Oracle® Hierarchical Storage Manager and StorageTek QFS Software

インストールおよび構成ガイド

リリース 6.0

E56768-02

2015 年 3 月

ORACLE

Oracle® Hierarchical Storage Manager and StorageTek QFS Software
インストールおよび構成ガイド

E56768-02

目次

はじめに

このドキュメントでは、Oracle HSM (旧 StorageTek Storage Archive Manager) and Oracle StorageTek QFS Software を使用してファイルシステムおよびアーカイブソリューションをインストールおよび構成するシステム管理者、ストレージおよびネットワーク管理者、およびサービスエンジニアに必要なものについて説明します。

ドキュメントのアクセシビリティー

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト (http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc) を参照してください。

Oracle Support へのアクセス
サポートをご契約のお客様には、My Oracle Support を通して電子支援サービスを提供しています。詳細情報は (http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info) か、聴覚に障害のあるお客様は (hhttp://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs) を参照してください。

このドキュメントを使用するための前提条件

このドキュメントでは、読者が Oracle Solaris オペレーティングシステムの管理、ディスクとテープストレージシステム、およびローカルエリアネットワークとストレージエリアネットワークに習熟していることを前提として説明します。Solaris のドキュメントとマニュアルページ、およびストレージハードウェアのドキュメントを参照し、関連するタスク、コマンド、および手順に関する情報を確認してください。

表記規則

このドキュメントでは、次の表記規則が使用されています。

- *イタリック*は、ドキュメントのタイトルおよび強調を表します。
- 太字はグラフィカルユーザーインタフェースの要素を表します。
- *等幅*は、コマンド、端末ウィンドウに表示されるテキスト、および構成ファイル、シェルスクリプト、ソースコードのファイル内容を表します。
- **等幅太字** は、ユーザー入力、コマンド行出力内の重要な変更、端末表示またはファイル内容を表します。ファイルまたは表示上で特に関連性の高い部分を強調する場合にも使用されます。

- ***等幅太字斜体***は、端末表示またはファイル上の変数の入力または出力を表します。
- *等幅斜体*は、端末表示またはファイル上のその他の変数を表します。
- `...` (3 点省略記号) は、例とは関係のないため、簡潔性および明確性を高めるために省略されたファイル内容やコマンド出力を表しています。
- `/` (バックスラッシュ) が例内の行の末尾で使用されている場合、それは改行を回避するためで、コマンドが次の行に続くことを表します。
- `[ - ]` (ハイフンで区切られた値を囲む大括弧) は値の範囲を区切ります。
- `[ ]` (大括弧) がコマンド構文の説明で使用されている場合は、オプションのパラメータであることを表します。
- `root@solaris:~#` および `[hostname]:root@solaris:~#` は、Solaris コマンドのシェルプロンプトを表します。
- `[root@linux ~]#` は、Linux コマンドシェルプロンプトを表します。

入手可能ドキュメント

『*Oracle Hierarchical Storage Manager and StorageTek QFS Software インストールおよび構成ガイド*』は、複数巻で構成されている*Oracle HSM お客様向けドキュメントライブラリ*の一部です (`docs.oracle.com/en/storage` から入手可能)。

Oracle Solaris オペレーティングシステムのドキュメントは、`http://docs.oracle.com/en/operating-systems/` で入手できます。

1

Oracle HSM ソリューションの配備

Oracle Hierarchical Storage Manager and StorageTek QFS Software (Oracle HSM) の配備は、基本的に非常に単純なプロセスです。ソフトウェアパッケージをインストールし、いくつかの構成ファイルを編集し、いくつかのコマンドを実行したあと、新しいファイルシステムをマウントして使用します。ただし、Oracle HSM には広範なオプションやチューニングパラメータが用意されています。これらの追加機能を使用すると、ほとんどの特殊なニーズに対応することができます。ただし、不要な機能によって配備が複雑化するため、結果として得られるソリューションの満足度が低下する原因になります。

したがってこのドキュメントでは、詳細なソリューション要件に厳密に従った Oracle HSM 配備の手順について説明しています。基本的な QFS および Oracle HSM ファイルシステムの作業、インストール、および構成から開始します。これらのファイルシステムは、単独ですべての要件を満たすか、より特殊なソリューションのための基礎となっています。基本機能の配備が完了したら、次に特定の環境や特殊なビジネスニーズをサポートする追加機能を構成する手順に進みます。次の中核となるタスクを実行します。

- 要件に合わせてハードウェアとオペレーティングシステムソフトウェアを構成します。
- 必要となる基本 QFS ファイルシステムまたは Oracle HSM ファイルシステム、あるいはその両方を構成し、可能なかぎりデフォルトを使用します。
- 要件を満たすための追加の Oracle HSM 機能をすべて構成します。
- 完成した構成のバックアップを取り、テスト用や本番用として引き渡します。

計画プロセスと配備プロセス全体を通して、QFS および Oracle HSM の設計は、パフォーマンスの最適化、データ保護、およびアーカイブ処理の複雑性が単純な UNIX ファイルシステムインタフェースの背後に隠されています。ユーザー、アプリケーション、およびほとんどの場合、管理者は、ディスクアレイとテープライブラリの混合に実装され、完全に最適化された Oracle HSM アーカイブ処理システムを、単一のローカルディスク上にある通常の UFS ファイルシステムと同様に取り扱うことができるべきです。Oracle HSM ソフトウェアは、インストールして構成したあとは、データとストレージリソースを可能なかぎりもっとも効率的かつ信頼できる方法で自動的に管理するはずであり、人間の介入は最小限で済みます。したがって、ファイルシステムやストレージリソースを極端に複雑に実装したり、過度に細かく管理する

と、Oracle HSM 配備の重要な目標が損なわれ、パフォーマンスや容量の利用率、データ保護などに悪影響を及ぼすことがあります。

この概要の残りの部分では、QFS ファイルシステムおよび Oracle HSM アーカイブファイルシステムについて簡単に説明します。この情報を大まかに把握するだけでも、後続の各構成手順の目的を理解しやすくなります。

1.1. QFS ファイルシステム

QFS ファイルシステムを使用すると、完全に最適化されたカスタムストレージソリューションと標準の UNIX インタフェースとを組み合わせることができます。これらは内部的に、厳格かつ高度に特殊化されることの多いパフォーマンス要件を満たせるように、物理ストレージデバイスを管理します。しかし、これ自体は外部ユーザー、アプリケーション、およびオペレーティングシステムに対しては通常の UNIX ファイルシステムとして提示されます。したがって、特殊なパフォーマンス要件やデータ保護要件を複雑な各種ストレージハードウェアを使用して実現すると同時に、既存のアプリケーションやビジネスプロセスとの単純な統合を保証することができます。

QFS ファイルシステムは、中核となる QFS ボリュームマネージャーを使用して独自の物理ストレージを管理します。QFS software は、標準インタフェースと完全な互換性のある高度に最適化された論理デバイスに標準の物理ストレージデバイスを編成します。その特殊な機能やカスタマイズはこのソフトウェアによってカプセル化されるため、オペレーティングシステムやアプリケーションからは見えないままになります。後者に対し、QFS software は、標準の Solaris デバイスドライバ経由で単一ディスクのように入出力要求を処理する論理ファミリセットデバイスを提供します。この標準準拠とチューニングのしやすさの組み合わせこそが、QFS とその他の UNIX ファイルシステムとの違いです。

このセクションの残りの部分では、まず QFS のデフォルトおよび入出力パフォーマンスのチューニングの目的について簡単に説明したあと、作成するファイルシステムの入出力動作の制御を可能にする各コアツールについて説明します。

- 柔軟な ディスク割り当て単位と論理デバイスタイプを使用した、読み取り/書き込みのサイズとファイルのサイズの一致。
- ストライプ化とラウンドロビン式の ファイル割り当て方式による、ファイル入出力とデバイスとの相互作用の方法の制御。
- 完全に構成可能な ストレージ割り当てと組み込みボリューム管理による、ファイルシステムとベースとなる物理ストレージとの相互作用の方法の制御。
- 一般用途と高パフォーマンスの ファイルシステムタイプの選択肢による、データとメタデータの入出力をそれぞれ異なるデバイス上で実行させるかどうかの選択。

1.1.1. QFS のデフォルトおよび入出力パフォーマンスのチューニングの目的

ディスクの入出力 (I/O) には、CPU 負荷の高いオペレーティングシステム要求と、時間のかかる機械的なプロセスが伴います。したがって、入出力パフォーマンスチューニングは、ある特定のデータ量を転送する際の入出力関連のシステムオーバーヘッドを最小化しつつ、機械的な作業量を必要最小限に抑えることに重点が置かれます。つまり、データ転送あたりの個別入出力の数 (CPU が実行する操作の数) を減らすとともに、各入出力の間のシーキング (読み取り/書き込みヘッドの再配置) を最小限に抑えます。したがって、入出力チューニングの基本目的は次のようになります。

- 平均ファイルサイズで割り切れるブロックの単位での、データの読み取りおよび書き込み。
- 大きなデータブロックの読み取りおよび書き込み。
- ベースとなるメディアの 512 バイトセクター境界に整列する単位でのブロックの書き込み。これにより、ディスクコントローラが新しいデータを書き込む前に、既存データの読み取りと変更が不要になります。
- 小さい入出力のキャッシュ内でのキュー、および大きい単位にまとめられた入出力のディスクへの書き込み。

Oracle HSM のデフォルト設定を使用すると、大部分の汎用ファイルシステムで典型的なさまざまなアプリケーションや使用パターンで、全体的に最良のパフォーマンスが得られます。ただし、必要であればアプリケーションから生成される入出力のタイプに応じてデフォルト動作を調整できます。連続する読み取りまたは書き込みの最小サイズを指定できます。ファイルがデバイスに格納される方法を最適化できます。一般的な使用または高いパフォーマンスのために最適化されたファイルシステムを選択できます。

1.1.2. ディスク割り当て単位と論理デバイスタイプ

ファイルシステムは、均一サイズのブロックの単位でディスクストレージを割り当てます。このサイズ、つまりディスク割り当て単位 (DAU) によって、書き込まれるデータ量に関係なく、各入出力操作で消費される連続した領域の最小量と、ある特定のサイズのファイル転送に最低限必要な入出力操作の回数が決まります。ブロックサイズがファイルの平均サイズに比べて大きすぎると、ディスク領域が無駄になります。ブロックサイズが小さすぎると、各ファイル転送でより多くの入出力操作が必要となり、パフォーマンスが低下します。したがって、入出力のパフォーマンスやストレージの効率が最大になるのは、ファイルサイズが基本ブロックサイズの偶数倍になっている場合です。

このため、QFS software は構成可能な DAU サイズの範囲をサポートしています。QFS ファイルシステムを作成する場合は、まずアクセスおよび格納する必要のあるデータファイルの

平均サイズを確認します。次に平均ファイルサイズをもっとも均等に分割する DAU を指定します。

まず、使用するデータにもっとも適した QFS デバイスタイプを選択します。次の 3 つのタイプがあります。

- *md* デバイス
- *mr* デバイス
- *gXXX* ストライプ化グループデバイス (ここで、*XXX* は *[0-127]* の範囲の整数)。

ファイルシステムに主に小さいファイルが含まれる、または小さいファイルと大きいファイルが混在する場合、通常 *md* デバイスが最適な選択肢です。*md* デバイスタイプでは柔軟なデュアル割り当てスキームが使用されます。ファイルシステムではファイルをデバイスに書き込む際に、最初の 8 つの書き込みに *4*K バイトの小さい DAU を使用します。その後、ユーザーが選択した大きな DAU (*16*、*32*、または *64*K バイト) を使用して残りのデータをすべて書き込みます。したがって、小さいファイルは適切な小さいブロックで書き込まれ、大きいファイルはその平均サイズに応じて調整された大きいブロックで書き込まれます。

ファイルシステムに主に大きなファイルが含まれる場合、均一サイズのファイルが含まれる場合、または大きなファイルと均一サイズのファイルが含まれる場合、*mr* デバイスが最適な選択肢となる場合があります。*mr* デバイスタイプでは、*[8-65528]*K バイトの範囲内で 8K バイトの増分値で調整可能な DAU が使用されます。ファイルは大きな均一ブロックで書き込まれ、平均的なファイルサイズにきわめて近くなるため、読み取り/変更/書き込みのオーバーヘッドが最小限に抑えられ、パフォーマンスが最大化されます。

ストライプ化グループは最大 128 個のデバイスをまとめたものであり、単一の論理デバイスとして扱われます。ストライプ化グループでは *mr* デバイスタイプの場合と同じく、*[8-65528]*K バイトの範囲内で 8K バイトの増分値で調整可能な DAU が使用されます。ファイルシステムは、ディスクごとに 1 DAU ずつ、ストライプ化グループのメンバーにデータを並列的に書き込みます。したがって、書き込み合計量が非常に大きくなる可能性があります。このため、きわめて大きなデータファイルを処理する必要のあるアプリケーションで、ストライプ化グループが役に立つ可能性があります。

1.1.3. ファイル割り当て方式

デフォルトでは、非共有の QFS ファイルシステムでストライプ化割り当てが使用され、共有ファイルシステムではラウンドロビン式割り当てが使用されます。ただし、必要であれば割り当てを変更できます。どちらの方法も状況によっては利点があります。

1.1.3.1. ストライプ化割り当て

ストライプ化割り当てが指定された場合、ファイルシステムは、使用可能なすべてのデバイス上で並列的に領域を割り当てます。ファイルシステムはデータファイルを複数のセグメントに分割し、各デバイスにセグメントを 1 つずつ書き込みます。各セグメントのサイズは、*ストライプ幅* (各デバイスに書き込まれる DAU の数) × ファミリセット内のデバイス数によって決まります。デバイスは *md* ディスクデバイス、*mr* ディスクデバイス、ストライプ化グループのいずれかになります。

一般に、ストライプ化によってパフォーマンスが向上しますが、これは、ファイルシステムが複数のファイルセグメントを順番にではなく同時に読み取るためです。複数の入出力操作が異なるデバイス上で並列して発生するため、デバイス当たりのシークオーバーヘッドが低減します。

ただし、ストライプ化割り当てでは、複数のファイルへの書き込みが一度に行われると、非常に多くのシーキングが発生する可能性があります。過剰なシーキングが発生するとパフォーマンスが大幅に低下する可能性があるため、複数ファイルへの同時入出力が予想される場合には、ラウンドロビン式割り当てを検討してください。

1.1.3.2. ラウンドロビン式割り当て

ラウンドロビン方式割り当てが指定された場合、ファイルシステムはストレージ領域を一度に 1 ファイル、一度に 1 デバイスずつ順次割り当てます。ファイルシステムは、使用可能な領域がある最初のデバイスにファイルを書き込みます。ファイルがデバイス上の残りの領域よりも大きい場合、ファイルシステムはその余剰分を、使用可能な領域のある次のデバイスに書き込みます。ファイルシステムは後続のファイルごとに、次に使用可能なデバイスへと移動し、このプロセスを繰り返します。ファイルシステムは、使用可能な最後のデバイスを使い終わると、また最初のデバイスに戻ります。デバイスは *md* ディスクデバイス、*mr* ディスクデバイス、ストライプ化グループのいずれかになります。

アプリケーションが複数ファイルへの入出力を同時に実行する場合、ラウンドロビン式割り当てを使用するとパフォーマンスが改善される可能性があります。またこれは、QFS 共有ファイルシステムのデフォルトでもあります (共有ファイルシステムの詳細については、「Oracle HSM ソフトウェアを使用した複数のホストからのファイルシステムへのアクセス」および *mount_samfs* のマニュアルページを参照)。

1.1.4. ストレージ割り当てと組み込みボリューム管理

1 つのデバイスや 1 つのデバイスの一部のみを対象とする UNIX ファイルシステムとは異なり、QFS ファイルシステムは独自にボリュームを管理します。各ファイルシステムは、物理

ストレージを提供する各デバイス間の関係を内部的に処理したあと、そのストレージを単一のファミリセットとしてオペレーティングシステムに提供します。入出力要求はほかの UNIX ファイルシステムと同じく、標準の Solaris デバイスドライバインタフェース経由で行われます。

1.1.5. ファイルシステムタイプ

QFS ファイルシステムには次の 2 つのタイプがあります。それぞれには独自の利点があります。

1.1.5.1. 汎用の ms ファイルシステム

QFS *ms* ファイルシステムは実装がもっとも簡単であり、一般用途の大部分に適しています。これらは同じ二重割り当て (*md* ディスクデバイス) のファイルデータとともにファイルシステムメタデータを格納します。この方法は、ハードウェア構成を簡略化し、ほとんどのニーズを満たします。

1.1.5.2. 高パフォーマンスの ma ファイルシステム

{ENT:QFS }*ma* ファイルシステムは、要求の多いアプリケーションでデータ転送速度を改善できます。これらのファイルシステムは、メタデータおよびデータを専用のデバイスに別個に格納します。メタデータは *mm* デバイスに保持され、データは一連の *md* ディスクデバイス、*mr* ディスクデバイス、またはストライプ化グループに保持されます。その結果、メタデータの更新はユーザーおよびアプリケーションの入出力と競合せず、デバイス構成は 2 つの異なる種類の入出力ワークロードに対応する必要はありません。たとえば、RAID-10 ミラー化ディスクにはメタデータを配置して冗長性や読み取り速度を改善させ、領域の使用効率の高い RAID-5 ディスクアレイにデータを保持できます。

1.2. Oracle HSM アーカイブファイルシステム

アーカイブファイルシステムは、1 つ以上の QFS *ma* または *ms* タイプのファイルシステムをアーカイブストレージおよび Oracle Hierarchical Storage Manager software と組み合わせます。Oracle HSM software は、ファイルシステムのディスクキャッシュからセカンダリディスクストレージまたはリムーバブルメディア、あるいはその両方にファイルをコピーします。ファイルシステムの中核となる部分としてコピーを管理します。したがって、ファイルシステムは、継続的なデータ保護機能、およびディスクまたはソリッドステートメディアに格納するとコストがかかりすぎる可能性がある大規模ファイルを柔軟かつ効率的に格納する機能の両方を提供します。

Oracle HSM ファイルシステムを適切に構成すると、別のバックアップアプリケーションを使用しなくても継続的なデータ保護を実現できます。ソフトウェアは、ファイルが作成または変

更されると、ユーザー定義ポリシーに指定されているとおりに、ファイルデータを自動的にコピーします。ローカルとリモートの両方のリソースを使用して、ディスクとテープメディアを含めて最大 4 つのコピーを維持できます。ファイルシステムのメタデータには、ファイルとそのすべてのコピーの場所が記録されます。ソフトウェアには、コピーをすばやく検索するための各種ツールが用意されています。このため、ファイルが失われたり破損したりしても、アーカイブからすぐに回復できます。しかも、バックアップコピーは標準の POSIX 準拠 *tar* (テープアーカイブ) 形式で保存されるため、Oracle HSM software を使用できない場合でもデータを回復できます。Oracle HSM は、入出力エラーを動的に検出して回復することによって、ファイルシステムメタデータの整合性を常に維持します。このため、時間のかかる整合性チェックを実行せずにファイルシステムのバックアップを実行できます。これは、何十万個ものファイルやペタバイトのデータが格納されている場合の重要な考慮事項になります。ファイルシステムのメタデータが別のデバイスに格納されており、データストレージディスクのみが問題になっている場合は、交換ディスク上でファイルシステムが構成された時点で回復が完了します。故障したディスク上にあったファイルがユーザーから要求された場合、Oracle HSM が自動的にそのバックアップコピーをテープから交換ディスクにステージングします。メタデータも失われていた場合、管理者は *samfsrestore* コマンドを使用して *samfsdump* バックアップファイルからそのメタデータを復元できます。メタデータの復元が完了すると、ユーザーからの要求に応じて再度テープからファイルを復元できるようになります。ディスクへのファイルの復元は要求に応じてのみ実行されるため、回復プロセスではネットワーク帯域幅を効率的に使用し、通常動作への影響も最小限に抑えられます。

Oracle HSM ファイルシステムはこのように、高パフォーマンスなプライマリディスクまたはソリッドステートメディア上のファイルと低コストでより高密度なセカンダリディスク、テープ、または光メディア上のファイルを同時に管理できる能力を備えているため、きわめて大きいファイルや使用頻度の低いファイルの経済的な格納には最適です。衛星画像や映像ファイルなど、順次アクセス方式の非常に大きなデータファイルは、磁気テープにのみ格納できます。ユーザーやアプリケーションがファイルにアクセスすると、選択されたファイル構成に応じて、ファイルシステムは自動的にファイルを元のディスクにステージングする、ファイルをテープからメモリー内に直接読み取ります。主に履歴や法令準拠を目的として保存されるレコードは、ファイル保存期間中のある時点で、ユーザーのアクセスパターンやコスト制約にもっとも適したメディアを使用して、階層的に格納できます。まず、ユーザーがときどきファイルにアクセスする場合は、低コストのセカンダリディスクデバイスにアーカイブできます。要求の減少に応じて、テープや光メディア上にのみコピーを維持します。しかし、法的証拠開示や規制プロセスへの対応などのためにデータが必要になった場合は、その場所に最初から存在していたように最小限の遅延で、必要な情報がファイルシステムによって自動的にプライマリディスクにステージングされます。法律および規制のために使用する場合、WORM に対応した Oracle HSM ファイルシステムを使用できます。WORM 対応のファイルシステムでは、デフォ

ルトおよびカスタマイズ可能なファイル保持期間、データとパスの不変性、および WORM 設定のサブディレクトリの継承がサポートされます。手動または自動、あるいはその両方のメディア検証を使用すると、長期間のデータの整合性をモニターできます。

アーカイブファイルシステムを管理および保守する基本的な Oracle HSM プロセスとして、次の 4 つがあります。

- アーカイブ処理
- ステージング
- 解放処理
- リサイクル処理.

1.2.1. アーカイブ処理

アーカイブ処理では、アクティブファイルのコピーの格納場所として予約されたアーカイブメディアにファイルシステム内のファイルをコピーします。アーカイブメディアには、磁気テープカートリッジなどのリムーバブルメディアボリュームや、磁気ディスクまたはソリッドステートストレージデバイス上に存在している 1 つ以上のファイルシステムを含めることができます。アーカイブファイルコピーは、アクティブファイルのバックアップ冗長性または非アクティブファイルの長期保存、あるいはその両方の何らかの組み合わせを提供することができます。

Oracle HSM アーカイブファイルシステムでは、アクティブなオンラインファイル、アーカイブコピー、および関連ストレージリソースが、単一の論理リソースであるアーカイブセットを形成します。アーカイブファイルシステム内のどのアクティブファイルも、属しているアーカイブセットは 1 つだけです。各アーカイブセットには、各ファイルの最大 4 つのアーカイブコピーと、そのアーカイブセットのアーカイブ処理を制御するポリシーを含めることができます。

アーカイブ処理は UNIX デーモン (サービス) の *sam-archiverd* によって管理されます。このデーモンは、アーカイブアクティビティーをスケジュールし、必要なタスクを実行するプロセス (*archiver*、*sam-arfind*、および *sam-arcopy*) を呼び出します。

archiver プロセスは、編集可能な構成ファイル *archiver.cmd* 内のアーカイブポリシーを読み取り、残りのアーカイブ処理を指示に従って設定します。このファイル内のディレクティブは、アーカイブ処理の全般的な動作を制御し、アーカイブセットをファイルシステム別に定義し、作成するコピーの数やそれぞれが使用するアーカイブメディアを指定します。

次に、*sam-archiverd* デーモンは現在マウントされているファイルシステムごとに *sam-arfind* プロセスを起動します。*sam-arfind* プロセスは割り当てられたファイルシステムをスキャンし、新しいファイル、変更済みのファイル、名前が変更されたファイル、および再アーカイブまたはアーカイブ解除する必要のあるファイルの有無を調べます。このプロセス

はデフォルトでファイルやディレクトリへの変更を継続的にスキャンするため、それによって全体のパフォーマンスが最大になります。ただし、たとえば古い StorageTek Storage Archive Manager 実装との互換性を維持する必要がある場合は、*archiver.cmd* ファイル内のアーカイブセット規則を編集し、いずれかの方法のうち 1 つを使用するスキャンをスケジュールすることもできます (詳細は、*sam-archiverd* のマニュアルページを参照)。

sam-arfind は、候補ファイルを特定し終わると、そのファイルのアーカイブポリシーを定義したアーカイブセットを特定します。*sam-arfind* プロセスは、ファイルの属性と各アーカイブセットに定義された選択条件を比較してアーカイブセットを特定します。これらの条件には次の 1 つ以上のファイル属性が含まれます。

- ファイルへのディレクトリパス、およびオプションで、1 つ以上の候補ファイルの名前に一致する正規表現
- 1 つ以上の候補ファイルの所有者に一致する、指定されたユーザー名
- ファイルに関連付けられたグループに一致する、指定されたグループ名
- 候補ファイルのサイズに等しいかそれより小さい、指定された最小ファイルサイズ
- 候補ファイルのサイズと等しいかそれより大きい、指定された最大ファイルサイズ。

正しいアーカイブセットと対応するアーカイブパラメータが特定されると、*sam-arfind* は、ファイルの*アーカイブ経過時間*が、アーカイブセットに指定されたしきい値以上になっているかどうかをチェックします。ファイルのアーカイブ経過時間とは、ファイルの作成、最後の変更 (デフォルト)、あるいは最後のアクセス以降に経過した秒数のことです。アーカイブ経過時間がポリシーに指定された経過時間の条件を満たす場合、*sam-arfind* はそのファイルをアーカイブセットの*アーカイブ要求*キューに追加し、優先順位を割り当てます。優先順位は、アーカイブセットに指定された規則、およびすでに存在しているアーカイブコピーの数、ファイルのサイズ、すべての未処理のオペレータ要求、アーカイブコピーの作成に依存するその他のすべての操作などの要因にも基づきます。

sam-arfind は、アーカイブが必要なファイルを特定して優先順位を付け、それらを各アーカイブセットのアーカイブ要求へ追加したあと、*sam-archiverd* デーモンに要求を返します。デーモンは各アーカイブ要求を*合成*します。これによってデータファイルがデータファイル内に配置され、メディアが効率的に活用され、ファイルがリムーバブルメディアに効率的に書き込まれ、あとでリムーバブルメディアから読み取られるようになります。デーモンは、*archiver.cmd* ファイルに設定したファイルソートパラメータやメディア制限をすべて遵守しますが (詳細は *archiver.cmd* のマニュアルページを参照)、通常、ソフトウェアが自由にメディアを選択できないように制限するとパフォーマンスが低下し、メディアの使用率も低下します。アーカイブファイルの組み立てが完了すると、*sam-archiverd* はアーカイブ要求の優先順位を決定し、コピープロセスが最小回数のマウント操作で最大数のファイルを転送

できるようにします (詳細は、*sam-archiverd* マニュアルページのスケジューリングに関するセクションを参照)。次に *sam-archiverd* は、コピー操作のスケジューリングを行い、どの時点でも、アーカイブセットポリシーやロボットライブラリで許可される最大ドライブ数を超えるドライブを必要としないようにします。

アーカイブ要求のスケジューリングが完了すると、*sam-archiverd* はスケジュールされたアーカイブ要求とドライブごとに、*sam-arcopy* プロセスのインスタンスを呼び出します。*sam-arcopy* インスタンスは、データファイルをアーカイブメディア上のアーカイブファイルにコピーし、アーカイブファイルシステムのメタデータを更新して新しいコピーの存在を反映させ、アーカイブログを更新します。

sam-arcopy プロセスが終了すると、*sam-archiverd* デーモンはアーカイブ要求をチェックし、キャッシュディスクからの読み取りエラー、リムーバブルメディアボリュームへの書き込みエラー、およびオープン、変更、または削除されたファイルに起因するエラーや省略の有無を調べます。アーカイブされなかったファイルがあれば、*sam-archiverd* が再度アーカイブ要求を合成します。

sam-arfind および *sam-arcopy* プロセスは、*syslog* 機能と *archiver.sh* を使用して、アーカイブアクティビティー、警告、および情報メッセージの継続的なレコードを作成できます。結果のアーカイバログには、アーカイブされたすべてのファイルのすべてのコピーに関する場所や処理の詳細な記録など、貴重な診断情報および履歴情報が含まれます。そのため、たとえば障害回復中に、それ以外の方法では回復不能な消失したデータファイルを回復するためにアーカイブログを使用できます (詳細は、お客様向けドキュメントライブラリの『*Oracle Hierarchical Storage Manager and StorageTek QFS Software ファイルシステム回復ガイド*』を参照)。ファイルシステム管理者は、アーカイバロギングを有効にし、*archiver.cmd* ファイルで *logfile=* ディレクティブを使用してログファイルを定義します。ログファイルの詳細については、*archiver.cmd* のマニュアルページを参照してください。

1.2.2. ステージング

ステージングプロセスは、ファイルデータをアーカイブストレージから元のプライマリディスクキャッシュ内にコピーして戻します。アプリケーションが *オフラインファイル* (プライマリストレージ内で現在使用できないファイル) にアクセスしようとすると、アーカイブコピーが自動的にステージングされます (プライマリディスクにコピーされます)。次に、ステージングの直後に読み取り操作で追跡を行うため、アプリケーションは完全なデータがディスクに書き込まれる前でも、すぐにファイルにアクセスできるようになります。メディアエラーが発生した場合や特定のメディアボリュームが使用不可能の場合、ステージング処理では最初に使用可能なデバイスを使用して、次に使用可能なアーカイブコピーがあればそれを自動的にロードしま

す。したがって、ステージングにより、ユーザーやアプリケーションに対してアーカイブストレージを透過的にします。すべてのファイルが常時、ディスク上で使用可能なように見えます。

大部分のファイルシステムでは、デフォルトのステージング動作で十分です。ただし、構成ファイル (*/etc/opt/SUNWsamfs/stager.cmd*) でディレクティブを挿入または変更することによりデフォルトを変更でき、ディレクトリごとまたはファイルごとにコマンド行からこれらのディレクティブをオーバーライドできます。たとえば、大規模ファイル内の小さなレコードにアクセスする場合、ファイルのステージングを行わない、アーカイブメディア内での直接データアクセスを選択することもできます。あるいは、*結合*ステージング機能を使用して、関連するファイルのいずれかがステージングされるたびに、それら関連ファイルのグループをステージングすることもできます。詳細は、*stage* および *stager.cmd* のマニュアルページを参照してください。

1.2.3. 解放処理

解放処理では、すでにアーカイブされたファイルのオンラインコピーのうち、現在使用されていないものを削除してプライマリディスクキャッシュ領域を解放します。ディスクアーカイブやテープボリュームなどのアーカイブメディアにコピーされたファイルは、アプリケーションからのアクセスがあった場合にステージングできます。したがって、ほかのファイル用の領域が必要な場合にそのファイルをディスクキャッシュ内に保持する必要はありません。ファイルシステムのサイズ増加に応じてプライマリストレージ容量が増えない場合でも、ディスクキャッシュからの不要なコピーの削除による解放では、新たに作成されたファイルや使用頻度の高いファイルに対して常にプライマリキャッシュストレージが使用可能であることを保証します。

解放処理は、キャッシュ使用率が*高位境界値*を超える場合、および*低位境界値*を上回ったままになっている場合に自動的に実行されます。この 2 つの構成可能しきい値は、アーカイブファイルシステムのマウント時に設定します。高位境界値が十分な空き領域が常に使用可能であることを保証するのに対し、低位境界値は、キャッシュ内で使用可能なファイルが常に妥当な数に保たれ、メディアのマウント操作が必要最小限に抑えられることを保証します。一般的な値は、最高値が 80%、最低値が 70% です。

大部分のファイルシステムでは、デフォルトの動作を使用した境界値による解放で十分です。ただし、構成ファイル (*/etc/opt/SUNWsamfs/releaser.cmd*) 内のディレクティブを変更したり追加したりしてデフォルトを変更でき、ディレクトリごとまたはファイルごとにコマンド行からそれらのデフォルトをオーバーライドできます。たとえば、順次アクセスされる大規模ファイルを部分的に解放すると、アプリケーションがディスク上に常に保持されているファイルの一部の読み取りを開始している間に、残りの部分をアーカイブメディアからステージングさせることができます。詳細は、*release* および *releaser.cmd* のマニュアルページを参照してください。

1.2.4. リサイクル処理

リサイクル処理では、使用されなくなったアーカイブコピーを削除してアーカイブメディア上の領域を解放します。ユーザーがファイルを変更すると、そのファイルの古いバージョンに関連付けられたアーカイブコピーは最終的に期限切れになります。リサイクラは、期限切れアーカイブコピーの割合がもっとも高いメディアボリュームを特定します。期限切れのファイルがアーカイブディスクボリュームに格納されている場合、リサイクラプロセスがそれを削除します。ファイルがリムーバブルメディア (テープボリュームなど) に存在する場合、リサイクラは、対象ボリュームに残っている期限切れではないコピーをほかのメディアに再アーカイブします。次に編集可能なスクリプト (*/etc/opt/SUNWsamfs/scripts/recycler.sh*) を呼び出し、リサイクルされたボリュームの再ラベル付け、ライブラリからのエクスポート、またはその他のユーザー定義アクションの実行を行います。

デフォルトでは、リサイクル処理は自動的に実行されません。Solaris *crontab* ファイルを構成して、都合のよいときにそれを実行できます。または、*/opt/SUNWsamfs/sbin/sam-recycler* コマンドを使用してコマンド行から必要に応じて実行することもできます。デフォルトのリサイクルパラメータを変更するには、ファイル */etc/opt/SUNWsamfs/archiver.cmd* を編集するか、あるいは別の */etc/opt/SUNWsamfs/recycler.cmd* ファイルを作成します。詳細は、対応するマニュアルページを参照してください。

2

ホストシステムの構成

インストールおよび構成に進む前に、Oracle Hierarchical Storage Manager and StorageTek QFS Software 用にホストオペレーティングシステムを構成します。この章では、次のトピックの概要について説明します。

- Oracle HSM 用の Oracle Solaris の構成
- Oracle HSM クライアント用の Linux の構成

2.1. Oracle HSM 用の Oracle Solaris の構成

Oracle HSM software および QFS ファイルシステムで使用するように Solaris ホストを構成するには、次のタスクを実行します。

- 最新オペレーティングシステムの更新のインストール
- 予測したファイルシステム入出力に合わせた Solaris システムおよびドライバパラメータの調整

2.1.1. 最新オペレーティングシステムの更新のインストール

可能なかぎり、常に Solaris オペレーティングシステムに対応した最新のパッチおよび更新をインストールします。Oracle Hierarchical Storage Manager and StorageTek QFS Software リリース 6.0 で使用可能な最新機能を使用する必要がある場合は、すべての Solaris ホスト上に Oracle Solaris 11 オペレーティングシステムソフトウェアをインストールする必要があります。ソフトウェアで使用するために推奨される最小のオペレーティングシステムリリースに関する詳細は、リリースノートおよび *`support.oracle.com`* を参照してください。

選択したバージョンの Solaris のインストール手順および更新手順については、対応するお客様向けドキュメントライブラリのインストールと管理のドキュメント、Oracle Technical Network (OTN)、および *`support.oracle.com`* のナレッジベースを参照してください。Image Packaging System (IPS) をはじめて使用する場合は、次の OTN 記事が特に役立つことがあります。

- 「*Introducing the Basics of Image Packaging System (IPS) on Oracle Solaris 11*」(Glynn Foster 著、2011 年 11 月)

- 「*How to Update Oracle Solaris 11 Systems From Oracle Support Repositories*」(Glynn Foster 著、2012 年 3 月)
- 「*More Tips for Updating Your Oracle Solaris 11 System from the Oracle Support Repository*」(Peter Dennis 著、2012 年 5 月)。

2.1.2. 予測したファイルシステム入出力に合わせた Solaris システムおよびドライバパラメータの調整

システム全体のエンドツーエンド入出力 (I/O) のパフォーマンスは、オペレーティングシステム、ドライバ、ファイルシステム、およびアプリケーションによって、不要に断片化および再キャッシュする必要のない単位でデータが転送されるときに最大になります。そのため、使用中のアプリケーションおよびファイルシステムで予測される最大のデータ転送に対応するように Solaris を設定してください。次のように進めます。

1. Oracle HSM ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. */etc/system* ファイルのバックアップコピーを作成してから、テキストエディタで */etc/system* を開きます。

   次の例では、*vi* エディタを使用します。

```
root@solaris:~# cp /etc/system /etc/system.backup
root@solaris:~# vi /etc/system
*ident  "%Z%%M% %I%     %E% SMI" /* SVR4 1.5 */
* SYSTEM SPECIFICATION FILE
...
```

3. */etc/system* ファイルで、*maxphys* (単一の単位としてドライバが処理できる最大の物理入出力要求のサイズ) を、使用中のアプリケーションおよびファイルシステムで行われる最大のデータ転送と等しい値に設定します。*set maxphys = 0xvalue* 形式の 1 行を入力します。ここで *value* は、バイト数を表す 16 進数です。その後、ファイルを保存してエディタを閉じます。

   maxphys を上回る要求は、ドライバによって *maxphys* のサイズのフラグメントに分割されます。デフォルト値は、オペレーティングシステムのリリースによって異なることがありますが、通常は 128K バイト前後です。この例では、*maxphys* を *0x800000* (8,388,608 バイトまたは 8M バイト) に設定します。

```
root@solaris:~# vi /etc/system
*ident  "%Z%%M% %I%     %E% SMI" /* SVR4 1.5 */
* SYSTEM SPECIFICATION FILE
...
set maxphys = 0x800000
:wq
root@solaris:~#
```

4. テキストエディタで */kernel/drv/sd.conf* ファイルを開きます。

   この例では、*vi* エディタを使用します。

```
root@solaris:~# vi /kernel/drv/sd.conf
# Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
name="sd" class="scsi" target=0 lun=0;
name="sd" class="scsi" target=1 lun=0;
...
# Associate the driver with devid resolution.
ddi-devid-registrant=1;
```

5. */kernel/drv/sd.conf* ファイルで、*sd_max_xfer_size* (SCSI ディスク (*sd*) ドライバで処理可能な最大のデータ転送サイズ) を *maxphys* に設定されている値に設定します。*sd_max_xfer_size=0xvalue;* 形式の 1 行を入力します。ここで *value* は、バイト数を表す 16 進数です。ファイルを保存して、エディタを閉じます。

   デフォルトは *0x100000* (1048576 バイトまたは 1M バイト) です。この例では、コメントを追加し、*sd_max_xfer_size* を *0x800000* (8,388,608 バイトまたは 8M バイト) に設定します。

```
...
# Associate the driver with devid resolution.
ddi-devid-registrant=1;
# Set SCSI disk maximum transfer size
sd_max_xfer_size=0x800000;
:wq
root@solaris:~#
```

6. テキストエディタで */kernel/drv/ssd.conf* ファイルを開きます。

   次の例では、*vi* エディタを使用します。

```
root@solaris:~# vi /kernel/drv/ssd.conf
# Copyright 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
name="ssd" parent="sf" target=0;
name="ssd" parent="fp" target=0;
...
name="ssd" parent="ifp" target=127;
```

7. */kernel/drv/ssd.conf* ファイルで、*ssd_max_xfer_size* (ファイバチャネルディスク (*ssd*) ドライバで処理可能な最大のデータ転送サイズ) を *maxphys* に設定されている値に設定します。*ssd_max_xfer_size=0xvalue;* 形式の 1 行を入力します。ここで *value* は、バイト数を表す 16 進数です。その後、ファイルを保存してエディタを閉じます。

   デフォルトは *0x100000* (1048576 バイトまたは 1M バイト) です。この例では、コメントを追加し、*ssd_max_xfer_size* を *0x800000* (8,388,608 バイトまたは 8M バイト) に設定します。

```
...
name="ssd" parent="ifp" target=127;
# Set Fibre Channel disk maximum transfer size
ssd_max_xfer_size=0x800000;
:wq
root@solaris:~#
```

8. システムを再起動します。コマンド *init 6* を使用します。

```
root@solaris:~# init 6
```

9. 追加の Solaris ホストを含めるソリューションを準備する場合、すべての Solaris ホストが構成されるまで、「Oracle HSM 用の Oracle Solaris の構成」で指定されているタスクを繰り返します。
10. 1 つ以上の Linux クライアントを含めるソリューションを準備する場合、「Oracle HSM クライアント用の Linux の構成」に移動します。
11. それ以外の場合、3章「*ストレージホストおよびデバイスの構成*」に進みます。

2.2. Oracle HSM クライアント用の Linux の構成

Oracle HSM クライアントソフトウェアをインストールする前に、次のように Linux オペレーティングシステムを準備する必要があります。

- 互換性のないオペレーティングシステム機能の無効化
- 必要なカーネル開発およびユーティリティーパッケージのインストール

2.2.1. 互換性のないオペレーティングシステム機能の無効化

1. Oracle HSM クライアントホストに *root* としてログインします。

```
[root@linux ~]#
```

2. SELinux (Secure Linux) がインストールされている場合は、それを無効にします。テキストエディタでファイル */etc/selinux/config* を開き、*SELINUX* フラグを *disabled* に設定し、ファイルを保存し、エディタを閉じて、リブートします。

   Oracle HSM では、Oracle Linux および Red Hat Enterprise Linux 上でデフォルトで有効になっている SELinux がサポートされていません。この例では、*vi* エディタでファイルを開きます。

```
[root@linux ~]# vi /etc/selinux/config
# This file controls the state of SELinux on the system.
...
#SELINUX=enforcing
#SELINUX=permissive
SELINUX=disabled
SELINUXTYPE=targeted
:wq
[root@linux ~]# reboot
```

3. AppArmor がインストールされている場合は、使用中の Linux ディストリビューションに対応したドキュメントで推奨される手順を使用して無効にします。

   AppArmor は、SELinux の代替として使用されることがありますが、Oracle HSM では AppArmor がサポートされていません。

4. 次に、必要なカーネル開発およびユーティリティーパッケージのインストールを実行します。

2.2.2. 必要なカーネル開発およびユーティリティーパッケージのインストール

Oracle HSM クライアントソフトウェアをインストールする前に、いくつかの指定したユーティリティーパッケージとともに、Linux カーネル開発パッケージをインストールする必要があります。必要なパッケージを特定してインストールするには、次の手順を使用します。

1. Linux クライアントホストに *root* としてログインします。

   この例では、クライアントは Oracle Linux 上でホストされています。

```
[root@linux ~]#
```

2. クライアント上にインストールされているカーネルバージョンを特定します。コマンド *uname -r* を使用します。

   この例では、カーネルバージョンは *2.6.9-89.0.0.0.1.EL* です。

```
[root@linux ~]# uname -r
2.6.9-89.0.0.0.1.EL
[root@linux ~]#
```

3. カーネル開発キット *kernel-devel-kernel-version* をインストールします。ここで *kernel-version* は、前のステップで特定したバージョン文字列です。

   Oracle HSM クライアントをインストールするには、このパッケージに含まれる *Module.symvers* が必要です。この例では、パラメータ *-y install* を付けて Oracle Linux コマンド *yum* を使用します (*-y* を付けると、すべてのプロンプトに自動的に「はい」で回答します)。

```
[root@linux ~]# yum -y install / kernel-devel-2.6.9-89.0.0.0.1.EL.i686.rpm
[root@linux ~]#
```

4. Korn シェル *ksh* がインストールされているかどうかを確認します。インストールされていない場合は、インストールします。

   この例では、Oracle Linux コマンド *rpm -qa* の出力を *grep* コマンドにパイプして、文字列 *ksh* を検索します。コマンドで出力が返されず、*ksh* がインストールされていないことを示します。そのため、コマンド *yum install ksh* を使用してインストールします。

```
[root@linux ~]# rpm -qa | grep ksh
[root@linux ~]#
[root@linux ~]# yum install ksh
...
--> Running transaction check
---> Package ksh-20100621-19.e16.x86_64 set to be installed

================================================================================
Package            Arch          Version                  Repository      Size
================================================================================
Installing:
 ksh               i686          2.6.9-89.0.0.0.1.EL       updates        506 k
...
Installed:
  ksh-2.6.9-89.0.0.0.1.EL.i686
Complete!
[root@linux ~]#
```

5. *cpio* ユーティリティーがインストールされているかどうかを確認します。インストールされていない場合は、インストールします。

   この例では、Oracle Linux コマンド *rpm -qa* の出力を *grep* コマンドにパイプして、文字列 *cpio* を検索します。コマンドでバージョン情報が返され、*cpio* ユーティリティーがインストールされていることを示します。

```
[root@linux ~]#  rpm -qa | grep cpio
cpio-2.10-10.e16.x86_64
[root@linux ~]#
```

6. *find* ユーティリティーがインストールされているかどうかを確認します。インストールされていない場合は、インストールします。

   この例では、Oracle Linux コマンド *rpm -qa* の出力を *grep* コマンドにパイプして、文字列 *findutils* を検索します。コマンドでバージョン情報が返され、*findutils* パッケージがインストールされていることを示します。

```
[root@linux ~]#  rpm -qa | grep findutils
```

```
findutils-4.4.2-6.e16.x86_64
[root@linux ~]#
```

7. *gcc* コンパイラがインストールされているかどうかを確認します。インストールされていない場合は、インストールします。

   この例では、Oracle Linux コマンド *rpm -qa* の出力を *grep* コマンドにパイプして、文字列 *gcc* を検索します。コマンドでバージョン情報が返され、*gcc* コンパイラがインストールされていることを示します。

```
[root@linux ~]#  rpm -qa | grep gcc
gcc-4.4.7-3.e16.x86_64
libgcc-4.4.7-3.e16.x86_64
[root@linux ~]#
```

8. *make* ユーティリティーがインストールされているかどうかを確認します。インストールされていない場合は、インストールします。

   この例では、Oracle Linux コマンド *rpm -qa* の出力を *grep* コマンドにパイプして、文字列 *make* を検索します。コマンドでバージョン情報が返され、*make* ユーティリティーがインストールされていることを示します。

```
[root@linux ~]#  rpm -qa | grep make
make-4.4.7-3.e16.x86_64
libmake-3.81.20.e16.x86_64
[root@linux ~]#
```

9. *binutils* パッケージがインストールされているかどうかを確認します。インストールされていない場合は、インストールします。

   Oracle HSM インストールソフトウェアで Linux カーネルを構築する必要がある場合は、このパッケージに含まれる *nm* ユーティリティーが必要です。この例では、Oracle Linux コマンド *rpm -qa* の出力を *grep* コマンドにパイプして、文字列 *nm* を検索します。コマンドでバージョン情報が返され、*nm* ユーティリティーがインストールされていることを示します。

```
[root@linux ~]#  rpm -qa | grep nm
binutils-2.20.51.0.2-5.34.e16.x86_64
```

```
[root@linux ~]#
```

10. *rpmbuild* パッケージがインストールされているかどうかを確認します。インストールされていない場合は、インストールします。

    この例では、Oracle Linux コマンド *rpm -qa* の出力を *grep* コマンドにパイプして、文字列 *rpmbuild* を検索します。コマンドでバージョン情報が返され、*rpmbuild* パッケージがインストールされていることを示します。

```
[root@linux ~]#  rpm -qa | grep rpmbuild
rpm-build-4.8.0-37.el6.x86_64
[root@linux ~]#
```

11. *rpm* パッケージがインストールされているかどうかを確認します。インストールされていない場合は、インストールします。

    Oracle HSM インストールソフトウェアで Linux カーネルを構築する必要がある場合は、このパッケージに含まれる *rpm2cpio* ユーティリティーが必要です。この例では、Oracle Linux コマンド *rpm -qa* の出力を *grep* コマンドにパイプして、文字列 *rpm* を検索します。コマンドでバージョン情報が返され、ユーティリティーがインストールされていることを示します。

```
[root@linux ~]#  rpm -qa | grep rpm
rpm-4.8.0-27.e16.x86_64
rpm-libs-4.8.0-27.e16.x86_64
rpm-python-4.8.0-27.e16.x86_64
[root@linux ~]#
```

12. 追加の Linux クライアントを含めるソリューションを準備する場合、すべての Linux クライアントが構成されるまで、「Oracle HSM クライアント用の Linux の構成」で指定されているタスクを繰り返します。
13. 次に、3章「*ストレージホストおよびデバイスの構成*」に進みます。

3

ストレージホストおよびデバイスの構成

Oracle HSM のインストールおよび構成に進む前に、この章に概要を示すストレージの構成タスクを実行します。この章では、次のトピックの概要について説明します。

3.1. プライマリストレージの構成

Oracle HSM ファイルシステムでは、プライマリディスクまたはソリッドステートディスクデバイスに、アクティブに使用および変更されているファイルが格納されます。キャッシュ用にディスクまたはソリッドステートディスクデバイスを構成する際には、次のガイドラインに従います。

3.1.1. プライマリキャッシュ用のデバイスの構成

1. プライマリキャッシュの開始容量を見積もるには、フルになった場合にそれぞれのファイルシステムで保持するデータの容量を決定します。
2. ファイルシステムのメタデータを考慮に入れるには、この開始容量を 10% 増やします。
3. 高パフォーマンスの *ma* タイプのファイルシステムを準備する場合、*mm* メタデータデバイス用にハードウェアを構成します。*mm* メタデータデバイスごとに、ハードウェアで制御された 4 ディスクの RAID 10 (1+0) ボリュームグループ 1 つが理想的です。パフォーマンスを最大にするためには、ソリッドステートディスクデバイスの使用を検討してください。

   ストライプ化ミラー RAID 10 アレイは、Oracle HSM メタデータの格納に適していることを特徴としています。RAID 10 ストレージハードウェアは冗長性が高いため、クリティカルなメタデータが保護されます。その他のほとんどの RAID 構成よりもスループットが高く、待機時間が短くなります。

   一般に、専用コントローラハードウェアで制御されるアレイは、共有の汎用プロセッサ上で動作しているソフトウェアで制御されるアレイよりもパフォーマンスに優れています。

ソリッドステートデバイスは、その特性上、頻繁に更新され、頻繁に読み取られるメタデータを格納する際に特に役立ちます。

4. プライマリキャッシュストレージ用に外部ディスクアレイを使用している場合は、ファイルシステム構成内の *md* または *mr* デバイスごとに、3+1 または 4+1 RAID 5 ボリュームを構成します。ボリュームグループごとに 1 つの論理ボリューム (LUN) を構成します。

   特定の数のディスクに対して、3+1 および 4+1 RAID 5 ボリュームグループを小さくすると、ボリュームグループを大きくするよりも並列性が高くなるため、入出力 (I/O) のパフォーマンスが高くなります。入出力の観点からは、RAID 5 ボリュームグループ内の個々のディスクデバイスは独立して動作せず、各ボリュームグループが単一のデバイスと同様に動作します。したがって、特定の数のディスクを 3+1 および 4+1 ボリュームグループに分割すると、対応する構成を大きくするよりもデバイスの独立性が高くなり、並列性が高くなり、入出力の競合が少なくなります。

   RAID グループを小さくすると、ストレージに対するパリティーの比率が高くなるため、容量も少なくなります。しかし、大部分のユーザーにとって、これはパフォーマンスの向上で十二分に相殺されます。アーカイブファイルシステムでは、ディスクキャッシュの容量が多少削減されても、多くの場合は、アーカイブ内で使用可能な容量がほぼ無制限であるため完全に相殺されます。

   1 つのボリュームグループに複数の論理ボリューム (LUN) を構成すると、論理的に別々のボリュームでの入出力によって、一度に 1 回の入出力にしか対応しないリソースセットに対する競合が生じます。これにより、入出力関連のオーバーヘッドが増加し、スループットが減少します。

5. 次に、アーカイブストレージの構成を開始します。

3.2. アーカイブストレージの構成

次のタスクを実行します。

- SAN 接続デバイスのゾーニング
- アーカイブディスクストレージの構成
- アーカイブテープストレージの構成

3.2.1. SAN 接続デバイスのゾーニング

1. ドライブとホストバスアダプタ間の通信を許可するには、ストレージエリアネットワーク (SAN) をゾーニングします。

2. ホストが SAN 上のデバイスを表示できることを確認します。*-al* (*接続ポイントリスト*) および *-o show_SCSI_LUN* オプションを指定して、Solaris 構成管理コマンド *cfgadm* を入力します。ドライブポートの World Wide Name (WWN) に関する出力を確認します。

   出力の 1 列目には、ホストバスアダプタのコントローラ番号と WWN をコロンで区切ったもので構成される接続ポイント ID (*Ap_id*) が表示されます。*-o show_SCSI_LUN* オプションは、ノードが ADI インタフェースを介してメディアチェンジャーを制御するブリッジドライブである場合に、ノード上のすべての LUN を表示します。

```
root@solaris:~# cfgadm -al -o show_SCSI_LUN
Ap_Id     Type Receptacle Occupant   Condition
c2::500104f000937528   tape connected  configured   unknown
c3::50060160082006e2,0 tape connected  unconfigured unknown
```

3. *cfgadm -al -o show_SCSI_LUN* の出力にドライブの WWN が一覧表示されていない場合は、ドライブが表示されません。SAN の構成に何らかの問題があります。このため、SAN 接続およびゾーン構成を再確認してください。次に、前のステップを繰り返します。
4. *cfgadm -al* コマンドの出力でドライブが構成されていないと表示される場合、次に *-c* (*構成*) スイッチを使用して、コマンドを再度実行します。

   このコマンドは、*/dev/rmt* に必要なデバイスファイルを構築します。

```
root@solaris:~# cfgadm -al
Ap_Id     Type Receptacle Occupant   Condition
c2::500104f000937528   tape connected  configured   unknown
c3::50060160082006e2,0 tape connected  unconfigured unknown
root@solaris:~# cfgadm -c configure 50060160082006e2,0
```

5. デバイス名と World Wide Name との関連付けを確認します。コマンド *ls -al /dev/rmt | grep WWN* を使用します。ここで *WWN* は World Wide Name です。

```
root@solaris:~# ls -al /dev/rmt | grep 50060160082006e2,0
lrwxrwxrwx 1 root root 94 May 20 05:05 3un -> /
../../devices/pci@1f,700000/SUNW,qlc@2/fp@0,0/st@w50060160082006e2,0:
```

6. 推奨される最小の Solaris パッチレベルが適用されている場合は、ここで停止して、アーカイブディスクストレージの構成に進みます。

7. それ以外の場合は、デバイスのターゲット ID を取得します。
8. */kernel/drv/st.conf* を編集します。前の手順で確認したターゲット ID を指定して、ベンダーで指定されたエントリーを *tape-config-list* に追加します。
9. 強制的に *st* モジュールをリロードします。コマンド *update_drv -f st* を使用します。

```
root@solaris:~# update_drv -f st
root@solaris:~#
```

10. 次に、アーカイブディスクストレージの構成に進みます。

3.2.2. アーカイブディスクストレージの構成

ディスクアーカイブ内のボリュームには、ZFS、UFS、QFS、または NFS ファイルシステムを使用できます。アーカイブとステージングのパフォーマンスを最適にするには、アーカイブとステージングで使用可能な帯域幅が最大になる一方で、アーカイブジョブとステージングジョブ間、および Oracle HSM とその他のアプリケーション間の競合発生が最小になるように、ファイルシステムおよびベースとなるストレージを構成します。次のガイドラインに従ってください。

1. ファイルシステムへのアクセスのために Oracle HSM とその他のアプリケーションやユーザー間で競合が発生しないように、専用のファイルシステムを使用します。
2. ファイルシステムまたは ZFS データセットごとに 1 つの Oracle HSM アーカイブディスクボリュームを構成し、アーカイブディスクボリュームで収容できるストレージ領域の量に割り当てを設定します。

   アーカイブボリュームのストレージ領域が共有ディスクデバイスのプールから動的に割り当てられる場合は、ベースとなる物理ストレージが過剰に割り当てられていないことを確認します。割り当ては、Oracle HSM のアーカイブ処理によって、使用する合計ストレージが使用可能な量を上回らないようにする際に役立ちます。
3. 可能であれば、各ファイルシステムのサイズは 10 - 20 テラバイトに設定します。
4. 使用可能なディスクリソースで許可されていれば、複数のファイルシステムを構成し、個々の Oracle HSM のアーカイブジョブとステージングジョブによるファイルシステムへのアクセスのために競合が発生しないようにします。アーカイブファイルシステムの数は 15 から 30 個が最適です。
5. 同一のベースとなるファイルシステムへのアクセスのために、個々のアーカイブジョブとステージングジョブが競合しないように、専用デバイス上に各ファイルシステムを構成します。

単一のファイルシステムのサブディレクトリは、個別のアーカイブボリュームとして使用しないでください。

同じ物理ドライブまたは RAID グループ上に存在する LUN 上には、複数のファイルシステムを構成しないでください。

6. 次に、アーカイブテープストレージの構成に進みます。

3.2.3. アーカイブテープストレージの構成

次のタスクを実行します。

- ドライブをライブラリに取り付ける順序の確認
- 直接接続ライブラリの構成 (存在する場合)。

3.2.3.1. ドライブをライブラリに取り付ける順序の確認

自動ライブラリに複数のドライブが含まれている場合は、Oracle HSM のマスター構成ファイル (*mcf*) 内のドライブの順序が、ライブラリコントローラに表示されるドライブの順序と同じである必要があります。この順序は、ホストで表示され、ホストの */var/adm/messages* ファイルで報告されるデバイスの順序とは異なる場合があります。

Oracle HSM メタデータサーバーとデータムーバーホストごとに、次に示すタスクを実行してドライブの順序を確認します。

- ライブラリと Solaris ホストのドライブ情報の収集
- 使用している装置に応じて、直接接続ライブラリ内のドライブを Solaris デバイス名にマップするか、ACSLS 接続ライブラリ内のドライブを Solaris デバイス名にマップします。

3.2.3.1.1. ライブラリと Solaris ホストのドライブ情報の収集

1. ライブラリのドキュメントを参照してください。ドライブとターゲットの識別方法を確認してください。ローカルオペレータパネルがある場合、これを使用してドライブの順序を判別する方法を参照してください。
2. ライブラリにローカルオペレータパネルがマウントされている場合、これを使用して、ドライブをコントローラに接続する順序を判別します。各ドライブの SCSI ターゲット ID または World Wide Name を判別します。
3. Solaris ホストに *root* としてログインします。

```
root@solaris:~#
```

4. */dev/scsi/changer/* 内に Solaris 論理デバイス名を一覧表示して、出力をテキストファイルにリダイレクトします。

   次の例では、*/dev/rmt/* のリストを *root* ユーザーのホームディレクトリ内のファイル *device-mappings.txt* にリダイレクトします。

```
root@solaris:~# ls -l /dev/rmt/ > /root/device-mappings.txt
```

5. 次に、直接接続ライブラリ内のドライブを Solaris デバイス名にマップ、または ACSLS 接続ライブラリ内のドライブを Solaris デバイス名にマップを実行します。

3.2.3.1.2. 直接接続ライブラリ内のドライブを Solaris デバイス名にマップ

/dev/rmt/ 内に一覧表示されている Solaris 論理ドライブ名ごと、およびライブラリが Oracle HSM サーバーホストに割り当てるドライブごとに、次の手順を実行します。

1. まだ Oracle HSM Solaris ホストにログインしていない場合、*root* としてログインします。

```
root@solaris:~#
```

2. テキストエディタで、「ライブラリと Solaris ホストのドライブ情報の収集」の手順で作成したデバイスマッピングファイルを開き、単純な表に整理します。

   後続の手順でこの情報を参照する必要があります。次の例では、*vi* エディタを使用して、権限、所有権、日付属性を */dev/rmt/* リストから削除して、ライブラリデバイス情報のヘッダーと領域を追加します。

```
root@solaris:~# vi /root/device-mappings.txt
LIBRARY SOLARIS      SOLARIS
DEVICE  LOGICAL      PHYSICAL
NUMBER  DEVICE       DEVICE
------- ----------   ----------------------------------------
        /dev/rmt/0 -> ../../devices/pci@1f,4000/scsi@2,1/st@2,0:
        /dev/rmt/1 -> ../../devices/pci@1f,4000/scsi@4,1/st@5,0:
        /dev/rmt/2 -> ../../devices/pci@1f,4000/scsi@4,1/st@6,0:
        /dev/rmt/3 -> ../../devices/pci@1f,4000/scsi@4/st@1,0:
lrwxrwxrwx 1 root root 40 Mar 18 2014 /dev/rmt/4 -> ../../devices/pci@1f,4000/scsi@4/st@2,0:
```

3. ライブラリで、すべてのドライブが空になっていることを確認します。
4. Solaris 論理デバイス名にまだマップしていないライブラリ内の最初のドライブにテープをロードします。

   次の例のために、LTO4 テープを HP Ultrium LTO4 テープドライブにロードします。
5. テープをマウントするドライブに対応する Solaris */dev/rmt/* エントリを識別します。ドライブを特定するまで、コマンド *mt -f /dev/rmt/number status* を実行します。ここで *number* は、*/dev/rmt/* 内のドライブを識別します。

   次の例では、*/dev/rmt/0* にあるドライブは空ですが、*/dev/rmt/1* にあるドライブにはテープが保持されています。そのため、ライブラリがドライブ 1 として識別するドライブは、Solaris */dev/rmt/1* に対応します。

```
root@solaris:~# mt -f /dev/rmt/0 status
/dev/rmt/0: no tape loaded or drive offline
root@solaris:~# mt -f /dev/rmt/1 status
HP Ultrium LTO 4 tape drive:
   sense key(0x0)= No Additional Sense   residual= 0   retries= 0
   file no= 0   block no= 3
```

6. デバイスマッピングファイルで、テープを収容する Solaris デバイスのエントリを特定して、指定された領域にライブラリのデバイス ID を入力します。次に、ファイルを保存します。

   次の例では、*/dev/rmt/1* の行の *LIBRARY DEVICE NUMBER* フィールドに *1* を入力します。

```
root@solaris:~# vi /root/device-mappings.txt
LIBRARY SOLARIS      SOLARIS
DEVICE  LOGICAL      PHYSICAL
NUMBER  DEVICE       DEVICE
------- ----------   ------------------------------------------
        /dev/rmt/0 -> ../../devices/pci@1f,4000/scsi@2,1/st@2,0:
   1    /dev/rmt/1 -> ../../devices/pci@1f,4000/scsi@4,1/st@5,0:
        /dev/rmt/2 -> ../../devices/pci@1f,4000/scsi@4,1/st@6,0:
        /dev/rmt/3 -> ../../devices/pci@1f,4000/scsi@4/st@1,0:
:w
```

7. テープをアンロードします。

8. ライブラリが Oracle HSM ホストに割り当てるすべてのデバイスの Solaris 論理デバイス名がデバイスマッピングファイルに保持されるまで、この手順を繰り返します。その後、ファイルを保存してエディタを閉じます。

```
root@solaris:~# vi /root/device-mappings.txt
LIBRARY SOLARIS      SOLARIS
DEVICE  LOGICAL      PHYSICAL
NUMBER  DEVICE       DEVICE
------- ----------   ----------------------------------------
   2    /dev/rmt/0 -> ../../devices/pci@1f,4000/scsi@2,1/st@2,0:
   1    /dev/rmt/1 -> ../../devices/pci@1f,4000/scsi@4,1/st@5,0:
   3    /dev/rmt/2 -> ../../devices/pci@1f,4000/scsi@4,1/st@6,0:
   4    /dev/rmt/3 -> ../../devices/pci@1f,4000/scsi@4/st@1,0:
:wq
root@solaris:~#
```

9. マッピングファイルを保存します。

   この情報は、6章「*基本ファイルシステムの構成* 」(6章「*基本ファイルシステムの構成* 」)で必要になり、13章「*Oracle HSM 構成のバックアップ*」(13章「*Oracle HSM 構成のバックアップ*」)の際に含めることもあります。

10. 次に、「直接接続ライブラリの構成」に進みます。

3.2.3.1.3. ACSLS 接続ライブラリ内のドライブを Solaris デバイス名にマップ

1. まだ Oracle HSM Solaris ホストにログインしていない場合、*root* としてログインします。

```
root@solaris:~#
```

2. テキストエディタで、「ライブラリと Solaris ホストのドライブ情報の収集」の手順で作成したデバイスマッピングファイルを開き、単純な表に整理します。

   後続の手順でこの情報を参照する必要があります。次の例では、*vi* エディタを使用して、権限、所有権、日付属性を */dev/rmt/* リストから削除して、ライブラリデバイス情報のヘッダーと領域を追加します。

```
root@solaris:~# vi /root/device-mappings.txt
```

```
LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
--------------  --------------------  --------------------------------
/dev/rmt/0
/dev/rmt/1
/dev/rmt/2
/dev/rmt/3
```

3. */dev/rmt/* で一覧表示される論理デバイス名ごとに、デバイスシリアル番号を表示します。コマンド *luxadm display /dev/rmt/number* を使用します。ここで *number* は、*/dev/rmt/* 内のドライブを識別します。

   この例では、デバイス */dev/rmt/0* のシリアル番号 *HU92K00200* を取得します。

```
root@solaris:~# luxadm display /dev/rmt/0
DEVICE PROPERTIES for tape: /dev/rmt/0
Vendor: HP
Product ID: Ultrium 4-SCSI
Revision: G25W
Serial Num: HU92K00200
...
Path status: Ready
root@solaris:~#
```

4. *device-mappings.txt* ファイルの対応する行にシリアル番号を入力します。

   次の例では、論理デバイス */dev/rmt/0* の行に、デバイス */dev/rmt/0* のシリアル番号 *HU92K00200* を記録します。

```
root@solaris:~# vi /root/device-mappings.txt
LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
--------------  --------------------  --------------------------------
/dev/rmt/0      HU92K00200
/dev/rmt/1
/dev/rmt/2
/dev/rmt/3
:wq
root@solaris:~#
```

5. */dev/rmt/* で一覧表示されるすべての論理デバイスのデバイスシリアル番号が識別され、その結果が *device-mappings.txt* ファイルに記録されるまで、前の 2 つのステップを繰り返します。

   この例では、4 つの論理デバイスを使用します。

```
root@solaris:~# vi /root/device-mappings.txt
LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
--------------  --------------------  --------------------------------
/dev/rmt/0      HU92K00200
/dev/rmt/1      HU92K00208
/dev/rmt/2      HU92K00339
/dev/rmt/3      HU92K00289
:w
root@solaris:~#
```

6. */dev/rmt/* にマップされているデバイスのシリアル番号ごとに、対応する ACSLS ドライブアドレスを取得します。ACSLS コマンド *display drive * -f serial_num* を使用します。

   次の例では、デバイス *HU92K00200* (*/dev/rmt/0*)、*HU92K00208* (*/dev/rmt/1*)、*HU92K00339* (*/dev/rmt/2*)、*HU92K00289* (*/dev/rmt/3*) の ACSLS アドレスを取得します。

```
ACSSA> display drive * -f serial_num
2014-03-29 10:49:12 Display Drive
Acs Lsm Panel Drive Serial_num
0   2   10    12    331000049255
0   2   10    16    331002031352
0   2   10    17    HU92K00200
0   2   10    18    HU92K00208
0   3   10    10    HU92K00339
0   3   10    11    HU92K00189
0   3   10    12    HU92K00289
```

7. *device-mappings.txt* ファイルの対応する行に、各 ACSLS ドライブのアドレスを記録します。ファイルを保存して、テキストエディタを閉じます。

```
root@solaris:~# vi /root/device-mappings.txt
LOGICAL DEVICE  DEVICE SERIAL NUMBER  ACSLS DEVICE ADDRESS
--------------  --------------------  ---------------------------------
/dev/rmt/0      HU92K00200            (acs=0, lsm=2, panel=10, drive=17)
/dev/rmt/1      HU92K00208            (acs=0, lsm=2, panel=10, drive=18)
/dev/rmt/2      HU92K00339            (acs=0, lsm=2, panel=10, drive=10)
/dev/rmt/3      HU92K00289            (acs=0, lsm=2, panel=10, drive=12)
:wq
```

8. マッピングファイルを保存します。

   この情報は、6章「*基本ファイルシステムの構成* 」(6章「*基本ファイルシステムの構成* 」)で必要になり、13章「*Oracle HSM 構成のバックアップ*」(13章「*Oracle HSM 構成のバックアップ*」)の際に含めることもあります。

9. アーカイブファイルシステムを構成するときに、Oracle StorageTek ACSLS ネットワーク接続ライブラリを構成します。そのため、高可用性ファイルシステムを計画している場合は、「高可用性ファイルシステム用のストレージの構成」に進みます。それ以外の場合は、4章「*Oracle HSM and QFS Software のインストール* 」に進みます。

3.2.3.2. 直接接続ライブラリの構成

直接接続テープライブラリを構成するには、ハードウェアを物理的に接続して、場合によってはSCSIドライバを構成する必要があります (Oracle HSMは、リリース5.4より前のSAM-QFSで使用される *samst* ドライバではなく、汎用 *sgen* ドライバを使用してライブラリロボットを制御します)。次のように進めます。

1. ライブラリおよびドライブをOracle HSMサーバーホストに物理的に接続します。
2. Solaris 11上ではじめてOracle HSMをインストールする場合や、Oracle HSMまたはSAM-QFS 5.4構成をアップグレードする場合は、ハードウェアが物理的に接続されたら停止します。

   Solaris 11では、*sgen* がデフォルトのSCSIドライバであるため、Oracle HSMインストールソフトウェアは、ドライバ別名と構成ファイルを自動的に更新できます。
3. Solaris 10システムにOracle HSMをインストールする場合、次のリストにあるいずれかのドライバ別名が *sgen* ドライバに割り当てられているかどうかを確認します。コマンド *grep scs.*,08 /etc/driver_aliases* を使用します。

   sgen ドライバには次のいずれかの別名が割り当てられていることがあります。

- *scsa,08.bfcp"* または *scsa,08.bvhci* (あるいはその両方)
- *scsiclass,08*

この例では、Solaris で *sgen* ドライバの別名として *scsiclass,08* が使用されています。

```
root@solaris:~# grep scs.*,08 /etc/driver_aliases
sgen "scsiclass,08"
root@solaris:~#
```

4. *grep* コマンドが *sgen "alias"* (*alias* は上のリスト内の別名) を返す場合、*sgen* ドライバがインストールされており、別名が正しく割り当てられています。そのため、高可用性ファイルシステムを構成する場合は、高可用性ファイルシステム用のストレージの構成を参照してください。それ以外の場合は、4章「*Oracle HSM and QFS Software のインストール*」に進みます。
5. *grep* コマンドで *some-driver "alias"* (*some-driver* は *sgen* 以外のドライバ、*alias* は上記の別名のいずれか) が返される場合は、その別名はすでに別のドライバに割り当てられています。そのため、**sgen** ドライバのパス指向の別名の作成を実行します。
6. コマンド *grep scs.*,08 /etc/driver_aliases* で出力が返されない場合は、*sgen* ドライバがインストールされていません。そのためこれをインストールします。コマンド *add_drv -i scsiclass,08 sgen* を使用します。

   この例では、*grep* コマンドで何も返されません。そのため *sgen* ドライバをインストールします。

```
root@solaris:~# grep scs.*,08 /etc/driver_aliases
root@solaris:~# add_drv -i scsiclass,08 sgen
```

7. コマンド *add_drv -i scsiclass,08 sgen* で「*Driver (sgen) is already installed*」というメッセージが返される場合は、ドライバがすでにインストールされていますが、接続されていません。そのためここで接続します。コマンド *update_drv -a -i scsiclass,08 sgen* を使用します。

   この例では、*add_drv* コマンドはドライバがすでにインストールされていることを示しています。そのためドライバを接続します。

```
root@solaris:~# add_drv -i scsiclass,08 sgen
```

```
Driver (sgen) is already installed.
root@solaris:~# update_drv -a -i scsiclass,08 sgen
```

8. コマンド *grep scs.*,08 /etc/driver_aliases* によって、別名 *scsiclass,08* が *sgen* ドライバに割り当てられていることが示される場合、ドライバは正しく構成されています。

```
root@solaris:~# grep scs.*,08 /etc/driver_aliases
sgen "scsiclass,08"
root@solaris:~#
```

9. 高可用性ファイルシステムを構成する場合は、高可用性ファイルシステム用のストレージの構成を参照してください。
10. それ以外の場合は、4章「*Oracle HSM and QFS Software のインストール*」に進みます。

3.2.3.3. sgen ドライバのパス指向の別名の作成

予定していた *sgen* 別名がすでに別のドライバに割り当てられている場合は、*sgen* を使用して、既存のドライバ割り当てを妨害せずに、指定したライブラリを接続するパス指向の別名を作成する必要があります。次のように進めます。

1. Oracle HSM サーバーホストに *root* としてログインします。

```
root@solaris:~#
```

2. システム構成を表示します。コマンド *cfgadm -vl* を使用します。

   cfgadm の出力は、2 行のヘッダーおよびレコードごとに 2 行で書式設定されています。

```
root@solaris:~# cfgadm -vl
Ap_Id                Receptacle  Occupant     Condition Information  When
Type        Busy  Phys_Id
c3                   connected   configured   unknown   unavailable
scsi-sas    n     /devices/pci@0/pci@0/pci@2/scsi@0:scsi
c5::500104f0008e6d78 connected   configured   unknown   unavailable
med-changer y     /devices/pci@0/.../SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78
...
root@solaris:~#
```

3. *cfgadm -vl* の出力で、ライブラリのレコードを検索します。各レコードの 2 行目の「*Type*」列で、*med-changer* を検索します。

   この例では、2 番目のレコードでライブラリを検索します。

```
root@solaris:~# cfgadm -vl
Ap_Id                Receptacle  Occupant     Condition Information  When
Type        Busy  Phys_Id
c3                   connected   configured   unknown   unavailable
scsi-sas    n     /devices/pci@0/pci@0/pci@2/scsi@0:scsi
c5::500104f0008e6d78 connected   configured   unknown   unavailable
med-changer y     /devices/pci@0/.../SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78
...
root@solaris:~#
```

4. 新しいパス指向の別名として機能する物理パスを取得します。*cfgadm -vl* の出力で「*Phys_Id*」列のエントリから、サブ文字列 */devices* を削除します。

   この例では、メディアチェンジャーレコードの「*Phys_Id*」にパス */devices/pci@0/pci@0/pci@9/SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78* が含まれているため、別名として */devices/* の後ろの文字列部分を選択します (この物理パスは、次に示す使用可能な領域に合わせて短縮されています)。

```
root@solaris:~# grep scsiclass,08 /etc/driver_aliases
sdrv "scsiclass,08"
root@solaris:~# cfgadm -vl
Ap_Id                Receptacle  Occupant     Condition Information  When
Type        Busy  Phys_Id
c3                   connected   configured   unknown   unavailable
scsi-sas    n     /devices/pci@0/pci@0/pci@2/scsi@0:scsi
c5::500104f0008e6d78 connected   configured   unknown   unavailable
med-changer y     /devices/pci@0/.../SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78
...
root@solaris:~#
```

5. パス指向の別名を作成して、*sgen* ドライバに割り当てます。コマンド *update_drv -d -i '"/path-to-library"' sgen* を使用します。ここで *path-to-library* は、前のステップで識別したパスです。

この例では、ライブラリパスを使用して、パス指向の別名 *'"/pci@0/pci@0/pci@9/SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78"'* を作成します (一重引用符と二重引用符に注意してください)。コマンドは 1 行ですが、ページレイアウトに合わせて 2 行として書式設定されています。

```
root@solaris:~# update_drv -d -i / '"/pci@0/pci@0/pci@9/SUNW,qlc@0,1/fp@0,0:fc::500104f0008e6d78"' sgen
root@solaris:~#
```

この時点で、ライブラリは *sgen* ドライバを使用して構成されています。

6. 高可用性ファイルシステムを構成する場合は、高可用性ファイルシステム用のストレージの構成に進みます。
7. それ以外の場合は、4章「*Oracle HSM and QFS Software のインストール*」に進みます。

3.3. 高可用性ファイルシステム用のストレージの構成

高可用性共有ファイルシステムを構成するには、使用しているバージョンの Solaris Cluster ソフトウェアに合ったハードウェア管理マニュアルの推奨事項に従う必要があります。これには、冗長パスとストレージデバイスの指定が含まれます。

ストレージエリアネットワーク接続で単一点障害が発生できないようにしてください。複数のインターコネクトと冗長スイッチを指定します。各ノードに複数のホストバスアダプタ (HBA) をインストールして、Oracle Solaris I/O マルチパスソフトウェアを使用します (詳細は、Oracle Solaris お客様向けドキュメントライブラリの『*Oracle Solaris SAN 構成およびマルチパス化ガイド*』と *stmsboot* のマニュアルページを参照)。

完全冗長プライマリストレージデバイスを構成します。ハードウェアで制御された RAID-10 ボリュームグループまたは RAID-1 Solaris Volume Manager ボリュームのいずれかで、ミラー化デバイス上に Oracle HSM ファイルシステムメタデータと構成ファイルを配置します。ハードウェアで制御された RAID-10 または RAID-5 ボリュームグループまたは RAID-1 Solaris Volume Manager ボリューム上にファイルシステムデータを配置します。

Solaris Volume Manager (SVM) マルチ所有者ディスクグループを使用して、デバイスの冗長性を確保する場合、現在のリリースの Solaris では、SVM ソフトウェアはデフォルトではインストールされなくなりました。Solaris 10 9/10 リリースに付属していたバージョンのソフトウェアをダウンロードしてインストールする必要があります。その後、Solaris Cluster ドキュメントの構成の推奨事項に従ってください。

4

Oracle HSM and QFS Software のインストール

Oracle HSM では、Oracle Solaris 11 の標準となった Image Packaging System (IPS) が使用されています。IPS はネットワークを中心としたパッケージ管理システムであり、ソフトウェアパッケージのインストール、アップグレード、および削除を効率化し、調整します。これにより、パッチの管理が大幅に簡素化され、本番環境への配備も容易になります。

管理者は Solaris Package Managerグラフィカルデスクトップアプリケーションまたは IPS 端末コマンドを使って Oracle Solaris のソフトウェアリポジトリにアクセスしたり、必要なソフトウェアパッケージを特定、ダウンロード、およびインストールしたりしますが、依存関係のチェックやパッケージの検証については IPS が自動的に処理します。IPS は、保守期間中に混乱を生じさせずに新しいソフトウェアを配備できるように、システムのスナップショットに変更を加えます。そのため、必要に応じて変更をロールバックできます。このため、稼働中の本番システムにインストールや更新を安全に適用できます。

Oracle HSM software をインストールするには、次のタスクを実行します。

- ソフトウェアの取得
- Solaris Cluster ソフトウェアのインストール (高可用性構成のみ)
- 共有 Oracle HSM ファイルシステムのアップグレード (該当する場合)
- ホストで Oracle HSM Software をインストール、アップグレード、またはダウングレードする

この章の最後では、Oracle HSM Software のアンインストールについて簡単にふれます。

4.1. ソフトウェアの取得

このセクションでは、必要なインストールソフトウェアやソフトウェア更新を取得するプロセスの概要を説明します。次のセクションを参照してください。

- インストール要件のチェック
- ソフトウェアインストールパッケージのダウンロード。

4.1.1. インストール要件のチェック

Oracle Solaris および Linux オペレーティングシステム、Oracle Cluster ソフトウェアのサポートされるバージョンや、その他の必須ソフトウェアパッケージやサポートされるソフトウェアパッケージなど、インストール要件の最新情報については、Oracle HSM リリースノート、Oracle サポートサービス (*support.oracle.com*)、および Oracle HSM WIKI ページ (*wikis.oracle.com/display/SAMQFS/Home*) を参照してください。

4.1.2. ソフトウェアインストールパッケージのダウンロード

Oracle ソフトウェア製品のインストールパッケージは、Oracle Software Delivery Cloud からダウンロードします。基本手順はすべての Oracle 製品で類似しています。

Oracle HSM リリース 6.0 パッケージをダウンロードするには、次の手順を実行します。

1. Web ブラウザウィンドウで *edelivery.oracle.com* を開きます。
2. サイトをまだ使用したことがない場合は、登録します。
3. 登録資格を使ってサインインします。
4. 該当するソフトウェアライセンスを確認するチェックボックスにチェックマークを付けます。
5. ソフトウェアに適用される輸出規制に同意するチェックボックスにチェックマークを付けます。
6. 「メディア・パック検索」ページの「製品パックを選択」コントロールにあるリストから「Oracle StorageTek Products」を選択します。
7. 「プラットフォーム」リストから、Oracle HSM software をホストするプラットフォームアーキテクチャーで「Oracle Solaris」を選択します。
8. 「実行」ボタンを押します。
9. 結果リストが表示されたら、Oracle Hierarchical Storage Manager メディアパックに対応するラジオボタンをクリックし、「続行」を押します。
10. 「Oracle Hierarchical Storage Manager and StorageTek QFS Software Media Pack for Oracle Solaris」ページが表示されたら「Readme」ボタンを押し、ダウンロードの手順を読みます。
11. 「Oracle Hierarchical Storage Manager and StorageTek QFS Software Media Pack for Oracle Solaris」ページがまだ表示されている間に「ダイジェストの表示」ボタンを押し、ダイジェスト値を保存します。

    ダイジェストとは、暗号化ハッシュ関数によって作成されるチェックサムのことです。発行されたダイジェストと、ダウンロードしたファイルからローカルで計算したダイジェストとを比較すると、ダウンロードしたファイルが完全で改ざんされていないことを確認できます。

ファイルからチェックサムを計算する手順については、Solaris の *dgst* および *md5* マニュアルページを参照してください。

12. 「Oracle Hierarchical Storage Manager and StorageTek QFS Software Media Pack for Oracle Solaris」ページがまだ表示されている間に、ライセンスを受けている製品に対応する「ダウンロード」ボタンを押します。

    リストには Oracle Hierarchical Storage Manager and StorageTek QFS Software の別個のエントリが含まれます。Oracle Hierarchical Storage Manager メディアパックには、アーカイブとファイルシステムソフトウェアの両方が含まれます。Oracle StorageTek QFS Software メディアパックには、ファイルシステムソフトウェアのみが含まれます。

13. プロンプトが表示されたら、「Readme」ページの説明に従って ZIP アーカイブをローカルディレクトリに保存します。

    選択したディレクトリは、すべての Oracle HSM ホストからローカルネットワーク経由でアクセス可能である必要があります。この章の例では、*sw_install* という名前のネットワークファイルサーバーの */hsmqfs* ディレクトリにファイルをダウンロードします。

14. 数回試しても必要なファイルをダウンロードできない場合は、Software Delivery Customer Service (*edelivery_ww@oracle.com*) に連絡し、ご相談ください。

15. ZIP ファイルのダウンロードが完了したら、ローカルディレクトリ内でファイルを解凍します。

    この例では、*/hsmqfs* サブディレクトリ内で Oracle Hierarchical Storage Manager and StorageTek QFS Software ファイル *Q12345-01.zip* を解凍したあと、その内容を一覧表示します。

```
[sw_install]root@solaris:~# cd /hsmqfs
[sw_install]root@solaris:~# unzip Q12345-01.zip
[sw_install]root@solaris:~# ls Q12345-01/
./       COPYRIGHT.txt       linux.iso             README.txt
../      iso.md5             Oracle-HSM_6.0/
[sw_install]root@solaris:~# ls Oracle-HSM_6.0/
total 42
./        COPYRIGHT.txt      linux1/       solaris_sparc/../       README.txt         linux2/
 solaris_x64/
```

16. 高可用性ファイルシステムを準備している場合は、次の「Solaris Cluster ソフトウェアのインストール (高可用性構成のみ)」に進みます。

17. マルチホスト共有ファイルシステムをアップグレードする場合は、次の「共有 Oracle HSM ファイルシステムのアップグレード」に進みます。
18. それ以外の場合は、直接「ホストで Oracle HSM Software をインストール、アップグレード、またはダウングレードする」に進みます。

4.2. Solaris Cluster ソフトウェアのインストール (高可用性構成のみ)

Oracle HSM の高可用性構成を準備する場合、次の手順を実行します。

1. Solaris Cluster ソフトウェア向けのオンラインの *Information Library* に含まれているインストールドキュメントやデータサービス管理ドキュメントの説明に従って、Oracle Solaris Cluster ソフトウェアおよび `SUNW.HAStoragePlus` データサービスソフトウェアを各ホストにインストールします。
2. 次に、「ホストで Oracle HSM Software をインストール、アップグレード、またはダウングレードする」に進みます。

4.3. 共有 Oracle HSM ファイルシステムのアップグレード

アップグレードプロセス中に使用可能な状態にしておく必要がある共有ファイルシステムのソフトウェアをアップグレードする場合、*ローリングアップグレード*を検討してください。アクティブなサーバーに加え、1 つ以上の潜在的なメタデータサーバーを構成する場合、アクティブでないサーバーを更新し、更新したサーバーをアクティブにできます。それから、残りの潜在的なメタデータサーバーおよびクライアントをアップグレードする前に、プライマリサーバーを構成および再アクティブ化します。このローリングアップグレードプロセスでは、ファイルシステムデータがクライアントからアクセスできる状態が維持されるように、アクティブな Oracle HSM メタデータサーバーが常に利用可能な状態に保たれます。

ローリングアップグレードを実行するには、次のタスクを実行します。

- かなり古いリリースの Oracle HSM のアップグレード
- ローリングアップグレードの実行

4.3.1. かなり古いリリースの Oracle HSM のアップグレード

どの時点でも、共有ファイルシステムのメタデータサーバー上およびクライアント上の Oracle HSM ソフトウェアのリリースは、古い場合でも 1 つ前までである必要があります。アップグレード先のリリースより 2 つ以上前のリリースの Oracle HSM (または SAM-QFS) software を実行しているホストが共有ファイルシステムの構成に含まれている場合、修正アクションを実行するまで目的のリリースにはアップグレードできません。

次のように進めます。

1. メタデータサーバーと同じリリースの Oracle HSM (または SAM-QFS) software を実行しているクライアントホストが存在しない場合、ソフトウェアをサーバーで使用されているリリースにアップグレードしてから、次の処理に進みます。
2. アクティブなメタデータサーバーの Oracle HSM (または SAM-QFS) software がアップグレード先のリリースより 2 つ以上前のリリースであり、かつアップグレード中にファイルシステムのマウント*状態を維持する必要がある*場合は、すべてのホストが完全に最新になるまで、ローリングアップグレードの実行を繰り返し実行し、リリースレベルを毎回 1 つずつ上げます。
3. アクティブなメタデータサーバーの Oracle HSM (または SAM-QFS) software がアップグレード先のリリースより 2 つ以上前のリリースであっても、アップグレード中にファイルシステムのマウント*状態を維持する必要*がない場合は、ローリングアップグレードを試行しないでください。「ホストで Oracle HSM Software をインストール、アップグレード、またはダウングレードする」の説明に従って、アーカイブ処理およびステージング処理を停止し、ファイルシステムをアンマウントして、各ホストを個別にアップグレードします。

4.3.2. ローリングアップグレードの実行

1. 続行する前に、必ず かなり古いリリースの Oracle HSM のアップグレードを実行します。

   ローリングアップグレードを試行する際にアップグレード先のリリースより 2 つ以上前のリリースのホストが存在していた場合、アップグレードが失敗し、よくてもファイルシステムが不整合な状態になります。
2. 現時点でアクティブな (最初の) メタデータサーバーに *`root`* としてログインします。次に、現時点で潜在的な (2 番目の) メタデーターサーバーに、同じく *`root`* としてログインします。

   この例では、アクティブなメタデータサーバー *`first-mds`* にログインします。次に、2 つ目の端末ウィンドウでセキュアシェル (*`ssh`*) を使用して、非アクティブで潜在的なメタデータサーバー *`second-mds`* にログインします。

```
[first-mds]root@solaris:~#


[first-mds]root@solaris:~# ssh root@second-mds
Password:
[second-mds]root@solaris:~#
```

3. 現在非アクティブ状態になっている 2 番目のメタデーターサーバーをアップグレードします。「ホストで Oracle HSM Software をインストール、アップグレード、またはダウングレードする」の手順に従って、更新された Oracle HSM ソフトウェアをインストールします。
4. アップグレードが完了したら、2 番目のサーバーをアクティブにする準備を整えます。最初のアクティブなメタデータサーバーで Oracle HSM または SAM-QFS アーカイブファイルシステムがマウントされている場合、新しいアーカイブおよびステージングアクティビティーをすべて停止し、メディアドライブをアイドル状態にして、現在のジョブが終了するまで待ちます。その後、ライブラリ制御デーモンを停止します。

   アーカイブアクティビティーの停止方法の完全な説明については、『Oracle Hierarchical Storage Manager and StorageTek QFS Software 保守および管理ガイド』を参照してください。

```
[first-mds]root@solaris:~# samcmd aridle
[first-mds]root@solaris:~# samcmd stidle
[first-mds]root@solaris:~# samcmd 901 idle
...
[first-mds]root@solaris:~# samcmd a
 ...
Waiting for :arrun
[first-mds]root@solaris:~# samcmd r
...
ty   eq status      act  use  state  vsn
li  801  ---------p    0   0%  off
         empty
...
[first-mds]root@solaris:~# samd stop
[first-mds]root@solaris:~#
```

5. 2 番目のメタデータサーバーで、Oracle HSM 構成ファイルをロードし、Oracle HSM プロセスを起動します。コマンド *samd config* を使用します。

```
[second-mds]root@solaris:~# samd config
[second-mds]root@solaris:~#
```

6. 2 番目のメタデータサーバーで Oracle HSM ファイルシステムをマウントします。

```
[second-mds]root@solaris:~# mount sharefs1
```

```
[second-mds]root@solaris:~#
```

7. 新しく更新された 2 番目のメタデータサーバーをアクティブにします。2 番目のメタデータサーバーから、コマンド *samsharefs -s server file-system* を発行します。ここで *server* は新しく更新されたメタデータサーバーのホスト名、*file-system* は Oracle HSM 共有ファイルシステムの名前です。

   この例では、潜在的なメタデータサーバーは *second-mds*、ファイルシステム名は *sharefs1* です。

```
[second-mds]root@solaris:~# samsharefs -s second-mds sharefs1
[second-mds]root@solaris:~#
```

8. 非アクティブ状態になった最初のメタデータサーバーをアップグレードします。「ホストで Oracle HSM Software をインストール、アップグレード、またはダウングレードする」の手順に従って、更新された Oracle HSM ソフトウェアをインストールします。
9. アップグレード手順を完了したら、最初のメタデータサーバーを再度アクティブにする準備を整えます。現在アクティブ状態になっている 2 番目のメタデータサーバーで Oracle HSM アーカイブファイルシステムがマウントされている場合、新しいアーカイブおよびステージングアクティビティーをすべて停止し、メディアドライブをアイドル状態にして、現在のジョブが終了するまで待ちます。その後、ライブラリ制御デーモンを停止します。

```
[second-mds]root@solaris:~# samcmd aridle
[second-mds]root@solaris:~# samcmd stidle
...
[second-mds]root@solaris:~# samd stop
[second-mds]root@solaris:~#
```

10. 最初のメタデータサーバーで、Oracle HSM 構成ファイルをロードし、Oracle HSM プロセスを起動します。コマンド *samd config* を使用します。

```
[first-mds]root@solaris:~# samd config
[first-mds]root@solaris:~#
```

11. 最初のメタデータサーバーで Oracle HSM ファイルシステムをマウントします。

```
[first-mds]root@solaris:~# mount sharefs1
[first-mds]root@solaris:~#
```

12. 最初のメタデータサーバーを再度アクティブにします。最初のメタデータサーバーから、コマンド *samsharefs -s server file-system* を発行します。ここで *server* は潜在的なメタデータサーバーのホスト名、*file-system* は Oracle HSM 共有ファイルシステムの名前です。

    この例では、潜在的なメタデータサーバーは *first-mds*、ファイルシステム名は *sharefs1* です。

```
[first-mds]root@solaris:~# samsharefs -s first-mds sharefs1
[first-mds]root@solaris:~#
```

13. 残りのクライアントを更新します。「ホストで Oracle HSM Software をインストール、アップグレード、またはダウングレードする」の手順に従って、更新された Oracle HSM ソフトウェアをインストールします。
14. ここで停止します。アップグレードが完了しました。

4.4. ホストで Oracle HSM Software をインストール、アップグレード、またはダウングレードする

個々のホストで Oracle HSM software をインストール、アップグレード、またはダウングレードするには、次のタスクを実行します。

- Oracle Solaris ホストでの Oracle HSM Software のインストール、アップグレード、またはダウングレード.
- Linux ホストに Oracle HSM クライアントソフトウェアをインストールまたは更新する (存在する場合)。

4.4.1. Oracle Solaris ホストでの Oracle HSM Software のインストール、アップグレード、またはダウングレード

Solaris ホストで Oracle HSM パッケージをインストール、アップグレード、またはダウングレードするには、次のタスクの実行から開始します。

- ソフトウェアの変更に対してホストを準備する.
- ホストのアーキテクチャーに対するパッケージの特定.

実際の状況にもっとも適合するインストールタスクを実行します。

- 新しいソフトウェアをインストールし、かつホストのオペレーティングシステムが Solaris 11 以降である場合、Image Packaging System (IPS) を使用してソフトウェアをインストールする (コマンド *pkg install*)。
- IPS コマンド *pkg install* を使用してインストールされたソフトウェアをアップグレードまたはダウングレードする場合、Image Packaging System (IPS) を使用してソフトウェアをアップグレードまたはダウングレードする (コマンド *pkg update*)。
- Solaris 10 のホストに新しいソフトウェアをインストールする場合、SVR4 *pkgrm* および *pkgadd* コマンドを使用してソフトウェアをインストールするを実行します。
- SVR4 コマンド *pkgadd* を使用してインストールされたソフトウェアをアップグレードする場合、SVR4 *pkgrm* および *pkgadd* コマンドを使用してソフトウェアをアップグレードまたはダウングレードするを実行します。

4.4.1.1. ソフトウェアの変更に対してホストを準備する

1. Oracle HSM software がホストシステムに現在インストールされていない場合は、「ホストのアーキテクチャーに対するパッケージの特定」に進みます。
2. それ以外の場合は、Oracle HSM サーバーに *root* としてログインします。

```
[samqfs1host]root@solaris:~#
```

3. Oracle HSM software が現在ホストシステムにインストールされている場合は、すべてのアーカイブ処理をアイドル状態にします。コマンド *samcmd aridle* を使用します。

   このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

```
[samqfs1host]root@solaris:~# samcmd aridle
[samqfs1host]root@solaris:~#
```

4. すべてのステージングプロセスをアイドル状態にします。コマンド *samcmd stidle* を使用します。

   このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

```
[samqfs1host]root@solaris:~# samcmd stidle
[samqfs1host]root@solaris:~#
```

5. アクティブなアーカイブジョブが完了するまで待機します。コマンド *samcmd a* を使用して、アーカイブプロセスのステータスを確認します。

   アーカイブプロセスが *Waiting for :arrun* の場合、アーカイブプロセスはアイドル状態になっています。

```
[samqfs1host]root@solaris:~# samcmd a
Archiver status samcmd     6.0 10:20:34 Feb 20 2015
samcmd on samqfs1host
sam-archiverd:  Waiting for :arrun
sam-arfind: ...
Waiting for :arrun
```

6. アクティブなステージングジョブが完了するまで待機します。コマンド *samcmd u* を使用してステージングプロセスのステータスを確認します。

   ステージングプロセスが *Waiting for :strun* の場合、ステージングプロセスはアイドル状態になっています。

```
[samqfs1host]root@solaris:~# samcmd u
Staging queue samcmd      6.0  10:20:34 Feb 20 2015
samcmd on solaris.demo.lan
Staging queue by media type: all
sam-stagerd:  Waiting for :strun
[samqfs1host]root@solaris:~#
```

7. すべてのリムーバブルメディアドライブをアイドル状態にしてから、続行します。ドライブごとに、コマンド *samcmd equipment-number idle* を使用します。ここで *equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイル内のドライブに割り当てられている装置の順序番号です。

   このコマンドはドライブを「*off*」にする前に、現在のアーカイブジョブおよびステージングジョブを完了できますが、新しいジョブは開始されません。この例では、4 つのドライブ (順序番号 *801*、*802*、*803*、*804*) をアイドル状態にします。

```
[samqfs1host]root@solaris:~# samcmd 801 idle
[samqfs1host]root@solaris:~# samcmd 802 idle
[samqfs1host]root@solaris:~# samcmd 803 idle
```

```
[samqfs1host]root@solaris:~# samcmd 804 idle
[samqfs1host]root@solaris:~#
```

8. 実行中のジョブが完了するまで待機します。

   コマンド *samcmd r* を使用すると、ドライブのステータスを確認できます。すべてのドライブが「*notrdy*」または「*empty*」の場合は、続行できる状態になっています。

```
[samqfs1host]root@solaris:~# samcmd r
Removable media samcmd     6.0 10:37:09 Feb 20 2014
samcmd on samqfs1host
ty   eq   status      act  use  state  vsn
li  801   ---------p    0   0%  notrdy
          empty
li  802    ---------p    0   0%  notrdy
          empty
li  803   ---------p    0   0%  notrdy
          empty
li  804   ---------p    0   0%  notrdy
          empty
[samqfs1host]root@solaris:~#
```

9. アーカイバおよびステージャープロセスがアイドル状態で、テープドライブがすべて「*notrdy*」になっている場合は、ライブラリ制御デーモンを停止します。コマンド *samd stop* を使用します。

```
[samqfs1host]root@solaris:~# samd stop
[samqfs1host]root@solaris:~#
```

10. NFS または SMB/CIFS 経由でファイルシステムが共有されている場合、ファイルシステムの共有を解除します。メタデータサーバーでコマンド *unshare mount-point* を使用します。ここで *mount-point* は、Oracle HSM ファイルシステムのマウントポイントディレクトリです。

    最初の例では、Oracle HSM スタンドアロンファイルシステム *samqfs1* の NFS 共有を停止します。

```
[samqfs1host]root@solaris:~# unshare /hsmqfs1
```

```
[samqfs1host]root@solaris:~#
```

2 番目の例では、Oracle HSM 共有ファイルシステム *samqfs2* の NFS 共有を停止します。

```
[samqfs2server]root@solaris:~# unshare /hsmqfs2
[samqfs2server]root@solaris:~#
```

11. すべての Oracle HSM ファイルシステムをアンマウントします。

    最初の例では、共有されていないスタンドアロンファイルシステム *samqfs1* をアンマウントします。

```
[samqfs1host]root@solaris:~# umount samqfs1
```

2 番目の例では、共有ファイルシステム *samqfs1* をアンマウントしていますが、クライアントがアンマウントするまでの時間として、*60* 秒が与えられており、まずクライアントからアンマウントし、そのあとでサーバーからアンマウントしています。

```
[samqfs2server]root@solaris:~# ssh root@samqfs2client1
Password:
[samqfs2client1]root@solaris:~# umount /hsmqfs2
[samqfs2client1]root@solaris:~# exit
[samqfs2server]root@solaris:~#

[samqfs2server]root@solaris:~# ssh root@samqfs2client1
Password:
[samqfs2client2]root@solaris:~# umount /hsmqfs2
[samqfs2client2]root@solaris:~# exit
[samqfs2server]root@solaris:~# umount -o await_clients=60 /sharefs2
```

12. 現時点で SAM-QFS 5.3 以前がインストールされている場合、すべてのパッケージをアンインストールします。コマンド *pkgrm SUNWsamfsu SUNWsamfsr* (QFS のみがインストールされている場合は *pkgrm SUNWqfsu SUNWqfsr*) を使用します。

パッケージは指定された順番で削除します (最初に *SUNWsamfsu*、最後に *SUNWsamfsr*)。この例では、すべての質問が自動的に回答されるように、コマンドに応答 *yes* をパイプしています。

```
[host1]root@solaris:~# yes | pkgrm SUNWsamfsu SUNWsamfsr
```

13. 次に、ホストのアーキテクチャーに対するパッケージの特定を実行します。

4.4.1.2. ホストのアーキテクチャーに対するパッケージの特定

1. Oracle HSM ホストに *root* としてログインします。

```
root@solaris:~#
```

2. Oracle HSM ダウンロードファイルが解凍されたディレクトリに移動して、目的のバージョンのパッケージが格納されているサブディレクトリを探します。

   最初にリリースされたパッケージは *Oracle_HSM_release-number* (または *STK_QFS_release-number*) サブディレクトリに格納されています。ここで、*release-number* はメジャーおよびマイナーリリース番号で、ドットで連結されます (*Oracle_HSM_6.0+/*)。パッチリリース (ある場合) は、*-patch-number* 接尾辞が付けられた類似したサブディレクトリ内に配置されます。ここで、*patch-number* は 2 桁のパッチのシーケンス番号です (*Oracle_HSM_6.0-01/*)。

   この例では、ソフトウェアの初期リリース (*Oracle_HSM_6.0/*) のダウンロードディレクトリに移動し、その内容を一覧表示します。

```
root@solaris:~# cd /net/sw-install/hsmqfs/Oracle_HSM_6.0/
root@solaris:~# ls -1
./
../
linux1/
linux2/
Notices/
README.txt
solaris_sparc/
solaris_x64/
```

3. ホストのアーキテクチャーに対応するサブディレクトリ (*solaris_sparc/*、*solaris_x64/* のいずれか) に移動し、内容を一覧表示します。

   この例では、*solaris_sparc/* サブディレクトリに移動します。

```
root@solaris:~# cd solaris_sparc/
root@solaris:~# ls -1
./
../
S10/
S11/
S11_ips/
fsmgr_5.4.zip
fsmgr_setup*
```

4. Solaris 11 以降がホストにインストールされている場合、Image Packaging System を使用してソフトウェアをインストールできます。IPS を使用するには、サブディレクトリ *S11_ips/* に移動し、Image Packaging System (IPS) を使用してソフトウェアをインストールする、または Image Packaging System (IPS) を使用してソフトウェアをアップグレードまたはダウングレードするを実行します。

```
root@solaris:~# cd S11_ips/
```

5. また、Solaris 11 以降がホストにインストールされている場合は、*pkgadd* の方法でソフトウェアをインストールすることもできます。この場合は、サブディレクトリ *S11/* に移動し、SVR4 *pkgrm* および *pkgadd* コマンドを使用してソフトウェアをアップグレードまたはダウングレードするを実行します。

```
root@solaris:~# cd S11_ips/
```

6. Solaris 10 がホストにインストールされている場合は、サブディレクトリ *S10/* に移動し、SVR4 *pkgrm* および *pkgadd* コマンドを使用してソフトウェアをアップグレードまたはダウングレードするを実行します。

```
root@solaris:~# cd S10/
```

4.4.1.3. Image Packaging System (IPS) を使用してソフトウェアをインストールする

通常、Image Packaging System (IPS) コマンドを使用して、Solaris 11 以降を実行しているホストに Oracle HSM software をインストール、アップグレード、またはダウングレードしてください。メタデータサーバーや共有ファイルシステムクライアント (存在する場合) など、ホストごとに次の手順を実行します。

1. ホストのアーキテクチャーに対するパッケージの特定をまだ実行していない場合は、これを実行します。
2. Solaris 11 IPS パッケージのリポジトリディレクトリ (*repo.samqfs/*) に移動します。

   この例では、Oracle HSM 6.0 のリポジトリディレクトリ (*Oracle_HSM_6.0/solaris_sparc/S11_ips/repo.samqfs*) に移動します。

```
root@solaris:~# cd repo.samqfs/
root@solaris:~#
```

3. Oracle Hierarchical Storage Manager and StorageTek QFS Software パッケージを両方ともインストールするには、コマンド *pkg install -g . --accept SUNWsamfs SUNWsamqassy* を使用します。ここで、*.* は現在のディレクトリ (リポジトリ)、*SUNWsamfs* および *SUNWsamqassy* は Oracle HSM の Image Packaging System パッケージ名です。

```
root@solaris:~# pkg install -g . --accept SUNWsamfs SUNWsamqassy
Creating plan
...
* The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
           Packages to install:   2
       Create boot environment:  No
Create backup boot environment: Yes
DOWNLOAD                                  PKGS         FILES    XFER (MB)   SPEED
Completed                                  2/2       520/520    21.4/21.4    0B/s
PHASE                                        ITEMS
Installing new actions                     693/693
```

```
Updating package state database                 Done
Updating image state                            Done
Creating fast lookup database                   Done
```

4. QFS Software パッケージのみをインストールするには、コマンド *pkg install -g . --accept SUNWqfs SUNWsamqassy* を使用します。ここで、*.* は現在のディレクトリ (リポジトリ)、*SUNWqfs* および *SUNWsamqassy* は Oracle HSM の Image Packaging System パッケージ名です。

```
root@solaris:~# pkg install -g . --accept SUNWqfs SUNWsamqassy
Creating plan
...
* The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
           Packages to install:   2
       Create boot environment:  No
Create backup boot environment: Yes
DOWNLOAD                                  PKGS         FILES    XFER (MB)   SPEED
Completed                                  2/2       520/520    21.4/21.4    0B/s
PHASE                                        ITEMS
Installing new actions                     693/693
Updating package state database              Done
Updating image state                         Done
Creating fast lookup database                Done
```

5. パッケージのインストールが完了したら、インストール後スクリプト *SAM-QFS-post-install* を実行します。これは、Oracle HSM インストールディレクトリ (*/opt/SUNWsamfs/*、*/opt/SUNWqfs/* のいずれか) の *util/* サブディレクトリ内にあります。

   この例では、*/opt/SUNWsamfs/util/SAM-QFS-post-install* を実行します。

```
root@solaris:~# /opt/SUNWsamfs/util/SAM-QFS-post-install
- The administrator commands will be executable by root only (group bin).
If this is the desired value, enter "y".  If you want to change
```

```
the specified value enter "c".
...
root@solaris:~#
```

6. Oracle HSM ディレクトリ */opt/SUNWsamfs/bin* と */opt/SUNWsamfs/sbin* (または */opt/SUNWqfs/bin* と */opt/SUNWqfs/sbin*) をシステムの *PATH* 変数に追加します (まだパスに含まれていない場合)。
7. Oracle HSM ディレクトリ */opt/SUNWsamfs/man* (または */opt/SUNWqfs/man*) をシステムの *MANPATH* 変数に追加します (まだマニュアルパスに含まれていない場合)。
8. 計画している Oracle HSM 構成に追加の Solaris ホストが含まれる場合は、すべてのホストにソフトウェアがインストールされるまで、この手順を最初から繰り返します。
9. 計画している Oracle HSM 構成に Linux ホストが共有ファイルシステムクライアントとして含まれる場合は、Linux ホストに Oracle HSM クライアントソフトウェアをインストールまたは更新するに進みます。
10. それ以外の場合は、5章「*samsetup 構成ウィザードの使用*」または 6章「*基本ファイルシステムの構成*」に進みます。

4.4.1.4. Image Packaging System (IPS) を使用してソフトウェアをアップグレードまたはダウングレードする

最初に Image Packaging System (IPS) を使用してインストールした Oracle HSM software を、IPS コマンドを使用してアップグレードまたはダウングレードします。

メタデータサーバーや共有ファイルシステムクライアント (存在する場合) など、ホストごとに次の手順を実行します。

1. 行なっていなければ、ホストのアーキテクチャーに対するパッケージの特定を実行します。
2. リポジトリ内の Oracle Hierarchical Storage Manager and StorageTek QFS Software パッケージを最新バージョンにアップグレードするには、コマンド *pkg update -g . --accept SUNWsamfs SUNWsamqassy* を使用します。ここで、*.* は現在のディレクトリ (リポジトリ)、*SUNWsamfs* および *SUNWsamqassy* は Oracle HSM の Image Packaging System パッケージ名です。

```
root@solaris:~# pkg update -g . --accept SUNWsamfs SUNWsamqassy
...
root@solaris:~#
```

3. リポジトリ内の QFS Software パッケージのみを最新バージョンにアップグレードするには、コマンド *pkg update -g . --accept SUNWqfs SUNWsamqassy* を使用します。ここで、*.* は現在のディレクトリ (リポジトリ)、*SUNWqfs* および *SUNWsamqassy* は Oracle HSM の Image Packaging System パッケージ名です。

```
[host1]root@solaris:~# pkg update -g . --accept SUNWqfs SUNWsamqassy
...
root@solaris:~#
```

4. Oracle HSM パッケージを指定したバージョンにダウングレードまたはアップグレードするには、最初に目的のパッケージの障害管理リソース ID (FMRI) を取得します。コマンド *pkg info -r -g . package-name* を使用します。ここで、*.* は現在のディレクトリを指定し、*package-name* は Oracle HSM パッケージの名前です。

   この例では、Oracle HSM バージョン 6.0.0 がホストにインストールされています。

```
root@solaris:~# samcmd l
Usage information samcmd      6.0.0 14:06:20 Feb  20 2015 ...
root@solaris:~#
```

   SAM-QFS 5.4.6 にダウングレードする必要があります。したがって、バージョン 5.4.6 の IPS リポジトリ (*Oracle_HSM_6.0/solaris_sparc/S11_ips/repo.samqfs*) の *SUNWsamfs* および *SUNWsamqassy* に対して *pkg info* コマンドを実行します。

```
root@solaris:~# pwd
/net/Oracle_HSM_6.0/solaris_sparc/S11_ips/repo.samqfs
root@solaris:~# pkg info -r -g . SUNWsamfs
          Name: SUNWsamfs
       Summary: StorageTek SAM and StorageTek SAM-QFS software
   Description: StorageTek Storage and Archive Manager File System
      Category: System/File System
         State: Not installed
     Publisher: samqfs
       Version: 5.4
 Build Release: 5.11
        Branch: None
```

```
Packaging Date: Tue Jul 08 22:56:56 2014
          Size: 88.64 MB
          FMRI: pkg://hsmqfs/SUNWsamfs@5.4,5.11:20140708T225656Z

root@solaris:~# pkg info -r -g . SUNWsamqassy
          Name: SUNWsamqassy
       Summary: StorageTek QFS and Storage Archive Manager SAM-QFS IPS assembly
 services
   Description: SAM-QFS IPS Assembly Services
      Category: System/File System
         State: Installed
     Publisher: samqfs
       Version: 5.4
 Build Release: 5.11
        Branch: None
Packaging Date: Fri Sep 26 17:21:35 2014
          Size: 15.15 kB
          FMRI: pkg://hsmqfs/SUNWsamqassy@5.4,5.11:20140926T172135Z
root@solaris:~#
```

5. その後、指定したバージョンに Oracle HSM パッケージをダウングレードまたはアップグレードするには、コマンド *pkg update -g . fmri* を実行します。ここで、. は現在のディレクトリを指定し、*fmri* は目的のソフトウェアバージョンの障害管理リソース ID を指定します。

   この例では、*SUNWsamfs* および *SUNWsamqassy* パッケージの 5.4.6 バージョンの FMRI を指定します。

```
root@solaris:~# pkg update -g . SUNWsamfs@5.4,5.11:20140708T225656Z
           Packages to update:   1
       Create boot environment:  No
Create backup boot environment: Yes
DOWNLOAD                                PKGS         FILES    XFER (MB)
 SPEEDCompleted                                1/1        160/160    19.2/19.2  3.4M/s
PHASE                                          ITEMS
Updating modified actions                     172/172
Updating package state database                 Done
```

```
Updating package cache                           1/1
Updating image state                            Done
Creating fast lookup database                   Done
Updating package cache                           3/3
root@solaris:~# pkg update -g . SUNWsamqassy@5.4,5.11:20140926T172135Z
...
root@solaris:~#
```

6. 計画している Oracle HSM 構成に追加の Solaris ホストが含まれる場合は、すべてのホストでソフトウェアが更新またはダウングレードされるまで、この手順を最初から繰り返します。
7. 計画している Oracle HSM 構成に Linux ホストが共有ファイルシステムクライアントとして含まれる場合は、Linux ホストに Oracle HSM クライアントソフトウェアをインストールまたは更新するに進みます。

4.4.1.5. SVR4 *pkgrm* および *pkgadd* コマンドを使用してソフトウェアをインストールする

Solaris 10 を実行するホストに Oracle HSM software をインストールする場合、また SVR4 コマンドを使用して最初にインストールされたソフトウェアをアップグレードする場合に、SVR4 パッケージコマンドを使用します。

メタデータサーバーや共有ファイルシステムクライアント (存在する場合) など、Oracle HSM Solaris ホストごとに次の手順を実行します。

1. ホストのアーキテクチャーに対するパッケージの特定をまだ実行していない場合は、これを実行します。
2. パッケージ Oracle Hierarchical Storage Manager and StorageTek QFS Software をどちらもインストールするには、コマンド *`pkgadd -d . SUNWsamfsr SUNWsamfsu`* を使用し、すべてのデフォルトを受け入れます。

   `SUNWsamfsu` パッケージをインストールする前に *`SUNWsamfsr`* パッケージをインストールする必要があります。この例では、オペレーティングシステムのディレクトリ (*`Oracle_HSM_6.0/solaris_sparc/S10`*) に位置していることを確認します。すべての質問が自動的に回答されるように、コマンドに応答 *`yes`* をパイプします。

```
root@solaris:~# pwd
/net/Oracle_HSM_6.0/solaris_sparc/s10
```

```
root@solaris:~# yes | pkgadd -d . SUNWsamfsr SUNWsamfsu
```

3. QFS Software パッケージのみをインストールするには、コマンド *pkgadd -d . SUNWqfsr SUNWqfsu* を使用し、すべてのデフォルトを受け入れます。

   SUNWqfsu パッケージをインストールする前に *SUNWqfsr* パッケージをインストールする必要があります。この例では、すべての質問が自動的に回答されるように、コマンドに応答 *yes* をパイプしています。

```
root@solaris:~# yes | pkgadd -d . SUNWqfsr SUNWqfsu
```

4. 計画している Oracle HSM 構成に Linux ホストが共有ファイルシステムクライアントとして含まれる場合は、Linux ホストに Oracle HSM クライアントソフトウェアをインストールまたは更新するに進みます。
5. それ以外の場合は、5章「*samsetup 構成ウィザードの使用*」または 6章「*基本ファイルシステムの構成*」に進みます。

4.4.1.6. SVR4 *pkgrm* および *pkgadd* コマンドを使用してソフトウェアをアップグレードまたはダウングレードする

Solaris 10 を実行するホストで Oracle HSM software をアップグレードまたはダウングレードする場合、および SVR4 コマンドを使用して最初にインストールされたソフトウェアをアップグレードまたはダウングレードする場合に、SVR4 パッケージコマンドを使用します。

メタデータサーバーや共有ファイルシステムクライアント (存在する場合) など、Oracle HSM Solaris ホストごとに次の手順を実行します。

1. Oracle HSM software を SAM-QFS 5.3 にダウングレードする場合は、最初に、以前のソフトウェアで指定されていた場所に構成ファイルを復元します。コマンド */opt/SUNWsamfs/sbin/backto 5.3* を使用します。

   backto コマンドは、ファイルを以前の場所と形式に復元します。詳細は、*backto* のマニュアルページを参照してください。

   この例では、Oracle HSM 6.0 構成ファイルを Oracle HSM 5.3 で使用するように変換します。

```
root@solaris:~# /opt/SUNWsamfs/sbin/backto 5.3 ...
root@solaris:~#
```

2. 現在インストールされているすべての Oracle HSM パッケージをアンインストールします。コマンド *pkgrm SUNWsamfsu SUNWsamfsr* (QFS のみがインストールされている場合は *pkgrm SUNWqfsu SUNWqfsr*) を使用します。

   パッケージは指定された順番で削除します (最初に *SUNWsamfsu*、最後に *SUNWsamfsr*)。この例では、すべての質問が自動的に回答されるように、コマンドに応答 *yes* をパイプしています。

```
root@solaris:~# yes | pkgrm SUNWsamfsu SUNWsamfsr
```

3. ホストのアーキテクチャーに対するパッケージの特定をまだ実行していない場合は、これを実行します。
4. パッケージ Oracle Hierarchical Storage Manager and StorageTek QFS Software をどちらもインストールするには、コマンド *pkgadd -d . SUNWsamfsr SUNWsamfsu* を使用し、すべてのデフォルトを受け入れます。

   SUNWsamfsu パッケージをインストールする前に *SUNWsamfsr* パッケージをインストールする必要があります。この例では、オペレーティングシステムの適切なディレクトリ (*Oracle_HSM_6.0/solaris_sparc/S10*) に位置していることを確認します。すべての質問が自動的に回答されるように、コマンドに応答 *yes* をパイプします。

```
root@solaris:~# pwd
/net/Oracle_HSM_6.0/solaris_sparc/s10
root@solaris:~# yes | pkgadd -d . SUNWsamfsr SUNWsamfsu
```

5. QFS Software パッケージのみをインストールするには、コマンド *pkgadd -d . SUNWqfsr SUNWqfsu* を使用し、すべてのデフォルトを受け入れます。

   SUNWqfsu パッケージをインストールする前に *SUNWqfsr* パッケージをインストールする必要があります。この例では、すべての質問が自動的に回答されるように、コマンドに応答 *yes* をパイプしています。

```
root@solaris:~# pwd
/net/Oracle_HSM_6.0/solaris_sparc/s10
root@solaris:~# yes | pkgadd -d . SUNWqfsr SUNWqfsu
```

6. 計画している Oracle HSM 構成に Linux ホストが共有ファイルシステムクライアントとして含まれる場合は、Linux ホストに Oracle HSM クライアントソフトウェアをインストールまたは更新するに進みます。
7. それ以外の場合は、5章「*samsetup 構成ウィザードの使用*」または 6章「*基本ファイルシステムの構成*」に進みます。

4.4.2. Linux ホストに Oracle HSM クライアントソフトウェアをインストールまたは更新する

Oracle HSM 共有ファイルシステムの Linux クライアントごとに、次の手順を実行します。

1. Linux クライアントに *root* としてログインします。

```
[root@linux ~]#
```

2. マウントされているすべての Oracle HSM ファイルシステムをアンマウントします。
3. 古い Oracle HSM パッケージをアンインストールします。スクリプト */var/opt/SUNWsamfs/Uninstall* を実行します。

```
[root@linux ~]# /var/opt/SUNWsamfs/Uninstall
```

4. Linux クライアントの ISO イメージを特定します。ISO イメージは、Oracle HSM インストールソフトウェアをダウンロードしたディレクトリにあります (「ソフトウェアの取得」を参照)。

   この例では、*ssh* を使用してリポジトリホスト *sw-install* にログインします (IP アドレスは *192.168.0.2*)。ディレクトリ */hsmqfs* でソフトウェアを見つけます。

```
[root@linux ~]# ssh root@sw-install
Password:
[sw_install]root@solaris:~# ls -1 /hsmqfs
./       COPYRIGHT.txt       linux.iso            README.txt
../      iso.md5             Oracle-HSM_6.0/
```

5. Linux ホストで、一時ディレクトリを作成します。

   この例では、ディレクトリ */hsmtemp* を作成します。

```
[root@linux ~]# mkdir /hsmtemp
```

```
[root@linux ~]#
```

6. *linux.iso* イメージを Linux ホストで使用できるようにします。作成した一時ディレクトリにイメージを保持するリモートディレクトリを NFS マウントします。コマンド *mount -t nfs repository-host-IP:hsm-repository-dir temp-dir* を使用します。ここでは:
   - *-t nfs* はマウントするファイルシステムのタイプを識別します。
   - *repository-host-IP* はインストールソフトウェアをホストするサーバーの IP アドレスです。
   - *hsm-repository-dir* は Oracle HSM インストールソフトウェアを保持するディレクトリです。
   - *temp-dir* は Linux ホストに作成した一時ディレクトリです。

   この例では、ホスト *sw-install* (*192.168.0.2*) のディレクトリ */hsmqfs* をマウントポイントディレクトリ */hsmtemp* に NFS マウントします。

```
[root@linux ~]# mount -t nfs 192.168.0.2:/hsmqfs /hsmtemp
[root@linux ~]#
```

7. Linux ホストに *linux.iso* イメージをマウントします。コマンド *mount -o ro,loop -t iso9660 temp-dir/linux.iso /mnt* を使用します。ここでは:
   - *-o* はマウントオプションのリストを指定します。
   - *ro* はイメージを読み取り専用でマウントします。
   - *loop* はループデバイスとしてイメージをマウントします。
   - *-t iso9660* はマウントするファイルシステムのタイプを識別します。
   - *temp-dir* はリモートイメージリポジトリのディレクトリがマウントされる一時ディレクトリです。
   - */mnt* は Linux システムの標準の一時マウントポイントのディレクトリです。

   この例では、ISO イメージは */hsmtemp* にあります。

```
[root@linux ~]# mount -o ro,loop -t iso9660 /hsmtemp/linux.iso /mnt
[root@linux ~]#
```

8. インストーラを実行します。コマンド */mnt/linux1/Install* を使用します。

```
[root@linux ~]# /mnt/linux1/Install
```

9. インストールプログラムでインストールされている Linux カーネルのバージョンが認識されなかった場合、ユーザーにカスタムカーネルの作成が要求されます。「*Yes*」と入力します。

```
[root@linux ~]# ./Install
...
A direct match for your kernel wasn't found. Attempt creating a custom rpm for your kernel (yes/
no)? yes
```

Linux カーネルには多くのバリエーションが存在しています。できるだけ多くのバリエーションをサポートできるように、Oracle HSM インストールプログラムにはカスタムカーネルモジュールが組み込まれています。

10. 画面に表示される手順に従います。
11. SuSE Linux クライアントをインストールする場合は、マニュアルページを認識するようにシステムを構成します。テキストエディタで */etc/manpath.config* ファイルを開き、*SECTION* パラメータの値に *1m* を追加します。

    この例では、*vi* エディタを使用します。

```
[root@linux ~]# vi /etc/manpath.config
...
#-------------------------------------------------------------
# Section names. Manual sections will be searched in the order listed here;
# the default is 1, n, l, 8, 3, 2, 5, 4, 9, 6, 7. Multiple SECTION
# directives may be given for clarity, and will be concatenated together in
# the expected way.
# If a particular extension is not in this list (say, 1mh), it will be
# displayed with the rest of the section it belongs to. The effect of this
# is that you only need to explicitly list extensions if you want to force a
# particular order. Sections with extensions should usually be adjacent to
# their main section (e.g. "1 1mh 8 ...").
SECTION 1 1m n l 8 3 2 3posix 3pm 3perl 5 4 9 6 7
```

12. 計画している Oracle HSM 構成に追加の Linux クライアントホストが含まれる場合は、すべてのホストにクライアントソフトウェアがインストールされるまで、この手順を最初から繰り返します。
13. それ以外の場合は、5章「*samsetup 構成ウィザードの使用*」または 6章「*基本ファイルシステムの構成*」に進みます。

4.5. Oracle HSM Software のアンインストール

このセクションでは次の手順の概要を示します。

- Solaris ホストの Oracle HSM をアンインストールする
- Linux ホスト上の Oracle HSM クライアントのアンインストール.

> **注意:**
>
> 既存の構成を使用して Oracle HSM をアップグレードまたは再インストールする場合は、ソフトウェアをアンインストールしないでください。アンインストールすると、すべての構成ファイルが削除されます。代わりに、「Oracle Solaris ホストでの Oracle HSM Software のインストール、アップグレード、またはダウングレード」に示されているアップグレード方法の 1 つを使用します。

4.5.1. Solaris ホストの Oracle HSM をアンインストールする

ソフトウェアを完全にアンインストールし、構成ファイルを削除するには、次の手順に従います。

1. ホストに *root* としてログインします。

```
root@solaris:~#
```

2. Solaris Image Packaging System を使用して Solaris 11 以降にソフトウェアをインストールした場合、コマンド *pkg uninstall SUNWsamfs SUNWsamqassy* (QFS software のみがインストールされている場合は *pkg uninstall SUNWqfs SUNWsamqassy*) を使用してソフトウェアをアンインストールします。

```
root@solaris:~#  pkg uninstall SUNWsamfs SUNWsamqassy
```

3. SVR4 *pkginstall* メソッドを使用して Solaris 10 または Solaris 11 にソフトウェアをインストールした場合、コマンド *pkgrm SUNWsamfsu SUNWsamfsr* (QFS software のみがインストールされている場合は、*pkgrm SUNWqfsu SUNWqfsr*) を使用してアンインストールします。

パッケージは指定された順番で削除します (最初に *SUNWsamfsu*、最後に *SUNWsamfsr*)。この例では、すべての質問が自動的に回答されるように、コマンドに応答 *yes* をパイプしています。

```
root@solaris:~# yes | pkgrm SUNWsamfsu SUNWsamfsr
```

4. SVR4 *pkginstall* メソッドを使用して Solaris 10 または Solaris 11 にソフトウェアをインストールした場合は、不要になった構成ファイルとログファイルを削除します。

```
root@solaris:~# rm -R /var/opt/SUNWsamfs/
root@solaris:~# rm -R /etc/opt/SUNWsamfs/
root@solaris:~# rm -R /var/adm/sam-log/
root@solaris:~#
```

5. ホストをリブートします。

```
root@solaris:~#  reboot
```

6. ここで停止します。

4.5.2. Linux ホスト上の Oracle HSM クライアントのアンインストール

Linux クライアントソフトウェアをアンインストールして完全に削除するには、次の手順を実行します。

1. Linux クライアントホストに *root* としてログインします。

```
[root@linux ~]#
```

2. Oracle HSM スクリプト */var/opt/SUNWsamfs/Uninstall* (QFS のみがインストールされている場合は */var/opt/SUNWqfs/Uninstall*) を実行します。

   ほかのメソッドを使用しないでください。*rpm -e* などのほかの方法は、ソフトウェアのアンインストールまたは再インストール時に予期しない結果や問題を引き起こす可能性があります。そのため、常にこのスクリプトを使用してください。

```
[root@linux ~]#  /var/opt/SUNWsamfs/Uninstall
```

5

samsetup 構成ウィザードの使用

samsetup ウィザードは、テキストベースでメニュー駆動の単純なユーティリティーであり、もっとも頻繁に発生する要件を満たす Oracle HSM ファイルシステムの迅速な作成および構成に使用できます。このウィザードでは、次のすべての基本タスクが順に指示されます。

- 単一のホストでマウントされる QFS スタンドアロンファイルシステムの作成
- 複数のホストでマウントされる QFS 共有ファイルシステムの作成
- QFS ファイルシステムの Oracle HSM アーカイブの構成
- プライマリ (キャッシュ) ディスクストレージ、アーカイブディスクストレージ、リムーバブルメディアライブラリ、ドライブ、メディアなどのストレージハードウェアの構成。

ウィザードの出力は有効な Oracle HSM 構成スクリプトであり、特殊なソリューションを作成する際の開始点にもなります。

samsetup ウィザードは基本的に自己記述式であり、メニューやプロンプトによってプロセスに進むことができ、コンテキストオンラインヘルプがすぐに利用可能となっています。したがって、この章ではツール自体から提供される情報を繰り返すことはしません。

ただし、特に Oracle HSM をはじめて使用する場合には、このウィザードを使用する前にこのマニュアルの以降のセクションを確認する必要があります。

- 6章「*基本ファイルシステムの構成*」では、Oracle HSM の動作方法に関する重要な情報を提供し、構成ファイルおよびファイルシステム作成プロセスについて説明します。構成ファイルを自分で作成および編集する必要性がまったくないと判断できる場合でも、ウィザードで提供されるオプションを十分に理解するには、この情報が必要になります。
- Oracle HSM アーカイブファイルシステムが必要な場合、「ファイルシステムの保護の構成」の情報が必要になります。*samsetup* ウィザードでは、重要なファイルシステムのメタデータやログのバックアップのスケジューリングは構成されません。
- Oracle HSM 共有ファイルシステムが必要な場合は、7章「*複数のホストからのファイルシステムへのアクセス*」も確認してください。「Oracle HSM ソフトウェアを使用した複数のホストからのファイルシステムへのアクセス」および「Oracle HSM 共有ファイルシステムの構成」のセクションが特に関連性があります。

- Oracle HSM の追加機能を使用する必要がある場合は、このマニュアルの関連するセクションで追加の構成手順を参照する必要があります。たとえば、「アーカイブメディア検証の構成」、「Write Once Read Many (WORM) ファイルのサポートの有効化」、「Linear Tape File System (LTFS) のサポートの有効化」、9章「*高可用性ソリューションの準備* 」、8章「*SAM-Remote の構成* 」、および 10章「*レポートデータベースの構成*」を参照してください。
- 最後に、ファイルシステムの構成時に `samsetup` を使用する場合も、コマンド行または Oracle HSM Manager ユーザーインタフェースを使用する場合も、13章「*Oracle HSM 構成のバックアップ*」の説明に従って作業内容を保護してください。

6

基本ファイルシステムの構成

QFS ファイルシステムは、すべての Oracle HSM ソリューションの基本的な構築単位です。これは単独で使用され、高いパフォーマンス、事実上無制限の容量、および非常に大きいファイルのサポートを提供します。Oracle Hierarchical Storage Manager および適切に構成されたアーカイブストレージとともに使用すると、Oracle HSM アーカイブファイルシステムになります。これにより、アーカイブおよび非アーカイブの両方の QFS ファイルシステムが、より複雑な複数ホスト構成と高可用性構成の基礎となります。そのため、この章には、作成および構成時に必要な基本的なタスクの概要が記載されています。

- QFS ファイルシステムの構成
- Oracle HSM アーカイブファイルシステムの構成

6.1. QFS ファイルシステムの構成

基本的な QFS ファイルシステムの作成および構成は簡単です。ケースごとに次のタスクを実行します。

- ファイルシステムをサポートするディスクデバイスを準備します。
- マスター構成ファイル (*mcf*) を作成します。
- */opt/SUNWsamfs/sbin/sammkfs* コマンドを使用してファイルシステムを作成します。
- */etc/vfstab* ファイルを編集して、新しいファイルシステムをホストの仮想ファイルシステム構成に追加します。
- 新しいファイルシステムをマウントします。

プロセスは、Oracle HSM Manager のグラフィカルインタフェースか、テキストエディタとコマンド行端末のいずれかを使用して実行できます。ただし、この例では、より理解しやすくするためにエディタとコマンド行による方法を使用してプロセスの一部を明確にしています。

簡単にするためと便宜のため、このセクションの手順では、Oracle HSM の初期構成中に Solaris 仮想ファイルシステム */etc/vfstab* の構成ファイルでファイルシステムのマウントオプションを設定します。ただし、ほとんどのオプションは、オプションの */etc/opt/*

SUNWsamfs/samfs.cmd ファイルで設定することも、コマンド行から設定することもできます。詳細は、*samfs.cmd* および *mount_samfs* のマニュアルページを参照してください。

6.1.1. QFS ファイルシステムのディスクストレージの準備

構成プロセスを開始する前に、計画した構成に必要なディスクリソースを選択します。raw デバイススライスである ZFS *zvol* ボリュームまたは Solaris Volume Manager ボリュームを使用できます。

6.1.2. 汎用の ms ファイルシステムの構成

1. ファイルシステムホストに *root* としてログインします。ゾーンを含むホストが構成されている場合、大域ゾーンにログインします。

```
root@solaris:~#
```

2. ファイル */etc/opt/SUNWsamfs/mcf* を作成します。

   mcf (マスター構成ファイル) は、空白文字で区切られた 6 列からなる表で、それぞれの列は、QFS ファイルシステムを定義する次のパラメータの 1 つを表します。*Equipment Identifier*、*Equipment Ordinal*、*Equipment Type*.*Family Set*、*Device State*、および *Additional Parameters*。表内の行は、ストレージデバイスとデバイスのグループ (ファミリセット) の両方を含むファイルシステム装置を表します。

   mcf ファイルを作成するには、Oracle HSM Manager のグラフィカルユーザーインタフェースでオプションを選択するか、テキストエディタを使用します。次の例では、*vi* テキストエディタを使用します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
~
~
"/etc/opt/SUNWsamfs/mcf"  [New File]
```

3. わかりやすくするために、列見出しはコメントとして入力します。

   コメント行は、シャープ記号 (#) で始まります。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
```

4. 最初の行の *Equipment Identifier* フィールド (最初の列) に、新しいファイルシステムの名前を入力します。

   この例では、ファイルシステムの名前は *qfsms* です。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms
```

5. *Equipment Ordinal* フィールド (2 番目の列) に、ファイルシステムを一意に識別する番号を入力します。

   装置番号は、Oracle HSM によって制御されるすべての装置を一意に識別します。この例では、*qfsms* ファイルシステムに *100* を使用します。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100
```

6. 「*Equipment Type*」フィールド (3 番目の列) に、汎用 QFS ファイルシステムの装置タイプ *ms* を入力します。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100        ms
```

7. *Family Set* フィールド (4 番目の列) に、ファイルシステムの名前を入力します。

   Family Set パラメータは、無人のテープライブラリとその常駐テープドライブや、ファイルシステムとそのコンポーネントディスクデバイスなど、ユニットを形成するために一緒に構成されている装置のグループを定義します。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
```

```
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100        ms         qfsms
```

8. *Device State* 列に *on* と入力して、*Additional Parameters* 列はブランクのままにします。

   この行は完成です。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100        ms         qfsms      on
```

9. 新しい行を開始します。*Equipment Identifier* フィールド (最初の列) で選択したディスクデバイスの 1 つの ID を入力して、*Equipment Ordinal* フィールド (2 番目の列) に一意の番号を入力します。

   この例では、デバイスが *qfsms* ファイルシステムファミリセットの一部であることを強調するためにデバイス行をインデントして、デバイス番号 (この場合は *101*) を作成するためにファミリセットのデバイス番号を増分します。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100        ms         qfsms      on
  /dev/dsk/c1t3d0s3   101
```

10. ディスクデバイス行 (3 番目の列) の *Equipment Type* フィールドに、ディスクデバイスの装置タイプ *md* を入力します。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
qfsms                100        ms         qfsms      on
  /dev/dsk/c1t3d0s3   101        md
```

11. ディスクデバイス行 (4 番目の列) の *Family Set* フィールドにファイルシステムのファミリセット名を入力し、*Device State* フィールド (5 番目の列) に *on* と入力し、*Additional Parameters* フィールド (6 番目の列) はブランクのままにします。

    ファミリセット名 *qfsms* は、ファイルシステムのハードウェアの一部としてディスク装置を識別します。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#----------------   ---------  ---------  ---------  ------   ----------------
qfsms                100        ms         qfsms      on
  /dev/dsk/c1t3d0s3   101        md         qfsms      on
```

12. 次に、残りのディスクデバイスのエントリを追加して、ファイルを保存し、エディタを終了します。

```
# Equipment          Equipment  Equipment  Family     Device   Additional
# Identifier         Ordinal    Type       Set        State    Parameters
#----------------   ---------  ---------  ---------  ------   ----------------
qfsms                100        ms         qfsms      on
  /dev/dsk/c1t3d0s3   101        md         qfsms      on
  /dev/dsk/c1t4d0s5   102        md         qfsms      on
:wq
root@solaris:~#
```

13. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認します。

    sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーを検出すると停止します。

```
root@solaris:~# sam-fsd
```

14. *sam-fsd* コマンドによって *mcf* ファイルでエラーが見つかった場合は、ファイルを編集してエラーを修正し、前の手順の説明に従って再確認します。

    次の例では、*sam-fsd* によって、デバイスに関して何らかの問題があることが指摘されています。

```
root@solaris:~# sam-fsd
```

```
Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem qfsms
sam-fsd: Problem with file system devices.
```

通常、このようなエラーの原因は不注意なタイプミスです。ここで、エディタで *mcf* ファイルを開くと、デバイス *102* (2 番目の *md* デバイス) の装置名のスライス番号の部分に 0 ではなく文字 *o* を入力したことがわかりました。

```
qfsms                100        ms         qfsms      on
  /dev/dsk/c0t0d0s0   101        md          qfsms      on
  /dev/dsk/c0t3d0so   102        md          qfsms      on
```

15. *sam-fsd* コマンドがエラーなしで実行された場合、*mcf* ファイルは正確です。次の手順に進みます。

    この例は、エラーのない出力の一部です。

```
root@solaris:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal ipc misc proc date
              size    10M  age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
```

16. 新しいファイルシステム用のマウントポイントディレクトリを作成し、マウントポイントに対するアクセス権を設定します。

ユーザーはマウントポイントポイントディレクトリに移動し、マウントしたファイルシステム内のファイルにアクセスするための実行権 (*x*) を持っている必要があります。この例では、*/qfsms* マウントポイントディレクトリを作成し、アクセス権を *755* (*-rwxr-xr-x*) に設定します。

```
root@solaris:~# mkdir /qfsms
root@solaris:~# chmod 755 /qfsms
```

17. *mcf* ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM ソフトウェアに指示します。コマンド *samd config* を使用します。

```
root@solaris:~# samd config
Configuring SAM-FS
root@solaris:~#
```

18. コマンド *samd config* が失敗し、「*You need to run /opt/SUNWsamfs/util/SAM-QFS-post-install*」というメッセージが表示された場合、ソフトウェアのインストール時にインストール後スクリプトを実行することを忘れています。今すぐ実行します。

```
root@solaris:~# /opt/SUNWsamfs/util/SAM-QFS-post-install
- The administrator commands will be executable by root only (group bin).
If this is the desired value, enter "y".  If you want to change
the specified value enter "c".
...
root@solaris:~#
```

19. */opt/SUNWsamfs/sbin/sammkfs* コマンドとファイルシステムのファミリセット名を使用してファイルシステムを作成します。

    Oracle HSM ソフトウェアは、*md* デバイスのデュアル割り当てサイズとデフォルトのディスク割り当て単位 (DAU) サイズを使用します。これは、大きいファイルと小さいファイルの両方と入出力要求を格納できるため、汎用ファイルシステムに適した選択肢です。この例では、デフォルトを受け入れます。

```
root@solaris:~# sammkfs qfsms
Building 'qfsms' will destroy the contents of devices:
  /dev/dsk/c1t3d0s3
```

```
  /dev/dsk/c1t4d0s5
Do you wish to continue? [y/N]yes
total data kilobytes       = ...
```

デフォルト以外の、入出力要件をより適切に満たした DAU サイズを指定するために必要な *mr* デバイスを使用するには、*-a* オプションを指定して *sammkfs* コマンドを使用します。

```
root@solaris:~# sammkfs -a 16 qfs2ma
```

追加情報については、*sammkfs* のマニュアルページを参照してください。

20. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

21. 新しいファイルシステムをオペレーティングシステムの仮想ファイルシステム構成に追加します。ファイルをテキストエディタで開き、*qfsms* ファミリセットデバイスの行を開始します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
qfsms      -       /qfsms    samfs   -
```

22. */etc/vfstab* ファイルの 6 番目の列 *Mount at Boot* には、ほとんどの場合 *no* を入力します。

```
root@solaris:~# vi /etc/vfstab
# File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
```

```
...
qfsms      -        /qfsms    samfs   -     no
```

23. ラウンドロビン式割り当てを指定するには、*stripe=0* マウントオプションを追加します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
qfsms      -        /qfsms    samfs   -     no       stripe=0
```

24. ストライプ化割り当てを指定するには、*stripe=stripe-width* マウントオプションを追加します。ここで、*stripe-width* は、ストライプ内の各ディスクに書き込まれるディスク割り当て単位 (DAU) の数です。

    この例では、ストライプ幅を 1 つの DAU に設定します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
qfsms      -        /qfsms    samfs   -     no       stripe=1
```

    ここで、*stripe=1* オプションは、ストライプ幅として 1 つの DAU、書き込みサイズとして 2 つの DAU を指定します。そのため、ファイルシステムが一度に 2 つの DAU を書き込む場合、*qfsms* ファミリセット内の 2 つの *md* ディスクデバイスのそれぞれに、DAU を 1 つずつ書き込みます。

25. その他の必要な変更を */etc/vfstab* ファイルに行います。

たとえば、メタデータサーバーが応答していない場合にファイルシステムをバックグラウンドでマウントするには、*bg* マウントオプションを *Mount Options* フィールドに追加します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
qfsms      -        /qfsms    samfs   -     no       stripe=1,bg
```

26. *vfstab* ファイルを保存して、エディタを閉じます。

```
...
qfsms      -        /qfsms    samfs   -     no       stripe=1
:wq
root@solaris:~#
```

27. 新しいファイルシステムをマウントします。

```
root@solaris:~# mount /qfsms
```

28. ファイルシステムが完成し、使用する準備ができました。

次の手順:

- Oracle Hierarchical Storage Manager を使用してアーカイブファイルシステムを設定する場合、「Oracle HSM アーカイブファイルシステムの構成」を参照してください
- ファイルシステムで WORM (書き込み 1 回、読み取り複数回) 機能を有効にする必要がある場合、「Write Once Read Many (WORM) ファイルのサポートの有効化」を参照してください。
- LTFS を使用するシステムと相互作業する必要がある場合、またはリモートサイト間で大量のデータを転送する必要がある場合、「Linear Tape File System (LTFS) のサポートの有効化」を参照してください。

- 複数ホストのファイルシステムアクセスや高可用性構成などの追加の要件がある場合、「応用編」を参照してください。

6.1.3. 高パフォーマンス ma ファイルシステムの構成

Oracle HSM ソフトウェアをファイルシステムホストにインストールしたら、後述のとおりに *ma* ファイルシステムを構成します。

1. ファイルシステムホストに *root* としてログインします。ゾーンを含むホストが構成されている場合、大域ゾーンにログインします。

```
root@solaris:~#
```

2. メタデータを保持するディスクデバイスを選択します。
3. データを保持するディスクデバイスを選択します。
4. *mcf* ファイルを作成します。

   mcf ファイルを作成するには、Oracle HSM Manager のグラフィカルユーザーインタフェースでオプションを選択するか、テキストエディタを使用します。次の例では、*vi* テキストエディタを使用します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
~
"/etc/opt/SUNWsamfs/mcf"  [New File]
```

5. わかりやすくするために、列見出しはコメントとして入力します。

   コメント行は、シャープ記号 (#) で始まります。

```
# Equipment           Equipment  Equipment  Family  Device   Additional
# Identifier          Ordinal    Type       Set     State    Parameters
#------------------   ---------  ---------  ------  ------   ----------------
```

6. ファイルシステムファミリセットのエントリを作成します。

   この例では、ファイルシステムを *qfsma* と指定し、装置番号を *200* に増分し、装置タイプを *ma* に設定し、ファミリセット名を *qfsma* に設定して、デバイスの状態 *on* を設定します。

```
# Equipment           Equipment  Equipment  Family  Device   Additional
```

```
# Identifier          Ordinal    Type       Set     State    Parameters
#------------------   ---------  ---------  ------  ------   -----------------
qfsma          200       ma        qfsma   on
```

7. メタデータデバイスごとにエントリを追加します。装置識別子の列で選択したディスクデバイスの識別子を入力し、装置番号を設定して、装置タイプを *mm* に設定します。

   ファイルシステムのサイズに必要なメタデータを保持するために十分なメタデータデバイスを追加します。この例では、単一のメタデータデバイスを追加します。

```
# Equipment           Equipment  Equipment  Family  Device   Additional
# Identifier          Ordinal    Type       Set     State    Parameters
#------------------   ---------  ---------  ------  ------   -----------------
qfsma                 200        ma         qfsma   on
  /dev/dsk/c0t0d0s0    201        mm         qfsma   on
```

8. 次に、データデバイスのエントリを追加して、ファイルを保存し、エディタを終了します。

   これらは、*md*、*mr*、またはストライプグループ (*gxxx*) のいずれかのデバイスです。この例では、*md* デバイスを指定します。

```
# Equipment          Equipment  Equipment  Family  Device   Additional
# Identifier         Ordinal    Type       Set     State    Parameters
#------------------  ---------  ---------  ------  ------   -----------------
qfsma                200        ma         qfsma   on
  /dev/dsk/c0t0d0s0   201        mm         qfsma   on
  /dev/dsk/c0t3d0s0   202        md         qfsma   on
  /dev/dsk/c0t3d0s1   203        md         qfsma   on
:wq
root@solaris:~#
```

9. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーを検出すると停止します。

```
root@solaris:~# sam-fsd
```

10. *sam-fsd* コマンドによって *mcf* ファイルでエラーが見つかった場合は、ファイルを編集してエラーを修正し、前の手順の説明に従って再確認します。

    次の例では、*sam-fsd* によって、デバイスに関して何らかの問題があることが指摘されています。

```
root@solaris:~# sam-fsd
Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem qfsma
sam-fsd: Problem with file system devices.
```

    通常、このようなエラーの原因は不注意なタイプミスです。ここで、エディタで *mcf* ファイルを開くと、デバイス *202* (最初の *md* デバイス) の装置名のスライス番号の部分に 1 ではなく感嘆符 (*!*) を入力したことがわかりました。

```
sharefs1              200         ma         qfsma   on
  /dev/dsk/c0t0d0s0    201         mm         qfsma   on
  /dev/dsk/c0t0d0s!    202         md         qfsma   on
  /dev/dsk/c0t3d0s0    203         md         qfsma   on
```

11. *sam-fsd* コマンドがエラーなしで実行された場合、*mcf* ファイルは正確です。次の手順に進みます。

    この例は、エラーのない出力の一部です。

```
root@solaris:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal ipc misc proc date
              size    10M  age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
...
Would start sam-archiverd()
```

```
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
```

12. */opt/SUNWsamfs/sbin/sammkfs* コマンドとファイルシステムのファミリセット名を使用してファイルシステムを作成します。

    この例では、*md* デバイスを持つ *ma* ファイルシステムのデフォルトのディスク割り当て単位 (DAU) サイズである *64*K バイトを使用してファイルシステムを作成します。

```
root@solaris:~# sammkfs qfsma
Building 'qfsma' will destroy the contents of devices:
  /dev/dsk/c0t0d0s0
  /dev/dsk/c0t3d0s0
  /dev/dsk/c0t3d0s1
Do you wish to continue? [y/N]yes
total data kilobytes       = ...
```

    デフォルトは適切で汎用的な選択肢です。ただし、ファイルシステムが主に、小さいファイル、または少量のデータの読み取りおよび書き込みを行うアプリケーションをサポートする場合、DAU サイズに *16*K バイトまたは *32*K バイトを指定することもできます。16K バイトの DAU を指定するには、*-a* オプションを指定して *sammkfs* コマンドを使用します。

```
root@solaris:~# sammkfs -a 16 qfsma
```

    mr デバイスと *gXXX* ストライプグループの DAU は、*8 - 65528*K バイトの範囲内で、8K バイト単位の増分で完全に調整可能です。デフォルトは、*mr* デバイスの場合は *64*K バイトで、*gXXX* ストライプグループの場合は *256*K バイトです。詳細は、*sammkfs* のマニュアルページを参照してください。

13. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

14. 新しいファイルシステムをオペレーティングシステムの仮想ファイルシステム構成に追加します。テキストエディタで */etc/vfstab* ファイルを開き、*qfsma* ファミリセットの行を開始します。

```
root@solaris:~# vi /etc/vfstab
# File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
...
qfsma      -        /qfsma    samfs   -
```

15. */etc/vfstab* ファイルの 6 番目の列 *Mount at Boot* に、*no* と入力します。

```
root@solaris:~# vi /etc/vfstab
# File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
...
qfsma      -        /qfsma    samfs   -     no
```

16. ラウンドロビン式割り当てを指定するには、*stripe=0* マウントオプションを追加します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
...
qfsma      -        /qfsma    samfs   -     no       stripe=0
```

17. ストライプ化割り当てを指定するには、*stripe=stripe-width* マウントオプションを追加します。ここで、*stripe-width* は、ストライプ内の各ディスクに書き込まれるディスク割り当て単位 (DAU) の数を表す *[1-255]* の範囲の整数です。

    ストライプ化割り当てを指定すると、データは並行してデバイスに書き込まれます。そのため、最高のパフォーマンスを確保するには、ストレージハードウェアで使用可能な帯域幅を完全に使用するストライプ幅を選択してください。特定のストライプ幅に転送され

るデータボリュームは、ハードウェアの構成方法によって異なります。単一のディスクボリュームに実装される *md* デバイスの場合、ストライプ幅 **1** は、1 つの 64K バイトの DAU を 2 つのディスクのそれぞれに書き込みます (合計 128K バイト)。3+1 RAID 5 ボリュームグループに実装されている *md* デバイスの場合、同じストライプ幅は、1 つの 64K バイトの DAU を 2 つの各デバイス上の 3 つのデータディスクのそれぞれに転送します (転送ごとに合計 6 つの DAU 、つまり 384K バイト)。この例では、ストライプ幅を 1 つの DAU に設定します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
...
qfsma      -        /qfsma    samfs   -     no       stripe=1
```

18. 使用可能なハードウェアをよりよく活用するために、ストライプ幅の調整を試すことができます。ファイルシステムの *Mount Options* フィールドで、*stripe=n* マウントオプションを設定します。ここで、*n* は、ファイルシステムに指定された DAU サイズの倍数です。ファイルシステムの入出力パフォーマンスをテストして、必要に応じて設定を再調整します。

    stripe=0 を設定すると、Oracle HSM は、ラウンドロビン式割り当てを使用してデバイスにファイルを書き込みます。1 つのデバイスがいっぱいになるまで、各ファイルが完全にそのデバイスに割り当てられます。ラウンドロビン式は、共有ファイルシステムとマルチストリーム環境に適しています。

    この例では、RAID-5 ボリュームグループの帯域幅がストライプ幅 1 では十分に使用されていないことがわかったため、*stripe=2* を試します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
qfsma      -        /qfsma    samfs   -     no       ...,stripe=2
```

19. それ以外の場合は、*vfstab* ファイルを保存します。

```
...
qfsma      -        /qfsma    samfs   -     no       stripe=1
:wq
root@solaris:~#
```

20. 新しいファイルシステムをマウントします。

```
root@solaris:~# mount /qfsms
```

基本ファイルシステムが完成し、使用する準備ができました。

21. Oracle Hierarchical Storage Manager を使用してアーカイブファイルシステムを設定する場合、「Oracle HSM アーカイブファイルシステムの構成」を参照してください。
22. ファイルシステムで WORM (書き込み 1 回、読み取り複数回) 機能を有効にする必要がある場合、「Write Once Read Many (WORM) ファイルのサポートの有効化」を参照してください。
23. LTFS を使用するシステムと相互作業する必要がある場合、またはリモートサイト間で大量のデータを転送する必要がある場合、「Linear Tape File System (LTFS) のサポートの有効化」を参照してください。
24. 複数ホストのファイルシステムアクセスや高可用性構成などの追加の要件がある場合、「応用編」を参照してください。
25. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

6.2. Oracle HSM アーカイブファイルシステムの構成

アーカイブファイルシステムは、1 つ以上の QFS *ma*- タイプまたは *ms*- タイプのファイルシステムをアーカイブストレージおよび Oracle HSM software と組み合わせます。Oracle HSM software は、セカンダリディスクストレージまたはリムーバブルメディア、あるいはその両方を基本ファイルシステムの操作と統合するため、ファイルはさまざまメディアの複数のコピーに保持されます。この冗長性により、データの保護は連続して行われ、非常に大きいファイルのポリシー主導型の保存と効率的なストレージがサポートされます。

- アーカイブコピー用のディスクストレージの準備
- アーカイブファイルシステムの構成
- アーカイブファイルシステムのマウント
- アーカイブプロセスの構成

- ネットワーク接続テープライブラリに格納されているアーカイブメディアのカタログ作成
- ファイルシステムの保護の構成
- アーカイブメディア検証の構成
- Oracle HSM ファイルシステムでの WORM サポートの有効化

6.2.1. アーカイブコピー用のディスクストレージの準備

Oracle HSM アーカイブファイルシステムは、プライマリファイルシステムディスクキャッシュから、テープボリューム、またはディスクアーカイブとして構成されているディスクベースのファイルシステムのいずれかにファイルをコピーできます。後者の場合、Oracle HSM は、各ファイルシステムをテープカートリッジのように使用して、割り当てられたボリュームシリアル番号 (VSN) を使用してそれらを操作します。ランダムアクセスディスクデバイスでは、順次アクセステープデバイスに関連するマウントと位置付けのオーバーヘッドが発生しないため、小さいファイルのアーカイブ、再アクセス、変更が頻繁に行われる場合、ディスクアーカイブボリュームの反応性が大幅に向上することがあります。

1. 必要になる可能性があるファイルシステムの数を決定します。最適なパフォーマンスを確保するには、1 回の Oracle HSM 操作では、テープボリュームの場合と同様に、一度につき 1 つのディスクボリュームの読み取りまたは書き込みを行うようにします。そのため、必要なボリュームの数は、要件の収集と定義時に特定したワークロードによって異なります。

   一般的な配備では、通常ボリューム数 15 - 30 が妥当です。
2. ディスクアーカイブに使用可能にできるディスクリソースと合計容量を特定します。
3. 使用可能なリソースから実際に作成できるディスクボリュームの数を計算します。ボリュームあたり 10T - 20T バイトを確保します。使用可能な合計容量が 10T バイト未満の場合は、単一のアーカイブボリュームを作成できます。
4. アーカイブボリュームごとに 1 つのファイルシステムを構成します。

   任意の組み合わせのローカルまたは NFS マウント済みの QFS ファイルシステム、ZFS ファイルシステム、または UFS ファイルシステムをアーカイブボリュームとして使用できます (NFS マウント済みボリュームは、オフサイトのアーカイブコピーを作成するときに特に役立ちます)。

   単一のファイルシステムのサブディレクトリをアーカイブボリュームとして使用しないでください。複数のボリュームが物理デバイスの単一のセット上に定義されている場合、複数の Oracle HSM 操作が同じリソースを使用しようとします。この状況により、ディスクのオーバーヘッドが大幅に増加し、パフォーマンスが著しく低下する可能性があります。

このセクションの例では、15 個のファイルシステムを作成します。

- *DISKVOL1* は、アーカイブストレージとして使用するために特に作成するローカル QFS ファイルシステムです。
- *DISKVOL2* - *DISKVOL15* は、*server* という名前のリモートサーバーにマウントされている UFS ファイルシステムです。

5. 1 つ以上の QFS ファイルシステムをアーカイブストレージボリュームとして構成する場合、それらのファイルシステムをアーカイブストレージボリュームとして明確に識別するファミリセット名と装置番号の範囲をそれぞれに割り当てます。

   QFS アーカイブストレージファイルシステムをほかの Oracle HSM プライマリファイルシステムと明確に区別することで、構成の理解と維持が簡単になります。この例では、新しいファイルシステム *DISKVOL1* はその機能を示しています。*mcf* ファイルでは、この名前と装置番号 *800* によって、ディスクアーカイブが、後続の例で Oracle HSM アーカイブファイルシステムを作成するときに使用するファミリセット名と装置番号である *samms* および *100* と区別されます。

```
# Archiving file systems:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------

# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------
DISKVOL1                 800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7         801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7         802       md        DISKVOL1  on
```

6. Oracle HSM ホストで、物理テープライブラリでアーカイブテープボリュームを保持する場合と同様に、アーカイブディスクボリュームのマウントポイントを保持するための単一の親ディレクトリを作成します。

   この例では、ディレクトリ */diskvols* を作成します。

```
root@solaris:~# mkdir /diskvols
```

7. 親ディレクトリで、アーカイブファイルシステムごとにマウントポイントディレクトリを作成します。

   この例では、マウントポイントディレクトリ *DISKVOL1* および *DISKVOL2* - *DISKVOL15* を作成します。

```
root@solaris:~# mkdir /diskvols/DISKVOL1
root@solaris:~# mkdir /diskvols/DISKVOL2
...
root@solaris:~# mkdir /diskvols/DISKVOL15
```

8. Oracle HSM ホストで、*/etc/vfstab* ファイルをバックアップします。次に、これをエディタで開いて、アーカイブファイルシステムごとにエントリを追加し、マウントオプション *nosam* を各 QFS ファイルシステムに追加します。ファイルを保存して、エディタを閉じます。

   nosam マウントオプションは、QFS ファイルシステムに格納されたアーカイブコピーがそれ自体をアーカイブしないようにします。

   この例では、*vi* エディタを使用して、*DISKVOL1* および *DISKVOL2* - *DISKVOL15* にエントリを追加します。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~# vi /etc/vfstab
#File
#Device           Device   Mount               System  fsck  Mount    Mount
#to Mount         to fsck  Point               Type    Pass  at Boot  Options
#--------         -------  ------------------  ------  ----  -------  ---------
/devices          -        /devices            devfs   -     no       -
...
DISKVOL1          -        /diskvols/DISKVOL1  samfs   -     yes      nosam
server:/DISKVOL2  -        /diskvols/DISKVOL2  nfs     -     yes
server:/DISKVOL3  -        /diskvols/DISKVOL3  nfs     -     yes
...
server:/DISKVOL15 -        /diskvols/DISKVOL15 nfs     -     yes
```

```
:wq
root@solaris:~#
```

9. Oracle HSM ホストで、アーカイブファイルシステムをマウントします。

   この例では、*DISKVOL1* および *DISKVOL2* - *DISKVOL15* をマウントします。

```
root@solaris:~# mount /diskvols/DISKVOL1
root@solaris:~# mount /diskvols/DISKVOL2
...
root@solaris:~# mount /diskvols/DISKVOL15
```

10. 次に、リムーバブルメディアライブラリとドライブの準備を実行します。

6.2.2. リムーバブルメディアライブラリとドライブの準備

このセクションでは、次のタスクについて説明します。

- Oracle StorageTek ACSLS ネットワーク接続自動ライブラリの構成
- バーコード付きリムーバブルメディアのラベル付け動作の構成
- ドライブ時間値の設定

6.2.2.1. Oracle StorageTek ACSLS ネットワーク接続自動ライブラリの構成

Oracle StorageTek ACSLS ネットワーク接続ライブラリを使用している場合、次のように構成するか、Oracle HSM Managerのグラフィカルユーザーインタフェースを使用してライブラリを自動的に検出して構成できます (Oracle HSM Manager の使用手順については、オンラインヘルプを参照してください)。

次のように進めます。

1. Oracle HSM サーバーホストに *root* としてログインします。

```
root@solaris:~#
```

2. */etc/opt/SUNWsamfs* ディレクトリに移動します。

```
root@solaris:~# cd /etc/opt/SUNWsamfs
```

3. テキストエディタで、構成するネットワーク接続ライブラリのタイプに対応する名前で新しいファイルを開始します。

   この例では、Oracle StorageTek ACSLS ネットワーク接続ライブラリのパラメータファイルを開始します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/acsls1params
# Configuration File for an ACSLS Network-Attached Tape Library 1
```

4. ACSLS 接続ライブラリとの通信時に Oracle HSM ソフトウェアが使用するパラメータと値を入力します。

   Oracle HSM ソフトウェアは、次の Oracle StorageTek 自動カートリッジシステムアプリケーションプログラミングインタフェース (ACSAPI) パラメータを使用して、ACSLS 管理によるライブラリを制御します (詳細は、*stk* のマニュアルページを参照してください)。

   - *access=user-id* は、アクセス制御のオプションのユーザー識別子の値を指定します。デフォルトでは、ユーザー識別子ベースのアクセス制御はありません。
   - *hostname=hostname* は、StorageTek ACSLS インタフェースを実行するサーバーのホスト名を指定します。
   - *portnum=portname* は、ACSLS と Oracle HSM ソフトウェアとの間の通信に使用されるポート番号を指定します。
   - *ssihost=hostname* は、ACSLS ホストに接続するネットワークに対するマルチホーム Oracle HSM サーバーを識別するホスト名を指定します。デフォルトは、ローカルホストの名前。
   - *ssi_inet_port=ssi-inet-port* は、ACSLS サーバーのシステムインタフェースが着信 ACSLS 応答に使用する必要がある固定のファイアウォールポートを指定します。*0*、または [*1024-65535*] の範囲の値のいずれかを指定します。デフォルトの *0* を使用すると、ポートを動的に割り当てることができます。
   - *csi_hostport=csi-port* は、Oracle HSM がその ACSLS 要求を送信する ACSLS サーバーでのクライアントシステムインタフェースのポート番号を指定します。*0*、または [*1024-65535*] の範囲の値のいずれかを指定します。デフォルトの *0* を使用すると、システムは、ACSLS サーバーのポートマッパーでポートの照会を行います。
   - *capid=(acs=acsnum, lsm=lsmnum, cap=capnum)* は、カートリッジアクセスポート (CAP) の ACSLS アドレスを指定します。ここで、*acsnum* は、ライブラリの自動カートリッジシステム (ACS) 番号、*lsmnum* は、CAP を保持するモジュールのライブラリスト

レージモジュール (LSM) 番号、*capnum* は、必要な CAP の識別番号です。完全なアドレスは括弧で囲まれています。

- *capacity=(index-value-list)* は、リムーバブルメディアカートリッジの容量を指定します。ここで、*index-value-list* は、*index=value* ペアのコンマ区切りリストです。リスト内のそれぞれの *index* は、ACSLS によって定義されたメディアタイプのインデックスで、それぞれの *value* は、対応するボリューム容量 (1024 バイト単位) です。

  ACSLS ファイル */export/home/ACSSS/data/internal/mixed_media/media_types.dat* は、メディアタイプインデックスを定義します。サポートされる容量をオーバーライドする必要がある場合、通常指定する必要があるのは新しいカートリッジタイプの容量エントリのみです。

- *device-path-name=(acs=ACSnumber, lsm=LSMnumber, panel=Panelnumber, drive=Drivenumber)[shared]* は、クライアントに接続されているドライブの ACSLS アドレスを指定します。ここで、*device-path-name* は、Oracle HSM サーバーでデバイスを識別し、*acsnum* は、ライブラリの自動カートリッジシステム (ACS) 番号、*lsmnum* は、ドライブを制御するモジュールのライブラリストレージモジュール (LSM) 番号、*Panelnumber* は、ドライブが取り付けられているパネルの識別番号、*Drivenumber* はドライブの識別番号です。完全なアドレスは括弧で囲まれています。

  ACSLS アドレスのあとにオプションの *shared* キーワードを追加すると、複数の Oracle HSM サーバーが独自のメディアをそれぞれ排他的に制御できるかぎり、これらのサーバーはドライブを共有できます。デフォルトの場合、共有ドライブ内のカートリッジは、60 秒間のアイドル状態のあとにロード解除されます。

この例では、*acslserver1* を ACSLS ホストとして識別して、アクセスを *sam_user* に制限し、動的ポート割り当てを指定し、カートリッジアクセスポートと 2 つのドライブをマップします。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/acsls1params
# Configuration File for an ACSLS Network-Attached Tape Library 1
hostname = acslserver1
portnum = 50014
access = sam_user
ssi_inet_port = 0
csi_hostport = 0
capid = (acs=0, lsm=1, cap=0)
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1)
```

```
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2)
```

5. ファイルを保存して、エディタを閉じます。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/acsls1params
# /etc/opt/SUNWsamfs/acslibrary1
# Configuration File for an ACSLS Network-Attached Tape Library
...
/dev/rmt/0cbn = (acs=0, lsm=1, panel=0, drive=1)
/dev/rmt/1cbn = (acs=0, lsm=1, panel=0, drive=2)
:wq
root@solaris:~#
```

6. 必要に応じて、バーコード付きリムーバブルメディアのラベル付け動作の構成 、または ドライブ時間値の設定を実行します。
7. それ以外の場合は、「アーカイブファイルシステムの構成」に進みます。

6.2.2.2. バーコード付きリムーバブルメディアのラベル付け動作の構成

バーコードリーダーを使用するテープライブラリがある場合、*defaults.conf* ファイルの *labels* ディレクティブを使用すると、バーコード上のボリュームラベルに基づいて Oracle HSM を構成できます。次のように進めます。

1. Oracle HSM ホストに *root* としてログインします。

```
root@solaris:~#
```

2. メディアのバーコードの最初の 6 文字を使用してライブラリで各ボリュームに自動的にラベル付けする必要があり、デフォルトを変更していない場合、ここで停止します。必要に応じて、ドライブ時間値の設定を実行します。それ以外の場合は、「アーカイブファイルシステムの構成」に進みます。

   デフォルトでは、ライブラリがバーコードリーダーとバーコード付きメディアを保持している場合、Oracle HSM ソフトウェアは、バーコードの最初の 6 文字でボリュームを自動的にラベル付けします。
3. デフォルト以外の動作が必要な場合、または以前にデフォルトをオーバーライドした場合は、テキストエディタでファイル */etc/opt/SUNWsamfs/defaults.conf* を開きます。

この例では、*vi* エディタでファイルを開きます。

```
root@solaris:~# vi /opt/SUNWsamfs/examples/defaults.conf
...
```

4. ディレクティブ行 *labels* = が存在する場合はこれを見つけ、存在しない場合は追加します。

   この例では、ディレクティブを追加します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.
...
labels =
```

5. バーコードの最初の 6 文字に基づく自動的なラベル付け (デフォルト) を再度有効にするには、*labels* ディレクティブの値を *barcodes* に設定します。ファイルを保存して、エディタを閉じます。

   これで、Oracle HSM ソフトウェアは、テープのバーコードの最初の 6 文字をラベルとして使用して、ラベルが付いていないテープに自動的に再ラベル付けします。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
...
labels = barcodes
:wq
root@solaris:~#
```

6. テープのバーコードの最後の 6 文字に基づいた自動ラベル付けを有効にするには、*labels* ディレクティブの値を *barcodes_low* に設定します。ファイルを保存して、エディタを閉じます。

   labels ディレクティブを *barcodes_low* に設定すると、Oracle HSM ソフトウェアは、テープのバーコードの最後の 6 文字をラベルとして使用して、ラベルが付いていないテープを自動的に再ラベル付けします。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
...
```

```
labels = barcodes_low
:wq
root@solaris:~#
```

7. 自動ラベル付けを無効にして、テープからラベルを読み取るように Oracle HSM を構成するには、*labels* ディレクティブの値を *read* に設定します。ファイルを保存して、エディタを閉じます。

   labels ディレクティブが値 *read* に設定されている場合、Oracle HSM software はテープを自動的に再ラベル付けできません。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
...
labels = read
idle_unload = 0
...
:wq
root@solaris:~#
```

8. 必要に応じて、ドライブ時間値の設定を実行します。
9. それ以外の場合は、「アーカイブファイルシステムの構成」に進みます。

6.2.2.3. ドライブ時間値の設定

デフォルトでは、Oracle HSM ソフトウェアは、次のようにしてドライブ時間パラメータを設定します。

- 指定されたデバイスタイプがメディアをマウント解除するまでに必要な最小経過時間は、*60* 秒です。
- SCSI *unload* コマンドに応答しているライブラリに対して新しいコマンドを発行するまでに Oracle HSM ソフトウェアが待機する時間は、*15* 秒です。
- アイドル状態のドライブをアンロードするまでに Oracle HSM ソフトウェアが待機する時間は、*600* 秒 (10 分) です。
- 複数の Oracle HSM サーバーによって共有されるアイドル状態のドライブをアンロードするまでに Oracle HSM ソフトウェアが待機する時間は、*600* 秒 (10 分) です。

デフォルトの時間値を変更するには、次のように進めます。

1. ログインしていない場合は、Oracle HSM ホストに *root* としてログインします。

```
root@solaris:~#
```

2. テキストエディタで */etc/opt/SUNWsamfs/defaults.conf* ファイルを開きます。

   この例では、*vi* エディタを使用します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
```

3. 必要に応じて、指定されたデバイスタイプがメディアをマウント解除するまでに必要な最小経過時間を指定します。*defaults.conf* ファイルに、ディレクティブを *equipment-type_delay* = *number-of-seconds* 形式で追加します。ここで、*equipment-type* は、構成するドライブタイプを示す 2 文字の Oracle HSM コード、*number-of-seconds* は、このデバイスタイプのデフォルトの秒数を表す整数です。

   装置タイプコードと対応する装置のリストについては、付録A「*装置タイプの用語集*」を参照してください。この例では、LTO ドライブ (装置タイプ *li*) のアンロード遅延をデフォルト値 (*60* 秒) から *90* 秒に変更します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
li_delay = 90
```

4. 必要に応じて、SCSI *unload* コマンドに応答しているライブラリに新しいコマンドを発行するまでに Oracle HSM ソフトウェアが待機する時間を指定します。*defaults.conf* ファイルに、ディレクティブを *equipment-type_unload* = *number-of-seconds* 形式で追加します。ここで、*equipment-type* は、構成するドライブタイプを示す 2 文字の Oracle HSM コード、*number-of-seconds* は、このデバイスタイプの秒数を表す整数です。

   装置タイプコードと対応する装置のリストについては、付録A「*装置タイプの用語集*」を参照してください。最悪の場合に、*unload* コマンドへの応答時にライブラリで必要になる

可能性がある最大時間を設定します。この例では、LTO ドライブ (装置タイプ *li*) のアンロード遅延をデフォルト値 (*15* 秒) から *35* 秒に変更します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
li_delay = 90
li_unload = 35
```

5. 必要に応じて、アイドル状態のドライブをアンロードするまでに Oracle HSM ソフトウェアが待機する時間を指定します。*defaults.conf* ファイルで、ディレクティブを *idle_unload = number-of-seconds* 形式で追加します。ここで、*number-of-seconds* は、指定された秒数を表す整数です。

   この機能を無効にするには、*0* を指定します。この例では、デフォルト値 (*600* 秒) を *0* に変更することでこの機能を無効にします。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
li_delay = 90
li_unload = 35
idle_unload = 0
```

6. 必要に応じて、アイドル状態の共有ドライブをアンロードするまでに Oracle HSM ソフトウェアが待機する時間を指定します。*defaults.conf* ファイルで、ディレクティブを *shared_unload = number-of-seconds* 形式で追加します。ここで、*number-of-seconds* は、指定された秒数を表す整数です。

   リムーバブルメディアドライブを共有するように Oracle HSM サーバーを構成できます。このディレクティブは、ロードされたメディアを所有するサーバーが実際にはドライブを使用していない場合に、ほかのサーバーで使用できるようにドライブを解放します。この機能を無効にするには、*0* を指定します。この例では、デフォルト値 (*600* 秒) を *0* に変更することでこの機能を無効にします。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
```

```
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
idle_unload = 600
li_delay = 90
li_unload = 35
idle_unload = 0
shared_unload = 0
```

7. ファイルを保存して、エディタを閉じます。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character and change the value.
...
idle_unload = 600
li_delay = 90
li_unload = 35
idle_unload = 0
shared_unload = 0
:wq
root@solaris:~#
```

8. それ以外の場合、アーカイブファイルシステムの構成を実行します。

6.2.3. アーカイブファイルシステムの構成

アーカイブファイルシステムを構成するための手順は、データファイルの追加のコピーを格納するためのデバイスを追加する点を除き、非アーカイブファイルシステムを作成する場合と同じです。

1. QFS ファイルシステムの構成から開始します。汎用の **ms** ファイルシステムの構成、または 汎用の **ms** ファイルシステムの構成を実行できます。

   Oracle HSM Manager のグラフィカルユーザーインタフェースを使用してファイルシステムを作成できますが、このセクションの例では *vi* エディタを使用します。ここでは、ファミリセット名が *samms* で、装置番号が *100* の汎用 *ms* ファイルシステムを作成します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
```

```
# Archiving file systems:
#
# Equipment              Equipment Equipment Family  Device Additional
# Identifier             Ordinal   Type      Set     State  Parameters
#----------------------- --------- --------- ------  ------ ------------------
samms                   100       ms        samms   on
 /dev/dsk/c1t3d0s3       101       md        samms   on
 /dev/dsk/c1t3d0s4       102       md        samms   on
```

2. アーカイブテープストレージを追加するには、ライブラリのエントリの追加から開始します。「Equipment Identifier」フィールドで、ライブラリのデバイス ID を入力し、装置番号を割り当てます。

   この例では、ライブラリの装置 ID は */dev/scsi/changer/c1t0d5* です。装置番号は *900* (ディスクアーカイブ用に選択された範囲に続く範囲) に設定します。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
DISKVOL1                 800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7        801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7        802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5 900
```

3. 装置タイプを汎用 SCSI 接続テープライブラリ *rb* に設定して、テープライブラリファミリセットの名前を指定し、デバイスの状態を *on* に設定します。

   この例では、ライブラリ *library1* を使用しています。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
DISKVOL1                 800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7        801       md        DISKVOL1  on
```

```
 /dev/dsk/c4t0d2s7        802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5 900       rb        library1  on
```

4. オプションで、*Additional Parameters* 列に、ライブラリカタログを格納するパスを入力します。

   カタログパスを指定することを選択しない場合、ソフトウェアによってデフォルトのパスが設定されます。

   ドキュメントのレイアウトの制限のために、例ではライブラリカタログの長いパス *var/opt/SUNWsamfs/catalog/library1cat* は省略されています。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------
DISKVOL1                 800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7        801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7        802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5 900       rb        library1  on     ...catalog/library1cat
```

5. 次に、ライブラリファミリセットの一部である各テープドライブのエントリを追加します。各ドライブは、ライブラリに物理的に取り付けられている順序で追加します。

   「ドライブをライブラリに取り付ける順序の確認」で作成したドライブマッピングファイルに表示されているドライブの順序に従います。この例では、*/dev/rmt/1*、*/dev/rmt/0*、*/dev/rmt/2*、および */dev/rmt/3* で Solaris に接続されているドライブはそれぞれ、ライブラリ内のドライブ *1*、*2*、*3*、および *4* です。そのため、*/dev/rmt/1* は、*mcf* ファイル内でデバイス *901* として最初に表示されています。装置タイプ *tp* は、汎用 SCSI 接続テープドライブを指定します。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------  ------ -----------------
DISKVOL1                 800       ms        DISKVOL1 on
```

```
 /dev/dsk/c6t0d1s7         801       md        DISKVOL1 on
 /dev/dsk/c4t0d2s7         802       md        DISKVOL1 on
/dev/scsi/changer/c1t0d5  900       rb        library1  on     ...catalog/library1cat
 /dev/rmt/1cbn             901       tp        library1  on
 /dev/rmt/0cbn             902       tp        library1  on
 /dev/rmt/2cbn             903       tp        library1  on
 /dev/rmt/3cbn             904       tp        library1  on
```

6. 最後に、Oracle HSM ヒストリアンを自分で構成する場合、装置タイプ *hy* を使用してエントリを追加します。「Family Set」列と「Device State」列にハイフンを入力し、「Additional Parameters」列にヒストリアンのカタログへのパスを入力します。

   ヒストリアンは、アーカイブからエクスポートされたボリュームをカタログする仮想ライブラリです。ヒストリアンを構成しない場合、指定された最大の装置番号に 1 を加えた値を使用して、ソフトウェアによってヒストリアンが自動的に作成されます。

   例では、ページレイアウトのためにヒストリアンカタログの長いパスは省略されています。フルパスは */var/opt/SUNWsamfs/catalog/historian_cat* です。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ -----------------
DISKVOL1                 800       ms        DISKVOL1  on
 /dev/dsk/c6t0d1s7        801       md        DISKVOL1  on
 /dev/dsk/c4t0d2s7        802       md        DISKVOL1  on
/dev/scsi/changer/c1t0d5 900       rb        library1  on     ...catalog/SL150cat
 /dev/rmt/0cbn            901       tp        library1  on
 /dev/rmt/1cbn            902       tp        library1  on
 /dev/rmt/2cbn            903       tp        library1  on
 /dev/rmt/3cbn            904       tp        library1  on
historian                999       hy        -         -      ...catalog/historian_cat
```

7. *mcf* ファイルを保存して、エディタを閉じます。

```
...
 /dev/rmt/3cbn            904       tp        library1  on
```

```
historian               999       hy         -       -      ...catalog/historian_cat
:wq
root@solaris:~#
```

8. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認します。見つかったエラーを修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーを検出すると停止します。

```
root@solaris:~# sam-fsd
Trace file controls:
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
root@solaris:~#
```

9. 1 つ以上のファイルシステムをアーカイブストレージボリュームとして使用している場合、テキストエディタで */etc/opt/SUNWsamfs/diskvols.conf* ファイルを作成して、ボリュームシリアル番号 (VSN) を各ファイルシステムに割り当てます。ファイルシステムごとに、必要なボリュームシリアル番号、空白文字、およびファイルシステムのマウントポイントのパスで構成される新しい行を開始します。次に、ファイルを保存します。

   この例では、3 つのディスクベースのアーカイブボリュームがあります。*DISKVOL1* は、この目的のためにローカルに作成した QFS ファイルシステムです。*DISKVOL2* - *DISKVOL15* は UFS ファイルシステムです。すべてが */diskvols/* ディレクトリにマウントされています。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/diskvols.conf
# Volume
# Serial     Resource
# Number     Path
# ------     ----------------------
DISKVOL1     /diskvols/DISKVOL1
DISKVOL2     /diskvols/DISKVOL2
...
```

```
DISKVOL15     /diskvols/DISKVOL3
```

10. 新しいファイルシステム用のマウントポイントディレクトリを作成し、マウントポイントに対するアクセス権を設定します。

    ユーザーはマウントポイントポイントディレクトリに移動し、マウントしたファイルシステム内のファイルにアクセスするための実行権 (*x*) を持っている必要があります。この例では、*/samms* マウントポイントディレクトリを作成し、アクセス権を *755* (*-rwxr-xr-x*) に設定します。

```
root@solaris:~# mkdir /samms
root@solaris:~# chmod 755 /samms
```

11. *mcf* ファイルを再度読み取り、ソフトウェア自体を適宜再構成するように Oracle HSM ソフトウェアに指示します。報告されたエラーをすべて修正して、必要に応じて繰り返します

```
root@solaris:~# /opt/SUNWsamfs/sbin/samd config
Configuring SAM-FS
root@solaris:~#
```

12. 次に、アーカイブファイルシステムのマウントを実行します。

6.2.4. アーカイブファイルシステムのマウント

1. ファイルシステムホストに *root* としてログインします。ゾーンを含むホストが構成されている場合、大域ゾーンにログインします。
2. Solaris */etc/vfstab* ファイルをバックアップして、テキストエディタで開きます。

   次の例では、*vi* エディタを使用します。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
...
samms      -        /samms    samfs   -     yes      -
```

3. *高位境界値*を設定します。これは、Oracle HSM が以前アーカイブしたファイルをディスクから解放するディスクキャッシュ利用率です。Oracle HSM ファイルシステムのエントリの最後の列に、マウントオプション *high=percentage* を入力します。ここで、*percentage* は、[*0-100*] の範囲の数値です。

   ディスクのストレージ容量、平均ファイルサイズ、特定の時間にアクセスされるファイル数の見積もりに基づいてこの値を設定します。ユーザーが作成する新しいファイルと、ユーザーがアクセスする必要があるアーカイブ済みファイルの両方について十分なキャッシュ領域を常に確保する必要があります。ただし、リムーバブルメディアボリュームのマウントに関連するオーバーヘッドを回避できるように、ステージングを可能なかぎり少なくする必要もあります。

   最新の高速ディスクまたはソリッドステートデバイスを使用してプライマリキャッシュが実装されている場合、高位境界値を 95% に設定します。それ以外の場合は、80 - 85% を使用します。この例では、高位境界値を 85% に設定します。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
...
samms      -        /samms    samfs   -     yes      high=85
```

4. Oracle HSM が以前にアーカイブしたファイルをディスクから解放しないようにするディスクキャッシュ利用率である*低位境界値*を設定します。Oracle HSM ファイルシステムのエントリの最後の列に、マウントオプション *low=percentage* を入力します。ここで、*percentage* は、[*0-100*] の範囲の数値です。

   ディスクのストレージ容量、平均ファイルサイズ、特定の時間にアクセスされるファイル数の見積もりに基づいてこの値を設定します。パフォーマンス上の理由により、ファイルが頻繁に要求および変更される場合は特に、最近アクティブになっているファイルを可能なかぎり多くキャッシュに保持する必要があります。これによって、ステージング関連のオーバーヘッドが最小限に抑えられます。ただし、アーカイブコピーからディスクにステー

ジングする必要がある新しいファイルと新たにアクセスされるファイルのために必要な領域を、以前にキャッシュに入れたファイルが使用するのは望ましくありません。

最新の高速ディスクまたはソリッドステートデバイスを使用してプライマリキャッシュが実装されている場合、低位境界値を 90% に設定します。それ以外の場合は、70 - 75% を使用します。この例では、ローカルの要件に基づいて高位境界値を 75% に設定します。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
...
samms      -        /samms    samfs   -     yes      high=85,low=75
```

5. 以前にアーカイブされたファイルがディスクから解放されるときに、ユーザーが一部のファイルデータをディスクキャッシュに保持する必要がある場合、Oracle HSM ファイルシステムエントリの最後の列に部分的解放のマウントオプションを入力します。

   部分的解放により、Oracle HSM は、ディスク領域を回復するためにアーカイブ済みファイルを解放する際に、指定されたファイルの最初の部分をディスクキャッシュに残します。この方法によって、アプリケーションはファイルの先頭にあるデータに即時にアクセスでき、残りのデータはテープなどのアーカイブメディアからステージングされます。次のマウントオプションは部分的解放を管理します。

   - *maxpartial=value* は、ファイルが部分的に解放されるときにディスクキャッシュに残すことができるファイルデータの最大量を *value* に設定します。ここで、*value* は、*0-2097152* の範囲内の K バイト数です (*0* は部分開放を無効にします)。デフォルトは *16* です。
   - *partial=value* は、ファイルが部分的に解放されたあとでディスクキャッシュに残るファイルデータのデフォルトの量を *value* に設定します。ここで、*value* は、[*0-maxpartial*] の範囲内の K バイト数です。デフォルトは *16* です。ただし、ファイルの保持される部分では常に、少なくとも 1 つのディスク割り当て単位 (DAU) と等しい K バイトが使用されます。
   - *partial_stage=value* は、部分的に解放されたファイル全体がステージングされる前に読み取られる必要があるファイルデータの最小量を *value* に設定します。ここ

で、*value* は、[*0-maxpartial*] の範囲内の K バイト数です。デフォルトは、設定する場合は *-o partial* で指定される値で、それ以外の場合は *16* です。

- *stage_n_window=value* は、自動ステージングなしでテープメディアから直接読み取られるファイルから一度に読み取ることができるデータの最大量を設定します。指定される *value* は、[*64-2048000*] の範囲内の K バイト数です。デフォルトは *256* です。

  テープメディアから直接読み取られるファイルの詳細は、*stage* のマニュアルページの *-n* の下にある「*OPTIONS*」セクションを参照してください。

この例では、アプリケーションの特性に基づいて *maxpartial* を *128*、*partial* を *64* に設定し、それ以外はデフォルト値を受け入れます。

```
root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
...
samms      -        /samms    samfs   -     yes      ... maxpartial=128,partial=64
```

6. QFS ファイルシステムをアーカイブから除外する必要がある場合、*nosam* マウントオプションをそれぞれの */etc/vfstab* エントリに追加します。

   この例では、ディスクアーカイブである *DISKVOL1* ファイルシステムの *nosam* オプションが設定されています。ここで、*nosam* マウントオプションにより、アーカイブコピー自体がアーカイブされないようにします。

```
#File
#Device           Device   Mount               System  fsck  Mount    Mount
#to Mount         to fsck  Point               Type    Pass  at Boot  Options
#--------         -------  ------------------  ------  ----  -------  ---------
/devices          -        /devices            devfs   -     no       -
...
samms             -        /samms              samfs   -     yes      ... ,partial=64
DISKVOL1          -        /diskvols/DISKVOL1  samfs   -     yes      nosam
server:/DISKVOL2  -        /diskvols/DISKVOL2  nfs     -     yes
```

```
...
server:/DISKVOL15 -        /diskvols/DISKVOL15 nfs     -     yes
```

7. */etc/vfstab* ファイルを保存して、エディタを閉じます。

```
...
server:/DISKVOL15 -        /diskvols/DISKVOL15 nfs     -     yes
:wq
root@solaris:~#
```

8. Oracle HSM アーカイブファイルシステムをマウントします

```
root@solaris:~# mount /samms
```

9. 次に、アーカイブプロセスの構成を実行します。

6.2.5. アーカイブプロセスの構成

アーカイブファイルシステムを作成してマウントしたら、通常はアーカイブ要件のすべてまたはほとんどに対応でき、追加の構成はほとんどありません。ほとんどの場合、ファイルシステムを識別し、それぞれのアーカイブコピーの数を指定して、各コピーにメディアボリュームを割り当てるテキストファイル *archiver.cmd* を作成する以外の作業は必要ありません。

Oracle HSM アーカイブ処理には多数のチューンアップパラメータがありますが、明確な特殊要件がない場合、通常はデフォルト設定を受け入れるべきです。デフォルトは、可能なかぎり広範な状況でメディアマウントの数を最小限に抑えて、メディアの利用率を最大化し、エンドツーエンドのアーカイブパフォーマンスを最適化するために慎重に選択されています。そのため、調整を行う必要がある場合は、アーカイバが作業のスケジューリングとメディアの選択を行うための自由を不必要に制限する変更については特に気をつけてください。ストレージ操作の細かい管理を試みるときには、場合によってはパフォーマンスと全体的な効率が大幅に低下することがあります。

ただし、ほとんどすべての状況でアーカイブのロギングを有効にするべきです。ログファイルは適切に管理しないと過剰なサイズに達する可能性があるため、アーカイブロギングはデフォルトでは有効になっていません (管理については、『*Oracle Hierarchical Storage Manager and StorageTek QFS Software 保守および管理ガイド*』で説明します)。ただし、ファイルシステムが破損したり失われたりした場合、アーカイブログファイルを使用すると、通常であれば簡単には復元できないファイルを回復できます。ファイルシステムの保護の構成を

実行して、適切に維持すれば、回復ポイントファイル内のファイルシステムのメタデータを使用して、アーカイブコピーに格納されているデータからファイルシステムをすぐに再構築できます。ただし、ファイルシステムが破損したり失われたりする*前*、かつ最後の回復ポイントが生成された*あと*で、必然的にいくつかのファイルがアーカイブされます。この状況では、アーカイブメディアは有効なコピーを保持しますが、ファイルシステムのメタデータがない場合、コピーを自動的に見つけることができません。ファイルシステムのアーカイブログには、各アーカイブコピーと各ボリューム内での対応する *tar* ファイルの位置を保持するメディアのボリュームシリアル番号が記録されるため、*tar* ユーティリティーを使用すると、これらのファイルを回復し、ファイルシステムを完全に復元できます。

archiver.cmd ファイルを作成して、アーカイブ処理を構成するには、次のように進めます。

1. ホストに *root* としてログインします。

```
root@solaris:~#
```

2. テキストエディタで新しい */etc/opt/SUNWsamfs/archiver.cmd* ファイルを開きます。

   archiver.cmd 内の各行は、空白文字で区切られた 1 つ以上のフィールドで構成されます (先頭の空白文字は無視されます)。

   この例では、*vi* エディタを使用してファイルを開き、コメントを入力します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# Configuration file for archiving file systems
```

3. *archiver.cmd* ファイルの最初の部分に、必要な汎用アーカイブディレクティブを入力します。

   汎用ディレクティブは、2 番目のフィールドに等号 (=) 文字を含むか、またはそのほかのフィールドを持ちません。ほとんどの場合、汎用ディレクティブを設定せず、デフォルト値を使用できます (詳細は、*archiver.cmd* マニュアルページの「*GENERAL DIRECTIVES SECTION*」を参照してください)。

   このセクションは空のままにできますが、この例では、2 つの汎用ディレクティブの形式を説明するために、そのデフォルト値を入力しました。

   - *archivemeta = off* ディレクティブは、メタデータをアーカイブしないようにアーカイブ処理に指示します。

- *examine = noscan* ディレクティブは、ファイルが変更されたことがファイルシステムによって報告されるたびに、アーカイブ処理を必要とするファイルがないかどうかを検査するようアーカイブ処理に指示します (デフォルト)。

  旧バージョンの Oracle HSM では、ファイルシステム全体が定期的にスキャンされていました。通常、旧バージョンの Oracle HSM 構成との互換性のために必要な場合を除き、このディレクティブを変更するべきではありません。

```
# Configuration file for archiving file systems
#------------------------------------------------------------------------
# General Directives
archivemeta = off                                                  # default
examine = noscan                                                   # default
```

4. 必要な汎用アーカイブディレクティブをすべて入力したら、アーカイブセットへのファイルの割り当てを開始します。新しい行に、割り当てディレクティブ *fs = filesystem-name* を入力します。ここで、*filesystem-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルで定義したファイルシステムのファミリセット名です。

   割り当てディレクティブは、指定されたファイルシステム内のファイルセットをアーカイブメディア上のコピーセットにマップします。ファイルセットは、すべてのファイルシステム全体と同じ大きさにすることも、少数のファイルと同じくらい小さくすることもできます。ただし、最高のパフォーマンスと効率を確保するには、必要以上に大きい値を指定しないでください。過剰なメディアマウント、不必要なメディアの再配置、およびメディアの全体的な利用率低下の原因となる可能性があるため、必要以上のアーカイブセットを作成しないでください。ほとんどの場合、ファイルシステムごとに 1 つのアーカイブセットを割り当てます。

   この例では、アーカイブファイルシステム *samms* のアーカイブセット割り当てディレクティブを開始します。

```
# Configuration file for archiving file systems
#------------------------------------------------------------------------
# General Directives
archivemeta = off                                                  # default
examine = noscan                                                   # default
#------------------------------------------------------------------------
# Archive Set Assignments
```

```
fs = samms                                          # Archiving File System
```

5. 次の行で、アーカイブロギングを有効にします。*logfile = path/filename* ディレクティブを入力します。ここで *path/filename* は、場所とファイル名を指定します。

   前述のとおり、アーカイブログデータは、ファイルシステムの損失後の完全な回復のために不可欠です。そのため、*/var/adm/* など、Oracle HSM 以外のディレクトリにアーカイバログを書き込み、コピーを定期的に保存するように Oracle HSM を構成します。すべてのファイルシステムに関するアーカイバアクティビティーを一緒に記録するグローバル *archiver.log* を作成できますが、ファイルシステムごとにログを構成すると、ファイル回復中のログの検索が容易になります。そのため、この例では、ファイルシステム割り当てディレクティブを使用して、ここで */var/adm/samms.archiver.log* を指定します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                           # Archiving File System
logfile = /var/adm/samms.archiver.log
```

6. 次の行で、このファイルシステムからアーカイブセットにファイルを割り当てます。作成する必要があるアーカイブセットごとに、ディレクティブ *archiveset-name starting-directory expression* を入力します。ここでは:

   - *archiveset-name* は、新しいアーカイブセットに選択する名前です。
   - *starting-directory* は、Oracle HSM がファイルの検索を開始するディレクトリのパス (ファイルシステムのマウントポイントに対して相対的) です。
   - *expression* は、Solaris *find* コマンドによって定義されるブール型の式の 1 つです。

   ほとんどの場合、アーカイブセット定義は可能なかぎり包括的で単純なままにするべきです。ただし、状況によって必要な場合、ユーザーまたはグループのファイル所有権、ファイルサイズ、ファイルの日付/時間のスタンプ、ファイル名など、(正規表現を使用して) より制約的な修飾子を追加で指定することで、アーカイブセットのメンバーシップを制限できます。詳細は、*archiver.cmd* のマニュアルページを参照してください。

   この例では、*samms* ファイルシステムで検出されたすべてのファイルを、*allsamms* という名前の単一のアーカイブセットに入れます。マウントポイントディレクトリ自体 (*/samms*) で検索を開始するには、ドット (*.*) を使用してパスを指定します。

```
...
#------------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                        # Archiving File System
logfile = /var/adm/samms.archiver.log
allsamms .
```

7. 次に、*samms* ファイルシステムの *allsamms* アーカイブセットのコピーディレクティブを追加します。コピーごとに、1 つ以上の空白文字がある行を開始して、ディレクティブ *copy-number -release -norelease archive-age unarchive-age* を入力します。ここでは:

   - *copy-number* は整数です。
   - *-release* および *-norelease* は、コピーの作成後にディスクキャッシュ領域の管理方法を制御するオプションのパラメータです。*-release* を単独で使用すると、ディスク領域は、対応するコピーが作成されるとすぐに自動的に解放されます。*-norelease* を単独で使用すると、*-norelease* が設定された*すべての*コピーが作成され、*さらに*リリーサの処理が実行されるまで、ディスク領域は解放されません。*-release* および *-norelease* を一緒に使用すると、*-norelease* が設定されたすべてのコピーが作成されるとすぐに、ディスクキャッシュ領域が自動的に解放されます。
   - *archive-age* は、ファイルがアーカイブされる前に経過していることが必要な、最後に変更された時間からの時間です。時間は、整数と、識別子 *s* (秒)、*m* (分)、*h* (時)、*d* (日)、*w* (週)、および *y* (年) の任意の組み合わせで表します。デフォルトは *4m* です。
   - *unarchive-age* は、ファイルがアーカイブ解除されるまでに必要な、最後に変更された時間からの経過時間です。デフォルトでは、コピーはアーカイブ解除されません。

   完全な冗長性を確保するには、常に*アーカイブセットごとに少なくとも 2 つのコピーを指定します* (最大は 4)。この例では、コピーが 15 分のアーカイブ経過時間に達するまでそれぞれ *-norelease* を使用して 3 つのコピーを指定します。コピー *1* は、アーカイブディスクボリュームを使用して作成されるのに対して、コピー *2* および *3* はテープメディアに対して作成されます。

```
...
#------------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                        # Archiving File System
logfile = /var/adm/samms.archiver.log
```

```
allsamms .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
```

8. 残りのファイルシステムのアーカイブセットを定義します。

   この例では、QFS ファイルシステム *DISKVOL1* をコピープロセスのためのアーカイブメディアとして構成しました。そのため、*fs = DISKVOL1* のエントリを開始します。ただし、アーカイブコピーのアーカイブコピーを作成する必要はありません。そのため、ログファイルは指定せず、このファイルシステム内のファイルのアーカイブが行われないようにする *no_archive* という特殊なアーカイブセットを使用します。

```
...
#------------------------------------------------------------------------
# Archive Set Assignments
fs = samms                                          # Archiving File System
logfile = /var/adm/samms.archiver.log
allsamms .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
fs = DISKVOL1                           # QFS File System (Archival Media)
no_archive .
```

9. 次に、コピーの作成方法を管理するディレクティブを入力します。新しい行で、キーワード *params* を入力して、コピーパラメータセクションを開始します。

```
...
fs = DISKVOL1                            # QFS File System (Archival Media)
no_archive .
#------------------------------------------------------------------------
# Copy Parameter Directives
params
```

10. すべてのアーカイブセットのすべてのコピーに適用される共通のコピーパラメータを設定する必要がある場合、*allsets -param value ...* 形式の行を入力します。ここ

で、*allsets* は、すべての構成済みのアーカイブセットを表す特殊なアーカイブセットで、*-param value ...* は、スペースで区切られた 1 つ以上のパラメータ/値のペアを表します。

パラメータとその値の詳細は、*archiver.cmd* マニュアルページの「*ARCHIVE SET COPY PARAMETERS SECTION*」セクションを参照してください。

例のディレクティブはほとんどのファイルシステムに最適です。特殊な *allsets* アーカイブセットは、最適なパフォーマンスと簡単な管理のために、すべてのアーカイブセットが均一に処理されるようにします。*-sort path* パラメータは、同じディレクトリ内のファイルがアーカイブメディア上にともに残るように、すべてのアーカイブセットのすべてのコピーのテープアーカイブ (*tar*) ファイルがパスでソートされるようにします。*-offline_copy stageahead* パラメータは、オフラインファイルのアーカイブ時にパフォーマンスを向上させることができます。

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
```

11. すべてのアーカイブセット内の特定のコピーのコピーパラメータを設定する必要がある場合、*allfiles.copy-number -param value ...* 形式の行を入力します。ここで、*allsets* は、すべての構成済みアーカイブセットを表す特殊なアーカイブセット、*copy-number* は、ディレクティブが適用されるコピーの番号で、*-param value ...* は、空白文字で区切られた 1 つ以上のパラメータ/値のペアを表します。

    パラメータとその値の詳細は、*archiver.cmd* マニュアルページの「*ARCHIVE SET COPY PARAMETERS SECTION*」を参照してください。

    この例では、ディレクティブ *allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G* は、ディスクボリュームのコピー *1* を最適化します。アーカイブ処理のために最初に選択されたファイルが 10 分間待機するか、待機しているすべてのファイルの合計サイズが 500M バイト以上になると、アーカイブ処理が開始されます。最大 10 台のドライブを使用してコピーを作成でき、コピー内の各 *tar* ファイルは最大 1G バイトです。

残りのディレクティブ *allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set* および *allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set* は、テープメディアのコピー *2* および *3* を最適化します。アーカイブ処理のために最初に選択されたファイルがそれぞれ 24 時間または 48 時間待機するか、待機しているすべてのファイルの合計サイズが 20G バイト以上になると、アーカイブ処理が開始されます。最大 2 台のドライブを使用して、これらのコピーを作成でき (この数値は、インフラストラクチャーに合うように調整します)、コピー内の各 *tar* ファイルは最大 24G バイトです。*-reserve set* は、各アーカイブセットのコピー *2* および *3* が、同じアーカイブセットのコピーのみを含むテープメディアを使用して作成されるようにします。

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
```

このセクションの例では、アーカイブ処理にディスクボリュームが使用されることを想定しています。テープボリュームのみを使用する場合は、2 つのコピーを指定して、より頻繁にテープにアーカイブします。指定されたドライブ数をインフラストラクチャーに合うように調整すれば、次の構成はほとんどのシステムにとって最適です。

```
allsets -sort path -offline_copy stageahead -reserve set
allfiles.1 -startage 8h -startsize 8G -drives 2 -archmax 10G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G
```

12. 特定のアーカイブセットとコピーにディレクティブを設定する必要がある場合、*archive-set-name.copy-number -param value ...* 形式の行を入力します。ここで、*archive-set-name* は、アーカイブセットに使用される名前、*copy-number* は、ディレクティブが適用されるコピーの番号で、*-param value ...* は、スペースで区切られた 1 つ以上のパラメータ/値のペアを表します。

パラメータとその値の詳細は、*archiver.cmd* マニュアルページの「*ARCHIVE SET COPY PARAMETERS SECTION*」を参照してください。

次の例では、2 つのアーカイブセット *hq* および *branches* が *corpfs* ファイルシステムに対して定義されています。*hq.1* および *hq.2* のコピーディレクティブは、アーカイブセット *hq* にのみ適用されます。アーカイブセット *branches* は影響を受けません。

```
#------------------------------------------------------------------------
# Archive Set Assignments
fs = corpfs
logfile = /var/adm/corporatefs.archive.log
hq /corpfs/hq/
    1 -norelease 15m
    2 -norelease 15m
branches /corpfs/branches/
    1 -norelease 15m
    2 -norelease 15m
#------------------------------------------------------------------------
# Copy Parameter Directives
params
hq.1 -drives 4
hq.2 -drives 2
```

13. 必要なすべてのコピーパラメータを設定したら、新しい行に *endparams* キーワードを入力して、コピーパラメータリストを閉じます。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
endparams
```

14. オプションで、*vsnpools* キーワード、*pool-name media-type volumes* 形式の 1 つ以上のディレクティブを入力して、メディアプールを定義できます。ここで、*pool-name* は、プールに割り当てた名前、*media-type* は、付録A「*装置タイプの用語集*」で定義されているメディアタイプコードの 1 つ、*volumes* は、1 つ以上のボリュームシリアル番号 (VSN) と一致する正規表現です。*endvsnpools* キーワードを使用してディレクティブのリストを閉じます。

    メディアプールはオプションであり、通常はアーカイブ処理に使用可能なメディアを制限する必要はありません。そのため、これらの例ではメディアプールを定義しません。詳細は、*archiver.cmd* のマニュアルページの「*VSN POOL DEFINITIONS SECTION*」を参照してください。

15. 次に、アーカイブセットコピーで使用するべきアーカイブメディアの特定を開始します。新しい行に、キーワード *vsns* を入力します。

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
```

16. *archive-set-name.copy-number media-type volumes* 形式の行を入力して、アーカイブセットコピーごとにメディアを指定します。ここで、*archive-set-name.copy-number* は、ディレクティブが適用されるアーカイブセットとコピーを指定し、*media-type* は、付録A「*装置タイプの用語集*」で定義されているメディアタイプコードの 1 つで、*volumes* は、1 つ以上のボリュームシリアル番号 (VSN) と一致する正規表現です。

    完全な冗長性を確保するには、両方のアーカイブセットコピーが同じ物理ボリューム上に存在することがないように、それぞれのコピーは常にメディアの異なる範囲に割り当ててください。可能な場合は、常に少なくとも 1 つのコピーをテープなどのリムーバブルメディアに割り当ててください。

この例では、すべてのアーカイブセットの最初のコピーを、*DISKVOL1 - DISKVOL15* の範囲内のボリュームシリアル番号が指定されたアーカイブディスクメディア (タイプ *dk*) に送信します。すべてのアーカイブセットの 2 番目のコピーを、*VOL000 - VOL199* の範囲内のボリュームシリアル番号が指定されたテープメディア (タイプ *tp*) に、3 番目のコピーを、*VOL200 - VOL399* の範囲内のボリュームシリアル番号が指定されたテープメディア (タイプ *tp*) に送信します。

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk DISKVOL[1-15]
allfiles.2 tp VOL[0-1][0-9][0-9]
allfiles.2 tp VOL[2-3][0-9][0-9]
```

17. すべてのアーカイブセットコピーにメディアを指定したら、新しい行に *endvsns* キーワードを入力して、*vsns* ディレクティブのリストを閉じます。ファイルを保存して、エディタを閉じます。

```
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -drives 10 -archmax 1G
allfiles.2 -startage 24h -startsize 20G -drives 2 -archmax 24G -reserve set
allfiles.3 -startage 48h -startsize 20G -drives 2 -archmax 24G -reserve set
endparams
```

```
#-----------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk DISKVOL[1-15]
allfiles.2 tp VOL[0-1][0-9][0-9]
allfiles.2 tp VOL[2-3][0-9][0-9]
endvsns
:wq
root@solaris:~#
```

18. *archiver.cmd* ファイルでエラーを調べます。コマンド *archiver -lv* を使用します。

    archiver -lv コマンドは、*archiver.cmd* ファイルを画面に出力し、エラーが見つからない場合は構成レポートを生成します。それ以外の場合、エラーを記録して停止します。この例では、エラーがあります。

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
...
13: # File System Directives
14: #
15: fs = samms
16: logfile = /var/adm/samms.archiver.log
17: all .
18:     1 -norelease 15m
19:     2 -norelease 15m
20: fs=DISKVOL1                          # QFS File System (Archival Media)
21:
...
42: endvsns
DISKVOL1.1 has no volumes defined
1 archive set has no volumes defined
root@solaris:~#
```

19. *archiver.cmd* ファイル内にエラーが見つかった場合、修正して、ファイルを再度チェックします。

上の例では、ディスクアーカイブとして構成した QFS ファイルシステムであるファイルシステムディレクティブ *DISKVOL1* に対して *no_archive* ディレクティブを入力することを忘れていました。欠落を修正すると、*archiver -lv* はエラーなしで実行されます。

```
root@solaris:~# archiver -lv
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
...
20: fs=DISKVOL1                          # QFS File System (Archival Media)
21: no_archive .
...
42: endvsns
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
...
allfiles.1
    startage: 10m startsize: 500M drives 10: archmax: 1G
 Volumes:
  DISKVOL1 (/diskvols/DISKVOL15)
  ...
  DISKVOL15 (/diskvols/DISKVOL3)
 Total space available:   150T
allfiles.2
    startage: 24h startsize: 20G drives: 2 archmax: 24G reserve: set
 Volumes:
   VOL000
...
   VOL199
 Total space available:  300T
allfiles.3
    startage: 48h startsize: 20G drives: 2 archmax: 24G reserve: set
 Volumes:
   VOL200
...
   VOL399
 Total space available:  300T
root@solaris:~#
```

20. *archiver.cmd* ファイルを再度読み取り、それに従って Oracle HSM ソフトウェア自体を再構成するようにこのソフトウェアに指示します。*samd config* コマンドを使用します。

```
root@solaris:~# /opt/SUNWsamfs/sbin/samd config
Configuring SAM-FS
root@solaris:~#
```

21. テキストエディタで */etc/opt/SUNWsamfs/releaser.cmd* ファイルを開き、行 *list_size = 300000* を追加して、ファイルを保存してエディタを閉じます。

    list_size ディレクティブは、ファイルシステムから一度に解放できるファイルの数を、[*10-2147483648*] の範囲内の整数に設定します。*.inodes* ファイルに 100 万の i ノードに対する領域が十分にある (i ノードごとに 512 バイトを確保できる) 場合、デフォルト値は *100000* です。それ以外の場合、デフォルトは *30000* です。この数値を *300000* に増やすと、小さいファイルが多数含まれる標準的なファイルシステムにより適合するようになります。

    この例では、*vi* エディタを使用します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/releaser.cmd
# releaser.cmd
logfile = /var/opt/SUNWsamfs/releaser.log
list_size = 300000
:wq
root@solaris:~#
```

22. テキストエディタで */etc/opt/SUNWsamfs/stager.cmd* ファイルを開き、行 *maxactive = stage-requests* を追加します。ここで、*stage-requests* は、8G バイト以上の RAM が搭載されたホストでは *500000*、8G バイト未満の RAM が搭載されたホストでは *100000* です。ファイルを保存して、エディタを閉じます。

    maxactive ディレクティブは、一度にアクティブであることが可能なステージング要求の最大数を、[*1-500000*] の範囲内の整数に設定します。デフォルトでは、ホストメモリー 1G バイトあたり 5000 のステージング要求が許可されます。

    この例では、*vi* エディタを使用します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/stager.cmd
# stager.cmd
```

```
logfile = /var/opt/SUNWsamfs/stager.log
maxactive = 300000
:wq
root@solaris:~#
```

23. リサイクルはデフォルトでは有効になっていません。そのため、リムーバブルメディアボリュームのリサイクルが必要な場合、「リサイクルプロセスの構成」に進みます。
24. Oracle HSM アーカイブファイルシステムの *mcf* ファイルのアーカイブ装置セクションに、ネットワーク接続テープライブラリが含まれている場合、「ネットワーク接続テープライブラリに格納されているアーカイブメディアのカタログ作成」に進みます。
25. アーカイブテープボリュームの整合性を検証できる必要がある場合、「アーカイブメディア検証の構成」に進みます。
26. それ以外の場合は、「ファイルシステムの保護の構成」を実行します。

6.2.6. リサイクルプロセスの構成

リムーバブルメディアボリュームに含まれている有効なアーカイブセットの数が、ユーザー指定の数より少ない場合、リサイクラはほかのボリュームに関する有効なデータを統合して、元のボリュームを長期間のストレージのためにエクスポートしたり、再使用のために再ラベル付けできるようにします。リサイクルは、次の 2 つの方法のいずれかで構成できます。

- アーカイブセット単位でのリサイクルの構成

  アーカイブセット単位でメディアをリサイクルする場合は、リサイクルディレクティブを *archiver.cmd* ファイルに追加します。各アーカイブセットコピー内のメディアをリサイクルする方法を正確に指定できます。アーカイブセットのメンバーのみが考慮されるため、リサイクル基準はより厳密に適用されます。

  可能な場合は、ライブラリ単位ではなくアーカイブセット単位でメディアをリサイクルしてください。Oracle HSM アーカイブファイルシステムでは、リサイクルは論理的に、ライブラリ管理ではなくファイルシステム操作の一部です。リサイクルによって、アーカイブ処理、解放処理、およびステージングが補完されます。そのため、これをアーカイブ処理の一部として構成することは理にかなっています。構成に、アーカイブディスクボリュームや SAM-Remote が含まれている場合、リサイクルをアーカイブセット単位で構成する必要があります。
- ライブラリ単位でのリサイクルの構成

ライブラリ単位でメディアをリサイクルする場合は、リサイクルディレクティブを *recycler.cmd* ファイルに追加します。そのため、指定したライブラリに含まれているすべてのメディアに共通のリサイクルパラメータを設定できます。リサイクルディレクティブは、ライブラリ内のすべてのボリュームに適用されるため、アーカイブセットに固有のディレクティブよりも本質的に粒度が低くなります。指定したボリュームシリアル番号 (VSN) を検査から明示的に除外できます。ただし、それ以外の場合、リサイクル処理では単に、現在有効なアーカイブファイルと見なされないものを含むボリュームが検索されます。

その結果、ライブラリ単位でリサイクルすると、リサイクルされるファイルシステムの一部ではないファイルが破壊されることがあります。リサイクルディレクティブで明示的に除外しない場合、アーカイブログやライブラリカタログのバックアップコピーなどの役に立つデータや、ほかのファイルシステムのアーカイブメディアが危険にさらされることがあります。このため、SAM-Remote を使用している場合は、ライブラリ単位ではリサイクルできません。SAM-Remote サーバーによって制御されるライブラリ内のボリュームには、サーバーではなくクライアントによって所有される外部アーカイブファイルが含まれています。

6.2.6.1. アーカイブセット単位でのリサイクルの構成

1. Oracle HSM ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. */etc/opt/SUNWsamfs/archiver.cmd* ファイルをテキストエディタで開き、コピーの *params* セクションまでスクロールダウンします。

   次の例では、*vi* エディタを使用します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 6h  -startsize 6G  -startcount 500000
allfiles.2 -startage 24h -startsize 20G -startcount 500000 -drives 5
```

3. *archiver.cmd* ファイルの *params* セクションに、アーカイブセットごとにリサイクラディレクティブを *archive-set directive-list* 形式で入力します。ここで、archive-

set は、アーカイブセットの 1 つで、*directive-list* は、ディレクティブの名前/値のペアのスペースで区切られたリストです (リサイクルディレクティブのリストについては、*archiver.cmd* のマニュアルページを参照してください)。その後、ファイルを保存してエディタを閉じます。

この例では、アーカイブセット *allfiles.1* および *allfiles.2* のリサイクルディレクティブを追加します。*-recycle_mingain 30* および *-recycle_mingain 90* ディレクティブでは、ボリュームの容量の少なくともそれぞれ 30% と 90% を回復できる場合を除き、ボリュームはリサイクルされません。*-recycle_hwm 60* ディレクティブは、リムーバブルメディア容量の 60% が使用されるとリサイクルを開始します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameters Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 6h  -startsize 6G -startcount 500000
allfiles.1 -recycle_mingain 30 -recycle_hwm 60
allfiles.2 -startage 6h -startsize 6G -startcount 500000
allfiles.2 -recycle_mingain 90 -recycle_hwm 60
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk DISKVOL1
allfiles.2 tp VOL0[0-1][0-9]
endvsns
:wq
[root@solaris:~#
```

4. *archiver.cmd* ファイルでエラーを調べます。コマンド *archiver -lv* を使用します。

   コマンド *archiver -lv* は *archiver.cmd* を読み取って、エラーが見つからない場合は構成レポートを生成します。それ以外の場合、エラーを記録して停止します。この例では、ファイルにはエラーは含まれていません。

```
root@solaris:~# archiver -lv
```

```
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
...
   VOL399
 Total space available:  300T
root@solaris:~#
```

5. *archiver.cmd* ファイル内にエラーが見つかった場合、修正して、ファイルを再度チェックします。
6. テキストエディタで *recycler.cmd* ファイルを作成します。リサイクラログのパスとファイル名を指定します。その後、ファイルを保存してエディタを閉じます。

   /var/adm/ など、Oracle HSM 以外のディレクトリにログを書き込むよう Oracle HSM を構成します。この例では、*vi* エディタを使用し、*/var/adm/recycler.log* を指定します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
:wq
root@solaris:~#
```

7. テキストエディタで */etc/opt/SUNWsamfs/scripts/recycler.sh* スクリプトを開き、リサイクルしたリムーバブルメディアボリュームを処理するためのシェルコマンドを入力します。

   有効なアーカイブコピーが排出されたリムーバブルメディアボリュームがリサイクル処理で識別されると、リサイクルされたメディアを処理するために設計された C シェルスクリプトである *recycler.sh* ファイルが呼び出されます。

   ボリュームのリサイクルの準備ができたことを管理者に通知するタスク、再利用のためにボリュームを再ラベル付けするタスク、または長期間の履歴保存のためにライブラリからボリュームをエクスポートするタスクなど、必要なタスクを実行するようにファイルを編集します。

   デフォルトでは、スクリプトは、スクリプトを設定するよう *root* ユーザーに通知します。
8. Oracle HSM アーカイブファイルシステムの *mcf* ファイルのアーカイブ装置セクションに、ネットワーク接続テープライブラリが含まれている場合、「ネットワーク接続テープライブラリに格納されているアーカイブメディアのカタログ作成」に進みます。
9. それ以外の場合は、「ファイルシステムの保護の構成」に進みます。

6.2.6.2. ライブラリ単位でのリサイクルの構成

1. Oracle HSM ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. テキストエディタで */etc/opt/SUNWsamfs/recycler.cmd* ファイルを作成します。

   次の例では、*vi* エディタを使用します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#--------------------------------------------------------------------
```

3. *logfile* ディレクティブを使用して、リサイクラログのパスとファイル名を指定します。

   /var/adm/ など、Oracle HSM 以外のディレクトリにログを書き込むよう Oracle HSM を構成します。この例では、*/var/adm/recycler.log* を指定します。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#--------------------------------------------------------------------
logfile = /var/adm/recycler.log
```

4. リサイクルするべきではないボリュームがアーカイブメディアライブラリ内に存在する場合、ディレクティブ *no_recycle media-type volumes* を入力します。ここで、*media-type* は、付録A「*装置タイプの用語集*」で定義されているメディアタイプコードの 1 つで、*volumes* は、1 つ以上のボリュームシリアル番号 (VSN) と一致する正規表現です。

   この例では、[*VOL020-VOL999*] の範囲内のボリュームのリサイクルを無効にします。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#--------------------------------------------------------------------
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
```

5. 新しい行に、ディレクティブ *library parameters* を入力します。ここで、*library* は、*/etc/opt/SUNWsamfs/mcf* ファイルがリムーバブルメディアライブラリに割り当てるファ

ミリセット名で、*parameters* は、次のリストから取得されたパラメータ/値のペアの空白文字区切りリストです。

- *-dataquantity size* は、再アーカイブのために一度にスケジュールできるデータの最大量を *size* に設定します。ここで、*size* はバイト数です。デフォルトは 1G バイトです。
- *-hwm percent* は、使用した場合にリサイクルをトリガーするメディアの合計容量の割合である、ライブラリの*高位境界値*を設定します。高位境界値は、[*0-100*] の範囲内の数値の *percent* として表されます。デフォルトは *95* です。
- *recycler.cmd* ファイルを非破壊的にテストできるように、*-ignore* は、このライブラリがリサイクルされないようにします。
- *-mail address* は、リサイクルメッセージを *address* に送信します。ここで、*address* は有効な電子メールアドレスです。デフォルトでは、メッセージは送信されません。
- *-mingain percent* は、合計容量の割合として表される最小容量以上に、使用可能な空き領域を増やすことができるボリュームのリサイクルを制限します。この最小増量率は、[*0-100*] の範囲内の数値である *percent* として指定されます。デフォルトは、合計容量が 200G バイト未満のボリュームでは *60*、容量が 200G バイト以上のボリュームでは *90* です。
- *-vsncount count* は、一度に再アーカイブ対象としてスケジュールできるボリュームの最大数を *count* に設定します。デフォルトは *1* です。

この例では、ライブラリ *library1* の高位境界値を 95% に設定し、カートリッジごとに 60% の最小容量増量率を必要とします。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#---------------------------------------------------------------------
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
library1 -hwm 95 -mingain 60
```

6. Oracle HSM 構成の一部であるほかのライブラリについて上述の手順を繰り返します。次に、*recycler.cmd* ファイルを保存して、エディタを閉じます。

```
root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for archiving file systems
#---------------------------------------------------------------------
```

```
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
library1 -hwm 95 -mingain 60
:wq
root@solaris:~#
```

7. Oracle HSM アーカイブファイルシステムの *mcf* ファイルのアーカイブ装置セクションにネットワーク接続テープライブラリが含まれている場合、「ネットワーク接続テープライブラリに格納されているアーカイブメディアのカタログ作成」に進みます
8. それ以外の場合は、「ファイルシステムの保護の構成」に進みます。

6.2.7. ネットワーク接続テープライブラリに格納されているアーカイブメディアのカタログ作成

ファイルシステムをマウントしたあと、Oracle HSM ソフトウェアが、*mcf* ファイルで構成されている各自動ライブラリのカタログを作成します。ただし、ネットワーク接続ライブラリがある場合は、カタログを生成するためにいくつかの追加の手順を行う必要があります。

次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. アーカイブファイルシステムが Oracle StorageTek ACSLS 接続テープライブラリを使用する場合、必要な Oracle HSM アーカイブメディアをライブラリのスクラッチプールから取得して、カタログを自動的に生成します。コマンド *samimport -c volumes -s pool* を使用します。ここで、*volumes* は必要なボリュームの数で、*pool* はライブラリに対して定義されているスクラッチメディアプールの名前です。ここで停止します。

   この例では、*scratch* というプールから取得された *20* のテープボリュームを要求します。

```
root@solaris:~# samimport -c 20 -s scratch
```

3. アーカイブファイルシステムが *単一の非共有論理ライブラリとして構成された* IBM 3494 ネットワーク接続ライブラリを使用する場合、必要なテープボリュームをライブラリメールスロット内に配置して、ライブラリで自動的にカタログ化できるようにします。ここで停止します。

mcf ファイルの *Additional Parameters* フィールドに *access=private* が指定されている場合、IBM 3494 ライブラリは単一の論理ライブラリとして構成されます。*access=shared* の場合、IBM 3494 ライブラリは複数の論理ライブラリに分割されるため、次に指定されている方法を使用する必要があります。

4. それ以外の場合、アーカイブファイルシステムが IBM 3494 の共有ネットワーク接続ライブラリまたはその他のネットワーク接続ライブラリを使用するときは、テキストエディタを使用してカタログ入力ファイルを作成します。

   この例では、*vi* エディタを使用して、ファイル *input3494cat* を作成します。

```
root@solaris:~# vi input3494cat
~
"~/input3494cat" [New File]
```

5. レコードの *index* を入力して、レコードを開始します。最初のレコードには常に *0* (ゼロ) を入力し、次に後続のレコードごとにインデックスを増分してください。フィールドの終わりを示すスペースを入力します。

   行は、*build_cat* 入力ファイル内のレコードとスペース区切りフィールドを定義します。最初のフィールドの値である *index* は単に、Oracle HSM カタログ内のレコードを識別する *0* から始まる連続する整数です。この例では、これは最初のレコードであるため、*0* を入力します。

```
0
~
"~/input3494cat" [New File]
```

6. レコードの 2 番目のフィールドには、テープボリュームのボリュームシリアル番号 (VSN) を入力するか、VSN がない場合は単一の *?* (疑問符) を入力します。次に、フィールドの終わりを示すスペースを入力します。

   空白文字がある場合、空白文字が含まれている値を *"VOL 01"* のように二重引用符で囲みます。この例では、最初のボリュームの VSN には空白文字が含まれていません。

```
0 VOL001
~
"~/input3494" [New File]
```

7. 3 番目のフィールドには、ボリュームのバーコード (ボリュームシリアル番号と異なる場合) とボリュームシリアル番号を入力し、ボリュームシリアル番号がない場合は、文字列 *NO_BAR_CODE* を入力します。次に、フィールドの終わりを示すスペースを入力します。

   この例では、最初のボリュームのバーコードの値は VSN と同じです。

```
0 VOL001 VOL001
~
"~/input3494cat" [New File]
```

8. 最後に 4 番目のフィールドに、ボリュームのメディアタイプを入力します。次に、フィールドの終わりを示すスペースを入力します。

   メディアタイプは、*li* (LTO メディアの場合) などの 2 文字のコードです (メディア装置タイプの総合リストについては、付録A「*装置タイプの用語集*」を参照)。この例では、LTO テープドライブを持つ IBM 3494 ネットワーク接続テープライブラリを使用しているため、*li* (最後の空白文字を含める) と入力します。

```
0 VOL001 VOL001 li
~
"~/input3494cat" [New File]
```

9. Oracle HSM で使用するボリュームごとに追加のレコードを作成するには、手順 3 - 6 を繰り返します。次に、ファイルを保存します。

```
0 VOL001 VOL001 li
1 VOL002 VOL002 li
...
13 VOL014 VOL014 li
:wq
root@solaris:~#
```

10. *build_cat input-file catalog-file* コマンドを使用してカタログを作成します。ここで、*input-file* は、入力ファイルの名前で、*catalog-file* は、ライブラリカタログのフルパスです。

    mcf ファイルの *Additional Parameters* フィールドにカタログ名を指定した場合、その名前を使用します。それ以外の場合、カタログを作成しないときは、Oracle HSM

ソフトウェアで、ファイル名 *family-set-name* を使用して */var/opt/SUNWsamfs/catalog/* ディレクトリにデフォルトのカタログが作成されます。ここで、*family-set-name* は、*mcf* ファイルでライブラリに使用する装置名です。この例では、ファミリセット i*3494* を使用します。

```
root@solaris:~# build_cat input_vsns /var/opt/SUNWsamfs/catalog/i3494
```

11. アーカイブファイルシステムが共有されている場合、使用する可能性がある各メタデータサーバーで前述の手順を繰り返します。

    アーカイブファイルシステムが完成し、使用する準備ができました。
12. 次に、ファイルシステムの保護の構成を実行します。

6.2.8. ファイルシステムの保護の構成

ファイルシステムを保護するには、2 つのことを実行する必要があります。

- データが保持されているファイルを保護する必要があります。
- データを使用、整理、検索、アクセス、および管理できるように、ファイルシステム自体を保護する必要があります。

Oracle HSM アーカイブファイルシステムでは、ファイルデータはアーカイバによって自動的に保護されます。変更されたファイルは、テープなどのアーカイブストレージメディアに自動的にコピーされます。ただし、ファイルしかバックアップしていないときに、ディスクデバイスまたは RAID グループに回復不能な障害が発生した場合は、データは保持されますが使用することは難しくなります。代替のファイルシステムの作成、各ファイルの特定、新しいファイルシステム内の適切な場所の決定、そのファイルの取り込み、およびそのファイルとユーザー、アプリケーション、その他のファイルとの間の失われた関係の再作成を行う必要があります。このような回復は、最善の状況でも、面倒で時間のかかるプロセスとなります。

したがって、すばやく効率的に回復するには、ファイルおよびアーカイブコピーを使用可能にするファイルシステムのメタデータを積極的に保護する必要があります。リムーバルメディア上でアーカイブされたコピーに、ディレクトリパス、i ノード、アクセス制御、シンボリックリンク、およびポインタをバックアップする必要があります。

Oracle HSM ファイルシステムのメタデータを保護するには、*回復ポイント*をスケジュールし、アーカイブログを保存します。回復ポイントは、Oracle HSM ファイルシステムのメタデータのポイントインタイムバックアップコピーを格納する圧縮ファイルです。データの損失 (ユーザーファイルの誤った削除から、ファイルシステム全体の壊滅的な損失まで) が発生した場合は、

ファイルまたはファイルシステムが元の状態のままである最新の回復ポイントを見つければ、即座にファイルまたはファイルシステムの既知の良好な最新状態まで回復できます。次に、その時点で記録されたメタデータを復元し、メタデータに示されているファイルをアーカイブメディアからディスクキャッシュにステージングするか、または可能であれば、ユーザーおよびアプリケーションがファイルにアクセスするときに必要に応じてファイルシステムでファイルをステージングするようにします。

ポイントインタイムバックアップコピーと同様に、回復ポイントが障害が発生した時点のファイルシステムの状態の完全なレコードであることは、ほとんどありません。必然的に、1 つの回復ポイントが完成してから、次の回復ポイントが作成されるまで、少なくとも数個のファイルが作成および変更されます。ファイルシステムが使用されていないときに回復ポイントを頻繁に作成するようにスケジュールすれば、この問題を最小限にできます。ただし、現実には、ファイルシステムは使用するために存在するため、スケジューリングには妥協が必要です。

このため、アーカイバログファイルのポイントインタイムコピーを保存する必要もあります。それぞれのデータファイルがアーカイブされると、ログファイルには、アーカイブメディアのボリュームシリアル番号、アーカイブセットとコピー番号、メディアでのアーカイブ (*tar*) ファイルの位置、および *tar* ファイル内でのデータファイルのパスと名前が記録されます。この情報があれば、Solaris または Oracle HSM *tar* ユーティリティーを使用して、失われたファイルを回復ポイントから回復できます。ただし、この情報は変動します。大部分のシステムログと同様に、アーカイバログは急速に増加するため、頻繁に上書きされてしまいます。定期的にコピーして回復ポイントを補完していなければ、必要なときにログ情報がないことになります。

そのため、ファイルシステムの保護にはいくつかの計画が必要です。一方で、回復ポイントとログファイルのコピーを十分な頻度で作成して、失われたり破損したりしたファイルとファイルシステムを回復するために最適な機会を得るのに十分な期間保持する必要があります。他方、データファイルが頻繁に変更されていて、使用しているディスク領域を認識する必要があるときは、回復ポイントとログファイルのコピーを作成することは望ましくありません (回復ポイントファイルとログが大きい可能性があります)。そのため、このセクションでは、変更なしで多数のファイルシステム構成で使用でき、幅広く適用可能な構成を推奨しています。変更が必要な場合は、推奨される構成に問題が説明されており、開始点として優れた役割を果たします。このセクションの残りでは、回復ポイントを作成および管理する手順について説明します。次のサブセクションが含まれます。

- 回復ポイントファイルとアーカイバログのコピーを格納する場所の作成
- 回復ポイントの自動作成とアーカイバログの保存

6.2.8.1. 回復ポイントファイルとアーカイバログのコピーを格納する場所の作成

構成したアーカイブファイルシステムごとに、次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. 回復ポイントファイルのストレージ場所を選択します。ファイルシステムのホストにマウントできる独立したファイルシステムを選択します。
3. 新しい回復ポイントファイルと、指定した任意の時点で保持する予定の数の回復ポイントファイルの両方を格納するのに十分な領域を、選択したファイルシステムに確保してください。

   回復ポイントファイルは大きいサイズになる可能性があり、作成する頻度と保持する期間によっては、多数のファイルを格納する必要があります。
4. 選択したファイルシステムが、どの物理デバイスもアーカイブファイルシステムと共有しないようにしてください。

   保護対象のファイルシステムに回復ポイントファイルを格納しないでください。アーカイブファイルシステムもホストしている物理デバイス上にある論理デバイス (パーティションや LUN など) に回復ポイントファイルを格納しないでください。
5. 選択したファイルシステムで、回復ポイントファイルを保持するディレクトリを作成します。コマンド *mkdir mount-point/path* を使用します。ここで、*mount-point* は、選択した独立ファイルシステム用のマウントポイント、*path* は、選択したディレクトリのパスと名前です。

   複数のアーカイブファイルシステムの回復ポイントファイルを単一のキャッチオールディレクトリに格納しないでください。回復ポイントファイルが編成され、必要に応じて簡単に見つけられるように、それぞれに別個のディレクトリを作成してください。

   この例では、アーカイブファイルシステム */samms* 用の回復ポイントを構成します。そのため、独立ファイルシステム */zfs1* にディレクトリ */zfs1/samms_recovery* を作成しました。

```
root@solaris:~# mkdir /zfs1/samms_recovery
```

6. ファイルシステムでアーカイブファイルシステムと物理デバイスを共有しない場合、ファイルシステムのアーカイバログのポイントインタイムコピーを格納するためのサブディレクトリを作成します。

   この例では、ログのコピーをホストのルートファイルシステムの */var* ディレクトリに格納することを選択します。アーカイブファイルシステム */samms* のファイルシステムの保護を構成します。そのため、ディレクトリ */var/samms_archlogs* を作成します。

```
root@solaris:~# mkdir /var/samms_archlogs
```

7. 次に、回復ポイントの自動作成とアーカイバログの保存を実行します。

6.2.8.2. 回復ポイントの自動作成とアーカイバログの保存

crontab ファイルでエントリを作成するか、Oracle HSM Manager のグラフィカルユーザーインタフェースのスケジューリング機能を使用することで、メタデータの回復ポイントファイルを自動的に作成できますが、後者の方法ではアーカイバログデータは自動的には保存されません。そのため、このセクションでは、*crontab* の方法に焦点を当てます。グラフィカルユーザーインタフェースを使用して回復ポイントをスケジュールする場合、Manager のオンラインヘルプを参照してください。

次の手順では、毎日実行される 2 つの *crontab* エントリを作成します。1 つは、期限切れの回復ポイントファイルを削除してから新しい回復ポイントを作成し、もう 1 つは、アーカイバログを保存します。構成したアーカイブファイルシステムごとに、次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. 編集のために *root* ユーザーの *crontab* ファイルを開きます。コマンド *crontab -e* を使用します。

   crontab コマンドは、*root* ユーザーの *crontab* ファイルの編集可能なコピーを、*EDITOR* 環境変数で指定されたテキストエディタで開きます (詳細は、Solaris *crontab* のマニュアルページを参照してください)。この例では、*vi* エディタを使用します。

```
root@solaris:~# crontab -e
...
# The root crontab should be used to perform accounting data collection.
```

```
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
```

3. 最初に、期限切れの回復ポイントファイルを削除して新しい回復ポイントを作成するエントリを作成します。新しい行で、作業を行う時間を指定します。*minutes hour* `* * *` と入力します。ここでは:
   - *minutes* は、ジョブを開始する分を指定する、[*0-59*] の範囲内の整数です。
   - *hour* は、ジョブを開始する時を指定する、[*0-23*] の範囲内の整数です。
   - `*` (アスタリスク) は未使用の値を指定します。

     毎日実行されるタスクの場合は、日 [*1-31*]、月 [*1-12*]、および曜日 [*0-6*] の値は未使用です。
   - 空白は、時間の指定のフィールドを区切ります。
   - *minutes hour* は、ファイルが作成または変更されていない時間を指定します。

     ファイルシステムのアクティビティーが最小限のときに回復ポイントファイルを作成すると、アーカイブの状態が可能なかぎり正確かつ完全にファイルに反映されます。新しいファイルと変更されたファイルのすべてが、指定した時間の前にアーカイブされていることが理想です。

   この例では、作業が毎日午前 2:10 に開始されるようにスケジュールします。

```
...
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * *
```

4. 続けて同じ行に、古い回復ポイントファイルをクリーンアップするシェルコマンドを入力します。テキスト *(find directory -type f -mtime +retention -print | xargs -l1 rm -f;* を入力します。ここでは:
   - *(* (左括弧) は、*crontab* エントリによって実行されるコマンドシーケンスの開始をマークします。
   - *directory* は、回復ポイントファイルが格納されるディレクトリのパスとディレクトリ名であり、Solaris *find* コマンドで検索を開始するポイントです。

- *-type f* は、プレーンファイル (ブロック特殊ファイル、文字特殊ファイル、ディレクトリ、パイプなどの反対) を指定する *find* コマンドオプションです。
- *-mtime +retention* は、*retention* (回復ポイントファイルが保持される時間の数を表す整数) を超えて変更されていないファイルを指定する *find* コマンドオプションです。
- *-print* は、見つかったすべてのファイルを標準出力に一覧表示する *find* コマンドオプションです。
- *|xargs -l1 rm -f* は、 *-print* の出力を Solaris コマンド *xargs -l1* にパイプ処理します。このコマンドは、一度に 1 行を引数として Solaris コマンド *rm -f* に送信し、次のこのコマンドが検出されたそれぞれのファイルを削除します。
- *;* (セミコロン) は、コマンド行の終わりをマークします。

この例では、*crontab* エントリは、72 時間 (3 日) 以上変更されていないファイルをディレクトリ */zfs1/samms_recovery* で検索し、見つかったファイルをすべて削除します。*crontab* エントリは引き続き 1 行です。改行はバックスラッシュでエスケープされます。

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /zfs1/samms_recovery -type f -mtime +72 -print | /
xargs -l1 rm -f;
```

5. 同じ行で続けて、回復ポイントが作成されるディレクトリに変更するシェルコマンドを入力します。テキスト *cd mount-point;* を入力します。ここで、*mount-point* は、アーカイブファイルシステムのルートディレクトリで、セミコロン (;) はコマンド行の終わりをマークします。

   回復ポイントファイルを作成するコマンド *samfsdump* は、現在のディレクトリとすべてのサブディレクトリ内にあるすべてのファイルのメタデータをバックアップします。この例では、保護するファイルシステムのマウントポイントである */samms* ディレクトリに移動します。

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
```

```
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /zfs1/samms_recovery -type f -mtime +72 -print | /
xargs -l1 rm -f; cd /samms;
```

6. 同じ行に続けて、毎日新しい回復ポイントを作成するシェルコマンドを入力します。テキスト */opt/SUNWsamfs/sbin/samfsdump -f directory/'date +/%y/%m/%d')* を作成します。ここでは:
   - */opt/SUNWsamfs/sbin/samfsdump* は、回復ポイントを作成するコマンドです (詳細は、マニュアルページを参照してください)。
   - *-f* は、回復ポイントファイルを保存する場所を指定する *samfsdump* コマンドオプションです。
   - *directory* は、このファイルシステムの回復ポイントを保持するために作成したディレクトリです。
   - *'date +/%y/%m/%d'* は、Solaris *date* コマンドと、回復ポイントファイルの名前 *YYMMDD* を作成するフォーマットテンプレートです。ここで、*YYMMDD* は、現在の年の最後の 2 桁、現在の月の 2 桁の数値、2 桁の日 (たとえば、*150122* は、2015 年 1 月 22 日) です。
   - ;(セミコロン) は、コマンド行の終わりをマークします。
   - *)* (右括弧) は、*crontab* エントリによって実行されるコマンドシーケンスの終わりをマークします。

   この例では、上で作成した回復ポイントディレクトリ */zfs1/samms_recovery* を指定します。*crontab* エントリは引き続き 1 行です。改行はバックスラッシュでエスケープされます。

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /zfs1/samms_recovery -type f -mtime +72 -print | /
xargs -l1 rm -f; cd /samms ; /opt/SUNWsamfs/sbin/samfsdump /
-f /zfs1/samms_recovery/'date +/%y/%m/%d')
```

7. 次に、アーカイバログを保存するエントリを作成します。新しい行に、*minutes hour * * **を入力して、作業を行う時間を指定します。ここでは:

   - *minutes* は、ジョブを開始する分を指定する、[*0-59*] の範囲内の整数です。
   - *hour* は、ジョブを開始する時を指定する、[*0-23*] の範囲内の整数です。
   - ** (アスタリスク) は未使用の値を指定します。

     毎日実行されるタスクの場合は、日 [*1-31*]、月 [*1-12*]、および曜日 [*0-6*] の値は未使用です。
   - 空白は、時間の指定のフィールドを区切ります。
   - *minutes hour* は、ファイルが作成または変更されていない時間を指定します。

   この例では、作業が毎週日曜日午前 3:15 に開始されるようにスケジュールします。

```
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /zfs1/samms_recovery -type f -mtime +72 -print | /
xargs -l1 rm -f; cd /samms ; /opt/SUNWsamfs/sbin/samfsdump /
-f /zfs1/samms_recovery/'date +/%y/%m/%d')
15 3 * * 0
```

8. 同じ行に続けて、現在のアーカイバログをバックアップの場所に移動して一意の名前を指定するシェルコマンドを入力します。テキスト *(mv /var/adm/samms.archive .log /var/samms_archlogs/"date +%y%m%d";* を入力します。

   この手順によって、アクティブなログファイルに残された場合に上書きされるログエントリが保存されます。例では、*samms* ファイルシステムのアーカイバログを選択した場所 */var/samms_archlogs/* に移動して、*YYMMDD* に名前変更します。ここで、*YYMMDD* は、現在の年の最後の 2 桁、現在の月の 2 桁の数値、および 2 桁の日 (たとえば、*150122* は、2015 年 1 月 22 日です) です。

```
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /samms_recovery/dumps -type f -mtime +72 -print | xargs -l1 rm -f; / cd /samms ; /
opt/SUNWsamfs/sbin/samfsdump -f /zfs1/samms_recovery/'date +/%y/%m/%d')
15 3 * * 0 ( mv /var/adm/samms.archiver.log /var/samms_archlogs/"date +%y%m%d";
```

9. 同じ行に続けて、アーカイバログファイルを再初期化するシェルコマンドを入力します。テキスト *touch /var/adm/samms.archive.log)* を入力します。

   この例では、*crontab* エントリは引き続き 1 行です。改行はバックスラッシュでエスケープされます。

```
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /samms_recovery/dumps -type f -mtime +72 -print | xargs -l1 rm -f; / cd /samms ; /
opt/SUNWsamfs/sbin/samfsdump -f /zfs1/samms_recovery/'date +/%y/%m/%d')
15 3 * * 0 ( mv /var/adm/samms.archive.log /var/samms_archlogs/"date +%y%m%d";/
touch /var/adm/samms.archiver.log )
```

10. ファイルを保存して、エディタを閉じます。

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
15 3 * * 0 [ -x /usr/lib/fs/nfs/nfsfind ] && /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 2 * * * ( find /samms_recovery/dumps -type f -mtime +72 -print | xargs -l1 rm -f; / cd /samms ; /
opt/SUNWsamfs/sbin/samfsdump -f /zfs1/samms_recovery/'date +/%y/%m/%d')
15 3 * * 0 ( mv /var/adm/samms.archive.log /var/samms_archlogs/"date +%y%m%d";/ touch /var/adm/samms
.archive.log )
:wq
root@solaris:~#
```

11. ファイルシステムで WORM (書き込み 1 回、読み取り複数回) 機能を有効にする必要がある場合、「Write Once Read Many (WORM) ファイルのサポートの有効化」を参照してください。
12. LTFS を使用するシステムと相互作業する必要がある場合、またはリモートサイト間で大量のデータを転送する必要がある場合、「Linear Tape File System (LTFS) のサポートの有効化」を参照してください。
13. アーカイブテープボリュームの整合性を検証できる必要がある場合、「アーカイブメディア検証の構成」に進みます。
14. 複数ホストのファイルシステムアクセスや高可用性構成などの追加の要件がある場合、「応用編」を参照してください。

15. それ以外の場合は、11章「*通知とロギングの構成*」に進みます。

6.2.9. アーカイブメディア検証の構成

メディア検証は、SCSI *verify* コマンドを使用してテープメディアのデータ整合性を評価する手法です。ホストの SCSI ドライバは、ドライブに書き込むデータの論理ブロックの CRC チェックサムを計算して、*verify* コマンドを送信します。ドライブはデータブロックを読み取り、独自のチェックサムを計算して、結果をドライバによって提供された値と比較します。矛盾があった場合はエラーを返します。ドライブは、チェックサムが完了すると読み取ったデータをすぐに破棄するため、ホストで追加の入出力関連のオーバーヘッドは発生しません。

Oracle HSM では、次の 2 つの方法でのメディア検証がサポートされます。

- データ整合性検証 (DIV) をサポートするための Oracle HSM の構成を行なって、Oracle HSM の定期的なメディア検証によって、StorageTek T10000 テープメディアでのデータ検証を手動または自動で行うことができます。
- また、Oracle HSM の定期的なメディア検証の構成を行なって、StorageTek T10000 テープメディアと、LTO Ultrium などのその他のフォーマットの両方にあるデータを自動的に検証することもできます。

6.2.9.1. データ整合性検証 (DIV) をサポートするための Oracle HSM の構成

データ整合性検証 (DIV) は、Oracle StorageTek テープドライブの機能であり、格納されているデータの整合性を保証するために Oracle HSM software とともに機能します。この機能が有効になっている (*div = on* または *div = verify*) 場合、サーバーホストとドライブの両方が、入出力中にチェックサムを計算して比較します。書き込み操作中に、サーバーは、データブロックごとに 4 バイトチェックサムを計算して、チェックサムをデータとともにドライブに渡します。次に、テープドライブはチェックサムを再計算し、結果をサーバーによって提供された値と比較します。値が一致した場合、ドライブはデータブロックとチェックサムの両方をテープに書き込みます。読み取り操作中に、ドライブとホストの両方が、データブロックとその関連するチェックサムをテープからを読み取ります。それぞれが、データブロックからチェックサムを再計算し、結果を格納されているチェックサムと比較します。どの時点でもチェックサムが一致しない場合、ドライブは、エラーが発生したことをアプリケーションソフトウェアに通知します。

div = verify オプションは、データの書き込み時に保護機能を強化します。書き込み操作が完了すると、ホストはデータを再検証するようテープドライブに指示します。次に、ドライブはデータを再スキャンし、チェックサムを再計算して、結果をテープに格納されているチェック

サムと比較します。ドライブは、すべての操作を内部で実行し、追加の入出力は行われない (データは破棄されます) ため、ホストシステムで追加のオーバーヘッドは発生しません。必要に応じて、Oracle HSM *tpverify* (tape-verify) コマンドを使用して、この手順を実行することもできます。

データ整合性検証を構成するには、次のように進めます。

1. Oracle HSM サーバーに *root* としてログインします。

   この例では、メタデータサーバーの名前は *samfs-mds* です。

```
[samfs-mds]root@solaris:~#
```

2. メタデータサーバーで Oracle Solaris 11 以上が実行されていることを確認します。

```
[samfs-mds]root@solaris:~# uname -r
5.11
[samfs-mds]root@solaris:~#
```

3. Oracle HSM *mcf* ファイルで定義されているアーカイブストレージ装置に、互換性のあるテープドライブである StorageTek T10000C (最小のファームウェアレベルは 1.53.315) または T10000D が含まれていることを確認します。
4. アーカイブプロセスがある場合、すべてアイドル状態にします。コマンド *samcmd aridle* を使用します。

   このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

```
[samfs-mds]root@solaris:~# samcmd aridle
[samfs-mds]root@solaris:~#
```

5. ステージングプロセスがある場合、すべてアイドル状態にします。コマンド *samcmd stidle* を使用します。

   このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

```
[samfs-mds]root@solaris:~# samcmd stidle
[samfs-mds]root@solaris:~#
```

6. アクティブなアーカイブジョブが完了するまで待機します。コマンド *samcmd a* を使用して、アーカイブプロセスのステータスを確認します。

   アーカイブプロセスが *Waiting for :arrun* の場合、アーカイブプロセスはアイドル状態になっています。

```
[samfs-mds]root@solaris:~# samcmd a
Archiver status samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samfs-mds
sam-archiverd:  Waiting for :arrun
sam-arfind: ...
Waiting for :arrun
```

7. アクティブなステージングジョブが完了するまで待機します。コマンド *samcmd u* を使用してステージングプロセスのステータスを確認します。

   ステージングプロセスが *Waiting for :strun* の場合、ステージングプロセスはアイドル状態になっています。

```
[samfs-mds]root@solaris:~# samcmd u
Staging queue samcmd      6.0 14:20:34 Feb 22 2015
samcmd on solaris.demo.lan
Staging queue by media type: all
sam-stagerd:  Waiting for :strun
root@solaris:~#
```

8. すべてのリムーバブルメディアドライブをアイドル状態にしてから、続行します。ドライブごとに、コマンド *samcmd equipment-number idle* を使用します。ここで *equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイル内のドライブに割り当てられている装置の順序番号です。

   このコマンドはドライブを「*off*」にする前に、現在のアーカイブジョブおよびステージングジョブを完了できますが、新しいジョブは開始されません。この例では、4 つのドライブ (順序番号 *801*、*802*、*803*、*804*) をアイドル状態にします。

```
[samfs-mds]root@solaris:~# samcmd 801 idle
[samfs-mds]root@solaris:~# samcmd 802 idle
```

```
[samfs-mds]root@solaris:~# samcmd 803 idle
[samfs-mds]root@solaris:~# samcmd 804 idle
[samfs-mds]root@solaris:~#
```

9. 実行中のジョブが完了するまで待機します。

   コマンド *samcmd r* を使用すると、ドライブのステータスを確認できます。すべてのドライブが「*notrdy*」または「*empty*」の場合は、続行できる状態になっています。

```
[samfs-mds]root@solaris:~# samcmd r
Removable media samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samqfs1host
ty   eq   status      act  use  state  vsn
li  801   ---------p    0   0%  notrdy
          empty
li  802   ---------p    0   0%  notrdy
          empty
li  803   ---------p    0   0%  notrdy
          empty
li  804   ---------p    0   0%  notrdy
          empty
[samfs-mds]root@solaris:~#
```

10. アーカイバおよびステージャープロセスがアイドル状態で、テープドライブがすべて「*notrdy*」になっている場合は、ライブラリ制御デーモンを停止します。コマンド *samd stop* を使用します。

```
[samfs-mds]root@solaris:~# samd stop
[samfs-mds]root@solaris:~#
```

11. */etc/opt/SUNWsamfs/defaults.conf* ファイルをテキストエディタで開きます。必要に応じて、行 *#div = off* のコメントを解除するか、この行が存在しない場合は追加します。

    デフォルトでは、*div* (データ整合性検証) は *off* (無効) です。

    この例では、*vi* エディタでファイルを開き、行のコメントを解除します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
div = off
```

12. データ整合性検証の読み取り、書き込み、および検証操作を有効にするには、行 *#div = off* を *div = on* に変更して、ファイルを保存します。

    各ブロックの書き込みおよび読み取りが行われるたびにデータが検証されますが、Oracle HSM アーカイバソフトウェアは、ファイルコピーがアーカイブされたあとで完全なファイルコピーを検証しません。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
div = on
:wq
[samfs-mds]root@solaris:~#
```

13. データ整合性検証機能の書き込み後の検証オプションを有効にするには、行 *#div = off* を *div = verify* に変更して、ファイルを保存します。

    各ブロックの書き込みまたは読み取りが行われると、ホストとドライブはデータ整合性検証を実行します。さらに、完全なアーカイブ要求がテープに書き込まれるたびに、ドライブは新たに格納されたデータとチェックサムを再度読み取り、再計算して、計算結果を格納されている結果と比較します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
div = verify
```

```
:wq
[samfs-mds]root@solaris:~#
```

14. *defaults.conf* ファイルを再度読み取り、それに従って Oracle HSM software 自体を再構成するようにこのソフトウェアに指示します。*samd config* コマンドを使用します。

```
[samfs-mds]root@solaris:~# /opt/SUNWsamfs/sbin/samd config
```

15. 前の手順で Oracle HSM 操作を停止した場合、この時点で *samd start* コマンドを使用して再開します。

```
[samfs-mds]root@solaris:~# samd start
[samfs-mds]root@solaris:~#
```

これで、データ整合性検証が構成されました。

16. データ整合性検証を自動化する必要がある場合、「Oracle HSM の定期的なメディア検証の構成」に進みます。
17. ファイルシステムで WORM (書き込み 1 回、読み取り複数回) 機能を有効にする必要がある場合、「Write Once Read Many (WORM) ファイルのサポートの有効化」を参照してください。
18. LTFS を使用するシステムと相互作業する必要がある場合、またはリモートサイト間で大量のデータを転送する必要がある場合、「Linear Tape File System (LTFS) のサポートの有効化」を参照してください。
19. 複数ホストのファイルシステムアクセスや高可用性構成などの追加の要件がある場合、「応用編」を参照してください。

6.2.9.2. Oracle HSM の定期的なメディア検証の構成

Oracle HSM アーカイブファイルシステムの定期的なメディア検証 (PMV) を設定できます。定期的なメディア検証は、ファイルシステム内のリムーバブルメディアのデータ整合性を自動的にチェックします。StorageTek データ整合性検証を使用して StorageTek T10000 メディアをチェックし、幅広くサポートされる SCSI *verify(6)* コマンドを使用してその他のドライブをチェックします。

定期的なメディア検証機能は、定期的に *tpverify* コマンドを適用する Oracle HSM デーモン *verifyd* の追加、検出されたエラーの記録、管理者への通知、および指定された回復アクションの自動的な実行を行います。定期的なメディア検証を構成するには、構成ファイ

ル *verifyd.cmd* でポリシーディレクティブを設定します。ポリシーでは、検証スキャンが実行される時間、行われるスキャンのタイプ、使用できるライブラリとドライブ、スキャンするべきテープボリューム、およびエラーの検出時に Oracle HSM で行うアクションを指定できます。Oracle HSM は、たとえば、エラーが含まれているファイルを自動的に再アーカイブしたり、エラーが含まれているテープボリュームをリサイクルしたりできます。

1. まだ行なっていない場合は、データ整合性検証 (DIV) をサポートするための Oracle HSM の構成を行います。
2. Oracle HSM サーバーに *root* としてログインします。

   この例では、メタデータサーバーの名前は *samfs-mds* です。

```
[samfs-mds]root@solaris:~#
```

3. メタデータサーバーで Oracle Solaris 11 以上が実行されていることを確認します。

```
[samfs-mds]root@solaris:~# uname -r
5.11
[samfs-mds]root@solaris:~#
```

4. テキストエディタで */etc/opt/SUNWsamfs/verifyd.cmd* ファイルを開きます。

   この例では、*vi* エディタでファイルを開きます。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
# For additional information about the format of the verifyd.cmd file,
# type "man verifyd.cmd".
# Enable Oracle HSM Periodic Media Validation (PMV)
pmv = off
```

5. 定期的なメディア検証を有効にするには、行 *pmv = on* を入力します。

   デフォルトでは、定期的なメディア検証は *off* です。この例では、これを *on* に設定します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
# For additional information about the format of the verifyd.cmd file,
# type "man verifyd.cmd".
# Enable Oracle HSM Periodic Media Validation (PMV)
```

```
pmv = on
```

6. 実行時間を設定します。行 *run_time = always* (連続して検証を実行する場合)、または *run_time = HHMM hhmm DD dd* を入力します。ここで、*HHMM* と *hhmm* は、それぞれ開始時間と終了時間で、*DD dd* は、オプションの開始日と終了日です。

   HH と *hh* は、*00-24* の範囲内の時間、*MM* と *mm* は、*00-60* の範囲内の分数、*DD* と *dd* は、*[0-6]* の範囲内の曜日で、*0* は日曜日、*6* は土曜日です。デフォルトは *2200 0500 6 0* です。

   ただし、検証は、重要なファイルシステム操作とはすぐには競合しません。検証プロセスによって、アーカイバとステージャーで必要なテープボリュームやドライブを自動的に得られます。そのため、この例では、実行時間を *always* に設定します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
# For additional information about the format of the verifyd.cmd file,
# type "man verifyd.cmd".
# Enable Oracle HSM Periodic Media Validation (PMV)
pmv = on
# Run all of the time.  PMV will yield VSNs and drives when
# resources are wanted by the SAM-QFS archiver and stager.
run_time = always
```

7. 検証方法を指定します。行 *pmv_method = specified-method* を入力します。ここで、*specified-method* は次のいずれかです。

   - *standard* メソッドは、Oracle StorageTek T10000C 以降のテープドライブで使用します。速度のために最適化された *standard* メソッドでは、メディアの端、先頭、終わり、および最初の 1,000 ブロックが検証されます。
   - *complete* メソッドも、Oracle StorageTek T10000C 以降のテープドライブで使用します。これは、メディアのすべてのブロックでメディアエラー訂正コード (ECC) を検証します。
   - *complete plus* メソッドも、Oracle StorageTek T10000C 以降のテープドライブで使用します。これは、メディアの各ブロックについてメディアのエラー訂正コード (ECC) とデータ整合性検証チェックサムの両方を検証します (「データ整合性検証 (DIV) をサポートするための Oracle HSM の構成」を参照)。
   - *legacy* メソッドは、その他すべてのテープドライブで使用でき、メディアがカタログ内で不良とマークされているときと、*verifyd.cmd* ファイルで指定された方法がドライ

ブでサポートされないときに自動的に使用されます。これは、6 バイトの固定ブロックモードの SCSI 検証コマンドを実行し、前に記録された不具合をスキップします。新たに永続的なメディアエラーが見つかった場合、*legacy* メソッドは次のファイルにスキップし、新たに検出されたエラーをメディア不具合データベースに記録します。

- メディア情報領域 (MIR) が欠落または破損している場合に、*mir rebuild* メソッドは、Oracle StorageTek テープカートリッジの MIR を再構築します。これは、メディアカタログで不良とマークされているメディアで機能し、MIR の破損が検出された場合に自動的に指定されます。

この例では、LTO ドライブを使用するため、*legacy* を指定します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
# resources are wanted by the SAM-QFS archiver and stager.
run_time = always
pmv_method = legacy
```

8. 使用可能なすべてのライブラリとドライブを検証に使用するには、行 *pmv_scan = all* を入力します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = all
```

9. 指定したライブラリ内のすべての使用可能なドライブを検証に使用するには、行 *pmv_scan = library equipment-number* を入力します。ここで、*equipment-number* は、ファイルシステムの *mcf* ファイルでライブラリに割り当てられた装置番号です。

   この例では、ライブラリ *800* 内のすべてのドライブを検証プロセスで使用できるようにします。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = library 800
```

10. 指定したライブラリ内で検証プロセスが使用できるドライブの数を制限するには、行 *pmv_scan = library equipment-number max_drives number* を使用します。ここで、*equipment-number* は、ファイルシステムの *mcf* ファイルでライブラリに割り当てられた装置番号で、number は、使用できるドライブの最大数です。

    この例では、ライブラリ *800* 内の最大 *2* 台のドライブを検証プロセスで使用できるようにします。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = library 800 max_drives 2
```

11. 指定したライブラリ内で検証プロセスが使用できるドライブを指定するには、行 *pmv _scan = library equipment-number drive drive-numbers* を入力します。ここで、*equipment-number* は、ファイルシステムの *mcf* ファイルでライブラリに割り当てられた装置番号で、*drive-numbers* は、*mcf* ファイルで指定されたドライブに割り当てられた装置番号の空白文字区切りリストです。

    この例では、ライブラリ *900* 内のドライブ *903* と *904* を検証プロセスで使用できるようにします。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = library 900 drive 903 904
```

12. 複数のライブラリで検証プロセスが使用できるドライブを指定するには、行 *pmv_scan = library-specification library-specification...* を入力します。ここで、*equipment-number* は、ファイルシステムの *mcf* ファイルでライブラリに割り当てられた装置番号で、*drive-numbers* は、*mcf* ファイルでの指定に割り当てられた装置番号のスペース区切りリストです。

    この例では、ライブラリ *800* 内の最大 *2* 台のドライブと、ライブラリ *900* 内のドライブ *903* と *904* を検証プロセスで使用できるようにします。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
```

```
pmv_method = legacy
pmv_scan = library 800 max_drives 2 library 900 drive 903 904
```

13. 定期的なメディア検証を無効にして、どの装置も使用されないようにするには、行 *pmv_scan = off* を入力します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_method = legacy
pmv_scan = off
```

14. 指定された数の永続的なエラーが定期的なメディア検証で検出されたあとで、リサイクル対象としてメディアに自動的にフラグを設定するには、行 *action = recycle perms number-errors* を入力します。ここで、*number-errors* はエラーの数です。

    この例では、*10* 個のエラーが検出されたあとで、リサイクル対象としてメディアにフラグを設定するように Oracle HSM を構成します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = recycle perms 10
```

15. 指定された期間エラーが累積されたあとで、不良ブロックが含まれているファイルを自動的に再アーカイブするには、行 *action = rearch age time* を入力します。ここで、*time* は、*SECONDSs*、*MINUTESm*、*HOURSh*、*DAYSd*、または *YEARSy* の任意の組み合わせのスペース区切りリストで、*SECONDS*、*MINUTES*、*HOURS*、*DAYS*、および *YEARS* は整数です。

    もっとも古いメディアの不具合が指定の期間経過したあとで、アーカイブが必要なファイルがファイルシステムでスキャンされます。この例では、再アーカイブ経過時間を *1* 分に設定します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = rearch age 1m
```

16. 永続的なメディアエラーが定期的なメディア検証で検出されたときにメディアを不良とマークして、それ以外の場合はアクションを行わない場合は、行 *action = none* を入力します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = none
```

17. 定期的に検証するべきテープボリュームを指定します。行 *pmv_vsns = selection-criterion* を入力します。ここで、*selection-criterion* は、*all* か、1 つ以上のボリュームシリアル番号 (VSN) を指定する正規表現のスペース区切りリストです。

    デフォルトは *all* です。この例では、3 つの正規表現を指定します。*^VOL0[01][0-9]* と *^VOL23[0-9]* は、それぞれ *VOL000* - *VOL019* および *VOL230* - *VOL239* の範囲内のボリュームシリアル番号を持つ 2 つのボリュームのセットを指定し、*VOL400* は、特定のボリュームシリアル番号を持つボリュームを指定します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = none
pmv_vsns = ^VOL0[01][0-9] ^VOL23[0-9] VOL400
```

    ボリュームが監査を必要とする場合、リサイクル対象としてスケジュールする場合、使用不可の場合、外部 (Oracle HSM 以外の) ボリュームである場合、またはデータを含んでいない場合、Oracle HSM はボリュームの検証を試行しません。クリーニングカートリッジ、ラベルなしのボリューム、および重複したボリュームシリアル番号が付いたボリュームも除外されます。

18. 必要な検証ポリシーを定義します。行 *pmv_policy = verified age vertime [modified age modtime] [mounted age mnttime]* を入力します。ここでは:
    - *verified age* は、ボリュームが最後に検証されてから経過している必要がある最小時間を指定します。
    - *modified age* (オプション) は、ボリュームが最後に変更されてから経過している必要がある最小時間を指定します。

- *mounted age* (オプション) は、ボリュームが最後にマウントされてから経過している必要がある最小時間を指定します。
- パラメータ値 *vertime*、*modtime*、および *mnttime* は、負ではない整数と時間単位 *y* (年)、*m* (月)、*d* (日)、*H* (時)、*M* (分)、および *S* (秒) の組み合わせです。

Oracle HSM は、ボリュームが最後に検証され、オプションで変更またはマウントされてから経過した時間に基づいて、検証の候補を識別して、ランク付けします。デフォルトのポリシーは、単一のパラメータである *verified age 6m* (6 か月) です。この例では、最終の verified age を 3 か月に設定し、最終の modified age を 15 か月に設定します。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_scan = all
action = none
pmv_vsns = ^VOL0[01][0-9] ^VOL23[0-9] VOL400
pmv_policy = verified age 3m modified age 15m
```

19. */etc/opt/SUNWsamfs/verifyd.cmd* ファイルを保存して、エディタを閉じます。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/verifyd.cmd
...
pmv_vsns = ^VOL0[01][0-9] ^VOL23[0-9] VOL400
pmv_policy = verified age 3m modified age 15m
:wq
root@solaris:~#
```

20. *tpverify -x* コマンドを入力して、*verifyd.cmd* ファイルでエラーを確認します。見つかったエラーを修正します。

tpverify -x コマンドは *verifyd.cmd* を読み取って、エラーが検出された場合は停止します。

```
root@solaris:~# tpverify -x
Reading '/etc/opt/SUNWsamfs/verifyd.cmd'.
PMV: off
     Run-time:
     Start Time: 2200
End Time: 0500
```

```
PMV Scan: all
PMV Method: legacy
STA Scan: off
Action: none
PMV VSNs: all
PMV Policy:
     Last Verified Age: 6m
root@solaris:~#
```

21. 新しい *verifyd.cmd* ファイルを使用して、検証サービスを再開します。*tpverify -r* コマンドを入力します。

```
root@solaris:~# tpverify -r
root@solaris:~#
```

定期的なメディア検証の構成が終了しました。

22. ファイルシステムで WORM (書き込み 1 回、読み取り複数回) 機能を有効にする必要がある場合、「Write Once Read Many (WORM) ファイルのサポートの有効化」を参照してください。
23. LTFS を使用するシステムと相互作業する必要がある場合、またはリモートサイト間で大量のデータを転送する必要がある場合、「Linear Tape File System (LTFS) のサポートの有効化」を参照してください。
24. 複数ホストのファイルシステムアクセスや高可用性構成などの追加の要件がある場合、「応用編」を参照してください。
25. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

6.3. Write Once Read Many (WORM) ファイルのサポートの有効化

WORM (Write Once Read Many) ファイルは、法律上およびアーカイブ上の理由で多くのアプリケーションで使用されます。WORM 対応の Oracle HSM ファイルシステムでは、デフォルトおよびカスタマイズ可能なファイル保持期間、データとパスの不変性、および WORM 設定のサブディレクトリの継承がサポートされます。次の 2 つの WORM モードのいずれかを使用できます。

- 標準のコンプライアンスモード (デフォルト)

ユーザーがディレクトリまたは実行可能ではないファイルで UNIX *setuid* 権限を設定する (*chmod 4000 directory|file*) と、標準の WORM モードは WORM 保持期間を開始します。実行可能ファイルに対して設定 *setuid* (*set user ID upon execution (実行時にユーザー ID を設定)*) 権限を設定すると、セキュリティー上のリスクが生じるため、UNIX 実行権限も持つファイルはこのモードを使用して保持できません。

- エミュレーションモード

  ユーザーが書き込み可能ファイルまたはディレクトリを読み取り専用にする (*chmod 444 directory|file*) と、WORM エミュレーションモードは WORM 保持期間を開始するため、実行可能ファイルを保持できます。

標準モードとエミュレーションモードの両方に、厳密な WORM 実装と、*root* ユーザーのために一部の制限が緩和された制限の少ないライト実装があります。厳密な実装とライト実装のどちらでも、ファイルまたはディレクトリに対する保持が起動されたあとは、データまたはパスに対する変更は許可されません。厳密な実装では、だれも指定された保存期間 (デフォルトでは 43,200 分/30 日) を短縮したり、保存期間の終了前にファイルまたはディレクトリを削除したりすることはできません。また、*sammkfs* を使用して、現在保持されているファイルとディレクトリが格納されているボリュームを削除することもできません。そのため、厳密な実装は、法律および規制上のコンプライアンス要件を満たすのに適しています。ライト実装では、*root* ユーザーは、ファイルシステムの作成コマンド *sammkfs* を使用して保存期間の短縮、ファイルとディレクトリの削除、ボリュームの削除を行うことができます。そのため、データ整合性と柔軟な管理の両方が主な要件である場合に、ライト実装が適した選択肢である可能性があります。

WORM 実装を選択するとき、ファイルでの保存を有効にするときは、注意が必要です。通常、要件と一致するもっとも制約の少ないオプションを使用してください。標準モードからエミュレーションモード、またはその逆に変更することはできません。そのため、慎重に選択してください。管理の柔軟性が優先される場合、または保存の要件があとで変わる可能性がある場合、ライト実装を選択します。あとで必要なことがわかった場合、ライトバージョンの WORM モードから、厳密なバージョンにアップグレードできます。ただし、厳密な実装からライト実装に変更することはできません。厳密な WORM 実装が有効になったら、指定された保存期間を通じてファイルを保持する必要があります。そのため、保存期間は、要件と一致する最短値に設定します。

6.3.1. Oracle HSM ファイルシステムでの WORM サポートの有効化

マウントオプションを使用して、ファイルシステムで WORM サポートを有効にします。次のように進めます。

1. *root* としてログインします。

```
root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. テキストエディタで */etc/vfstab* ファイルを開き、WORM サポートを有効にする Oracle HSM ファイルシステムのエントリを見つけます。

   この例では、*/etc/vfstab* ファイルを *vi* エディタで開き、アーカイブファイルシステム *worm1* を見つけます。

```
root@solaris:~# vi /etc/vfstab
#File
#Device    Device  Mount    System fsck Mount   Mount
#to Mount to fsck Point    Type   Pass at Boot Options
#-------- ------- -------- ------ ---- ------- -------------------------
/devices  -       /devices devfs  -    no      -
/proc     -       /proc    proc   -    no      -
...
worm1     -       /worm1   samfs  -    yes     -
```

4. 標準の WORM コンプライアンスモードの厳密な実装を有効にするには、*vfstab* ファイルの *Mount Options* 列に *worm_capable* オプションを入力します。

```
#File
#Device    Device  Mount    System fsck Mount   Mount
#to Mount to fsck Point    Type   Pass at Boot Options
#-------- ------- -------- ------ ---- ------- -------------------------
/devices  -       /devices devfs  -    no      -
/proc     -       /proc    proc   -    no      -
```

```
...
worm1     -       /worm1   samfs  -    yes     worm_capable
```

5. 標準の WORM コンプライアンスモードのライト実装を有効にするには、*vfstab* ファイルの *Mount Options* 列に *worm_lite* オプションを入力します。

```
#File
#Device   Device  Mount    System fsck Mount   Mount
#to Mount to fsck Point    Type   Pass at Boot Options
#-------- ------- -------- ------ ---- ------- -------------------------
/devices  -       /devices devfs  -    no      -
/proc     -       /proc    proc   -    no      -
...
worm1     -       /worm1   samfs  -    yes     worm_lite
```

6. WORM エミュレーションモードの厳密な実装を有効にするには、*vfstab* ファイルの *Mount Options* 列に *worm_emul* オプションを入力します。

```
#File
#Device   Device  Mount    System fsck Mount   Mount
#to Mount to fsck Point    Type   Pass at Boot Options
#-------- ------- -------- ------ ---- ------- -------------------------
/devices  -       /devices devfs  -    no      -
/proc     -       /proc    proc   -    no      -
...
worm1     -       /worm1   samfs  -    yes     worm_emul
```

7. WORM エミュレーションモードのライト実装を有効にするには、*vfstab* ファイルの *Mount Options* 列に *emul_lite* オプションを入力します。

```
#File
#Device   Device  Mount    System fsck Mount   Mount
#to Mount to fsck Point    Type   Pass at Boot Options
#-------- ------- -------- ------ ---- ------- -------------------------
/devices  -       /devices devfs  -    no      -
/proc     -       /proc    proc   -    no      -
...
```

```
worm1     -      /worm1    samfs  -    yes     emul_lite
```

8. 保存期間が明示的に割り当てられていないファイルのデフォルトの保存期間を変更するには、*def_retention=period* オプションを *vfstab* ファイルの *Mount Options* 列に追加します。ここで、*period* は、次の段落で説明されている形式の 1 つを取ります。

   period の値には、次の 3 つの形式のいずれかを使用できます。

   - *permanent* または *0* は、永続的な保存を指定します。
   - *YEARSyDAYSdHOURShMINUTESm*。ここで、*YEARS*、*DAYS*、*HOURS*、および *MINUTES* は、負以外の整数で、指定子は省略できます。そのため、たとえば、*5y3d1h4m*、*2y12h*、および *365d* はすべて有効です。
   - *MINUTES*。ここで、*MINUTES* は *[1-2147483647]* の範囲内の整数です。

   2038 年を超える保存期間を設定する必要がある場合、デフォルトの保存期間を設定します。*touch* などの UNIX ユーティリティーは、符号付き 32 ビット整数を使用して、1970 年 1 月 1 日以降に経過した秒数として時間を表します。32 ビット整数が表すことができる最大秒数は、2038 年 1 月 18 日午後 10:14 に変換されます

   値が指定されていない場合、*def_retention* はデフォルトで *43200* 分 (30 日) に設定されます。この例では、標準の WORM 対応ファイルシステムの保存期間を *777600* 分 (540 日) に設定します。

```
#File
#Device   Device  Mount    System fsck Mount   Mount
#to Mount to fsck Point    Type   Pass at Boot Options
#-------- ------- -------- ------ ---- ------- -------------------------
/devices  -       /devices devfs  -    no      -
/proc     -       /proc    proc   -    no      -
...
worm1     -       /worm1   samfs  -    no     worm_capable,def_retention=777600
```

9. *vfstab* ファイルを保存して、エディタを閉じます。

   ファイルシステムは WORM 対応です。1 つ以上の WORM ファイルがファイルシステムにある場合、Oracle HSM ソフトウェアは、WORM 機能を反映するためにファイルシステムのスーパーブロックを更新します。厳密な *worm_capable* または *worm_emul* マウント

オプションを使用してファイルシステムがマウントされている場合、その後に *sammkfs* でファイルシステムを再構築しようとすると失敗します。

10. LTFS を使用するシステムと相互作業する必要がある場合、またはリモートサイト間で大量のデータを転送する必要がある場合、「Linear Tape File System (LTFS) のサポートの有効化」を参照してください
11. 複数ホストのファイルシステムアクセスや高可用性構成などの追加の要件がある場合、「応用編」を参照してください。
12. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

6.4. Linear Tape File System (LTFS) のサポートの有効化

Oracle HSM は、データを Linear Tape File System (LTFS) ボリュームからインポートしたり、このボリュームにエクスポートしたりできます。この機能により、LTFS を標準のテープ形式として使用するシステムとの相互作用が容易になります。また、一般的なワイドエリアネットワーク (WAN) 接続が遅すぎるか、タスクにとって高価すぎる場合に、大量のデータをリモート Oracle HSM サイト間で簡単に転送できます。

LTFS ボリュームの使用と管理については、*samltfs* のマニュアルページと『*Oracle Hierarchical Storage Manager and StorageTek QFS Software 保守および管理ガイド*』を参照してください。

Oracle HSM LTFS のサポートを有効にするには、次のように進めます。

1. Oracle HSM メタデータサーバーに *root* としてログインします。

```
[samfs-mds]root@solaris:~#
```

2. アーカイブプロセスがある場合、すべてアイドル状態にします。コマンド *samcmd aridle* を使用します。

   このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

```
[samfs-mds]root@solaris:~# samcmd aridle
[samfs-mds]root@solaris:~#
```

3. ステージングプロセスがある場合、すべてアイドル状態にします。コマンド *samcmd stidle* を使用します。

このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

```
[samfs-mds]root@solaris:~# samcmd stidle
[samfs-mds]root@solaris:~#
```

4. アクティブなアーカイブジョブが完了するまで待機します。コマンド *samcmd a* を使用して、アーカイブプロセスのステータスを確認します。

   アーカイブプロセスが *Waiting for :arrun* の場合、アーカイブプロセスはアイドル状態になっています。

```
[samfs-mds]root@solaris:~# samcmd a
Archiver status samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samfs-mds
sam-archiverd:  Waiting for :arrun
sam-arfind: ...
Waiting for :arrun
```

5. アクティブなステージングジョブが完了するまで待機します。コマンド *samcmd u* を使用してステージングプロセスのステータスを確認します。

   ステージングプロセスが *Waiting for :strun* の場合、ステージングプロセスはアイドル状態になっています。

```
[samfs-mds]root@solaris:~# samcmd u
Staging queue samcmd      6.0 14:20:34 Feb 22 2015
samcmd on solaris.demo.lan
Staging queue by media type: all
sam-stagerd:  Waiting for :strun
root@solaris:~#
```

6. すべてのリムーバブルメディアドライブをアイドル状態にしてから、続行します。ドライブごとに、コマンド *samcmd equipment-number idle* を使用します。ここで *equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイル内のドライブに割り当てられている装置の順序番号です。

このコマンドはドライブを「*off*」にする前に、現在のアーカイブジョブおよびステージングジョブを完了できますが、新しいジョブは開始されません。この例では、4 つのドライブ (順序番号 *801*、*802*、*803*、*804*) をアイドル状態にします。

```
[samfs-mds]root@solaris:~# samcmd 801 idle
[samfs-mds]root@solaris:~# samcmd 802 idle
[samfs-mds]root@solaris:~# samcmd 803 idle
[samfs-mds]root@solaris:~# samcmd 804 idle
[samfs-mds]root@solaris:~#
```

7. 実行中のジョブが完了するまで待機します。

   コマンド *samcmd r* を使用すると、ドライブのステータスを確認できます。すべてのドライブが「*notrdy*」または「*empty*」の場合は、続行できる状態になっています。

```
[samfs-mds]root@solaris:~# samcmd r
Removable media samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samqfs1host
ty   eq   status      act  use  state  vsn
li  801   ---------p    0   0%  notrdy
          empty
li  802   ---------p    0   0%  notrdy
          empty
li  803   ---------p    0   0%  notrdy
          empty
li  804   ---------p    0   0%  notrdy
          empty
[samfs-mds]root@solaris:~#
```

8. アーカイバおよびステージャープロセスがアイドル状態で、テープドライブがすべて「*notrdy*」になっている場合は、ライブラリ制御デーモンを停止します。コマンド *samd stop* を使用します。

```
[samfs-mds]root@solaris:~# samd stop
[samfs-mds]root@solaris:~#
```

9. テキストエディタで */etc/opt/SUNWsamfs/defaults.conf* を開きます。

この例では、*vi* エディタでファイルを開きます。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
```

10. *defaults.conf* ファイルで、行 *ltfs = mountpoint workers volumes* を追加します。ここで、mountpoint は、LTFS ファイルシステムがマウントされているホストファイルシステム内のディレクトリ、*workers* は、LTFS に使用するドライブのオプションの最大数で、*volumes* は、ドライブごとのテープボリュームのオプションの最大数です。次に、ファイルを保存し、エディタを閉じます。

    この例では、LTFS マウントポイント */mnt/ltfs* を指定して、ほかのパラメータではデフォルトを受け入れます。

```
[samfs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
ltfs = /mnt/ltfs
:wq
[samfs-mds]root@solaris:~#
```

11. *defaults.conf* ファイルを再度読み取り、それに従って Oracle HSM ソフトウェア自体を再構成するようにこのソフトウェアに指示します。報告されたエラーをすべて修正して、必要に応じて繰り返します。

```
[samfs-mds]root@solaris:~# /opt/SUNWsamfs/sbin/samd config
```

12. 前の手順で Oracle HSM 操作を停止した場合、この時点で *samd start* コマンドを使用して再開します。

```
[samfs-mds]root@solaris:~# samd start
```

13. これで、LTFS の Oracle HSM のサポートが有効になりました。複数ホストのファイルシステムアクセスや高可用性構成などの追加の要件がある場合、「応用編」を参照してください。
14. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

6.5. 応用編

これで、Oracle HSM ファイルシステムの基本的なインストールと構成が完了します。この時点で、さまざまな目的のために最適に構成されている、完全に機能するファイルシステムが設定されています。

このマニュアルの残りの章では、より専門的なニーズに対処します。そのため、次に概要が説明されている追加の調整タスクと機能の実装タスクを開始する前に、要件を慎重に評価してください。次に、高可用性や共有ファイルシステム構成など、追加の機能が必要な場合は、基本構成から追加の機能を慎重に実装できます。ただし、これまでに行なった作業がニーズを満たしていると判断した場合は、追加の変更が向上につながる可能性はないと考えられます。単にメンテナンスと管理を複雑にする場合があります。

- アプリケーションで異常に大きいか、異常に均一な量のデータをファイルシステムに転送する場合、追加のマウントオプションを設定することで、ファイルシステムのパフォーマンス向上が可能になることがあります。詳細は、12章「*特殊なニーズのための入出力特性の調整*」を参照してください。
- ファイルシステムへの共有アクセスを構成する必要がある場合、「Oracle HSM ソフトウェアを使用した複数のホストからのファイルシステムへのアクセス」や「NFS と SMB/CIFS を使用した複数のホストからファイルシステムへのアクセス」を参照してください。
- 高可用性 QFS ファイルシステムまたは Oracle HSM アーカイブファイルシステムを構成する必要がある場合、9章「*高可用性ソリューションの準備* 」を参照してください。
- リモートの場所にホストされたアーカイブストレージを共有するように Oracle HSM アーカイブファイルシステムを構成する必要がある場合、8章「*SAM-Remote の構成* 」を参照してください。
- サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
- それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

7

複数のホストからのファイルシステムへのアクセス

Oracle HSM ファイルシステムは、いくつかの方法のいずれかを使用して複数のホスト間で共有できます。それぞれの方法には、ある状況では一定の利点がある一方で、別の状況では著しい欠点もあります。したがって、方法の選択は固有の要件によって異なります。共有する方法は次のとおりです。

- Oracle HSM ソフトウェアを使用した複数のホストからのファイルシステムへのアクセス
- NFS と SMB/CIFS を使用した複数のホストからファイルシステムへのアクセス

7.1. Oracle HSM ソフトウェアを使用した複数のホストからのファイルシステムへのアクセス

Oracle HSM では、すべてのファイルシステムを同時にマウントする 1 つのサーバーおよび 1 つ以上のクライアントを構成することによって、ファイルシステムを複数のホストで使用できるようにします。その後、ファイルデータは、NFS および CIFS 共有に関連するネットワークおよび中間サーバーの待ち時間なしで、高パフォーマンスのローカルパス入出力によってディスクデバイスからホストに直接渡されます。メタデータサーバーとして同時にアクティブにできるホストは 1 つだけですが、冗長化の目的では、任意の数のクライアントを潜在的なメタデータサーバーとして構成できます。ファイルシステムのマウントポイント数には制限がありません。

Oracle HSM では、アーカイブ処理を使用するかどうかに関係なく、複数読み取り/単一書き込み構成と共有構成の両方で高パフォーマンス (*ma*) と汎用 (*ms*) の両方のファイルシステムへの複数ホストアクセスがサポートされています。制限事項がわずかにあります。

- ブロック (*b–*) 特殊ファイルはサポートされていません。
- 文字 (*c–*) 特殊ファイルはサポートされていません。
- FIFO 名前付きパイプ (*p–*) 特殊ファイルはサポートされていません。
- セグメント化ファイルはサポートされていません。

  セグメント化ファイル環境では、Oracle HSM 共有ファイルシステムを実装できません。
- 必須のロックはサポートされていません。

必須のロックが設定されている場合は、*EACCES* エラーが返されます。ただし、アドバイザリロックはサポートされています。アドバイザリロックの詳細は、*fcntl* のマニュアルページを参照してください。

Oracle HSM software ホストは、2 つの構成のいずれかを使用してファイルシステムデータにアクセスできますが、それぞれの構成に特定のアプリケーションでの独自の利点と制限があります。

*複数読み取り/単一書き込み*構成では、単一のホストに読み取り/書き込みアクセス権を付与してファイルシステムをマウントし、その他のすべてのホストではそれを読み取り専用でマウントします。構成は、単にマウントポイントオプションを設定するだけで済みます。単一のホストがファイルに対するすべての変更を行うため、追加のファイルロックや整合性チェックなしでも、ファイルの整合性およびデータの完全性が保証されます。パフォーマンスを最適にするために、すべてのホストがディスクから直接メタデータとデータを読み取ります。ただし、すべてのホストがファイルシステムのメタデータにアクセスする必要があるため、*ma* ファイルシステム内のすべてのホストがデータとメタデータデバイスの両方へのアクセス権を持っている必要があります。

*共有*構成では、単一のホストが特定の期間内に特定の方法でファイルにアクセスすることを許可するリースを使用することで、すべてのホストがファイルデータの読み取り、書き込み、および追加を行うことができます。メタデータサーバーは*読み取り*、*書き込み*、および*追加*のリースを発行し、更新および競合リースの要求を管理します。共有ファイルシステムでは、高い柔軟性が提供されますが、構成が多少複雑になるため、ファイルシステムのオーバーヘッドが増加します。すべてのホストはディスクから直接ファイルデータを読み取りますが、クライアントはネットワークを介してメタデータにアクセスします。そのため、メタデータデバイスへのアクセス権が不足しているクライアントでも、*ma* ファイルシステムを共有できます。

複数のホストからのデータへのアクセスを構成するには、2 つの方法のいずれかを選択します。

- Oracle HSM 単一書き込み/複数読み取りファイルシステムの構成
- Oracle HSM 共有ファイルシステムの構成.

7.1.1. Oracle HSM 単一書き込み/複数読み取りファイルシステムの構成

単一書き込み/複数読み取りファイルシステムを構成するには、次のタスクを実行します。

- 書き込みでのファイルシステムの作成

- 読み取りの構成

7.1.1.1. 書き込みでのファイルシステムの作成

次のように進めます。

1. *root* アカウントを使用して、*writer*として機能するホストにログインします。

   この例では、*writer* ホストの名前は *swriterfs-mds-writer* です。

```
[swriterfs1-mds-writer]root@solaris:~#
```

2. *writer* として機能するホスト上で、*/etc/opt/SUNWsamfs/mcf* ファイルをテキストエディタで開き、QFS ファイルシステムを追加します。汎用の **ms** ファイルシステムの構成、または 高パフォーマンス **ma** ファイルシステムの構成を実行できます。

   個別のメタデータデバイスを持つ *ma* ファイルシステム上で、ファイルシステムのメタデータサーバーを書き込み側として構成します。次の例では、*vi* テキストエディタを使用して、ホスト *swriterfs1-mds-writer* 上の *mcf* ファイルを編集します。この例では、装置 ID およびファミリセット名 *swriterfs1* と装置番号 *300* を使用して *ma* ファイルシステムを指定します。

```
[swriterfs1-mds-writer]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family      Device   Additional
# Identifier          Ordinal    Type       Set         State    Parameters
#------------------  ---------  ---------  ---------   ------   ---------------
swriterfs1            300         ma          swriterfs1  on
 /dev/dsk/c0t0d0s0    301        mm         swriterfs1  on
 /dev/dsk/c0t3d0s0    302        mr         swriterfs1  on
 /dev/dsk/c0t3d0s1    303        mr         swriterfs1  on
```

3. */etc/opt/SUNWsamfs/mcf* ファイルを保存して、エディタを終了します。

   この例では、変更を保存して、*vi* エディタを終了します。

```
# Equipment           Equipment  Equipment  Family      Device   Additional
# Identifier          Ordinal    Type       Set         State    Parameters
#------------------  ---------  ---------  ---------   ------   ---------------
swriterfs1            300        ma         swriterfs1  on
 /dev/dsk/c0t0d0s0     301        mm          swriterfs1  on
```

```
 /dev/dsk/c0t3d0s0    302        mr         swriterfs1  on
 /dev/dsk/c0t3d0s1    303        mr         swriterfs1  on
:wq
[swriterfs1-mds-writer]root@solaris:~#
```

4. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、エラーがあれば修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーを検出すると停止します。

```
[swriterfs1-mds-writer]root@solaris:~# sam-fsd
...
Would start sam-stagerd()
Would start sam-amld()
[swriterfs1-mds-writer]root@solaris:~#
```

5. Oracle HSM サービスに、*mcf* ファイルを再度読み取り、それ自体を適宜再構成するように指示します。コマンド *samd config* を使用します。

```
[swriterfs1-mds-writer]root@solaris:~# samd config
Configuring SAM-FS
[swriterfs1-mds-writer]root@solaris:~#
```

6. 「高パフォーマンス **ma** ファイルシステムの構成」で説明したとおりに、*sammkfs* コマンドおよびファイルシステムのファミリセット名を使用して、ファイルシステムを作成します。

   この例では、コマンドは単一書き込み/複数読み取りファイルシステム *swriterfs1* を作成します。

```
[swriterfs1-mds-writer]root@solaris:~# sammkfs swriterfs1
Building 'swriterfs1' will destroy the contents of devices:
  /dev/dsk/c0t0d0s0
  /dev/dsk/c0t3d0s0
  /dev/dsk/c0t3d0s1
Do you wish to continue? [y/N]yes ...
```

7. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
[swriterfs1-mds-writer]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
[swriterfs1-mds-writer]root@solaris:~#
```

8. 「高パフォーマンス **ma** ファイルシステムの構成」で説明したとおりに、オペレーティングシステムの */etc/vfstab* ファイルに新しいファイルシステムを追加します。

   この例では、*vi* テキストエディタで */etc/vfstab* ファイルを開き、*swriterfs1* ファミリセットデバイスの行を追加します。

```
[swriterfs1-mds-writer]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount       System  fsck  Mount    Mount
#to Mount  to fsck  Point       Type    Pass  at Boot  Options
#--------  -------  --------    ------  ----  -------  -----------------------
/devices   -        /devices    devfs   -     no       -
/proc      -        /proc       proc    -     no       -
...
swriterfs1 -        /swriterfs1 samfs   -     no
```

9. */etc/vfstab* ファイルの「*Mount Options*」列に、*writer* マウントオプションを入力します。

   注意:

   常に 1 つのホストのみが *writer* になっていることを確認します。*writer* オプションを使用して、複数のホストによる複数読み取り/単一書き込みファイルシステムのマウントを許可すると、ファイルシステムが破損する可能性があります。

```
#File
#Device    Device   Mount       System  fsck  Mount    Mount
#to Mount  to fsck  Point       Type    Pass  at Boot  Options
#--------  -------  --------    ------  ----  -------  -----------------------
/devices   -        /devices    devfs   -     no       -
/proc      -        /proc       proc    -     no       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       writer
```

10. その他の必要な変更を */etc/vfstab* ファイルに行います。コンマを区切り文字として使用して、マウントオプションを追加します。

    たとえば、最初の試行に失敗した場合にバックグラウンドでファイルシステムをマウントするには、「*Mount Options*」フィールドに *bg* マウントオプションを追加します (指定可能なマウントオプションの包括的なリストについては、*mount_samfs* のマニュアルページを参照)。

```
#File
#Device    Device   Mount       System  fsck  Mount    Mount
#to Mount  to fsck  Point       Type    Pass  at Boot  Options
#--------  -------  --------    ------  ----  -------  -----------------------
/devices   -        /devices    devfs   -     no       -
/proc      -        /proc       proc    -     no       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       writer,bg
```

11. */etc/vfstab* ファイルを保存して、エディタを終了します。

```
#File
#Device    Device   Mount       System  fsck  Mount    Mount
#to Mount  to fsck  Point       Type    Pass  at Boot  Options
#--------  -------  --------    ------  ----  -------  -----------------------
/devices   -        /devices    devfs   -     no       -
/proc      -        /proc       proc    -     no       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       writer,bg
:wq
[swriterfs1-mds-writer]root@solaris:~#
```

12. */etc/vfstab* ファイルで指定されたマウントポイントを作成し、そのマウントポイントに対するアクセス権を設定します。

    マウントポイントディレクトリに移動し、マウント済みファイルシステム内のファイルにアクセスするには、すべてのホスト上でマウントポイントアクセス権が同じである必要があり、ユーザーが実行 (*x*) 権限を持っている必要があります。この例では、*/swriterfs1* マウントポイントディレクトリを作成し、アクセス権を *755* (*-rwxr-xr-x*) に設定します。

```
[swriterfs1-mds-writer]root@solaris:~# mkdir /swriterfs1
[swriterfs1-mds-writer]root@solaris:~# chmod 755 /swriterfs1
[swriterfs1-mds-writer]root@solaris:~#
```

13. 新しいファイルシステムをマウントします。

```
[swriterfs1-mds-writer]root@solaris:~# mount /swriterfs1
[swriterfs1-mds-writer]root@solaris:~#
```

14. 共有ファイルシステムが作成されたら、読み取りの構成を実行します。

7.1.1.2. 読み取りの構成

*読み取り*は、ファイルシステムを読み取り専用でマウントするホストです。読み取りとして構成するホストごとに、次の手順を実行します。

1. ホストに *root* としてログインします。

   この例では、*reader* ホストの名前は *swriterfs-reader1]* です。

```
[swriterfs-reader1]root@solaris:~#
```

2. 端末ウィンドウで *samfsconfig device-path* を使用して、複数読み取り/単一書き込みファイルシステムの構成情報を取得します。ここで *device-path* は、コマンドがファイルシステムディスクデバイスの検索を開始する場所 (*/dev/dsk/** など) です。

   samfsconfig ユーティリティーは、*sammkfs* が Oracle HSM ファイルシステムに含まれている各デバイス上に書き込む識別スーパーブロックを読み取ることで、ファイルシステムの構成情報を取得します。このコマンドは、現在のホストから始まる、構成内の各デバイスへの正確なパスを返し、到達できないデバイスにフラグを付けます (コマンドの構文およびパラメータの詳細は、*samfsconfig* のマニュアルページを参照)。

   この例では、*samfsconfig* 出力は、デバイスへのパスがホスト *swriterfs1-reader1* から指定されている点を除き、*swriterfs1-mds-writer* 上の *mcf* ファイルに一覧表示されるものと同じ装置を示しています。

```
[swriterfs1-reader1]root@solaris:~# samfsconfig /dev/dsk/*
# Family Set 'swriterfs1' Created Thu Nov 21 07:17:00 2013
# Generation 0 Eq count 4 Eq meta count 1
```

```
#
sharefs              300        ma         sharefs   -
 /dev/dsk/c1t0d0s0    301        mm         sharefs   -
 /dev/dsk/c1t3d0s0    302        mr         sharefs   -
 /dev/dsk/c1t3d0s1    303        mr         sharefs   -
```

3. *samfsconfig* の出力から共有ファイルシステムのエントリをコピーします。次に、2 つ目のウィンドウで */etc/opt/SUNWsamfs/mcf* ファイルをテキストエディタで開き、コピーしたエントリをファイルにペーストします。

   あるいは、*samfsconfig* の出力を *mcf* ファイルにリダイレクトできます。または、*samd buildmcf* コマンドを使用して *samfsconfig* を実行し、クライアント *mcf* ファイルを自動的に作成できます。

   この例では、コメントアウトした列見出しを追加したあとのホスト *swriterfs1-reader1* の *mcf* ファイルが次のように表示されます。

```
[swriterfs1-reader1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#-----------------    ---------  ---------  ---------  ------   ---------------
sharefs              300        ma         sharefs    -
 /dev/dsk/c1t0d0s0    301        mm         sharefs    -
 /dev/dsk/c1t3d0s0    302        mr         sharefs    -
 /dev/dsk/c1t3d0s1    303        mr         sharefs    -
```

4. すべてのデバイスで「*Device State*」フィールドが *on* に設定されていることを確認します。次に、*mcf* ファイルを保存します。

```
[swriterfs1-reader1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs              300        ma         sharefs    on
 /dev/dsk/c1t0d0s0    301        mm         sharefs    on
 /dev/dsk/c1t3d0s0    302        mr         sharefs    on
 /dev/dsk/c1t3d0s1    303        mr         sharefs    on
:wq
```

```
[swriterfs1-reader1]root@solaris:~#
```

5. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、エラーがあれば修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーを検出すると停止します。

```
[swriterfs1-reader1]root@solaris:~# sam-fsd
...
Would start sam-stagerd()
Would start sam-amld()
[swriterfs1-reader1]root@solaris:~#
```

6. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
[swriterfs1-reader1]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
[swriterfs1-reader1]root@solaris:~#
```

7. ホストオペレーティングシステムの */etc/vfstab* ファイルに単一書き込み/複数読み取りファイルシステムを追加します。

   この例では、*vi* テキストエディタで */etc/vfstab* ファイルを開き、*swriterfs1* ファミリセットデバイスの行を追加します。

```
[swriterfs1-reader1]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount       System  fsck  Mount    Mount
#to Mount  to fsck  Point       Type    Pass  at Boot  Options
#--------  -------  --------    ------  ----  -------  -----------------------
/devices   -        /devices    devfs   -     no       -
/proc      -        /proc       proc    -     no       -
...
swriterfs1 -        /swriterfs1 samfs   -     no
```

8. */etc/vfstab* ファイルの「*Mount Options*」列に、*reader* オプションを入力します。

注意:

ホストが *reader* オプションを使用してファイルシステムをマウントすることを確認します。誤って複数のホスト上で *writer* マウントオプションを使用すると、ファイルシステムが破損する可能性があります。

```
#File
#Device    Device   Mount       System  fsck  Mount    Mount
#to Mount  to fsck  Point       Type    Pass  at Boot  Options
#--------  -------  --------    ------  ----  -------  ----------------------
/devices   -        /devices    devfs   -     no       -
/proc      -        /proc       proc    -     no       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       reader
```

9. セパレータとしてコンマを使用して、その他の必要なマウントオプションを追加し、その他の必要な変更を */etc/vfstab* ファイルに加えます。次に、*/etc/vfstab* ファイルを保存します。

```
#File
#Device    Device   Mount       System  fsck  Mount    Mount
#to Mount  to fsck  Point       Type    Pass  at Boot  Options
#--------  -------  --------    ------  ----  -------  ----------------------
/devices   -        /devices    devfs   -     no       -
/proc      -        /proc       proc    -     no       -
...
swriterfs1 -        /swriterfs1 samfs   -     no       writer,bg
:wq
[swriterfs1-reader1]root@solaris:~#
```

10. */etc/vfstab* ファイルで指定されたマウントポイントを作成し、そのマウントポイントに対するアクセス権を設定します。

    マウントポイントディレクトリに移動し、マウント済みファイルシステム内のファイルにアクセスするには、すべてのホスト上でマウントポイントアクセス権が同じである必要があり、ユーザーが実行 (*x*) 権限を持っている必要があります。この例では、書き込みホストで実行したときと同様に、*/swriterfs1* マウントポイントディレクトリを作成し、アクセス権を *755* (*-rwxr-xr-x*) に設定します。

```
[swriterfs1-reader1]root@solaris:~# mkdir /swriterfs1
[swriterfs1-reader1]root@solaris:~# chmod 755 /swriterfs1
[swriterfs1-reader1]root@solaris:~#
```

11. 新しいファイルシステムをマウントします。

```
[swriterfs1-reader1]root@solaris:~# mount /swriterfs1
[swriterfs1-reader1]root@solaris:~#
```

12. ファイルシステムを読み取り専用でマウントするようにすべての読み取りホストが構成されるまで、この手順を繰り返します。
13. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
14. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

7.1.2. Oracle HSM 共有ファイルシステムの構成

Oracle HSM 共有ファイルシステムでは、複数の Oracle HSM ホストにファイルへの読み取り、書き込み、および追加のアクセス権が付与されます。すべてのホストがファイルシステムをマウントし、ストレージデバイスに直接接続します。さらに、1 つのホストであるメタデータサーバー (MDS) がファイルシステムのメタデータを排他的に制御し、同じファイルへのアクセスを求めるホスト間を調整します。サーバーは読み取り、書き込み、追加リースを発行、更新、および取り消すことで、Ethernet ローカルネットワーク経由でクライアントホストにメタデータの更新を提供し、ファイルアクセスを制御します。高パフォーマンス *ma* または汎用 *ms* タイプの非アーカイブファイルシステムとアーカイブファイルシステムの両方を共有できます。

共有ファイルシステムを構成するには、次のタスクを実行します。

- 共有するファイルシステムメタデータサーバーの構成
- 共有するファイルシステムクライアントの構成
- 共有ファイルシステム用のアーカイブストレージの構成

7.1.2.1. 共有するファイルシステムメタデータサーバーの構成

共有ファイルシステムがサポートされるようにメタデータサーバーを構成するには、次に示すタスクを実行します。

- アクティブおよび潜在的なメタデータサーバーでのホストファイルの作成
- アクティブなサーバーでの共有ファイルシステムの作成

- アクティブなサーバーでの共有ファイルシステムのマウント

7.1.2.1.1. アクティブおよび潜在的なメタデータサーバーでのホストファイルの作成

アクティブおよび潜在的なメタデータサーバー上で、共有ファイルシステムのサーバーおよびクライアントに関するネットワークアドレス情報を一覧表示する hosts ファイルを作成する必要があります。hosts ファイルは、*/etc/opt/SUNWsamfs/* ディレクトリに *mcf* ファイルとともに格納されています。共有ファイルシステムの初期作成中に、*sammkfs -S* コマンドを実行すると、このファイルに格納されている設定を使用して共有が構成されます。ここで、次の手順を使用して作成します。

1. サーバーに *root* としてログインします。

   この例では、サーバーの名前は *sharefs-mds* です。

```
[sharefs-mds]root@solaris:~#
```

2. テキストエディタを使用して、メタデータサーバー上で */etc/opt/SUNWsamfs/hosts.family-set-name* を作成します。*family-set-name* は、共有する予定のファイルシステムのファミリセット名で置き換えます。

   この例では、*vi* テキストエディタを使用してファイル *hosts.sharefs* を作成します。いくつかのオプションの見出しを追加します。各行は、コメントを示すシャープ記号 (*#*) で始めます。

```
[sharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                         Server   On/  Additional
#Host Name            Network Interface   Ordinal  Off  Parameters
#------------------   ------------------- -------  ---  ----------
```

3. メタデータサーバーのホスト名と IP アドレスまたはドメイン名を 2 列で、空白文字で区切って追加します。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                         Server   On/  Additional
#Host Name            Network Interface   Ordinal  Off  Parameters
#------------------   ------------------- -------  ---  ----------
```

```
sharefs-mds            10.79.213.117
```

4. 3 列目を、空白文字でネットワークアドレスと区切って追加します。この列に、アクティブなメタデータサーバーの順序番号である *1* を入力します。

   この例では、メタデータサーバーは 1 つだけであるため、*1* を入力します。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                        Server   On/  Additional
#Host Name            Network Interface  Ordinal  Off  Parameters
#------------------   ------------------ -------  ---  ----------
sharefs-mds           10.79.213.117      1
```

5. 4 列目を、空白文字でネットワークアドレスと区切って追加します。この列には、*0* (ゼロ) を入力します。

   4 列目の *0*、- (ハイフン)、または空白値は、ホストが「*on*」(共有ファイルシステムへのアクセスありで構成) であることを示します。*1* (数字の 1) は、ホストが「*off*」(ファイルシステムへのアクセスなしで構成) であることを示します (共有ファイルシステムを管理する際のこれらの値の使用については、*samsharefs* のマニュアルページを参照してください)。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                        Server   On/  Additional
#Host Name            Network Interface  Ordinal  Off  Parameters
#------------------   ------------------ -------  ---  ----------
sharefs-mds          10.79.213.117      1        0
```

6. 5 列目を、空白文字でネットワークアドレスと区切って追加します。この列には、現在アクティブなメタデータサーバーを示すキーワード「*server*」を入力します。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                        Server   On/  Additional
#Host Name            Network Interface  Ordinal  Off  Parameters
#------------------   ------------------ -------  ---  ----------
sharefs-mds           10.79.213.117      1        0    server
```

7. 潜在的なメタデータサーバーとして 1 つ以上のホストを追加する予定である場合は、それぞれのエントリを作成します。そのたびに、サーバー番号を増分します。ただし、「*server*」キーワードは含めないでください (アクティブなメタデータサーバーは、ファイルシステムごとに 1 つのみです)。

   この例では、ホスト *sharefs-mds_alt* は、サーバー番号が *2* の潜在的なメタデータサーバーです。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                         Server   On/  Additional
#Host Name            Network Interface   Ordinal  Off  Parameters
#------------------   ------------------- -------  ---  ----------
sharefs-mds           10.79.213.117       1        0    server
sharefs-mds_alt       10.79.213.217       2        0
```

8. クライアントホストごとに 1 行追加して、それぞれのサーバー番号の値を *0* に指定します。

   サーバー番号 *0* は、クライアントとしてのホストを示します。この例では、2 つのクライアント (*sharefs-client1* と *sharefs-client2*) を追加します。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                         Server   On/  Additional
#Host Name            Network Interface   Ordinal  Off  Parameters
#------------------   ------------------- -------  ---  ----------
sharefs-mds           10.79.213.117       1        0    server
sharefs-mds_alt       10.79.213.217       2        0
sharefs-client1       10.79.213.133       0        0
sharefs-client2       10.79.213.147       0        0
```

9. */etc/opt/SUNWsamfs/hosts.family-set-name* ファイルを保存して、エディタを終了します。

   この例では、*/etc/opt/SUNWsamfs/hosts.sharefs* への変更を保存して、*vi* エディタを終了します。

```
# /etc/opt/SUNWsamfs/hosts.sharefs
#                                         Server   On/  Additional
```

```
#Host Name            Network Interface       Ordinal  Off  Parameters
#------------------   ---------------------   -------  ---  ----------
sharefs-mds           10.79.213.117           1        0    server
sharefs-mds_alt       10.79.213.217           2        0
sharefs-client1       10.79.213.133           0        0
sharefs-client2       10.79.213.147           0        0
:wq
[sharefs-mds]root@solaris:~#
```

10. 共有ファイルシステムの構成に含まれる任意の潜在的なメタデータサーバー上に、新しい */etc/opt/SUNWsamfs/hosts.family-set-name* ファイルのコピーを配置します。
11. 次に、アクティブなサーバーでの共有ファイルシステムの作成を実行します。

7.1.2.1.2. アクティブなサーバーでの共有ファイルシステムの作成

次のように進めます。

1. サーバーに *root* としてログインします。

   この例では、サーバーの名前は *sharefs-mds* です。

```
[sharefs-mds]root@solaris:~#
```

2. メタデータサーバー (MDS) 上で、*/etc/opt/SUNWsamfs/mcf* ファイルをテキストエディタで開き、QFS ファイルシステムを追加します。汎用の **ms** ファイルシステムの構成、または 高パフォーマンス **ma** ファイルシステムの構成を実行できます。

   次の例では、*vi* テキストエディタを使用して、ホスト *sharefs-mds* 上の *mcf* ファイルを編集します。この例では、装置 ID およびファミリセット名 *sharefs* と装置番号 *300* を使用して *ma* ファイルシステムを指定します。

```
[sharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300        ma         sharefs    on
 /dev/dsk/c0t0d0s0    301        mm         sharefs    on
 /dev/dsk/c0t3d0s0    302        mr         sharefs    on
 /dev/dsk/c0t3d0s1    303        mr         sharefs    on
```

3. *ma* ファイルシステム装置に対応する行の「*Additional Parameters*」フィールドに *shared* パラメータを入力します。

```
# Equipment           Equipment   Equipment  Family     Device   Additional
# Identifier          Ordinal     Type       Set        State    Parameters
#-----------------    ---------   ---------  ---------  ------   ---------------
sharefs               300         ma         sharefs    on       shared
 /dev/dsk/c0t0d0s0     301         mm         sharefs    on
 /dev/dsk/c0t3d0s0     302         mr         sharefs    on
 /dev/dsk/c0t3d0s1     303         mr         sharefs    on
```

4. */etc/opt/SUNWsamfs/mcf* ファイルを保存して、エディタを終了します。

   この例では、変更を保存して、*vi* エディタを終了します。

```
sharefs               300         ma         sharefs    on      shared
 /dev/dsk/c0t0d0s0     301         mm         sharefs    on
 /dev/dsk/c0t3d0s0     302         mr         sharefs    on
 /dev/dsk/c0t3d0s1     303         mr         sharefs    on
:wq
[sharefs-mds]root@solaris:~#
```

5. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、エラーがあれば修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーを検出すると停止します。

```
[sharefs-mds]root@solaris:~# sam-fsd
...
Would start sam-stagerd()
Would start sam-amld()
[sharefs-mds]root@solaris:~#
```

6. *mcf* ファイルを再度読み取り、それ自体を適宜再構成するように Oracle HSM サービスに指示します。報告されたエラーをすべて修正して、必要に応じて繰り返します。

```
[sharefs-mds]root@solaris:~# samd config
[sharefs-mds]root@solaris:~#
```

7. 「高パフォーマンス **ma** ファイルシステムの構成」で説明したとおりに、*sammkfs -S* コマンドおよびファイルシステムのファミリセット名を使用して、ファイルシステムを作成します。

   sammkfs コマンドは、*hosts.family-set-name* および *mcf* ファイルを読み取って、指定されたプロパティーを使用して共有ファイルシステムを作成します。この例では、コマンドは *hosts.sharefs* ファイルから共有パラメータを読み取り、共有ファイルシステム *sharefs* を作成します。

```
[sharefs-mds]root@solaris:~# sammkfs -S sharefs
Building 'sharefs' will destroy the contents of devices:
  /dev/dsk/c0t0d0s0
  /dev/dsk/c0t3d0s0
  /dev/dsk/c0t3d0s1
Do you wish to continue? [y/N]yes ...
[sharefs-mds]root@solaris:~#
```

8. 次に、アクティブなサーバーでの共有ファイルシステムのマウントを実行します。

7.1.2.1.3. アクティブなサーバーでの共有ファイルシステムのマウント

1. サーバーに *root* としてログインします。

   この例では、サーバーの名前は *sharefs-mds* です。

```
[sharefs-mds]root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
[sharefs-mds]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
[sharefs-mds]root@solaris:~#
```

3. 「高パフォーマンス **ma** ファイルシステムの構成」で説明したとおりに、オペレーティングシステムの */etc/vfstab* ファイルに新しいファイルシステムを追加します。

   この例では、*vi* テキストエディタで */etc/vfstab* ファイルを開き、*sharefs* ファミリセットデバイスの行を追加します。

```
[sharefs-mds]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
sharefs    -        /sharefs  samfs   -     no
```

4. 「*Mount Options*」列に *shared* オプションを入力します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
sharefs    -        /sharefs  samfs   -     no       shared
```

5. その他の必要な変更を */etc/vfstab* ファイルに行います。

   たとえば、最初の試行に失敗した場合にバックグラウンドでファイルシステムのマウントを再試行するには、「*Mount Options*」フィールドに *bg* マウントオプションを追加します(指定可能なマウントオプションの詳細は、*mount_samfs* のマニュアルページを参照)。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
```

```
/proc      -        /proc     proc    -     no       -
...
sharefs    -        /sharefs  samfs   -     no       shared,bg
```

6. */etc/vfstab* ファイルを保存して、エディタを終了します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
sharefs    -        /sharefs  samfs   -     no       shared,bg
:wq
[sharefs-mds]root@solaris:~#
```

7. */etc/vfstab* ファイルで指定されたマウントポイントを作成し、そのマウントポイントに対するアクセス権を設定します。

   マウントポイントディレクトリに移動し、マウント済みファイルシステム内のファイルにアクセスするには、メタデータサーバー上およびすべてのクライアント上でマウントポイントアクセス権が同じである必要があり、ユーザーが実行 (*x*) 権限を持っている必要があります。この例では、*/sharefs* マウントポイントディレクトリを作成し、アクセス権を *755* (*-rwxr-xr-x*) に設定します。

```
[sharefs-mds]root@solaris:~# mkdir /sharefs
[sharefs-mds]root@solaris:~# chmod 755 /sharefs
[sharefs-mds]root@solaris:~#
```

8. 新しいファイルシステムをマウントします。

```
[sharefs-mds]root@solaris:~# mount /sharefs
[sharefs-mds]root@solaris:~#
```

9. ホストに複数のネットワークインタフェースが構成されている場合、「ローカル hosts ファイルを使用したホストネットワーク通信のルーティング」を実行できます。

10. それ以外の場合は、メタデータサーバー上に共有ファイルシステムが作成されたら、「共有するファイルシステムクライアントの構成」を開始します。

7.1.2.2. 共有するファイルシステムクライアントの構成

クライアントには、純粋にクライアントとして構成されているホストと、潜在的なメタデータサーバーとして構成されているホストの両方が含まれています。ほとんどの点で、クライアントの構成はサーバーの構成と同じです。各クライアントには、サーバーとまったく同じデバイスが含まれています。マウントオプションとデバイスへの正確なパスのみが異なります (コントローラ番号は各クライアントホストで割り当てられるため、異なる可能性があります)。

共有ファイルシステムがサポートされるように 1 つ以上のクライアントを構成するには、次に示すタスクを実行します。

- Solaris クライアントでの共有ファイルシステムの作成
- Solaris クライアントでの共有ファイルシステムのマウント
- Linux クライアントでの共有ファイルシステムの作成 (存在する場合)
- Linux クライアントでの共有ファイルシステムのマウント (存在する場合)。

7.1.2.2.1. Solaris クライアントでの共有ファイルシステムの作成

クライアントごとに、次の手順を実行します。

1. クライアント上で、*root* としてログインします。

   この例では、サーバーは *sharefs-client1* という名前です。

```
[sharefs-client1]root@solaris:~#
```

2. 端末ウィンドウで、コマンド *samfsconfig device-path* を入力します。ここで *device-path* は、コマンドがファイルシステムディスクデバイスの検索を開始する場所 (*/dev/dsk/** や */dev/zvol/dsk/rpool/** など) です。

   samfsconfig コマンドは、共有ファイルシステムの構成情報を取得します。

```
[sharefs-client1]root@solaris:~# samfsconfig /dev/dsk/*
```

3. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っているため、潜在的なメタデータサーバーとしての使用に適している場合は、*samfsconfig* の出力が、ファイルシステムのメタデータサーバーで作成された *mcf* ファイルに酷似しています。

この例では、ホスト *sharefs-client1* がメタデータデバイス (装置タイプ *mm*) へのアクセス権を持っているため、コマンドの出力にサーバー *sharefs-mds* 上の *mcf* ファイルに一覧表示されるものと同じ装置が表示されます。ホストで割り当てられたデバイスコントローラ番号のみが異なります。

```
[sharefs-client1]root@solaris:~# samfsconfig /dev/dsk/*
# Family Set 'sharefs' Created Thu Feb 21 07:17:00 2013
# Generation 0 Eq count 4 Eq meta count 1
#
sharefs             300         ma         sharefs   -
/dev/dsk/c1t0d0s0    301         mm         sharefs   -
/dev/dsk/c1t3d0s0    302         mr         sharefs   -
/dev/dsk/c1t3d0s1    303         mr         sharefs   -
```

4. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていない場合は、*samfsconfig* コマンドではメタデータデバイスを検索できません。したがって、検出された Oracle HSM デバイスをファイルシステム構成に合わせることはできません。コマンドの出力では、「*Missing Slices*」の下に「*Ordinal 0*」(メタデータデバイス) が一覧表示されますが、ファイルシステムファミリセットを識別する行を含めることができず、データデバイスの一覧がコメントアウトされています。

   この例では、ホスト *sharefs-client2* はデータデバイスへのアクセス権のみを持っています。したがって、*samfsconfig* の出力は次のように表示されます。

```
[sharefs-client2]root@solaris:~# samfsconfig /dev/dsk/*
# Family Set 'sharefs' Created Thu Feb 21 07:17:00 2013
#
# Missing slices
# Ordinal 0
# /dev/dsk/c4t3d0s0    302         mr         sharefs   -
# /dev/dsk/c4t3d0s1    303         mr         sharefs   -
```

5. *samfsconfig* の出力から共有ファイルシステムのエントリをコピーします。次に、2 つ目のウィンドウからテキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開き、コピーしたエントリをファイルにペーストします。

1 つ目の例では、ホスト *sharefs-client1* がファイルシステムのメタデータデバイスへのアクセス権を持っているため、*mcf* ファイルの始まりは次のように表示されます。

```
[sharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300        ma         sharefs    -
 /dev/dsk/c1t0d0s0     301        mm         sharefs    -
 /dev/dsk/c1t3d0s0     302        mr         sharefs    -
 /dev/dsk/c1t3d0s1     303        mr         sharefs    -
```

2 つ目の例では、ホスト *sharefs-client2* がファイルシステムのメタデータデバイスへのアクセス権を持っていないため、*mcf* ファイルの始まりは次のように表示されます。

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
# /dev/dsk/c4t3d0s0    302         mr         sharefs   -
# /dev/dsk/c4t3d0s1    303         mr         sharefs   -
```

6. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っている場合は、共有ファイルシステムのエントリの「*Additional Parameters*」フィールドに、*shared* パラメータを追加します。

   この例では、ホスト *sharefs-client1* はメタデータへのアクセス権を持っています。

```
[sharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300        ma         sharefs    -        shared
 /dev/dsk/c1t0d0s0     301        mm         sharefs    -
 /dev/dsk/c1t3d0s0     302        mr         sharefs    -
 /dev/dsk/c1t3d0s1     303        mr         sharefs    -
```

7. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていない場合は、共有ファイルシステムの行を追加し、*shared* パラメータを追加します。

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
sharefs               300        ma         sharefs    -        shared
# /dev/dsk/c4t3d0s0    302         mr         sharefs   -
# /dev/dsk/c4t3d0s1    303         mr         sharefs   -
```

8. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていない場合は、メタデータデバイスの行を追加します。「*Equipment Identifier*」フィールドを「*nodev*」(デバイスなし) に設定し、残りのフィールドはメタデータサーバーの場合とまったく同じ値に設定します。

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   ---------------
sharefs               300         ma        sharefs    on       shared
 nodev                301         mm        sharefs    on
# /dev/dsk/c4t3d0s0    302         mr         sharefs    -
# /dev/dsk/c4t3d0s1    303         mr         sharefs    -
```

9. ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていない場合は、データデバイスのエントリのコメントを解除します。

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#-----------------   ---------  ---------  ---------  ------   -----------------
sharefs              300         ma        sharefs     on       shared
 nodev                301         mm        sharefs     on
 /dev/dsk/c4t3d0s0    302         mr        sharefs     -
 /dev/dsk/c4t3d0s1    303         mr        sharefs     -
```

10. すべてのデバイスで「*Device State*」フィールドが「*on*」に設定されていることを確認し、*mcf* ファイルを保存します。

    1 つ目の例では、ホスト *sharefs-client1* がファイルシステムのメタデータデバイスへのアクセス権を持っているため、*mcf* ファイルの終わりは次のように表示されます。

```
[sharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300        ma         sharefs    on       shared
 /dev/dsk/c1t0d0s0    301        mm         sharefs    on
 /dev/dsk/c1t3d0s0     302        mr         sharefs    on
 /dev/dsk/c1t3d0s1     303        mr         sharefs    on
:wq
[sharefs-client1]root@solaris:~#
```

    2 つ目の例では、ホスト *sharefs-client2* がファイルシステムのメタデータデバイスへのアクセス権を持っていないため、*mcf* ファイルの終わりは次のように表示されます。

```
[sharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300         ma        sharefs    on       shared
 nodev                301         mm        sharefs    on
 /dev/dsk/c4t3d0s0     302         mr        sharefs    on
 /dev/dsk/c4t3d0s1     303         mr        sharefs    on
:wq
[sharefs-client2]root@solaris:~#
```

11. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、エラーがあれば修正します。

    sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、*sharefs-client1* 上で *mcf* ファイルを確認します。

```
[sharefs-client1]root@solaris:~# sam-fsd
...
Would start sam-stagerd()
Would start sam-amld()
[sharefs-client1]root@solaris:~#
```

12. この時点で、ホストに複数のネットワークインタフェースが構成されている場合は、ローカル hosts ファイルを使用したホストネットワーク通信のルーティングを実行できます。
13. 次に、Solaris クライアントでの共有ファイルシステムのマウントを実行します。

7.1.2.2.2. Solaris クライアントでの共有ファイルシステムのマウント

クライアントごとに、次の手順を実行します。

1. Solaris クライアント上で、*root* としてログインします。

   この例では、サーバーは *sharefs-client1* という名前です。

```
[sharefs-client1]root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
[sharefs-client1]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
[sharefs-client1]root@solaris:~#
```

3. テキストエディタで */etc/vfstab* ファイルを開き、共有ファイルシステムの行を追加します。

   この例では、*vi* テキストエディタでファイルを開き、*sharefs* ファミリセットデバイスの行を追加します。

```
[sharefs-client1]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
```

```
/proc      -        /proc     proc    -     no       -
...
sharefs    -        /sharefs  samfs   -     no
```

4. セパレータとしてコンマを使用して、その他の必要なマウントオプションを追加し、その他の必要な変更を */etc/vfstab* ファイルに加えます。次に、*/etc/vfstab* ファイルを保存します。

   この例では、マウントオプションを追加しません。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ------------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
sharefs    -        /sharefs  samfs   -     no       -
:wq
[sharefs-client1]root@solaris:~#
```

5. */etc/vfstab* ファイルで指定されたマウントポイントを作成し、そのマウントポイントに対するアクセス権を設定します。

   マウントポイントのアクセス権は、メタデータサーバーおよびその他のすべてのクライアントと同じにする必要があります。ユーザーはマウントポイントポイントディレクトリに移動し、マウントしたファイルシステム内のファイルにアクセスするための実行権 (*x*) を持っている必要があります。この例では、*/sharefs* マウントポイントディレクトリを作成し、アクセス権を *755* (*-rwxr-xr-x*) に設定します。

```
[sharefs-client1]root@solaris:~# mkdir /sharefs
[sharefs-client1]root@solaris:~# chmod 755 /sharefs
[sharefs-client1]root@solaris:~#
```

6. 共有ファイルシステムをマウントします。

```
[sharefs-client1]root@solaris:~# mount /sharefs
[sharefs-client1]root@solaris:~#
```

7. 共有ファイルシステムに Linux クライアントが含まれている場合は、Linux クライアントでの共有ファイルシステムの作成を実行します。
8. Oracle HSM 共有アーカイブファイルシステムを構成している場合は、「共有ファイルシステム用のアーカイブストレージの構成」の次のタスクに進みます。
9. それ以外の場合は、ここで終了します。Oracle HSM 共有ファイルシステムが構成されました。

7.1.2.2.3. Linux クライアントでの共有ファイルシステムの作成

クライアントごとに、次の手順を実行します。

1. Linux クライアント上で、*root* としてログインします。

   この例では、Linux クライアントホストの名前は *sharefs-clientL* です。

```
[sharefs-clientL][root@linux ~]#
```

2. 端末ウィンドウで、コマンド *samfsconfig device-path* を入力します。ここで *device-path* は、コマンドがファイルシステムディスクデバイスの検索を開始する場所 (*/dev/** など) です。

   samfsconfig コマンドは、共有ファイルシステムの構成情報を取得します。Linux ホストがファイルシステムのメタデータデバイスへのアクセス権を持っていないため、*samfsconfig* コマンドではメタデータデバイスを検索できません。したがって、検出された Oracle HSM デバイスをファイルシステム構成に合わせることはできません。コマンドの出力では、「*Missing Slices*」の下に「*Ordinal 0*」(メタデータデバイス) が一覧表示されますが、ファイルシステムファミリセットを識別する行を含めることができず、データデバイスの一覧がコメントアウトされています。

   この例では、Linux ホスト *sharefs-clientL* 用の *samfsconfig* 出力が次のように表示されます。

```
[sharefs-clientL][root@linux ~]# samfsconfig /dev/*
# Family Set 'sharefs' Created Thu Feb 21 07:17:00 2013
#
# Missing slices
# Ordinal 0
# /dev/sda4            302         mr         sharefs   -
```

```
# /dev/sda5            303         mr         sharefs   -
```

3. *samfsconfig* の出力から共有ファイルシステムのエントリをコピーします。次に、2 つ目のウィンドウからテキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開き、コピーしたエントリをファイルにペーストします。

   この例では、Linux ホスト *sharefs-clientL* 用の *mcf* ファイルの始まりが次のように表示されます。

```
[sharefs-clientL][root@linux ~]# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
# /dev/sda4           302        mr         sharefs   -
# /dev/sda5           303        mr         sharefs   -
```

4. **mcf** ファイルに共有ファイルシステムの行を挿入し、**shared** パラメータを追加します。

```
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300        ma         sharefs    -        shared
# /dev/sda4            302         mr         sharefs   -
# /dev/sda5            303         mr         sharefs   -
```

5. *mcf* ファイルに、ファイルシステムのメタデータデバイスの行を挿入します。Linux ホストはメタデータデバイスへのアクセス権を持っていないため、「*Equipment Identifier*」フィールドを「*nodev*」(デバイスなし) に設定し、残りのフィールドはメタデータサーバーの場合とまったく同じ値に設定します。

```
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300         ma        sharefs    on       shared
 nodev                301         mm        sharefs    on
# /dev/sda4            302         mr         sharefs   -
# /dev/sda5            303         mr         sharefs   -
```

6. **mcf** ファイルで、データデバイスのエントリをコメント解除します。

```
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs              300         ma        sharefs     on       shared
 nodev                301         mm        sharefs     on
 /dev/sda4           302         mr        sharefs     -
 /dev/sda5           303         mr        sharefs     -
```

7. すべてのデバイスで「*Device State*」フィールドが「*on*」に設定されていることを確認し、*mcf* ファイルを保存します。

```
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#------------------   ---------  ---------  ---------  ------   ---------------
sharefs               300         ma        sharefs    on       shared
 nodev                 301         mm        sharefs    on
 /dev/sda4             302         mr        sharefs    on
 /dev/sda5             303         mr        sharefs    on
:wq
[sharefs-clientL][root@linux ~]#
```

8. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認し、エラーがあれば修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、Linux クライアント *sharefs-clientL* 上で *mcf* ファイルを確認します。

```
[sharefs-clientL][root@linux ~]# sam-fsd
...
Would start sam-stagerd()
Would start sam-amld()
[sharefs-clientL][root@linux ~]#
```

9. 次に、Linux クライアントでの共有ファイルシステムのマウントを実行します。

7.1.2.2.4. Linux クライアントでの共有ファイルシステムのマウント

クライアントごとに、次の手順を実行します。

1. Linux クライアント上で、*root* としてログインします。

   この例では、Linux クライアントホストの名前は *sharefs-clientL* です。

```
[sharefs-clientL][root@linux ~]#
```

2. オペレーティングシステムの */etc/fstab* ファイルをバックアップします。

```
[sharefs-clientL][root@linux ~]# cp /etc/fstab /etc/fstab.backup
```

3. テキストエディタで */etc/fstab* ファイルを開き、共有ファイルシステムの行を開始します。

   この例では、*sharefs-clientL* 上で */etc/fstab* ファイルをバックアップしたあとに、*vi* テキストエディタでファイルを開き、*sharefs* ファミリセットデバイスの行を追加します。

```
[sharefs-clientL][root@linux ~]# vi /etc/fstab
#File
#Device    Mount    System    Mount                     Dump      Pass
#to Mount  Point    Type      Options                   Frequency Number
#--------  -------  --------  ------------------------  --------- ------
...
/proc      /proc     proc     defaults
sharefs    /sharefs  samfs
```

4. ファイルの 4 列目で、必須の *shared* マウントオプションを追加します。

```
#File
#Device    Mount    System    Mount                     Dump      Pass
#to Mount  Point    Type      Options                   Frequency Number
#--------  -------  --------  ------------------------  --------- ------
...
/proc      /proc     proc     defaults
```

```
sharefs    /sharefs  samfs    shared
```

5. ファイルの 4 列目で、セパレータとしてコンマを使用して、その他の必要なマウントオプションを追加します。

   Linux クライアントでは、次の追加マウントオプションがサポートされています。

   - *rw*, *ro*
   - *retry*
   - *meta_timeo*
   - *rdlease*, *wrlease*, *aplease*
   - *minallocsz*, *maxallocsz*
   - *noauto*, *auto*

   この例では、オプション *noauto* を追加します。

```
#File
#Device    Mount    System    Mount                      Dump      Pass
#to Mount  Point    Type      Options                    Frequency Number
#--------  -------  --------  -------------------------  --------- ------
...
/proc      /proc     proc     defaults
sharefs    /sharefs  samfs    shared,noauto
```

6. ファイルの残りの 2 列には、それぞれゼロ (*0*) を入力します。次に、*/etc/fstab* ファイルを保存します。

```
#File
#Device    Mount    System    Mount                      Dump      Pass
#to Mount  Point    Type      Options                    Frequency Number
#--------  -------  --------  -------------------------  --------- ------
...
/proc      /proc     proc     defaults
sharefs    /sharefs  samfs    shared,noauto              0         0
:wq
[sharefs-clientL][root@linux ~]#
```

7. */etc/fstab* ファイルで指定されたマウントポイントを作成し、そのマウントポイントに対するアクセス権を設定します。

   マウントポイントのアクセス権は、メタデータサーバーおよびその他のすべてのクライアントと同じにする必要があります。ユーザーはマウントポイントポイントディレクトリに移動し、マウントしたファイルシステム内のファイルにアクセスするための実行権 (*x*) を持っている必要があります。この例では、*/sharefs* マウントポイントディレクトリを作成し、アクセス権を *755* (*-rwxr-xr-x*) に設定します。

```
[sharefs-clientL][root@linux ~]# mkdir /sharefs
[sharefs-clientL][root@linux ~]# chmod 755 /sharefs
```

8. 共有ファイルシステムをマウントします。コマンド *mount mountpoint* を使用します。ここで *mountpoint* は、*/etc/fstab* ファイルで指定されたマウントポイントです。

   例で示すように、*mount* コマンドは警告を生成します。これは通常の動作であり、無視できます。

```
[sharefs-clientL][root@linux ~]# mount /sharefs
Warning: loading SUNWqfs will taint the kernel: SMI license
See http://www.tux.org/lkml/#export-tainted for information
about tainted modules. Module SUNWqfs loaded with warnings
[sharefs-clientL][root@linux ~]#
```

9. Oracle HSM 共有アーカイブファイルシステムを構成している場合は、「共有ファイルシステム用のアーカイブストレージの構成」の次のタスクに進みます
10. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
11. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

7.1.2.2.5. ローカル hosts ファイルを使用したホストネットワーク通信のルーティング

個別のホストには、ローカル hosts ファイルは必要ありません。ファイルシステムは、すべてのファイルシステムホストについて、アクティブなメタデータサーバーとアクティブおよび潜在的なメタデータサーバーのネットワークインタフェースを識別します (「アクティブおよび潜在的なメタデータサーバーでのホストファイルの作成」を参照)。ただし、複数のネットワークインタ

フェースを持つファイルシステムホスト間で、ネットワークトラフィックを選択的にルーティングする必要がある場合は、ローカル hosts ファイルが役立ちます。

それぞれのファイルシステムホストは、メタデータサーバー上のほかのホストでネットワークインタフェースを検索します。ファイルシステムのグローバル hosts ファイル */etc/opt/SUNWsamfs/hosts.family-set-name* にホスト名と IP アドレスが一覧表示されます。ここで *family-set-name* は、共有ファイルシステムのファミリセット名です。その後、ホストはローカル hosts ファイル */etc/opt/SUNWsamfs/hosts.family-set-name.local* を検索します。

ローカル hosts ファイルがない場合、ホストは、グローバル hosts ファイルで指定されたインタフェースアドレスを使用します。ホストは、グローバルファイルで指定された順序で使用されます。

ローカル hosts ファイルが存在する場合、ホストはグローバルファイルと比較して、両方のファイルに一覧表示されたインタフェースのみを使用します。ホストは、ローカルファイルで指定された順序で使用されます。

そのため、各ファイルでさまざまなアドレスを使用すると、さまざまなホストで使用されているインタフェースを制御できます。ローカル hosts ファイルを構成するには、次に概要を示す手順を使用します。

1. それぞれのアクティブおよび潜在的なメタデータサーバーホスト上で、共有ファイルシステムのグローバル hosts ファイルを編集して、必要な方法でサーバーとホストの通信がルーティングされるようにします。

   このセクションの例では、共有ファイルシステム *sharefs2nic* に、アクティブなメタデータサーバー *sharefs2-mds*、および潜在的なメタデータサーバー *sharefs2-mds_alt* (それぞれが 2 つのネットワークインタフェースを持つ) が含まれています。また、2 つのクライアント (*sharefs2-client1* と *sharefs2-client2*) も存在します。

   アクティブおよび潜在的なメタデータサーバーが、プライベートネットワークアドレスを使用して相互に通信し、DNS (Domain Name Service) でパブリック LAN (Local Area Network) 上のアドレスに解決できるホスト名を使用してクライアントと通信する必要があります。

   そのため、ファイルシステムのグローバルホストファイルである */etc/opt/SUNWsamfs/hosts.sharefs2* を編集します。アクティブおよび潜在的なサーバーには、プライベートネットワークインタフェースアドレスを指定します。ただし、クライアントには、アドレスではなくホスト名を指定します。

```
[sharefs2-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2
# /etc/opt/SUNWsamfs/hosts.sharefs2
#                                   Server   On/  Additional
#Host Name        Network Interface  Ordinal  Off  Parameters
#--------------  -----------------  -------  ---  ----------
sharefs2-mds      172.16.0.129       1        0   server
sharefs2-mds_alt  172.16.0.130       2        0
sharefs2-client1  sharefs2-client1   0        0
sharefs2-client2  sharefs2-client2   0        0
:wq
[sharefs2-mds]root@solaris:~#
```

2. パスとファイル名 */etc/opt/SUNWsamfs/hosts.family-set-name.local* を使用して、それぞれのアクティブおよび潜在的なメタデータサーバー上でローカル hosts ファイルを作成します。ここで *family-set-name* は、共有ファイルシステムの装置識別子です。アクティブおよび潜在的なサーバーで使用するネットワーク用のインタフェースのみを含めてください。

   この例では、アクティブおよび潜在的なメタデータサーバーがプライベートネットワークを介して相互に通信するように、各サーバー上のローカル hosts ファイル */hosts.sharefs2.local* には、2 つのホスト (アクティブおよび潜在的なメタデータサーバー) のプライベートアドレス のみが一覧表示されています。

```
[sharefs2-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2.local
# /etc/opt/SUNWsamfs/hosts.sharefs2 on sharefs2-mds
#                                   Server   On/  Additional
#Host Name        Network Interface  Ordinal  Off  Parameters
#--------------  -----------------  -------  ---  ----------
sharefs2-mds      172.16.0.129       1        0   server
sharefs2-mds_alt  172.16.0.130       2        0
:wq
[sharefs2-mds]root@solaris:~# ssh root@sharefs2-mds_alt
Password:


[sharefs2-mds_alt]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2.local
# /etc/opt/SUNWsamfs/hosts.sharefs2.local on sharefs2-mds_alt
#                                   Server   On/  Additional
```

```
#Host Name        Network Interface  Ordinal  Off  Parameters
#--------------  -----------------  -------  ---  ----------
sharefs2-mds      172.16.0.129        1        0   server
sharefs2-mds_alt  172.16.0.130        2        0
:wq
[sharefs2-mds_alt]root@solaris:~# exit
[sharefs2-mds]root@solaris:~#
```

3. パスとファイル名 */etc/opt/SUNWsamfs/hosts.family-set-name.local* を使用して、各クライアント上でローカル hosts ファイルを作成します。ここで *family-set-name* は、共有ファイルシステムの装置識別子です。クライアントで使用するネットワーク用のインタフェースのみを含めてください。

   この例では、クライアントはパブリックネットワーク経由でのみサーバーと通信する必要があります。そのため、ファイルには、2 つのホスト (アクティブおよび潜在的なメタデータサーバー) のホスト名のみを含めます。

```
[sharefs2-mds]root@solaris:~# ssh root@sharefs2-client1
Password:
[sharefs2-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2.local
# /etc/opt/SUNWsamfs/hosts.sharefs2.local on sharefs2-client1
#                                    Server   On/  Additional
#Host Name        Network Interface  Ordinal  Off  Parameters
#--------------  -----------------  -------  ---  ----------
sharefs2-mds      sharefs2-mds        1        0   server
sharefs2-mds_alt  sharefs2-mds_alt    2        0
:wq
[sharefs2-client1]root@solaris:~# exit
[sharefs2-mds]root@solaris:~# ssh root@sharefs2-client2
Password:

[sharefs2-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sharefs2.local
# /etc/opt/SUNWsamfs/hosts.sharefs2.local on sharefs2-client2
#                                    Server   On/  Additional
#Host Name        Network Interface  Ordinal  Off  Parameters
#--------------  -----------------  -------  ---  ----------
sharefs2-mds      sharefs2-mds        1        0    server
```

```
sharefs2-mds_alt   sharefs2-mds_alt     2         0
:wq
[sharefs2-client2]root@solaris:~# exit
[sharefs2-mds]root@solaris:~#
```

4. サーバーの構成が完了したときに、この手順を開始した場合は、「アクティブなサーバーでの共有ファイルシステムのマウント」に進みます。
5. クライアントを構成しているときに、この手順を開始した場合は、ここで「Solaris クライアントでの共有ファイルシステムのマウント」を実行する必要があります。

7.1.2.3. 共有ファイルシステム用のアーカイブストレージの構成

アーカイブ Oracle HSM 共有ファイルシステム用にアーカイブストレージを設定するには、次のタスクを実行します。

- 永続的なバインドを使用したサーバーおよびデータムーバーホストへのテープドライブの接続
- アーカイブストレージを使用するためのアーカイブファイルシステムのホストの構成
- 共有アーカイブファイルシステムのホスト間でのテープ入出力の分散 (必要な場合)。

7.1.2.3.1. 永続的なバインドを使用したサーバーおよびデータムーバーホストへのテープドライブの接続

共有アーカイブファイルシステムでは、すべての潜在的なメタデータサーバーにライブラリおよびテープドライブへのアクセス権が必要です。共有アーカイブファイルシステムのホスト間でのテープ入出力の分散を実行すると決定した場合は、1 つ以上のクライアントにもドライブへのアクセス権が必要です。したがって、それぞれのドライブに整合性のある方法で対処できるように、これらの各ホストを構成する必要があります。

Solaris オペレーティングシステムは、起動時にデバイスが検出される順序でドライブをシステムデバイスツリーに追加します。この順序によって、その他のファイルシステムホストでデバイスが検出される順序や、リムーバブルメディアライブラリに物理的にインストールされる順序が反映される場合と、反映されない場合があります。したがって、その他のホストにバインドするときと同じ方法、およびリムーバブルメディアライブラリにインストールされるときと同じ順序で、デバイスを各ホストに永続的にバインドする必要があります。

次の手順では、必要な手順の概要を示します (永続的なバインドの作成についての詳細は、*devfsadm* と *devlinks* のマニュアルページ、および使用中の Solaris オペレーティングシステムバージョンに対応した管理ドキュメントを参照)。

1. アクティブなメタデータサーバーに *root* としてログインします。

```
[sharefs-mds]root@solaris:~#
```

2. ライブラリ内のドライブの現在の物理的な順序を把握していない場合は、「ドライブをライブラリに取り付ける順序の確認」で説明したとおりに、マッピングファイルを作成します。

   この例では、*device-mappings.txt* ファイルは次のように表示されます。

```
LIBRARY SOLARIS          SOLARIS
DEVICE  LOGICAL          PHYSICAL
NUMBER  DEVICE           DEVICE
------- -------------    --------------------------------------------------
   2    /dev/rmt/0cbn -> ../../devices/pci@8,.../st@w500104f00093c438,0:cbn
   1    /dev/rmt/1cbn -> ../../devices/pci@8,.../st@w500104f0008120fe,0:cbn
   3    /dev/rmt/2cbn -> ../../devices/pci@8,.../st@w500104f000c086e1,0:cbn
   4    /dev/rmt/3cbn -> ../../devices/pci@8,.../st@w500104f000b6d98d,0:cbn
```

3. テキストエディタで */etc/devlink.tab* ファイルを開きます。

   この例では、*vi* エディタを使用します。

```
[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
# Copyright (c) 1993, 2011, Oracle and/or its affiliates. All rights reserved.
# This is the table used by devlinks
# Each entry should have 2 fields; but may have 3.  Fields are separated
# by single tab ('/t') characters.
...
```

4. ガイドとして *device-mappings.txt* ファイルを使用して、Solaris テープドライブツリー内の開始ノード *rmt/node-number* をライブラリ内の 1 番目のドライブに再マッピングする 1 行を */etc/devlink.tab* ファイルに追加します。*type=ddi_byte:tape; addr=device_address,0; rmt/node-number/M0* 形式で行を入力します。ここで *device_address* は、デバイスの物理アドレスで、*node-number* は、Solaris で自動的に構成されるデバイスとの競合を回避するために十分に大きい Solaris デバイスツリー内の位置です (Solaris はノード *0* から起動されます)。

この例では、ライブラリ内の 1 番目のデバイス *1* のデバイスアドレス *w500104f0008120fe* を書き留め、デバイスが *rmt/1* にあるホストに現在接続されていることを確認します。

```
[sharefs-mds] vi /root/device-mappings.txt
LIBRARY SOLARIS          SOLARIS
DEVICE  LOGICAL          PHYSICAL
NUMBER  DEVICE           DEVICE
------- -----------   ------------------------------------------------------
   2    /dev/rmt/0cbn -> ../../devices/pci@8,.../st@w500104f00093c438,0:cbn
   1    /dev/rmt/1cbn -> ../../devices/pci@8,.../st@w500104f0008120fe,0:cbn
   3    /dev/rmt/2cbn -> ../../devices/pci@8,.../st@w500104f000c086e1,0:cbn
   4    /dev/rmt/3cbn -> ../../devices/pci@8,.../st@w500104f000b6d98d,0:cbn
```

そのため、競合しないノード *rmt/60* をライブラリ *w500104f0008120fe* 内の番号 *1* のドライブに再マッピングする行を */etc/devlink.tab* に作成します。

```
[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
# Copyright (c) 1993, 2011, Oracle and/or its affiliates. All rights reserved.
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
:w
```

5. メタデータサーバー上のデバイスツリー内のドライブ順序とライブラリへのインストール順序が一致するように、Oracle HSM アーカイブ用に割り当てられているテープデバイスごとに、*/etc/devlink.tab* ファイルへの行の追加を繰り返します。ファイルを保存します。

   この例では、残りの 3 つのデバイス (*w500104f00093c438* にあるライブラリドライブ *2*、*w500104f000c086e1* にあるライブラリドライブ *3*、*w500104f000c086e1* にあるライブラリドライブ *4*) の順序とアドレスを書き留めます。

```
[sharefs-mds]root@solaris:~# vi /root/device-mappings.txt
...
   2    /dev/rmt/0cbn -> ../../devices/pci@8,.../st@w500104f00093c438,0:cbn
   1    /dev/rmt/1cbn -> ../../devices/pci@8,.../st@w500104f0008120fe,0:cbn
```

```
3    /dev/rmt/2cbn -> ../../devices/pci@8,.../st@w500104f000c086e1,0:cbn
4    /dev/rmt/3cbn -> ../../devices/pci@8,.../st@w500104f000b6d98d,0:cbn
```

その後、ライブラリ内と同じ順序を維持して、デバイスアドレスを次の 3 つの Solaris デバイスノード (**rmt/61**、**rmt/62**、および **rmt/63**) にマッピングします。

```
[sharefs-mds]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;    rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;    rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;    rmt/63/M0
:wq
[sharefs-mds]root@solaris:~#
```

6. */dev/rmt* 内のテープデバイスへの既存のリンクをすべて削除します。

```
[sharefs-mds]root@solaris:~# rm /dev/rmt/*
```

7. */etc/devlink.tab* ファイル内のエントリから、新しい永続的なテープデバイスのリンクを作成します。コマンド *devfsadm -c tape* を使用します。

   devfsadm コマンドを実行するたびに、*/etc/devlink.tab* ファイルで指定された構成を使用して、そのファイルに指定されたデバイス用に新しいテープデバイスのリンクが作成されます。*-c tape* オプションを指定すると、テープクラスデバイス用の新しいリンクのみが作成されるようにコマンドが制限されます。

```
[sharefs-mds]root@solaris:~# devfsadm -c tape
```

8. 共有ファイルシステム構成内のそれぞれの潜在的なメタデータサーバーとデータムーバー上で同じ永続的なテープデバイスリンクを作成します。*/etc/devlink.tab* ファイルに同じ行を追加し、*/dev/rmt* 内のリンクを削除して、*devfsadm -c tape* を実行します。

   この例では、潜在的なメタデータサーバー *sharefs-mds_alt* およびデータムーバークライアント *sharefs-client1* があります。したがって、アクティブなサーバー *sharefs-mds* 上のファイルと一致するように、それぞれの */etc/devlink.tab* ファイルを編集し

ます。次に、*sharefs-mds_alt* および *sharefs-client1* 上の */dev/rmt* にある既存のリンクを削除して、それぞれで *devfsadm -c tape* を実行します。

```
[sharefs-mds]root@solaris:~# ssh sharefs-mds_alt
Password:
[sharefs-mds_alt]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;    rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;    rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;    rmt/63/M0
:wq
[sharefs-mds_alt]root@solaris:~# rm /dev/rmt/*
[sharefs-mds_alt]root@solaris:~# devfsadm -c tape
[sharefs-mds_alt]root@solaris:~# exit
[sharefs-mds]root@solaris:~# ssh sharefs-client1
Password:
[sharefs-client1]root@solaris:~# vi /etc/devlink.tab
...
type=ddi_byte:tape;addr=w500104f0008120fe,0;    rmt/60/M0
type=ddi_byte:tape;addr=w500104f00093c438,0;    rmt/61/M0
type=ddi_byte:tape;addr=w500104f000c086e1,0;    rmt/62/M0
type=ddi_byte:tape;addr=w500104f000b6d98d,0;    rmt/63/M0
:wq
[sharefs-client1]root@solaris:~# rm /dev/rmt/*
[sharefs-client1]root@solaris:~# devfsadm -c tape
[sharefs-client1]root@solaris:~# exit
[sharefs-mds]root@solaris:~#
```

9. 次に、アーカイブストレージを使用するためのアーカイブファイルシステムのホストの構成を実行します。

7.1.2.3.2. アーカイブストレージを使用するためのアーカイブファイルシステムのホストの構成

アクティブなメタデータサーバー、およびそれぞれの潜在的なメタデータサーバーとデータムーバークライアントで、次の手順を実行します。

1. ホストに *root* としてログインします。

```
[sharefs-host]root@solaris:~#
```

2. テキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開きます。

   次の例では、*vi* エディタを使用します。

```
[sharefs-host]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
sharefs                  100       ms        sharefs on
 /dev/dsk/c1t3d0s3        101       md        sharefs on
 /dev/dsk/c1t3d0s4        102       md        sharefs on
...
```

3. */etc/opt/SUNWsamfs/mcf* ファイル内のファイルシステム定義のあとに、アーカイブストレージ装置のセクションを開始します。

   この例では、わかりやすくするため見出しをいくつか追加します。

```
[sharefs-host]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
```

4. アーカイブテープストレージを追加するには、ライブラリのエントリの追加から開始します。「Equipment Identifier」フィールドで、ライブラリのデバイス ID を入力し、装置番号を割り当てます。

   この例では、ライブラリの装置 ID は */dev/scsi/changer/c1t0d5* です。装置番号は *900* (ディスクアーカイブ用に選択された範囲に続く範囲) に設定します。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
/dev/scsi/changer/c1t0d5 900
```

5. 装置タイプを汎用 SCSI 接続テープオプションライブラリ *rb* に設定して、テープライブラリファミリセットの名前を指定して、デバイスの状態を *on* に設定します。

   この例では、ライブラリ *library1* を使用しています。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
/dev/scsi/changer/c1t0d5 900       rb        library1  on
```

6. 「*Additional Parameters*」列で、ライブラリカタログのオプションのユーザー定義のパスと名前を入力できます。

   オプションのデフォルト以外のパスは 127 文字を超えることはできません。この例では、ユーザー定義のカタログファイル名 *library1cat* とともにデフォルトのパス *var/opt/SUNWsamfs/catalog/* を使用します。ドキュメントのレイアウト制限のために、例ではパスが省略されています。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
/dev/scsi/changer/c1t0d5 900       rb        library1  on     .../library1cat
```

7. 次に、テープドライブごとにエントリを追加します。「永続的なバインドを使用したサーバーおよびデータムーバーホストへのテープドライブの接続 」の手順で設定した永続的な装置 ID を使用します。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------  ------ -----------------
DISKVOL1                  800       ms       DISKVOL1  on
```

```
 /dev/dsk/c6t0d1s7         801       md       DISKVOL1  on
 /dev/dsk/c4t0d2s7         802       md       DISKVOL1  on
/dev/scsi/changer/c1t0d5  900       rb       library1  on     .../library1cat
 /dev/rmt/60cbn            901       tp       library1 on
 /dev/rmt/61cbn            902       tp       library1  on
 /dev/rmt/62cbn            903       tp       library1  on
 /dev/rmt/63cbn            904       tp       library1  on
```

8. 最後に、Oracle HSM ヒストリアンを自分で構成する場合、装置タイプ *hy* を使用してエントリを追加します。「Family Set」列と「Device State」列にハイフンを入力し、「Additional Parameters」列にヒストリアンのカタログへのパスを入力します。

   ヒストリアンは、アーカイブからエクスポートされたボリュームをカタログする仮想ライブラリです。ヒストリアンを構成しない場合、指定された最大の装置番号に 1 を加えた値を使用して、ソフトウェアによってヒストリアンが自動的に作成されます。

   ページレイアウトの都合上、この例では、ヒストリアンカタログへのパスが短縮されていることに注意してください。フルパスは */var/opt/SUNWsamfs/catalog/historian_cat* です。

```
# Archival storage for copies:
#
# Equipment              Equipment Equipment Family    Device Additional
# Identifier             Ordinal   Type      Set       State  Parameters
#----------------------- --------- --------- --------- ------ ----------------
/dev/scsi/changer/c1t0d5 900       rb        library1  on     ...catalog/library1cat
 /dev/rmt/60cbn           901       tp        library1  on
 /dev/rmt/61cbn           902       tp        library1  on
 /dev/rmt/62cbn           903       tp        library1  on
 /dev/rmt/63cbn           904       tp        library1  on
historian                999       hy        -         -      .../historian_cat
```

9. *mcf* ファイルを保存して、エディタを閉じます。

```
...
 /dev/rmt/3cbn            904       tp        library1  on
historian                999       hy        -      -      .../historian_cat
:wq
```

```
[sharefs-host]root@solaris:~#
```

10. *sam-fsd* コマンドを実行して、*mcf* ファイルにエラーがないかどうかを確認します。見つかったエラーを修正します。

    sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーを検出すると停止します。

```
[sharefs-host]root@solaris:~# sam-fsd
...
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[sharefs-host]root@solaris:~#
```

11. *mcf* ファイルを再度読み取り、それ自体を適宜再構成するように Oracle HSM サービスに指示します。報告されたエラーをすべて修正して、必要に応じて繰り返します。

```
[sharefs-host]root@solaris:~# samd config
Configuring SAM-FS
[sharefs-host]root@solaris:~#
```

12. すべてのアクティブおよび潜在的なメタデータサーバーと、すべてのデータムーバークライアントがアーカイブストレージを使用するように構成されるまで、この手順を繰り返します。
13. 必要に応じて、共有アーカイブファイルシステムのホスト間でのテープ入出力の分散を実行します。
14. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
15. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

7.1.2.3.3. 共有アーカイブファイルシステムのホスト間でのテープ入出力の分散

Oracle HSM リリース 6.0 以降、Oracle Solaris 11 以上が実行されている共有アーカイブファイルシステムのクライアントは、テープドライブを接続して、ファイルシステムの代わりにテープ入出力を実行できます。これらのデータムーバーホスト間でテープ入出力を分散させると、サーバーのオーバーヘッドが大幅に削減されるため、ファイルシステムのパフォーマンスが

向上し、Oracle HSM 実装のスケーリング時の柔軟性が大幅に向上します。アーカイブの必要性が高くなれば、Oracle HSM メタデータサーバーをより強力なシステムに交換するか (垂直スケーリング)、より多くのクライアント間で負荷を分散させるか (水平スケーリング) の選択肢があります。

共有ファイルシステムホスト間でテープ入出力を分散させるには、次の手順を実行します。

1. テープ入出力を処理するファイルシステムメタデータサーバーおよびすべてのファイルシステムクライアントに、分散入出力で使用されるデバイスをすべて接続します。
2. まだ実行していない場合は、データムーバーとして機能するクライアントごとに、永続的なバインドを使用したサーバーおよびデータムーバーホストへのテープドライブの接続 を実行します。その後、ここに戻ります。
3. 共有アーカイブファイルシステムのメタデータサーバーに *root* としてログインします。

   この例では、サーバーのホスト名は *samsharefs-mds* です。

```
[samsharefs-mds]root@solaris:~#
```

4. メタデータサーバーで Oracle Solaris 11 以上が実行されていることを確認します。

```
[samsharefs-mds]root@solaris:~# uname -r
5.11
[samsharefs-mds]root@solaris:~#
```

5. データムーバーとして動作するすべてのクライアントで、Oracle Solaris 11 以上が実行されていることを確認します。

   この例では、*ssh* を使用してクライアントホスト *samsharefs-client1* および *samsharefs-client2* にリモートログインして、ログインバーナーから Solaris のバージョンを取得します。

```
[samsharefs-mds]root@solaris:~# ssh root@samsharefs-client1
Password:
Oracle Corporation      SunOS 5.11      11.1    September 2013
[samsharefs-client1]root@solaris:~# exit
[samsharefs-mds]root@solaris:~# ssh root@samsharefs-client2
Password:
Oracle Corporation      SunOS 5.11      11.1    September 2013
[samsharefs-client2]root@solaris:~# exit
```

```
[samsharefs-mds]root@solaris:~#
```

6. メタデータサーバー上で、テキストエディタを使用して */etc/opt/SUNWsamfs/defaults.conf* を開きます。必要に応じて、行 *#distio = off* のコメントを解除するか、行がまったく存在しない場合は追加します。

   デフォルトでは、「*distio*」は「*off*」(無効) になっています。

   この例では、*vi* エディタでファイルを開き、行 *distio = on* を追加します。

```
[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
distio = on
```

7. 次に、分散入出力に追加するデバイスタイプを選択します。分散入出力でデバイスタイプ *dev* を使用するには、*defaults.conf* ファイルに行 *dev_distio = on* を追加します。分散入出力からデバイスタイプ *dev* を除外するには、行 *dev_distio = off* を追加します。ファイルを保存します。

   デフォルトでは、StorageTek T10000 ドライブおよび LTO ドライブを分散入出力に追加できますが (*ti_distio = on* および *li_distio = on*)、その他のタイプはすべて除外されています。この例では、LTO ドライブを除外します。

```
[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
distio = on
li_distio = off
:wq
[samsharefs-mds]root@solaris:~#
```

8. データムーバーとして動作する各クライアント上で、サーバー上のファイルと一致するように *defaults.conf* ファイルを編集します。

9. データムーバーとして機能する各クライアント上で、*/etc/opt/SUNWsamfs/mcf* ファイルをテキストエディタで開き、メタデータサーバーが分散テープ入出力で使用しているテープデバイスがすべて含まれるように、ファイルを更新します。デバイスの順序および装置番号がメタデータサーバー上の *mcf* ファイルのものと同じであることを確認します。

   この例では、*vi* エディタを使用して、ホスト *samsharefs-client1* 上に *mcf* ファイルを構成します。

```
[samsharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment              Equipment Equipment Family      Device Additional
# Identifier             Ordinal   Type      Set         State  Parameters
#----------------------- --------- --------- ----------  ------ --------------
samsharefs               800       ms        samsharefs  on
...
# Archival storage for copies:
 /dev/rmt/60cbn          901       ti                    on
 /dev/rmt/61cbn          902       ti                    on
 /dev/rmt/62cbn          903       ti                    on
 /dev/rmt/63cbn          904       ti                    on
```

10. データムーバーとして動作するクライアント上に、メタデータサーバー上の */etc/opt/SUNWsamfs/mcf* ファイルに一覧表示されたテープライブラリが構成されている場合は、分散テープ入出力用に使用されているテープデバイスのファミリセット名としてライブラリファミリセットを指定します。ファイルを保存します。

    この例では、ホスト *samsharefs-client1* 上にライブラリが構成されているため、テープデバイスのファミリセット名 *library1* を使用します。

```
[samsharefs-client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment              Equipment Equipment Family      Device Additional
# Identifier             Ordinal   Type      Set         State  Parameters
#----------------------- --------- --------- ----------  ------ --------------
samsharefs               800       ms        samsharefs  on
...
# Archival storage for copies:
/dev/scsi/changer/c1t0d5 900       rb        library1    on     .../library1cat
 /dev/rmt/60cbn           901       ti        library1    on
 /dev/rmt/61cbn           902       ti        library1    on
```

```
 /dev/rmt/62cbn           903       ti        library1    on
 /dev/rmt/63cbn           904       ti        library1    on
:wq
[samsharefs-client1]root@solaris:~#
```

11. データムーバーとして動作するクライアント上に、メタデータサーバー上の */etc/opt/SUNWsamfs/mcf* ファイルに一覧表示されたテープライブラリが構成されていない場合は、分散テープ入出力用に使用されているテープデバイスのファミリセット名としてハイフン (-) を使用します。その後、ファイルを保存してエディタを閉じます。

    この例では、ホスト *samsharefs-client2* 上にライブラリが構成されていないため、テープデバイスのファミリセット名としてハイフンを使用します。

```
[samsharefs-client2]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment              Equipment Equipment Family      Device Additional
# Identifier             Ordinal   Type      Set         State  Parameters
#----------------------- --------- --------- ----------  ------ --------------
samsharefs               800       ms        samsharefs  on
...
# Archival storage for copies:
 /dev/rmt/60cbn           901       ti        -           on
 /dev/rmt/61cbn           902       ti        -           on
 /dev/rmt/62cbn           903       ti        -           on
 /dev/rmt/63cbn           904       ti        -           on
:wq
[samsharefs-client2]root@solaris:~#
```

12. 特定のアーカイブセットのコピーで分散テープ入出力を有効または無効にする必要がある場合は、サーバーにログインし、テキストエディタで */etc/opt/SUNWsamfs/archiver.cmd* ファイルを開き、*-distio* パラメータをコピーディレクティブに追加します。分散入出力を有効にするには *-distio on*、無効にするには *-distio off* を設定します。ファイルを保存します。

    この例では、サーバー *samsharefs-mds* にログインし、*vi* エディタを使用して、コピー *1* の分散入出力を *off* にします。

```
[samsharefs-mds]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
# archiver.cmd
```

```
...
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 500M -startcount 500000 -distio off
allfiles.2 -startage 24h -startsize 20G  -startcount 500000 -reserve set
:wq
[samsharefs-mds]root@solaris:~#
```

13. *sam-fsd* コマンドを実行して、構成ファイルにエラーがないかどうかを確認します。見つかったエラーを修正します。

    sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、サーバー *sharefs-mds* 上でコマンドを実行します。

```
[sharefs-mds]root@solaris:~# sam-fsd
```

14. 変更済みの構成ファイルを読み取り、それ自体を適宜再構成するように、Oracle HSM サービスに指示します。報告されたエラーをすべて修正して、必要に応じて繰り返します。

```
[sharefs-mds]root@solaris:~# samd config
```

15. 分散入出力が正常にアクティブになったことを確認するには、コマンド *samcmd g* を使用します。クライアントの出力に *DATAMOVER* フラグが表示された場合は、分散入出力が正常にアクティブ化されています。

    この例では、フラグが表示されています。

```
[samsharefs-mds]root@solaris:~# samcmd g
Shared clients samcmd 6.0.dist_tapeio 11:09:13 Feb 20 2014
samcmd on samsharefs-mds
samsharefs is shared, server is samsharefs-mds, 2 clients 3 max
ord hostname             seqno nomsgs status   config  conf1  flags
  1 samsharefs-mds          14      0   8091  808540d   4051      0 MNT SVR

    config   :   CDEVID      ARCHIVE_SCAN    GFSID   OLD_ARCHIVE_FMT
    "        :   SYNC_META   TRACE   SAM_ENABLED     SHARED_MO
    config1  :   NFSV4_ACL   MD_DEVICES      SMALL_DAUS      SHARED_FS
```

```
    flags    :
    status   :   MOUNTED     SERVER  SAM     DATAMOVER
    last_msg :  Wed Jul  2 10:13:50 2014

  2 samsharefs-client1     127      0   a0a1  808540d   4041      0 MNT CLI

    config   :   CDEVID      ARCHIVE_SCAN    GFSID   OLD_ARCHIVE_FMT
    "        :   SYNC_META   TRACE   SAM_ENABLED     SHARED_MO
    config1  :   NFSV4_ACL   MD_DEVICES      SHARED_FS
    flags    :
    status   :   MOUNTED     CLIENT  SAM     SRVR_BYTEREV
    "        :   DATAMOVER
...
```

16. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
17. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

7.2. NFS と SMB/CIFS を使用した複数のホストからファイルシステムへのアクセス

Oracle HSM ソフトウェアによる複数のホストファイルシステムアクセスのネイティブサポートの代わりに、またはそのサポートに加えて、Network File System (NFS) または Server Message Block (SMB)/Common Internet File System (CIFS) を使用すると、複数のホストが Oracle HSM ファイルシステムにアクセスできます (「Oracle HSM ソフトウェアを使用した複数のホストからのファイルシステムへのアクセス」を参照)。次のセクションでは、基本的な構成手順の概要を示します。

- NFS を使用した Oracle HSM ファイルシステムの共有
- SMB/CIFS を使用した Oracle HSM ファイルシステムの共有

7.2.1. NFS を使用した Oracle HSM ファイルシステムの共有

次のタスクを実行します。

- NFS 4 を使用して Oracle HSM 共有ファイルシステムを共有する前の委任の無効化
- 必要に応じて、WORM ファイルおよびディレクトリを共有する NFS サーバーおよびクライアントの構成。
- Oracle HSM ホストでの NFS サーバーの構成

- NFS 共有としての Oracle HSM ファイルシステムの共有
- NFS クライアントでの NFS で共有された Oracle HSM ファイルシステムのマウント

7.2.1.1. NFS 4 を使用して Oracle HSM 共有ファイルシステムを共有する前の委任の無効化

NFS を使用して Oracle HSM 共有ファイルシステムを共有する場合は、Oracle HSM software が NFS からの干渉なしでアクセスを制御していることを確認する必要があります。一般に、NFS サーバーがそのクライアントの代わりにファイルにアクセスするときは、Oracle HSM 共有ファイルシステムのクライアントとしてアクセスするため、これは問題ではありません。ただし、NFS バージョン 4 のサーバーが読み取り/書き込みアクセス権への制御をクライアントに *委任* するように構成されている場合は、問題が発生する可能性があります。委任の長所は、サーバーが競合の発生を防止する場合にしか介入する必要がない点にあります。サーバーのワークロードが NFS クライアント間で部分的に分散され、ネットワークトラフィックが削減されます。ただし、委任では、独自の共有ファイルシステムクライアントからのアクセスも制御する Oracle HSM サーバーとは別にアクセス権 (特に書き込みアクセス権) が付与されます。競合の発生およびファイル破損の可能性を回避するには、委任を無効にする必要があります。次のように進めます。

1. NFS 共有として構成する Oracle HSM ファイルシステムのホストにログインします。*root* としてログインします。

   ファイルシステムが Oracle HSM 共有ファイルシステムである場合は、そのファイルシステムのメタデータサーバーにログインします。次の例では、サーバー名は *qfsnfs* です。

```
[qfsnfs]root@solaris:~#
```

2. NFS バージョン 4 を使用していて、NFS サーバーで Solaris 11.1 以降が動作している場合は、Service Management Facility (SMF) の *sharectl set -p* コマンドを使用して、NFS *server_delegation* プロパティーを *off* にします。

```
[qfsnfs]root@solaris:~# sharectl set -p server_delegation=off
```

3. NFS バージョン 4 を使用していて、NFS サーバーで Solaris 11.0 以前が動作している場合は、テキストエディタで */etc/default/nfs* ファイルを開き、*NFS_SERVER_DELEGATION* パラメータを *off* に設定して、委任を無効にします。ファイルを保存して、エディタを閉じます。

   この例では、*vi* エディタを使用します。

```
[qfsnfs]root@solaris:~# vi /etc/default/nfs
# ident "@(#)nfs        1.10    04/09/01 SMI"
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
...
NFS_SERVER_DELEGATION=off
:wq
[qfsnfs]root@solaris:~#
```

4. 共有する予定の Oracle HSM ファイルシステムで Write-Once Read-Many (WORM) 機能がサポートされている場合は、ここで WORM ファイルおよびディレクトリを共有する NFS サーバーおよびクライアントの構成を実行します。
5. それ以外の場合は、Oracle HSM ホストでの NFS サーバーの構成を実行します。

7.2.1.2. WORM ファイルおよびディレクトリを共有する NFS サーバーおよびクライアントの構成

1. NFS を使用して共有する Oracle HSM ファイルシステムのホストにログインします。*root* としてログインします。

   ファイルシステムが Oracle HSM 共有ファイルシステムである場合は、そのファイルシステムのメタデータサーバーにログインします。次の例では、サーバー名は *qfsnfs*、クライアント名は *nfsclient1* です。

```
[qfsnfs]root@solaris:~#
```

2. 共有する予定の Oracle HSM ファイルシステムが WORM 機能を使用していて、Oracle Solaris 10 以降で動作しているサーバー上でホストされている場合は、NFS サーバー上およびすべてのクライアント上で NFS バージョン 4 が有効になっていることを確認します。

   この例では、サーバー *qfsnfs* およびクライアント *nfsclient1* を確認します。どちらの場合でも、*uname -r* コマンドを使用して、最初に Solaris のバージョンレベルを確認します。次に、*modinfo* コマンドの出力を *grep* および正規表現にパイプして、NFS のバージョン情報を検索します。

```
[qfsnfs]root@solaris:~# uname -r
5.11
```

```
[qfsnfs]root@solaris:~# modinfo | grep -i "nfs.* version 4"
258 7a600000  86cd0  28   1  nfs (network filesystem version 4)
[qfsnfs]root@solaris:~# ssh root@nfsclient1
Pasword: ...
[nfsclient1]root@solaris:~# uname -r
5.11
[nfsclient1]root@solaris:~# modinfo | grep -i "nfs.* version 4"
278 ffffffff8cba000  9df68  27   1  nfs (network filesystem version 4)
[nfsclient1]root@solaris:~# exit
[qfsnfs]root@solaris:~#
```

3. Oracle Solaris 10 以降で動作しているサーバー上で NFS バージョン 4 が有効になっていない場合は、サーバーおよび各クライアントに *root* としてログインします。次に、*sharectl set* コマンドを使用して、NFS 4 を有効にします。

```
[qfsnfs]root@solaris:~# sharectl set -p server_versmax=4 nfs
[qfsnfs]root@solaris:~# ssh root@nfsclient1
Password ...
[nfsclient1]root@solaris:~# sharectl set -p server_versmax=4 nfs
[nfsclient1]root@solaris:~# exit
[qfsnfs]root@solaris:~#
```

4. 次に、Oracle HSM ホストでの NFS サーバーの構成を実行します

7.2.1.3. Oracle HSM ホストでの NFS サーバーの構成

クライアントがネットワークファイルシステム (NFS) を使用して正常に Oracle HSM ファイルシステムをマウントできるようにするには、ホスト上でファイルシステムが正常にマウントされるまで、Oracle HSM ファイルシステムの共有が試行されないように NFS サーバーを構成する必要があります。Oracle Solaris 10 以降のオペレーティングシステムバージョンでは、Service Management Facility (SMF) がブート時にファイルシステムのマウントを管理します。次の手順を使用して NFS を構成しない場合は、QFS マウントと NFS 共有のどちらか一方は成功しますが、他方には失敗します。

1. NFS 共有として構成する Oracle HSM ファイルシステムのホストにログインします。*root* としてログインします。

ファイルシステムが Oracle HSM 共有ファイルシステムである場合は、そのファイルシステムのメタデータサーバーにログインします。次の例では、サーバー名は *qfsnfs* です。

```
[qfsnfs]root@solaris:~#
```

2. *svccfg export /network/nfs/server* コマンドの出力をリダイレクトして、既存の NFS 構成を XML マニフェストファイルにエクスポートします。

   この例では、エクスポートされた構成をマニフェストファイル */var/tmp/server.xml* に転送します。

```
[qfsnfs]root@solaris:~# svccfg export /network/nfs/server > /var/tmp/server.xml
[qfsnfs]root@solaris:~#
```

3. テキストエディタでマニフェストファイルを開き、*filesystem-local* 依存関係を探します。

   この例では、*vi* エディタでファイルを開きます。*filesystem-local* 依存関係のエントリは、*nfs-server_multi-user-server* 依存関係のエントリの直前に一覧表示されます。

```
[qfsnfs]root@solaris:~# vi /var/tmp/server.xml
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='export'>
  <service name='network/nfs/server' type='service' version='0'>
    ...
    <dependency name='filesystem-local' grouping='require_all' restart_on='error' type='service'>
      <service_fmri value='svc:/system/filesystem/local'/>
    </dependency>
    <dependent name='nfs-server_multi-user-server' restart_on='none'
        grouping='optional_all'>
      <service_fmri value='svc:/milestone/multi-user-server'/>
    </dependent>
    ...
```

4. *filesystem-local* 依存関係の直後に、QFS 共有ファイルシステムをマウントする *qfs* 依存関係を追加します。次に、ファイルを保存して、エディタを終了します。

   これにより、サーバーが NFS 経由で共有を試行する前に、Oracle HSM 共有ファイルシステムがマウントされます。

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='export'>
  <service name='network/nfs/server' type='service' version='0'>
    ...
    <dependency name='filesystem-local' grouping='require_all' restart_on='error' type='service'>
      <service_fmri value='svc:/system/filesystem/local'/>
    </dependency>
    <dependency name='qfs' grouping='require_all' restart_on='error' type='service'>
      <service_fmri value='svc:/network/qfs/shared-mount:default'/>
    </dependency>
    <dependent name='nfs-server_multi-user-server' restart_on='none'
        grouping='optional_all'>
      <service_fmri value='svc:/milestone/multi-user-server'/>
    </dependent>
:wq
[qfsnfs]root@solaris:~#
```

5. *svccfg validate* コマンドを使用して、マニフェストファイルを検証します。

```
[qfsnfs]root@solaris:~# svccfg validate /var/tmp/server.xml
```

6. *svccfg validate* コマンドでエラーがレポートされた場合は、そのエラーを修正してから、ファイルを再検証します。

   この例では、*svccfg validate* コマンドで XML 解析エラーが返されます。ファイルを保存するときに、誤って末尾のタグ *</dependency>* を付け忘れました。そのため、*vi* エディタでファイルを再度開いて、問題を修正します。

```
[qfsnfs]root@solaris:~# svccfg validate /var/tmp/server.xml
/var/tmp/server.xml:75: parser error : Opening and ending tag mismatch: dependency line 29 and
 service
```

```
  </service>
            ^
/var/tmp/server.xml:76: parser error : expected '>'
</service_bundle>
          ^
/var/tmp/server.xml:77: parser error : Premature end of data in tag service_bundle line 3
^
svccfg: couldn't parse document
[qfsnfs]root@solaris:~# vi /var/tmp/server.xml
...
:wq
[qfsnfs]root@solaris:~#
```

7. *svccfg validate* コマンドがエラーなしで完了したら、*svcadm disable nfs/server* コマンドを使用して NFS を無効にします。

   この例では、*svccfg validate* コマンドで出力が返されなかったため、ファイルは有効であり、NFS を無効にできます。

```
[qfsnfs]root@solaris:~# svccfg validate /var/tmp/server.xml
[qfsnfs]root@solaris:~# svcadm disable nfs/server
```

8. *svccfg delete nfs/server* コマンドを使用して、既存の NFS サーバー構成を削除します。

```
[qfsnfs]root@solaris:~# svccfg delete nfs/server
```

9. *svccfg import* コマンドを使用して、マニフェストファイルを Service Management Facility (SMF) にインポートします。

```
[qfsnfs]root@solaris:~# svccfg import /var/tmp/server.xml
```

10. *svcadm enable nfs/server* コマンドを使用して、NFS を再度有効にします。

    NFS は、更新済みの構成を使用するように構成されます。

```
[qfsnfs]root@solaris:~# svcadm enable nfs/server
```

11. *qfs* 依存関係が適用されたことを確認します。コマンド *svcs -d svc:/network/nfs/server:default* で */network/qfs/shared-mount:default* サービスが表示されることを確認します。

```
[qfsnfs]root@solaris:~# svcs -d svc:/network/nfs/server:default
STATE          STIME    FMRI
...
online         Nov_01   svc:/network/qfs/shared-mount:default
...
```

12. 次に、NFS 共有としての Oracle HSM ファイルシステムの共有を実行します。

7.2.1.4. NFS 共有としての Oracle HSM ファイルシステムの共有

使用中の Oracle Solaris オペレーティングシステムバージョンに対応した管理ドキュメントに記載されている手順を使用して、Oracle HSM ファイルシステムを共有します。次の手順では、Solaris 11.1 での手順について簡単に説明します。

1. NFS を使用して共有する Oracle HSM ファイルシステムのホストにログインします。*root* としてログインします。

   ファイルシステムが Oracle HSM 共有ファイルシステムである場合は、そのファイルシステムのメタデータサーバーにログインします。次の例では、サーバー名は *qfsnfs* です。

```
[qfsnfs]root@solaris:~#
```

2. コマンド行 *share -F nfs -o sharing-options sharepath* を入力します。ここで *-F* スイッチは *nfs* 共有プロトコルを指定し、*sharepath* は共有リソースへのパスです。オプションの *-o* パラメータを使用する場合、*sharing-options* には次のいずれかを含めることができます。
   - *rw* は、読み取りおよび書き込み権限のある *sharepath* をすべてのクライアントが使用できるようにします。
   - *ro* は、読み取り専用権限のある *sharepath* をすべてのクライアントが使用できるようにします。
   - *rw=clients* は、読み取りおよび書き込み権限のある *sharepath* を *clients* (共有へのアクセス権を持つ 1 つ以上のクライアントのコロン区切りリスト) が使用できるようにします。

- *ro=clients* は、読み取り専用権限のある *sharepath* を *clients* (共有へのアクセス権を持つ 1 つ以上のクライアントのコロン区切りリスト) が使用できるようにします。

この例では、*/qfsms* ファイルシステムを読み取り/書き込みでクライアント *nfsclient1* および *nfsclient2* と共有して、読み取り専用で *nfsclient3* と共有します (次のコマンドは 1 行で入力します。改行はバックスラッシュでエスケープされます)。

```
[qfsnfs]root@solaris:~# share -F nfs -o rw=nfsclient1:nfsclient2 /
 ro=nfsclient3 /qfsms
```

このコマンドを入力すると、自動的に NFS サーバーデーモン *nfsd* が再起動されます。追加のオプションおよび詳細は、*share_nfs* のマニュアルページを参照してください。

3. コマンド行 *share -F nfs* を使用して、共有パラメータを確認します。

   この例では、共有が適切に構成されたことがコマンドの出力に表示されます。

```
[qfsnfs]root@solaris:~# share -F nfs
/qfsms   sec=sys,rw=nfsclient1:nfsclient2,ro=nfsclient3
[qfsnfs]root@solaris:~#
```

4. 次に、NFS クライアントでの NFS で共有された Oracle HSM ファイルシステムのマウントを実行します。

7.2.1.5. NFS クライアントでの NFS で共有された Oracle HSM ファイルシステムのマウント

クライアントシステム上の適切なマウントポイントに、NFS サーバーのファイルシステムをマウントします。クライアントごとに、次の手順を実行します。

1. クライアントに *root* としてログインします。

   この例では、NFS クライアントの名前は *nfsclient1* です。

```
[nfsclient1]root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
[nfsclient1]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
[nfsclient1]root@solaris:~#
```

3. テキストエディタで */etc/vfstab* ファイルを開きます。

   次の例では、*vi* エディタを使用します。

```
[nfsclient1]root@solaris:~# vi /etc/vfstab
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
```

4. */etc/vfstab* ファイルの 1 列目に、NFS サーバーの名前と共有するファイルシステムのマウントポイントをコロンで区切って指定して、マウントするファイルデバイスの名前を指定します。

   この例では、NFS サーバーの名前は *qfsnfs*、共有ファイルシステムの名前は *qfsms*、サーバー上のマウントポイントは */qfsms* です。

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms
```

5. ローカルシステムがリモートのファイルシステムの整合性チェックを試行しないように、*/etc/vfstab* ファイルの 2 列目に、ハイフン (-) を入力します。

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -
```

6. */etc/vfstab* ファイルの 3 列目に、リモートのファイルシステムをマウントするローカルのマウントポイントを入力します。

   この例では、マウントポイントはディレクトリ */qfsnfs* です。

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs
```

7. */etc/vfstab* ファイルの 4 列目に、ファイルシステムタイプ *nfs* を入力します。

```
#File         Device                           Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs
```

   クライアントは NFS ファイルシステムとしてリモートの QFS ファイルシステムをマウントするため、*nfs* ファイルシステムタイプを使用します。

8. ローカルシステムではリモートのファイルシステムの整合性チェックが行われないため、*/etc/vfstab* ファイルの 5 列目に、ハイフン (-) を入力します。

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs    -
```

9. */etc/vfstab* ファイルの 6 列目に、ブート時にリモートのファイルシステムをマウントする場合は *yes*、要求に応じて手動でマウントする場合は *no* を入力します。

   この例では、*yes* を入力します。

```
#File         Device                        Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs    -     yes
```

10. */etc/vfstab* ファイルの最後の列に、*hard* および *intr* NFS マウントオプションを入力して、無制限で中断なしの再試行を強制するか、または *retrans* を *120* 以上に設定して *timeo* を *3000* 1/10 秒に設定し、*soft*、*retrans*、および *timeo* マウントオプションを入力することによって、特定の再試行回数を設定します。

    hard 再試行オプションを設定するか、または *soft* オプションに十分に長いタイムアウト値と十分な再試行回数を指定すると、すぐにマウントできないリムーバブルボリューム上に要求されたファイルが存在するときにも、NFS 要求が失敗しなくなります。これらのマウントオプションの詳細は、Solaris *mount_nfs* のマニュアルページを参照してください。

    この例では、*soft* マウントオプションを入力します。

```
#File         Device                        Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs    -     yes    soft,retrans=120,timeo=3000
```

11. NFS 2 を使用している場合は、*rsize* マウントパラメータを *32768* に設定します。

    その他の NFS バージョンを使用している場合は、デフォルト値を受け入れます。

rsize マウントパラメータは、読み取りバッファーのサイズを *32768* バイトに設定します (デフォルトは *8192* バイト)。この例では、NFS 2 での構成方法を示します。

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs2:/qfs2 -       /qfsnfs2  nfs    -     yes    ...,rsize=32768
```

12. NFS 2 を使用している場合は、*wsize* マウントパラメータを *32768* に設定します。

    その他の NFS バージョンを使用している場合は、デフォルト値を受け入れます。

    wsize マウントパラメータは、書き込みバッファーのサイズを指定したバイト数に設定します (デフォルトは *8192* バイト)。この例では、NFS 2 での構成方法を示します。

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs2:/qfs2 -       /qfsnfs2  nfs    -     yes    ...,wsize=32768
```

13. */etc/vfstab* ファイルを保存して、エディタを終了します。

```
#File         Device                         Mount
#Device       to      Mount     System fsck  at     Mount
#to Mount     fsck    Point     Type   Pass  Boot   Options
#------------ ------  --------- ------ ----  -----  ----------------
/devices      -       /devices  devfs  -     no     -
...
qfsnfs:/qfsms -       /qfsnfs   nfs    -     yes    soft,retrans=120,timeo=3000
:wq
[nfsclient1]root@solaris:~#
```

14. 共有ファイルシステムのマウントポイントディレクトリを作成します。

    この例では、*/qfsnfs* という名前のディレクトリ上で共有ファイルシステムをマウントします。

```
[nfsclient1]root@solaris:~# mkdir /qfsnfs
[nfsclient1]root@solaris:~#
```

15. */etc/vfstab* ファイルで指定されたマウントポイントを作成し、そのマウントポイントに対するアクセス権を設定します。

    ユーザーはマウントポイントポイントディレクトリに移動し、マウントしたファイルシステム内のファイルにアクセスするための実行権 (*x*) を持っている必要があります。この例では、*/qfsnfs* マウントポイントディレクトリを作成し、アクセス権を *755* (*-rwxr-xr-x*) に設定します。

```
[nfsclient1]root@solaris:~# mkdir /qfsnfs
[nfsclient1]root@solaris:~# chmod 755 /qfsnfs
[nfsclient1]root@solaris:~#
```

16. 共有ファイルシステムをマウントします。

```
[nfsclient1]root@solaris:~# mount /qfsnfs
[nfsclient1]root@solaris:~#
```

17. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
18. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

7.2.2. SMB/CIFS を使用した Oracle HSM ファイルシステムの共有

SMB を使用すると、Oracle HSM が Microsoft Windows ホストにアクセスできるようになり、大文字と小文字の区別のなし、DOS 属性のサポート、および NFSv4 のアクセス制御リスト (ACL) のサポートなどの相互運用性機能が提供されます。Oracle Solaris OS は、サーバーメッセージブロック (SMB) プロトコルのサーバーおよびクライアント実装を提供し、これに

は、NT LM 0.12 や共通インタフェースファイルシステム (CIFS) などの多数の SMB ダイアレクトのサポートが含まれます。

Oracle HSM では、Windows Security Identifier (SID) がサポートされます。Windows アイデンティティーは、*idmap* サービスを使用して明示的に定義したり、Active Directory サービスで提供したりする必要がなくなりました。

Oracle HSM ファイルシステムで SMB サービスを構成するには、次のタスクを実行します。

- Oracle Solaris SMB 構成および管理ドキュメントの確認.
- SMB サーバー用の Windows アイデンティティーの明示的なマップ (オプション).
- SMB/CIFS を使用して共有する Oracle HSM ファイルシステムの構成.
- Windows Active Directory ドメインまたはワークグループ用の SMB サーバーの構成.
- SMB/CIFS 共有としての Oracle HSM ファイルシステムの共有.

7.2.2.1. Oracle Solaris SMB 構成および管理ドキュメントの確認

次のセクションでは、Oracle HSM ファイルシステムに適用される SMB 構成プロセスの一部について概要を示します。これらは包括的なものではなく、考えられるシナリオをすべて網羅しているわけではありません。そのため、Oracle Solaris SMB サーバーの構成、既存の Windows 環境へのサーバーの統合、および Solaris システムでの SMB 共有のマウントについては、完全な手順を確認してください。完全な手順は、*Oracle Solaris 11.1 Information Library* で *Oracle Solaris 11.1 での SMB と Windows の相互運用性*に関するドキュメントで見つかります。

7.2.2.2. SMB サーバー用の Windows アイデンティティーの明示的なマップ (オプション)

Oracle HSM では、Windows Security Identifier (SID) が完全にサポートされるようになりましたが、一部の状況では、UNIX アイデンティティーと SID 間の関係を明示的に定義することが引き続き役立つ場合があります。たとえば、ユーザーが UNIX と Windows の両方のアイデンティティーを持っている異種環境では、*idmap* サービスまたは Active Directory サービスを使用して、明示的なマッピングを作成する場合があります。SMB と Windows の完全な相互運用性情報については、使用している Oracle Solaris バージョンの製品ドキュメントを参照してください。

7.2.2.3. SMB/CIFS を使用して共有する Oracle HSM ファイルシステムの構成

SMB/CIFS を使用して共有する Oracle HSM ファイルシステムでは、ネットワークファイルシステム (NFS) Version 4 で採用され、Solaris 11 で導入されたアクセス制御リスト (ACL) の実装を使用する必要があります。旧バージョンの Solaris および NFS では、Windows ACL 実装と互換性のない POSIX ドラフト仕様に基づいた ACL が使用されていました。

Solaris 11 では、Oracle HSM で作成する新しいファイルシステムは、デフォルトで NFS バージョン 4 ACL を使用します。ただし、SMB/CIFS クライアントを使用して既存の Oracle HSM ファイルシステムを共有する必要がある場合は、適切な手順を使用して既存の POSIX スタイルの ACL を変換する必要があります。

- POSIX スタイルの ACL を使用する Oracle HSM 非共有ファイルシステムの変換
- POSIX スタイルの ACL を使用する Oracle HSM 共有ファイルシステムの変換

7.2.2.3.1. POSIX スタイルの ACL を使用する Oracle HSM 非共有ファイルシステムの変換

次のように進めます。

1. ホストに *root* としてログインします。

   この例では、ホスト *qfs-host* にログインします。

```
[qfs-host]root@solaris:~#
```

2. ホストで Oracle Solaris 11.1 以上が実行されていることを確認します。コマンド *uname -r* を使用します。

```
[qfs-host]root@solaris:~# uname -r
5.11
[qfs-host]root@solaris:~#
```

3. コマンド *umount mount-point* を使用して、ファイルシステムをアンマウントします。ここで *mount-point* は、Oracle HSM ファイルシステムのマウントポイントです。

   詳細については、*umount_samfs* のマニュアルページを参照してください。次の例では、サーバー名は *qfs-host*、ファイルシステムは */qfsms* です。

```
[qfs-host]root@solaris:~# umount /qfsms
```

4. *samfsck -F -A file-system* コマンドを使用して、ファイルシステムを変換します。ここで *-F* オプションを指定すると、ファイルシステムがチェックおよび修復され、*-A* オプションを指定すると ACL が変換されます。*file-system* は、変換する必要があるファイルシステムの名前です。

   -A オプションが指定されているときは、*-F* オプションが必須です。*samfsck -F -A* コマンドでエラーが返された場合は、プロセスが異常終了し、ACL は変換されません (これらのオプションの詳細は、*samfsck* のマニュアルページを参照)。

```
[qfs-host]root@solaris:~# samfsck -F -A /qfsms
```

5. エラーが返され、ACL が変換されない場合は、*samfsck -F -a file-system* コマンドを使用して、強制的に ACL を変換します。

   -a オプションを指定すると、強制的に変換されます。*-a* オプションが指定されているときは、*-F* オプションが必須です (これらのオプションの詳細は、*samfsck* のマニュアルページを参照)。

```
[qfs-host]root@solaris:~# samfsck -F -a /qfsms
```

6. 次に、Windows Active Directory ドメインまたはワークグループ用の SMB サーバーの構成を実行します。

7.2.2.3.2. POSIX スタイルの ACL を使用する Oracle HSM 共有ファイルシステムの変換

1. ファイルシステムのメタデータサーバーに *root* としてログインします。

   この例では、メタデータサーバー *sharedqfs-mds* にログインします。

```
[sharedqfs-mds]root@solaris:~#
```

2. メタデータサーバーで Oracle Solaris 11.1 以上が実行されていることを確認します。コマンド *uname -r* を使用します。

```
[sharedqfs-mds]root@solaris:~# uname -r
```

```
5.11
[sharedqfs-mds]root@solaris:~#
```

3. 各 Oracle HSM クライアントに *root* としてログインして、各クライアントで Oracle Solaris 11.1 以上が実行されていることを確認します。

   この例では、端末ウィンドウを開き、*ssh* を使用してクライアントホスト *sharedqfs-client1* および *sharedqfs-client2* にリモートログインして、ログインバナーから Solaris のバージョンを取得します。

```
[sharedqfs-mds]root@solaris:~# ssh root@sharedqfs-client1
Password:
Oracle Corporation      SunOS 5.11      11.1    September 2013
[sharedqfs-client1]root@solaris:~#


[sharedqfs-mds]root@solaris:~# ssh root@sharedqfs-client2
Password:
Oracle Corporation      SunOS 5.11      11.1    September 2013
[sharedqfs-client2]root@solaris:~#
```

4. コマンド *umount mount-point* を使用して、各 Oracle HSM クライアントから Oracle HSM 共有ファイルシステムをアンマウントします。ここで *mount-point* は、Oracle HSM ファイルシステムのマウントポイントです。

   詳細については、*umount_samfs* のマニュアルページを参照してください。この例では、2 つのクライアント (*sharedqfs-client1* と *sharedqfs-client2*) から */sharedqfs1* をアンマウントします。

```
Oracle Corporation      SunOS 5.11      11.1    September 2013
[sharedqfs-client1]root@solaris:~# umount /sharedqfs
[sharedqfs-client1]root@solaris:~#


Oracle Corporation      SunOS 5.11      11.1    September 2013
[sharedqfs-client2]root@solaris:~# umount /sharedqfs
[sharedqfs-client1]root@solaris:~#
```

5. コマンド *umount -o await_clients=interval mount-point* を使用して、メタデータサーバーから Oracle HSM 共有ファイルシステムをアンマウントします。ここで *mount-*

point は、Oracle HSM ファイルシステムのマウントポイントで、interval は *-o await_clients* オプションの遅延実行で指定された遅延 (秒) です。

Oracle HSM 共有ファイルシステムのメタデータサーバー上で *-o await_clients* オプションを付けて *umount* コマンドを発行すると、クライアントが共有をアンマウントする時間を持てるように、*umount* は指定された秒数間待機します。非共有ファイルシステムをアンマウントする場合や、Oracle HSM クライアント上でコマンドを発行する場合は影響がありません。詳細については、*umount_samfs* のマニュアルページを参照してください。

この例では、クライアントのアンマウントを *60* 秒間許可し、メタデータサーバー *sharedqfs-mds* から */sharedqfs* ファイルシステムをアンマウントします。

```
[sharedqfs-mds]root@solaris:~# umount -o await_clients=60 /sharedqfs
```

6. ファイルシステムを POSIX スタイルの ACL から NFS バージョン 4 の ACL に変換します。メタデータサーバー上で、コマンド *samfsck -F -A file-system* を使用します。ここで *-F* オプションを指定するとファイルシステムがチェックおよび修復され、*-A* オプションを指定すると ACL が変換されます。*file-system* は、変換する必要があるファイルシステムの名前です。

   -A オプションが指定されているときは、*-F* オプションが必須です。*samfsck -F -A file-system* コマンドでエラーが返された場合は、プロセスが異常終了し、ACL は変換されません (これらのオプションの詳細は、*samfsck* のマニュアルページを参照)。この例では、*/sharedqfs* という名前の Oracle HSM ファイルシステムを変換します。

```
[sharedqfs-mds]root@solaris:~# samfsck -F -A /sharedqfs
```

7. エラーが返され、ACL が変換されない場合は、強制的に ACL を変換します。メタデータサーバー上で、*samfsck -F -a file-system* コマンドを使用します。

   -a オプションを指定すると、強制的に変換されます。*-a* オプションが指定されているときは、*-F* オプションが必須です (これらのオプションの詳細は、*samfsck* のマニュアルページを参照)。この例では、*/qfsma* という名前の Oracle HSM ファイルシステムを強制的に変換します。

```
[sharedqfs-mds]root@solaris:~# samfsck -F -a /sharedqfs
```

8. 次に、Windows Active Directory ドメインまたはワークグループ用の SMB サーバーの構成を実行します。

7.2.2.4. Windows Active Directory ドメインまたはワークグループ用の SMB サーバーの構成

Oracle Solaris SMB サービスは、ドメインとワークグループという、相互に排他的な 2 つのモードのいずれかで動作できます。環境と認証のニーズに基づいて、いずれか一方を選択します。

- Active Directory ドメインユーザーに Solaris SMB サービスへのアクセス権を付与する必要がある場合は、ドメインモードでの SMB サーバーの構成を実行します。
- ローカルの Solaris ユーザーに SMB サービスへのアクセス権を付与する必要があるが、Active Directory ドメインを持っていないか、または Active Directory ドメインユーザーにサービスへのアクセス権を付与する必要がない場合は、ワークグループモードでの SMB サーバーの構成を実行します。

7.2.2.4.1. ドメインモードでの SMB サーバーの構成

1. Windows Active Directory 管理者に連絡して、次の情報を取得します。
   - Active Directory ドメインに参加する際に使用する必要がある認証済みの Active Directory ユーザーアカウントの名前
   - アカウント用のデフォルトの *Computers* コンテナの代わりに使用する必要がある組織単位 (存在する場合)
   - Oracle HSM ファイルシステムが共有されるドメインの完全修飾 LDAP/DNS ドメイン名。
2. SMB/CIFS 共有として構成する Oracle HSM ファイルシステムのホストにログインします。*root* としてログインします。

   ファイルシステムが Oracle HSM 共有ファイルシステムである場合は、そのファイルシステムのメタデータサーバーにログインします。次の例では、サーバー名は *qfssmb* です。

```
[qfssmb]root@solaris:~#
```

3. 単一の Oracle Solaris システム上では、オープンソースの Samba と SMB サーバーを同時に使用できません。したがって、Samba サービスが実行されているかどうかを確認します。サービスステータスコマンド *svcs* の出力を *grep* および正規表現 *samba* にパイプします。

   この例では、*svcs* コマンドの出力に正規表現の一致が含まれているため、SMB サービスは実行中です。

```
[qfssmb]root@solaris:~# svcs | grep samba
legacy_run     Nov_03   lrc:/etc/rc3_d/S90samba
```

4. Samba サービス (*svc:/network/samba*) が実行中の場合は、Windows Internet Naming Service/WINS (*svc:/network/wins*) (実行中の場合) とともに無効にします。コマンド *svcadm disable* を使用します。

```
[qfssmb]root@solaris:~# svcadm disable svc:/network/samba
[qfssmb]root@solaris:~# svcadm disable svc:/network/wins
```

5. ここで、*svcadm enable -r smb/server* コマンドを使用して、SMB サーバーおよびこれが依存するサービスを起動します。

```
[qfssmb]root@solaris:~# svcadm enable -r smb/server
```

6. Oracle HSM ホスト上のシステムクロックが Microsoft Windows ドメインコントローラのシステムクロックの 5 分以内であることを確認します。
   - Windows ドメインコントローラで Network Time Protocol (NTP) サーバーが使用されている場合は、同じサーバーを使用するように Oracle HSM ホストを構成します。Oracle HSM ホスト上に */etc/inet/ntpclient.conf* ファイルを作成し、*svcadm enable ntp* コマンドを使用して *ntpd* デーモンを起動します (詳細は、*ntpd* のマニュアルページおよび Oracle Solaris 管理ドキュメントを参照)。
   - それ以外の場合は、*ntpdate domain-controller-name* コマンドを実行して、Oracle HSM ホストとドメインコントローラを同期するか (詳細は、*ntpdate* のマニュアルページを参照)、または Oracle HSM ホスト上のシステムクロックをドメインコントローラのシステムクロックで表示される時間に手動で設定します。
7. コマンド *smbadm join -u username -o organizational-unit domain-name* を使用して、Windows ドメインに参加します。ここで *username* は、Active Directory 管理者によって指定されたユーザーアカウントの名前、オプションの *organizational-unit* は指定されたアカウントコンテナ (存在する場合)、*domain-name* は指定された完全修飾 LDAP または DNS ドメイン名です。

   この例では、ユーザーアカウントを使用して、Windows ドメイン *this.example.com* に参加します。

```
[qfssmb]root@solaris:~# smbadm join -u admin -o smbsharing this.example.com
```

8. 次に、SMB/CIFS 共有としての Oracle HSM ファイルシステムの共有を実行します。

7.2.2.4.2. ワークグループモードでの SMB サーバーの構成

1. Contact the Windows ネットワーク管理者に連絡して、Oracle HSM ファイルシステムのホストが参加する必要のある Windows ワークグループの名前を取得します。

   デフォルトのワークグループの名前は *WORKGROUP* です。
2. Oracle HSM ファイルシステムのホストにログインします。*root* としてログインします。

   ファイルシステムが Oracle HSM 共有ファイルシステムである場合は、そのファイルシステムのメタデータサーバーにログインします。次の例では、サーバー名は *qfssmb* です。

```
[qfssmb]root@solaris:~#
```

3. 単一の Oracle Solaris システム上では、オープンソースの Samba と SMB サーバーを同時に使用できません。したがって、Samba サービスが実行されているかどうかを確認します。*svcs* サービスステータスコマンドの出力を *grep* および正規表現 *samba* にパイプします。

   この例では、*svcs* コマンドの出力に正規表現の一致が含まれているため、SMB サービスは実行中です。

```
[qfssmb]root@solaris:~# svcs | grep samba
legacy_run     Nov_03   lrc:/etc/rc3_d/S90samba
```

4. Samba サービス (*svc:/network/samba*) が実行中の場合は、Windows Internet Naming Service/WINS (*svc:/network/wins*) サービス (実行中の場合) とともに無効にします。コマンド *svcadm disable* を使用します。

   単一の Oracle Solaris システム上では、Samba と SMB サーバーを同時に使用できません。

```
[qfssmb]root@solaris:~# svcadm disable svc:/network/samba
[qfssmb]root@solaris:~# svcadm disable svc:/network/wins
```

5. ここで、コマンド *svcadm enable -r smb/server* を使用して、SMB サーバーおよびこれが依存するサービスを起動します。

```
[qfssmb]root@solaris:~# svcadm enable -r smb/server
```

6. ワークグループに参加します。*-w* (ワークグループ) スイッチ、および Windows ネットワーク管理者によって指定されたワークグループの名前を指定して、コマンド *smbadm join* を使用します。

   この例では、指定されたワークグループの名前は *crossplatform* です。

```
[qfssmb]root@solaris:~# smbadm join -w crossplatform
```

7. SMB パスワードが暗号化されるように Oracle HSM ホストを構成します。テキストエディタで */etc/pam.d/other* ファイルを開き、コマンド行 *password required pam_smb_passwd.so.1 nowarn* を追加して、ファイルを保存します。

   この例では、*vi* エディタを使用します。

```
[qfssmb]root@solaris:~# vi /etc/pam.d/other
# Copyright (c) 2012, Oracle and/or its affiliates. All rights reserved.
#
# PAM configuration
#
# Default definitions for Authentication management
# Used when service name is not explicitly mentioned for authentication
#
auth definitive  pam_user_policy.so.1
...
password required pam_authtok_store.so.1
password required pam_smb_passwd.so.1 nowarn
:wq
[qfssmb]root@solaris:~#
```

   詳細は、*pam_smb_passwd* のマニュアルページを参照してください。

8. *pam_smb_passwd* モジュールがインストールされたら、SMB サーバーが Windows ワークグループにログインできるように、*passwd local-username* コマンドを使用して、ユーザー *local-username* の暗号化バージョンのパスワードを生成します。

SMB サーバーのユーザー認証では、Solaris オペレーティングシステムで使用されるものと同じ暗号化バージョンのパスワードを使用できません。この例では、ユーザー *smbsamqfs* の暗号化された SMB パスワードを生成します。

```
[qfssmb]root@solaris:~# passwd smbsamqfs
```

9. 次に、SMB/CIFS 共有としての Oracle HSM ファイルシステムの共有を実行します。

7.2.2.5. SMB/CIFS 共有としての Oracle HSM ファイルシステムの共有

使用中の Oracle Solaris オペレーティングシステムバージョンに対応した管理ドキュメントに記載されている手順を使用して、Oracle HSM ファイルシステムを共有します。次の手順では、Solaris 11.1 での手順について簡単に説明します。

1. SMB/CIFS 共有として構成する Oracle HSM ファイルシステムのホストにログインします。*root* としてログインします。

   ファイルシステムが Oracle HSM 共有ファイルシステムである場合は、そのファイルシステムのメタデータサーバーにログインします。次の例では、サーバー名は *qfssmb* です。

```
[qfssmb]root@solaris:~#
```

2. 共有を構成します。コマンド *share -F smb -o specific-options sharepath sharename* を使用します。ここで *-F* スイッチは *smb* 共有プロトコルを指定し、*sharepath* は共有リソースへのパス、*sharename* は共有で使用する名前です。オプションの *-o* パラメータの値 *sharing-options* には、次のいずれかを含めることができます。

   - *abe=[true|false]*

     共有のアクセスベースの列挙 (ABE) ポリシーが *true* になっている場合は、クライアントに返されるディレクトリリストから、要求するユーザーがアクセス権を持っていないディレクトリのエントリが削除されます。

   - *ad-container=cn=user,ou=organization,dc=domain-dns*

     Active Directory コンテナでは、共有アクセスは、LDAP (Lightweight Directory Access Protocol) 相対識別名 (RDN) 属性値 *cn* (ユーザーオブジェクトクラス)、*ou*

(組織単位オブジェクトクラス)、および *dc* (ドメイン DNS オブジェクトクラス) で指定されたドメインオブジェクトに制限されます。

SMB/CIFS での Active Directory コンテナの使用の詳細は、*Internet Engineering Task Force Request For Comment (RFC) 2253* および Microsoft Windows ディレクトリサービスのドキュメントを参照してください。

- *catia=[true|false]*

  CATIA 文字置き換えが *true* になっている場合は、Windows で無効な CATIA バージョン 4 ファイル名内の文字が有効な文字で置き換えられます。置き換えリストについては、*share_smb* のマニュアルページを参照してください。

- *csc=[manual|auto|vdo|disabled]*

  クライアント側キャッシュ (*csc*) のポリシーによって、オフラインで使用されるファイルのクライアント側キャッシュが制御されます。*manual* ポリシーを使用すると、クライアントはユーザーから要求があればファイルをキャッシュに入れることができますが、自動的なファイルごとの再統合は無効になります (これがデフォルトです)。*auto* ポリシーを使用すると、クライアントは自動的にファイルをキャッシュに入れることができ、ファイルごとの自動的な再統合が有効になります。*vdo* ポリシーを使用すると、クライアントは自動的にオフラインで使用されるファイルをキャッシュに入れることができ、ファイルごとの再統合が有効になり、オフラインでもクライアントがローカルキャッシュから動作できます。*disabled* ポリシーを使用すると、クライアント側キャッシュが許可されません。

- *dfsroot=[true|false]*

  Microsoft 分散ファイルシステム (DFS) では、ルート共有 (*dfsroot=true*) は、幅広く分散している共有フォルダのグループを、より簡単に管理できる単一の DFS ファイルシステムにまとめる共有です。詳細は、Microsoft Windows Server のドキュメントを参照してください。

- *guestok=[true|false]*

  guestok ポリシーが *true* になっている場合は、ローカルで定義されている *guest* アカウントが共有にアクセスできます。*false* または未定義のまま (デフォルト) になっている場合は、*guest* アカウントが共有にアクセスできません。このポリシーを使用すると、Windows *Guest* ユーザーをローカルで定義されている UNIX ユーザー名 (*guest* や *nobody* など) にマップできます。

```
# idmap add winname:Guest unixuser:guest
```

その後、必要に応じて */var/smb/smbpasswd* に格納されているパスワードと照合して、ローカルで定義されているアカウントを認証できます。詳細は、*idmap* のマニュアルページを参照してください。

- *rw=[*|[[-]criterion][:[-]criterion*]...

  rw ポリシーでは、指定されたアクセスリストに一致するクライアントへのアクセスが許可または拒否されます。

  アクセスリストには、すべてを意味する単一のアスタリスク (*) またはクライアントアクセス条件のコロン区切りのリストのいずれかが含まれています。ここでそれぞれの *criterion* は、*拒否*を意味するオプションのマイナス記号 (-) と、その後に続くホスト名、ネットワークグループ、完全な LDAP または DNS ドメイン名、@ 記号、およびすべてまたは一部の IP アドレスまたはドメイン名で構成されます。アクセスリストは、クライアントでいずれかの基準が満たされるまで、左から右へと評価されます。詳細は、*share_smb* のマニュアルページを参照してください。

- *ro=[*|[[-]criterion][:[-]criterion*]...

  ro ポリシーでは、アクセスリストに一致するクライアントへの読み取り専用アクセスが許可または拒否されます。

- *none=[*|[[-]criterion][:[-]criterion*]...

  none ポリシーでは、アクセスリストに一致するクライアントへのアクセスが拒否されます。アクセスリストがアスタリスク (*) になっている場合は、*ro* および *rw* ポリシーで *none* ポリシーをオーバーライドできます。

この例では、クライアント *smbclient1* および *smbclient2* とは読み取り/書き込み、*smbclient3* とは読み取り専用で、*/qfsms* ファイルシステムを共有します。

```
[qfssmb]root@solaris:~# share -F smb -o rw=smbclient1:smbclient2
ro=smbclient3 /qfsms
```

このコマンドを入力すると、自動的に SMB サーバーデーモン *smbd* が再起動されます。

3. 共有パラメータを確認します。コマンド *share -F nfs* を使用します。

   この例では、共有が適切に構成されたことがコマンドの出力に表示されます。

```
[qfssmb]root@solaris:~# share -F smb /qfsms
sec=sys,rw=smbclient1:smbclient2,ro=smbclient3
[qfssmb]root@solaris:~#
```

4. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
5. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

8

SAM-Remote の構成

Oracle Hierarchical Storage Manager software の SAM-Remote 機能を使用すると、Oracle HSM ファイルシステムホストは、リモートの Oracle HSM ファイルシステムホストにホストされているテープメディアとドライブにアクセスできます。ローカルホストは、SAM-Remote サーバーとして機能するリモートホストの SAM-Remote クライアントとしてテープリソースにアクセスします。クライアントのアーカイブポリシーでは通常、ローカルの磁気またはソリッドステート (SSD) ディスクアーカイブで 1 または 2 つのコピーが維持され、サーバーによって提供されるリモートテープで 1 つまたは 2 つのコピーが維持されます。各ホスト上のマスター構成ファイル `/etc/opt/SUNWsamfs/mcf` は、特殊な SAM-Remote 装置タイプを使用して、共有リソースとクライアント/サーバーの関係を定義します。

SAM-Remote クライアントおよびサーバーの多数のアーカイブとデータ保護の要件に対処できます。

- テープアーカイブの利点を、ライブラリとドライブがない Oracle HSM ホストに拡張できます。
- 地域のオフィスや衛星キャンパスにホストされた Oracle HSM ファイルシステムのテープリソースの保守と管理を一元化できます。

  中心となるメインオフィスのデータセンターでは、Oracle HSM ファイルシステムホストにはテープライブラリが接続されており、このホストが SAM-Remote サーバーとして動作します。より小さい分散型オフィスでは、Oracle HSM ファイルシステムホストにはディスクアーカイブのみがあり、このホストは SAM-Remote クライアントとして機能します。すべてのホストが、アーカイブ済みデータのローカルコピーとテープコピーの両方を維持します。ただし、ハードウェアとメディアのインベントリは中央のデータセンターに集中しており、ここではもっとも効率的かつ最小限のコストで保守できます。
- バックアップと障害回復のために、オフサイトのテープコピーを自動的に作成して保守できます。

  すべての Oracle HSM ファイルシステムホストにテープライブラリが接続されています。それぞれのホストは、SAM-Remote クライアントと、オフサイトの場所にあるクライアントに対

するサーバーとして動作します。それぞれの Oracle HSM ホストは、ローカルリソースを使用してローカルディスクとテープコピーを作成します。それぞれのホストが、対応するホストによって提供されるリソースを使用してリモートテープコピーを作成し、それぞれが対応するホストにテープリソースを提供します。そのため、通常のアーカイブ処理の一部として、2つのファイルシステムのオフサイトコピーが自動的に作成されます。

- ローカルリソースが使用できないときに、リモートアーカイブストレージリソースにアクセスするように Oracle HSM ファイルシステムホストを構成できます。

  すべての Oracle HSM ファイルシステムホストでテープライブラリが接続され、それぞれのファイルシステムホストが、SAM-Remote クライアントと、別の場所にあるクライアントに対するサーバーとして機能します。それぞれの Oracle HSM ホストは、ローカルリソースを使用してローカルディスクとテープコピーを作成します。ただし、ホストがそのローカルライブラリにアクセスできない場合でも、そのリモートホストによって提供されたメディアとリソースを使用してファイルをアーカイブして取得できます。

この章では、SAM-Remote クライアント/サーバーネットワークを構成するプロセスの概要を示します。このセクションでは、次のタスクについて説明します。

8.1. すべての SAM-Remote ホストで同じソフトウェアが使用されていることの確認

SAM-Remote クライアントとサーバーには、同じリビジョンの Oracle HSM ソフトウェアがインストールされている必要があります。次の手順を使用して、リビジョンレベルを確認します。

1. SAM-Remote サーバーホストに *root* としてログインします。

   この例では、サーバーホストは *server1* です。

```
[server1]root@solaris:~#
```

2. SAM-Remote クライアントホストに *root* としてログインします。

この例では、端末ウィンドウを開き、*ssh* を使用してホスト *client1* にログインします。

```
[server1]root@solaris:~# ssh root@client1
Password: ...
[client1]root@solaris:~#
```

3. すべての SAM-Remote サーバーおよびクライアント上で、Oracle HSM パッケージのリビジョンレベルが同じであることを確認します。それぞれの SAM-Remote ホストで、コマンド *samcmd l* を使用して、構成の詳細を一覧表示します。結果を比較します。

   この例では、*server1* の結果と *client1* の結果を比較します。両方で同じリリースの Oracle HSM ソフトウェアが使用されています。

```
[server1]root@solaris:~# samcmd l
Usage information samcmd     6.0  10:20:34 Feb 20 2015
samcmd on server1
...
[server1]root@solaris:~#

[client1]root@solaris:~# samcmd l
Usage information samcmd     6.0  10:20:37 Feb 20 2015
samcmd on client1
...
[server1]root@solaris:~#
```

4. 4章「*Oracle HSM and QFS Software のインストール*」の手順を使用して、すべての SAM-Remote サーバーとクライアントが同じリビジョンレベルになるまで、必要に応じてホストソフトウェアを更新します。
5. 次に、Oracle HSM プロセスの停止を実行します。

8.2. Oracle HSM プロセスの停止

1. SAM-Remote サーバーホストに *root* としてログインします。

   この例では、サーバーの名前は *server1* です。

```
[server1]root@solaris:~#
```

2. 構成されているデバイスの装置番号を取得します。コマンド *samcmd c* を使用します。

   この例では、デバイスの番号は *801*、*802*、*803*、および *804* です。

```
[server1]root@solaris:~# samcmd c
Device configuration samcmd     6.0  10:20:34 Feb 20 2015
samcmd on server1
Device configuration:
ty  eq  state   device_name                        fs   family_set
rb  800 on      /dev/scsi/changer/c1t0d5           800  rb800
tp  801 on      /dev/rmt/0cbn                      801  rb800
tp  802 on      /dev/rmt/1cbn                      802  rb800
tp  803 on      /dev/rmt/2cbn                      803  rb800
tp  804 on      /dev/rmt/3cbn                      804  rb800
```

3.
4. アーカイブプロセスがある場合、すべてアイドル状態にします。コマンド *samcmd aridle* を使用します。

   このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

```
[samfs-mds]root@solaris:~# samcmd aridle
[samfs-mds]root@solaris:~#
```

5. ステージングプロセスがある場合、すべてアイドル状態にします。コマンド *samcmd stidle* を使用します。

   このコマンドは現在のアーカイブおよびステージングを完了できますが、新しいジョブは開始されません。

```
[samfs-mds]root@solaris:~# samcmd stidle
[samfs-mds]root@solaris:~#
```

6. アクティブなアーカイブジョブが完了するまで待機します。コマンド *samcmd a* を使用して、アーカイブプロセスのステータスを確認します。

アーカイブプロセスが *Waiting for :arrun* の場合、アーカイブプロセスはアイドル状態になっています。

```
[samfs-mds]root@solaris:~# samcmd a
Archiver status samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samfs-mds
sam-archiverd:  Waiting for :arrun
sam-arfind: ...
Waiting for :arrun
```

7. アクティブなステージングジョブが完了するまで待機します。コマンド *samcmd u* を使用してステージングプロセスのステータスを確認します。

   ステージングプロセスが *Waiting for :strun* の場合、ステージングプロセスはアイドル状態になっています。

```
[samfs-mds]root@solaris:~# samcmd u
Staging queue samcmd      6.0 14:20:34 Feb 22 2015
samcmd on solaris.demo.lan
Staging queue by media type: all
sam-stagerd:  Waiting for :strun
root@solaris:~#
```

8. すべてのリムーバブルメディアドライブをアイドル状態にしてから、続行します。ドライブごとに、コマンド *samcmd equipment-number idle* を使用します。ここで *equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイル内のドライブに割り当てられている装置の順序番号です。

   このコマンドはドライブを「*off*」にする前に、現在のアーカイブジョブおよびステージングジョブを完了できますが、新しいジョブは開始されません。この例では、4 つのドライブ (順序番号 *801*、*802*、*803*、*804*) をアイドル状態にします。

```
[samfs-mds]root@solaris:~# samcmd 801 idle
[samfs-mds]root@solaris:~# samcmd 802 idle
[samfs-mds]root@solaris:~# samcmd 803 idle
[samfs-mds]root@solaris:~# samcmd 804 idle
[samfs-mds]root@solaris:~#
```

9. 実行中のジョブが完了するまで待機します。

   コマンド *samcmd r* を使用すると、ドライブのステータスを確認できます。すべてのドライブが「*notrdy*」または「*empty*」の場合は、続行できる状態になっています。

```
[samfs-mds]root@solaris:~# samcmd r
Removable media samcmd     6.0 14:20:34 Feb 22 2015
samcmd on samqfs1host
ty   eq   status      act  use  state  vsn
li  801   ---------p    0   0%  notrdy
          empty
li  802   ---------p    0   0%  notrdy
          empty
li  803   ---------p    0   0%  notrdy
          empty
li  804   ---------p    0   0%  notrdy
          empty
[samfs-mds]root@solaris:~#
```

10. アーカイバおよびステージャープロセスがアイドル状態で、テープドライブがすべて「*notrdy*」になっている場合は、ライブラリ制御デーモンを停止します。コマンド *samd stop* を使用します。

```
[samfs-mds]root@solaris:~# samd stop
[samfs-mds]root@solaris:~#
```

11. 次に、SAM-Remote サーバーの構成を実行します。

8.3. SAM-Remote サーバーの構成

SAM-Remote サーバーは、接続されているロボットテープライブラリとテープドライブを、それ自体が Oracle HSM ファイルシステムホストであるリモートクライアントで使用できるようにする、Oracle HSM ファイルシステムホストです。Oracle HSM プロセスを起動するには、SAM-Remote サーバーで少なくとも 1 つの QFS ファイルシステムをマウントする必要があります。

SAM-Remote サーバーを構成するには、次のタスクを実行します。

- SAM-Remote サーバーの **mcf** ファイルでのリモート共有アーカイブ装置の定義
- **samremote** サーバー構成ファイルの作成

8.3.1. SAM-Remote サーバーの mcf ファイルでのリモート共有アーカイブ装置の定義

1. SAM-Remote サーバーホストに *root* としてログインします。

   この例では、サーバーの名前は *server1* です。

```
[server1]root@solaris:~#
```

2. サーバー上で、*/etc/opt/SUNWsamfs/mcf* ファイルをテキストエディタで開き、アーカイブ装置の定義までスクロールダウンします。

   次の例では、*vi* エディタを使用します。このファイルでは、1 つの Oracle HSM アーカイブファイルシステム *fs600*と、4 台のドライブを保持する 1 つのテープライブラリ *rb800* を定義します。この例では、明確にするために実際のファイルには存在しない可能性のある見出しも含まれ、長いデバイスパスが短縮されていることに注意してください。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
#------------------------------------------------------------------------
# Oracle HSM archiving file system fs600
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
fs600                      600       ms        fs600   on
 /dev/dsk/c9t60...F4d0s7    610       md        fs600   on
 /dev/dsk/c9t60...81d0s7    611       md        fs600   on
#------------------------------------------------------------------------
# Local tape archive rb800
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/dev/scsi/changer/c1t0d5   800       rb        rb800   on
 /dev/rmt/0cbn              801       tp        rb800   on
 /dev/rmt/1cbn              802       tp        rb800   on
 /dev/rmt/2cbn              803       tp        rb800   on
 /dev/rmt/3cbn              804       tp        rb800   on
```

3. アーカイブ装置の定義が終了したら、テープリソースをクライアントで使用可能にするサーバーのエントリを開始します。「*Equipment Identifier*」フィールドに SAM-Remote サーバー構成ファイル */etc/opt/SUNWsamfs/samremote* へのパスを入力し、装置番号を割り当てます。

   この例では、いくつかの見出しをコメントとして追加し、装置番号 *500* をサーバー *samremote* に割り当てます。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#================================================================================
# Server samremote shares tape hardware and media with clients
# Equipment                   Equipment Equipment Family  Device Additional
# Identifier                  Ordinal   Type      Set     State  Parameters
#--------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/samremote  500
```

4. 新しいエントリの「*Equipment Type*」フィールドに、SAM-Remote サーバー装置を表す *ss* を入力します。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#================================================================================
# Server samremote shares tape hardware and media with clients
# Equipment                   Equipment Equipment Family  Device Additional
# Identifier                  Ordinal   Type      Set     State  Parameters
#--------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/samremote  500       ss
```

5. すべてのホストおよびサーバー間で一意の「*Family Set*」の名前を割り当てて、デバイスを *on* に設定します。

   この例では、新しい装置にファミリセット名 *ss500* を割り当てます。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#================================================================================
```

```
# Server samremote shares tape hardware and media with clients
# Equipment                   Equipment Equipment Family  Device Additional
# Identifier                  Ordinal   Type      Set     State  Parameters
#--------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/samremote  500       ss        ss500   on
```

6. 10 個を超える SAM-Remote クライアントを構成する場合、1 - 10 個までの連続するクライアントグループごとに、サーバー装置 (タイプ *ss*) のエントリを追加します。
7. ファイルを保存して、エディタを閉じます。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
/etc/opt/SUNWsamfs/samremote  500       ss        ss500   on
:wq
[server1]root@solaris:~#
```

8. 次に、**samremote** サーバー構成ファイルの作成を実行します。

8.3.2. samremote サーバー構成ファイルの作成

SAM-Remote サーバー構成ファイルでは、各クライアントで使用されるディスクバッファー特性とメディアを定義します。構成する必要のあるサーバーごとに、次の手順を実行します。

1. SAM-Remote サーバーホストに *root* としてログインします。

   この例では、サーバーの名前は *server1* です。

```
[server1]root@solaris:~#
```

2. サーバー上で、*/etc/opt/SUNWsamfs/samremote* ファイルをテキストエディタで作成します。

   この例では、*vi* エディタを使用してファイルを作成します。ハッシュ (*#*) 記号で示された説明的なコメントがいくつか含まれるファイルを作成することから始めます。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/samremote
# Server Configuration File:
# Defines the disk buffer and media that is available to each client.
```

3. 1 番目のクライアントエントリは、改行して、1 列目にクライアントのホスト名、IP アドレス、または完全修飾ドメイン名を入力することから開始します。

   クライアント識別子の行は、空白以外の文字で開始する必要があります。この例では、ホスト名 *client1* を使用してクライアントを識別します。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/samremote
# Server Configuration File:
# Defines the disk buffer and media that is available to each client.
client1
```

4. クライアントと共有するメディアの識別を開始します。*indent media* 形式の新しい行を開始します。ここで *indent* は 1 つ以上の空白文字、*media* は SAM-remote キーワードです。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/samremote
# Server Configuration File:
# Defines the disk buffer and media that is available to each client.
client1
        media
```

5. *indent equipment-number media-type VSNs* 形式の新しい行を使用して、各メディアタイプおよびソースを識別します。ここでは:

   - *indent* は、1 つ以上の空白文字です。
   - *equipment-number* は、*mcf* ファイルでアーカイブストレージ装置を識別する装置番号です。
   - *media-type* は、この装置で使用されるテープメディアのメディア識別子です (Oracle HSM メディアタイプの完全なリストについては、付録A「*装置タイプの用語集*」を参照)。
   - *VSNs* は、1 つ以上のボリュームシリアル番号 (最大 31 文字の英数字の文字列) を空白文字で区切ったリストです。

   この例では、共有メディアの 1 つのソースである、装置番号が *800* のテープライブラリに収容されている一連のテープボリューム (タイプ *tp*) を識別します。使用可能なボリュームは、括弧で囲んだ正規表現で指定されます。式 *VOL0[0-1][0-9]* は、*client1* をボリューム *VOL000-VOL019* に制限します。

```
client1
       media
         800 tp (VOL0[0-1][0-9])
```

各行には、1 つのメディアタイプしか指定できないことに注意してください。そのため、ライブラリで複数のメディアタイプがサポートされている場合は、新しいエントリに各タイプを指定します。

```
      media
         800 ti VOL500 VOL501
         800 li (VOL0[0-1][0-9])
```

6. クライアントと共有するメディアの識別が完了したら、SAM-Remote キーワード *endmedia* を入力して、リストを閉じます。

   この例では、この時点で *client1* が完全に構成されました。

```
client1
       media
         800 tp (VOL0[0-1][0-9])
       endmedia
```

7. 追加のクライアントを構成する必要がある場合は、ここで実行します。それぞれ最大 10 個までの新しいクライアント構成レコードを追加します。その後、ファイルを保存してエディタを閉じます。

   ボリュームの競合やデータ損失の可能性を回避するには、*クライアントが同じリムーバブルメディアボリュームを共有していないことを確認します。*

   この例では、1 つの追加クライアント *client2* を構成します。2 番目のクライアントは、*client1* (装置番号 *800*) と同じテープライブラリに収容されている一連のテープボリュームにアクセスします。ただし、この構成内の正規表現では別のボリュームセット *VOL020-VOL039* が指定されています。

```
# Server Configuration File:
# Defines the disk buffer and media that is available to each client.
client1
```

```
        media
          800 tp (VOL0[0-1][0-9])
        endmedia
client2
        media
          800 tp (VOL02-3][0-9])
        endmedia
:wq
[server1]root@solaris:~#
```

8. 次に、SAM-Remote クライアントの構成を実行します。

8.4. SAM-Remote クライアントの構成

SAM-Remote クライアントごとに、次のタスクを実行します。

- SAM-Remote クライアントの MCF ファイルでのリモートアーカイブ装置の定義
- SAM-Remote クライアント構成ファイルの作成
- SAM-Remote クライアントの MCF ファイルでのリモートアーカイブ装置の定義
- SAM-Remote クライアント構成ファイルの作成
- SAM-Remote クライアントでの **`archiver.cmd`** ファイルの構成

8.4.1. SAM-Remote クライアントの MCF ファイルでのリモートアーカイブ装置の定義

1. SAM-Remote クライアントホストに *`root`* としてログインします。

   この例では、SAM-Remote クライアントの名前は *`client1`* です。

```
[client1]root@solaris:~#
```

2. クライアント上で、*`/etc/opt/SUNWsamfs/mcf`* ファイルをテキストエディタで開き、アーカイブ装置の定義までスクロールダウンします。

   次の例では、*`vi`* エディタを使用します。このファイルでは、1 つの Oracle HSM アーカイブファイルシステム *`fs100`* を定義します。ローカルコピーは、ローカル ZFS ファイルシステムのディスクアーカイブ *`DISKVOL1`* に格納されます。この例では、明確にするために実際のファイルには存在しない可能性のある見出しも含まれ、長いデバイスパスが短縮されていることに注意してください。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Client's /etc/opt/SUNWsamfs/mcf file
#========================================================================
# Oracle HSM archiving file system "fs100"
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
fs100                      100       ms        fs100   on
 /dev/dsk/c10t60...7Bd0s7   110       md        fs100   on
 /dev/dsk/c10t60...48d0s7   111       md        fs100   on
#========================================================================
# Disk archive "/diskvols/DISKVOL1" stores local archive copies
```

3. アーカイブ装置の定義が終了したら、サーバーをクライアントで使用可能にするため、装置に対して入力を開始します。「*Equipment Identifier*」フィールドに、SAM-Remote サーバー構成ファイルへのパスを入力して、装置番号を割り当てます。

   この例では、クライアント構成 */etc/opt/SUNWsamfs/sc400* を指定して、クライアントに装置番号 *400* を割り当てます。いくつかの見出しもコメントとして追加します。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
# Disk archive "/diskvols/DISKVOL1" stores local archive copies
#
#========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/sc400   400
```

4. 新しいエントリの「*Equipment Type*」フィールドに、SAM-Remote クライアント装置を表す *sc* を入力します。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#========================================================================
```

```
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc
```

5. 「*Family Set*」フィールドに、すべてのホストおよびサーバー間で一意のファミリセット名を割り当て、デバイスを *on* に設定します。

   この例では、ファミリセット名 *ss500* を新しい装置に割り当てます。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#===========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
```

6. SAM-Remote サーバーによって使用可能になるテープドライブごとに、SAM-Remote 擬似デバイスを SAM-Remote クライアント *sc* 装置に追加します。「*Equipment Identifier*」フィールドに、*/dev/samrd/rddevice-number* 形式のエントリを追加します。ここで *device-number* は整数です。

   この例では、2 つの擬似デバイス */dev/samrd/rd 0* と */dev/samrd/rd 1* に対して入力を開始します。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#===========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/sc400   400       sc        sc400   on
 /dev/samrd/rd0
 /dev/samrd/rd1
```

7. 各擬似デバイスの「*Equipment Ordinal*」フィールドに、*sc* 装置に割り当てた範囲内の番号を入力します。

   この例では、*/dev/samrd/rd0* に装置番号 *410*、*/dev/samrd/rd1* に装置番号 *420* を割り当てます。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#================================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
 /dev/samrd/rd0             410
 /dev/samrd/rd1             420
```

8. 各 SAM-Remote 擬似デバイスの「*Equipment Type*」フィールドに、*rd* (SAM-Remote 擬似デバイスの装置タイプ) を入力します。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#================================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
 /dev/samrd/rd0             410       rd
 /dev/samrd/rd1             420       rd
```

9. 各擬似デバイスの「*Family Set*」フィールドに、*sc* 装置のファミリセット名を入力します。

   この例では、ファミリセット名 *ss500* を使用します。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
```

```
#========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
 /dev/samrd/rd0             410       rd        ss500
 /dev/samrd/rd1             420       rd        ss500
```

10. 各擬似デバイスの「*Device State*」フィールドに、*on* を入力します。その後、ファイルを保存してエディタを閉じます。

    この例では、*/dev/samrd/rd0* に装置番号 *410*、*/dev/samrd/rd1* に装置番号 *420* を割り当てます。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
...
#========================================================================
# Client "sc400" accesses tape resources on server "samremote" (ss500)
# Equipment                Equipment Equipment Family  Device Additional
# Identifier               Ordinal   Type      Set     State  Parameters
#------------------------  --------- --------- ------  ------ ----------
/etc/opt/SUNWsamfs/ss500   400       sc        ss500   on
 /dev/samrd/rd0             410       rd        ss500   on
 /dev/samrd/rd1             420       rd        ss500   on
:wq
[client1]root@solaris:~#
```

11. 次に、SAM-Remote クライアント構成ファイルの作成を実行します。

8.4.2. SAM-Remote クライアント構成ファイルの作成

SAM-Remote クライアントごとに、次の手順を実行します。

1. SAM-Remote クライアントホストに *root* としてログインします。

   この例では、SAM-Remote クライアントの名前は *client1* です。

```
[client1]root@solaris:~#
```

2. クライアント上で、*/etc/opt/SUNWsamfs/family-set-name* ファイルをテキストエディタで開きます。ここで *family-set-name* は、*mcf* ファイルで使用されるリモート装置のファミリセット名です。

   この例では、*vi* エディタを使用してファイルを作成して、ファミリセット *ss500* に指定します。また、ハッシュ (*#*) 記号で示された説明的なコメントがいくつか含まれるファイルも作成します。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/sc400
# Client's SAM-Remote client configuration file: /opt/SUNWsamfs/sc400
# This file identifies the host of the SAM-Remote server.
```

3. 改行して、1 列目にサーバーのホスト名、IP アドレス、または完全修飾ドメイン名を入力することで、サーバーに対応する単一のエントリを追加します。その後、ファイルを保存してエディタを閉じます。

   この行は、空白以外の文字で開始する必要があります。この例では、ホスト名 *server1* を使用してサーバーを識別します。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/samremote
# Client's SAM-Remote server configuration file: /opt/SUNWsamfs/sc400
# This file identifies the host of the SAM-Remote server.
server1
:wq
[client1]root@solaris:~#
```

4. 次に、SAM-Remote クライアントでの **archiver.cmd** ファイルの構成を実行します。

8.4.3. SAM-Remote クライアントでの archiver.cmd ファイルの構成

1. SAM-Remote クライアントホストに *root* としてログインします。

   この例では、SAM-Remote クライアントの名前は *client1* です。

```
[client1]root@solaris:~#
```

2. テキストエディタで */etc/opt/SUNWsamfs/archiver.cmd* ファイルを開き、キーワード *params* で始まり、キーワード *endparams* で終わるコピーパラメータディレクティブまでスクロールダウンします。

   この例では、*vi* エディタでファイルを開きます。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#------------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy direct
allfiles.1 -startage 10m -startsize 500M -drives 10
allfiles.2 -startage 24h -startsize 20G -drives 2 -reserve set
endparams
```

3. リモートメディア上にアーカイブされるすべてのアーカイブセットのコピーパラメータを確認します。これらのいずれかに *-tapenonstop* または *-offline_copy direct* ディレクティブ (あるいはその両方) が含まれている場合は、この時点で、これらのディレクティブを削除します。

   この例では、*all* パラメータで、すべてのコピーに *-offline_copy direct* ディレクティブが指定されています。したがって、リモートメディア *allfiles.3* に送信する予定のコピーに *-offline_copy none* を指定することによって、このディレクティブをオーバーライドします。

```
#------------------------------------------------------------------------
# Copy Parameter Directives
# Copy Parameter Directives
params
allsets -sort path -offline_copy direct
allfiles.1 -startage 10m -startsize 500M -drives 10
allfiles.2 -startage 24h -startsize 20G -drives 2 -reserve set offline_copy none
endparams
```

4. SAM-Remote キーワード *vsns* で始まり、キーワード *endvsns* で終わる VSN ディレクティブまでスクロールダウンします。

次の例では、*vi* エディタを使用します。現在メディアが割り当てられている唯一のコピー *allfiles.1* は、ローカルのディスクアーカイブボリューム *qfs200* を使用して作成されます。

```
...
endparams
#-------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk qfs200
endvsns
```

5. サーバーの */etc/opt/SUNWsamfs/samremote* ファイルで、このクライアント用に指定されているリモートメディアに、アーカイブコピーを割り当てます。その後、ファイルを保存してエディタを閉じます。

   この例では、*client1* を構成します。コピー *allfiles.2* は、*samremote* サーバー構成ファイルに指定された、*VOL000-VOL019* の範囲のリモートテープボリュームを使用して作成されます。

```
...
endparams
#-------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk qfs200
allfiles.2 tp VOL0[0-1][0-9]
endvsns
:wq
[client1]root@solaris:~#
```

6. 次に、SAM-Remote サーバーでのアーカイブ構成の検証を実行します。

8.5. SAM-Remote サーバーでのアーカイブ構成の検証

1. SAM-Remote サーバーホストに *root* としてログインします。

   この例では、SAM-Remote サーバーの名前は *server1* です。

```
[server1]root@solaris:~#
```

2. サーバー上で Oracle HSM プロセスを起動します。コマンド *samd start* を使用します。

```
[server1]root@solaris:~# samd start
```

3. サーバーホスト上で、共有デバイスサーバーのステータスを確認します。コマンド *samcmd s* を使用します。

   この例では、装置番号が *500* の SAM-Remote サーバー装置 (タイプ *ss*) が *on* になっていて、正常に動作しています。

```
[server1]root@solaris:~# samcmd s
Device status samcmd     6.0  11:20:34 Feb 20 2015
samcmd on server1
ty    eq  state   device_name                    fs     status
rb    800 on      /dev/scsi/changer/c1t0d5       800    m--------r
tp    801 on      /dev/rmt/0cbn                  800    ---------p
  empty
tp    802 on      /dev/rmt/1cbn                  800    ---------p
  empty
tp    803 on      /dev/rmt/2cbn                  800    ---------p
  empty
tp    804 on      /dev/rmt/3cbn                  800    ---------p
  empty
ss    500 on      /etc/opt/SUNWsamfs/samremote     ss500  -------o-r
[server1]root@solaris:~#
```

4. 共有デバイスサーバーが *on* ではない場合、サーバーホストの */etc/opt/SUNWsamfs/mcf* ファイルで正しく定義されていることを確認します。*/etc/opt/SUNWsamfs/samremote* ファイルが正しいこと、およびこのファイルが正しい場所にあることを確認してください。

   手順「SAM-Remote サーバーの **mcf** ファイルでのリモート共有アーカイブ装置の定義」および「**samremote** サーバー構成ファイルの作成」を参照してください。

5. サーバー上で、SAM-Remote クライアントの接続ステータスを確認します。コマンド *samcmd R* を使用します。

この例では、*client1* と *client2* の両方の状態が *0005* になっているため、*connected* になります (状態 *0004* は接続されていないことを示します)。

```
[server1]root@solaris:~# samcmd R
Remote server eq: 500 addr: 00003858 samcmd 6.0  11:20:44 Feb 20 2015
samcmd on server1
message:
Client IPv4: client1 192.10.10.3 port - 5000
 client index - 0 port - 31842 flags - 0005 connected
Client IPv4: client2 10.1.229.97 port - 5000
 client index - 1 port - 32848 flags - 0005 connected
[server1]root@solaris:~#
```

6. 共有デバイスクライアントが接続されていない (状態 *0004*) 場合は、ネットワーク接続を確認してください。サーバーとクライアント (複数) が相互にホスト名とアドレスを解決できることを確認します。サーバーとクライアント (複数) が相互にアクセスできることを確認します。

   この例では、*getent* および *ping* コマンドとともに *ssh* を使用して、各ホストから SAM-Remote 構成内のその他の各ホストまでの接続を確認します。

```
[server1]root@solaris:~# getent hosts client1
192.10.10.3  client1
[server1]root@solaris:~# getent hosts 192.10.10.3
192.10.10.3  client1
[server1]root@solaris:~# ping 192.10.10.3
192.10.10.31 is alive
[server1]root@solaris:~# getent hosts client2
10.1.229.97 client2
[server1]root@solaris:~# getent hosts 10.1.229.97
10.1.229.97 client2
[server1]root@solaris:~# ping 10.1.229.97
192.10.10.31 is alive
[server1]root@solaris:~# ssh root@client1
Password: ...
[client1]root@solaris:~# getent hosts server1
192.10.201.12  server1
```

```
...
[client1]root@solaris:~# exit
[server1]root@solaris:~# ssh root@client2
Password: ...
[client2]root@solaris:~# getent hosts server1
192.10.201.12  server1
...
[client2]root@solaris:~# exit
[server1]root@solaris:~#
```

7. 共有デバイスクライアントが接続されていない (状態 *0004*) 場合は、クライアントホストの */etc/opt/SUNWsamfs/mcf* ファイルで現在定義されていることを確認します。サーバーホストが */etc/opt/SUNWsamfs/family-set-name* ファイルで正しく識別されていて、そのファイルがクライアントホスト上の適切な場所に配置されていることを確認します。次に、クライアントホストがサーバーホスト上の */etc/opt/SUNWsamfs/samremote* ファイルで正しく識別されていることを確認します。

   手順「SAM-Remote クライアントの MCF ファイルでのリモートアーカイブ装置の定義」および「SAM-Remote クライアント構成ファイルの作成」を参照してください。
8. クライアント上で、サーバーホストが */etc/opt/SUNWsamfs/family-set-name* ファイルで正しく識別されていて、そのファイルがクライアントホスト上の適切な場所に配置されていることを確認します。

   「SAM-Remote クライアント構成ファイルの作成」の手順を参照してください。
9. 共有デバイスクライアントが接続されておらず (状態 *0004*)、クライアント側の構成ファイルに問題がない場合は、サーバーを確認してください。クライアントホストが */etc/opt/SUNWsamfs/samremote* ファイルで正しく識別されていることを確認します。

   「**samremote** サーバー構成ファイルの作成」の手順を参照してください。
10. サーバー上で、各クライアントが共有テープライブラリのカタログにアクセスし、使用可能なボリュームを表示できることを確認します。コマンド *samcmd v equipment-number* を使用します。ここで *equipment-number* は、クライアントの *mcf* ファイルで SAM-Remote クライアント装置に割り当てられている装置番号です。

    この例では、*client1* を確認するため、SAM-Remote クライアント装置 */etc/opt/SUNWsamfs/sc400* の装置番号は *400* です。出力には、*client1* がアクセスできるボリューム (*VOL000* - *VOL019*) が正しく一覧表示されています。

```
[server1]root@solaris:~# samcmd v 400
Robot catalog samcmd     6.0  12:20:40 Feb 20 2015
samcmd on server1
Robot VSN catalog by slot       : eq 400
slot     access time  count use flags         ty vsn
   3     none         0     0%  -il-o-b-----  li VOL000
   7     none         0     0%  -il-o-b-----  li VOL001
...
  24     none         0     0%  -il-o-b-----  li VOL019
[server1]root@solaris:~#
```

11. 共有装置クライアントが適切なボリュームを表示できない場合は、ホストファイルを確認してください。サーバーホスト上で、割り当てられているボリュームが */etc/opt/SUNWsamfs/samremote* ファイルで正しく識別されていることを確認します。クライアントホスト上で、*/etc/opt/SUNWsamfs/family-set-name* ファイルでサーバーホストが正しく識別されていることを確認します。

    手順「**samremote** サーバー構成ファイルの作成」および「SAM-Remote クライアント構成ファイルの作成」を参照してください。
12. 次に、各 SAM-Remote クライアントでのアーカイブ構成の検証を実行します。

8.6. 各 SAM-Remote クライアントでのアーカイブ構成の検証

SAM-Remote クライアントごとに、次の手順を実行します。

1. SAM-Remote クライアントホストに *root* としてログインします。

   この例では、SAM-Remote クライアントの名前は *client1* です。

```
[client1]root@solaris:~#
```

2. クライアントホスト上で Oracle HSM プロセスを起動します。コマンド *samd start* を使用します。

```
[client1]root@solaris:~# samd start
[client1]root@solaris:~#
```

3. クライアントホスト上で、共有デバイスクライアントのステータスを確認します。コマンド *samcmd s* を使用します。

   この例では、装置番号が *400* の SAM-Remote クライアント装置 (タイプ *sc*) が *on* になっていて、正常に動作しています。

```
[client1]root@solaris:~# samcmd s
Device status samcmd     6.0  12:20:49 Feb 20 2015
samcmd on client1
ty    eq  state   device_name                    fs     status
sc    400 on      /etc/opt/SUNWsamfs/sc400       sc400  -------o-r
```

4. 共有デバイスクライアントが *on* になっていない場合は、*sc* デバイスが正しく定義されていることを確認します。クライアントホスト上で、*/etc/opt/SUNWsamfs/mcf* ファイルを確認し、*/etc/opt/SUNWsamfs/family-set-name* ファイルが正しいこと、およびこのファイルが正しい場所にあることを確認します。

   手順「SAM-Remote クライアントの MCF ファイルでのリモートアーカイブ装置の定義」および「SAM-Remote クライアント構成ファイルの作成」を参照してください。

5. クライアントホスト上で、*/etc/opt/SUNWsamfs/archiver.cmd* ファイルによって、リモートメディアに適切なボリュームシリアル番号が指定されていることを確認します。コマンド *archiver -A* を使用してファイルを一覧表示します。

   この例では、*client1* を構成します。コピー *allfiles.2* は、*samremote* サーバー構成ファイルに指定された、*VOL000-VOL019* の範囲のリモートテープボリュームの 1 つを使用して作成されます。

```
[client1]root@solaris:~# archiver -A
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
1: # archiver.cmd
2: #---------------------------------------------------------------------
3: # Global Directives
4: archivemeta = off
5: examine = noscan
...
30: #---------------------------------------------------------------------
31: # VSN Directives
32: vsns
```

```
33: allfiles.1 dk qfs200
34: allfiles.2 tp VOL0[0-1][0-9]
36: endvsns
[client1]root@solaris:~#
```

6. *archiver.cmd* ファイルで相違に気付いた場合は、それを修正してから続行してください。
7. リサイクル処理を構成する場合は、SAM-Remote のリサイクル処理の構成を参照してください。

8.7. SAM-Remote のリサイクル処理の構成

SAM-Remote が構成されている場合は、あるホストでのリサイクル処理によって別のホスト上の有効なデータが破棄されないことを保証する必要があります。SAM-Remote サーバー上に構成されているリサイクルディレクティブがリサイクルするメディアは、サーバーが独自のアーカイブセットで使用するメディアのみに制限されます。サーバーは、SAM-Remote クライアントで使用可能なメディアボリュームのリサイクルを試行できません。同様に、SAM-Remote クライアント上に構成されているリサイクルディレクティブは、ローカルまたはサーバーで使用可能にすると指定されたボリュームで、アーカイブされているクライアントデータを保持するメディアのみをリサイクルします。

SAM-Remote 環境でリサイクラの使用を試行する前に、リサイクルプロセスについて十分に理解するようにしてください。そのため、「リサイクル処理」、および *sam-recycler*、*archiver.cmd*、*recycler.cmd*、および *recycler.sh* のマニュアルページを参照してください。

リサイクル処理の動作を理解している場合は、次のタスクを実行します。

- SAM-Remote サーバーでのリサイクル処理の構成
- SAM-Remote クライアントでのリサイクル処理の構成.

8.7.1. SAM-Remote サーバーでのリサイクル処理の構成

SAM-Remote サーバーがホストしているファイルシステム用にリサイクル処理を構成する必要がある場合は、次の手順を実行します。

1. SAM-Remote サーバーに *root* としてログインします。

   この例では、SAM-Remote サーバーの名前は *server1* です。

```
[server1]root@solaris:~#
```

2. テキストエディタで */etc/opt/SUNWsamfs/archiver.cmd* ファイルを開きます。*params* セクションまでスクロールダウンします。

   この例では、*vi* エディタでファイルを開きます。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#-----------------------------------------------------------------------
# Copy Parameter Directives
params
allsets -sort path -offline_copy direct
allfiles.1 -startage 10m -startsize 500M -drives 10
allfiles.2 -startage 24h -startsize 20G -drives 2 -reserve set
endparams
```

3. *archive-set directive-list* の形式で、アーカイブセット単位のリサイクラディレクティブを入力します。ここで archive-set は、アーカイブセットの 1 つ、*directive-list* は、ディレクティブの名前と値のペアを空白文字で区切ったリストです (リサイクルディレクティブの完全なリストについては、*archiver.cmd* のマニュアルページを参照)。

   SAM-Remote の使用時には、*archiver.cmd* ファイルの *params* セクションで、アーカイブセット単位でのリサイクル処理を構成する必要があります。ライブラリ単位でのリサイクル処理は指定できません。

   この例では、アーカイブセット *allfiles.1* および *allfiles.2* のリサイクルディレクティブを追加します。*-recycle_mingain 90* ディレクティブは、少なくともボリューム容量の 90% を回復できなければ、ボリュームをリサイクルしません。*-recycle_hwm 60* ディレクティブは、リムーバブルメディア容量の 60% が使用されるとリサイクルを開始します。*-recycle_vsncount 1* は、一度にリサイクルするリムーバブルメディアボリュームを 1 つだけスケジュールします。

```
#-----------------------------------------------------------------------
#  Copy Parameters Directives
params
allsetsallfiles. -sort path -offline_copy direct
```

```
allfiles.1 -startage 10m -startsize 500M -drives 10
allfiles.1 -recycle_mingain 90
allfiles.2 -startage 24h -startsize 20G -drives 2 -reserve set offline_copy none
allfiles.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
endparams
```

SAM-Remote サーバーで定義されたリサイクルディレクティブは、サーバーが独自のアーカイブセットに使用するアーカイブボリュームにのみ適用されます。サーバーのリサイクルディレクティブは、クライアントからアクセス可能なボリュームには適用されません。

この例では、コピー *allfiles.2* のサーバーのリサイクルディレクティブは、*VSN Directives* セクション *VOL100-VOL199* にサーバーの使用のために一覧表示されているテープボリュームに適用されます。サーバーのリサイクルディレクティブは、*client1* 用に予約済みのボリューム *VOL000-VOL019*、または *client2* 用に予約済みのボリューム *VOL020-VOL039* には適用されません。

```
...
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk DISKVOL1
allfiles.2 tp VOL1[0-9][0-9]
endvsns
```

4. *archiver.cmd* ファイルを保存して、エディタを閉じます。

```
...
endvsns
:wq
[server1]root@solaris:~#
```

5. サーバー上で、*recycler.cmd* ファイルをテキストエディタで作成します。リサイクラログのパスとファイル名を指定します。

   次の例では、*vi* エディタを使用します。ログファイルのデフォルトの場所を指定します。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
```

6. サーバー上の *recycler.cmd* ファイルに、*no-recyle media-type volumes* 形式のディレクティブを追加します。ここで *media-type* は付録A「*装置タイプの用語集*」で指定されたメディアタイプのいずれかであり、*volumes* は SAM-Remote クライアントに割り当てられている各アーカイブストレージボリュームのボリュームシリアル番号を指定する空白文字区切りのリストまたは正規表現です。ファイルを保存して、エディタを閉じます。

   no-recyle ディレクティブは、クライアント専用のストレージリソースを追加で保護します。これは、指定したボリュームがスキップされるように、ホストのリサイクル処理の順序を明示的に変更します。

   この例では、*VOL000-VOL019* と *VOL020-VOL039* の範囲のメディアタイプ *tp* (テープ) ボリュームで *no-recyle* ディレクティブを追加します。

```
[server1]root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/opt/SUNWsamfs/recycler/recycler.log
no_recycle tp VOL0[0-1][0-9] VOL0[2-3][0-9]
:wq
[server1]root@solaris:~#
```

7. 次に、SAM-Remote クライアントでのリサイクル処理の構成を実行します。

8.7.2. SAM-Remote クライアントでのリサイクル処理の構成

クライアントごとに、次の手順を実行します。

1. SAM-Remote クライアントに *root* としてログインします。

   この例では、SAM-Remote クライアントの名前は *client1* です。

```
[client1]root@solaris:~#
```

2. クライアント上で、*/etc/opt/SUNWsamfs/archiver.cmd* ファイルをテキストエディタで開き、コピー *params* セクションまでスクロールダウンします。

   この例では、*vi* エディタでファイルを開きます。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#-----------------------------------------------------------------------
# Copy Parameters Directives
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 6h  -startsize 6G  -startcount 500000
allfiles.2 -startage 24h -startsize 20G -startcount 500000 -archmax 24G
endparams
#-----------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk qfs200
allfiles.2 tp VOL0[0-1][0-9]
endvsns
```

3. *archiver.cmd* ファイルの *params* セクションに、アーカイブセットごとにリサイクラディレクティブを *archive-set directive-list* 形式で入力します。ここで、archive-set は、アーカイブセットの 1 つで、*directive-list* は、ディレクティブの名前/値のペアのスペースで区切られたリストです (リサイクルディレクティブのリストについては、*archiver.cmd* のマニュアルページを参照してください)。その後、ファイルを保存してエディタを閉じます。

   SAM-Remote の使用時には、*archiver.cmd* ファイルの *params* セクションで、アーカイブセット単位でのリサイクル処理を構成する必要があります。ライブラリ単位でのリサイクル処理は指定できません。

   この例では、アーカイブセット *allfiles.1* および *allfiles.2* のリサイクルディレクティブを追加します。*-recycle_mingain 90* ディレクティブは、少なくともボリューム容量の 90% を回復できなければ、ボリュームをリサイクルしません。*-recycle_hwm 60* ディレクティブは、リムーバブルメディア容量の 60% が使用されるとリサイクルを開始します。*-recycle_vsncount 1* ディレクティブは、一度にリサイクルするリムーバブルメディアボリュームを 1 つだけスケジュールします。

```
#-----------------------------------------------------------------------
# Copy Parameters Directives
params
```

```
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 6h  -startsize 6G  -startcount 500000
allfiles.1 -recycle_mingain 90
allfiles.2 -startage 24h -startsize 20G -startcount 500000 -archmax 24G
allsets.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
endparams
```

クライアント上で定義されているリサイクルディレクティブは、クライアントが独自のアーカイブセットで使用するメディアにのみ適用されます。この例では、コピー *allfiles.2* のクライアントのリサイクルディレクティブは、範囲 *VOL000-VOL019* 内のサーバー指定のリモートテープボリュームに適用されます。client2 用に予約済みの範囲 *VOL020-VOL039* 内のボリュームや、サーバー用に予約済みの範囲 *VOL100-VOL119* 内のボリュームには適用されません。

```
...
endparams
#------------------------------------------------------------------------
# VSN Directives
vsns
allfiles.1 dk qfs200
allfiles.2 tp VOL0[0-1][0-9]
endvsns
:wq
[client1]root@solaris:~#
```

4. *archiver.cmd* ファイルを保存して、エディタを閉じます。

```
...
endvsns
:wq
[client]root@solaris:~#
```

5. クライアント上で、*recycler.cmd* ファイルをテキストエディタで作成します。リサイクラログのパスとファイル名を指定します。その後、ファイルを保存してエディタを閉じます。

サーバーおよびクライアントは、クライアントがサーバーまたは *client2* で使用されているどのアーカイブメディアにもアクセスできないように構成されています。そのため、*no-recyle* ディレクティブを追加する必要はありません。

次の例では、*vi* エディタを使用します。ログファイルのデフォルトの場所を指定します。

```
[client1]root@solaris:~# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
:wq
[client1]root@solaris:~#
```

6. すべての SAM-Remote クライアントが構成されるまで、この手順を繰り返します。
7. コマンド *sam-recycler -dvxn* を使用します。各パラメータの効果は次のとおりです。
   - *-d* は、各ボリュームをリサイクル対象として選択した、または選択しなかった理由を示すボリューム選択メッセージを表示します。
   - *-v* は、リサイクル対象のマークが付けられた各ボリュームに存在し、移動する必要のあるファイルを一覧表示します。
   - *-x* は、ボリュームにラベルが付けられた時点よりも古いために回復不可能であるアーカイブコピーが一覧表示された場合に、エラーを返して停止します。
   - *-n* は、実際のリサイクル処理を回避します。リサイクル処理は、*archiver.cmd* ファイル内のすべてのアーカイブセット定義に *-recycle_ignore* が含まれる場合と同様に動作するため、リサイクル構成を壊さずにテストできます。
8. すべての SAM-Remote クライアントおよびサーバーが構成されたら、サイドバンドデータベース機能を使用する予定の場合は、10章「*レポートデータベースの構成*」に進みます。
9. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

9

高可用性ソリューションの準備

Oracle Hierarchical Storage Manager and StorageTek QFS Software 高可用性構成は、ファイルシステムとアーカイブサービスを中断なしでメンテナンスするために設計されています。高可用性ソリューションでは、Oracle Hierarchical Storage Manager または QFS software は、Oracle Solaris Cluster ソフトウェア、冗長なハードウェア、および冗長な通信と統合されています。そのため、ホストシステムまたはコンポーネントで障害が発生したり、管理者によってサービスが停止されたりした場合に、Oracle HSM サービスは、ユーザーとアプリケーションがアクセスできる代替のホストに自動的にフェイルオーバーします。したがって、高可用性構成により、装置とシステムの障害による停止時間を最小限に抑えられます。

ただし、高可用性構成は複雑であり、予期しない相互作用と、場合によってはデータの破損を防ぐために、慎重に設計して配備する必要があります。そのため、この章はサポートされる構成の説明から始まります。このセクションを確認して、可用性の要件にもっとも適う構成を選択してください。後続のセクションでは、選択した構成の設定方法について説明します。

Oracle Solaris Cluster の共有構成ではハードウェアアーキテクチャーを混在させることはできません。すべてのノードが、SPARC アーキテクチャー、x86-64 アーキテクチャー (Solaris 11 .1 のみ)、または 32 ビットの x86 アーキテクチャー (Solaris 10 以前) のいずれかを使用する必要があります。

9.1. サポートされる高可用性構成について

クラスタ化されたマルチホストソリューションでは、ファイルシステム、アプリケーション、オペレーティングシステム、クラスタソフトウェア、およびストレージ間の相互作業は、格納されているデータの整合性を確保するために慎重に制御する必要があります。複雑さと潜在的リスクを最小限に抑えるために、サポートされる高可用性 Oracle HSM 構成は、特定の 4 セットの配備要件に合わせて調整されています。

- HA-QFS、高可用性 QFS の非共有、スタンドアロンのファイルシステム構成
- HA-COTC、高可用性メタデータサーバーを備えた QFS 共有ファイルシステム
- HA-SAM、高可用性、アーカイブ、QFS 共有ファイルシステム構成
- SC-RAC、Oracle RAC の高可用性 QFS 共有ファイルシステム構成.

9.1.1. HA-QFS、高可用性 QFS の非共有、スタンドアロンのファイルシステム構成

高可用性 QFS (HA-QFS) 構成は、ホストの障害時に QFS 非共有のスタンドアロンファイルシステムがアクセス可能な状態のままにします。ファイルシステムは、Solaris Cluster ソフトウェアがタイプ *SUNW.HAStoragePlus* のリソースとして管理する 2 ノードクラスタ内の両方のノード上に構成されます。ただし、どんな場合でも QFS ファイルシステムをマウントするのは 1 つのノードのみです。ファイルシステムをマウントするノードで障害が発生した場合、クラスタリングソフトウェアは、自動的にフェイルオーバーを開始して、残りのノードでファイルシステムを再マウントします。

クライアントは、ファイルサーバーとして機能するアクティブなクラスタノードを使用して、ネットワークファイルシステム (NFS)、高可用性 NFS (HA-NFS)、または SMB/CIFS 共有経由でデータにアクセスします。

実装手順については、「高可用性 QFS 非共有ファイルシステム」を参照してください。

9.1.2. HA-COTC、高可用性メタデータサーバーを備えた QFS 共有ファイルシステム

クラスタ外部の高可用性クライアント (HA-COTC) 構成では、サーバーで障害が発生した場合でも、QFS ファイルシステムクライアントがデータに引き続きアクセスできるように、QFS メタデータサーバーの可用性が維持されます。ファイルシステムは共有されます。QFS のアクティブおよび潜在的なメタデータサーバーは、Solaris Cluster ソフトウェアによって管理される 2 ノードクラスタにホストされます。タイプ *SUNW.qfs* の Oracle HSM 高可用性リソースは、クラスタ内の共有ファイルシステムサーバーのフェイルオーバーを管理します。すべてのクライアントがクラスタの外部にホストされます。クラスタ化されたサーバーは、メタデータの可用性を確保し、入出力ライセンスを発行し、ファイルシステムの整合性を維持します。

アクティブなメタデータサーバーをホストするノードで障害が発生した場合、Solaris Cluster ソフトウェアは、正常なノードで潜在的な MDS を自動的にアクティブにし、フェイルオーバーを開始します。QFS ファイルシステムは共有されているため、新たにアクティブにされたメタデータサーバーノードにすでにマウントされており、クライアントにマウントされたままになります。クライアントは、引き続きメタデータの更新と入出力リースを受信するため、ファイルシステムは中断なしで続行できます。

HA-COTC 構成では、*mm* メタデータデバイスと *mr* データデバイスが物理的に切り離された高パフォーマンスの *ma* ファイルシステムを使用する必要があります。汎用 *ms* ファイルシステムと *md* デバイスはサポートされません。

標準のネットワークファイルシステム (NFS) または SMB/CIFS を使用して、Oracle HSM を実行しないクライアントと HA-COTC ファイルシステムを共有できます。ただし、HA-NFS はサポートされていません。

実装手順については、「高可用性 QFS 共有ファイルシステム、クラスタの外部にあるクライアント」を参照してください。

9.1.3. HA-SAM、高可用性、アーカイブ、QFS 共有ファイルシステム構成

高可用性 Oracle Hierarchical Storage Manager (HA-SAM) 構成では、サーバーホストで障害が発生した場合でも QFS メタデータサーバーと Oracle Hierarchical Storage Manager アプリケーションが動作を継続できるようにすることにより、アーカイブファイルシステムの可用性が維持されます。ファイルシステムは、Solaris Cluster ソフトウェアによって管理される 2 ノードクラスタにホストされた、アクティブおよび潜在的な QFS メタデータサーバーの間で共有されます。タイプ *SUNW.qfs* の Oracle HSM 高可用性リソースは、サーバーのフェイルオーバーを管理します。

アクティブな Oracle HSM メタデータサーバーノードで障害が発生した場合、クラスタリングソフトウェアは、潜在的なメタデータサーバーノードを自動的にアクティブにして、フェイルオーバーを開始します。QFS ファイルシステムは共有されており、すべてのノードにすでにマウントされているため、データとメタデータへのアクセスは中断されません。

クライアントは、ファイルサーバーとして機能するアクティブなクラスタノードを使用して、高可用性ネットワークファイルシステム (HA-NFS)、NFS、または SMB/CIFS 共有経由でデータにアクセスします。

実装手順については、「高可用性 Oracle HSM 共有アーカイブファイルシステム」を参照してください。

9.1.4. SC-RAC、Oracle RAC の高可用性 QFS 共有ファイルシステム構成

Solaris Cluster-Oracle Real Application Cluster (SC-RAC) 構成では、QFS ファイルシステムを使用する高可用性のデータベースソリューションがサポートされます。RAC ソフトウェアは入出力要求の調整、ワークロードの分散、およびクラスタのノードで実行されている複数の Oracle データベースインスタンスの単一で一貫性のあるデータベースファイルセットの維持を行います。SC-RAC 構成では、Oracle データベース、Oracle Real Application Cluster (RAC)、および QFS software は、クラスタ内の複数のノードで実行されます。Solaris

Cluster ソフトウェアは、タイプ *SUNW.qfs* のリソースとしてクラスタを管理します。1 つのノードは、QFS 共有ファイルシステムのメタデータサーバー (MDS) として構成されています。残りのノードは、クライアントとしてファイルシステムを共有する潜在的なメタデータサーバーとして構成されています。アクティブなメタデータサーバーノードで障害が発生した場合、Solaris Cluster ソフトウェアは、正常なノード上の潜在的なメタデータサーバーを自動的にアクティブにして、フェイルオーバーを開始します。QFS ファイルシステムは共有されており、すべてのノードにすでにマウントされているため、データへのアクセスは中断されません。

9.2. 高可用性 QFS 非共有ファイルシステム

高可用性 QFS (HA-QFS) ファイルシステムを構成するには、タイプ *SUNW.HAStoragePlus* のリソースとして管理される 2 ノードの Solaris Cluster に 2 つの同一のホストを設定します。その後、両方のノードで QFS 非共有ファイルシステムを構成します。いつでも 1 つのノードのみがファイルシステムをマウントします。ただし、1 つのノードで障害が発生した場合、クラスタリングソフトウェアは、自動的にフェイルオーバーを開始して、残っているノードでファイルシステムを再マウントします。

高可用性 QFS (HA-QFS) ファイルシステムを設定するには、次のように進めます。

- 両方のクラスタノード上での非共有 QFS ファイルシステムの作成
- 高可用性 QFS ファイルシステムの構成
- 必要に応じて、高可用性ネットワークファイルシステム (HA-NFS) 共有を構成します。

  HA-NFS を設定するための詳細な手順は、*Oracle Solaris Cluster* オンラインドキュメントライブラリに含まれている『*Oracle Solaris Cluster Data Service for Network File System (NFS) ガイド*』に記載されています。

9.2.1. 両方のクラスタノード上での非共有 QFS ファイルシステムの作成

1. クラスタノードの 1 つに *root* としてログインします。

   この例では、ホストは *qfs1mds-node1* および *qfs1mds-node2* です。ホスト *qfs1mds-node1* にログインします。

```
[qfs1mds-node1]root@solaris:~#
```

2. ホストで必要な QFS ファイルシステムを構成しますが、マウントしないでください。

「汎用の **ms** ファイルシステムの構成」または「高パフォーマンス **ma** ファイルシステムの構成」の手順を使用して、ファイルシステムを構成します。HA-QFS 構成では QFS 共有ファイルシステムはサポートされません。

3. 残りのクラスタノードに *root* としてログインします。

   この例では、*ssh* を使用してホスト *qfs1mds-node2* にログインします。

```
[qfs1mds-node1]root@solaris:~# ssh root@qfs1mds-node2
Password:
[qfs1mds-node2]root@solaris:~#
```

4. 2 番目のノードで同一の QFS ファイルシステムを構成します。
5. 次に、高可用性 QFS ファイルシステムの構成を実行します。

9.2.2. 高可用性 QFS ファイルシステムの構成

次のように進めます。

1. クラスタノードの 1 つに *root* としてログインします。

   この例では、ホストは *qfs1mds-node1* および *qfs1mds-node2* です。ホスト *qfs1mds-node1* にログインします。

```
[qfs1mds-node1]root@solaris:~#
```

2. Solaris Cluster ソフトウェアで *SUNW.HAStoragePlus* リソースタイプを定義します (まだ定義していない場合)。コマンド *clresourcetype register SUNW.HAStoragePlus* を使用します。

   HAStoragePlus は、ディスクデバイスグループ、クラスタファイルシステム、およびローカルファイルシステムの間の依存関係を定義して管理する Solaris Cluster リソースタイプです。これは、サービスの再起動の試行時に必要なすべてのコンポーネントの準備が整うように、フェイルオーバー後のデータサービスの開始を調整します。詳細は、*SUNW.HAStoragePlus* のマニュアルページを参照してください。

```
[qfs1mds-node1]root@solaris:~# clresourcetype register SUNW.HAStoragePlus
[qfs1mds-node1]root@solaris:~#
```

3. タイプ *SUNW.HAStoragePlus* の新しい Solaris Cluster リソースと、そのリソースを含める新しいリソースグループを作成します。コマンド */usr/global/bin/clresource create -g resource-group -t SUNW.HAStoragePlus -x FilesystemMountPoints=/global/mount-point -x FilesystemCheckCommand=/bin/true QFS-resource* を使用します。ここでは:
   - *resource-group* は、ファイルシステムリソースグループに選択した名前です。
   - *mount-point* は、QFS ファイルシステムがマウントされているディレクトリです。
   - *QFS-resource* は、*SUNW.HAStoragePlus* リソースに選択した名前です。

   この例では、マウントポイントディレクトリ */global/qfs1* および *SUNW.HAStoragePlus* リソース *haqfs* とともにリソースグループ *qfsrg* を作成します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs1mds-node1]root@solaris:~# clresource create -g qfsrg -t SUNW.HAStoragePlus /
-x FilesystemMountPoints=/global/hsmqfs1/qfs1 /
-x FilesystemCheckCommand=/bin/true haqfs
[qfs1mds-node1]root@solaris:~#
```

4. クラスタ内のノードを表示します。コマンド *clresourcegroup status* を使用します。

   この例では、QFS ファイルシステムホストノードは *qfs1mds-1* と *qfs1mds-2* です。ノード *qfs1mds-1* は *Online* であるため、ファイルシステムをマウントして *qfsrg* リソースグループをホストするプライマリノードです。

```
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name     Node Name     Suspended    Status
----------     ---------     ---------    ------
qfsrg          qfs1mds-1     No           Online
               qfs1mds-2     No           Offline
[qfs1mds-node1]root@solaris:~#
```

5. リソースグループをセカンダリノードに移動して、リソースグループが正しくフェイルオーバーするようにします。Solaris Cluster コマンド *clresourcegroup switch -n node2 group-name* を使用します。ここで *node2* はセカンダリノードの名前で、*group-name* は HA-QFS リソースグループに選択した名前です。次に、*clresourcegroup status* を使用して結果を確認します。

この例では、*haqfs* リソースグループを *qfs1mds-node2* に移動して、指定したノードでリソースグループがオンラインになることを確認します。

```
[qfs1mds-node1]root@solaris:~# clresourcegroup switch -n qfs1mds-node2 qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name     Node Name     Suspended    Status
----------     ---------     ---------    ------
qfsrg          qfs1mds-1     No           Offline
               qfs1mds-2     No           Online
[qfs1mds-node1]root@solaris:~#
```

6. リソースグループをプライマリノードに戻します。Solaris Cluster コマンド *clresourcegroup switch -n node1 group-name* を使用します。ここで *node1* はプライマリノードの名前で、*group-name* は HA-QFS リソースグループに選択した名前です。次に、*clresourcegroup status* を使用して結果を確認します。

   この例では、*qfsrg* リソースグループを *qfs1mds-node1* に正常に戻します。

```
[qfs1mds-node1]root@solaris:~# clresourcegroup switch -n qfs1mds-node1 qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsrg       qfs1mds-node1   No          Online
            qfs1mds-node2   No          Offline
[qfs1mds-node1]root@solaris:~#
```

7. 高可用性ネットワークファイルシステム (HA-NFS) 共有を構成する必要がある場合、この時点で行います。手順については、Oracle Solaris Cluster *オンラインドキュメントライブラリに含まれている*『Oracle Solaris Cluster Data Service for Network File System (NFS) ガイド』を参照してください。
8. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
9. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

9.3. 高可用性 QFS 共有ファイルシステム、クラスタの外部にあるクライアント

クラスタ外部の高可用性クライアント (HA-COTC) 構成は、Solaris Cluster ソフトウェアによって管理される高可用性クラスタのノードに重要なメタデータサーバー (MDS) をホストする非アーカイブ QFS 共有ファイルシステムです。この配置によって、QFS メタデータとファイルアクセスのリースでフェイルオーバーが保護されるため、サーバーで障害が発生した場合にファイルシステムクライアントはデータへのアクセスを失いません。ただし、Solaris Cluster が QFS 共有データの制御のために QFS software と競合しないように、ファイルシステムクライアントとデータデバイスはクラスタの外部にとどまります。

HA-COTC ファイルシステムを構成するには、次のタスクを実行します。

- 両方の HA-COTC クラスタノードにおける QFS 共有ファイルシステムの hosts ファイルの作成
- QFS サーバーおよび HA-COTC クラスタの外部にあるクライアントでのローカル hosts ファイルの作成
- プライマリ HA-COTC クラスタノード上でのアクティブな QFS メタデータサーバーの構成
- セカンダリ HA-COTC クラスタノード上での潜在的な QFS メタデータサーバーの構成
- HA-COTC メタデータサーバーのフェイルオーバーの構成
- HA-COTC クラスタの外部にあるホストを QFS 共有ファイルシステムクライアントとして構成
- 必要に応じて、「NFS と SMB/CIFS を使用した複数のホストからファイルシステムへのアクセス」の説明に従ってネットワークファイルシステム (NFS) 共有を構成します。高可用性 NFS (HA-NFS) はサポートされません。

9.3.1. 両方の HA-COTC クラスタノードにおける QFS 共有ファイルシステムの hosts ファイルの作成

QFS 共有ファイルシステムでは、すべてのホストがファイルシステムのメタデータにアクセスできるように、メタデータサーバーで hosts ファイルを構成する必要があります。hosts ファイルは、*/etc/opt/SUNWsamfs/* ディレクトリに *mcf* ファイルとともに格納されています。共有ファイルシステムの初期作成中に、*sammkfs -S* コマンドを実行すると、このファイルに格納されている設定を使用して共有が構成されます。ここで、次の手順を使用して作成します。

1. HA-COTC クラスタのプライマリノードに *root* としてログインします。

   この例では、ホストは *qfs1mds-node1* および *qfs1mds-node2* です。ホスト *qfs1mds-node1* にログインします。

```
[qfs1mds-node1]root@solaris:~#
```

2. クラスタ構成を表示します。*/usr/global/bin/cluster show* コマンドを使用します。それぞれの *Node Name* のレコードを見つけて、各ネットワークアダプタの *privatehostname*、*Transport Adapter* の名前、および *ip_address* プロパティーを記録します。

   コマンドの出力はかなり長くなる可能性があるため、次の例では、長い表示は省略記号 (...) を使用して省略されています。

   この例では、それぞれのノードには、*qfe3* と *hme0* の 2 つのネットワークインタフェースがあります。

   - *hme0* アダプタには、クラスタがノード間の内部通信に使用するプライベートネットワークの IP アドレスがあります。Solaris Cluster ソフトウェアは、各プライベートアドレスに対応するプライベートホスト名を割り当てます。

     デフォルトでは、プライマリノードのプライベートホスト名は *clusternode1-priv* で、セカンダリノードのプライベートホスト名は *clusternode2-priv* です。
   - *qfe3* アダプタには、クラスタがデータ転送に使用するパブリック IP アドレスとパブリックホスト名である *qfs1mds-node1* と *qfs1mds-node2* があります。

```
[qfs1mds-node1]root@solaris:~# cluster show
...
  === Cluster Nodes ===
  Node Name:                                qfs1mds-node1...
    privatehostname:                           clusternode1-priv...
    Transport Adapter List:                    qfe3, hme0...
    Transport Adapter:                      qfe3...
      Adapter Property(ip_address):            172.16.0.12...
    Transport Adapter:                      hme0...
      Adapter Property(ip_address):            10.0.0.129...
  Node Name:                                qfs1mds-node2...
    privatehostname:                           clusternode2-priv...
    Transport Adapter List:                    qfe3, hme0...
      Adapter Property(ip_address):            172.16.0.13...
    Transport Adapter:                      hme0
      Adapter Property(ip_address):            10.0.0.122
```

3. テキストエディタを使用して、メタデータサーバーでファイル */etc/opt/SUNWsamfs/hosts.family-set-name* を作成します。ここで、*family-set-name* は、ファイルシステムのファミリセット名です。

   この例では、*vi* テキストエディタを使用してファイル *hosts.qfs1* を作成します。ホストテーブル内の列を示すために、コメントを示すハッシュ記号 (#) で各行を開始して、いくつかのオプションの見出しを追加します。

```
[qfs1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1
# /etc/opt/SUNWsamfs/hosts.qfs1
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
```

4. テーブルの最初の列に、プライマリメタデータサーバーノードとセカンダリメタデータサーバーノードのホスト名、その後にいくつかの空白文字を入力します。それぞれのエントリを別の行に入力してください。

   hosts ファイルでは、行は行 (レコード) で、空白文字は列 (フィールド) 区切り文字です。この例では、最初の 2 行の「*Host Name*」列には、ファイルシステムのメタデータサーバーをホストするクラスタノードのホスト名である、値 *qfs1mds-node1* と *qfs1mds-node2* が含まれています。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1
qfs1mds-node2
```

5. 各行の 2 番目の列では、ホスト *Host Name* の *Network Interface* 情報の指定を開始します。それぞれの HA-COTC クラスタノードの Solaris Cluster プライベートホスト名またはプライベートネットワークアドレスと、その後に続けてコンマを入力します。

   HA-COTC サーバーノードは、高可用性クラスタ内のサーバー間の通信にプライベートホスト名を使用します。この例では、Solaris Cluster ソフトウェアによって割り当てられたデフォルトの名前であるプライベートホスト名 *clusternode1-priv* および *clusternode2-priv* を使用します。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv,
qfs1mds-node2  clusternode2-priv,
```

6. 各行の 2 番目の列にあるコンマのあとに、アクティブなメタデータサーバーの仮想パブリックホスト名と、その後に空白文字を続けて入力します。

   HA-COTC サーバーノードは、パブリックデータネットワークを使用して、すべてがクラスタの外部にあるクライアントと通信します。アクティブなメタデータサーバーの IP アドレスとホスト名はフェイルオーバー中に変わる (*qfs1mds-node1* から *qfs1mds-node2*、およびその逆) ため、両方に仮想ホスト名 *qfs1mds* を使用します。あとで、*qfs1mds* の要求をアクティブなメタデータサーバーに常にルーティングするように Solaris Cluster ソフトウェアを構成します。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds
qfs1mds-node2  clusternode2-priv,qfs1mds
```

7. 各行の 3 番目の列に、サーバーの番号 (アクティブなメタデータサーバーの場合は *1*、潜在的なメタデータサーバーの場合は *2*) とその後にスペースを続けて入力します。

   この例では、メタデータサーバーは 1 つしかなく、プライマリノード *qfs1mds-node1* がアクティブなメタデータサーバーであるため番号は *1* で、セカンダリノード *qfs1mds-node2* の番号は **2** です。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds      1
qfs1mds-node2  clusternode2-priv,qfs1mds      2
```

8. 各行の 4 番目の列に、*0* (ゼロ) とその後に空白文字を続けて入力します。

4 列目の *0* (ゼロ)、- (ハイフン)、または空白値は、ホストが「*on*」(共有ファイルシステムへのアクセスありで構成) であることを示します。*1* (数字の 1) は、ホストが「*off*」(ファイルシステムへのアクセスなしで構成) であることを示します (共有ファイルシステムを管理する際のこれらの値の使用については、*samsharefs* のマニュアルページを参照してください)。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds  1        0
qfs1mds-node2  clusternode2-priv,qfs1mds  2        0
```

9. プライマリノードの行の 5 番目の列に、キーワード *server* を入力します。

   server キーワードは、デフォルトのアクティブなメタデータサーバーを示します。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds  1        0    server
qfs1mds-node2  clusternode2-priv,qfs1mds  2        0
```

10. クライアントホストごとに 1 行追加して、*Server Ordinal* 値を *0* に設定します。その後、ファイルを保存してエディタを閉じます。

    サーバー番号 *0* は、サーバーではなくクライアントとしてのホストを示します。HA-COTC クライアントはクラスタのメンバーではないため、クラスタのパブリックデータネットワークを介してのみ通信します。パブリック IP アドレスのみが指定されています。この例では、2 つのクライアント *qfs1client1* および *qfs1client2* を、ホスト名ではなくそのパブリック IP アドレス *172.16.0.133* および *172.16.0.147* を使用して追加します。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv,qfs1mds  1        0    server
qfs1mds-node2  clusternode2-priv,qfs1mds  2        0
qfs1client1    172.16.0.133               0        0
```

```
qfs1client2    172.16.0.147                    0         0
:wq
[qfs1mds-node1]root@solaris:~#
```

11. グローバル */etc/opt/SUNWsamfs/hosts.family-set-name* ファイルのコピーを潜在的な QFS メタデータサーバー (2 番目の HA-COTC クラスタノード) に配置します。
12. 次に、QFS サーバーおよび HA-COTC クラスタの外部にあるクライアントでのローカル hosts ファイルの作成を実行します。

9.3.2. QFS サーバーおよび HA-COTC クラスタの外部にあるクライアントでのローカル hosts ファイルの作成

クラスタの外部にあるクライアントとファイルシステムを共有する高可用性構成では、クライアントが、Solaris Cluster ソフトウェアによって定義されたパブリックデータネットワークを使用してファイルシステムサーバーのみと通信するようにする必要があります。これを行うには、特別に構成された QFS ローカル hosts ファイルを使用して、クライアントと、サーバー上の複数のネットワークインタフェース間のネットワークトラフィックを選択的にルーティングします。

それぞれのファイルシステムホストは、メタデータサーバー上の */etc/opt/SUNWsamfs/hosts.family-set-name* ファイルを最初にチェックすることで、ほかのホストのネットワークインタフェースを識別します。次に、個別の */etc/opt/SUNWsamfs/hosts.family-set-name.local* ファイルを確認します。ローカル hosts ファイルが存在しない場合、ホストはグローバル hosts ファイルに指定されたインタフェースアドレスをグローバルファイルに指定された順序で使用します。ただし、ローカル hosts ファイルが存在する場合、ホストはグローバルファイルと比較して、両方のファイルに一覧表示されたインタフェースのみをローカルファイルに指定された順序で使用します。各ファイルでさまざまな配列のさまざまなアドレスを使用すると、さまざまなホストで使用されているインタフェースを制御できます。

ローカル hosts ファイルを構成するには、次に概要を示す手順を使用します。

1. HA-COTC クラスタのプライマリノードに *root* としてログインします。

   この例では、ホストは *qfs1mds-node1* および *qfs1mds-node2* です。ホスト *qfs1mds-node1* にログインします。

```
[qfs1mds-node1]root@solaris:~#
```

2. アクティブおよび潜在的なそれぞれのメタデータサーバーでローカル hosts ファイルを作成します。パスとファイル名 */etc/opt/SUNWsamfs/hosts.family-set-*

name.local を使用します。ここで *family-set-name* は、共有ファイルシステムの装置 ID です。*アクティブおよび潜在的なサーバーで使用するネットワーク用のインタフェースのみを含めてください。*

この例では、アクティブなメタデータサーバーと潜在的なメタデータサーバーが、プライベートネットワークを介して相互に通信し、クライアントとはパブリックネットワークを介して通信するようにします。そのため、アクティブなサーバーと潜在的なサーバーのローカル hosts ファイル *hosts.qfs1.local* には、アクティブなサーバーと潜在的なサーバーのクラスタのプライベートアドレスのみが表示されています。

```
[qfs1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1.local
# /etc/opt/SUNWsamfs/hosts.qfs1.local
#                                         Server   On/  Additional
#Host Name     Network Interface           Ordinal  Off  Parameters
#------------  --------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv           1        0    server
qfs1mds-node2  clusternode2-priv           2        0
qfs1client1    172.16.0.133                  0        0
qfs1client2    172.16.0.147                  0        0
:wq
[qfs1mds-node1]root@solaris:~# ssh root@qfs1mds-node2
Password:


[qfs1mds-node2]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1.local
# /etc/opt/SUNWsamfs/hosts.qfs1.local
#                                         Server   On/  Additional
#Host Name     Network Interface           Ordinal  Off  Parameters
#------------  --------------------------  -------  ---  ----------
qfs1mds-node1  clusternode1-priv           1        0    server
qfs1mds-node2  clusternode2-priv           2        0
qfs1client1    172.16.0.133                  0        0
qfs1client2    172.16.0.147                  0        0
:wq
[qfs1mds-node2]root@solaris:~# exit
[qfs1mds-node1]root@solaris:~#
```

3. テキストエディタを使用して、各クライアントでローカル hosts ファイルを作成します。パスとファイル名 */etc/opt/SUNWsamfs/hosts.family-set-name.local* を使用します。ここで *family-set-name* は、共有ファイルシステムの装置 ID です。*クライアントで使用するネットワーク用のインタフェースのみを含めてください。*その後、ファイルを保存してエディタを閉じます。

   この例では、*vi* エディタを使用します。クライアントがパブリックデータネットワークのみを介してサーバーのみと通信するようにします。そのため、ファイルには、アクティブなメタデータサーバー *qfs1mds* の仮想ホスト名のみが含まれています。Solaris Cluster ソフトウェアは、*qfs1mds* の要求をどちらかアクティブな方のサーバーノードにルーティングします。

```
[qfs1mds-node1]root@solaris:~# ssh root@qfsclient1
Password:
[qfs1client1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1.local
# /etc/opt/SUNWsamfs/hosts.qfs1.local
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  ---------------------------  -------  ---  ----------
qfs1mds        qfs1mds                      1        0    server
:wq
qfs1client1]root@solaris:~# exit
[qfs1mds-node1]root@solaris:~# ssh root@qfs1client2
Password:
[qfs1client2]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1.local
# /etc/opt/SUNWsamfs/hosts.qfs1.local
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  ---------------------------  -------  ---  ----------
qfs1mds        qfs1mds                      1        0    server
:wq
[qfs1client2]root@solaris:~# exit
[qfs1mds-node1]root@solaris:~#
```

4. 次に、プライマリ HA-COTC クラスタノード上でのアクティブな QFS メタデータサーバーの構成を実行します。

9.3.3. プライマリ HA-COTC クラスタノード上でのアクティブな QFS メタデータサーバーの構成

アクティブなメタデータサーバーを構成するには、次のタスクを実行します。

- プライマリ HA-COTC ノード上での高パフォーマンス QFS ファイルシステムの作成
- クラスタ制御からのデータデバイスの除外
- プライマリ HA-COTC ノード上での QFS ファイルシステムのマウント.

9.3.3.1. プライマリ HA-COTC ノード上での高パフォーマンス QFS ファイルシステムの作成

1. HA-COTC クラスタのプライマリノードと QFS 共有ファイルシステムのアクティブなメタデータサーバーの両方としての役割を果たすクラスタノードを選択します。*root* としてログインします。

   この例では、*qfs1mds-node1* がプライマリノードおよびアクティブなメタデータサーバーです。

```
[qfs1mds-node1]root@solaris:~#
```

2. QFS ファイルシステムに使用されるグローバルストレージデバイスを選択します。Solaris Cluster コマンド */usr/global/bin/cldevice list -v* を使用します。

   Solaris Cluster ソフトウェアは、クラスタノードに接続されているすべてのデバイスに一意のデバイス ID (DID) を割り当てます。*グローバルデバイス*は、クラスタ内のすべてのノードからアクセスできるのに対して、*ローカルデバイス*は、ローカルデバイスをマウントするホストからのみアクセス可能です。グローバルデバイスはフェイルオーバー後にアクセス可能な状態のままになります。ローカルデバイスはそうではありません。

   この例では、デバイス *d1*、*d2*、*d7*、および *d8* は両方のノードからアクセス可能ではありません。そのため、高可用性 QFS 共有ファイルシステムの構成時にデバイス *d3*、*d4*、および *d5* の中から選択します。

```
[qfs1mds-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  qfs1mds-node1:/dev/rdsk/c0t0d0
d2                  qfs1mds-node1:/dev/rdsk/c0t6d0
d3                  qfs1mds-node1:/dev/rdsk/c1t1d0
```

```
d3                  qfs1mds-node2:/dev/rdsk/c1t1d0
d4                  qfs1mds-node1:/dev/rdsk/c1t2d0
d4                  qfs1mds-node2:/dev/rdsk/c1t2d0
d5                  qfs1mds-node1:/dev/rdsk/c1t3d0
d5                  qfs1mds-node2:/dev/rdsk/c1t3d0
d6                  qfs1mds-node2:/dev/rdsk/c0t0d0
d7                  qfs1mds-node2:/dev/rdsk/c0t1d0
```

3. 選択したプライマリノードで、*md* または *mr* データデバイスを使用する高パフォーマンス *ma* ファイルシステムを作成します。テキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開きます。

   この例では、ファイルシステム *qfs1* を構成します。デバイス *d3* をメタデータデバイス (装置タイプ *mm*) として構成して、*d4* および *d5* をデータデバイス (装置タイプ *mr*) として使用します。

```
[qfs1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#-----------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1     -
 /dev/did/dsk/d3s0    101        mm         qfs1     -
 /dev/did/dsk/d4s0    102        mr         qfs1     -
 /dev/did/dsk/d5s1    103        mr         qfs1     -
```

4. */etc/opt/SUNWsamfs/mcf* ファイルで、ファイルシステムエントリの *Additional Parameters* 列に *shared* パラメータを入力します。ファイルを保存します。

```
[qfs1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#-----------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1     -        shared
 /dev/did/dsk/d3s0     101        mm         qfs1     -
 /dev/did/dsk/d4s0     102        mr         qfs1     -
 /dev/did/dsk/d5s1     103        mr         qfs1     -
:wq
```

```
[qfs1mds-node1]root@solaris:~#
```

5. エラーを *mcf* ファイルで確認します。コマンド */opt/SUNWsamfs/sbin/sam-fsd* を使用して、見つかったエラーを修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、ホスト *qfs1mds-node1* で *mcf* ファイルを確認します。

```
[qfs1mds-node1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1mds-node1]root@solaris:~#
```

6. ファイルシステムを作成します。コマンド */opt/SUNWsamfs/sbin/sammkfs -S family-set-name* を使用します。ここで *family-set-name* は、ファイルシステムの装置 ID です。

   sammkfs コマンドは、プライマリノード *qfs1mds-node1* 上の *hosts.family-set-name* および *mcf* ファイルを読み取り、指定されたプロパティーを使用して共有ファイルシステムを作成します。

```
[qfs1mds-node1]root@solaris:~# sammkfs -S qfs1
Building 'qfs1' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[qfs1mds-node1]root@solaris:~#
```

7. 次に、クラスタ制御からのデータデバイスの除外を実行します。

9.3.3.2. クラスタ制御からのデータデバイスの除外

デフォルトでは、Solaris Cluster ソフトウェアは、クラスタの排他的使用のためにディスクデバイスを隔離します。ただし、HA-COTC 構成では、メタデータ (*mm*) デバイスのみがクラスタの一部です。データ (*mr*) デバイスは、クラスタの外部にあるファイルシステムクライアントと共

有され、クライアントホストに直接接続されています。そのため、データ (*mr*) デバイスは、クラスタソフトウェアの制御の範囲外にする必要があります。これは、次の 2 つの方法のいずれかで実現できます。

- HA-COTC クラスタ内の QFS データデバイスのフェンシングの無効化、または
- HA-COTC クラスタ上のローカル専用デバイスグループに共有データデバイスを配置.

9.3.3.2.1. HA-COTC クラスタ内の QFS データデバイスのフェンシングの無効化

1. HA-COTC クラスタのプライマリノードおよび QFS 共有ファイルシステムのアクティブなメタデータサーバーにログインします。*root* としてログインします。

   この例では、*qfs1mds-node1* がプライマリノードおよびアクティブなメタデータサーバーです。

```
[qfs1mds-node1]root@solaris:~#
```

2. */etc/opt/SUNWsamfs/mcf* ファイルで定義されているデータ (*mr*) デバイスごとに、フェンシングを無効にします。コマンド *cldevice set -p default_fencing=nofencing-noscrub device-identifier* を使用します。ここで *device-identifier* は、*mcf* ファイルの最初の列に表示されているデバイスのデバイス ID です。

   メタデータ (*mm*) デバイスのフェンシングを無効にしないでください。HA-COTC 構成では、QFS メタデータ (*mm*) デバイスはクラスタの一部ですが、QFS 共有データ (*mr*) デバイスはそうではありません。データデバイスは、クラスタの外部にあるクライアントに直接接続されています。このため、HA-COTC データ (*mr*) デバイスは、Solaris Cluster ソフトウェアによって管理されないローカルデバイスとして管理する必要があります。それ以外の場合、Solaris Cluster ソフトウェアと QFS は相反する目的で動作し、データが破損する可能性があります。

   上の例では、デバイス *d4* および *d5* をファイルシステム *qfs1* のデータデバイスとして構成しました。そのため、これらのデバイスのフェンシングをグローバルに無効にします (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs1mds-node1]root@solaris:~# cldevice set -p /
default_fencing=nofencing-noscrub d4
[qfs1mds-node1]root@solaris:~# cldevice set -p /
```

```
default_fencing=nofencing-noscrub d5
```

3. 次に、プライマリ HA-COTC ノード上での QFS ファイルシステムのマウントを実行します。

9.3.3.2.2. HA-COTC クラスタ上のローカル専用デバイスグループに共有データデバイスを配置

1. HA-COTC クラスタのプライマリノードおよび QFS 共有ファイルシステムのアクティブなメタデータサーバーにログインします。*root* としてログインします。

   この例では、*qfs1mds-node1* がプライマリノードおよびアクティブなメタデータサーバーです。

```
[qfs1mds-node1]root@solaris:~#
```

2. ファイルシステムの一部であるすべてのデータ (*mr*) デバイスを *localonly* デバイスグループに配置します。コマンド *cldevicegroup set -d device-identifier-list -p localonly=true -n active-mds-node device-group* を使用します。ここで、*device-list* は、デバイス ID のコンマ区切りリスト、*active-mds-node* は、アクティブなメタデータサーバーが通常常駐しているプライマリノード、*device-group* は、デバイスグループに選択した名前です。

   次の例では、データデバイス *d4* および *d5* (*mcf* 装置番号 *102* と *103*) をプライマリノード上のローカルデバイスグループ *mdsdevgrp* に配置します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs1mds-node1]root@solaris:~# cldevicegroup set -d d4,d5 -p localonly=true /
-n node1mds mdsdevgrp
[qfs1mds-node1]root@solaris:~#
```

3. 次に、プライマリ HA-COTC ノード上での QFS ファイルシステムのマウントを実行します。

9.3.3.3. プライマリ HA-COTC ノード上での QFS ファイルシステムのマウント

1. HA-COTC クラスタのプライマリノードおよび QFS 共有ファイルシステムのアクティブなメタデータサーバーにログインします。*root* としてログインします。

この例では、*qfs1mds-node1* がプライマリノードおよびアクティブなメタデータサーバーです。

```
[qfs1mds-node1]root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
[qfs1mds-node1]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. テキストエディタでオペレーティングシステムの */etc/vfstab* ファイルを開き、新しいファイルシステムの行を開始します。最初の列 (「*Device to Mount*」) にファイルシステム名を入力して、その後に 1 つ以上の空白文字を続けて入力します。

   この例では、*vi* テキストエディタを使用します。*qfs1* ファイルシステムの行を開始します。

```
[qfs1mds-node1]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount         System  fsck  Mount    Mount
#to Mount  to fsck  Point         Type    Pass  at Boot  Options
#--------  -------  ------------  ------  ----  -------  ---------------------
/devices   -        /devices      devfs   -     no       -
/proc      -        /proc         proc    -     no       -
...
qfs1       -
```

4. */etc/vfstab* ファイルの 2 番目の列 (「*Device to fsck*」) に、ハイフン (-) とその後に 1 つ以上の空白文字を続けて入力します。

   ハイフンは、ファイルシステムの整合性チェックをスキップすることをオペレーティングに示します。これらのチェックは、SAMFS ファイルシステムではなく UFS を対象としています。

```
[qfs1mds-node1]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount         System  fsck  Mount    Mount
#to Mount  to fsck  Point         Type    Pass  at Boot  Options
#--------  -------  ------------  ------  ----  -------  ---------------------
```

```
/devices   -        /devices      devfs   -     no       -
/proc      -        /proc         proc    -     no       -
...
qfs1       -
```

5. */etc/vfstab* ファイルの 3 番目の列に、クラスタを基準とした相対的なファイルシステムのマウントポイントを入力します。システムのルートディレクトリの直下にはないサブディレクトリを選択します。

   QFS 共有ファイルシステムをルートの直下にマウントすると、*SUNW.qfs* リソースタイプの使用時にフェイルオーバーの問題が発生する可能性があります。この例では、クラスタ上のマウントポイントを */global/ha-cotc/qfs1* に設定します。

```
#File
#Device    Device   Mount                 System  fsck  Mount    Mount
#to Mount  to fsck  Point                 Type    Pass  at Boot  Options
#--------  -------  --------------------  ------  ----  -------  --------------
/devices   -        /devices              devfs   -     no       -
/proc      -        /proc                 proc    -     no       -
...
qfs1       -        /global/ha-cotc/qfs1
```

6. QFS 共有ファイルシステムの場合と同様に、*/etc/vfstab* ファイルレコードの残りのフィールドに入力します。次に、ファイルを保存し、エディタを閉じます。

```
#File
#Device    Device   Mount                 System  fsck  Mount    Mount
#to Mount  to fsck  Point                 Type    Pass  at Boot  Options
#--------  -------  --------------------  ------  ----  -------  --------------
/devices   -        /devices              devfs   -     no       -
/proc      -        /proc                 proc    -     no       -
...
qfs1       -        /global/ha-cotc/qfs1  samfs   -     no       shared
:wq
[qfs1mds-node1]root@solaris:~#
```

7. 高可用性共有ファイルシステムのマウントポイントを作成します。

-p (*親*) オプションを指定して *mkdir* コマンドを使用すると、*/global* ディレクトリが作成されます (まだ存在しない場合)。

```
[qfs1mds-node1]root@solaris:~# mkdir -p /global/ha-cotc/qfs1
```

8. プライマリノードで高可用性共有ファイルシステムをマウントします。

```
[qfs1mds-node1]root@solaris:~# mount /global/ha-cotc/qfs1
```

9. 次に、セカンダリ HA-COTC クラスタノード上での潜在的な QFS メタデータサーバーの構成 を実行します。

9.3.4. セカンダリ HA-COTC クラスタノード上での潜在的な QFS メタデータサーバーの構成

2 ノードクラスタのセカンダリノードは、潜在的なメタデータサーバーとして機能します。潜在的なメタデータサーバーは、メタデータデバイスにアクセスできるホストであるため、メタデータサーバーの役割を担うことができます。そのため、プライマリノード上のアクティブなメタデータサーバーで障害が発生した場合、Solaris Cluster ソフトウェアはセカンダリノードにフェイルオーバーし、潜在的なメタデータサーバーをアクティブにできます。潜在的なメタデータサーバーを構成するには、次のタスクを実行します。

- セカンダリ HA-COTC ノード上での高パフォーマンス QFS ファイルシステムの作成
- セカンダリ HA-COTC ノードでの QFS ファイルシステムのマウント.

9.3.4.1. セカンダリ HA-COTC ノード上での高パフォーマンス QFS ファイルシステムの作成

1. HA-COTC クラスタのセカンダリノードに *root* としてログインします。

   この例では、*qfs1mds-node2* がセカンダリノードおよび潜在的なメタデータサーバーです。

```
[qfs1mds-node2]root@solaris:~#
```

2. */etc/opt/SUNWsamfs/mcf* ファイルをプライマリノードからセカンダリノードにコピーします。
3. エラーを *mcf* ファイルで確認します。コマンド */opt/SUNWsamfs/sbin/sam-fsd* を使用して、見つかったエラーを修正します。

sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、ホスト *qfs1mds-node1* で *mcf* ファイルを確認します。

```
[qfs1mds-node2]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1mds-node2]root@solaris:~#
```

4. 次に、セカンダリ HA-COTC ノードでの QFS ファイルシステムのマウントを実行します。

9.3.4.2. セカンダリ HA-COTC ノードでの QFS ファイルシステムのマウント

1. HA-COTC クラスタのセカンダリノードに *root* としてログインします。

   この例では、*qfs1mds-node2* がセカンダリノードです。

```
[qfs1mds-node2]root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
[qfs1mds-node2]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. テキストエディタでオペレーティングシステムの */etc/vfstab* ファイルを開き、新しいファイルシステムの行を追加します。次に、ファイルを保存し、エディタを閉じます。

   この例では、*vi* エディタを使用します。

```
[qfs1mds-node2]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount                 System  fsck  Mount    Mount
#to Mount  to fsck  Point                 Type    Pass  at Boot  Options
#--------  -------  --------------------  ------  ----  -------  -------------
/devices   -        /devices              devfs   -     no       -
/proc      -        /proc                 proc    -     no       -
```

```
...
qfs1        -         /global/ha-cotc/qfs1  samfs    -      no        shared
:wq
[qfs1mds-node2]root@solaris:~#
```

4. セカンダリノードで高可用性共有ファイルシステムのマウントポイントを作成します。

```
[qfs1mds-node2]root@solaris:~# mkdir -p /global/ha-cotc/qfs1
[qfs1mds-node2]root@solaris:~#
```

5. セカンダリノードで高可用性共有ファイルシステムをマウントします。

```
[qfs1mds-node2]root@solaris:~# mount /global/ha-cotc/qfs1
[qfs1mds-node2]root@solaris:~#
```

6. 次に、HA-COTC メタデータサーバーのフェイルオーバーの構成を実行します。

9.3.5. HA-COTC メタデータサーバーのフェイルオーバーの構成

Solaris Cluster ソフトウェアによって管理されるクラスタ内に Oracle HSM 共有ファイルシステムをホストする場合、Oracle HSM software によって定義されるリソースタイプである *SUNW.qfs* クラスタリソースを作成することで、メタデータサーバーのフェイルオーバーを構成します (詳細は、*SUNW.qfs* のマニュアルページを参照)。HA-COTC 構成のリソースを作成して構成するには、次のように進めます。

1. HA-COTC クラスタのプライマリノードに *root* としてログインします。

   この例では、*qfs1mds-node1* がプライマリノードです。

```
[qfs1mds-node1]root@solaris:~#
```

2. Solaris Cluster ソフトウェアの QFS リソースタイプ *SUNW.qfs* を定義します。コマンド *clresourcetype register SUNW.qfs* を使用します。

```
[qfs1mds-node1]root@solaris:~# clresourcetype register SUNW.qfs
[qfs1mds-node1]root@solaris:~#
```

3. 登録ファイルが見つからないために登録が失敗した場合、Solaris Cluster がリソースタイプの登録ファイル */opt/cluster/lib/rgm/rtreg/* を保持するディレクトリに、*/opt/SUNWsamfs/sc/etc/* ディレクトリへのシンボリックリンクを配置します。

   Oracle HSM ソフトウェアのインストール前に、Oracle Solaris Cluster ソフトウェアをインストールしませんでした。通常、Oracle HSM は、インストール時に Solaris Cluster を検出すると、*SUNW.qfs* 登録ファイルの場所を自動的に指定します。そのため、リンクを手動で作成する必要があります。

```
[qfs1mds-node1]root@solaris:~# cd /opt/cluster/lib/rgm/rtreg/
[qfs1mds-node1]root@solaris:~# ln -s /opt/SUNWsamfs/sc/etc/SUNW.qfs SUNW.qfs
[qfs1mds-node1]root@solaris:~#
```

4. QFS メタデータサーバーのリソースグループを作成します。Solaris Cluster コマンド *clresourcegroup create -n node-list group-name* を使用します。ここで *node-list* は、2 つのクラスタノード名のコンマ区切りリストで、*group-name* はリソースグループに使用する名前です。

   この例では、HA-COTC サーバーノードをメンバーとして使用してリソースグループ *qfsrg* を作成します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs1mds-node1]root@solaris:~# clresourcegroup create -n / qfs1mds-node1,qfs1mds-node2 qfsrg
[qfs1mds-node1]root@solaris:~#
```

5. 新しいリソースグループで、アクティブなメタデータサーバーの仮想ホスト名を設定します。Solaris Cluster コマンド *clreslogicalhostname create -g group-name virtualMDS* を使用します。ここで *group-name* は QFS リソースグループの名前で、*virtualMDS* は仮想ホスト名です。

   共有ファイルシステムの hosts ファイルで使用したものと同じ仮想ホスト名を使用します。この例では、仮想ホスト *qfs1mds* を *qfsr* リソースグループ内に作成します。

```
[qfs1mds-node1]root@solaris:~# clreslogicalhostname create -g qfsrg qfs1mds
```

6. QFS ファイルシステムリソースをリソースグループに追加します。コマンド *clresource create -g group-name -t SUNW.qfs -x QFSFileSystem=mount-point -y Resource_dependencies=virtualMDS resource-name* を使用します。ここでは:

- *group-name* は、QFS リソースグループの名前です。
- *mount-point* は、クラスタ内のファイルシステムのマウントポイントである、システムのルートディレクトリの直下にはないサブディレクトリです。

  QFS 共有ファイルシステムをルートの直下にマウントすると、*SUNW.qfs* リソースタイプの使用時にフェイルオーバーの問題が発生する可能性があります。
- *virtualMDS* は、アクティブなメタデータサーバーの仮想ホスト名です。
- *resource-name* は、リソースに指定する名前です。

この例では、リソースグループ *qfsrg* にタイプ *SUNW.qfs* の *hasqfs* という名前のリソースを作成します。*SUNW.qfs* 拡張プロパティー *QFSFileSystem* を */global/ha-cotc/qfs1* マウントポイントに設定して、標準プロパティー *Resource_dependencies* をアクティブなメタデータサーバー *qfs1mds* の論理ホストに設定します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs1mds-node1]root@solaris:~# clresource create -g qfsrg -t SUNW.qfs /
-x QFSFileSystem=/global/ha-cotc/qfs1 -y Resource_dependencies=qfs1mds hasqfs
```

7. リソースグループをオンラインにします。コマンド *clresourcegroup online -emM group-name* を使用します。ここで *group-name* は、QFS リソースグループの名前です。

   この例では、*qfsr* リソースグループをオンラインにします。

```
[qfs1mds-node1]root@solaris:~# clresourcegroup manage qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup online -emM qfsrg
```

8. QFS リソースグループを必ずオンラインにしてください。Solaris Cluster *clresourcegroup status* コマンドを使用します。

   この例では、*qfsrg* リソースグループは、プライマリノード *sam1mds-node1* では *online* です。

```
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsrg       qfs1mds-node1   No          Online
            qfs1mds-node2   No          Offline
```

9. リソースグループをセカンダリノードに移動して、リソースグループが正しくフェイルオーバーするようにします。Solaris Cluster コマンド *clresourcegroup switch -n node2 group-name* を使用します。ここで *node2* はセカンダリノードの名前で、*group-name* は HA-QFS リソースグループに選択した名前です。次に、*clresourcegroup status* を使用して結果を確認します。

   この例では、*qfsrg* リソースグループを *qfs1mds-node2* に移動して、指定したノードでリソースグループがオンラインになることを確認します。

```
[qfs1mds-node1]root@solaris:~# clresourcegroup switch -n qfs1mds-node2 qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsrg       qfs1mds-node1   No          Offline
            qfs1mds-node2   No          Online
```

10. リソースグループをプライマリノードに戻します。Solaris Cluster コマンド *clresourcegroup switch -n node1 group-name* を使用します。ここで *node1* はプライマリノードの名前で、*group-name* は HA-QFS リソースグループに選択した名前です。次に、*clresourcegroup status* を使用して結果を確認します。

    この例では、*qfsrg* リソースグループを *qfs1mds-node1* に正常に戻します。

```
[qfs1mds-node1]root@solaris:~# clresourcegroup switch -n qfs1mds-node1 qfsrg
[qfs1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsrg       qfs1mds-node1   No          Online
            qfs1mds-node2   No          Offline
```

11. 次に、HA-COTC クラスタの外部にあるホストを QFS 共有ファイルシステムクライアントとして構成を実行します。

9.3.6. HA-COTC クラスタの外部にあるホストを QFS 共有ファイルシステムクライアントとして構成

クライアントがクラスタ内のメタデータサーバーの高可用性構成を妨げないように、各ホストを、ファイルシステムのメタデータデバイスにアクセスできない QFS クライアントとして構成します。

HA-COTC 共有ファイルシステムのクライアントごとに、次のように進めます。

1. HA-COTC クラスタ内のプライマリノードにログインします。*root* としてログインします。

```
[qfs1mds-node1]root@solaris:~#
```

2. クラスタのデバイス構成を表示します。Solaris Cluster コマンド */usr/global/bin/cldevice list -v* を使用します。

```
[qfs1mds-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  qfs1mds-node1:/dev/rdsk/c0t0d0
d2                  qfs1mds-node1:/dev/rdsk/c0t6d0
...
d7                  qfs1mds-node2:/dev/rdsk/c0t1d0
[qfs1mds-node1]root@solaris:~#
```

3. *cldevice list -v* コマンドの出力を調べます。各 QFS データ (*mr*) デバイスのデバイス ID に対応する */dev/rdsk/* パスをメモします。

   この例では、QFS データデバイスは *d4* および *d5* です。

```
[qfs1mds-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  qfs1mds-node1:/dev/rdsk/c0t0d0
d2                  qfs1mds-node1:/dev/rdsk/c0t6d0
d3                  qfs1mds-node1:/dev/rdsk/c1t1d0
d3                  qfs1mds-node2:/dev/rdsk/c1t1d0
d4                  qfs1mds-node1:/dev/rdsk/c1t2d0
d4                  qfs1mds-node2:/dev/rdsk/c1t2d0
```

```
d5                   qfs1mds-node1:/dev/rdsk/c1t3d0
d5                   qfs1mds-node2:/dev/rdsk/c1t3d0
d6                   qfs1mds-node2:/dev/rdsk/c0t0d0
d7                   qfs1mds-node2:/dev/rdsk/c0t1d0
[qfs1mds-node1]root@solaris:~#
```

4. HA-COTC クラスタのクライアントホストに *root* としてログインします。

   この例では、*qfs1client1* がクライアントホストです。

```
[qfs1mds-node1]root@solaris:~# ssh root@qfs1client1
[qfs1client1]root@solaris:~#
```

5. クライアントホスト上で、共有ファイルシステムの構成情報を取得します。*samfsconfig /dev/rdsk/** コマンドを使用します。

   *samfsconfig /dev/rdsk/** コマンドは、指定されたパスで、QFS ファイルシステムに属する接続済みデバイスを検索します。この例では、このコマンドは、*qfs1* データ (*mr*) デバイスのパスを検索します。想定どおり、メタデータ (*mm*) デバイスが見つからないため、共有データデバイスが一覧表示される前に、「*Missing slices*」および「*Ordinal 0*」メッセージが返されます。

```
[qfs1client1]root@solaris:~# samfsconfig /dev/rdsk/*
# Family Set 'qfs1' Created Thu Dec 21 07:17:00 2013
# Missing slices
# Ordinal 0
# /dev/rdsk/c1t2d0s0   102         mr       qfs1   -
# /dev/rdsk/c1t3d0s1   103         mr       qfs1   -
```

6. *samfsconfig* コマンドの出力を、サーバー上の Solaris Cluster *cldevice list* コマンドの出力と比較します。両方で *mr* データデバイスの同じデバイスパスが報告されることを確認します。

   コントローラ番号 (*cN*) は異なる可能性がありますが、*samfsconfig* および *cldevice list* コマンドは、同じデバイスを示します。この例では、*samfsconfig* および *cldevice list* コマンドは、同じデバイスを指しています。

メタデータサーバーノードでは、*/etc/opt/SUNWsamfs/mcf* ファイルは、クラスタデバイス ID *d4* および *d5* を使用して、共有の *mr* データデバイス *102* および *103* を識別します。

```
/dev/did/dsk/d4s0     102        mr         qfs1  -
/dev/did/dsk/d5s1     103        mr         qfs1  -
```

メタデータサーバーノードで *cldevice list* コマンドを使用すると、クラスタデバイス ID *d4* および *d5* がパス */dev/rdisk/c1t2d0* および */dev/rdisk/c1t3d0* にマップされます。

```
d4                    qfs1mds-node1:/dev/rdsk/c1t2d0
d5                    qfs1mds-node1:/dev/rdsk/c1t3d0
```

クライアントノードでは、*samfsconfig* コマンドは、パス */dev/rdisk/c1t2d0* および */dev/rdisk/c1t3d0* を持つ共有の *mr* データデバイス *102* および *103* も識別します。

```
/dev/rdsk/c1t2d0s0    102        mr       qfs1  -
/dev/rdsk/c1t3d0s1    103        mr       qfs1  -
```

7. テキストエディタでクライアントの */etc/opt/SUNWsamfs/mcf* ファイルを開きます。HA-COTC 共有ファイルシステムのエントリを追加します。このエントリは、メタデータサーバー *mcf* ファイル内の対応するエントリと正確に一致するべきです。

   この例では、*vi* エディタを使用して、QFS 共有ファイルシステム *qfs1* (装置番号 *100*) のエントリを作成します。

```
[qfs1client1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1     -        shared
```

8. 新しい行で、HA-COTC 共有ファイルシステムのメタデータ (*mm*) デバイスのエントリを開始します。最初の列 (「*Equipment Identifier*」) に、キーワード *nodev* を入力します。

```
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1      -       shared
 nodev
```

9. HA-COTC ファイルシステムのメタデータ (*mm*) デバイスの残りのフィールドに、メタデータサーバー *mcf* ファイルで使用される同じ装置番号、ファミリセット、およびデバイス状態パラメータを入力します。

```
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   ----------------
qfs1                  100        ma         qfs1      -       shared
 nodev                 101        mm         qfs1     -
```

10. データ (*mr*) デバイスの完全なエントリを *samfsconfig* 出力からコピーします。クライアントの */etc/opt/SUNWsamfs/mcf* ファイルにエントリを貼り付けます。*samfsconfig* によって挿入される先頭のコメント (*#*) 記号を削除します。次に、ファイルを保存し、エディタを閉じます。

```
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   -----------------
qfs1                  100        ma         qfs1      -       shared
 nodev                 101        mm         qfs1     -
 /dev/rdsk/c1t2d0s0    102        mr         qfs1     -
 /dev/rdsk/c1t3d0s1    103        mr         qfs1     -
:wq
[qfs1client1]root@solaris:~#
```

11. エラーを *mcf* ファイルで確認します。コマンド */opt/SUNWsamfs/sbin/sam-fsd* を使用して、見つかったエラーを修正します。

    sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、ホスト *qfs1client1* で *mcf* ファイルを確認します。

```
[qfs1client1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1client1]root@solaris:~#
```

12. クライアントオペレーティングシステムの */etc/vfstab* ファイルをテキストエディタで開いて、サーバーで使用される同じパラメータを使用して新しいファイルシステムのエントリを追加します。次に、ファイルを保存し、エディタを閉じます。

    この例では、*vi* エディタを使用します。

```
[qfs1client1]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount                 System  fsck  Mount    Mount
#to Mount  to fsck  Point                 Type    Pass  at Boot  Options
#--------  -------  --------------------  ------  ----  -------  -------------
/devices   -        /devices              devfs   -     no       -
/proc      -        /proc                 proc    -     no       -
...
qfs1       -        /global/ha-cotc/qfs1  samfs   -     no       shared
:wq
[qfs1client1]root@solaris:~#
```

13. クライアントで、高可用性共有ファイルシステムのマウントポイントを作成します。

```
[qfs1client1]root@solaris:~# mkdir -p /global/qfs1
[qfs1client1]root@solaris:~#
```

14. クライアントで、高可用性共有ファイルシステムをマウントします。

    例

```
[qfs1client1]root@solaris:~# mount /global/qfs1
[qfs1client1]root@solaris:~#
```

15. すべての HA-COTC クライアントが構成されるまでこの手順を繰り返します。
16. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
17. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

9.4. 高可用性 Oracle HSM 共有アーカイブファイルシステム

高可用性 Oracle Hierarchical Storage Manager (HA-SAM) 構成では、サーバーホストで障害が発生した場合でも QFS メタデータサーバーと Oracle Hierarchical Storage Manager アプリケーションが動作を継続できるようにすることにより、アーカイブファイルシステムの可用性が維持されます。ファイルシステムは、Solaris Cluster ソフトウェアによって管理される 2 ノードクラスタにホストされた、アクティブおよび潜在的な QFS メタデータサーバーの間で共有されます。アクティブなクラスタノードで障害が発生した場合、クラスタリングソフトウェアは、残っているノードで潜在的な Oracle HSM サーバーをアクティブにして、実行中の操作に対する制御を渡します。QFS ファイルシステムと Oracle HSM アプリケーションのローカルストレージディレクトリは共有され、すでにマウントされているため、データとメタデータへのアクセスは中断されません。

HA-SAM 構成では、アクティブなメタデータサーバーを介してすべての入出力を送信することで、クラスタ環境でのファイルシステムの整合性が確保されます。HA-SAM ファイルシステムはアクセシビリティーの理由でのみ共有します。潜在的なメタデータサーバーホストは、ほかの SAM-QFS 共有ファイルシステム構成の場合のようにファイルシステムクライアントとしては使用できません。潜在的なメタデータサーバーは、ノードのフェイルオーバー中にアクティブにしないかぎり入出力を実行しません。HA-SAM ファイルシステムは NFS を使用してクライアントと共有できます。ただし、共有がアクティブなメタデータサーバーノードから排他的にエクスポートされるようにする必要があります。

高可用性アーカイブファイルシステムは、3 つの Solaris Cluster リソースタイプに依存します。

- *SUNW.hasam*

  プライマリホストで障害が発生した場合、*SUNW.hasam* リソースは Oracle Hierarchical Storage Manager アプリケーションのフェイルオーバーを管理します。*SUNW.hasam* ソフトウェアは、Oracle HSM software ディストリビューションに付属しています。

- *SUNW.qfs*

プライマリホストで障害が発生した場合、*SUNW.qfs* リソースは、QFS メタデータサーバーのフェイルオーバーを管理します。*SUNW.qfs* ソフトウェアは、Oracle HSM software ディストリビューションに付属しています (詳細は、*SUNW.qfs* のマニュアルページを参照)。

- *SUNW.HAStoragePlus*

  プライマリホストで障害が発生した場合、*SUNW.HAStoragePlus* リソースは、Oracle Hierarchical Storage Manager のローカルストレージのフェイルオーバーを管理します。Oracle HSM アプリケーションは、変わりやすいアーカイブ情報 (ジョブキューやリムーバブルメディアカタログ) をサーバーホストのローカルファイルシステムに保持します。*SUNW.HAStoragePlus* は、標準のリソースタイプとして Solaris Cluster ソフトウェアに含まれています (リソースタイプの詳細は、*Oracle Solaris Cluster* ドキュメントライブラリで『*データサービス計画および管理*』ドキュメントを参照してください)。

必要なコンポーネントのインスタンスを構成して、動作中の HA-SAM アーカイブ構成に統合するには、次のタスクを実行します。

- 両方の HA-SAM クラスタノードでのグローバル hosts ファイルの作成
- 両方の HA-SAM クラスタノードでのローカル hosts ファイルの作成
- プライマリ HA-SAM クラスタノード上でのアクティブな QFS メタデータサーバーの構成
- セカンダリ HA-SAM クラスタノード上での潜在的な QFS メタデータサーバーの構成
- Oracle HSM 構成ファイルの高可用性ローカルファイルシステムの構成
- 高可用性ローカルファイルシステムへの Oracle HSM 構成ファイルの再配置
- 高可用性ローカルファイルシステムを使用するための HA-SAM クラスタの構成
- QFS ファイルシステムメタデータサーバーのフェイルオーバーの構成
- Oracle Hierarchical Storage Manager アプリケーションのフェイルオーバーの構成
- HA-SAM ソリューションのクラスタリソースの依存関係の定義
- HA-SAM リソースグループのオンライン化および構成のテスト
- 必要に応じて、高可用性ネットワークファイルシステム (HA-NFS) 共有を構成します。

  HA-NFS を設定するための詳細な手順は、*Oracle Solaris Cluster* オンラインドキュメントライブラリに含まれている『*Oracle Solaris Cluster Data Service for Network File System (NFS) ガイド*』に記載されています。

9.4.1. 両方の HA-SAM クラスタノードでのグローバル hosts ファイルの作成

Oracle HSM アーカイブ共有ファイルシステムでは、両方のノードのホストがファイルシステムのメタデータにアクセスできるように、メタデータサーバーで hosts ファイルを構成する必要があります。hosts ファイルは、*/etc/opt/SUNWsamfs/* ディレクトリに *mcf* ファイルとともに格納されています。共有ファイルシステムの初期作成中に、*sammkfs -S* コマンドを実行すると、このファイルに格納されている設定を使用して共有が構成されます。ここで、次の手順を使用して作成します。

1. HA-SAM クラスタのプライマリノードに *root* としてログインします。

   この例では、*sam1mds-node1* がプライマリノードです。

```
[sam1mds-node1]root@solaris:~#
```

2. クラスタ構成を表示します。*/usr/global/bin/cluster show* コマンドを使用します。出力で、それぞれの *Node Name* のレコードを見つけて、各ネットワークアダプタの *privatehostname*、*Transport Adapter* の名前、および *ip_address* プロパティーをメモします。

   この例では、それぞれのノードには、*hme0* と *qfe3* の 2 つのネットワークインタフェースがあります。

   - *hme0* アダプタには、クラスタがノード間の内部通信に使用するプライベートネットワークの IP アドレスがあります。Solaris Cluster ソフトウェアは、各プライベートアドレスに対応する *privatehostname* を割り当てます。

     デフォルトでは、プライマリノードのプライベートホスト名は *clusternode1-priv* で、セカンダリノードのプライベートホスト名は *clusternode2-priv* です。
   - *qfe3* アダプタには、クラスタがデータ転送に使用するパブリック IP アドレスとパブリックホスト名である *sam1mds-node1* と *sam1mds-node2* があります。

   表示は、省略記号 (*...*) を使用して省略されています。

```
[sam1mds-node1]root@solaris:~# cluster show
...
  === Cluster Nodes ===
  Node Name:                                    sam1mds-node1...
    privatehostname:                               clusternode1-priv...
```

```
    Transport Adapter List:                        qfe3, hme0...
    Transport Adapter:                           qfe3...
      Adapter Property(ip_address):                172.16.0.12...
    Transport Adapter:                           hme0...
      Adapter Property(ip_address):                10.0.0.129...
  Node Name:                                     sam1mds-node2...
    privatehostname:                               clusternode2-priv...
    Transport Adapter List:                        qfe3, hme0...
      Adapter Property(ip_address):                172.16.0.13...
    Transport Adapter:                           hme0...
      Adapter Property(ip_address):                10.0.0.122
```

3. テキストエディタを使用して、ファイル */etc/opt/SUNWsamfs/hosts.family-set-name* を作成します。ここで、*family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルによってファイルシステム装置に割り当てられるファミリセット名です。

   この例では、*vi* テキストエディタを使用してファイル *hosts.sam1* を作成します。ホストテーブル内の列を示すために、コメントを示すハッシュ記号 (#) で各行を開始して、いくつかのオプションの見出しを追加します。

```
[sam1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sam1
# /etc/opt/SUNWsamfs/hosts.sam1
#                                          Server   On/  Additional
#Host Name     Network Interface             Ordinal  Off  Parameters
#------------  ----------------------------  -------  ---  ----------
```

4. テーブルの最初の列に、プライマリメタデータサーバーノードとセカンダリメタデータサーバーノードのホスト名のあとにいくつかの空白文字を付けて、各行に 1 つのエントリを入力します。

   hosts ファイルでは、行は行 (レコード) で、空白文字は列 (フィールド) 区切り文字です。この例では、最初の 2 行の「*Host Name*」列には、ファイルシステムのメタデータサーバーをホストするクラスタノードのホスト名である、値 *sam1mds-node1* と *sam1mds-node2* が含まれています。

```
#                                          Server   On/  Additional
#Host Name     Network Interface             Ordinal  Off  Parameters
#------------  ----------------------------  -------  ---  ----------
```

```
sam1mds-node1
sam1mds-node2
```

5. 各行の 2 番目の列では、*Host Name* 列に一覧表示されているホストの *Network Interface* 情報の指定を開始します。それぞれの HA-SAM クラスタノードの Solaris Cluster プライベートホスト名またはプライベートネットワークアドレスと、その後にコンマを続けて入力します。

   HA-SAM サーバーノードは、高可用性クラスタ内のサーバー間の通信にプライベートホスト名を使用します。この例では、Solaris Cluster ソフトウェアによって割り当てられたデフォルトの名前であるプライベートホスト名 *clusternode1-priv* および *clusternode2-priv* を使用します。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,
sam1mds-node2  clusternode2-priv,
```

6. 各行の 2 番目の列にあるコンマのあとに、アクティブなメタデータサーバーのパブリックホスト名と、その後に空白文字を続けて入力します。

   HA-SAM サーバーノードは、パブリックデータネットワークを使用して、クラスタの外部にあるホストと通信します。アクティブなメタデータサーバーの IP アドレスとホスト名はフェイルオーバー中に変わる (*sam1mds-node1* から *sam1mds-node2*、およびその逆) ため、両方に仮想ホスト名 *sam1mds* を使用します。あとで、*sam1mds* の要求をアクティブなメタデータサーバーに常にルーティングするように Solaris Cluster ソフトウェアを構成します。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,sam1mds
sam1mds-node2  clusternode2-priv,sam1mds
```

7. 各行の 3 番目の列に、サーバーの番号 (アクティブなメタデータサーバーの場合は *1*、潜在的なメタデータサーバーの場合は *2*) とその後にスペースを続けて入力します。

この例では、メタデータサーバーは 1 つしかなく、プライマリノード *sam1mds-node1* がアクティブなメタデータサーバーであるため番号は *1* で、セカンダリノード *sam1mds-node2* の番号は *2* です。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,sam1mds  1
sam1mds-node2  clusternode2-priv,sam1mds  2
```

8. 各行の 4 番目の列に、*0* (ゼロ) とその後に空白文字を続けて入力します。

   4 列目の *0*、- (ハイフン)、または空白値は、ホストが「*on*」(共有ファイルシステムへのアクセスありで構成) であることを示します。*1* (数字の 1) は、ホストが「*off*」(ファイルシステムへのアクセスなしで構成) であることを示します (共有ファイルシステムを管理する際のこれらの値の使用については、*samsharefs* のマニュアルページを参照してください)。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,sam1mds  1        0
sam1mds-node2  clusternode2-priv,sam1mds  2        0
```

9. プライマリノードの行の 5 番目の列に、キーワード *server* を入力します。その後、ファイルを保存してエディタを閉じます。

   server キーワードは、デフォルトのアクティブなメタデータサーバーを示します。

```
#                                       Server   On/  Additional
#Host Name     Network Interface         Ordinal  Off  Parameters
#------------  ------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv,sam1mds  1        0    server
sam1mds-node2  clusternode2-priv,sam1mds  2        0
:wq
[sam1mds-node1]root@solaris:~#
```

10. グローバル */etc/opt/SUNWsamfs/hosts.family-set-name* ファイルのコピーを潜在的なメタデータサーバーに格納します。
11. 次に、両方の HA-SAM クラスタノードでのローカル hosts ファイルの作成を実行します。

9.4.2. 両方の HA-SAM クラスタノードでのローカル hosts ファイルの作成

高可用性アーカイブ共有ファイルシステムでは、Solaris Cluster ソフトウェアによって定義されたプライベートネットワークを使用してサーバーが相互に通信するようにする必要があります。これを行うには、特別に構成されたローカル hosts ファイルを使用して、サーバー上のネットワークインタフェース間のネットワークトラフィックを選択的にルーティングします。

それぞれのファイルシステムホストは、メタデータサーバー上の */etc/opt/SUNWsamfs/hosts.family-set-name* ファイルを最初にチェックすることで、ほかのホストのネットワークインタフェースを識別します。次に、個別の */etc/opt/SUNWsamfs/hosts.family-set-name.local* ファイルを確認します。ローカル hosts ファイルが存在しない場合、ホストはグローバル hosts ファイルに指定されたインタフェースアドレスをグローバルファイルに指定された順序で使用します。ただし、ローカル hosts ファイルが存在する場合、ホストはグローバルファイルと比較して、両方のファイルに一覧表示されたインタフェースのみをローカルファイルに指定された順序で使用します。各ファイルでさまざまな配列のさまざまなアドレスを使用すると、さまざまなホストで使用されているインタフェースを制御できます。

ローカル hosts ファイルを構成するには、次に概要を示す手順を使用します。

1. HA-SAM クラスタのプライマリノードに *root* としてログインします。

   この例では、*sam1mds-node1* がプライマリノードです。

```
[sam1mds-node1]root@solaris:~#
```

2. テキストエディタで、パスとファイル名 */etc/opt/SUNWsamfs/hosts.family-set-name.local* を使用して、アクティブなメタデータサーバー上でローカル hosts ファイルを作成します。ここで、*family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルによってファイルシステム装置に割り当てられるファミリセット名です。*潜在的なサーバーとの通信時にアクティブなサーバーに使用させるネットワークインタフェースのみを含めます。*その後、ファイルを保存してエディタを閉じます。

   この例では、アクティブなメタデータサーバーと潜在的なメタデータサーバーがプライベートネットワークを介して相互に通信するようにします。そのため、アクティブなメタデー

タサーバーである *hosts.sam1.local* のローカル hosts ファイルには、アクティブなサーバーと潜在的なサーバーのクラスタのプライベートアドレスのみが表示されています。

```
[sam1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sam1.local
#                                        Server   On/  Additional
#Host Name     Network Interface          Ordinal  Off  Parameters
#------------  -------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv          1        0    server
sam1mds-node2  clusternode2-priv          2        0
:wq
[sam1mds-node1]root@solaris:~#
```

3. セカンダリクラスタノードに *root* としてログインします。

   この例では、*sam1mds-node2* がセカンダリノードです。

```
[sam1mds-node1]root@solaris:~# ssh root@sam1mds-node2
Password:
[sam1mds-node2]root@solaris:~#
```

4. テキストエディタを使用して、潜在的なメタデータサーバーでローカル hosts ファイルを作成します。パスとファイル名 */etc/opt/SUNWsamfs/hosts.family-set-name.local* を使用します。ここで、*family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルによってファイルシステム装置に割り当てられるファミリセット名です。アクティブなサーバーとの通信時に潜在的なサーバーに使用させるネットワークインタフェースのみを含めます。その後、ファイルを保存してエディタを閉じます。

   この例では、アクティブなメタデータサーバーと潜在的なメタデータサーバーがプライベートネットワークを介して相互に通信するようにします。そのため、潜在的なメタデータサーバーである *hosts.sam1.local* のローカル hosts ファイルには、アクティブなサーバーと潜在的なサーバーのクラスタのプライベートアドレスのみが表示されています。

```
[sam1mds-node2]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.sam1.local
#                                        Server   On/  Additional
#Host Name     Network Interface          Ordinal  Off  Parameters
#------------  -------------------------  -------  ---  ----------
sam1mds-node1  clusternode1-priv          1        0    server
```

```
sam1mds-node2  clusternode2-priv              2        0
:wq
[sam1mds-node2]root@solaris:~# exit
[sam1mds-node1]root@solaris:~#
```

5. 次に、プライマリ HA-SAM クラスタノード上でのアクティブな QFS メタデータサーバーの構成を実行します。

9.4.3. プライマリ HA-SAM クラスタノード上でのアクティブな QFS メタデータサーバーの構成

1. HA-SAM クラスタのプライマリノードと QFS 共有ファイルシステムのアクティブなメタデータサーバーの両方としての役割を果たすクラスタノードを選択します。*root* としてログインします。

   この例では、*sam1mds-node1* がプライマリノードです。

```
[sam1mds-node1]root@solaris:~#
```

2. QFS ファイルシステムに使用されるグローバルストレージデバイスを選択します。コマンド */usr/global/bin/cldevice list -v* を使用します。

   Solaris Cluster ソフトウェアは、クラスタノードに接続されているすべてのデバイスに一意のデバイス ID (DID) を割り当てます。*グローバル*デバイスは、クラスタ内のすべてのノードからアクセスできるのに対して、*ローカル*デバイスは、ローカルデバイスをマウントするホストからのみアクセス可能です。グローバルデバイスはフェイルオーバー後にアクセス可能な状態のままになります。ローカルデバイスはそうではありません。

   この例では、デバイス *d1*、*d2*、*d7*、および *d8* は両方のノードからアクセス可能ではありません。そのため、高可用性 QFS 共有ファイルシステムの構成時にデバイス *d3*、*d4*、および *d5* の中から選択します。

```
[sam1mds-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  sam1mds-node1:/dev/rdsk/c0t0d0
d2                  sam1mds-node1:/dev/rdsk/c0t6d0
d3                  sam1mds-node1:/dev/rdsk/c1t1d0
d3                  sam1mds-node2:/dev/rdsk/c1t1d0
```

```
d4                  sam1mds-node1:/dev/rdsk/c1t2d0
d4                  sam1mds-node2:/dev/rdsk/c1t2d0
d5                  sam1mds-node1:/dev/rdsk/c1t3d0
d5                  sam1mds-node2:/dev/rdsk/c1t3d0
d6                  sam1mds-node2:/dev/rdsk/c0t0d0
d7                  sam1mds-node2:/dev/rdsk/c0t1d0
```

3. 選択したプライマリノードで、*mr* データデバイスを使用する高パフォーマンスの *ma* ファイルシステムを作成します。テキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開きます。

   この例では、ファイルシステム *sam1* を構成します。デバイス *d3* をメタデータデバイス (装置タイプ *mm*) として構成して、*d4* および *d5* をデータデバイス (装置タイプ *mr*) として使用します。

```
[sam1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   -----------------
sam1                  100        ma         sam1     -
 /dev/did/dsk/d3s0    101        mm         sam1     -
 /dev/did/dsk/d4s0    102        mr         sam1     -
 /dev/did/dsk/d5s1    103        mr         sam1     -
```

4. */etc/opt/SUNWsamfs/mcf* ファイルで、ファイルシステムエントリの *Additional Parameters* 列に *shared* パラメータを入力します。ファイルを保存します。

```
[sam1mds-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#------------------   ---------  ---------  -------  ------   -----------------
sam1                  100        ma         sam1     -        shared
 /dev/did/dsk/d3s0     101        mm         sam1     -
 /dev/did/dsk/d4s0     102        mr         sam1     -
 /dev/did/dsk/d5s1     103        mr         sam1     -
:wq
[sam1mds-node1]root@solaris:~#
```

5. エラーを *mcf* ファイルで確認します。コマンド */opt/SUNWsamfs/sbin/sam-fsd* を使用して、見つかったエラーを修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、ホスト *sam1mds-node1* で *mcf* ファイルを確認します。

```
[sam1mds-node1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[sam1mds-node1]root@solaris:~#
```

6. ファイルシステムを作成します。コマンド */opt/SUNWsamfs/sbin/sammkfs -S family-set-name* を使用します。ここで、*family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルによってファイルシステム装置に割り当てられるファミリセット名です。

   sammkfs コマンドは、*hosts.family-set-name* および *mcf* ファイルを読み取って、指定されたプロパティーを使用して Oracle HSM ファイルシステムを作成します。

```
[sam1mds-node1]root@solaris:~# sammkfs -S sam1
Building 'sam1' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[sam1mds-node1]root@solaris:~#
```

7. テキストエディタでオペレーティングシステムの */etc/vfstab* ファイルを開き、新しいファイルシステムの行を開始します。最初の列にファイルシステム名、空白文字、2 番目の列にハイフン、さらに空白を入力します。

   この例では、*vi* テキストエディタを使用します。*sam1* ファイルシステムの行を開始します。ハイフンは、オペレーティングシステムが UFS ツールを使用してファイルシステムの整合性チェックを試行しないようにします。

```
[sam1mds-node1]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount         System  fsck  Mount    Mount
#to Mount  to fsck  Point         Type    Pass  at Boot  Options
#--------  -------  ------------  ------  ----  -------  ---------------------
/devices   -        /devices      devfs   -     no       -
/proc      -        /proc         proc    -     no       -
...
sam1       -
```

8. */etc/vfstab* ファイルの 3 番目の列に、クラスタを基準とした相対的なファイルシステムのマウントポイントを入力します。システムのルートディレクトリの直下にはないサブディレクトリを選択します。

   QFS 共有ファイルシステムをルートの直下にマウントすると、*SUNW.qfs* リソースタイプの使用時にフェイルオーバーの問題が発生する可能性があります。この例では、クラスタ上のマウントポイントを */global/ha-sam/sam1* に設定します。

```
#File
#Device    Device   Mount               System  fsck  Mount    Mount
#to Mount  to fsck  Point               Type    Pass  at Boot  Options
#--------  -------  ------------------  ------  ----  -------  ---------------
/devices   -        /devices            devfs   -     no       -
/proc      -        /proc               proc    -     no       -
...
sam1       -        /global/ha-sam/sam1
```

9. Oracle HSM 共有ファイルシステムの場合と同様に、*/etc/vfstab* ファイルレコードの残りのフィールドに入力します。次に、ファイルを保存し、エディタを閉じます。

```
#File
#Device    Device   Mount               System  fsck  Mount    Mount
#to Mount  to fsck  Point               Type    Pass  at Boot  Options
#--------  -------  ------------------  ------  ----  -------  ---------------
/devices   -        /devices            devfs   -     no       -
/proc      -        /proc               proc    -     no       -
...
```

```
sam1       -        /global/ha-sam/sam1 samfs   -     no       shared
:wq
[sam1mds-node1]root@solaris:~#
```

10. 高可用性ファイルシステムのマウントポイントを作成します。

    -p (*親*) オプションを指定して *mkdir* コマンドを使用すると、*/global* ディレクトリが作成されます (まだ存在しない場合)。

```
[sam1mds-node1]root@solaris:~# mkdir -p /global/ha-sam/sam1
```

11. プライマリノードで高可用性共有ファイルシステムをマウントします。

```
[sam1mds-node1]root@solaris:~# mount /global/ha-sam/sam1
```

12. 次に、セカンダリ HA-SAM クラスタノード上での潜在的な QFS メタデータサーバーの構成 を実行します。

9.4.4. セカンダリ HA-SAM クラスタノード上での潜在的な QFS メタデータサーバーの構成

2 ノードクラスタのセカンダリノードは、潜在的なメタデータサーバーとして機能します。潜在的なメタデータサーバーは、メタデータデバイスにアクセスできるホストであるため、メタデータサーバーの役割を担うことができます。そのため、プライマリノード上のアクティブなメタデータサーバーで障害が発生した場合、Solaris Cluster ソフトウェアはセカンダリノードにフェイルオーバーし、潜在的なメタデータサーバーをアクティブにできます。

1. HA-SAM クラスタのセカンダリノードに *root* としてログインします。

   この例では、*sam1mds-node2* がセカンダリノードです。

```
[sam1mds-node2]root@solaris:~#
```

2. */etc/opt/SUNWsamfs/mcf* ファイルをプライマリノードからセカンダリノードにコピーします。
3. エラーを *mcf* ファイルで確認します。コマンド */opt/SUNWsamfs/sbin/sam-fsd* を使用して、見つかったエラーを修正します。

sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、ホスト *sam1mds-node1* で *mcf* ファイルを確認します。

```
[sam1mds-node2]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[sam1mds-node2]root@solaris:~#
```

4. ファイルシステムを作成します。コマンド */opt/SUNWsamfs/sbin/sammkfs -S family-set-name* を使用します。ここで、*family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルによってファイルシステム装置に割り当てられるファミリセット名です。

   sammkfs コマンドは、*hosts.family-set-name* および *mcf* ファイルを読み取って、指定されたプロパティーを使用して Oracle HSM ファイルシステムを作成します。

```
[sam1mds-node2]root@solaris:~# sammkfs sam1
Building 'sam1' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[sam1mds-node2]root@solaris:~#
```

5. テキストエディタでオペレーティングシステムの */etc/vfstab* ファイルを開き、新しいファイルシステムの行を追加します。次に、ファイルを保存し、エディタを閉じます。

   この例では、*vi* エディタを使用します。

```
[sam1mds-node2]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount               System  fsck  Mount    Mount
#to Mount  to fsck  Point               Type    Pass  at Boot  Options
#--------  -------  ------------------- ------  ----  -------  --------------
/devices   -        /devices            devfs   -     no       -
```

```
/proc       -         /proc                proc     -      no        -
...
sam1        -         /global/ha-sam/sam1 samfs    -      no        shared
:wq
[sam1mds-node2]root@solaris:~#
```

6. セカンダリノードで高可用性共有ファイルシステムのマウントポイントを作成します。

```
[sam1mds-node2]root@solaris:~# mkdir -p /global/ha-sam/sam1
```

7. セカンダリノードで高可用性共有ファイルシステムをマウントします。

```
[sam1mds-node2]root@solaris:~# mount /global/ha-sam/sam1
```

8. 次に、HA-SAM クラスタリソースグループの作成を実行します。

9.4.5. HA-SAM クラスタリソースグループの作成

HA-SAM ソリューションのために高可用性リソースを管理するリソースグループを作成します。

1. HA-SAM クラスタのプライマリクラスタノードに *root* としてログインします。

   この例では、プライマリノードは *sam1mds-node1* です。

```
[sam1mds-node1]root@solaris:~#
```

2. HA-SAM ソリューションリソースを管理するための Solaris Cluster リソースグループを作成します。コマンド *clresourcegroup create -n node1,node2 groupname* を使用します。ここでは:
   - *node1* は、プライマリクラスタノードのホスト名です。
   - *node2* は、セカンダリクラスタノードのホスト名です。
   - *groupname* は、HA-SAM リソースグループに選択した名前です。

   この例では、*has-rg* という名前のリソースグループを作成して、ホスト *sam1mds-node1* および *sam1mds-node2* を含めます (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# clresourcegroup create /
```

```
-n sam1mds-node1,sam1mds-node2 has-rg
```

3. 次に、Oracle HSM 構成ファイルの高可用性ローカルファイルシステムの構成を実行します。

9.4.6. Oracle HSM 構成ファイルの高可用性ローカルファイルシステムの構成

次のフェイルオーバーを正常に回復するには、Oracle HSM software は、フェイルオーバーの発生時に実行されていたアーカイブ操作を再開する必要があります。アーカイブ操作を再開するには、ソフトウェアは、システム構成と、通常はアクティブなメタデータサーバーのローカルファイルシステムに格納されている状態情報にアクセスできる必要があります。そのため、クラスタ内の両方のノードから常にアクセス可能な状態になっている高可用性ローカルファイルシステムに必要な情報を移動する必要があります。

必要なファイルシステムを作成するには、次のように進めます。

1. HA-SAM クラスタのプライマリノードに *root* としてログインします。

   この例では、プライマリノードは *sam1mds-node1* です。

```
[sam1mds-node1]root@solaris:~#
```

2. プライマリクラスタノードで、グローバルデバイスの空きスライス上に UFS ファイルシステムを作成します。コマンド *newfs /dev/global/dsk/dXsY* を使用します。ここで *X* は、グローバルデバイスのデバイス ID (DID) 番号で、*Y* はスライス番号です。

   この例では、*/dev/global/dsk/d10s0* 上に新しいファイルシステムを作成します。

```
[sam1mds-node1]root@solaris:~# newfs /dev/global/dsk/d10s0
newfs: construct a new file system /dev/global/dsk/d10s0: (y/n)? y
/dev/global/dsk/d10s0: 1112940 sectors in 1374 cylinders of 15 tracks,
54 sectors 569.8MB in 86 cyl groups (16 c/g, 6.64MB/g, 3072 i/g)
super-block backups(for fsck -b #) at:
32, 13056, 26080, 39104, 52128, 65152, 78176, 91200, 104224, . . .
[sam1mds-node1]root@solaris:~#
```

3. プライマリクラスタノードで、オペレーティングシステムの */etc/vfstab* ファイルをテキストエディタで開きます。新しい UFS ファイルシステムの行を追加します。ファイルを保存して、エディタを閉じます。

   新しい行は、*/dev/global/dsk/dXsY /dev/global/dsk/dXsY / global/mount_point ufs 5 no global* 形式の空白文字区切りリストにしてください。ここでは:

   - *X* は、ファイルシステムを保持するグローバルデバイスのデバイス ID (DID) 番号です。
   - *Y* は、ファイルシステムを保持するスライスの番号です。
   - */dev/global/dsk/dXsY* は、マウントされるファイルシステムデバイスの名前です。
   - */dev/global/dsk/dXsY* は、*fsck* コマンドによってチェックされるファイルシステムデバイスの名前です。
   - *mount_point* は、UFS ファイルがマウントされるサブディレクトリの名前です。
   - *ufs* は、ファイルシステムタイプです。
   - *5* は、推奨される *fsck* パス番号です。
   - *no* は、起動時にファイルシステムをマウントしないようにオペレーティングシステムに指示します。
   - *global* は、両方のノードからアクセスできるようにファイルシステムをマウントします。

   次の例では、*vi* エディタを使用します。ファイルシステム名は */dev/global/dsk/ d10s0* で、マウントポイントは */global/hasam_cfg* です (ファイルシステムエントリは 1 行です。ページに収めるために改行が挿入され、バックスラッシュ文字でエスケープされています)。

```
[sam1mds-node1]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount              System  fsck  Mount    Mount
#to Mount  to fsck  Point              Type    Pass  at Boot  Options
#--------  -------  ------------------ ------  ----  -------  --------------
/devices   -        /devices           devfs   -     no       -
/proc      -        /proc              proc    -     no       -
...
sam1       -        /global/ha-samsam1  samfs   -     no       shared
/dev/global/dsk/d10s0   /dev/global/rdsk/d10s0  /global/hasam_cfg  ufs  5 /
     no   global
```

```
:wq
[sam1mds-node2]root@solaris:~#
```

4. プライマリクラスタノードで、高可用性ローカルファイルシステムのマウントポイントを作成します。コマンド *mkdir -p /global/mount_point* を使用します。ここで *mount_point* は、選択したマウントポイントディレクトリです。

   この例では、ディレクトリ */global/hasam_cfg* を作成します。

```
[sam1mds-node1]root@solaris:~# mkdir -p /global/hasam_cfg
```

5. セカンダリクラスタノードに *root* としてログインします。

   この例では、セカンダリノードは *sam1mds-node2* です。*ssh* を使用してログインします。

```
[sam1mds-node1]root@solaris:~# ssh root@sam1mds-node2
Password:
[sam1mds-node2]root@solaris:~#
```

6. セカンダリノードで、オペレーティングシステムの */etc/vfstab* ファイルをテキストエディタで開きます。新しい UFS ファイルシステム用の同一のエントリを追加します。ファイルを保存して、エディタを閉じます。

```
[sam1mds-node1]root@solaris:~# ssh root@sam1mds-node2
Password:
[sam1mds-node2]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount               System  fsck  Mount    Mount
#to Mount  to fsck  Point               Type    Pass  at Boot  Options
#--------  -------  ------------------- ------  ----  -------  --------------
/devices   -        /devices            devfs   -     no       -
/proc      -        /proc               proc    -     no       -
...
sam1       -        /global/ha-samsam1  samfs   -     no       shared
/dev/global/dsk/d10s0   /dev/global/rdsk/d10s0  /global/hasam_cfg  ufs  5  /
    no   global
:wq
[sam1mds-node1]root@solaris:~#
```

7. セカンダリノードで、同じマウントポイントを作成します。

   この例では、*/global/hasam_cfg* ディレクトリを作成します。次に、*ssh* セッションを閉じて、プライマリノードでの作業を再開します。

```
[sam1mds-node2]root@solaris:~# mkdir -p /global/hasam_cfg
[sam1mds-node2]root@solaris:~# exit
[sam1mds-node1]root@solaris:~#
```

8. プライマリノードで、高可用性ローカルファイルシステムをマウントします。コマンド *mount /global/mount_point* を使用します。ここで *mount_point* は、選択したマウントポイントディレクトリです。

   このコマンドは、両方のノードで UFS ファイルシステムをマウントします。この例では、*/global/hasam_cfg* でファイルシステムをマウントします。

```
[sam1mds-node1]root@solaris:~# mount /global/hasam_cfg
[sam1mds-node1]root@solaris:~#
```

9. プライマリノードで、Oracle HSM ステージング情報を保持するためのサブディレクトリを作成します。コマンド *mkdir -p /global/mount_point/catalog* を使用します。ここで *mount_point* は、選択したマウントポイントディレクトリです。

```
[sam1mds-node1]root@solaris:~# mkdir /global/hasam_cfg/catalog
[sam1mds-node1]root@solaris:~#
```

10. プライマリノードで、Oracle HSM アーカイブカタログを保持するためのサブディレクトリを作成します。コマンド *mkdir -p /global/mount_point/stager* を使用します。ここで *mount_point* は、選択したマウントポイントディレクトリです。

```
[sam1mds-node1]root@solaris:~# mkdir /global/hasam_cfg/stager
[sam1mds-node1]root@solaris:~#
```

11. 次に、高可用性ローカルファイルシステムへの Oracle HSM 構成ファイルの再配置を実行します。

9.4.7. 高可用性ローカルファイルシステムへの Oracle HSM 構成ファイルの再配置

1. HA-SAM クラスタのプライマリノードに *root* としてログインします。

   この例では、プライマリノードは *sam1mds-node1* です。

```
[sam1mds-node1]root@solaris:~#
```

2. プライマリノードで、*catalog/* および *stager/* ディレクトリを */var/opt/SUNWsamfs/* 内のデフォルトの場所から一時的な場所にコピーします。

   この例では、ディレクトリを */var/tmp/* に再帰的にコピーします (次の最初のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# cp -r /var/opt/SUNWsamfs/catalog /
/var/tmp/catalog
[sam1mds-node1]root@solaris:~# cp -r /var/opt/SUNWsamfs/stager /var/tmp/stager
[sam1mds-node1]root@solaris:~#
```

3. プライマリノードで、*catalog/* および *stager/* ディレクトリを */var/opt/SUNWsamfs/* から削除します。

```
[sam1mds-node1]root@solaris:~# rm -rf /var/opt/SUNWsamfs/catalog
[sam1mds-node1]root@solaris:~# rm -rf /var/opt/SUNWsamfs/stager
[sam1mds-node1]root@solaris:~#
```

4. プライマリノードで、カタログ情報のデフォルトの場所から高可用性 UFS ローカルファイルシステム内の新しい場所へのシンボリックリンクを作成します。コマンド *ln -s /global/mount_point/catalog /var/opt/SUNWsamfs/catalog* を使用します。ここでは:

   - *mount_point* は、高可用性ローカルファイルシステムがノードのルートファイルシステムに接続するサブディレクトリの名前です。
   - */var/opt/SUNWsamfs/catalog* はデフォルトの場所です。

   シンボリックリンクは、カタログ情報の要求を新しい場所に自動的にリダイレクトします。この例では、新しい場所 */global/hasam_cfg/catalog* を指す *catalog* リンクを作

成します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# ln -s /global/hasam_cfg/catalog /
/var/opt/SUNWsamfs/catalog
[sam1mds-node1]root@solaris:~#
```

5. プライマリノードで、ステージング情報のデフォルトの場所から高可用性 UFS ローカルファイルシステム内の新しい場所へのシンボリックリンクを作成します。コマンド *ln -s / global/mount_point/stager /var/opt/SUNWsamfs/stager* を使用します。ここでは:
   - *mount_point* は、高可用性ローカルファイルシステムがノードのルートファイルシステムに接続するサブディレクトリの名前です。
   - */var/opt/SUNWsamfs/stager* はデフォルトの場所です。

   シンボリックリンクは、ステージャー情報の要求を新しい場所に自動的にリダイレクトします。この例では、新しい場所 */global/hasam_cfg/stager* を指す *stager* リンクを作成します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# ln -s /global/hasam_cfg/stager / /var/opt/SUNWsamfs/stager
[sam1mds-node1]root@solaris:~#
```

6. プライマリノードで、シンボリックリンクによってデフォルトの */var/opt/SUNWsamfs/ catalog* および */var/opt/SUNWsamfs/stager* ディレクトリが置き換えられていることを確認します。リンクが高可用性ファイルシステムの新しい場所を指すことを確認します。

   この例では、リンクは適切です。

```
[sam1mds-node1]root@solaris:~# ls -l /var/opt/SUNWsamfs/catalog
lrwxrwxrwx 1 root other ... /var/opt/SUNWsamfs/catalog -> /global/hasam_cfg/catalog
[sam1mds-node1]root@solaris:~# ls -l /var/opt/SUNWsamfs/stager
lrwxrwxrwx 1 root other ... /var/opt/SUNWsamfs/stager -> /global/hasam_cfg/stager
[sam1mds-node1]root@solaris:~#
```

7. *catalog/* および *stager/* ディレクトリの内容を一時的な場所から高可用性ファイルシステムにコピーします。

この例では、*catalog/* および *stager/* ディレクトリを */var/tmp/* から新しい場所 */global/hasam_cfg/stager* にコピーします (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# cp -rp /var/tmp/catalog/* /
/var/opt/SUNWsamfs/catalog
[sam1mds-node1]root@solaris:~# cp -rp /var/tmp/stager/* /
/var/opt/SUNWsamfs/stager
[sam1mds-node1]root@solaris:~#
```

8. HA-SAM クラスタのセカンダリノードに *root* としてログインします。

   この例では、*ssh* (セキュアシェル) を使用してセカンダリノード *sam1mds-node2* にログインします。

```
[sam1mds-node1]root@solaris:~# ssh root@sam1mds-node2
Password:
[sam1mds-node2]root@solaris:~#
```

9. セカンダリノードで、カタログ情報のデフォルトの場所から高可用性 UFS ローカルファイルシステム内の新しい場所へのシンボリックリンクを作成します。コマンド *ln -s /global/mount_point/catalog /var/opt/SUNWsamfs/catalog* を使用します。ここでは:

   - *mount_point* は、高可用性ローカルファイルシステムがノードのルートファイルシステムに接続するサブディレクトリの名前です。
   - */var/opt/SUNWsamfs/catalog* はデフォルトの場所です。

   シンボリックリンクは、カタログ情報の要求を新しい場所に自動的にリダイレクトします。この例では、新しい場所 */global/hasam_cfg/catalog* を指す *catalog* リンクを作成します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node2]root@solaris:~# ln -s /global/hasam_cfg/catalog /
/var/opt/SUNWsamfs/catalog
[sam1mds-node2]root@solaris:~#
```

10. セカンダリノードで、ステージング情報のデフォルトの場所から高可用性 UFS ローカルファイルシステム内の新しい場所へのシンボリックリンクを作成します。コマンド *ln -s /global/mount_point/stager /var/opt/SUNWsamfs/stager* を使用します。ここでは:

    - *mount_point* は、高可用性ローカルファイルシステムがノードのルートファイルシステムに接続するサブディレクトリの名前です。
    - */var/opt/SUNWsamfs/stager* はデフォルトの場所です。

    シンボリックリンクは、ステージャー情報の要求を新しい場所に自動的にリダイレクトします。この例では、新しい場所 */global/hasam_cfg/stager* を指す *stager* リンクを作成します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node2]root@solaris:~# ln -s /global/hasam_cfg/stager / /var/opt/SUNWsamfs/stager
[sam1mds-node2]root@solaris:~#
```

11. セカンダリノードで、シンボリックリンクによってデフォルトの */var/opt/SUNWsamfs/catalog* および */var/opt/SUNWsamfs/stager* ディレクトリが置き換えられていることを確認します。リンクが高可用性ファイルシステムの新しい場所を指すことを確認します。

    この例では、リンクは適切です。そのため、*ssh* セッションを閉じて、プライマリノードで作業を再開します。

```
[sam1mds-node2]root@solaris:~# ls -l /var/opt/SUNWsamfs/catalog
lrwxrwxrwx 1 root other ... /var/opt/SUNWsamfs/catalog -> /global/hasam_cfg/catalog
[sam1mds-node2]root@solaris:~# ls -l /var/opt/SUNWsamfs/stager
lrwxrwxrwx 1 root other ... /var/opt/SUNWsamfs/stager -> /global/hasam_cfg/stager
[sam1mds-node2]root@solaris:~# exit
[sam1mds-node1]root@solaris:~#
```

12. 次に、高可用性ローカルファイルシステムを使用するための HA-SAM クラスタの構成を実行します。

9.4.8. 高可用性ローカルファイルシステムを使用するための HA-SAM クラスタの構成

1. HA-SAM クラスタのプライマリノードで、*SUNW.HAStoragePlus* リソースタイプをクラスタ構成の一部として登録します。Solaris Cluster コマンド *clresourcetype register SUNW.HAStoragePlus* を使用します。

```
[sam1mds-node1]root@solaris:~# clresourcetype register SUNW.HAStoragePlus
[sam1mds-node1]root@solaris:~#
```

2. プライマリノードで、*SUNW.HAStoragePlus* リソースタイプの新しいインスタンスを作成して、これを Solaris Cluster リソースグループに関連付けます。コマンド *clresource create -g groupname -t SUNW.HAStoragePlus -x FilesystemMountPoints=mountpoint -x AffinityOn=TRUE resourcename* を使用します。ここでは:

   - *groupname* は、HA-SAM リソースグループに選択した名前です。
   - *SUNW.HAStoragePlus* は、ローカルファイルシステムのフェイルオーバーをサポートする Solaris Cluster リソースタイプです。
   - *mountpoint* は、カタログとステージャーファイルを保持する高可用性ローカルファイルシステムのマウントポイントです。
   - *resourcename* は、リソース自体に選択した名前です。

   この例では、タイプ *SUNW.HAStoragePlus* の *has-cfg* という名前のリソースを作成します。新しいリソースをリソースグループ *has-rg* に追加します。次に、リソースの拡張プロパティーを構成します。*FilesystemMountPoints* を */global/hasam_cfg* に設定して、*AffinityOn* を *TRUE* に設定します (次のコマンドはそれぞれ 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# clresource create -g has-rg /
-t SUNW.HAStoragePlus -x FilesystemMountPoints=/global/hasam_cfg /
-x AffinityOn=TRUE has-cfg
[sam1mds-node1]root@solaris:~#
```

3. 次に、QFS ファイルシステムメタデータサーバーのフェイルオーバーの構成を実行します。

9.4.9. QFS ファイルシステムメタデータサーバーのフェイルオーバーの構成

Oracle HSM software によって定義されたリソースタイプである *SUNW.qfs* クラスタリソースを作成して、メタデータサーバーのフェイルオーバーを構成します (詳細は、*SUNW.qfs* のマニュアルページを参照)。HA-SAM 構成のリソースを作成して構成するには、次のように進めます。

1. HA-SAM クラスタのプライマリクラスタノードに *root* としてログインします。

   この例では、プライマリノードは *sam1mds-node1* です。

```
[sam1mds-node1]root@solaris:~#
```

2. Solaris Cluster ソフトウェア用にリソースタイプ *SUNW.qfs* を定義します。コマンド *clresourcetype register SUNW.qfs* を使用します。

```
[sam1mds-node1]root@solaris:~# clresourcetype register SUNW.qfs
[qfs1mds-node1]root@solaris:~#
```

3. 登録ファイルが見つからないために登録が失敗した場合、Solaris Cluster がリソースタイプの登録ファイル */opt/cluster/lib/rgm/rtreg/* を保持するディレクトリに、*/opt/SUNWsamfs/sc/etc/* ディレクトリへのシンボリックリンクを配置します。

   Oracle HSM software をインストールする前に Oracle Solaris Cluster ソフトウェアをインストールしなかった場合、登録は失敗します。通常、Oracle HSM は、インストール時に Solaris Cluster を検出すると、*SUNW.qfs* 登録ファイルの場所を自動的に指定します。この例では、リンクを手動で作成します。

```
[qfs1mds-node1]root@solaris:~# cd /opt/cluster/lib/rgm/rtreg/
[qfs1mds-node1]root@solaris:~# ln -s /opt/SUNWsamfs/sc/etc/SUNW.qfs SUNW.qfs
[qfs1mds-node1]root@solaris:~#
```

4. 新しいリソースグループで、アクティブなメタデータサーバーの仮想ホスト名を設定します。Solaris Cluster コマンド *clreslogicalhostname create -g group-name virtualMDS* を使用します。ここでは:
   - *group-name* は、QFS リソースグループの名前です。
   - *virtualMDS* は仮想ホスト名です。

共有ファイルシステムの hosts ファイルで使用したものと同じ仮想ホスト名を使用します。この例では、仮想ホスト名 *sam1mds* を *has-rg* リソースグループに追加します。

```
[sam1mds-node1]root@solaris:~# clreslogicalhostname create -g has-rg sam1mds
[qfs1mds-node1]root@solaris:~#
```

5. Oracle HSM ファイルシステムリソースをリソースグループに追加します。コマンド *clresource create -g groupname -t SUNW.qfs -x QFSFileSystem=mount-point* を使用します。ここでは:
   - *groupname* は、HA-SAM リソースグループに選択した名前です。
   - *SUNW.qfs* は、QFS ファイルシステムメタデータサーバーのフェイルオーバーをサポートする Solaris Cluster リソースタイプです。
   - *mount-point* は、クラスタ内のファイルシステムのマウントポイントである、システムのルートディレクトリの直下にはないサブディレクトリです。

     QFS 共有ファイルシステムをルートの直下にマウントすると、*SUNW.qfs* リソースタイプの使用時にフェイルオーバーの問題が発生する可能性があります。
   - *resource-name* は、リソース自体に選択した名前です。

   この例では、リソースグループ *has-rg* にタイプ *SUNW.qfs* の *has-qfs* という名前のリソースを作成します。*SUNW.qfs* 拡張プロパティー *QFSFileSystem* を */global/ha-sam/sam1* マウントポイントに設定します。標準プロパティー *Resource_dependencies* を、アクティブなメタデータサーバーを表す仮想ホスト名である *sam1mds* に設定します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# clresource create -g has-rg -t SUNW.qfs /
-x QFSFileSystem=/global/ha-sam/sam1 -y Resource_dependencies=sam1mds has-qfs
[sam1mds-node1]root@solaris:~#
```

6. 次に、Oracle Hierarchical Storage Manager アプリケーションのフェイルオーバーの構成を実行します。

9.4.10. Oracle Hierarchical Storage Manager アプリケーションのフェイルオーバーの構成

Oracle HSM *SUNW.hasam* リソースを作成することで、Oracle Hierarchical Storage Manager アプリケーションのフェイルオーバーを構成します。このリソースタイプは、Oracle HSM プロセスの正常なシャットダウンと再起動を調整します。

Oracle HSM アプリケーションのフェイルオーバーを構成するには、次のように進めます。

1. HA-SAM クラスタのプライマリクラスタノードに *root* としてログインします。

   この例では、プライマリノードは *sam1mds-node1* です。

```
[sam1mds-node1]root@solaris:~#
```

2. Solaris Cluster ソフトウェア用にリソースタイプ *SUNW.hasam* を定義します。コマンド *clresourcetype register SUNW.hasam* を使用します。

```
[sam1mds-node1]root@solaris:~# clresourcetype register SUNW.hasam
[sam1mds-node1]root@solaris:~#
```

3. Oracle HSM *SUNW.hasam* リソースをリソースグループに追加します。コマンド *clresource create -g groupname -t SUNW.hasam -x QFSName=fs-name -x CatalogFileSystem=mount-point resource-name* を使用します。ここでは:
   - *groupname* は、HA-SAM リソースグループに選択した名前です。
   - *SUNW.hasam* は、Oracle Hierarchical Storage Manager アプリケーションのフェイルオーバーをサポートする Solaris Cluster リソースタイプです。
   - *mount-point* は、Oracle HSM アーカイブカタログを保持するグローバルファイルシステムのマウントポイントです。
   - *resource-name* は、リソース自体に選択した名前です。

   この例では、リソースグループ *has-rg* にタイプ *SUNW.hasam* の *has-sam* という名前のリソースを作成します。*SUNW.hasam* 拡張プロパティー *QFSName* を、*mcf* ファイルで指定された QFS ファイルシステム名 *sam1* に設定します。*SUNW.hasam* 拡張プロパティー *CatalogFileSystem* を */global/hasam_cfg* マウントポイントに設定します。

```
[sam1mds-node1]root@solaris:~# clresource create -g has-rg -t SUNW.hasam /
-x QFSName=sam1 -x CatalogFileSystem=/global/hasam_cfg has-sam
[sam1mds-node1]root@solaris:~#
```

4. 次に、HA-SAM ソリューションのクラスタリソースの依存関係の定義を実行します。

9.4.11. HA-SAM ソリューションのクラスタリソースの依存関係の定義

1. HA-SAM クラスタのプライマリクラスタノードに *root* としてログインします。

   この例では、プライマリノードは *sam1mds-node1* です。

```
[sam1mds-node1]root@solaris:~#
```

2. 高可用性ローカルファイルシステムが使用可能である場合を除き、QFS ファイルシステムは起動すべきではありません。そのため、*SUNW.qfs* リソースを *SUNW.HAStoragePlus* リソースに依存させます。Solaris Cluster コマンド *clresource set -p Resource_dependencies=dependency resource-name* を使用します。ここでは:

   - *dependency* は、*SUNW.HAStoragePlus* リソースの名前です。
   - *resource-name* は、*SUNW.qfs* リソースの名前です。

   この例では、*SUNW.qfs* リソースを *SUNW.HAStoragePlus* リソース *has-cfg* に依存させます (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# clresource set /
-p Resource_dependencies=has-cfg has-qfs
[sam1mds-node1]root@solaris:~#
```

3. アクティブな QFS メタデータサーバーがオンラインである場合を除き、クラスタは仮想ホスト名を使用可能にするべきではありません。そのため、仮想ホスト名は *SUNW.qfs* リソースに依存させます。Solaris Cluster コマンド *clresource set -p Resource_dependencies=virtualMDS resource-name* を使用します。ここでは:

   - *virtualMDS* は、アクティブな Oracle HSM メタデータサーバーを表す仮想ホスト名です。
   - *resource-name* は、*SUNW.qfs* リソースの名前です。

この例では、*SUNW.qfs* リソースの設定時に作成した仮想ホスト名は *sam1mds* です。リソース自体の名前は *has-qfs* です (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# clresource set /
-p Resource_dependencies=sam1mds has-qfs
[sam1mds-node1]root@solaris:~#
```

4. *SUNW.qfs* リソースをリソースグループに依存させます。Solaris Cluster コマンド *clresource set -p Resource_dependencies=dependency resource-name* を使用します。ここでは:
   - *dependency* は、*SUNW.HAStoragePlus* リソースの名前です。
   - *resource-name* は、*SUNW.qfs* リソースの名前です。

   この例では、*has-qfs* を *has-rg* リソースグループに依存させます (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[sam1mds-node1]root@solaris:~# clresource set /
-p Resource_dependencies=has-qfs has-rg
[sam1mds-node1]root@solaris:~#
```

5. 次に、HA-SAM リソースグループのオンライン化および構成のテストを実行します。

9.4.12. HA-SAM リソースグループのオンライン化および構成のテスト

1. HA-SAM クラスタのプライマリクラスタノードに *root* としてログインします。

   この例では、プライマリノードは *sam1mds-node1* です。

```
[sam1mds-node1]root@solaris:~#
```

2. リソースグループをオンラインにします。Solaris Cluster コマンド *clresourcegroup manage groupname*, and *clresourcegroup online -emM groupname* を使用します。ここで *groupname* は HA-SAM リソースグループの名前です。

   この例では、*has-rg* リソースグループをオンラインにします。

```
[sam1mds-node1]root@solaris:~# clresourcegroup manage has-rg
[sam1mds-node1]root@solaris:~# clresourcegroup online -emM has-rg
[sam1mds-node1]root@solaris:~#
```

3. HA-SAM リソースグループを必ずオンラインにしてください。Solaris Cluster *clresourcegroup status* コマンドを使用します。

   この例では、*has-rg* リソースグループは、プライマリノード *sam1mds-node1* 上では *online* です。

```
[sam1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
has-rg      sam1mds-node1   No          Online
            sam1mds-node2   No          Offline
[sam1mds-node1]root@solaris:~#
```

4. 次に、リソースグループを正しくフェイルオーバーさせます。リソースグループをセカンダリノードに移動します。Solaris Cluster コマンド *clresourcegroup switch -n node2 groupname* を使用します。ここで *node2* は、セカンダリノードの名前で、*groupname* は、HA-SAM リソースグループに選択した名前です。次に、*clresourcegroup status* を使用して結果を確認します。

   この例では、*has-rg* リソースグループを *sam1mds-node2* に移動して、指定したノードでリソースグループがオンラインになることを確認します。

```
[sam1mds-node1]root@solaris:~# clresourcegroup switch -n sam1mds-node2 has-rg
[sam1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
has-rg      sam1mds-node1   No          Offline
            sam1mds-node2   No          Online
[sam1mds-node1]root@solaris:~#
```

5. リソースグループをプライマリノードに戻します。Solaris Cluster コマンド *clresourcegroup switch -n node1 groupname* を使用します。ここで *node1* は、プライマリノードの名前で、*groupname* は、HA-SAM リソースグループに選択した名前です。次に、*clresourcegroup status* を使用して結果を確認します。

   この例では、*has-rg* リソースグループを *sam1mds-node1* に正常に戻します。

```
[sam1mds-node1]root@solaris:~# clresourcegroup switch -n sam1mds-node1 has-rg
[sam1mds-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
has-rg      sam1mds-node1   No          Online
            sam1mds-node2   No          Offline
[sam1mds-node1]root@solaris:~#
```

6. 必要に応じて、この時点で高可用性ネットワークファイルシステム (HA-NFS) 共有を構成します。

   HA-NFS を設定するための詳細な手順は、*Oracle Solaris Cluster* オンラインドキュメントライブラリに含まれている『*Oracle Solaris Cluster Data Service for Network File System (NFS) ガイド*』に記載されています。
7. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
8. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

9.5. 高可用性 QFS 共有ファイルシステムおよび Oracle RAC

Solaris Cluster-Oracle Real Application Cluster (SC-RAC) 構成では、Solaris Cluster ソフトウェアは、Oracle Database および Oracle Real Application Cluster (RAC) ソフトウェアもホストするノードにマウントされている *SUNW.qfs* リソースとして QFS 共有ファイルシステムを管理します。すべてのノードが QFS サーバーとして構成され、そのうちの 1 つがアクティブなメタデータサーバー、その他は潜在的なメタデータサーバーとなります。アクティブなメタデータサーバーノードで障害が発生した場合、Solaris Cluster ソフトウェアは、正常なノード上の潜在的なメタデータサーバーを自動的にアクティブにして、フェイルオーバーを開始します。入出力は Oracle RAC を使用して調整され、QFS ファイルシステムは共有され、すべてのノード上にすでにマウントされています。そのため、データへのアクセスは中断されません。

SC-RAC 構成では、RAC ソフトウェアは入出力要求の調整、ワークロードの分散、およびクラスタノードで実行されている複数の Oracle データベースインスタンスの単一で一貫性のあるデータベースファイルセットの維持を行います。RAC ではファイルシステムの整合性が確保されているため、QFS の潜在的なメタデータサーバーは、共有ファイルシステムのクライアントとして入出力を実行できます。追加情報については、*Oracle Solaris Cluster オンラインドキュメントライブラリ*で Oracle Real Application Cluster に関する Oracle Solaris Cluster Data Service のドキュメントを参照してください。

SC-RAC ファイルシステムを構成するには、次のタスクを実行します。

- すべての SC-RAC クラスタノードにおける QFS 共有ファイルシステムの hosts ファイルの作成
- QFS サーバーおよび HA-COTC クラスタの外部にあるクライアントでのローカル hosts ファイルの作成
- プライマリ SC-RAC クラスタノード上でのアクティブな QFS メタデータサーバーの構成、または ソフトウェア RAID ストレージを使用した SC-RAC ノード上での QFS メタデータサーバーの構成
- 残りの SC-RAC クラスタノード上での潜在的な QFS メタデータサーバーの構成
- SC-RAC メタデータサーバーのフェイルオーバーの構成
- 必要に応じて、「NFS と SMB/CIFS を使用した複数のホストからファイルシステムへのアクセス」の説明に従ってネットワークファイルシステム (NFS) 共有を構成します。高可用性 NFS (HA-NFS) はサポート*されません*。

9.5.1. すべての SC-RAC クラスタノードにおける QFS 共有ファイルシステムの hosts ファイルの作成

QFS 共有ファイルシステムでは、すべてのホストがファイルシステムのメタデータにアクセスできるように、メタデータサーバーで hosts ファイルを構成する必要があります。hosts ファイルは、*/etc/opt/SUNWsamfs/* ディレクトリに *mcf* ファイルとともに格納されています。共有ファイルシステムの初期作成中に、*sammkfs -S* コマンドを実行すると、このファイルに格納されている設定を使用して共有が構成されます。ここで、次の手順を使用して作成します。

1. SC-RAC クラスタのプライマリクラスタノードに *root* としてログインします。

   この例では、プライマリノードは *qfs1rac-node1* です。

```
[qfs1rac-node1]root@solaris:~#
```

2. クラスタ構成を表示します。コマンド */usr/global/bin/cluster show* を使用します。出力で、それぞれの *Node Name* のレコードを見つけて、各ネットワークアダプタの *privatehostname*、*Transport Adapter* の名前、および *ip_address* プロパティーを記録します。

   この例では、それぞれのノードには、*qfe3* と *hme0* の 2 つのネットワークインタフェースがあります。

   - *hme0* アダプタには、クラスタがノード間の内部通信に使用するプライベートネットワークの IP アドレスがあります。Solaris Cluster ソフトウェアは、各プライベートアドレスに対応する *privatehostname* を割り当てます。

     デフォルトでは、プライマリノードのプライベートホスト名は *clusternode1-priv* で、セカンダリノードのプライベートホスト名は *clusternode2-priv* です。
   - *qfe3* アダプタには、クラスタがデータ転送に使用するパブリック IP アドレスとパブリックホスト名である *qfs1rac-node1* と *qfs1rac-node2* があります。

   表示は、省略記号 (*...*) を使用して省略されています。

```
[qfs1rac-node1]root@solaris:~# cluster show
...
  === Cluster Nodes ===
  Node Name:                                    qfs1rac-node1...
    privatehostname:                               clusternode1-priv...
    Transport Adapter List:                        qfe3, hme0...
    Transport Adapter:                          qfe3...
      Adapter Property(ip_address):                172.16.0.12...
    Transport Adapter:                          hme0...
      Adapter Property(ip_address):                10.0.0.129...
  Node Name:                                    qfs1rac-node2...
    privatehostname:                               clusternode2-priv...
    Transport Adapter List:                        qfe3, hme0...
      Adapter Property(ip_address):                172.16.0.13...
    Transport Adapter:                          hme0
      Adapter Property(ip_address):                10.0.0.122...
  Node Name:                                    qfs1rac-node3...
    privatehostname:                               clusternod3-priv...
    Transport Adapter List:                        qfe3, hme0...
      Adapter Property(ip_address):                172.16.0.33...
```

```
    Transport Adapter:                          hme0
      Adapter Property(ip_address):                10.0.0.092
```

3. テキストエディタを使用して、ファイル */etc/opt/SUNWsamfs/hosts.family-set-name* を作成します。ここで、*family-set-name* は、*/etc/opt/SUNWsamfs/mcf* ファイルによってファイルシステム装置に割り当てられるファミリセット名です。

   この例では、*vi* テキストエディタを使用してファイル *hosts.qfs1rac* を作成します。ホストテーブル内の列を示すために、コメントを示すハッシュ記号 (#) で各行を開始して、いくつかのオプションの見出しを追加します。

```
[qfs1rac-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/hosts.qfs1rac
# /etc/opt/SUNWsamfs/hosts.qfs1rac
#                                             Server   On/  Additional
#Host Name     Network Interface               Ordinal  Off  Parameters
#------------  ------------------------------  -------  ---  ----------
```

4. テーブルの最初の列に、プライマリメタデータサーバーノードとセカンダリメタデータサーバーノードのホスト名、その後にいくつかの空白文字を入力します。それぞれのエントリを別の行に入力してください。

   hosts ファイルでは、行は行 (レコード) で、空白文字は列 (フィールド) 区切り文字です。この例では、最初の 2 行の「Host Name」列には、クラスタノードのホスト名 *qfs1rac-node1*、*qfs1rac-node2*、および *qfs1rac-node3* が一覧表示されます。

```
#                                             Server   On/  Additional
#Host Name     Network Interface               Ordinal  Off  Parameters
#------------  ------------------------------  -------  ---  ----------
qfs1rac-node1
qfs1rac-node2
qfs1rac-node3
```

5. 各行の 2 番目の列では、ホスト *Host Name* の *Network Interface* 情報の指定を開始します。それぞれの SC-RAC クラスタノードの Solaris Cluster プライベートホスト名またはプライベートネットワークアドレスと、その後に続けてコンマを入力します。

   SC-RAC サーバーノードは、高可用性クラスタ内のサーバー間の通信にプライベートホスト名を使用します。この例では、Solaris Cluster ソフトウェアによって割り当てられたデ

フォルトの名前であるプライベートホスト名 *clusternode1-priv*、*clusternode2-priv*、および *clusternode3-priv* を使用します。

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  ------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,
qfs1rac-node2  clusternode2-priv,
qfs1rac-node3  clusternode3-priv,
```

6. 各行の 2 番目の列にあるコンマのあとに、アクティブなメタデータサーバーのパブリックホスト名と、その後に空白文字を続けて入力します。

   SC-RAC サーバーノードは、パブリックデータネットワークを使用して、すべてがクラスタの外部にあるクライアントと通信します。アクティブなメタデータサーバーの IP アドレスとホスト名はフェイルオーバー中に変わる (たとえば、*qfs1rac-node1* から *qfs1rac-node2*) ため、仮想ホスト名 *qfs1rac-mds* を使用してアクティブなサーバーを表します。あとで、*qfs1rac-mds* の要求を、アクティブなメタデータサーバーを現在ホストしているノードに常にルーティングするように Solaris Cluster ソフトウェアを構成します。

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  ------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,qfs1rac-mds
qfs1rac-node2  clusternode2-priv,qfs1rac-mds
qfs1rac-node3  clusternode3-priv,qfs1rac-mds
```

7. 各行の 3 番目の列に、サーバーの番号 (アクティブなメタデータサーバーの場合は *1*、潜在的なメタデータサーバーの場合は *2*) とその後にスペースを続けて入力します。

   この例では、プライマリノード *qfs1rac-node1* がアクティブなメタデータサーバーです。そのため、これは番号 *1* です。2 番目のノード *qfs1rac-node2* は、番号 *2* などです。

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  ------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,qfs1rac-mds     1
qfs1rac-node2  clusternode2-priv,qfs1rac-mds     2
```

```
qfs1rac-node3  clusternode3-priv,qfs1rac-mds     3
```

8. 各行の 4 番目の列に、*0* (ゼロ) とその後に空白文字を続けて入力します。

   4 列目の *0*、- (ハイフン)、または空白値は、ホストが「*on*」(共有ファイルシステムへのアクセスありで構成) であることを示します。*1* (数字の 1) は、ホストが「*off*」(構成済みだが、ファイルシステムへのアクセスなし) であることを示します (共有ファイルシステムを管理する際のこれらの値の使用については、*samsharefs* のマニュアルページを参照してください)。

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  ------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,qfs1rac-mds     1        0
qfs1rac-node2  clusternode2-priv,qfs1rac-mds     2        0
qfs1rac-node3  clusternode3-priv,qfs1rac-mds     3        0
```

9. プライマリノードの行の 5 番目の列に、キーワード *server* を入力します。ファイルを保存して、エディタを閉じます。

   server キーワードは、デフォルトのアクティブなメタデータサーバーを示します。

```
#                                          Server   On/  Additional
#Host Name     Network Interface            Ordinal  Off  Parameters
#------------  ------------------------------  -------  ---  ----------
qfs1rac-node1  clusternode1-priv,qfs1rac-mds     1        0    server
qfs1rac-node2  clusternode2-priv,qfs1rac-mds     2        0
qfs1rac-node3  clusternode3-priv,qfs1rac-mds     2        0
:wq
[qfs1rac-node1]root@solaris:~#
```

10. 各ノード上のグローバル */etc/opt/SUNWsamfs/hosts.family-set-name* ファイルのコピーを SC-RAC クラスタに格納します。
11. 次に、プライマリ SC-RAC クラスタノード上でのアクティブな QFS メタデータサーバーの構成を実行します。

9.5.2. プライマリ SC-RAC クラスタノード上でのアクティブな QFS メタデータサーバーの構成

1. SC-RAC クラスタのプライマリノードと QFS 共有ファイルシステムのアクティブなメタデータサーバーの両方としての役割を果たすクラスタノードを選択します。*root* としてログインします。

   この例では、プライマリノードは *qfs1rac-node1* です。

```
[qfs1rac-node1]root@solaris:~#
```

2. QFS ファイルシステムに使用されるグローバルストレージデバイスを選択します。コマンド */usr/global/bin/cldevice list -v* を使用します。

   Solaris Cluster ソフトウェアは、クラスタノードに接続されているすべてのデバイスに一意のデバイス ID (DID) を割り当てます。*グローバルデバイス*は、クラスタ内のすべてのノードからアクセスできるのに対して、*ローカルデバイス*は、ローカルデバイスをマウントするホストからのみアクセス可能です。グローバルデバイスはフェイルオーバー後にアクセス可能な状態のままになります。ローカルデバイスはそうではありません。

   この例では、デバイス *d1*、*d2*、*d6*、*d7*、および *d8* は、すべてのノードからアクセス可能ではありません。そのため、高可用性 QFS 共有ファイルシステムの構成時にデバイス *d3*、*d4*、および *d5* の中から選択します。

```
[qfs1rac-node1]root@solaris:~# cldevice list -v
DID Device          Full Device Path
----------          ----------------
d1                  qfs1rac-node1:/dev/rdsk/c0t0d0
d2                  qfs1rac-node1:/dev/rdsk/c0t6d0
d3                  qfs1rac-node1:/dev/rdsk/c1t1d0
d3                  qfs1rac-node2:/dev/rdsk/c1t1d0
d3                  qfs1rac-node3:/dev/rdsk/c1t1d0
d4                  qfs1rac-node1:/dev/rdsk/c1t2d0
d4                  qfs1rac-node2:/dev/rdsk/c1t2d0
d4                  qfs1rac-node3:/dev/rdsk/c1t2d0
d5                  qfs1rac-node1:/dev/rdsk/c1t3d0
d5                  qfs1rac-node2:/dev/rdsk/c1t3d0
d5                  qfs1rac-node3:/dev/rdsk/c1t3d0
d6                  qfs1rac-node2:/dev/rdsk/c0t0d0
```

```
d7                  qfs1rac-node2:/dev/rdsk/c0t1d0
d8                  qfs1rac-node3:/dev/rdsk/c0t1d0
```

3. *mr* データデバイスを使用する高パフォーマンスの *ma* 共有ファイルシステムを作成します。テキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開きます。

   この例では、ファイルシステム *qfs1rac* を構成します。デバイス *d3* をメタデータデバイス (装置タイプ *mm*) として構成して、*d4* および *d5* をデータデバイス (装置タイプ *mr*) として使用します。

```
[qfs1rac-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#-----------------   ---------  ---------  -------  ------   --------------
qfs1rac               100        ma         qfs1rac  -
 /dev/did/dsk/d3s0    101        mm         qfs1rac  -
 /dev/did/dsk/d4s0    102        mr         qfs1rac  -
 /dev/did/dsk/d5s0    103        mr         qfs1rac  -
...
```

4. */etc/opt/SUNWsamfs/mcf* ファイルで、ファイルシステムエントリの *Additional Parameters* 列に *shared* パラメータを入力します。ファイルを保存します。

```
[qfs1rac-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family   Device   Additional
# Identifier          Ordinal    Type       Set      State    Parameters
#-----------------   ---------  ---------  -------  ------   --------------
qfs1rac               100        ma         qfs1rac  -        shared
 /dev/did/dsk/d3s0     101        mm         qfs1rac  -
 /dev/did/dsk/d4s0     102        mr         qfs1rac  -
 /dev/did/dsk/d5s0     103        mr         qfs1rac  -
...
:wq
[qfs1rac-node1]root@solaris:~#
```

5. エラーを *mcf* ファイルで確認します。コマンド */opt/SUNWsamfs/sbin/sam-fsd* を使用して、見つかったエラーを修正します。

sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、ホスト *qfs1rac-node1* 上で *mcf* ファイルを確認します。

```
[qfs1rac-node1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1rac-node1]root@solaris:~#
```

6. ファイルシステムを作成します。コマンド */opt/SUNWsamfs/sbin/sammkfs -S family-set-name* を使用します。ここで、*family-set-name* は、ファイルシステムの装置 ID です。

   sammkfs コマンドは、*hosts.family-set-name* および *mcf* ファイルを読み取って、指定されたプロパティーを使用して共有ファイルシステムを作成します。

```
[qfs1rac-node1]root@solaris:~# sammkfs -S qfs1rac
Building 'qfs1rac' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[qfs1rac-node1]root@solaris:~#
```

7. テキストエディタでオペレーティングシステムの */etc/vfstab* ファイルを開き、新しいファイルシステムの行を開始します。最初の列にファイルシステム名、空白文字、2 番目の列にハイフン、さらに空白を入力します。

   この例では、*vi* テキストエディタを使用します。*qfs1rac* ファイルシステムの行を開始します。ハイフンは、オペレーティングシステムが UFS ツールを使用してファイルシステムの整合性チェックを試行しないようにします。

```
[qfs1rac-node1]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount           System  fsck  Mount    Mount
#to Mount  to fsck  Point           Type    Pass  at Boot  Options
```

```
#--------  -------  ---------------  ------  ----  -------  ------------------
/devices   -        /devices         devfs   -     no       -
/proc      -        /proc            proc    -     no       -
...
qfs1rac    -
```

8. */etc/vfstab* ファイルの 3 番目の列に、クラスタを基準とした相対的なファイルシステムのマウントポイントを入力します。システムのルートディレクトリの直下にはないサブディレクトリを指定します。

   QFS 共有ファイルシステムをルートの直下にマウントすると、*SUNW.qfs* リソースタイプの使用時にフェイルオーバーの問題が発生する可能性があります。この例では、*qfs1rac* ファイルシステムのマウントポイントは */global/sc-rac/qfs1rac* です。

```
#File
#Device    Device   Mount                    System  fsck  Mount    Mount
#to Mount  to fsck  Point                    Type    Pass  at Boot  Options
#--------  -------  -----------------------  ------  ----  -------  ----------
/devices   -        /devices                 devfs   -     no       -
/proc      -        /proc                    proc    -     no       -
...
qfs1rac    -        /global/sc-rac/qfs1rac
```

9. 4 番目の列にファイルシステムタイプ *samfs*、5 番目の列に - (ハイフン)、6 番目の列に *no* を入力します。

```
#File
#Device    Device   Mount                    System  fsck  Mount    Mount
#to Mount  to fsck  Point                    Type    Pass  at Boot  Options
#--------  -------  -----------------------  ------  ----  -------  ----------
/devices   -        /devices                 devfs   -     no       -
/proc      -        /proc                    proc    -     no       -
...
qfs1rac    -        /global/sc-rac/qfs1rac   samfs   -     no
:wq
[qfs1rac-node1]root@solaris:~#
```

10. */etc/vfstab* ファイルの 7 番目の列に、次に一覧表示されているマウントオプションを入力します。次に、ファイルを保存し、エディタを閉じます。

    SC-RAC クラスタ構成では、次のマウントオプションが推奨されます。これらは、この時点で */etc/vfstab* 内で指定することも、またはその方が便利な場合はファイル */etc/opt/SUNWsamfs/samfs.cmd* 内で指定することもできます。

    - *shared*
    - *stripe=1*
    - *sync_meta=1*
    - *mh_write*
    - *qwrite*
    - *forcedirectio*
    - *notrace*
    - *rdlease=300*
    - *wrlease=300*
    - *aplease=300*

    この例では、リストはページレイアウトに合うように省略されています。

```
#File
#Device   Device  Mount                  System fsck Mount   Mount
#to Mount to fsck Point                  Type   Pass at Boot Options
#-------- ------- --------------------   ------ ---- ------- ------------
/devices  -       /devices               devfs  -    no      -
/proc     -       /proc                  proc   -    no      -
...
qfs1rac   -       /global/sc-rac/qfs1rac samfs  -    no      shared,...=300
:wq
[qfs1rac-node1]root@solaris:~#
```

11. 高可用性共有ファイルシステムのマウントポイントを作成します。

```
[qfs1rac-node1]root@solaris:~# mkdir -p /global/sc-rac/qfs1rac
[qfs1rac-node1]root@solaris:~#
```

12. プライマリノードで高可用性共有ファイルシステムをマウントします。

```
[qfs1rac-node1]root@solaris:~# mount /global/sc-rac/qfs1rac
[qfs1rac-node1]root@solaris:~#
```

13. 次に、残りの SC-RAC クラスタノード上での潜在的な QFS メタデータサーバーの構成を実行します。

9.5.3. 残りの SC-RAC クラスタノード上での潜在的な QFS メタデータサーバーの構成

クラスタの残りのノードは、潜在的なメタデータサーバーとして機能します。潜在的なメタデータサーバーは、メタデータデバイスにアクセスできるホストであり、メタデータサーバーの役割を担います。そのため、プライマリノード上のアクティブなメタデータサーバーで障害が発生した場合、Solaris Cluster ソフトウェアはセカンダリノードにフェイルオーバーし、潜在的なメタデータサーバーをアクティブにできます。

SC-RAC クラスタ内の残りのノードごとに、次のように進めます。

1. ノードに *root* としてログインします。

   この例では、現在のノードは *qfs1rac-node2* です。

```
[qfs1rac-node2]root@solaris:~#
```

2. */etc/opt/SUNWsamfs/mcf* ファイルをプライマリノードから現在のノードにコピーします。
3. エラーを *mcf* ファイルで確認します。コマンド */opt/SUNWsamfs/sbin/sam-fsd* を実行して、見つかったエラーを修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、ホスト *qfs1rac-node2* 上で *mcf* ファイルを確認します。

```
[qfs1rac-node2]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs1rac-node2]root@solaris:~#
```

4. テキストエディタでオペレーティングシステムの */etc/vfstab* ファイルを開き、新しいファイルシステムの行を開始します。

   この例では、*vi* エディタを使用します。

```
[qfs1rac-node2]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount                  System  fsck  Mount    Mount
#to Mount  to fsck  Point                  Type    Pass  at Boot  Options
#--------  -------  ---------------------- ------  ----  -------  ------------
/devices   -        /devices               devfs   -     no       -
/proc      -        /proc                  proc    -     no       -
...
qfs1rac    -        /global/sc-rac/qfs1rac samfs   -     no
```

5. */etc/vfstab* ファイルの 7 番目の列に、次に一覧表示されているマウントオプションを入力します。次に、ファイルを保存し、エディタを閉じます。

   SC-RAC クラスタ構成では、次のマウントオプションが推奨されます。これらは、この時点で */etc/vfstab* 内で指定することも、またはその方が便利な場合はファイル */etc/opt/SUNWsamfs/samfs.cmd* 内で指定することもできます。

   - *shared*
   - *stripe=1*
   - *sync_meta=1*
   - *mh_write*
   - *qwrite*
   - *forcedirectio*
   - *notrace*
   - *rdlease=300*
   - *wrlease=300*
   - *aplease=300*

   この例では、リストはページレイアウトに合うように省略されています。

```
#File
```

```
#Device    Device  Mount                   System  fsck  Mount   Mount
#to Mount to fsck Point                   Type    Pass  at Boot Options
#--------- ------- ---------------------- ------  ----  ------- ------------
/devices   -       /devices                devfs   -     no      -
/proc      -       /proc                   proc    -     no      -
...
qfs1rac    -       /global/sc-rac/qfs1rac samfs   -     no      shared,...=300
:wq
[qfs1rac-node2]root@solaris:~#
```

6. セカンダリノードで高可用性共有ファイルシステムのマウントポイントを作成します。

```
[qfs1rac-node2]root@solaris:~# mkdir -p /global/sc-rac/qfs1rac
[qfs1rac-node2]root@solaris:~#
```

7. セカンダリノードで高可用性共有ファイルシステムをマウントします。

```
[qfs1rac-node2]root@solaris:~# mount /global/sc-rac/qfs1rac
[qfs1rac-node2]root@solaris:~#
```

8. 次に、SC-RAC メタデータサーバーのフェイルオーバーの構成を実行します。

9.5.4. SC-RAC メタデータサーバーのフェイルオーバーの構成

Solaris Cluster ソフトウェアによって管理されるクラスタ内に Oracle HSM 共有ファイルシステムをホストする場合、Oracle HSM software によって定義されるリソースタイプである *SUNW.qfs* クラスタリソースを作成することで、メタデータサーバーのフェイルオーバーを構成します (詳細は、*SUNW.qfs* のマニュアルページを参照)。SC-RAC 構成のリソースを作成して構成するには、次のように進めます。

1. SC-RAC クラスタのプライマリノードに *root* としてログインします。

   この例では、プライマリノードは *qfs1rac-node1* です。

```
[qfs1rac-node1]root@solaris:~#
```

2. Solaris Cluster ソフトウェアの QFS リソースタイプ *SUNW.qfs* を定義します。コマンド *clresourcetype registerSUNW.qfs* を使用します。

```
[qfs1rac-node1]root@solaris:~# clresourcetype registerSUNW.qfs
[qfs1rac-node1]root@solaris:~#
```

3. 登録ファイルが見つからないために登録が失敗した場合、Solaris Cluster がリソースタイプの登録ファイル */opt/cluster/lib/rgm/rtreg/* を保持するディレクトリに、*/opt/SUNWsamfs/sc/etc/* ディレクトリへのシンボリックリンクを配置します。

   Oracle HSM ソフトウェアのインストール前に、Oracle Solaris Cluster ソフトウェアをインストールしませんでした。通常、Oracle HSM は、インストール時に Solaris Cluster を検出すると、*SUNW.qfs* 登録ファイルの場所を自動的に指定します。そのため、リンクを手動で作成する必要があります。

```
[qfs1rac-node1]root@solaris:~# cd /opt/cluster/lib/rgm/rtreg/
[qfs1rac-node1]root@solaris:~# ln -s /opt/SUNWsamfs/sc/etc/SUNW.qfs SUNW.qfs
[qfs1rac-node1]root@solaris:~#
```

4. QFS メタデータサーバーのリソースグループを作成します。Solaris Cluster コマンド *clresourcegroup create -n node-list group-name* を使用します。ここで *node-list* は、クラスタノードのコンマ区切りリストで、*group-name* は、リソースグループに使用する名前です。

   この例では、SC-RAC サーバーノードをメンバーとして使用してリソースグループ *qfsracrg* を作成します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs1rac-node1]root@solaris:~# clresourcegroup create /
-n qfs1rac-node1,qfs1rac-node2 qfsracrg
[qfs1rac-node1]root@solaris:~#
```

5. 新しいリソースグループで、アクティブなメタデータサーバーの仮想ホスト名を設定します。Solaris Cluster コマンド *clreslogicalhostname create -g group-name* を使用します。ここで *group-name* は QFS リソースグループの名前で、*virtualMDS* は仮想ホスト名です。

   共有ファイルシステムの hosts ファイルで使用したものと同じ仮想ホスト名を使用します。この例では、*qfsracrg* リソースグループに仮想ホスト *qfs1rac-mds* を作成しま

す (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs1rac-node1]root@solaris:~# clreslogicalhostname create /
-g qfsracrg qfs1rac-mds
[qfs1rac-node1]root@solaris:~#
```

6. QFS ファイルシステムリソースをリソースグループに追加します。コマンド *clresource create -g group-name -t SUNW.qfs -x QFSFileSystem=mount-point -y Resource_dependencies=virtualMDS resource-name* を使用します。ここでは:
   - *group-name* は、QFS リソースグループの名前です。
   - *mount-point* は、クラスタ内のファイルシステムのマウントポイントである、システムのルートディレクトリの直下にはないサブディレクトリです。

     QFS 共有ファイルシステムをルートの直下にマウントすると、*SUNW.qfs* リソースタイプの使用時にフェイルオーバーの問題が発生する可能性があります。
   - *virtualMDS* は、アクティブなメタデータサーバーの仮想ホスト名です。
   - *resource-name* は、リソースに指定する名前です。

   この例では、リソースグループ *qfsracrg* 内にタイプ *SUNW.qfs* の *scrac* という名前のリソースを作成します。*SUNW.qfs* 拡張プロパティー *QFSFileSystem* を */global/sc-rac/qfs1rac* マウントポイントに設定します。標準プロパティー *Resource_dependencies* をアクティブなメタデータサーバーの論理ホスト *qfs1rac-mds* に設定します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs1rac-node1]root@solaris:~# create -g qfsracrg -t SUNW.qfs /
-x QFSFileSystem=/global/sc-rac/qfs1rac /
-y Resource_dependencies=qfs1rac-mds scrac
[qfs1rac-node1]root@solaris:~#
```

7. リソースグループをオンラインにします。Solaris Cluster コマンド *clresourcegroup manage group-name* および *clresourcegroup online -emM group-name* を使用します。ここで *group-name* は、QFS リソースグループの名前です。

   この例では、*qfsracrg* リソースグループをオンラインにします。

```
[qfs1rac-node1]root@solaris:~# clresourcegroup manage qfsracrg
```

```
[qfs1rac-node1]root@solaris:~# clresourcegroup online -emM qfsracrg
[qfs1rac-node1]root@solaris:~#
```

8. QFS リソースグループを必ずオンラインにしてください。Solaris Cluster コマンド *clresourcegroup status* を使用します。

   この例では、*qfsracrg* リソースグループは、プライマリノード *qfs1rac-node1* 上では *online* です。

```
[qfs1rac-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsracrg    qfs1rac-node1   No          Online
            qfs1rac-node2   No          Offline
            qfs1rac-node3   No          Offline
[qfs1rac-node1]root@solaris:~#
```

9. リソースグループを正しくフェイルオーバーさせます。リソースグループをセカンダリノードに移動します。Solaris Cluster コマンド *clresourcegroup switch -n node2 group-name* を使用します。ここで *node2* は、セカンダリノードの名前で、*group-name* は、HA-SAM リソースグループに選択した名前です。次に、*clresourcegroup status* を使用して結果を確認します。

   この例では、*qfsracrg* リソースグループを *qfs1rac-node2* および *qfs1rac-node3* に移動して、指定したノードでリソースグループがオンラインになることを確認します。

```
[qfs1rac-node1]root@solaris:~# clresourcegroup switch -n qfs1rac-node2 qfsracrg
[qfs1rac-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsracrg    qfs1rac-node1   No          Offline
            qfs1rac-node2   No          Online
            qfs1rac-node3   No          Offline
[qfs1rac-node1]root@solaris:~# clresourcegroup switch -n qfs1rac-node3 qfsracrg
[qfs1rac-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
```

```
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
qfsracrg    qfs1rac-node1   No          Offline
            qfs1rac-node2   No          Offline
            qfs1rac-node3   No          Online
[qfs1rac-node1]root@solaris:~#
```

10. リソースグループをプライマリノードに戻します。Solaris Cluster コマンド *clresourcegroup switch -n node1 group-name* を使用します。ここで *node1* は、プライマリノードの名前で、*group-name* は、HA-SAM リソースグループに選択した名前です。次に、*clresourcegroup status* を使用して結果を確認します。

    この例では、*qfsracrg* リソースグループを *qfs1rac-node1* に正常に戻します。

```
[qfs1rac-node1]root@solaris:~# clresourcegroup switch -n qfs1rac-node1 qfsracrg
[qfs1rac-node1]root@solaris:~# clresourcegroup status
=== Cluster Resource Groups ===
Group Name  Node Name       Suspended   Status
----------  -------------   ---------   ------
samr        qfs1rac-node1   No          Online
            qfs1rac-node2   No          Offline
            qfs1rac-node3   No          Offline
[qfs1rac-node1]root@solaris:~#
```

11. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。
12. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

9.5.5. ソフトウェア RAID ストレージを使用した SC-RAC ノード上での QFS メタデータサーバーの構成

高可用性ファイルシステムには、冗長なプライマリストレージデバイス上のデータとメタデータを格納する必要があります。冗長なディスク配列ハードウェアは、メタデータには RAID-1 または RAID-10、データには RAID-5 を使用してこの冗長性を提供できます。ただし、標準のデュアルポート SCSI ディスクデバイスまたは JBOD (*just a bunch of disks*) 配列をプライマリストレージとして使用する必要がある場合、必要な冗長性をソフトウェアで提供する必要があります。

このため、SC-RAC 構成では、Oracle Solaris Volume Manager (SVM) マルチ所有者ディスクセットに基づくソフトウェアの RAID 構成がサポートされます。このセクションでは、このタイプの SC-RAC ファイルシステム構成を設定する際に行う必要がある基本的な手順の概要について説明します。

冗長なストレージアレイを管理するためには、Solaris Volume Manager のみを使用してください。異なるデバイス上のストレージを連結しないでください。これを行うと、入出力がコンポーネントデバイスに非効率的に分散され、QFS ファイルシステムのパフォーマンスが低下します。

次のタスクを実行します。

- Solaris 11+ 上での Solaris Volume Manager のインストール
- Solaris Volume Manager のマルチ所有者ディスクグループの作成
- QFS データおよびメタデータのミラー化ボリュームの作成
- ミラー化ボリュームを使用した SC-RAC クラスタ上での QFS 共有ファイルシステムの作成

9.5.5.1. Solaris 11+ 上での Solaris Volume Manager のインストール

Solaris Volume Manager (SVM) は Solaris 11 以降の Solaris には付属しなくなりました。ただし、Solaris Cluster 4 ソフトウェアでは引き続き Solaris Volume Manager がサポートされます。そのため、ソフトウェアを使用するには、Solaris 10 9/10 リリースに組み込まれているバージョンをダウンロードしてインストールする必要があります。クラスタ内のノードごとに、次のように進めます。

1. ノードに *root* としてログインします。

   次の例では、Solaris Image Packaging System (IPS) を使用してクラスタノード *qfs2rac-node1* を構成します。

```
[qfs2rac-node1]root@solaris:~#
```

2. ローカルで入手可能な Solaris Volume Manager (SVM) パッケージを確認します。コマンド *pkg info svm* を使用します。

```
[qfs2rac-node1]root@solaris:~# pkg info svm
pkg: info: no packages matching the following patterns you specified are
```

```
installed on the system.  Try specifying -r to query remotely:
        svm
[qfs2rac-node1]root@solaris:~#
```

3. パッケージがローカルで見つからない場合は、Solaris Image Packaging System (IPS) リポジトリを確認します。コマンド *pkg -r svm* を使用します。

```
[qfs2rac-node1]root@solaris:~# pkg -r svm
          Name: storage/svm
       Summary: Solaris Volume Manager
   Description: Solaris Volume Manager commands
      Category: System/Core
         State: Not installed
     Publisher: solaris
       Version: 0.5.11
 Build Release: 5.11
        Branch: 0.175.0.0.0.2.1
Packaging Date: October 19, 2011 06:42:14 AM
          Size: 3.48 MB
          FMRI: pkg://solaris/storage/svm@0.5.11,5.11-0.175.0.0.0.2.1:20111019T064214Z
[qfs2rac-node1]root@solaris:~#
```

4. パッケージをインストールします。コマンド *pkg install storage/svm* を使用します。

```
[qfs2rac-node1]root@solaris:~# pkg install storage/svm
           Packages to install:   1
       Create boot environment:  No
Create backup boot environment: Yes
            Services to change:   1
DOWNLOAD      PKGS       FILES    XFER (MB)
Completed      1/1     104/104      1.6/1.6
PHASE            ACTIONS
Install Phase    168/168
PHASEITEMS
Package State Update Phase         1/1
Image State Update Phase           2/2
[qfs2rac-node1]root@solaris:~#
```

5. インストールが終了したら、*metadb* の場所を確認します。コマンド *which metadb* を使用します。

```
[qfs2rac-node1]root@solaris:~# which metadb
/usr/sbin/metadb
[qfs2rac-node1]root@solaris:~#
```

6. インストールを確認します。コマンド *metadb* を使用します。

```
[qfs2rac-node1]root@solaris:~# metadb
[qfs2rac-node1]root@solaris:~#
```

7. *metadb* がエラーを返す場合は、*kernel/drv/md.conf* ファイルが存在するかどうかを確認します。

```
[qfs2rac-node1]root@solaris:~# metadb
metadb: <HOST>: /dev/md/admin: No such file or directory
[qfs2rac-node1]root@solaris:~# ls -l /kernel/drv/md.conf
-rw-r--r--   1 root     sys          295 Apr 26 15:07 /kernel/drv/md.conf
[qfs2rac-node1]root@solaris:~#
```

8. *kernel/drv/md.conf* ファイルが存在しない場合は、作成します。*root* をファイルの所有者にして、*sys* をグループの所有者にします。権限は *644* に設定します。

   この例では、*vi* エディタを使用してファイルを作成します。ファイルの内容は次のようになります。

```
[qfs2rac-node1]root@solaris:~# vi kernel/drv/md.conf
#####################################################
#pragma ident   "@(#)md.conf    2.1   00/07/07 SMI"
#
# Copyright (c) 1992-1999 by Sun Microsystems, Inc.
# All rights reserved.
#
name="md" parent="pseudo" nmd=128 md_nsets=4;
#####################################################
:wq
```

```
[qfs2rac-node1]root@solaris:~# chown root:sys kernel/drv/md.conf
[qfs2rac-node1]root@solaris:~# chmod 644
[qfs2rac-node1]root@solaris:~#
```

9. *md.conf* ファイルを動的に再スキャンして、デバイスツリーが更新されていることを確認します。コマンド *update_drv -f md* を使用します。

   この例では、デバイスツリーが更新されます。したがって、Solaris Volume Manager はインストールされています。

```
[qfs2rac-node1]root@solaris:~# update_drv -f md
[qfs2rac-node1]root@solaris:~# ls -l  /dev/md/admin
lrwxrwxrwx   1 root root 31 Apr 20 10:12 /dev/md/admin -> ../../devices/pseudo/md@0:admin
[qfs2rac-node1]root@solaris:~#
```

10. 次に、Solaris Volume Manager のマルチ所有者ディスクグループの作成を実行します。

9.5.5.2. Solaris Volume Manager のマルチ所有者ディスクグループの作成

1. SC-RAC 構成内のすべてのノードに *root* としてログインします。

   この例では、ノード *qfs2rac-node1* にログインします。次に、新しい端末ウィンドウを開き、*ssh* を使用してノード *qfs2rac-node2* および *qfs2rac-node3* にログインします。

```
[qfs2rac-node1]root@solaris:~#

[qfs2rac-node1]root@solaris:~# ssh root@qfs2rac-node2
Password:
[qfs2rac-node2]root@solaris:~#

[qfs2rac-node1]root@solaris:~# ssh root@qfs2rac-node3
Password:
[qfs2rac-node3]root@solaris:~#
```

2. Oracle Solaris Cluster 4.*x* を Solaris 11.*x* 以降で使用していて、各ノードで Solaris 11+ 上での Solaris Volume Manager のインストールをまだ行なっていない場合は、これを行なってから、先に進みます。

Solaris 11 以降、Solaris Volume Manager はデフォルトではインストールされません。

3. 各ノード上で新しい状態データベースデバイスを接続して、状態データベースの複製を 3 つ作成します。コマンド *metadb -a -f -c3 device-name* を使用します。ここで、*device-name* は、*cXtYdYsZ* 形式の物理デバイス名です。

   Solaris Cluster デバイス ID (DID) は使用しないでください。物理デバイス名を使用してください。この例では、3 つすべてのクラスタノード上に状態データベースデバイスを作成します。

```
[qfs2rac-node1]root@solaris:~# metadb -a -f -c3 /dev/rdsk/c0t0d0

[qfs2rac-node2]root@solaris:~# metadb -a -f -c3 /dev/rdsk/c0t6d0

[qfs2rac-node3]root@solaris:~# metadb -a -f -c3 /dev/rdsk/c0t4d0
```

4. 1 つのノード上に Solaris Volume Manager のマルチ所有者ディスクグループを 1 つ作成します。コマンド *metaset -sdiskset -M -a -h host-list* を使用します。ここで、*host-list* は、所有者の空白文字区切りリストです。

   Solaris Volume Manager では、ディスクセットごとに最大 4 つのホストがサポートされます。この例では、*qfs2rac-node1* でディスクグループ *datadisks* を作成して、3 つのノード *qfs2rac-node1*、*qfs2rac-node2*、および *qfs2rac-node3* を所有者として指定します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs2rac-node1]root@solaris:~# metaset -s datadisks -M -a -h qfs2rac-node1 /
qfs2rac-node2 qfs2rac-node3
```

5. ノードの 1 つでデバイスを表示します。Solaris Cluster コマンド *cldevice list -n -v* を使用します。

```
[qfs2rac-node1]root@solaris:~# cldevice list -n -v
DID Device  Full Device Path
----------  ----------------
d13         qfs2rac-node1:/dev/rdsk/c6t600C0FF00000000000332B62CF3A6B00d0
d14         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E950F1FD9600d0
d15         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E9124FAF9C00d0
```

```
...
[qfs2rac-node1]root@solaris:~#
```

6. *cldevice list -n -v* コマンドの出力で、ミラー化するデバイスを選択します。

   この例では、4 つのミラー用に 4 つのデバイスのペア *d21* と *d13*、*d14* と *d17*、*d23* と *d16*、および *d15* と *d19* を選択します。

```
[qfs2rac-node1]root@solaris:~# cldevice list -n -v
DID Device  Full Device Path
----------  ----------------
d13         qfs2rac-node1:/dev/rdsk/c6t600C0FF00000000000332B62CF3A6B00d0
d14         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E950F1FD9600d0
d15         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E9124FAF9C00d0
d16         qfs2rac-node1:/dev/rdsk/c6t600C0FF00000000000332B28488B5700d0
d17         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000086DB474EC5DE900d0
d18         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E975EDA6A000d0
d19         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000086DB47E331ACF00d0
d20         qfs2rac-node1:/dev/rdsk/c6t600C0FF0000000000876E9780ECA8100d0
d21         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000004CAD5B68A7A100d0
d22         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000086DB43CF85DA800d0
d23         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000004CAD7CC3CDE500d0
d24         qfs2rac-node1:/dev/rdsk/c6t600C0FF000000000086DB4259B272300d0
....
[qfs2rac-node1]root@solaris:~#
```

7. 選択したデバイスを同じノード上のディスクセットに追加します。コマンド *metaset -a devicelist* を使用します。ここで、*devicelist* は、1 つ以上のクラスタデバイス ID のスペース区切りリストです。

   この例では、一覧表示されているディスクをマルチ所有者ディスクセット *dataset1* に追加します (次のコマンドは 1 行で入力します。改行はバックスラッシュ文字でエスケープされます)。

```
[qfs2rac-node1]root@solaris:~# metaset -s dataset1 -M -a -h /dev/did/rdsk/d21 / /dev/did/rdsk/d13 /dev/did/
rdsk/d14 /dev/did/rdsk/d17 /dev/did/rdsk/d23 /
/dev/did/rdsk/d16 /dev/did/rdsk/d15 /dev/did/rdsk/d19
[qfs2rac-node1]root@solaris:~#
```

8. 次に、QFS データおよびメタデータのミラー化ボリュームの作成を実行します。

9.5.5.3. QFS データおよびメタデータのミラー化ボリュームの作成

1. コンポーネント間の関係を明確な状態に保つには、作成する RAID-0 論理ボリュームと RAID-1 ミラーの命名スキームを決定します。

   一般に、RAID-1 ミラーには *dn* (*n* は整数) という名前が付けられます。RAID-1 ミラーを構成する RAID-0 ボリュームには、*dnX* という名前が付けられます。ここで、*X* は、ミラー内でのデバイスの位置を表す整数 (通常、双方向ミラーの場合は *0* または *1*) です。

   この手順全体の例では、RAID-0 論理ボリュームのペアから双方向 RAID-1 ミラーを作成します。そのため、ミラーには、*d1*、*d2*、*d3*、*d4* などの名前を付けます。その後、RAID-0 ボリュームの各ペアには、そのペアを含む RAID-1 ミラーに基づいて*d10* と *d11*、*d20* と *d21*、*d30* と *d31*、*d40* と *d41* のように名前を付けます。
2. マルチ所有者ディスクセットを作成したノードにログインします。*root* としてログインします。

   上の例では、*qfs2rac-node1* 上にディスクセットを作成しました。

```
[qfs2rac-node1]root@solaris:~#
```

3. 最初の RAID-0 論理ボリュームを作成します。コマンド *metainit -s diskset-name device-name number-of-stripes components-per-stripe component-names* を使用します。ここでは:
   - *diskset-name* は、ディスクセットに選択した名前です。
   - *device-name* は、RAID-0 論理ボリュームに選択した名前です。
   - *number-of-stripes* は *1* です。
   - *components-per-stripe* は *1* です。
   - *component-name* は、RAID-0 ボリュームで使用するディスクセットコンポーネントのデバイス名です。

   この例では、マルチ所有者ディスクセット *dataset1* でクラスタ (DID)デバイス */dev/did/dsk/d21s0* を使用して、RAID-0 論理ボリューム *d10* を作成します。

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d10 1 1 /dev/did/dsk/d21s0
```

```
[qfs2rac-node1]root@solaris:~#
```

4. 残りの RAID-0 論理ボリュームを作成します。

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d11 1 1 /dev/did/dsk/d13s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d20 1 1 /dev/did/dsk/d14s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d21 1 1 /dev/did/dsk/d17s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d30 1 1 /dev/did/dsk/d23s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d31 1 1 /dev/did/dsk/d16s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d40 1 1 /dev/did/dsk/d15s0
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d41 1 1 /dev/did/dsk/d19s0
...
[qfs2rac-node1]root@solaris:~#
```

5. 最初の RAID-1 ミラーを作成します。コマンド *metainit -s diskset-name RAID-1-mirrorname -m RAID-0-volume0* を使用します。ここでは:
   - *diskset-name* は、マルチ所有者ディスクセットの名前です。
   - *RAID-1-mirrorname* は、RAID-1 ミラー化ボリュームの名前です。
   - *RAID-0-volume0* は、ミラーに追加する最初の RAID-0 論理ボリュームです。

   この例では、ミラー *d1* を作成して、最初の RAID-0 ボリューム *d10* をミラー内に追加します。

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d1 -m d10
[qfs2rac-node1]root@solaris:~#
```

6. 残りの RAID-0 ボリュームを最初の RAID-1 ミラーに追加します。コマンド *metattach -s diskset-name RAID-1-mirrorname RAID-0-volume* を使用します。ここでは:
   - *diskset-name* は、マルチ所有者ディスクセットの名前です
   - *RAID-1-mirrorname* は、RAID-1 ミラー化ボリュームの名前です
   - *RAID-0-volume* は、ミラーに追加する RAID-0 論理ボリュームです。

   この例では、*d1* は双方向ミラーであるため、単一の RAID-0 ボリューム *d11* を追加します。

```
[qfs2rac-node1]root@solaris:~# metattach -s dataset1 d11 d1
[qfs2rac-node1]root@solaris:~#
```

7. 残りのミラーを作成します。

   この例では、ミラー *d2*、*d3*、*d4* などを作成します。

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d2 -m d20
[qfs2rac-node1]root@solaris:~# metattach -s dataset1 d21 d2
[qfs2rac-node1]root@solaris:~# metainit -s dataset2 d3 -m d30
[qfs2rac-node1]root@solaris:~# metattach -s dataset2 d31 d3
[qfs2rac-node1]root@solaris:~# metainit -s dataset2 d4 -m d40
[qfs2rac-node1]root@solaris:~# metattach -s dataset2 d41 d4
...
[qfs2rac-node1]root@solaris:~#
```

8. QFS ファイルシステムのメタデータを保持するミラーを選択します。

   次の例では、ミラー *d1* および *d2* を選択します。

9. 選択したミラーで、QFS メタデータを保持するソフトパーティションを作成します。ミラーごとに、コマンド *metainit -s diskset-name partition-name -p RAID-1-mirrorname size* を使用します。ここでは:

   - *diskset-name* は、マルチ所有者ディスクセットの名前です。
   - *partition-name* は、新しいパーティションの名前です。
   - *RAID-1-mirrorname* は、ミラーの名前です。
   - *size* は、パーティションのサイズです。

   この例では、2 つの 500G バイトのパーティション *d53* および *d63* を、ミラー *d1* およびミラー *d2* 上にそれぞれ作成します。

```
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d53 -p d1 500g
[qfs2rac-node1]root@solaris:~# metainit -s dataset1 d63 -p d2 500g
```

10. 次に、ミラー化ボリュームを使用した SC-RAC クラスタ上での QFS 共有ファイルシステムの作成を実行します。

9.5.5.4. ミラー化ボリュームを使用した SC-RAC クラスタ上での QFS 共有ファイルシステムの作成

1. まだ実行していない場合は、手順「すべての SC-RAC クラスタノードにおける QFS 共有ファイルシステムの hosts ファイルの作成」を実行します。終了したら、ここに戻ります。
2. SC-RAC クラスタのプライマリノードと QFS 共有ファイルシステムのアクティブなメタデータサーバーの両方としての役割を果たすクラスタノードを選択します。*root* としてログインします。

   この例では、ノード *qfs2rac-node1* を選択します。

```
[qfs2rac-node1]root@solaris:~#
```

3. プライマリノード上で、高パフォーマンスの *ma* 共有ファイルシステムを作成します。Solaris Volume Manager のミラー化ディスクボリュームを *mm* メタデータデバイスおよび *mr* データデバイスとして使用します。テキストエディタで */etc/opt/SUNWsamfs/mcf* ファイルを開いて、必要な編集を行なって、ファイルを保存します。

   この例では、*vi* テキストエディタを使用して、ファイルシステム *qfs2rac* を作成します。ミラー化ボリューム *d1* および *d2* 上のパーティションは、ファイルシステムの 2 つの *mm* メタデータデバイス *110* および *120* として機能します。ミラー化ボリューム *d3* および *d4* は、ファイルシステムの 2 つの *mr* データデバイス *130* および *140* として機能します。

```
[qfs2rac-node1]root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# /etc/opt/SUNWsamfs/mcf file:
#
# Equipment               Equipment Equipment Family   Device  Additional
# Identifier              Ordinal   Type      Set      State   Parameters
# ----------------------- --------- --------  -------  ------  ----------
qfs2rac                   100         ma        qfs2rac  on      shared
 /dev/md/dataset1/dsk/d53  110         mm        qfs2rac  on
 /dev/md/dataset1/dsk/d63  120         mm        qfs2rac  on
 /dev/md/dataset1/dsk/d3   130         mr        qfs2rac  on
 /dev/md/dataset1/dsk/d4   140         mr        qfs2rac  on
:wq
[qfs2rac-node1]root@solaris:~#
```

4. エラーを *mcf* ファイルで確認します。コマンド */opt/SUNWsamfs/sbin/sam-fsd* を使用して、見つかったエラーを修正します。

   sam-fsd コマンドは、Oracle HSM 構成ファイルを読み取り、ファイルシステムを初期化します。エラーが発生した場合は停止します。この例では、ホスト *qfs2rac-node1* 上で *mcf* ファイルを確認します。

```
[qfs2rac-node1]root@solaris:~# sam-fsd
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
[qfs2rac-node1]root@solaris:~#
```

5. ファイルシステムを作成します。コマンド */opt/SUNWsamfs/sbin/sammkfs -S family-set-name* を使用します。ここで *family-set-name* は、ファイルシステムの装置 ID です。

   sammkfs コマンドは、*hosts.family-set-name* および *mcf* ファイルを読み取って、指定されたプロパティーを使用して共有ファイルシステムを作成します。

```
[qfs2rac-node1]root@solaris:~# sammkfs -S qfs2rac
Building 'qfs2rac' will destroy the contents of devices:
  ...
Do you wish to continue? [y/N]yes ...
[qfs2rac-node1]root@solaris:~#
```

6. テキストエディタでオペレーティングシステムの */etc/vfstab* ファイルを開き、新しいファイルシステムの行を開始します。最初の列にファイルシステム名、空白文字、2 番目の列にハイフン、さらに空白を入力します。

   この例では、*vi* テキストエディタを使用します。*qfs2rac* ファイルシステムの行を開始します。ハイフンは、オペレーティングシステムが UFS ツールを使用してファイルシステムの整合性チェックを試行しないようにします。

```
[qfs2rac-node1]root@solaris:~# vi /etc/vfstab
#File
```

```
#Device    Device  Mount                   System fsck Mount   Mount
#to Mount to fsck Point                   Type   Pass at Boot Options
#--------- ------- ---------------------- ------ ---- ------- ------------
/devices   -       /devices                devfs  -    no      -
/proc      -       /proc                   proc   -    no      -
...
qfs2rac    -
```

7. */etc/vfstab* ファイルの 3 番目の列に、クラスタを基準とした相対的なファイルシステムのマウントポイントを入力します。システムのルートディレクトリの直下にはないマウントポイントサブディレクトリを指定します。

   QFS 共有ファイルシステムをルートの直下にマウントすると、*SUNW.qfs* リソースタイプの使用時にフェイルオーバーの問題が発生する可能性があります。この例では、*qfs2rac* ファイルシステムのマウントポイントは */global/sc-rac/qfs2rac* です。

```
#File
#Device    Device  Mount                   System fsck Mount   Mount
#to Mount to fsck Point                   Type   Pass at Boot Options
#--------- ------- ---------------------- ------ ---- ------- ------------
/devices   -       /devices                devfs  -    no      -
/proc      -       /proc                   proc   -    no      -
...
qfs2rac    -       /global/sc-rac/qfs2rac samfs  -    no
```

8. */etc/vfstab* ファイルの 4 番目の列に、ファイルシステムタイプ (*samfs*) を入力します。

```
#File
#Device    Device  Mount                   System fsck Mount   Mount
#to Mount to fsck Point                   Type   Pass at Boot Options
#--------- ------- ---------------------- ------ ---- ------- ------------
/devices   -       /devices                devfs  -    no      -
/proc      -       /proc                   proc   -    no      -
...
qfs2rac    -       /global/sc-rac/qfs2rac samfs
```

9. */etc/vfstab* ファイルの 5 番目の列に、*fsck* パスオプション (-) を入力します。

```
#File
#Device   Device  Mount                  System fsck Mount   Mount
#to Mount to fsck Point                  Type   Pass at Boot Options
#-------- ------- ---------------------- ------ ---- ------- ------------
/devices  -       /devices               devfs  -    no      -
/proc     -       /proc                  proc   -    no      -
...
qfs2rac   -       /global/sc-rac/qfs2rac samfs  -
```

10. */etc/vfstab* ファイルの 6 番目の列に、ブート時のマウントオプション (*no*) を入力します。

```
#File
#Device   Device  Mount                  System fsck Mount   Mount
#to Mount to fsck Point                  Type   Pass at Boot Options
#-------- ------- ---------------------- ------ ---- ------- ------------
/devices  -       /devices               devfs  -    no      -
/proc     -       /proc                  proc   -    no      -
...
qfs2rac   -       /global/sc-rac/qfs2rac samfs  -    no
```

11. */etc/vfstab* ファイルの 7 番目の列に、*sw_raid* マウントオプションと、SC-RAC 構成で推奨されるマウントオプションを入力します。次に、ファイルを保存し、エディタを閉じます。

    次のマウントオプションが推奨されます。これらは、この時点で */etc/vfstab* 内で指定することも、またはその方が便利な場合はファイル */etc/opt/SUNWsamfs/samfs.cmd* 内で指定することもできます。

    - *shared*
    - *stripe=1*
    - *sync_meta=1*
    - *mh_write*
    - *qwrite*
    - *forcedirectio*

- *notrace*
- *rdlease=300*
- *wrlease=300*
- *aplease=300*

この例では、マウントオプションリストはページレイアウトに合うように省略されています。

```
#File
#Device    Device  Mount                  System fsck Mount   Mount
#to Mount to fsck Point                  Type   Pass at Boot Options
#-------- ------- ---------------------- ------ ---- ------- ------------
/devices   -       /devices               devfs  -    no      -
/proc      -       /proc                  proc   -    no      -
...
qfs2rac    -       /global/sc-rac/qfs2rac samfs  -    no      shared,...sw_raid
:wq
[qfs2rac-node1]root@solaris:~#
```

12. 高可用性共有ファイルシステムのマウントポイントを作成します。

```
[qfs2rac-node1]root@solaris:~# mkdir -p /global/sc-rac/qfs2rac
[qfs2rac-node1]root@solaris:~#
```

13. プライマリノードで高可用性共有ファイルシステムをマウントします。

```
[qfs2rac-node1]root@solaris:~# mount /global/sc-rac/qfs2rac
[qfs2rac-node1]root@solaris:~#
```

14. 2 番目のノードを設定します。「残りの SC-RAC クラスタノード上での潜在的な QFS メタデータサーバーの構成 」の手順を使用します。
15. フェイルオーバーを構成します。「SC-RAC メタデータサーバーのフェイルオーバーの構成」の手順を使用します。その後、ここに戻ります。

    Solaris Volume Manager のミラー化ボリュームを使用して、SC-RAC の構成を正常に行いました。
16. サイドバンドデータベース機能を使用する計画がある場合、10章「*レポートデータベースの構成*」に進みます。

17. それ以外の場合は、11章「*通知とロギングの構成* 」に進みます。

10

レポートデータベースの構成

Oracle HSM では、指定されたファイルシステムのすべてのファイルの現在のメタデータ情報を格納する、オプションのレポートデータベースがサポートされています。この *サイドバンド* データベースは、ファイルやファイルシステムのアクティビティーの管理とレポートに非常に役立つ場合があります。

Oracle HSM サイドバンドデータベースの実装は簡単です。`samdb` コマンドを使用して、提供されるデータベーススキーマ (または代替のカスタムデータベーススキーマ) および `samfsdump` コマンドによって生成される回復ポイントファイルによって、MySQL データベースを作成および構成します。Oracle HSM デーモンプロセスは、対応するファイルシステムが変更されると、データベースを自動的に更新します。追加の `samdb` コマンドを使用すると、データベースのクエリーや管理を実行できます。コマンドやオプションの完全な詳細については、`samdb` および `samdb.conf` のマニュアルページを参照してください。

サイドバンドデータベース機能を使用するには、次のタスクを実行します。

- MySQL サーバーソフトウェアのインストールおよび構成
- データベースロードファイルの作成
- サイドバンドデータベースの作成

10.1. MySQL サーバーソフトウェアのインストールおよび構成

`samdb` のレポート機能を有効にするには、MySQL データベースをインストールして構成する必要があります。次のように進めます。

1. `http://dev.mysql.com/doc/` から『*MySQL Reference Manual*』をダウンロードします。

   `samdb` のレポートを有効にする際に必要な MySQL タスクを特定するには、次の手順を使用します。ただし、次の手順はもともと完全ではなく、正式なものでもありません。これらは、『*MySQL Reference Manual*』を参照する際のガイドとして使用してください。
2. MySQL サーバーをホストするシステムに `root` としてログインします。

MySQL サーバーは、Oracle HSM メタデータサーバーホストにインストールすることも、独立した Solaris または Linux ホストにインストールすることもできます。

この例では、Solaris ホスト *samsql* に MySQL をインストールします。

```
[samsql]root@solaris:~#
```

3. 『*MySQL Reference Manual*』の指示に従って、MySQL サーバーソフトウェアをダウンロードしてインストールします。自動起動を有効にします。
4. *mysql* クライアントと *root* ユーザーアカウントを使用して MySQL サーバーに接続します。コマンド *mysql --user=root -p* を使用します。プロンプトが表示されたら、インストール中に *root* ユーザーに割り当てたパスワードを入力します。

   mysql コマンドシェルが起動します。

```
[samsql]root@solaris:~# mysql --user=root -p
Enter Password:
mysql>
```

5. Oracle HSM MySQL ユーザーを作成します。コマンド *CREATE USER 'user_name'@'host_name' IDENTIFIED BY 'user-password'* を使用します。ここでは:
   - *user_name* は Oracle HSM MySQL ユーザーの名前です
   - MySQL を Oracle HSM メタデータサーバーホストにインストールした場合、*host_name* は *localhost* です。それ以外の場合は、メタデータサーバーのホスト名または IP アドレスです。
   - *user-password* は、Oracle HSM MySQL ユーザーに割り当てるパスワードです。

   この例では、Oracle HSM メタデータサーバー *samqfs1mds* 上にユーザー *samsql* を作成します。設定するユーザーパスワード *samsqluserpassw0rd* はデモ用です (これは、本番データベース用としてはセキュアな選択肢ではありません)。

```
[samsql]root@solaris:~# mysql --user=root
Enter Password:
mysql> CREATE USER 'samsql'@'samqfs1mds' IDENTIFIED BY 'samsqluserpassw0rd'
mysql>
```

6. Oracle HSM ユーザーに必要な権限を付与します。コマンド *GRANT CREATE,DROP,INDEX,SELECT,INSERT,UPDATE,DELETE ON host_name TO 'user_name'@'host_name'* を使用します。

   この例では、メタデータサーバー *samqfs1mds* 上のユーザー *samsql* に権限を付与します。

```
[samsql]root@solaris:~# mysql --user=root -p
Enter Password:
mysql> CREATE USER 'samsql'@'host_name' IDENTIFIED BY 'samsqluserpassw0rd'
mysql> GRANT CREATE,DROP,INDEX,SELECT,INSERT,UPDATE,DELETE ON samqfs1mds TO /
'samsql'@'samqfs1mds'
mysql>
```

7. MySQL のコマンドインタフェースを閉じ、オペレーティングシステムのコマンドシェルに戻ります。MySQL コマンド *QUIT* を使用します。

```
[samsql]root@solaris:~# mysql --user=root -p
Enter Password:
mysql> CREATE USER 'samsql'@'host_name' IDENTIFIED BY 'samsqluserpassw0rd'
mysql> GRANT CREATE,DROP,INDEX,SELECT,INSERT,UPDATE,DELETE ON samqfs1mds TO /
'samsql'@'samqfs1mds'
mysql> QUIT
Bye
root@solaris:~#
```

8. 次に、データベースロードファイルの作成を実行します。

10.2. データベースロードファイルの作成

1. Oracle HSM メタデータサーバーホストに *root* としてログインします。

   この例では、ホスト *samqfs1mds* にログインします。

```
[samqfs1mds]root@solaris:~#
```

2. すでに最新の回復ポイントファイルがある場合は、回復ポイントファイルの内容からデータベースロードファイルを生成します。コマンド *samfsrestore -SZ output-path-name -f recoverypoint-file* を使用します。ここでは:

- *-f* は、入力ファイルのパス名とファイル名として *recoverypoint-file* を指定します。
- *-SZ* を指定した場合、コマンドは回復ポイントファイルをスキャンし、*output-path-name* に指定されたパス名とファイル名のあるデータベースロードファイルを出力します。

詳細については、*samfsdump* のマニュアルページを参照してください。

この例では、*samqfs1* ファイルシステムの構成時にスケジュールした日次回復ポイントファイル */zfs1/hsmqfs1_recovery/140129* (「ファイルシステムの保護の構成」を参照) を使用します。データベースロードファイル */root/hsmqfs1dataload* に出力を送信します (次のコマンドは 1 行で入力し、改行はバックスラッシュ文字でエスケープします)。

```
[samqfs1mds]root@solaris:~# samfsrestore -SZ /root/hsmqfs1dataload -f /
/zfs1/hsmqfs1_recovery/140129
```

3. 最新の回復ポイントファイルがない場合は、データベースロードファイルをここで作成します。Oracle HSM ファイルシステムのルートディレクトリに移動します。次に、コマンド *samfsdump -SZ output-path-name* を使用します。

   詳細については、*samfsdump* のマニュアルページを参照してください。この例では、*/hsmqfs1* ディレクトリに移動します。データベースロードファイル */root/hsmqfs1dataload* に出力を送信します。

```
[samqfs1mds]root@solaris:~# cd /hsmqfs1
[samqfs1mds]root@solaris:~# samfsdump -SZ /root/hsmqfs1dataload
```

4. 次に、サイドバンドデータベースの作成を実行します。

10.3. サイドバンドデータベースの作成

1. MySQL サーバーのホストに *root* としてログインします。

   この例では、MySQL サーバーは Solaris ホスト *samqfs1mds* 上にホストされています。

```
[samqfs1mds]root@solaris:~#
```

2. テキストエディタでファイル */etc/opt/SUNWsamfs/samdb.conf* を開きます。

次の例では、*vi* エディタを使用します。まず、見出し行をコメントとして追加します。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
```

3. *samdb.conf* ファイルの最初の列にはファイルシステムのファミリセット名を入力し、そのあとに列区切り文字のコロン (*:*) を入力します。

   この例では、ファミリセット名 *samqfs1* を入力します。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:
```

4. 2 番目の列には MySQL データベースサーバーのホスト名を入力し、そのあとに列区切り文字のコロン (*:*) を入力します。

   この例では、Oracle HSM メタデータサーバーホスト *samqfs1mds* 上でデータベースサーバーを共同でホストしています。したがって、ホスト名 *localhost* を入力します。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:
```

5. 3 番目の列には、Oracle HSM ソフトウェアが MySQL データベースへのアクセス時に使用するユーザー名を入力し、そのあとに列区切り文字のコロン (*:*) を入力します。

   この例では、データベースへのログイン用としてユーザー *samqfs* を作成しました。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:
```

6. 4 番目の列には、Oracle HSM ソフトウェアが MySQL データベースへのアクセス時に使用するパスワードを入力し、そのあとに列区切り文字のコロン (*:*) を入力します。

この例では、ダミーのパスワード *P^ssw0rd* を使用します。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:P^ssw0rd:
```

7. 5 番目の列には MySQL データベースの名前を入力し、そのあとに列区切り文字のコロン (*:*) を入力します。

   この例では、データベースの名前を *samqfs1db* にします。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:P^ssw0rd:samqfs1db:
```

8. 6 番目の列にはデータベースサーバーの TCP/IP ポートを入力し、そのあとに列区切り文字のコロン (*:*) を入力します。

   この例では、*0* (ゼロ) を入力します。リモートサーバーを使用している場合、ゼロ (または空) の値がデフォルトポート *3306* を指定します。しかし、ここでは *localhost* を使用しているので、このゼロは単なるプレースホルダーです。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:P^ssw0rd:samqfs1db:0:
```

9. 7 番目の列には MySQL クライアントフラグを入力し、そのあとに列区切り文字のコロン (*:*) を入力します。

   MySQL クライアントフラグは通常、*0* (ゼロ) に設定されます。しかし、さまざまな組み合わせの値を設定することで、特定の MySQL 機能を有効にできます。詳細は、*mysql_real_connect()* 関数についての MySQL のドキュメントを参照してください。

   この例では、*0* (ゼロ) を入力します。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:Pˆssw0rd:samqfs1db:0:0:
```

10. 8 番目の最後の列には、Oracle HSM ファイルシステムのマウントポイントを入力します。ファイルを保存して、エディタを閉じます。

    この例では、ファイルシステムは */hsmqfs/hsmqfs1* にマウントされます。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/samdb.conf
# /etc/opt/SUNWsamfs/samdb.conf
 #FS_NAME:HOST:USER:PASSWORD:NAME:PORT:CLIENT_FLAG:MOUNT_POINT
samqfs1:localhost:samqfs:Pˆssw0rd:samqfs1db:0:0:/hsmqfs/hsmqfs1
:wq
[samqfs1mds]root@solaris:~#
```

11. 新しいデータベースと関連するテーブルを作成します。コマンド *samdb create family_set* を使用します。ここで *family_set* は、*/etc/opt/SUNWsamfs/mcf* ファイルで Oracle HSM ファイルシステムに指定されたファミリセット名です。

    デフォルトのデータベーススキーマは */opt/SUNWsamfs/etc/samdb.schema* です。別のものを指定するには、コマンドを *samdb create family_set -s schema* と入力できます。ここで、*schema* はスキーマファイルのパスと名前です。

    この例では、デフォルトのスキーマを使用してファイルシステムファミリセット *samqfs1* のデータベースを作成します。

```
[samqfs1mds]root@solaris:~# samdb create samqfs1
```

12. 前の手順で作成したデータベースロードファイルに含まれるデータを、データベースに移入します。コマンド *samdb load family_set input_file* を使用します。ここで、*family_set* は */etc/opt/SUNWsamfs/mcf* ファイルでファイルシステムに指定されたファミリセット名、*input_file* はデータベースロードファイルのパスと名前です。

    この例では、データベースロードファイル */root/hsmqfs1dataload* を使用してファイルシステムファミリセット *samqfs1* のデータベースをロードします。

```
[samqfs1mds]root@solaris:~# samdb load samqfs1 /root/hsmqfs1dataload
```

13. データベースの整合性をチェックします。コマンド *samdb check family_set* を使用します。ここで *family_set* は、*/etc/opt/SUNWsamfs/mcf* ファイルでファイルシステムに指定されたファミリセット名です。

    samdb check コマンドは、データベースのエントリをファイルシステムの現在のメタデータと比較します。これは、ロード処理時に発生した不整合があれば指摘し、可能な場合は修正します。

    この例では、データベースロードファイル */root/hsmqfs1dataload* を使用してファイルシステムファミリセット *samqfs1* のデータベースをロードします。

```
[samqfs1mds]root@solaris:~# samdb check samqfs1
```

14. 次に、データベースサポートが有効な Oracle HSM ファイルシステムのマウントを実行します。

10.4. データベースサポートが有効な Oracle HSM ファイルシステムのマウント

1. Oracle HSM メタデータサーバーホストに *root* としてログインします。

```
[samqfs1mds]root@solaris:~#
```

2. */etc/vfstab* ファイルをバックアップします。

```
[samqfs1mds]root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
```

3. テキストエディタで */etc/vfstab* ファイルを開き、データベースを作成したファイルシステムのエントリにスクロールダウンします。

    次の例では、*vi* エディタを使用します。*samqfs1* ファイルシステムのエントリにスクロールダウンします。

```
[samqfs1mds]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
```

```
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
...
samqfs1    -       /hsmqfs1  samfs   -     yes      ... ,partial=64
```

4. */etc/vfstab* ファイルの最後の列で、ファイルシステムのマウントオプションリストに *sam_db* を追加します。その後、ファイルを保存してエディタを閉じます。

   この例では、*samqfs1* ファイルシステムのサイドバンドデータベースを有効にします。

```
[samqfs1mds]root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  -------------------------
/devices   -        /devices  devfs   -     no       -
...
samqfs1    -        /hsmqfs1  samfs   -     yes      ... ,partial=64,sam_db
:wq
root@solaris:~#
```

5. Oracle HSM アーカイブファイルシステムをマウントします。

   sam_db オプションを使用してファイルシステムをマウントすると、Oracle HSM software は、サイドバンドデータベースを更新するプロセスを起動します。

   この例では、ファイルシステム */hsmqfs1* をマウントします。

```
[samqfs1mds]root@solaris:~# mount /hsmqfs1
```

6. 次に、11章「*通知とロギングの構成*」に進みます。

11

通知とロギングの構成

Oracle HSM ファイルシステムでは、SNMP (Simple Network Management Protocol) を使用した自動的なリモート通知がサポートされ、包括的で構成可能なロギング機能が提供されています。この章では、次のトピックの概要について説明します。

- SNMP (Simple Network Management Protocol) の構成
- Oracle HSM ロギングの有効化
- デバイスロギングの構成
- ログローテーションの構成
- 電子メールアラートの有効化.

11.1. SNMP (Simple Network Management Protocol) の構成

ネットワーク管理アプリケーションでは、SNMP (Simple Network Management Protocol) を使用して Oracle HSM ファイルシステムをモニターできます。ネットワーク管理ステーションに障害や構成の変更を警告するトラップが自動的に送信されるように、SNMP エージェントを構成できます。

Oracle HSM 管理情報ベース (MIB) では、SNMP トラップで提供される情報のタイプを定義します。これには、構成エラー、SCSI *tapealert* イベント、およびさまざまな種類のシステムの異常なアクティビティーが含まれます。詳細は、管理情報ベース (MIB) ファイル (*/var/snmp/mib/SUN-SAM-MIB.mib*) を参照してください。

Oracle HSM トラップイベントが発生すると、Solaris カーネルシステムのイベント通知デーモン *sysevntd* によってスクリプト */etc/opt/SUNWsamfs/scripts/sendtrap* が呼び出されます。その後、このスクリプトは、ローカルホストまたは指定した管理ステーションにトラップを送信します。このスクリプトでは、以前の標準バージョンと下位互換性のあるバージョン 2c の SNMP 標準がサポートされています。バージョン 2c では、認証資格証明 (*community strings*) および管理データが平文で交換されます。詳細は、*sendtrap* のマニュアルページを参照してください。

SNMP 通知を構成するには、次のタスクを実行します。

- **/etc/hosts** ファイルにすべての SNMP 管理ステーションが一覧表示されていることの確認
- SNMP サポートの有効化
- トラップ受信者としての管理ステーションの指定と認証の構成.

このセクションには、任意の時点で SNMP サポートの無効化を実行する必要がある場合の指示も記載されています。

11.1.1. /etc/hosts ファイルにすべての SNMP 管理ステーションが一覧表示されていることの確認

1. Oracle HSM サーバーに *root* としてログインします。

   この例では、Oracle HSM サーバーホストは *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. */etc/hosts* ファイルをテキストエディタで開きます。SNMP 管理ステーションとして使用するホストごとにエントリを含めます。

   次の例では、*vi* エディタを使用します。目的の管理ステーションの 1 つ (*management1*) が一覧表示されています。ただし、別の 1 つ (*management2*) は一覧表示されていません。

```
[samqfs1mds]root@solaris:~# vi /etc/hosts
# Internet host table
::1 localhost
127.0.0.1 localhost loghost
10.0.0.10 server1
10.0.0.20 management1
```

3. */etc/hosts* ファイルに目的の SNMP 管理ステーションホストの一部またはすべてに対応するエントリが含まれていない場合は、必要なエントリを追加して、ファイルを保存します。

   この例では、*vi* エディタを使用して、欠落している管理ステーション *management2* を追加します。

```
[samqfs1mds]root@solaris:~# vi /etc/hosts
```

```
# Internet host table
::1 localhost
127.0.0.1 localhost loghost
10.0.0.10 server1
10.0.0.20 management1
10.0.0.30 management2
```

4. */etc/hosts* ファイルに目的の SNMP 管理ステーションホストに対応するエントリがすべて含まれている場合は、エディタを閉じます。

```
[samqfs1mds]root@solaris:~# vi /etc/hosts
...
10.0.0.20 management1
10.0.0.30 management2
:wq
[samqfs1mds]root@solaris:~#
```

5. 次に、SNMP サポートの有効化を実行します。

11.1.2. SNMP サポートの有効化

デフォルトでは、SNMP 通知のサポートが有効になっているため、ある時点で SNMP サポートを無効にしていないかぎり、アクションは必要ありません。SNMP サポートを再度有効にする必要がある場合は、次の手順に従います。

1. Oracle HSM サーバーに *root* としてログインします。

   この例では、Oracle HSM サーバーホストは *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. *file /etc/opt/SUNWsamfs/defaults.conf* をテキストエディタで開きます。行 *alerts = off* を探します。

   ディレクティブ *alerts = off* は、SNMP サポートを無効にします。この例では、*vi* エディタでファイルを開き、次の行を探します。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
```

```
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
alerts = off
```

3. SNMP 通知のサポートを有効にするには、*alerts* ディレクティブの値を *on* に変更します。その後、ファイルを保存してエディタを閉じます。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
alerts = on
:wq
[samqfs1mds]root@solaris:~#
```

4. *defaults.conf* ファイルを再度読み取り、サービス自体を適宜再構成するように Oracle HSM サービスに指示します。コマンド *samd config* を使用します。

```
[samqfs1mds]root@solaris:~# 1mds]samd config
[samqfs1mds]root@solaris:~#
```

5. 次に、トラップ受信者としての管理ステーションの指定と認証の構成を実行します。

11.1.3. トラップ受信者としての管理ステーションの指定と認証の構成

1. Oracle HSM サーバーに *root* としてログインします。

   この例では、Oracle HSM サーバーホストは *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. テキストエディタで */etc/opt/SUNWsamfs/scripts/sendtrap* ファイルを開き、*TRAP_DESTINATION=* で始まる行を探します。

   sendtrap ファイルは、構成可能なシェルスクリプトです。この例では、*vi* エディタでファイルを開きます。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/scripts/sendtrap
# /etc/opt/SUNWsamfs/scripts/sendtrap#!/usr/bin/sh
# sendtrap:
#  This script gets invoked by the sysevent configuration file.
#  This is not expected to be run as a stand-alone program
...
# CONFIGURATION PARAMETERS:
TRAP_DESTINATION=`hostname`
```

3. 行 *TRAP_DESTINATION=`* で、単一引用符に囲まれたテキストを、それぞれ *hostname*:*port* 形式で 1 つ以上のトラップ受信者の空白文字区切りリストで置き換えます。ここで、*hostname* は、*/etc/hosts* に一覧表示されている管理ステーションのホスト名で、*port* は、ホストがトラップを待機しているポートです。

   デフォルトでは、トラップは *localhost* の UDP ポート *161* に送信されます。この例では、ホスト *management1* と *management2* をデフォルトの *localhost* に追加します。*localhost* と *management1* ではデフォルトポートが使用されますが、*management2* ではカスタムポート *1161* が使用されます。

```
...
# CONFIGURATION PARAMETERS:
TRAP_DESTINATION=`localhost:161 management1:161 management1:1161`
```

4. コミュニティー文字列 *COMMUNITY="public"* が設定されている行までスクロールダウンします。

   コミュニティー文字列は、SNMP バージョン 2c でエージェントおよび管理ステーションを認証する平文の共有パスワードです。デフォルト値は SNMP 標準 *public* です。

```
...
# CONFIGURATION PARAMETERS:
TRAP_DESTINATION=`localhost:161 management1:161 management1:1161`
...
COMMUNITY="public"
```

5. *COMMUNITY=""* ディレクティブを管理ステーションで使用される値に設定します。その後、ファイルを保存してエディタを閉じます。

ファイル内のその他の部分は編集しないでください。編集可能なパラメータは、*COMMUNITY=""* と *TRAP_DESTINATION=``* のみです。

デフォルトの SNMP コミュニティー文字列 *public* はセキュアでないことに注意してください。そのため、ネットワーク管理者がよりセキュアな選択を指示する場合があります。SNMP バージョン 2c では、最大 32 文字の英数字を使用できます。この例では、コミュニティー文字列を *Iv0wQh2th74bVVt8of16t1m3s8it4wa9* に設定します。

```
...
# CONFIGURATION PARAMETERS:
TRAP_DESTINATION=`localhost:161 management1:163 management1:1162`
...
COMMUNITY="Iv0wQh2th74bVVt8of16t1m3s8it4wa9"
:wq
[samqfs1mds]root@solaris:~#
```

6. 次に、Oracle HSM アプリケーションロギングの有効化を実行します。

11.1.4. SNMP サポートの無効化

デフォルトの場合、リモート通知機能は使用可能です。リモート通知を無効にする必要がある場合は、次の手順を実行します。

1. Oracle HSM サーバーに *root* としてログインします。

   この例では、Oracle HSM サーバーホストは *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. */etc/opt/SUNWsamfs/defaults.conf* ファイルをテキストエディタで開きます。行 *#alerts = on* を探します。

   この例では、*vi* エディタを使用します。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
```

```
#alerts = on
[samqfs1mds]root@solaris:~#
```

3. SNMP 通知のサポートを無効にするには、行のコメントを解除するためにハッシュ文字 (*#*) を削除して、*alerts* の値を *off* に変更します。その後、ファイルを保存してエディタを閉じます。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
alerts = off
:wq
[samqfs1mds]root@solaris:~#
```

4. *defaults.conf* ファイルを再度読み取り、サービス自体を適宜再構成するように Oracle HSM サービスに指示します。コマンド *samd config* を使用します。

```
[samqfs1mds]root@solaris:~# samd config
[samqfs1mds]root@solaris:~#
```

5. ここで停止します。SNMP サポートが無効になります。

11.2. Oracle HSM ロギングの有効化

/var/adm/sam-log ファイルには、Oracle HSM アプリケーションおよびそのコンポーネントのデーモンとプロセスのステータスおよびエラー情報が記録されます。ロギングプロセスを設定するには、次の手順を実行します。

11.2.1. Oracle HSM アプリケーションロギングの有効化

1. Oracle HSM サーバーに *root* としてログインします。

   この例では、Oracle HSM サーバーホストは *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. テキストエディタで */etc/syslog.conf* ファイルを開きます。

この例では、*vi* エディタでファイルを開きます。

```
[samqfs1mds]root@solaris:~# vi /etc/syslog.conf
# syslog configuration file ...
*.err;kern.notice;auth.notice                  /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit       /var/adm/messages
*.alert;kern.err;daemon.err                    operator
*.alert                                        root
...
```

3. */etc/syslog.conf* ファイルで、文字列 *local7.debug*、1 つ以上のタブ文字、およびパス文字列 */var/adm/sam-log* で構成される 1 行を追加します。その後、ファイルを保存してエディタを閉じます。

   この例では、次のコメントも追加します。

```
[samqfs1mds]root@solaris:~# vi /etc/syslog.conf
# syslog configuration file ...
*.err;kern.notice;auth.notice                  /dev/sysmsg
*.err;kern.debug;daemon.notice;mail.crit       /var/adm/messages
*.alert;kern.err;daemon.err                    operator
*.alert                                        root
...
# Oracle HSM logging
local7.debug    /var/adm/sam-log
:wq
[samqfs1mds]root@solaris:~#
```

4. ログファイル */var/adm/sam-log* を作成します。コマンド *touch /var/adm/sam-log* を使用します。

```
[samqfs1mds]root@solaris:~# touch /var/adm/sam-log
[samqfs1mds]root@solaris:~#
```

5. Solaris *syslogd* デーモンに、その構成ファイルを再度読み取り、Oracle HSM ロギングを開始するように指示します。コマンド *pkill -HUP syslogd* を使用します。

HUP シグナルを受信するたびに、*syslogd* ロギングサービスによって */etc/syslog.conf* 構成ファイルが再度読み取られ、開いているログファイルがすべて閉じられてから、*syslog.conf* に一覧表示されているログファイルが開かれます。このコマンドを実行すると、Oracle HSM ロギングが有効になります。

```
[samqfs1mds]root@solaris:~# pkill -HUP syslogd
[samqfs1mds]root@solaris:~#
```

6. デバイスロギングの構成に進みます。

11.3. デバイスロギングの構成

デバイスロギング機能では、個々のハードウェアデバイスに固有のエラー情報が提供されます (ソフトメディアのエラーは収集されません)。各デバイスには、対応する装置番号を含む名前が付けられ、*/var/opt/SUNWsamfs/devlog/* ディレクトリに格納される独自のログファイルが存在します。

デバイスログは、急速に増大する可能性があります。そのため、デフォルトでは、限られたイベントデータセット (*err*、*retry*、*syserr*、および *date*) のログが記録されます。あとで問題が発生した場合は、*samset* コマンドを使用すると、デバイス単位で追加のイベントをログに記録できます (詳細は、*samset* のマニュアルページの *devlog* セクションを参照)。

11.3.1. defaults.conf ファイルでのデバイスログの有効化

基本的なデバイスロギングを有効にするには、次の手順を実行します。

1. Oracle HSM サーバーに *root* としてログインします。

   この例では、Oracle HSM サーバーホストは *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. テキストエディタで */etc/opt/SUNWsamfs/defaults.conf* を開きます。

   この例では、*vi* エディタでファイルを開きます。

```
[samqfs1mds]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
```

...

3. *defaults.conf* ファイルで、必要なデバイスロギングのデフォルトレベルを定義する 1 行を追加します。ディレクティブ *devlog equipment-number loggable-events* を入力します。ここでは:

   - *equipment-number* は、*/etc/opt/SUNWsamfs/mcf* ファイルで定義されたすべての装置を表すキーワード *all*、または *mcf* で定義された装置の特定部分を識別する装置番号です。
   - *loggable-events* は、デフォルト値を空白文字で区切ったリスト (*err retry syserr date*) です。

     イベントタイプの包括的なリストについては、*samset* のマニュアルページの *devlog* セクションを参照してください。ただし、ログのサイズを最小限に抑えるには、デフォルトの選択を使用します。診断のために、*samset* コマンドを使用して、必要に応じて追加のイベントを選択的に有効にできます。

   この例では、デフォルトのロギングレベルを使用して、*すべての*デバイスのデバイスロギングを有効にします。

```
[samfs-mds1]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
devlog all err retry syserr date
```

4. *defaults.conf* ファイルを保存して、エディタを閉じます。

```
[samfs-mds1]root@solaris:~# vi /etc/opt/SUNWsamfs/defaults.conf
# These are the defaults.  To change the default behavior, uncomment the
# appropriate line (remove the '#' character from the beginning of the line)
# and change the value.
...
devlog all err retry syserr date
:wq
[samfs-mds1]root@solaris:~#
```

5. *defaults.conf* ファイルを再度読み取り、サービス自体を適宜再構成するように Oracle HSM サービスに指示します。コマンド *samd config* を使用します。

```
[samfs-mds1]root@solaris:~# samd config
[samfs-mds1]root@solaris:~#
```

6. 次に、Oracle HSM ログファイルの自動ローテーションの設定を実行します。

11.4. ログローテーションの構成

ログファイルが急速に増大すると、大量の領域が消費され、ログの使用が困難になる可能性があります。そのため、Oracle HSM ログの自動ログローテーションを構成するようにしてください。この目的のために、ソフトウェアには Solaris *crontab* ファイルから実行できるスクリプト *log_rotate.sh* が含まれています。

ローテーションする目的のログごとに、2 つの *crontab* エントリを作成します。1 つ目のエントリは、目的の時間に *log_rotate.sh* スクリプトを実行します。ターゲットのログファイルが指定された最小サイズ (デフォルトは *100000* バイト) に達すると、スクリプトによって名前が変更され、もっとも古い既存のコピーが削除されます (同時に 7 つが常に保持されています)。2 つ目の *crontab* エントリは、Solaris ロギングデーモン *syslogd* に、新しいログファイルでロギングを再開するように指示します。

11.4.1. Oracle HSM ログファイルの自動ローテーションの設定

次のログをローテーションすることを検討します。

- */etc/syslog.conf* ファイルで指定された場所に配置されている Oracle HSM ログファイル *sam-log*。
- */var/opt/SUNWsamfs/devlog/* ディレクトリに配置されているデバイスログファイル。
- */etc/opt/SUNWsamfs/stager.cmd* ファイルで指定されているステージャーログファイル。
- */etc/opt/SUNWsamfs/releaser.cmd* ファイルで指定されているリリーサログファイル。
- */etc/opt/SUNWsamfs/recycler.cmd* ファイルで指定されているリサイクラログファイル。

アーカイバログファイルはローテーションしないでください。ログ情報は分析とファイルシステムの回復に有益です。アーカイバログの適切な処理については、「ファイルシステムの保護の構成」を参照してください。

ローテーション対象のログを決定したら、ログごとに次の手順を実行します。

1. Oracle HSM サーバーに *root* としてログインします。

   この例では、Oracle HSM サーバーホストは *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. スクリプトファイル *log_rotate.sh* を */opt/SUNWsamfs/examples/* (アンインストールされた場所) から */etc/opt/SUNWsamfs/scripts/* にコピーします。

   次のコマンドは 1 行で入力します。改行はバックスラッシュによってエスケープされます。

```
[samfs-mds1]root@solaris:~# cp /opt/SUNWsamfs/examples/log_rotate.sh /  /etc/opt/SUNWsamfs/scripts/
```

3. 編集のために *root* ユーザーの *crontab* ファイルを開きます。コマンド *crontab -e* を使用します。

   crontab コマンドは、*root* ユーザーの *crontab* ファイルの編集可能なコピーを、*EDITOR* 環境変数で指定されたテキストエディタで開きます (詳細は、Solaris *crontab* のマニュアルページを参照してください)。この例では、*vi* エディタを使用します。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
```

4. 新しい行で、*minutes hour * * day-of-the-week* と入力して、ログファイルをローテーションする日時を指定します。ここでは:
   - *minutes* は、ジョブを開始する分を指定する、[*0-59*] の範囲内の整数です。
   - *hour* は、ジョブを開始する時を指定する、[*0-23*] の範囲内の整数です。
   - *** (アスタリスク) は未使用の値を指定します。

     毎日実行するタスクの場合、日 [*1-31*] および月 [*1-12*] の値は使用されません。
   - *day-of-the-week* は、日曜日 (*0*) から始まる、[*0-6*] の範囲の整数です。

- 空白は、時間の指定のフィールドを区切ります。

この例では、毎週日曜日の午前 3 時 10 分にログローテーションが開始されるようにスケジュールします。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0
```

5. 引き続き同じ行で、Oracle HSM ログをローテーションするシェルスクリプトのパスと名前 */etc/opt/SUNWsamfs/scripts/log_rotate.sh* を入力し、そのあとに空白文字を追加します。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0 /etc/opt/SUNWsamfs/scripts/log_rotate.sh
```

6. 引き続き同じ行で、ローテーションする必要のあるログの名前およびローテーションするファイルの最小サイズを入力します。テキスト *samfslog* [*minimum-size*] を入力します。ここで *samfslog* は、Oracle HSM ログファイルへのパス、[*minimum-size*] は、スクリプトでローテーションされる最小ファイルサイズ (バイト単位) を指定するオプションの整数 (デフォルトは *100000*) です。

   この例では、*/var/adm/sam-log* をローテーションする必要があります。デフォルトの最小サイズを受け入れます。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0 /etc/opt/SUNWsamfs/scripts/log_rotate.sh /var/adm/sam-log
```

7. 改行します。*log_rotate.sh* スクリプトの 10 分後に開始される *crontab* エントリを作成します。このエントリは、古いログファイルを閉じて新しいファイルにロギングを再開するように Solaris *syslogd* デーモンに指示します。行 *minutes hour * * day-of-the-week /bin/kill -HUP `/bin/cat /etc/syslog.pid`* を入力します。ここで *minutes hour * * day-of-the-week* には、前のステップで指定した時間の 10 分後の時間を指定します。

   この例では、エントリは毎週日曜日の午前 3 時 20 分に Oracle HSM ロギングを再開します。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0 /etc/opt/SUNWsamfs/scripts/log_rotate.sh /var/adm/sam-log
20 3 * * 0  /bin/kill -HUP `/bin/cat /etc/syslog.pid`
```

8. ファイルを保存して、エディタを閉じます。

```
[samfs-mds1]root@solaris:~# crontab -e
#ident "%Z%%M% %I% %E% SMI"
# Copyright 2007 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
```

```
# The root crontab should be used to perform accounting data collection.
10 3 * * * /usr/sbin/logadm
...
30 0,9,12,18,21 * * * /usr/lib/update-manager/update-refresh.sh
10 3 * * 0 /etc/opt/SUNWsamfs/scripts/log_rotate.sh /var/adm/sam-log
20 3 * * 0  /bin/kill -HUP `/bin/cat /etc/syslog.pid`
:wq
[samfs-mds1]root@solaris:~#
```

9. 必要なすべてのログのログローテーションが構成されるまで、この手順を繰り返します。
10. 次に、電子メールアラートの有効化を参照してください。

11.5. 電子メールアラートの有効化

Oracle HSM Manager のグラフィカルユーザーインタフェースを使用すると、電子メールアラートが最適に設定されます。詳細は、オンラインヘルプを参照してください。

コマンド行インタフェースから電子メールアラートを構成する必要がある場合は、*defaults.conf*、*archiver.sh*、*dev_down.sh*、*load_notify.sh*、*recycler.sh*、*archiver.cmd*、*recycler.cmd*、および *notify.cmd* のマニュアルページを参照してください。

この時点で、Oracle HSM システムが構成されています。ただし、使用を開始する前に、作業内容を保護します。手順については、13章「*Oracle HSM 構成のバックアップ*」を参照してください。

12

特殊なニーズのための入出力特性の調整

これまでの章で説明した基本的なファイルシステム構成手順は、ほとんどの状況で、最適かつバランスの取れたパフォーマンスを提供します。したがって、アプリケーションの動作方法についてまったく確信が持てない場合は、通常、このセクションに含まれる設定をデフォルト値のままにしておくことをお勧めします。ただし、アプリケーションの入出力要求が非常に一貫している、あるいは非常に大きい場合、ファイルシステムによる物理入出力の処理方法を調整または変更すると、全体のパフォーマンスが向上する可能性があります。

物理入出力がもっとも効率的なのは、読み取りや書き込みのすべてあるいはほとんどが、ディスクセクターの 512 バイト境界で開始および終了するような場合です。ディスク入出力はセクターサイズのチャンクでしか発生できません。したがって、入出力要求がセクター境界をまたがる場合、システムはアプリケーションデータを同じセクター内の無関係なデータから分離する追加の操作を実行する必要があります。プロセスの後半の処理が失敗しないようにする必要があります。最悪の場合、セクターをまたがる書き込みの際に、ファイルシステムはセクターを読み取り、メモリー内のセクターデータを変更したあと、セクターをディスクに書き戻す必要があります。そのような読み取り/変更/書き込み操作は、機械的なアクティビティーの増加だけを考慮しても、パフォーマンス上きわめてコストが高くなります。

しかし、ほとんどのアプリケーションは、セクター境界に適切に整列されていないさまざまなサイズのデータの読み取りや書き込みを行う必要があります。このため、Oracle HSM では多くのファイルシステムと同様に、デフォルトでページ*入出力*が使用されます。ファイルシステムがアプリケーションからの即時入出力要求を処理する際は、Solaris ページメモリー内のデータキャッシュに対して読み取りや書き込みを行います。ファイルシステムが非同期でキャッシュを更新する際は、より効率的なサイズで、適切に整列された物理読み取り/書き込みを行います。ディスクからデータを読み取るたびに、次の読み取りを予測して同じ操作内で対応するデータをキャッシュにロードしておくことにより、物理入出力を最大限に活用できます。したがって、ほとんどの入出力要求は、追加の物理ディスクアクティビティーなしに、仮想メモリーページ内にキャッシュされたデータを使用して対応されます。ページ入出力ではメモリーが使用され、システム CPU にもある程度の負荷がかかりますが、そのコストはほとんどの場合、物理入出力の効率アップによって十分に相殺されます。

ただし、ある場合には、その利点によってもページ入出力に関連する追加のオーバーヘッドが相殺されないことがあります。常に適切に整列された入出力を常に実行するアプリケーションや、そのように調整可能なアプリケーションでは、ページキャッシュの利点はありません。きわめて大きな入出力を実行するアプリケーションでも、整列されていないセクターは最初と最後だけであり、入出力が大きすぎるとキャッシュに収まらないことがあるため、ページキャッシュの利点がほとんどない場合があります。最後に、テレメトリデータ、監視ビデオ、またはその他の種類のリアルタイム情報をストリーミングするアプリケーションでは、書き込みが即時に非揮発性ストレージに収容されない場合、回復不可能なデータが失われる危険性があります。このような場合は、*直接入出力*を使用したほうが良い可能性があります。直接入出力が指定された場合、ファイルシステムはアプリケーションのメモリーとディスクデバイスとの間でデータを直接転送し、ページキャッシュをバイパスします。

Oracle HSM では、入出力キャッシュ動作の選択や調整についてユーザーにさまざまな選択肢を用意しています。アプリケーションの入出力特性を理解し、「予測したファイルシステム入出力に合わせた Solaris システムおよびドライバパラメータの調整」で説明したタスクを完了したら、次のように方法を選択してください。

- アプリケーションが実行する入出力要求が常に小さい、サイズが一定していない、あるいは整列されていない場合は、Oracle HSM のデフォルト設定を受け入れます。このセクションに含まれる変更を一切行わないでください。
- アプリケーションが実行する入出力要求のサイズが一定していないが、平均より大きく、整列されていない場合は、大規模データ転送のためのページ入出力の最適化を実行します。
- アプリケーションが実行する入出力要求に、適切に整列されている、あるいはサイズが大きいものと、整列されていないサイズの小さいものが混在している場合は、ページ入出力と直接入出力の切り替えの有効化を実行します。
- アプリケーションが実行する入出力要求が*一貫して*適切に整列されている、または非常に大きい場合は、直接入出力を排他的に使用するためのファイルシステムの構成を実行します。
- アプリケーションが共有ファイルシステムクライアント上で実行され、常に多数のファイルを開いている場合は、ディレクトリ名参照キャッシュのサイズの増加を実行します。

12.1. 大規模データ転送のためのページ入出力の最適化

ページ入出力は、アプリケーションやハードウェアの特性と一致するように調整できます。キャッシュへの読み取りやキャッシュからの書き込みのサイズは、アプリケーションが転送する平均データ量、または物理ストレージが転送可能な最大データ量のいずれかの大きいほうを転送できる大きさにする必要があります。このいずれかに合わせてページキャッシュ動作

を調整できなかった場合、キャッシュ使用率が低下し、アプリケーションの入出力要求に対して必要な物理入出力が増加し、システム全体のパフォーマンスが低下します。

たとえば、単一のディスクボリューム上に実装された *md* データデバイスと、3+1 RAID 5 ボリュームグループ上に実装された *md* デバイスを比較することを検討します。アプリケーションからの各書き込み要求を処理する際に、マルチディスクデバイスで可能な追加の帯域幅を無視し、単一の 64K バイトディスク割り当て単位 (DAU)をキャッシュから後者のデバイスに書き込む場合、RAID デバイスは、その入出力を 3 つの小さな、効率の低い 21K/22K バイトのフラグメントに分割してから、RAID グループ内の 3 つのデータディスクにデータを書き出す必要があります。したがって、アプリケーションからの 64K バイトの入出力要求の処理に対して、この構成では、ページキャッシュによって複数の要求を結合して 3-DAU (192K バイト) の単一の入出力にする場合より必要な作業が著しく増加します。アプリケーションがデバイス帯域幅の偶数倍 (192、384、または 576K バイト) で入出力要求を実行できれば、またはそのように調整可能であれば、物理入出力ごとにキャッシュまたは転送できるデータ量を増やせるため、オーバーヘッドがさらに減少し、結果的にパフォーマンスが向上します。

そのためには、アプリケーションの入出力要件を明確にし、ハードウェアの入出力特性を理解しておきます。その後、次の手順を実行します。

1. ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~#
```

3. テキストエディタで */etc/vfstab* ファイルを開き、調整が必要なファイルシステムの行を検索します。

   この例では、ファイルシステムの名前は *qfsma* です。

```
root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  --------------------
/devices   -        /devices  devfs   -     no       -
```

```
...
qfsma      -        /qfsma    samfs   -     yes      ...
```

4. ファイルシステムの「*Mount Options*」フィールドに、*writebehind=n* マウントオプションを追加します。ここで、*n* は *8*K バイトの倍数です。マウントオプションを区切るにはコンマ (空白なし) を使用します。ファイルを保存して、エディタを閉じます。

   writebehind オプションは、ページキャッシュがディスクにフラッシュされるまでに、特定のファイルのどれだけの量がキャッシュの待ち行列に入れられるかを決定します。このパラメータの値の設定を大きくするとパフォーマンスが改善されますが、これは、キューが大きければ、多数の小さなアプリケーション書き込みが、少数の大きい、より効率的な物理入出力にまとめられるためです。このパラメータに設定する値が小さいほど、変更がすぐに非揮発性ストレージに書き込まれるため、データ保護が強化されます。

   デフォルト値は *512*K バイト (64K バイトの DAU 8 個分) ですが、これは一般に、大きなブロックの順次入出力に適しています。しかしこの例の場合、ファミリセットには、ストライプ化ファイル割り当ての *md* ディスクデバイスが 2 つ含まれています。ストライプ幅は 64K バイト (1 DAU) であり、2 つの *md* デバイスに 128K バイトが書き込まれることになります。*md* デバイスは 3+1 RAID 5 グループです。したがって、3 つのデータスピンドルのそれぞれに少なくとも 128K バイトを書き込む必要があるため、合計で *768*K バイト (96 グループがそれぞれ 8K バイト) 以上の書き込みが必要になります。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  --------------------
/devices   -        /devices  devfs   -     no       -
...
qfsma      -        /qfsma    samfs   -     yes      ...,writebehind=768
:wq
root@solaris:~#
```

5. ファイルシステムの入出力パフォーマンスをテストし、必要に応じて *writebehind* 設定を調整します。
6. テキストエディタで */etc/vfstab* ファイルを再度開きます。ファイルシステムの「*Mount Options*」フィールドに、*readahead=n* マウントオプションを追加します。ここで、*n* は 8K

バイトの倍数です。マウントオプションを区切るにはコンマ (空白なし) を使用します。ファイルを保存して、エディタを閉じます。

readahead オプションは、単一の物理読み取り時にキャッシュ内に読み取られるデータの量を決定します。アプリケーションが順次読み取りを実行していると考えられる場合、ファイルシステムは、物理読み取りのたびに、次に必要になるデータブロックをキャッシュします。このとき、一連のアプリケーション読み取り要求はキャッシュメモリーから処理でき、複数のアプリケーション読み取り要求が単一の物理入出力要求にまとめられます。

デフォルト値は *1024*K バイト (64K バイトの DAU 16 個分) ですが、これは一般に、大きなブロックの順次入出力に適しています。データベースやそれに類似したアプリケーションが独自に readahead を実行する場合、競合を避けるために、Oracle HSM *readahead* を *0* に設定します。それ以外の場合、*readahead* は一般に、単一の物理入出力で転送可能な最大のデータ量をキャッシュできるよう設定する必要があります。*readahead* の設定が、アプリケーションから通常要求されるデータ量や各デバイスから供給可能なデータ量よりも小さい場合、アプリケーションの入出力要求に対応するために、必要以上に多くの物理入出力が要求されます。ただし、*readahead* の設定値が高すぎると、メモリーの消費量が増大し、システム全体のパフォーマンスが低下する可能性があります。この例では、*readahead* を *736*K バイト (64K バイトの DAU 36 個分) に設定します。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  --------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
qfsma      -        /qfsma    samfs   -     yes      ...,readahead=736
:wq
root@solaris:~#
```

7. ファイルシステムの入出力パフォーマンスをテストし、必要に応じて *readahead* 設定を調整します。

   readahead パラメータのサイズを増やすと、ある時点までは大容量ファイル転送のパフォーマンスが向上します。そのため、*readahead* サイズをリセットしたあと、システムのパフォーマンスをテストします。次に転送速度がそれ以上上がらないと思うところまで *readahead* サイズを上向きに調整します。

12.2. ページ入出力と直接入出力の切り替えの有効化

ページ入出力と直接入出力を切り替えたほうがアプリケーションの入出力動作に適合する場合、これを切り替えられるように Oracle HSM ファイルシステムを構成できます。直接入出力の利点を引き出す読み取り/書き込みの特性として、セクター整列と最小サイズを指定し、切り替えをトリガーする適格の読み取り/書き込み個数を設定します。次のように進めます。

1. ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~#
```

3. テキストエディタで */etc/vfstab* ファイルを開き、構成するファイルシステムの行を検索します。

   この例では、ファイルシステムの名前は *qfsma* です。

```
root@solaris:~# vi /etc/vfstab
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ------------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   stripe=1
```

4. 512 バイトセクター境界に適切に整列された読み取り要求に対して直接入出力を開始するしきい値サイズを設定するには、ファイルシステムの「*Mount Options*」フィールドに *dio_rd_form_min=n* マウントオプションを追加します。ここで、*n* は K バイト数です。マウントオプションを区切るにはコンマ (空白なし) を使用します。

   デフォルトは *dio_rd_form_min=256*K バイトです。この例では、アプリケーションが 512K バイト以上の読み取りを要求するまで、適切に整列された読み取りが継続的に生

成されないことがわかっています。したがって、適切に整列された直接読み取りのしきい値サイズを、*512* に変更します。

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   stripe=1,dio_rd_form_min=512
```

5. 512 バイトセクター境界に適切に整列された書き込み要求に対して直接入出力を開始するしきい値サイズを設定するには、ファイルシステムの「*Mount Options*」フィールドに *dio_wr_form_min=n* マウントオプションを追加します。ここで、*n* は K バイト数です。マウントオプションを区切るにはコンマ (空白なし) を使用します。

   デフォルトは *dio_wr_form_min=256*K バイトです。この例では、アプリケーションが 1M バイト以上の書き込みを要求するまで、適切に整列された書き込みが継続的に生成されないことがわかっています。したがって、適切に整列された直接書き込みのしきい値サイズを、*1024*K バイトに変更します。

```
#File     Device                      Moun
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_form_min=1024
```

6. 512 バイトセクター境界に適切に整列されていない読み取り要求に対して直接入出力を開始するしきい値サイズを設定するには、ファイルシステムの「*Mount Options*」フィールドに *dio_rd_ill_min=n* マウントオプションを追加します。ここで、*n* は K バイト数です。マウントオプションを区切るにはコンマ (空白なし) を使用します。

   デフォルトは *dio_rd_ill_min=0*K バイトであり、整列されていない読み取りでは直接入出力は使用されません。この例では、アプリケーションが一般に小さなデータ単位に

対しては、整列されていない読み取り要求を行うことがわかっています。このデータの大部分はあとで再度読み取られます。したがって、これらの読み取りにはページキャッシュが適していると考えられます。直接入出力に切り替えると、不要な物理入出力が増加し、パフォーマンスが低下します。したがって、ここではデフォルトを受け入れ、*vfstab* ファイルに変更を加えません。

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_form_min=1024
```

7. 512 バイトセクター境界に適切に整列されていない書き込み要求に対して直接入出力を開始するしきい値サイズを設定するには、ファイルシステムの「*Mount Options*」フィールドに *dio_wr_ill_min=n* マウントオプションを追加します。ここで、*n* は K バイト数です。マウントオプションを区切るにはコンマ (空白なし) を使用します。

   デフォルトは *dio_wr_ill_min=0*K バイトであり、整列されていない書き込みでは直接入出力は使用されません。整列されていない書き込みでは、システムがセクターの読み取り、変更、書き込みを行う必要があるため、特にパフォーマンス上で高コストになることがあります。ただしこの例では、アプリケーションがセクター境界に位置しない単一の大規模な書き込み要求を行う場合があることがわかっています。読み取り/書き込み/変更操作は、連続するセクターで構成される大規模ブロックの先頭と末尾に限定されるため、直接入出力のメリットがページ入出力のメリットよりも大きくなります。したがって、ここでは *dio_wr_ill_min=2048*K バイトに設定します。

   この例では、整列されていないデータの書き込みで直接入出力を使用するしきい値のデフォルトを、*2048*K バイトに変更します。

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
```

```
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_ill_min=2048
```

8. 読み取りで直接入出力を有効にするには、「*Mount Options*」フィールドに *dio_rd_consec=n* マウントオプションを追加します。ここで *n* は、直接入出力への切り替えをトリガーするため、前述の指定のサイズと整列の要件を満たす必要のある連続する入出力転送の回数を示します。直接入出力の利点を引き出すアプリケーション操作のための値を選択します。マウントオプションを区切るにはコンマ (空白なし) を使用します。

   デフォルトは *dio_rd_consec=0* であり、入出力の切り替えは無効になっています。この例では、アプリケーションがいったん *dio_rd_form_min* に指定された最小サイズ (512K バイト) 以上の適切に整列された 3 つの連続する読み取りを要求すると、直接入出力の実行が有益になるまでその動作を続行することがわかっています。*dio_rd_form_min* に指定された最小サイズはデフォルト値の *0* であるため、直接入出力を有効にしても、整列されていない読み取り要求には影響しません。したがって、*dio_rd_consec=3* と設定します。

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_rd_consec=3
```

9. 書き込みで直接入出力を有効にするには、「*Mount Options*」フィールドに *dio_wr_consec=n* マウントオプションを追加します。ここで *n* は、直接入出力への切り替えをトリガーするため、前述の指定のサイズと整列の要件を満たす必要のある連続する入出力転送の回数を示します。直接入出力の利点を引き出すアプリケーション操作のための値を選択します。マウントオプションを区切るにはコンマ (空白なし) を使用します。

   デフォルトは *dio_wr_consec=0* であり、入出力の切り替えが無効です。この例では、アプリケーションがいったん *dio_wr_form_min* に指定された最小サイズ (1024K バイト) 以上の適切に整列された 2 つの連続する書き込みを要求すると、直接入出力の実行が有益になるまでその動作を続行することがわかっています。また、整列されていない連続する 2 つの書き込みが *dio_wr_form_min* (2048K バイト) を超える場合は、十分サイズ

が大きいため、整列されていなくても比較的問題にならないこともわかっています。したがって、*dio_wr_consec=2* と設定します。

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- ------ ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_consec=2
```

10. *vfstab* ファイルを保存して、エディタを閉じます。

```
#File     Device                      Mount
#Device   to     Mount    System fsck at    Mount
#to Mount fsck   Point    Type   Pass Boot  Options
#-------- ------ -------- -----  ---- ----- ----------------------------
/devices  -      /devices devfs  -    no    -
/proc     -      /proc    proc   -    no    -
...
qfsma     -      /qfsma   samfs  -    yes   ...,dio_wr_consec=2
:wq
root@solaris:~#
```

11. 変更されたファイルシステムをマウントします。

```
root@solaris:~# mount /qfsms
root@solaris:~#
```

12.3. 直接入出力を排他的に使用するためのファイルシステムの構成

アプリケーションの入出力特性から直接入出力の排他的な使用が必要とされる場合、*forcedirectio* マウントオプションを使用してファイルシステム全体をマウントできます(個々のファイルやディレクトリに直接入出力を指定する方法については、Oracle HSM *setfa* のマニュアルページを参照)。

直接入出力を排他的に使用するようにファイルシステムをマウントするには、次の手順を実行します。

1. ファイルシステムホストに *root* としてログインします。

```
root@solaris:~#
```

2. オペレーティングシステムの */etc/vfstab* ファイルをバックアップします。

```
root@solaris:~# cp /etc/vfstab /etc/vfstab.backup
root@solaris:~#
```

3. テキストエディタで */etc/vfstab* ファイルを開き、直接入出力を使用するファイルシステムの行を検索します。

   この例では、ファイルシステムの名前は *qfsma* です。

```
root@solaris:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  --------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
qfsma      -        /qfsma    samfs   -     yes      stripe=1
```

4. ファイルシステムの「*Mount Options*」フィールドに *forcedirectio* マウントオプションを追加します。マウントオプションを区切るにはコンマ (空白なし) を使用します。ファイルを保存して、エディタを閉じます。

```
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  --------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
```

```
qfsma      -        /qfsma    samfs   -     yes      stripe=1,forcedirectio
:wq
root@solaris:~#
```

5. 変更されたファイルシステムをマウントします。

```
root@solaris:~# mount /qfsms
root@solaris:~#
```

12.4. ディレクトリ名参照キャッシュのサイズの増加

共有ファイルシステムのクライアントが同時に多数のファイルを開く場合、メタデータサーバー上の Oracle Solaris ディレクトリ名参照キャッシュ (DNLC) のデフォルトサイズが不十分であることが判明することがあります。メタデータサーバーはすべてのクライアントに代わってファイル名を検索するため、このような条件下ではファイルシステムのパフォーマンスが低下することがあります。

この種の作業負荷が予想される場合は、ディレクトリ名参照キャッシュサイズパラメータ *ncsize* の値をデフォルトサイズの 2 倍か 3 倍に変更します。手順については、*Oracle Solaris Information Library* から入手可能な『*Oracle Solaris チューニング可能パラメータリファレンスマニュアル*』を参照してください (はじめにの「入手可能ドキュメント」セクションを参照)。

13

Oracle HSM 構成のバックアップ

Oracle Hierarchical Storage Manager and StorageTek QFS Software の構成が完了したら、構成ファイルおよび関連情報をバックアップして、投資を保護してください。次のタスクを実行します。

- Oracle HSM 構成のバックアップ場所の作成
- **`samexplorer`** の実行およびレポートの安全な格納
- Oracle HSM 構成の手動バックアップ.

13.1. Oracle HSM 構成のバックアップ場所の作成

次のように進めます。

1. ファイルシステムホストに *`root`* としてログインします。

```
root@solaris:~#
```

2. Oracle HSM 構成のバックアップコピーのストレージの場所を選択します。ファイルシステムのホストにマウントできる独立したファイルシステムを選択します。
3. 選択したファイルシステムが、どの物理デバイスもアーカイブファイルシステムと共有しないようにしてください。

   保護対象のファイルシステムに回復ポイントファイルを格納しないでください。アーカイブファイルシステムもホストしている物理デバイス上にある論理デバイス (パーティションや LUN など) に回復ポイントファイルを格納しないでください。
4. 選択したファイルシステムで、構成情報を保持するディレクトリを作成します。コマンド *`mkdir mount-point/path`* を使用します。ここで、*`mount-point`* は、選択した独立ファイルシステム用のマウントポイント、*`path`* は、選択したディレクトリのパスと名前です。

   この例では、独立ファイルシステム */zfs1* にディレクトリ */zfs1/sam_config* を作成しました。

```
root@solaris:~# mkdir /zfs1/sam_config
```

5. 次に、**samexplorer** の実行およびレポートの安全な格納を実行します。

13.2. samexplorer の実行およびレポートの安全な格納

samexplorer は、Oracle HSM ソフトウェアとファイルシステムに関する包括的な構成とステータス情報を収集してレポートする診断ツールです。Oracle のサポートサービス担当者は、トラブルシューティング時にこの出力を使用します。そのため、Oracle HSM ソフトウェアとファイルシステムを構成または再構成するたびに *samexplorer* のベースラインレポートを作成することをお勧めします。

1. ファイルシステムのメタデータサーバーホストに *root* としてログインします。

   この例では、ホスト名は *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. バックアップ構成情報が保持されているディレクトリに、*samexplorer* レポート用のサブディレクトリを作成します。コマンド *mkdir mount-point/path* を使用します。ここで、*mount-point* は、選択した独立ファイルシステム用のマウントポイント、*path* は、選択したディレクトリのパスと名前です。

   この例では、ディレクトリ */zfs1/sam_config/explorer* を作成します。

```
[samqfs1mds]root@solaris:~# mkdir /zfs1/sam_config/explorer
[samqfs1mds]root@solaris:~#
```

3. 選択したディレクトリに、*samexplorer* レポートを作成します。コマンド *samexplorer path/hostname.YYYYMMDD.hhmmz.tar.gz* を使用します。ここで、*path* は選択したディレクトリへのパス、*hostname* は Oracle HSM ファイルシステムホストの名前、*YYYYMMDD.hhmmz* は日付とタイムスタンプです。

   デフォルトのファイル名は */tmp/SAMreport.hostname.YYYYMMDD.hhmmz.tar.gz* です。

   この例では、ディレクトリ */zfs1/sam_config/explorer/* にファイル *samhost1.20140130.1659MST.tar.gz* を作成します (次のコマンドは 1 行で入力します。改行はバックスラッシュでエスケープされます)。

```
[samqfs1mds]root@solaris:~# samexplorer /
```

```
/zfs1/sam_config/explorer/samhost1.20140130.1659MST.tar.gz

     Report name:     /zfs1/sam_config/explorer/samhost1.20140130.1659MST.tar.gz
     Lines per file:  1000
     Output format:   tar.gz (default) Use -u for unarchived/uncompressed.

     Please wait.....................................
     Please wait.....................................
     Please wait.....................................

     The following files should now be ftp'ed to your support provider
     as ftp type binary.

     /zfs1/sam_config/explorer/samhost1.20140130.1659MST.tar.gz

[samqfs1mds]root@solaris:~#
```

4. ファイルシステムを大幅に再構成する際は必ず、この手順を繰り返してください。
5. 次に、Oracle HSM 構成の手動バックアップを実行します。

13.3. Oracle HSM 構成の手動バックアップ

samexplorer ユーティリティーは、完全な冗長性のために大部分の Oracle HSM 構成情報を収集しますが、主な構成作業のあとには次の手順を実行してください。

1. ファイルシステムホストに *root* としてログインします。

   この例では、ホスト名は *samqfs1mds* です。

```
[samqfs1mds]root@solaris:~#
```

2. バックアップ構成情報が保持されているディレクトリに、Oracle HSM 構成の手動バックアップコピー用のサブディレクトリを作成します。コマンド *mkdir mount-point/path* を使用します。ここで、*mount-point* は、選択した独立ファイルシステム用のマウントポイント、*path* は、選択したディレクトリのパスと名前です。

   次の例では、アーカイブファイルシステム */hsmqfs1* 用の回復ポイントを構成します。そのため、ディレクトリ */zfs1/sam_config/samconfig* を作成しました。

```
[samqfs1mds]root@solaris:~# mkdir /zfs1/sam_config/samconfig
[samqfs1mds]root@solaris:~#
```

3. 選択したディレクトリに、現在の Oracle HSM 構成用のサブディレクトリを作成します。コマンド *mkdir mount-point/path/subdirectory* を使用します。ここで *mount-point* は選択した独立ファイルシステム用のマウントポイント、*path/subdirectory* は選択したディレクトリのパスと名前です。

   この例では、日付を使用してサブディレクトリの名前を付けます。

```
samqfs1mdsroot@solaris:~# mkdir /zfs1/sam_config/samconfig/20140127
[samqfs1mds]root@solaris:~#
```

4. 構成ファイルを別のファイルシステムにコピーします。

```
/etc/opt/SUNWsamfs/
     mcf
     archiver.cmd
     defaults.conf
     diskvols.conf
     hosts.family-set-name
     hosts.family-set-name.local
     preview.cmd
     recycler.cmd
     releaser.cmd
     rft.cmd
     samfs.cmd
     stager.cmd
     inquiry.conf
     samremote                  # SAM-Remote server configuration file
     family-set-name           # SAM-Remote client configuration file
     network-attached-library   # Parameters file
     scripts/*                  # Back up all locally modified files
```

5. すべてのライブラリカタログデータ (ヒストリアンで保持されているデータを含む) をバックアップします。カタログごとに、コマンド */opt/SUNWsamfs/sbin/dump_cat -V*

catalog-file を使用します。ここで、*catalog-file* はカタログファイルのパスと名前です。出力を新しい場所にある *dump-file* にリダイレクトします。

この例では、*library1* のカタログデータを、NFS でマウントされた個別のファイルシステム *zfs1* 上のディレクトリにある *library1cat.dump* ファイルにダンプします (次のコマンドは 1 行で入力します。改行はバックスラッシュでエスケープされます)。

```
samqfs1mdsroot@solaris:~# dump_cat -V /var/opt/SUNWsamfs/catalog/library1cat /
> /zfs1/sam_config/20140513/catalogs/library1cat.dump
```

6. Oracle HSM のインストールおよび構成中に変更したシステム構成ファイルをコピーします。これには、次が含まれる可能性があります。

```
/etc/
      syslog.conf
      system
      vfstab
/kernel/drv/
      sgen.conf
      samst.conf
      samrd.conf
      sd.conf
      ssd.conf
      st.conf
/usr/kernel/drv/dst.conf
```

7. Oracle HSM 構成の一部として作成したカスタムシェルスクリプトおよび *crontab* エントリを選択したサブディレクトリにコピーします。

   たとえば、回復ポイントの作成とログのローテーションを管理するために *crontab* エントリを作成した場合は、ここでコピーを保存します。

8. 現在インストールされているソフトウェア (Oracle HSM、Solaris、Solaris Cluster (該当する場合) を含む) のリビジョンレベルを記録し、選択したサブディレクトリ内の *readme* ファイルに情報のコピーを保存します。
9. 必要になったときにソフトウェアをすばやく復元できるように、選択したサブディレクトリに、ダウンロードした Oracle HSM、Solaris、および Solaris Cluster パッケージのコピーを保存します。

10. ここで停止します。構成をバックアップして、ファイルシステムを使用する準備ができました。

付録A

装置タイプの用語集

マスター構成ファイル (*mcf*) の「*Equipment Type*」フィールドの値は、Oracle Hierarchical Storage Manager and StorageTek QFS Software 内のデバイスおよびデバイス構成を識別します。装置タイプは、2 文字のコードで指定します。この用語集に記載されているコードは、サンプルを使用する際や既存の *mcf* (詳細は、*mcf(4)* のマニュアルページを参照してください) の内容を解釈する際のクイックリファレンスとして使用できます。

便宜上、コードを 3 つのセクションに分けてアルファベット順に記載しています。

A.1. 推奨される装置およびメディアのタイプ

このセクションでは、通常必要になる装置コードについて説明します。汎用装置コード (*rb*、*tp*、*od*) を表すコード、およびネットワーク接続ライブラリインタフェースと Oracle HSM 履歴を表すコードです。

汎用装置コード *rb*、*tp*、および *od* は、SCSI 接続のライブラリ、テープドライブ、および光学ディスクデバイス全般に使用できる推奨の装置タイプです。汎用装置タイプを指定すると、Oracle HSM が SCSI ベンダーコードに基づいて正しいタイプを自動的に設定します。

gXXX
XXX は *[0-127]* の範囲の整数で、*ma* ディスクキャッシュファミリセットの一部であるディスクデバイスのストライプ化グループです。

hy
オプションの Oracle HSM 履歴仮想ライブラリです。メディアカタログが格納されますが、ハードウェアへの関連付けはありません。エクスポートしたメディアの追跡に使用します。

ma
1 つ以上の専用の *mm* ディスクデバイス上のファイルシステムメタデータが保持される高パフォーマンスの QFS ファイルシステム。ファイルデータは別の *md*、*mr*、または *gXXX* データデバイスに格納されます。

md
ma ファイルシステムのファイルデータや *ms* ファイルシステムのデータおよびメタデータを格納するディスクデバイス。*md* デバイスはファイルデータを、小さい 4K バイトのディス

ク割り当て単位 (DAU)、および大きい 16-、32-、または 64K バイトの DAU で格納します。DAU のデフォルトは 64K バイトです。

mm
高パフォーマンス *ma* ファイルシステムのファイルシステムメタデータを格納するディスクデバイス。

mr
ma ファイルシステムのファイルデータを格納するディスクデバイス。*mr* デバイスはファイルデータを、8-65528K バイトの範囲の 8K バイトの倍数で自由に調整できる大きなディスク割り当て単位 (DAU) で格納します。DAU のデフォルトは 64K バイトです。

ms
Oracle HSM ファイルシステムで、ファイルシステムのメタデータをファイルデータを格納しているのと同じデバイスに格納します。

od
SCSI 接続光学ディスク。Oracle HSM は、SCSI ベンダーコードを使用して適切な装置タイプを自動的に設定します。

rb
SCSI 接続テープライブラリ。Oracle HSM は、SCSI ベンダーコードを使用して適切な装置タイプを自動的に設定します。

rd
SAM-Remote 疑似デバイス。マスター構成ファイル (*mcf*) では、対応する「*Equipment Identifier*」フィールドに、疑似デバイスへのパス (*/dev/samrd/rd2* など) が含まれている必要があります。対応する「*Family Set*」フィールドには、SAM-Remote サーバーのホスト名が含まれている必要があります。

sc
SAM-Remote クライアントシステム。マスター構成ファイル (*mcf*) では、対応する「*Equipment Identifier*」フィールドに、クライアントの SAM-Remote クライアント構成ファイルへのパスが含まれている必要があります。対応する「*Family Set*」フィールドには、サーバーのファミリセット名が含まれている必要があります。「*Additional Parameters*」フィールドには、クライアントのライブラリカタログファイルへのフルパスが含まれている必要があります。

sk
ネットワーク接続ライブラリとの Oracle StorageTek ACSLS インタフェース。マスター構成ファイル (*mcf*) では、対応する「*Equipment Identifier*」フィールドに、ACSLS インタフェースのパラメータファイルへのパスが含まれている必要があります。詳細は、*stk(7)* のマニュアルページを参照してください。

ss
SAM-Remote サーバー。マスター構成ファイル (*mcf*) では、対応する「*Equipment Identifier*」フィールドに、SAM-Remote サーバー構成ファイルへのパスが含まれている必要があります。対応する「*Family Set*」フィールドには、サーバーのファミリセット名が含まれている必要があります。この名前は、クライアント上の *mcf* の「*Family Set*」フィールドで使用される名前と一致する必要があります。

tp
SCSI 接続テープドライブ。Oracle HSM は、SCSI ベンダーコードを使用して適切な装置タイプを自動的に設定します。いいえ。ただし、*li* や *ti* などのより固有の装置コードを

使用する場合、一貫した方法でこれを行う必要があります。たとえば、*mcf* ファイルで *li* (LTO) テープ装置を指定する場合は、アーカイバ *.cmd* ファイルで *tp* 装置と同じ装置を参照できません。

A.2. その他の装置タイプとメディアタイプ

このセクションに表示されている装置タイプもサポートされます。

ほとんどの場合、汎用装置タイプ *rb*、*tp*、および *od* を使用して、SCSI 接続のライブラリ、テープドライブ、および光ディスクデバイスを識別することをお勧めしています。汎用装置タイプは、ハードウェアを SCSI ベンダー ID を使用して動的に特定するよう Oracle HSM に指示します。次のタイプコードは、あるメディアタイプから別のメディアタイプに移行する際に不可欠であり、管理のために役に立つことがあります。ただし、たとえば、これらをマスター構成ファイル (*mcf*) で使用すると、ある時点で実際のハードウェアと一致しなくなる可能性がある静的な装置構成がハードコーディングされます。

ac
Sun 1800、3500、または L11000 テープライブラリ。

at
Sony AIT-4 または AIT-5 テープドライブ。

cy
Cygnet 光学ディスクライブラリ。

d3
StorageTek D3 テープドライブ。

dm
Sony DMF ライブラリ。

ds
DocuStore または Plasmon 光学ディスクライブラリ。

dt
DAT 4-mm テープドライブ。

e8
Exabyte X80 ライブラリ。

fd
Fujitsu M8100 128 トラックテープドライブ。

h4
HP SL48 または SL24 ライブラリ。

hc
Hewlett Packard L9-/L20-/L60 シリーズライブラリ。

i7
IBM 3570 テープドライブ。

ic
IBM 3570 メディアチェンジャー。

il
IBM 3584 テープライブラリ。

li
LTO-3 以降のテープドライブ。

lt
Digital Linear Tape (DLT)、Super DLT、DLT-S4 テープドライブ。

me
Metrum ライブラリ。

mf
IBM マルチファンクション光学ドライブ。

mo
5.25 インチ消去可能光学ドライブ。

o2
12 インチ WORM ドライブ。

ov
Overland Data Inc. Neo シリーズテープライブラリ。

pd
Plasmon D シリーズ DVD-RAM ライブラリ。

q8
Qualstar 42xx、62xx、または 82xx ライブラリ。

s3
StorageTek SL3000 ライブラリ。

s9
Oracle StorageTek 97xx シリーズライブラリ。

se
StorageTek 9490 テープドライブ。

sf
StorageTek T9940 テープドライブ。

sg
StorageTek 9840C 以降のテープドライブ。

sl
Spectra Logic または Qualstar テープライブラリ。

st
StorageTek 3480 テープドライブ。

ti
StorageTek T10000 (Titanium) テープドライブ。

vt
Metrum VHS (RSP-2150) テープドライブ。

wo
5.25 インチ光学 WORM ドライブ。

xt
Exabyte (850x) 8-mm テープドライブ。

付録B

共有ファイルシステムでのマウントオプション

Oracle Hierarchical Storage Manager and StorageTek QFS Software 共有ファイルシステムは、いくつかのマウントオプションを使用してマウントできます。この章では、これらのオプションのいくつかについて、その役割のコンテキスト内で説明します。

B.1. 共有ファイルシステムのマウントオプション

ほとんどのマウントオプションは、*mount* コマンドを使用するか、*/etc/vfstab* ファイルに入力するか、または *samfs.cmd* ファイルに入力することによって指定できます。たとえば、次の */etc/vfstab* ファイルには、共有ファイルシステムのためのマウントオプションが含まれています。

```
sharefs - /sfs samfs - no shared,mh_write
```

いくつかのマウントオプションは、*samu* オペレータユーティリティーを使用して動的に変更できます。これらのオプションの詳細は、『*Oracle Hierarchical Storage Manager and StorageTek QFS samu コマンドリファレンス*』を参照してください。

これらのマウントオプションの詳細は、*mount_samfs* のマニュアルページを参照してください。

B.1.1. bg: バックグラウンドでのマウント

bg マウントオプションは、最初のマウント操作が失敗した場合は、それ以降のマウントの試行をバックグラウンドで実行することを指定します。デフォルトでは、*bg* は有効ではなく、マウント試行はフォアグラウンドで継続されます。

B.1.2. retry: ファイルシステムのマウントの再試行

retry マウントオプションは、システムでファイルシステムのマウントを試行する回数を指定します。デフォルトは 10000 です。

B.1.3. shared: Oracle HSM 共有ファイルシステムの宣言

shared マウントオプションは、ファイルシステムを Oracle HSM 共有ファイルシステムにすることを宣言します。ファイルシステムを Oracle HSM 共有ファイルシステムとしてマウントする

には、このオプションを */etc/vfstab* ファイル内に指定する必要があります。このオプションが *samfs.cmd* ファイル内、または *mount* コマンド上に存在してもエラー条件は発生しませんが、ファイルシステムは共有ファイルシステムとしてマウントされません。

B.1.4. minallocsz および maxallocsz: 割り当てサイズの調整

mount コマンドの *minallocsz* および *maxallocsz* オプションは、容量を K バイト単位で指定します。これらのオプションは、最小のブロック割り当てサイズを設定します。ファイルサイズが大きくなる場合は、追加リースが認められると、メタデータサーバーによってブロックが割り当てられます。この割り当ての初期サイズを指定するには、*-o minallocsz=n* を使用します。メタデータサーバーは、アプリケーションのアクセスパターンに応じてブロック割り当てのサイズを *-o maxallocsz=n* の設定まで増やすことができますが、この値を超えることはできません。

mount コマンド行のこれらの *mount* オプションは、*/etc/vfstab* ファイルまたは *samfs.cmd* ファイル内に指定できます。

B.1.5. rdlease、wrlease、および aplease: Oracle HSM 共有ファイルシステムでのリースの使用

デフォルトでは、ホストがファイルを共有するときに、入出力 *lease* をそれ自体およびそのクライアントに発行すると、Oracle HSM メタデータサーバーでファイルシステムの整合性が保持されます。リースによって、指定された期間内でファイルを操作するためのアクセス権が共有ホストに付与されます。*read lease* を使用すると、ホストがファイルデータを読み取ります。*write lease* を使用すると、ホストが既存のファイルデータを上書きします。*append lease* を使用すると、ホストがファイルの末尾に追加データを書き込みます。メタデータサーバーは、必要に応じてリースを更新できます。

したがって、Oracle HSM 共有ファイルシステムに対する読み取りおよび書き込みは、データに対する POSIX の動作に類似しています。ただし、メタデータに対しては、アクセス時間が変化しても、ほかのホストにはすぐにわからないことがあります。ファイルへの変更は、書き込みリースの最後にディスクにプッシュされます。読み取りリースが取得されると、新しく書き込まれたデータを表示できるように、システムは期限切れのキャッシュページをすべて無効にします。

次のマウントオプションは、リースの期間を設定します。

- *-o rdlease=number-seconds* は、読み取りリースの最大時間 (秒単位) を指定します。
- *-o wrlease=number-seconds* は、書き込みリースの最大時間 (秒単位) を指定します。

- *-o aplease=number-seconds* は、追加リースの最長時間 (秒単位) を指定します。

これら 3 つのいずれの場合も、*number-seconds* は [*15-600*] の範囲の整数です。各リースのデフォルトの時間は *30* 秒です。リースが有効な場合、ファイルを切り捨てることはできません。これらのリースの設定の詳細は、*mount_samfs* のマニュアルページを参照してください。

現在のメタデータサーバーが停止したためにメタデータサーバーを変更する場合は、リース時間を切り替え時間に加える必要があります。これは、代替メタデータサーバーが制御を引き継ぐには、その前にすべてのリースが期限切れになっていることが必要であるためです。

リース時間を短く設定しておくと、リースが期限切れになるごとに更新する必要があるため、クライアントホストとメタデータサーバーの間のトラフィックが増加します。

B.1.6. mh_write: マルチホスト読み取りと書き込みの有効化

mh_write オプションでは、複数ホストから同一ファイルへの書き込みアクセスが制御されます。メタデータサーバーホスト上でマウントオプションとして *mh_write* が指定されている場合は、Oracle HSM 共有ファイルシステムにより、複数のホストから同じファイルへの同時の読み取りと書き込みが可能になります。メタデータサーバーホストで *mh_write* を指定しないと、同時にファイルに書き込みができるホストは 1 つだけになります。

デフォルトでは、*mh_write* は無効になっており、*wrlease* マウントオプションの期間中ファイルに書き込めるのは 1 つのホストだけです。*mh_write* オプションが有効になった状態で Oracle HSM 共有ファイルシステムがメタデータサーバー上にマウントされている場合は、複数のホストから同じファイルへの同時の読み取りと書き込みを実行できます。

メタデータサーバー上で *mh_write* が有効になっている場合は、Oracle HSM で次のことがサポートされます。

- 複数の読み取りホストとページ入出力
- 複数の読み取りホストおよび書き込みホストと、書き込みがあった場合にのみ直接入出力
- 1 つの追加ホスト (その他のホストは読み取りまたは書き込みを行う) と、書き込みがあった場合にのみ直接入出力。

mh_write オプションを使用してファイルシステムをマウントしても、ロック動作は変わりません。ファイルロックは、*mh_write* が有効かどうかには関係なく同じ動作を行います。ただし、その他の点では、動作は一定ではない可能性があります。読み取りと書き込みが同時にあった場合、Oracle HSM 共有ファイルシステムは、ファイルへのすべてのホストアクセスに直接入出力を使用します。そのため、ページ整合入出力がその他のホストにただちに表示される

はずです。ただし、非ページ整合入出力では期限切れのデータが表示されたり、場合によってはファイルに書き込まれたりしますが、これは、このような状況を防止している通常のリースメカニズムが無効になるためです。

このため、複数のホストが同じファイルに同時に書き込む必要がある場合、およびホストされているアプリケーションがページ整合入出力を実行し、競合する書き込みを調整する場合にかぎり、*mh_write* オプションを指定してください。それ以外の場合は、データの不一致が発生する可能性があります。*mh_write* を付けて *flock()* を使用してホスト間を調整しても、整合性は保証されません。詳細は、*mount_samfs* のマニュアルページを参照してください。

B.1.7. min_pool: 並行スレッドの最小数の設定

min_pool マウントオプションは、Oracle HSM 共有ファイルシステムの並行スレッドの最小数を設定します。Oracle Solaris システムでのデフォルト設定は *min_pool=64* です。この設定は、Oracle Solaris ではスレッドプール内にアクティブスレッドが少なくとも 64 個存在することを示します。共有ファイルシステムのアクティビティーに応じて、*min_pool* の設定を [*8-2048*] の範囲の任意の値に調整できます。

min_pool マウントオプションは *samfs.cmd* ファイルに設定してください。これを */etc/vfstab* ファイル内か、またはコマンド行で設定した場合は無視されます。

B.1.8. meta_timeo: キャッシュされた属性の保持

meta_timeo マウントオプションは、システムがメタデータ情報に対するチェックを待つ間隔の長さを決定します。デフォルトでは、システムはメタデータ情報を 3 秒ごとにリフレッシュします。たとえば、新しく作成されたファイルがいくつか含まれている共有ファイルシステムで *ls* コマンドを入力すると、3 秒が経過するまですべてのファイルに関する情報が返されない可能性があります。このオプションの構文は *meta_timeo=seconds* で、*seconds* は [*0-60*] の範囲の整数です。

B.1.9. stripe: ストライプ化割り当ての指定

デフォルトでは、共有ファイルシステム内のデータファイルは、ラウンドロビン式ファイル割り当て方式を使用して割り当てられます。ファイルデータが複数のディスクにわたってストライプ化されるように指定するには、メタデータホストとすべての潜在的なメタデータホスト上で *stripe* マウントオプションを指定できます。デフォルトでは、非共有ファイルシステムのファイルデータは、ストライプ化方式で割り当てられることに注意してください。

ラウンドロビン式割り当てでは、ファイルは、各スライスまたはストライプ化グループ上にラウンドロビン式で作成されます。1 つのファイルの最大のパフォーマンスは、スライスまたはスト

ライプ化グループの速度になります。ファイル割り当て方式の詳細は、『*Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド*』(*Oracle HSM お客様向けドキュメントライブラリ*、*docs.oracle.com/en/storage*) を参照してください。

B.1.10. sync_meta: メタデータが書き込まれる頻度の指定

sync_meta オプションを *sync_meta=1* または *sync_meta=0* に設定できます。

デフォルト設定は *sync_meta=1* です。これは、メタデータが変更されるたびに Oracle HSM 共有ファイルシステムがファイルのメタデータをディスクに書き込むことを示します。この設定によってデータのパフォーマンスが低下しますが、データの整合性は保証されます。メタデータサーバーを変更する場合は、この設定が有効である必要があります。

sync_meta=0 を設定した場合、Oracle HSM 共有ファイルシステムは、メタデータをバッファーに書き込んでからディスクに書き込みます。この遅延書き込みによってより高いパフォーマンスが実現されますが、マシンの予定外の停止が発生したあとのデータの整合性は低下します。

B.1.11. worm_capable および def_retention: WORM 機能の有効化

worm_capable マウントオプションにより、ファイルシステムでは WORM ファイルがサポートされます。*def_retention* マウントオプションは、*def_retention=MyNdOhPm* の形式を使用して、デフォルトの保存期間を設定します。

この形式では、*M*、*N*、*O*、および *P* は負でない整数であり、*y*、*d*、*h*、および *m* は、それぞれ年、日、時、分を表します。これらの単位を任意に組み合わせることができます。たとえば *1y5d4h3m* は 1 年、5 日、4 時間、3 分、*30d8h* は 30 日と 8 時間、*300m* は 300 分をそれぞれ表します。この形式は、保存時間を分単位で指定していた旧バージョンのソフトウェアと下位互換性があります。

詳細は、『*Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド*』(*Oracle HSM お客様向けドキュメントライブラリ*、*docs.oracle.com/en/storage*) を参照してください。

付録C

アーカイブのための構成ディレクティブ

この付録では、Oracle Hierarchical Storage Manager ファイルシステムを構成するディレクティブと、関連するソフトウェア操作の一覧を表示します。それぞれのディレクティブは、1 つ以上のコンマ区切りフィールドで構成された 1 つのテキスト行です。関連するディレクティブは、一緒に Oracle HSM コマンド (*.cmd*) ファイルに格納されます。

この付録の残りには、3 つの主要な種類のディレクティブの概要が記載されています。

- アーカイブディレクティブ
- ステージングディレクティブ
- プレビュー要求ディレクティブ

追加情報については、Oracle HSM のマニュアルページを参照してください。

Oracle HSM コマンドファイルは、ここで説明されているようにコマンド行から構成することも、Oracle HSM Manager software を使用して構成することもできます。Oracle HSM Manager の詳細は、オンラインヘルプを参照してください。

C.1. アーカイブディレクティブ

このセクションには、*archiver.cmd* ファイルを構成するアーカイブディレクティブの使用法に関する情報が記載されています。アーカイブディレクティブは、ファイルのコピー、使用されるメディア、およびアーカイブソフトウェアの全体的な動作を制御するアーカイブセットを定義します。

アーカイブディレクティブには 4 つの基本タイプがあります。

- グローバルアーカイブディレクティブ
- ファイルシステムディレクティブ
- コピーパラメータ
- ボリュームシリアル番号 (VSN) 関連付けディレクティブ

グローバルディレクティブとファイルシステムディレクティブの両方が、ファイルのアーカイブ方法を制御します。ただし、アーカイバは、グローバルディレクティブを評価する前にファイルシステム固有のディレクティブを評価します。そのため、ファイルシステムディレクティブは、競

合が存在する場合にグローバルディレクティブをオーバーライドします。同様に、ファイルシステムディレクティブ内では、最初に表示されるディレクティブによって後続の競合するディレクティブがオーバーライドされます。

C.1.1. グローバルアーカイブディレクティブ

グローバルディレクティブはアーカイバ全体の動作を制御し、構成済みのすべてのファイルシステムに合わせて動作を最適化することを可能にします。グローバルディレクティブは、単一のキーワード、またはキーワードとその後に続く等号 (=) と追加のデータフィールドで構成されます。グローバルディレクティブは、*archiver.cmd* ファイルを開始し、最初の ファイルシステムディレクティブで終了します。

C.1.1.1. archivemeta: メタデータをアーカイブするかどうかの制御

archivemeta 指示は、ファイルシステムメタデータをアーカイブするかどうかを制御します。ファイルシステム内でファイルの移動やディレクトリ構造の変更が頻繁に行われる場合は、ファイルシステムメタデータをアーカイブします。ただし、ディレクトリ構造が適度に安定している場合は、メタデータのアーカイブを無効にして、リムーバブルメディアドライブが行うアクションを減らすことができます。デフォルトでは、メタデータはアーカイブされません。

このディレクティブの形式は、次のとおりです。

```
archivemeta=state
```

state には、*on* または *off* を指定します。デフォルトは *off* です。

メタデータのアーカイブ処理は、Version 1 と Version 2 のどちらのスーパーブロックを使用するかによって、次のように異なります。

- Version 1 ファイルシステムの場合、アーカイバはディレクトリ、リムーバブルメディアファイル、セグメント索引 i ノード、シンボリックリンクをメタデータとしてアーカイブします。
- Version 2 ファイルシステムの場合、アーカイバはディレクトリおよびセグメントインデックス i ノードをメタデータとしてアーカイブします。リムーバブルメディアファイルおよびシンボリックリンクは、データブロックではなく i ノードに格納されます。これらはアーカイブされません。シンボリックリンクは、データとしてアーカイブされます。

C.1.1.2. archmax: アーカイブファイルサイズの制御

archmax ディレクティブは、アーカイブ (*.tar*) ファイルの最大サイズを指定します。*target-size* 値に達すると、アーカイブファイルにそれ以上のユーザーファイルが追加

されることはありません。複数のサイズの大きいユーザーファイルが、1 つのアーカイブファイルに書き込まれます。

デフォルト値を変更するには、次のディレクティブを使用します。

```
archmax=media target-size
```

ここで *media* は、付録A「*装置タイプの用語集*」および *mcf* のマニュアルページで定義されているメディアタイプの 1 つ、*target-size* は、アーカイブファイルの最大サイズです。この値はメディアによって異なります。デフォルトでは、光ディスクに書き込まれるアーカイブファイルは最大 5M バイトです。テープの場合の最大アーカイブファイルのデフォルトサイズは、512M バイトです。

アーカイブファイルのサイズとして、大きいサイズを設定した場合も、小さいサイズを設定した場合も、それぞれに利点と欠点があります。たとえば、テープにアーカイブする場合、*archmax* を大きいサイズに設定すると、テープドライブの停止と開始の頻度が下がります。しかし、大きいアーカイブファイルを書き込むと、テープの終わりが早すぎて、大量のテープが無駄になる可能性があります。ベストプラクティスとしては、メディア容量の 5% を超える値に *archmax* 指示を設定しないでください。

archmax 指示は、個々のアーカイブセットに対して設定することもできます。

C.1.1.3. bufsize: アーカイババッファーサイズの設定

デフォルトでは、アーカイブ対象ファイルは、メモリーバッファーを使用してアーカイブメディアにコピーされます。*bufsize* ディレクティブを使用すると、デフォルト値以外のバッファーサイズを指定したり、オプションでバッファーをロックしたりできます。これらのアクションにより、一部の状況ではパフォーマンスが向上することがあります。さまざまな *buffer-size* 値を試すことができます。このディレクティブの形式は、次のとおりです。

```
bufsize=media buffer-size [lock]
```

ここでは:

- *media* は、付録A「*装置タイプの用語集*」と *mcf* のマニュアルページで定義されているメディアタイプの 1 つです
- *buffer-size* は、[*2-1024*] の範囲の数値です。デフォルトは *4* です。この値にメディアタイプの *dev_blksize* 値が乗算され、その結果がバッファーサイズとして使

用されます。*dev_blksize* 値は *defaults.conf* ファイルで指定されます。詳細は、*defaults.conf* のマニュアルページを参照してください。

- *lock* は、アーカイブコピーの作成時にアーカイバがロックバッファーを使用できるかどうかを指示します。

  lock が指定されている場合、*sam-arcopy* の動作中は、アーカイバがメモリー内のアーカイブバッファーにファイルロックを設定します。この動作により、入出力リクエストごとにバッファーをロックまたはロック解除することに伴うオーバーヘッドが回避され、その結果システムの CPU 時間を短縮できます。

  lock 引数は、大容量メモリーを備えた大型システムだけで指定する必要があります。十分なメモリーがないと、メモリー不足状態となります。*lock* 引数が有益なのは、アーカイブ対象のファイルに対して直接入出力が使用可能となっている場合のみです。デフォルトでは、*lock* は指定されておらず、アーカイブ用を含むすべての直接入出力バッファーに、ファイルシステムによってロックが設定されています。

-bufsize および *-lock* アーカイブセットコピーパラメータを使用すると、アーカイブセットごとにバッファーサイズとロックを指定できます。詳細は、「アーカイブのコピーディレクティブ」を参照してください。

C.1.1.4. drives: アーカイブに使用するドライブ数の制御

デフォルトの場合、アーカイバはアーカイブ用自動ライブラリにあるすべてのドライブを使用します。使用するドライブ数を制限するには、*drives* ディレクティブを使用します。このディレクティブの形式は、次のとおりです。

```
drives=media-library count
```

ここで *media-library* は、*mcf* ファイルで定義された自動ライブラリのファミリセット名、*count* は、アーカイブで使用できるドライブの数です。

この目的でアーカイブセットコピーパラメータ *-drivemax*、*-drivemin*、および *-drives* を使用することもできます。詳細は、「アーカイブのコピーディレクティブ」を参照してください。

C.1.1.5. examine: アーカイブスキャンの制御

examine ディレクティブは、アーカイバがアーカイブの準備ができているファイルを特定するために使用する *method* を設定します。

```
examine=method
```

method は次のいずれかのディレクティブです。

- デフォルトである *noscan* は継続アーカイブを指定します。最初のスキャンのあと、内容が変更されてアーカイブが必要なときにのみ、ディレクトリがスキャンされます。ディレクトリおよび i ノード情報はスキャンされません。このアーカイブ方法は、特にファイル数が 1,000,000 を超えるファイルシステムで、スキャンアーカイブよりも高いパフォーマンスが得られます。
- *scan* は、スキャンアーカイブを指定します。ファイルシステムディレクトリがはじめてスキャンされた後で、常に i ノードがスキャンされます。
- *scandirs* はスキャンアーカイブを指定します。ディレクトリは常にスキャンされます。i ノード情報はスキャンされません。

  アーカイバは、*no_archive* 属性が設定されたディレクトリをスキャンしません。そのため、変更されていないファイルが含まれるディレクトリに対してこの属性を設定することで、スキャン時間を短縮できます。
- *scaninodes* は、スキャンアーカイブを指定します。i ノードは常にスキャンされます。ディレクトリ情報はスキャンされません。

C.1.1.6. interval: アーカイブ間隔の指定

アーカイバは、マウントされているすべてのアーカイブ対応ファイルシステムのステータスを定期的にチェックします。タイミングは、各ファイルシステムでのスキャン操作間の時間である*アーカイブ間隔*によって制御されます。アーカイブ間隔を変更するには、*interval* ディレクティブを使用します。

継続アーカイブが設定されておらず、*startage*、*startsize*、または *startcount* のどのパラメータも指定されていない場合にのみ、*interval* ディレクティブは完全スキャンを開始します。継続アーカイブが設定されている (*examine=noscan*) 場合、*interval* ディレクティブはデフォルトの *startage* 値として機能します。このディレクティブの形式は、次のとおりです。

```
interval=time
```

time には、ファイルシステムのスキャンを行う時間間隔を指定します。デフォルトでは、*time* は秒単位と見なされ、値は *600* (10 分) です。別の時間単位 (分や時など) も指定できます。

アーカイバは、*samu* ユーティリティーの *arrun* コマンドを受信すると、すべてのファイルシステムのスキャンをすぐに開始します。*archiver.cmd* ファイルで *examine=scan* ディレクティ

ブも指定されている場合は、*arrun* または *arscan* が実行されたあとで、スキャンが実行されます。

ファイルシステムに *hwm_archive* マウントオプションが設定されている場合は、アーカイブ間隔を自動的に短縮できます。ファイルシステムの利用率が高位境界値を超えると、アーカイバはスキャンを開始します。*high=percent* マウントオプションは、高位境界値をファイルシステムに設定します。

アーカイブ間隔の指定方法の詳細は、*archiver.cmd* および *mount_samfs* のマニュアルページを参照してください。

C.1.1.7. logfile: アーカイバログファイルの指定

アーカイバは、アーカイブ、再アーカイブ、またはアーカイブ解除された各ファイルに関する情報を含むログファイルを出力できます。ログファイルは、アーカイブアクションを連続的に記録したものです。デフォルトでは、アーカイバログファイルは有効になりません。ログファイルを指定するには、*logfile* ディレクティブを使用します。このディレクティブの形式は、次のとおりです。

```
logfile=pathname
```

pathname には、ログファイルの絶対パスとファイル名を指定します。*logfile* 指示は、個々のファイルシステムに対して設定することもできます。

アーカイバログファイルは、破損したり失われたりしたファイルシステムを回復するために不可欠であり、モニタリングと分析に役立つことがあります。そのため、アーカイバログを有効にして、バックアップしてください。詳細は、『*Oracle Hierarchical Storage Manager and StorageTek QFS インストールおよび構成ガイド*』を参照してください。

C.1.1.8. notify: イベント通知スクリプトの名前変更

notify 指示は、アーカイバのイベント通知スクリプトファイルの名前を設定します。このディレクティブの形式は、次のとおりです。

```
notify=filename
```

filename に、アーカイバのイベント通知スクリプトを含むファイルの名前、またはフルパスを指定します。デフォルトのファイル名は */etc/opt/SUNWsamfs/scripts/archiver.sh* です。

アーカイバはこのスクリプトを実行して、さまざまなイベントをサイト固有の方法で処理します。このスクリプトは、第 1 引数のキーワード *emerg*、*alert*、*crit*、*err*、*warning*、*notice*、*info*、*debug* のいずれかで呼び出されます。

そのほかの引数については、デフォルトのスクリプトで説明されています。詳細は、*archiver.sh* のマニュアルページを参照してください。

C.1.1.9. ovflmin: ボリュームオーバーフローの制御

ボリュームオーバーフローが有効になっていると、アーカイバは複数のボリュームにまたがるアーカイブファイルを作成できます。ファイルサイズが指定された最小サイズを超えると、アーカイバはこのファイルの残りの部分を同じタイプの別のボリュームに書き込みます。各ボリュームに書き込まれたファイル部分のことを、「セクション」と呼びます。*sls* コマンドは、アーカイブコピーの一覧を表示して、各ボリュームにあるファイルの各セクションを示します。

アーカイバは、*ovflmin* 指示によってボリュームオーバーフローを制御します。デフォルトでは、ボリュームオーバーフローは使用不可となっています。ボリュームオーバーフローを有効にするには、*archiver.cmd* ファイルで *ovflmin* ディレクティブを使用します。このディレクティブの形式は、次のとおりです。

```
ovflmin = media minimum-file-size
```

ここで *media* は付録A「*装置タイプの用語集*」および *mcf* のマニュアルページで定義されているメディアタイプの 1 つで、*minimum-file-size* はボリュームオーバーフローをトリガーする最小ファイルのサイズです。*ovflmin* 指示は、個々のアーカイブセットに対して設定することもできます。

ボリュームオーバーフローは、及ぼす影響を検討したうえで慎重に使用してください。複数のボリュームをまたぐファイルの場合は、障害回復とリサイクルが非常に難しくなります。ボリュームオーバーフローファイルでは、チェックサムは生成されません。チェックサムの使用方法の詳細は、*ssum* のマニュアルページを参照してください。

C.1.1.10. scanlist_squash: スキャンリストの連結の制御

scanlist_squash パラメータは、スキャンリストの連結を制御します。デフォルトの設定は無効 (*off*) です。このパラメータはグローバルに使用することも、特定のファイルシステム用に使用することもできます。

on にすると、アーカイバが共通の親ディレクトリから下に再帰的にスキャンできるように、このディレクティブはディレクトリツリー内のサブディレクトリのスキャンリストを統合します。多数

のファイルとサブディレクトリがファイルシステム内で変更されている場合、スキャンリストの統合によって、アーカイブパフォーマンスが大幅に低下することがあります。

C.1.1.11. setarchdone: archdone フラグ設定の制御

setarchdone グローバルディレクティブは、アーカイブされることがないファイルで *archdone* フラグが設定されるかどうかを制御します。このディレクティブの形式は、次のとおりです。

```
setarchdone=state
```

state は *on* または *off* のいずれかです。*examine* ディレクティブが *scandirs* または *noscan* に設定されている場合、デフォルトは *off* です。

archdone フラグは、フラグ付きのファイルを無視するようにアーカイブ処理に指示します。通常、ファイルの指定されたコピーがすべて作成されたら、ファイルがあとで変更されるまで、またはファイルがあとで変更されないかぎり、後続のアーカイブ操作でファイルがスキップされるように、アーカイブ処理は *archdone* フラグを設定します。

ただし、*setarchdone* が *on* に設定されている場合、アーカイブ処理は、アーカイブ基準を満たさないためにアーカイブされることがない、アーカイブされていないファイルを特定してフラグを付けます。これによって将来のアーカイブのオーバーヘッドを削減できますが、ファイルの評価によってオーバーヘッドが即時に増加して、パフォーマンスに悪影響を与える可能性があります。

C.1.1.12. wait: アーカイバ起動の遅延

wait ディレクティブを使用すると、アーカイバは、*samcmd* コマンド、*samu* インタフェース、または Oracle HSM Manager からの開始シグナルを待機します。このディレクティブの形式は、次のとおりです。

```
wait
```

デフォルトでは、*sam-fsd* 初期化コマンドの実行時にアーカイバは自動的に開始します。

wait 指示は、個々のファイルシステムに対して設定することもできます。

C.1.2. ファイルシステムディレクティブ

ファイルシステムディレクティブは、特定のファイルシステムのアーカイブ動作を定義します。

- **fs**: ファイルシステムの指定
- ***copy-number* [*archive-age*]**: ファイルシステムメタデータの複数コピーの指定
- ファイルシステムディレクティブとしての **interval**、**logfile**、**scanlist**

C.1.2.1. fs: ファイルシステムの指定

各 *fs=file-system-name* ディレクティブでは、指定されたファイルシステム *file-system-name* にのみ適用される一連のアーカイブディレクティブが導入されます。このディレクティブの形式は、次のとおりです。

```
fs=file-system-name
```

file-system-name は、*mcf* ファイルで定義されているファイルシステム名です。

fs= ディレクティブのあとに配置された汎用ディレクティブとアーカイブセット関連付けディレクティブは、指定したファイルシステムにのみ適用されます。

C.1.2.2. *copy-number* [*archive-age*]: ファイルシステムメタデータの複数コピーの指定

ファイルシステムメタデータには、ファイルシステムにおけるパス名が含まれます。メタデータの複数のコピーが必要な場合は、*archiver.cmd* ファイルの *fs=* ディレクティブの直後にコピー定義を配置します。

```
copy-number [archive-age]
```

時間は、整数と時間の単位の 1 つ以上の組み合わせで表されます。単位には、*s* (秒)、*m* (分)、*h* (時)、*d* (日)、*w* (週)、および *y* (年) が含まれます。ディレクトリが頻繁に変更される場合に、複数のメタデータコピーを指定すると、これによってファイルシステムがメタデータテープボリュームをマウントする頻度が高くなる可能性があります。そのため、デフォルトでは、Oracle HSM は、メタデータの単一のコピーのみを作成します。

この例では、*fs=samma1* ファイルシステムについてメタデータのコピー *1* が 4 時間 (*4h*) 後に、コピー *2* が 12 時間 (*12h*) 後に作成されます。

```
# General Directives
archivemeta = off
examine = noscan
# Archive Set Assignments
fs = samma1
1 4h
2 12h
```

C.1.2.3. ファイルシステムディレクティブとしての interval、logfile、scanlist

いくつかの指示は、すべてのファイルシステムを対象とするグローバル指示として指定することも、1 つのファイルシステムだけを対象とする指示として指定することもできます。これらの指示については、次のセクションで説明されています。

- **interval**: アーカイブ間隔の指定
- **logfile**: アーカイバログファイルの指定
- **scanlist_squash**: スキャンリストの連結の制御
- **wait**: アーカイバ起動の遅延

C.1.3. *archive-set-name*: アーカイブセット割り当てディレクティブ

アーカイブセット割り当てディレクティブは、同時にアーカイブされるファイルを指定します。次で説明する幅広い選択基準を使用すると、非常に細かくファイルを指定できます。ただし、やむをえない場合を除いて、使用は避けてください。通常は、可能なかぎり包括的なアーカイブセットを最小数だけ構成するようにしてください。アーカイブセットでは、アーカイブメディアのセットが排他的に使用されます。そのため、過度に制限された割り当て基準によって個別に定義されたアーカイブセットが多数あると、メディアの利用率が低くなり、システムのオーバーヘッドが高くなり、パフォーマンスが低下します。極端なケースでは、ライブラリ内に十分な容量が残っているにもかかわらず、使用可能なメディアの不足が原因でジョブが失敗する可能性があります。

アーカイブセット割り当て指示の形式は、次のとおりです。

```
archive-set-name path [-access interval [-nftv]] [-after date-time] [-minsize size] [-maxsize size] [-
user username] [-group groupname] [-name regex]
```

ここでは:

- *archive-set-name* は、管理者によって定義されたアーカイブセットの名前です。

  名前には、最大 29 文字の大文字と小文字 [*A-Za-z*]、数字 [*0-9*]、および下線 (_) を任意の組み合わせで含めることができますが、先頭文字は文字である必要があります。空白文字を含むその他の文字は含めることができず、独自のアーカイブセットには Oracle HSM の特殊なアーカイブセット *no_archive* と *all* の名前は使用できません。
- *path* は、ファイルシステム内でアーカイブ処理を開始するサブディレクトリのマウントポイントを基準とする相対パスを指定します。開始ディレクトリとそのサブディレクトリ内のすべ

てのファイルがアーカイブされます。ファイルシステム内のすべてのファイルを含めるには、ドット (*.*) 文字を使用します。パスの先頭にスラッシュ (*/*) を使用することはできません。

- *-access* は、*interval* によって指定された期間アクセスされていないファイルを再アーカイブします。*interval* は整数で、*s* (秒)、*m* (分)、*h* (時)、*d* (日)、*w* (週)、および *y* (年) のいずれかの単位があとに付きます。

  このパラメータを使用すると、あまり使用されないファイルの再アーカイブを、高コストのメディアから低コストのメディアにスケジュールできます。ソフトウェアは、ファイルのアクセス時間と変更時間の妥当性検査を行い、これらの時間がファイルの作成時間以降であり、さらにファイルの検証時間以前であることを確認します。*-nftv* (ファイル時間の検証なし) パラメータは、この検証を無効にします。
- *-after* は、*date-time* のあとで作成または変更されたファイルのみをアーカイブします。*date-time* は、*YYYY-MM-DD [hh:mm:ss] [Z]* 形式の式であり、*YYYY*、*MM*、*DD*、*hh*、*mm*、および *ss* は、それぞれ年、月、日、時、分、および秒を表す整数です。オプションの *Z* パラメータは、タイムゾーンを協定世界時 (UTC) に設定します。デフォルトは *00:00:00* と現地時間です。
- *-minsize* および *-maxsize* は、指定された *size* より大きいか小さいファイルのみをアーカイブします。*size* は整数で、*b* (バイト)、*k* (K バイト)、*M* (M バイト)、*G* (G バイト)、*T* (T バイト)、*P* (P バイト)、*E* (E バイト) のいずれかの単位があとに付きます。
- *-user username* および *-group groupname* は、指定されたユーザーまたはグループ (あるいはその両方) に属するファイルのみをアーカイブします。
- *-name* は、正規表現 *regex* で定義されたパターンに一致するパスおよびファイル名を含むすべてのファイルをアーカイブします。

C.1.3.1. アーカイブのコピーディレクティブ

アーカイバは、デフォルトでアーカイブセット内のファイルのアーカイブ経過時間が 4 分であるときに、それらのファイルに対してアーカイブのコピーを 1 つ書き込みます。デフォルトの動作を変更するには、アーカイブのコピーディレクティブを使用します。アーカイブのコピーディレクティブは、関連するアーカイブセット割り当てディレクティブの直あとに配置する必要があります。

アーカイブのコピーディレクティブは、*1*、*2*、*3*、*4* のいずれかの *copy-number* 値から始まります。この数字のあとに、そのコピーのアーカイブ特性を指定する 1 つまたは複数の引数が続きます。アーカイブのコピーディレクティブの形式は、次のとおりです。

```
copy-number [archive-age] [-release [attribute] [-norelease][-stage[attribute] [unarchive-age]
```

ここでは:

- オプションの *archive-age* パラメータは、新規または変更済みのファイルがアーカイブ対象になる前に、ディスクキャッシュ内に存在しなければならない時間です。整数と時間単位の 1 つ以上の組み合わせで *archive-age* を指定します。単位には **s** (秒)、**m** (分)、*h* (時間)、*d* (日)、*w* (週)、および *y* (年) が含まれます。デフォルトは *4m* (4 分) です。
- オプションの *-release* パラメータは、アーカイブコピーが作成された直後に、ファイルで使用されているディスク領域が解放されるように Oracle HSM リリーサソフトウェアをクリアします。オプションの解放*属性*は、*-a*、*-n*、または *-d* です。*-a* (結合ステージング) 属性は、アーカイブセットから解放されたファイルのいずれかにアクセスしたときに、ソフトウェアがこれらのファイルをすべてステージングするように要求します。*-n* 属性は、ソフトウェアがアーカイブメディアからファイルを直接読み取り、ステージングしないように要求します。*-d* 属性は、デフォルトのステージング動作をリセットします。
- オプションの *-norelease* パラメータは、*-norelease* マークが付けられたコピーがすべて作成されるまで、ファイルで使用されているディスク領域を解放するように Oracle HSM リリーサソフトウェアをクリアしません。
- *-release -norelease* は同時に使用され、*-release -norelease* フラグが付けられたすべてのコピーが作成された直後に、Oracle HSM ソフトウェアがファイルで使用しているディスク領域を解放するように要求します。Oracle HSM は、リリーサプロセスの実行を待機しません。
- オプションの *-stage* パラメータ。オプションの解放*属性*は *-a*、*-c copy-number*、*-f*、*-I*、*-i input_file*、*-w*、*-n*、*-p*、*-V*、*-x*、*-r*、*-d* です。ここでは:

  -a (結合ステージング) は、アーカイブセットのファイルのいずれかにアクセスしたときに、これらのファイルをすべてステージングするように要求します。

  -c copy-number は、ソフトウェアが指定したコピー番号からステージングするように要求します。

  -n は、ソフトウェアがアーカイブメディアからファイルを直接読み取り、ステージングしないように要求します。

  -w は、各ファイルが正常にステージングされるまでソフトウェアが待機してから、続行するように要求します (*-d* または *-n* では無効です)。

  -d は、デフォルトのステージング動作をリセットします。
- *unarchive-age* パラメータは、ファイルのアーカイブコピー再利用のため、メディア上の空き領域にアーカイブ解除する前に、アーカイブ内に存在しなければならない時間を指定します。時間は、整数と時間単位を 1 つ以上組み合わせて表現されます。単位には *s* (秒)、*m* (分)、*h* (時間)、*d* (日)、*w* (週)、および *y* (年) が含まれます。

下記の例には、アーカイブセット *allsamma1* の 2 つのコピーディレクティブが含まれています。1 つ目のディレクティブは、アーカイブ経過時間が 5 分 (*5m*) に達するまで、コピー *1* を解放しません。2 つ目のディレクティブは、アーカイブ経過時間が 1 時間 (*1h*) に達するまでコピー *2* を解放せず、アーカイブ解除期間の 7 年 6 か月 (*7y6m*) に達するとコピー *2* をアーカイブ解除します。

```
# Archive Set Assignments
fs = samma1
logfile = /var/adm/samma1.archive.log
allsamma1 .
    1 -norelease 5m
    2 -norelease 1h 7y6m
```

C.1.4. コピーパラメータ

コピーパラメータは、アーカイブセットで指定されたコピーの作成方法を定義します。*archiver.cmd* ファイルのアーカイブセットのコピーパラメータセクションは、*params* ディレクティブで始まり *endparams* ディレクティブで終わります。

```
params
allsets -sort path -offline_copy stageahead
allfiles.1 -startage 10m -startsize 10M -drives 10 -archmax 1G
allfiles.2 -startage 1h -startsize 1G -drives 2 -archmax 10G -reserve set
endparams
```

各コピーパラメータの形式は、次のとおりです。

archive-set-name[.copy-number][R] [-startage time] [-startcount count] [-startsize size] [-archmax maximum-size] [-bufsize=buffer-size] [-drivemax maximum-size] [-drivemin minimum-size] [-drives number] [-fillvsns] [-lock] [-offline_copy method] [-sort criterion] [-rsort criterion] [-recycle_dataquantity size] [-recycle_hwm percent] [-recycle_ignore] [-recycle_mailaddr mail-address] [-recycle_mingainpercentage] [-recycle_vsncountcount] [-recycle_minobs percentage] [-unarchagetime_ref] [-tapenonstop] [-reserve keyword] [-priority multiplier ranking]

ここでは:

- *archive-set-name* は、ファイルシステムディレクティブまたは特殊なディレクティブ *allsets* でアーカイブセット割り当てディレクティブによって定義されているアーカイブセットの名前です。このアーカイブセットは、指定されたコピーパラメータを定義済みのすべてのアーカイブセットに適用します。個々のアーカイブセットのパラメータを指定する前に、最初に *allsets* のパラメータを設定します。そうしないと、個々のアーカイブセットのパラメータによって *allsets* の指定がオーバーライドされ、その目的が果たされなくなります。
- *.copy-number* は、指定されたコピーパラメータの適用を、*copy-number* で指定されたアーカイブコピーに制限します。ここで *copy-number* は、[*1-4*] の整数の範囲であり、オプションの *R* は、パラメータの適用を再アーカイブのコピーに制限します。

- *-startage time* は、最初のファイルがアーカイブ要求に追加されたときから、アーカイブ処理が実際に開始されたときまでの間の間隔を指定します。*time* に整数と時間単位の 1 つ以上の組み合わせを指定します。単位には *s* (秒)、*m* (分)、*h* (時)、*d* (日)、*w* (週)、および *y* (年) が含まれます。デフォルトは *2h* (2 時間) です。
- *-startcount count* は、アーカイブ要求内のファイルの最小数を指定します。アーカイブ処理を待機しているファイルの数がこのしきい値に達すると、アーカイブ処理が開始されます。デフォルトでは、*count* は設定されません。
- *-startsize size* は、アーカイブ要求の最小サイズをバイト単位で指定します。アーカイブ処理を待機しているファイルの合計サイズがこのしきい値に達すると、アーカイブ処理が開始されます。デフォルトでは、*size* は設定されません。
- *-archmax* は、アーカイブファイルのサイズを *maximum-size* までに制限します。ここで *maximum-size* は、メディアによって異なります。磁気テープの場合、デフォルトの最大アーカイブファイルサイズは 512M バイトです。光ディスクに書き込まれるアーカイブファイルは、最大 5M バイトです。

  同じ名前のグローバルアーカイブディレクティブについては、「**archmax**: アーカイブファイルサイズの制御」を参照してください。
- *-bufsize= buffer-size* は、アーカイブメディアに書き出されるときにアーカイブファイルを保持するバッファーのサイズを *buffer-size*dev_blksize* に設定します。ここで *buffer-size* は、[*2-32*] の範囲の整数、*dev_blksize* は、*defaults.conf* ファイルでメディアタイプに指定されたブロックサイズです。デフォルトは *4* です。
- *-drivemax* は、1 台のドライブを使用してアーカイブされるデータの量を *maximum-size* メガバイト以下に制限します。ここで *maximum-size* は整数です。デフォルトでは、*maximum-size* は指定されていません。

  -drives パラメータを使用して複数のドライブが指定されている場合、1 台の任意のドライブに書き込まれるデータの量を制限すると、ワークロードを平準化して、ドライブ全体の利用率を改善できます。
- *-drivemin minimum-size* は、1 台のドライブを使用してアーカイブされるデータの量を *minimum-size* メガバイト以上に制限します。ここで *minimum-size* は整数です。デフォルトは、*-archmax* 値 (指定されている場合)、または *defaults.conf* ファイルに一覧表示されたメディアタイプの値です。

  ドライブに書き込まれるデータの量に下限を設定すると、ドライブの利用率と効率性を改善できます。*minimum-size* は、転送時間がメディアのロード、位置設定、およびアンロードに必要な時間を大幅に上回るよう十分な長さに設定してください。*-drivemin* を指定すると、データ転送が十分に長い場合にのみ複数のドライブが使用されます。

- *-drives number* は、アーカイブ処理で使用されるドライブ数を *number* までに制限します。ここで *number* は整数です。デフォルトは **1** です。

  アーカイブセットに含まれるファイルのサイズが大きい、またはファイルの数が多い場合、ドライブの最大数を大きく設定すると、パフォーマンスを改善できます。使用可能なドライブの動作速度が異なる場合は、複数のドライブを指定することで、このようなばらつきを平準化して、アーカイブ処理の効率性を高めることもできます。
- *-fillvsns* は、小さいアーカイブファイルを使用して、より完全にアーカイブメディアボリュームを満杯にするようにアーカイブ処理プロセスに強制します。

  デフォルトでは、アーカイバはアーカイブコピー内のすべてのファイルを保持するのに十分な領域のあるボリュームを選択します。このため、アーカイブファイルが大きくなり、多数のカートリッジにある残りの容量に収まらなくなります。その結果、メディア全体が十分に利用されなくなります。この問題は *-fillvsns* パラメータで対処できますが、その代償としてメディアのマウント、位置設定操作、およびアンマウントが多くなり、これらのすべてによってアーカイブ処理およびステージング処理のパフォーマンスが低下します。
- *-lock* は、直接入出力を使用してアーカイブコピーを作成するときに、ロックバッファーを使用するように指示します。ロックバッファーを使用することによってバッファーのページングが回避され、直接入出力のパフォーマンスが改善されます。

  使用可能なメモリーが制限されているシステム上で *-lock* パラメータを指定すると、メモリー不足の状態になる可能性があります。デフォルトでは、ロックバッファーは指定されていないため、ファイルシステムがアーカイブバッファーの制御を保持します。
- *-offline_copy method* は、すでにファイルがディスクキャッシュから解放されているときに、アーカイブコピーを作成する方法を指定します。*method* には *direct*、*stageahead*、*stageall*、または *none* を指定できます。

  単一のアーカイブコピーが作成された直後にファイルが解放される可能性があるため、残りのコピーはオフラインのコピーから作成する必要があります。*-offline_copy* メソッドを指定すると、使用可能にできるドライブの数とディスクキャッシュの空き領域に合わせて、コピープロセスを調整できます。

  direct は、2 台のドライブを使用して、オフラインボリュームからアーカイブボリュームに直接ファイルをコピーします。適切なバッファー領域を確保するには、このメソッドを使用するときに、*stage_n_window* マウントオプションで設定された値を大きくします。

  stageahead は、アーカイブファイルをコピー先に書き込む間に、次のアーカイブファイルをステージングします。

stageall は、1 台のドライブを使用して、アーカイブ処理の前にすべてのファイルをディスクキャッシュにステージングします。このメソッドを使用する場合は、ディスクキャッシュがファイルを保持できるだけの大きさであることを確認してください。

none (デフォルト) は、アーカイブボリュームにコピーする前に、必要に応じてファイルをディスクキャッシュにステージングします。

- *-sort* は、アーカイブ処理の前に、*criterion* でファイルをソートします。ここで *criterion* は、*age*、*priority*、*size*、または *none* です。

  age は、変更時間 (最古から最新まで) でのソートを指定します。

  path (デフォルト) は、フルパス名でのソートを指定します。これにより、同じディレクトリに存在するファイルがアーカイブメディア上にまとめられます。

  priority は、アーカイブ処理のソートを優先順位高から低の順に指定します。

  size は、ファイルをファイルサイズ最小から最大の順でソートします。

  none は、ソートを指定せず、ファイルシステムで発生した順序でファイルをアーカイブします。
- *-rsort criterion* は、*-sort* と同様に、ただし逆順でファイルを *criterion* でソートします。
- *-recycle_dataquantity size* は、リサイクラが再アーカイブ対象としてスケジュールするデータの量を *size* バイトに制限します。ここで *size* は整数です。

  リサイクラは、有効なアーカイブファイルのアーカイブボリュームを空にする必要があるときに、再アーカイブ処理をスケジュールします。リサイクル対象として選択したボリュームの実際の数は、*-recycle_vsncount* パラメータによって異なる可能性もあることに注意してください。デフォルトは *1073741824* (1G バイト) です。
- *-recycle_hwm percent* は、リムーバブルメディアのリサイクル処理が開始されるメディアの最大利用率 (高位境界値または *hwm*) を設定します。ディスクメディアの場合、このパラメータは無視されます (下記の *-recycle_minobs* を参照)。デフォルトは *95* です。
- *-recycle_ignore* では、リサイクルプロセスの通常実行は許可されますが、アーカイブセット内では実際にメディアのリサイクルは実行できません。テスト用に使用されます。
- *-recycle_mailaddr mail-address* は、リサイクラの情報メッセージを *mail-address* に転送します。デフォルトでは、メールは送信されません。

- *-recycle_mingain* は、リサイクル対象のボリュームの選択を、少なくとも指定した *percentage* の空き領域が増加する数に制限します。デフォルトは *50* です。
- *-recycle_vsncount* は、リサイクラが再アーカイブ処理をスケジュールするボリュームの数を *count* に制限します。リサイクル対象として選択したボリュームの実際の数は、*-recycle_dataquantity* パラメータによって異なる可能性もあります。ディスクメディアの場合、このパラメータは無視されます。デフォルトは *1* です。
- *-recycle_minobs* は、有効なファイルの再アーカイブ処理および元の *tar* ファイルの最終的な削除をトリガーする、ディスク常駐のアーカイブファイル内にある古いファイルの *percentage* を設定します。リムーバブルメディアの場合、このパラメータは無視されます (上記の *-recycle_hwm* を参照)。デフォルトは *50* です。
- *-unarchage* は、アーカイブ解除期間を計算するための参照時間を *time_ref* に設定します。ここで *time_ref* は、ファイルアクセス時間を表す *access* (デフォルト) または変更時間を表す *modify* です。
- *-tapenonstop* は、リムーバブルメディアファイルを閉じることなく、アーカイブファイルの最後に単一のテープマークとファイルの終わり (EOF) ラベルを書き込みます。これによって、複数のアーカイブファイルの転送が高速化されますが、アーカイブセット全体がテープに書き込まれるまでテープカートリッジをアンロードできなくなります。デフォルトでは、Oracle HSM ソフトウェアは、アーカイブファイルの最後にあるファイルの終わりラベルのあとに 2 つの追加のテープマークを書き込むことで、テープファイルを閉じます。
- *-reserve keyword* は、指定されたアーカイブセットを排他的に使用するためにリムーバブルメディアボリュームを予約します。アーカイブセットのファイルを保持するために最初にボリュームを使用すると、ソフトウェアは、指定された 1 つ以上のキーワード *fs*、*set* と、*dir* (ディレクトリ)、*user*、または *group* のいずれかまたはこれらの両方に基づいて、一意の予約名をボリュームに割り当てます。

  fs は、ファイルシステム名を予約名に含めます。*arset.1 -reserve fs*。

  set は、アーカイブセット割り当てディレクティブからのアーカイブセット名を予約名に含めます (*all -reserve set*)。

  dir は、アーカイブセット割り当てディレクティブで指定されたディレクトリパスの最初の 31 文字を予約名に含めます。

  user は、アーカイブファイルに関連付けられたユーザー名を含めます。*arset.1 -reserve user*。

  group は、アーカイブファイルに関連付けられたグループ名を含めます。*arset.1 -reserve group*。

状況によっては、セット別にボリュームを予約すると便利なことがあります。ただし、ソフトウェアでメディアを選択する場合よりも本質的に非効率です。ボリュームを予約する場合は、システムは、より頻繁にカートリッジのマウント、アンマウント、および配置を行う必要があり、オーバーヘッドが増加してパフォーマンスが低下します。予約スキームの制約が大きいと、使用可能なメディアが十分に利用されず、極端な例では使用可能なメディアがないためにアーカイブ処理が失敗することがあります。

- *-priority multiplier ranking* は、前述の *sort priority* パラメータとともに使用したときのファイルのアーカイブの優先順位を変更します。*ranking* は、[*(-3.400000000E+38)-3.400000000E+38*] (-3.402823466x10^{38} - 3.402823466x10^{38}) の範囲の実数です。*multiplier* は、相対的な *ranking* を変更するアーカイブ特性であり、リスト *age*、*archive_immediate*、*archive_overflow*、*archive_loaded*、*copies*、*copy1*、*copy2*、*copy3*、*copy4*、*offline*、*queuewait*、*re-archive*、*reqrelease*、*size*、*stage_loaded*、および *stage_overflow* から選択されます。

  優先順位の詳細は、*archiver* および *archiver.cmd* のマニュアルページを参照してください。

C.1.5. ボリュームシリアル番号 (VSN) プールディレクティブ

archiver.cmd ファイルの VSN プールセクションにより、ボリュームシリアル番号 (VSN) 関連付けディレクティブで 1 つの単位として指定できるアーカイブメディアボリュームの名前付きの集合が定義されます。

このセクションは *vsnpools* ディレクティブで始まり、*endvsnpools* ディレクティブまたは *archiver.cmd* ファイルの最後で終わります。VSN プール定義の構文は次のとおりです。

```
vsn-pool-name media-type volume-specification
```

ここでは:

- *vsn-pool-name* は、プールに割り当てる名前です。
- *media-type* は、付録A「*装置タイプの用語集*」および *mcf* のマニュアルページで一覧表示されている 2 文字の Oracle HSM メディアタイプ識別子です。
- *volume-specification* は、ボリュームシリアル番号に一致する 1 つ以上の正規表現のスペース区切りリストです。正規表現構文の詳細は、Solaris *regcmp* のマニュアルページを参照してください。

この例では、4 つの VSN プール (*users_pool*、*data_pool*、*proj_pool*、および *scratch_pool*) を定義します。スクラッチプールは、VSN 関連付け内の一部のボリューム

を使い切ったとき、または別の VSN プールが空の状態になったときに使用されるボリュームセットです。指定した 3 つのプールのいずれかがボリューム不足になった場合、アーカイバはスクラッチプール VSN を選択します。

```
vsnpools
users_pool li ^VOL2[0-9][0-9]
data_pool li ^VOL3.*
scratch_pool li ^VOL4[0-9][0-9]
proj_pool li ^VOL[56].*
endvsnpools
```

C.1.6. ボリュームシリアル番号 (VSN) 関連付けディレクティブ

archiver.cmd ファイルの VSN 関連付けセクションでは、アーカイブセットにアーカイブメディアボリュームを割り当てます。このセクションは *vsns* ディレクティブで始まり、*endvsns* ディレクティブで終わります。

ボリューム割り当てディレクティブの形式は、次のとおりです。

```
archive-set-name.copy-number [media-type volume-specification] [-pool vsn-pool-name]
```

ここでは:

- *archive-set-name* は、アーカイブセット割り当てディレクティブが指定のボリュームに関連付けるアーカイブセットに割り当てた名前です。
- *copy-number* は、アーカイブセット割り当てディレクティブが指定のボリュームに関連付けるコピーに割り当てた番号です。これは、[*1-4*] の範囲の整数です。
- *media-type* は、付録A「*装置タイプの用語集*」および *mcf* のマニュアルページで一覧表示されている 2 文字の Oracle HSM メディアタイプ識別子です。
- *volume-specification* は、ボリュームシリアル番号に一致する 1 つ以上の正規表現のスペース区切りリストです。正規表現構文の詳細は、Solaris *regcmp* のマニュアルページを参照してください。
- *-pool vsn-pool-name* は、事前に指定された、単位として指定できるアーカイブメディアボリュームの名前付きコレクションです。ボリュームシリアル番号 (VSN) プールディレクティブを参照してください。

この例では、メディアを 2 行の VSN 指定に関連付けるさまざまな方法を示します。

```
vsns
archiveset.1 lt VSN001 VSN002 VSN003 VSN004 VSN005
archiveset.2 lt VSN0[6-9] VSN10
archiveset.3 -pool data_pool
endvsns
```

C.2. ステージングディレクティブ

書き込みは、ニアラインまたはオフラインの記憶装置からオンライン記憶装置に、ファイルデータをコピーして戻すことです。

ステージャーは、*samd* デーモンが実行されたときに起動します。ステージャーのデフォルトの動作は次のとおりです。

- ステージャーは、ライブラリ内のすべてのドライブを使用しようとする。
- 書き込みバッファーサイズはメディアタイプ別に決定され、書き込みバッファーはロックされない。
- ログファイルへの書き込みは行われない。
- 一度にアクティブであることが可能な書き込み要求は、最大 1000 個。

/etc/opt/SUNWsamfs/stager.cmd ファイルにディレクティブを挿入すると、ステージャーの動作をサイトに合わせてカスタマイズできます。

ファイルが *-n* (*never stage*) オプションでアーカイブされていなければ、アプリケーションでオフラインファイルが必要なときに、そのアーカイブコピーがディスクキャッシュにステージングされます。アプリケーションがファイルをすぐに利用できるようにするため、読み取り操作が書き込み操作のすぐあとを追跡するので、ファイル全体が書き込まれる前にアクセスを開始できます。

書き込みエラーとしては、メディアエラー、メディアを利用できない、自動ライブラリを利用できない、などがあります。ステージングエラーが返された場合、Oracle HSM ソフトウェアは次に使用可能なファイルのコピーを検索しようとします (そのようなコピーが存在し、アーカイブコピーのメディアを読み取るために使用できるデバイスがある場合)。

C.2.1. stager.cmd ファイル

stager.cmd ファイルには、デフォルト動作をオーバーライドするための指示を指定できます。ステージャーを構成して、ファイルをただちに書き込んだり、ファイルをまったく書き込まなかったり、部分的に書き込んだり、ほかの書き込みアクションを指定したりできます。たとえば、非書き込み属性を使用すると、ファイルをオンラインで書き込まずにアーカイブメディアから直接データにアクセスできるため、大きいファイルから小さいレコードにアクセスするアプリケーションに有益です。

このセクションでは、ステージャー指示について説明します。ステージャーディレクティブの詳細は、*stager.cmd* のマニュアルページを参照してください。Oracle HSM Manager software

を使用している場合は、「ファイルシステムの概要」ページまたは「ファイルシステムの詳細」ページからステージングを制御できます。ファイルシステムをブラウズし、個々のファイルのステータスを表示できます。また、フィルタを使用して特定のファイルを表示し、書き込むファイルを選択することができます。書き込み元のコピーを選択することも、システムにコピーを選択させることもできます。

この例は、指定可能なディレクティブをすべて設定したあとの *stager.cmd* ファイルを示しています。

```
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

C.2.1.1. drives: ステージングに使用するドライブ数の指定

デフォルトでは、ステージャーはファイルの書き込みを行うときに利用可能なすべてのドライブを使用します。ステージャーによってすべてのドライブが使用中の状態のままになると、アーカイバのアクティビティーに支障を来す恐れがあります。*drives* 指示は、ステージャーが利用できるドライブの数を指定します。このディレクティブの形式は、次のとおりです。

```
drives=library count
```

ここでは:

- *library* は、*mcf*ファイルに表示されるライブラリのファミリセット名です。
- *count* は、使用されるドライブの最大数です。デフォルトでは、このライブラリ用として *mcf* ファイルに構成されているドライブ数。

この例では、*dog* ファミリセットのライブラリにある 1 台のドライブだけをファイルのステージングに使用するように指定します。

```
drives = dog 1
```

C.2.1.2. bufsize: ステージングバッファーサイズの設定

デフォルトでは、書き込み対象ファイルは、アーカイブメディアからディスクキャッシュに復元される前に、メモリーバッファーに読み取られます。*bufsize* ディレクティブを使用して、バッファーサイズを指定したり、オプションでバッファーをロックしたりできます。これらの操作により、パフォーマンスを向上させることができます。さまざまな *buffer-size* 値を試すことができます。このディレクティブの形式は、次のとおりです。

```
bufsize= media-type buffer-size [lock]
```

ここでは:

- *media-type* は、付録A「*装置タイプの用語集*」および *mcf* のマニュアルページで一覧表示されている 2 文字の Oracle HSM メディアタイプ識別子です。
- *buffer-size* は、[*2-8192*] の範囲の整数です。この値は *defaults.conf* ファイルに指定された *media-type_blksize* 値で乗算されます。*buffer-size* の値が高ければ高いほど、多くのメモリーが使用されます。デフォルトは *16* です。
- *lock* は、各ステージング操作の期間内にロックバッファーを使用するように指示します。これにより、入出力要求ごとのステージングバッファーのロックまたはロック解除に関連するオーバーヘッドが回避され、パフォーマンスが改善されます。使用可能メモリーが制限されたシステム上で *lock* パラメータを指定すると、メモリー不足の状態になる可能性があります。デフォルトでは、ロックバッファーは指定されていないため、ファイルシステムがアーカイブバッファーの制御を保持します。

  lock 引数が有効であるのは、ステージング対象のファイルで直接入出力が有効になっている場合のみです。直接入出力の有効化についての詳細は、*setfa*、*sam_setfa*、および *mount_samfs* のマニュアルページを参照してください。

C.2.1.3. logfile: ステージングログファイルの指定

Oracle HSM ソフトウェアがファイルステージングイベント情報を収集し、それをログファイルに書き込むように要求できます。デフォルトでは、ログファイルへの書き込みは行われません。*logfile* ディレクティブは、ステージャーがログ情報を書き込むことができるログファイルを指定します。ステージャーは、書き込みを行なったファイルごとに 1 つまたは複数の行をログファイルに書き込みます。この 1 行には、ファイル名、書き込みを行なった日時、ボリュームシリアル番号 (VSN) などが含まれます。このディレクティブの形式は、次のとおりです。

```
logfile=filename [event-list]
```

ここで *filename* は、ログファイルのフルパス名、*event-list* は、ログが記録されるイベントタイプのスペース区切りリストです。

- *all* は、すべてのステージングイベントのログを記録します。
- *start* は、ファイルのステージングが開始されたときにログを記録します。
- *finish* (デフォルト) は、ファイルのステージングが終了したときにログを記録します。
- *cancel* (デフォルト) は、オペレータがステージングを取り消したときにログを記録します。
- *error* (デフォルト) は、ステージングエラーのログを記録します。

次のディレクティブは、*/var/adm/* ディレクトリにステージングログを作成します。

```
logfile=/var/adm/stage.log
```

ステージャーログエントリの形式は、次のとおりです。

```
status date time media-
type volume position.offset inode filesize filename copy user group requestor equipment-
number validation
```

ここでは:

- *status* は、開始を表す *S*、取り消しを表す *C*、エラーを表す *E*、完了を表す *F* です。
- *date* は、*yyyy/mm/dd* 形式の日付です。ここで *yyyy* は年を表す 4 桁の数字、*mm* は月を表す 2 桁の数字、*dd* は日を表す 2 桁の数字です。
- *time* は、*hh:mm:ss* 形式の時間です。ここで *hh*、*mm*、および *ss* は、それぞれ時間、分、秒を表す 2 桁の数字です。
- *media-type* は、付録A「*装置タイプの用語集*」および *mcf* のマニュアルページで一覧表示されている 2 文字の Oracle HSM メディアタイプ識別子です。
- *volume* は、ステージング対象のファイルを保持するメディアのボリュームシリアル番号 (VSN) です。
- *position.offset* は、ボリューム上のアーカイブ (*tar*) ファイルの先頭位置を表す 16 進数と、アーカイブファイルの先頭から相対的なステージング済みファイルのオフセットを表す 16 進数をドットで区切ったペアです。
- *inode* は、ステージング済みファイルの i ノード番号と生成番号をドットで区切ったものです。
- *filesize* は、ステージング済みファイルのサイズです。
- *filename* は、ステージング済みファイルの名前です。
- *copy* は、ステージング済みファイルを含むコピーのアーカイブコピー番号です。
- *user* は、ファイルを所有するユーザーです。
- *group* は、ファイルを所有するグループです。
- *requestor* は、ファイルを要求したグループです。
- *equipment-number* は、*mcf* ファイルに定義された、ファイルのステージング元であるドライブの装置番号です。
- *validation* は、ステージング済みファイルが検証済み (*V*) または未検証 (*-*) のどちらであるかを示します。

次の例に、一般的なステージャーログの一部を示します。

```
S 2014/02/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1 root other root 0 -
F 2014/02/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1 root other root 0 -
S 2014/02/16 14:06:27 dk disk02 4.a68 1218.1387 519464 /sam1/testdir1/fileaq 1 root other root 0 -
S 2014/02/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1 root other root 0 -
F 2014/02/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1 root other root 0 -
```

C.2.1.4. maxactive: ステージング要求数の指定

maxactive ディレクティブでは、一度にアクティブにできる書き込みリクエストの数を指定できます。このディレクティブの形式は、次のとおりです。

```
maxactive=number
```

ここで *number* は、[*1-500000*] の範囲の整数です。デフォルトは *4000* です。

この例は、キューに同時に存在できるステージング要求が 500 個までであることを指定しています。

```
maxactive=500
```

C.2.1.5. copysel: ステージング時のコピー選択順序の指定

ステージングディレクティブ *copysel* は、ファイルシステムごとにステージャーのコピー選択順序を設定します。

```
copysel=selection-order
```

selection-order は、最初から最後の順でのコピー番号のコロン区切りリストです。デフォルトの選択順序は *1:2:3:4* です。

詳細は、*stager.cmd* のマニュアルページを参照してください。この例は、ファイルシステム *samfs1* および *samfs2* のデフォルト以外のコピー選択順序を設定する *stager.cmd* ファイルを示しています。

```
logfile = /var/opt/SUNWsamfs/log/stager
drives = hp30 1
fs = samfs1
copysel = 4:3:2:1
fs = samfs2
copysel = 3:1:4:2
```

C.3. プレビュー要求ディレクティブ

Oracle HSM プロセスで、現在ドライブにロードされていないリムーバブルメディアボリュームが要求されると、その要求がプレビューキューに追加されます。キューに入れられた要求は、デフォルトでは先入れ先出し (FIFO) 順で処理されます。ただし、ファイル */etc/opt/SUNWsamfs/preview.cmd* を編集することにより、デフォルトの動作をオーバーライドできま

す。Oracle HSM ライブラリ制御デーモン (*sam-amld*) は停止するまで、これらのディレクティブを使用の開始時に読み取ります。キューの優先順位を動的に変更することはできません。

ディレクティブには次の 2 つのタイプがあります。

- グローバルディレクティブはファイルの先頭に置かれ、すべてのファイルシステムに適用されます。
- ファイルシステムディレクティブは、*fs=directive* 形式を取り、個々のファイルシステムに固有です

次のセクションでは、プレビューキューを制御するように *preview.cmd* ファイルを編集する方法について説明します。

- グローバルディレクティブ
- グローバルディレクティブまたはファイルシステム固有のディレクティブ
- **preview.cmd** ファイルのサンプル

C.3.1. グローバルディレクティブ

純粋なグローバルディレクティブは、次のとおりです。

- **vsn_priority**: ボリューム優先順位の調整
- **age_priority**: キュー内で待機する時間に応じた優先順位の調整

C.3.1.1. vsn_priority: ボリューム優先順位の調整

vsn_priority ディレクティブは、優先順位の高いボリュームとしてフラグが付けられたボリューム (VSN) の優先順位を、指定された値だけ高くします。ディレクティブの形式は、次のとおりです。

```
vsn_priority=value
```

ここで *value* は実数です。デフォルトは *1000.0* です。

次のコマンドを使用して、ボリュームに高い優先順位のフラグを設定します。

```
chmed +p media-type.volume-serial-number
```

ここで *media-type* は、付録A「*装置タイプの用語集*」および *mcf* のマニュアルページに一覧表示されている 2 文字の Oracle HSM メディアタイプのいずれかです。*volume-serial-number* は、ライブラリ内にある優先順位の高いボリュームを一意に識別する英数字の文字列です。詳細な情報については、*chmed* のマニュアルページを参照してください。

C.3.1.2. age_priority: キュー内で待機する時間に応じた優先順位の調整

age_priority ディレクティブは、要求がキュー内に存在する時間に対して指定された相対的な優先順位を変更します。これにより、たとえば、優先順位が高く新しい要求より古い要求が無制限に優先されるようなことを回避できます。このディレクティブは、キュー内で待機する時間の相対的な重み付けを変更する乗数を指定します。形式は次のとおりです。

```
age_priority=weighting-factor
```

ここで *weighting-factor* は、*1.0* よりも大きい、よりも小さい、または等しい実数です。ここでは:

- *1.0* よりも大きい値を指定すると、集計優先順位を計算するときに、キュー内で待機する時間に指定された重み付けが大きくなります。
- *1.0* よりも小さい値を指定すると、合計優先順位を計算するときに、キュー内で待機する時間に指定された重み付けが小さくなります。
- *1.0* と等しい値を指定すると、キュー内で待機する時間に指定された相対的な重み付けは変更されません。

デフォルトは *1.0* です。

C.3.2. グローバルディレクティブまたはファイルシステム固有のディレクティブ

次のディレクティブは、グローバルに、またはファイルシステムごとに適用できます。

- **hwm_priority**: ディスクキャッシュがほぼ満杯になったときの優先順位の調整
- **hwm_priority**: ディスクキャッシュがほぼ空になったときの優先順位の調整
- **lhwm_priority**: ディスクキャッシュが満杯になったときの優先順位の調整
- **hlwm_priority**: ディスクキャッシュが空になったときの優先順位の調整

C.3.2.1. hwm_priority: ディスクキャッシュがほぼ満杯になったときの優先順位の調整

hwm_priority ディレクティブは、ファイルシステムの利用率が高位境界値 (*hwm*) を上回ったときに、アーカイブ要求とステージング要求に指定された相対的な重み付けを調整します。高位境界値は、リリーサプロセスが起動され、アーカイブメディア上のコピーを含むファイルで占有されたディスク領域の再利用が開始されるポイントです。このような状況で、アーカ

イブ処理用に指定された相対的な重み付けを大きくすると、解放プロセスはステージングされたアーカイブコピーおよび新しいファイル用に追加容量を解放できます。ディレクティブの形式は、次のとおりです。

```
hwm_priority=weighting-factor
```

ここで *weighting-factor* は実数です。デフォルトは *0.0* です。

C.3.2.2. hwm_priority: ディスクキャッシュがほぼ空になったときの優先順位の調整

lwm_priority ディレクティブは、ファイルシステムの利用率が低位境界値 (*lwm*) を下回ったときに、アーカイブ要求とステージング要求に指定された相対的な重み付けを調整します。低位境界値は、リリーサプロセスが停止するポイントです。このような状況で、アーカイブ処理用に指定された相対的な重み付けを小さくして、ステージング要求の優先順位を高くすると、ディスクキャッシュ内に配置されるファイルが増え、メディアマウントの要求が減り、ファイルシステムのパフォーマンスが向上します。ディレクティブの形式は、次のとおりです。

```
lwm_priority=weighting-factor
```

ここで *weighting-factor* は実数です。デフォルトは *0.0* です。

C.3.2.3. lhwm_priority: ディスクキャッシュが満杯になったときの優先順位の調整

hlwm_priority ディレクティブは、ディスクキャッシュが満杯に近づき、キャッシュの利用率が低位境界値と高位境界値 (*lwm* と *hwm*) の間であるときに、アーカイブ要求とステージング要求に指定された相対的な重み付けを調整します。このような状況で、アーカイブ処理用に指定された相対的な重み付けを大きくすると、解放プロセスはステージングされたアーカイブコピーおよび新しいファイル用に追加容量を解放できます。ディレクティブの形式は、次のとおりです。

```
lhwm_priority=weighting-factor
```

ここで *weighting-factor* は実数です。デフォルトは *0.0* です。

C.3.2.4. hlwm_priority: ディスクキャッシュが空になったときの優先順位の調整

hlwm_priority ディレクティブは、ディスクキャッシュが空に近づき、キャッシュの利用率が高位境界値と低位境界値 (*hwm* と *lwm*) の間であるときに、アーカイブ要求とステージング

要求に指定された相対的な重み付けを調整します。このような状況で、アーカイブ処理用に指定された相対的な重み付けを小さくして、ステージング要求の優先順位を高くすると、ディスクキャッシュ内に配置されるファイルが増え、メディアマウントの要求が減り、ファイルシステムのパフォーマンスが向上します。ディレクティブの形式は、次のとおりです。

```
hlwm_priority=weighting-factor
```

ここで *weighting-factor* は実数です。デフォルトは *0.0* です。

C.3.3. preview.cmd ファイルのサンプル

指定されたメディアマウント要求の集計優先順位は、次の式に従って、すべての重み付け係数で設定された値を使用することで決定されます。

```
priority = vsn_priority + wm_priority + (age_priority * time-waiting-in-queue)
```

ここで *wm_priority* は、現在有効な境界値優先順位 (*hwm_priority*、*lwm_priority*、*hlwm_priority*、または *lhwm_priority*) であり、*time-waiting-in-queue* は、ボリューム要求がキューに入れられている秒数です。優先順位の計算の詳細な説明については、*preview.cmd* のマニュアルページの *PRIORITY CALCULATION* のセクションを参照してください。

データへのアクセスが非常に重要である場合やリムーバブルメディアドライブが不足している場合などの特殊な状況では、*preview.cmd* ファイルでディレクティブを使用すると、さらにファイルシステムアクティビティーを操作上の要件および使用可能なリソースに適合させることができます。格納されたデータの整合性は、*preview.cmd* ファイルの設定による影響を受けないため、アーカイブ要求とステージング要求の適切なバランスが見つかるまで、自由に試すことができます。

次の理由の一方または両方のために、デフォルトの優先順位計算の調整が必要になることがあります。

- ユーザーおよびアプリケーションがファイルにアクセスしたときに使用できるように、アーカイブ要求の前にステージング要求が処理されるようにする。
- ファイルシステムが満杯に近づいたときに、アーカイブ要求にもっとも高い優先順位が与えられるようにする。

次の *preview.cmd* ファイルのサンプルは、上記に示した状況に対処します。

```
# Use default weighting value for vsn_priority:
vsn_priority=1000.0
```

```
age_priority = 1.0
# Insure that staging requests are processed before archive requests:
lwm_priority = -200.0
lhwm_priority = -200.0
hlwm_priority = -200.0
# Insure that archive requests gain top priority when a file system is about to fill up:
hwm_priority = 500.0
```

lwm_priority、*lhwm_priority*、および *hlwm_priority* の重み付けを負の値にすると、ディスクキャッシュ内の領域が使用可能であるときは常に、ステージング要求の優先順位がアーカイブ要求よりも高くなります。これにより、要求があったときは常にデータアクセスが可能になります。キュー内で複数の要求が 100 秒間待機していて、ファイルシステムが低位境界値を下回っている場合は、次のようになります。

- 優先ボリュームに対するアーカイブマウント要求の集計優先順位は *1000+(-200)+(1x100)=900*
- 優先ボリュームに対するステージングマウント要求の集計優先順位は *1000+0+(1x100)=1100*
- 非優先ボリュームに対するステージングマウント要求の集計優先順位は *0+0+(1x100)=100*

ただし、ディスクキャッシュがほぼ空になったときは、アーカイブ要求を優先する必要があります。ファイルシステムが満杯になったときに、アーカイブされているファイルが非常に少ない場合は、アーカイブされたファイルのステージングや新しいファイルの取り込みに使用できる領域がありません。キュー内で複数の要求が 100 秒間待機していて、ファイルシステムが高位境界値を上回っている場合、次のようになります。

- 優先ボリュームに対するアーカイブマウント要求の集計優先順位は *1000+500+(1x100)=1600*
- 優先ボリュームに対するステージングマウント要求の集計優先順位は *1000+0+(1x100)=1100*
- 非優先ボリュームに対するステージングマウント要求の集計優先順位は *0+0+(1x100)=100*

付録D

例

/opt/SUNWsamfs/examples/ サブディレクトリには、さまざまな機能とさまざまな装置に対するソリューションを説明する、サンプルの Oracle HSM 構成ファイル、シェルスクリプト、および *dtrace* プログラムが含まれています。次のファイルが含まれています。

```
01_example.archiver.cmd.simple.txt
01_example.mcf.simple.txt
01_example.vfstab.txt
02_example.archiver.cmd.disk.tape.txt
02_example.diskvols.conf.NFS.txt
02_example.mcf.shared.txt
02_example.vfstab.disk.archive.NFS.txt
03_example.archiver.cmd.dk.9840.9940.txt
03_example.diskvols.conf
03_example.mcf.dk.9840.9940.txt
03_example.stk.9840C_parms.txt
03_example.stk.9940B_parms.txt
03_example.vfstab.disk.archive.txt
04_example.archiver.cmd.9840.LTO.txt
04_example.mcf.ma.9840.LTO.txt
04_example.stk50c.txt
05_example.archiver.cmd.9840.9940.T10k.txt
05_example.mcf.veritas.9840.9940.T10K.txt
05_example.stk_params9840.txt
05_example.stk_params9940.txt
05_example.stk_paramsT10K.txt
05_example.vstab.txt
06_example.archiver.cmd.samremote.client.txt
06_example.local.copy.samremote.client.stk50.txt
06_example.mcf.samremote.client.txt
06_example.mcf.samremote.server.txt
06_example.samremote.client.setup.stk100.txt
06_example.samremote.client.vfstab.txt
06_example.samremote.configuration.samremote.server.txt
07_example.mcf.distio.client.txt
07_example.mcf.distio.mds.txt
archiver.sh
defaults.conf
dev_down.sh
dtrace/fs_mon
dtrace/ino_mon
hosts.shsam1
hosts.shsam1.local.client
hosts.shsam1.local.server
inquiry.conf
load_notify.sh
log_rotate.sh
metadata_config_samfs.xml
nrecycler.sh
preview.cmd
recover.sh
recycler.sh
```

```
restore.sh
samdb.conf
samfs.cmd
samst.conf
save_core.sh
sendtrap
ssi.sh
st.conf_changes
stageback.sh
syslog.conf_changes
tarback.sh
verifyd.cmd
```

付録E

製品のアクセシビリティー機能

視力障がいや色覚異常など、視覚に障がいのある方は、コマンド行インタフェースをとおして Oracle Hierarchical Storage Manager and StorageTek QFS Software (Oracle HSM) にアクセスできます。このテキストベースのインタフェースにはスクリーンリーダーとの互換性があり、すべての機能はキーボードを使用して制御します。

用語集

この用語集では、Oracle HSM ソフトウェアおよびファイルシステムに固有の用語に焦点を当てています。業界標準の定義については、Storage Networking Industry Association が保守している辞書 (*http://www.snia.org/education/dictionary/*) を参照してください。

vfstab ファイル
vfstab ファイルには、ファイルシステムのマウントオプションが含まれます。コマンド行で指定されたマウントオプションは、*/etc/vfstab* ファイル内の指定をオーバーライドし、*/etc/vfstab* ファイル内で指定されたマウントオプションは *samfs.cmd* ファイル内の指定をオーバーライドします。

Oracle HSM
1. Oracle Hierarchical Storage Manager の一般的な略語。
2. アーカイブ処理のために構成され、Oracle HSM software によって管理される **QFS** ファイルシステムを説明する形容詞。

QFS
単独で使用することも、Oracle Hierarchical Storage Manager によって制御されるアーカイブファイルシステムとして使用することもできる、高性能で大容量の UNIX ファイルシステムである Oracle HSM QFS Software 製品。

アーカイバ
リムーバブルカートリッジへのファイルのコピーを自動制御するアーカイブプログラム。

アーカイブストレージ
アーカイブメディアに作成されるデータストレージ領域。

アーカイブセット
アーカイブセットは、アーカイブされるファイルのグループを識別し、ファイルは、サイズ、所有権、グループ、またはディレクトリの場所に関する共通の条件を共有します。アーカイブセットは、任意のファイルシステムグループ間で定義できます。

アーカイブメディア
アーカイブファイルの書き込み先であるメディア。アーカイブメディアには、リムーバブルなテープカートリッジまたは光磁気カートリッジと、アーカイブ処理用に構成されたディスクファイルシステムの両方が含まれます。

アドレッサブルストレージ
Oracle HSM のファイルシステムを通じてユーザーが参照する、オンライン、ニアライン、オフサイト、およびオフラインストレージを包含するストレージ領域。

イーサネット
パケット交換ローカルエリア網のテクノロジ。

オフサイトストレージ
サーバーから離れた遠隔地にあって災害時の障害回復に使用されるストレージ。

オフラインストレージ
ロード時にオペレータの介入を必要とするストレージ。

オンラインストレージ
いつでも利用可能なストレージ (ディスクキャッシュストレージなど)。

カートリッジ

データストレージメディア (磁気テープ、光学メディアなど) の容器。ボリューム、テープ、メディアと呼ぶこともあります。ボリュームシリアル番号 (VSN)を参照してください。

カーネル

基本的なオペレーティングシステム機能を提供するプログラム。UNIX カーネルは、プロセスの作成と管理を行い、ファイルシステムにアクセスする機能を提供し、一般的なセキュリティーを提供し、通信機能を用意します。

回復ポイント

Oracle HSM ファイルシステムのメタデータについてポイントインタイムのバックアップコピーを格納する圧縮ファイル。

ユーザーファイルを不意に削除してしまった場合からファイルシステム全体が壊滅的に失われた場合に至るまで、データ損失時に管理者は、ファイルまたはファイルシステムが完全なままの時点の最新の回復ポイントを見つけるとほぼすぐに、ファイルまたはファイルシステムを最新の既知の良好な状態に回復できます。次に、管理者はその時点で記録されたメタデータを復元します。そして、メタデータに示されているファイルを管理者がアーカイブメディアからディスクキャッシュに書き込むか、または可能であれば、ファイルシステムがユーザーおよびアプリケーションがファイルにアクセスするときに必要に応じてファイルを書き込むようにします。

外部配列

ファイルに割り当てられた各データブロックのディスク上の位置を定義する、ファイルの i ノード内の配列。

解放優先順位

ファイルシステム内のファイルがアーカイブ後に解放される優先順位。解放優先順位は、ファイル属性のさまざまなウェイトを掛け合わせてから、その結果を合計することで計算されます。

カタログ

自動ライブラリにあるリムーバブルメディアボリュームのレコード。1 つの自動ライブラリにつき 1 つのカタログがあり、1 つのサイトの自動ライブラリすべてにつき 1 つの履歴があります。ボリュームは、ボリュームシリアル番号 (VSN)を使用して識別および追跡されます。

監査 (完全)

カートリッジをロードしてカートリッジの VSN を検証する処理。光磁気カートリッジの容量と領域に関する情報が確認され、自動ライブラリのカタログに入力されます。ボリュームシリアル番号 (VSN)を参照してください。

間接ブロック

ストレージブロックのリストが入っているディスクブロック。ファイルシステムには、最大 3 レベルの間接ブロックがあります。第 1 レベルの間接ブロックには、データストレージに使用されるブロックのリストが入っています。第 2 レベルの間接ブロックには、第 1 レベルの間接ブロックのリストが入っています。第 3 レベルの間接ブロックには、第 2 レベルの間接ブロックのリストが入っています。

管理セット ID

共通の特性を共有するユーザーやグループについて、ストレージ管理者が定義したセット。通常、管理セットは、複数のグループからのユーザーが関与

し、複数のファイルおよびディレクトリにまたがっているようなプロジェクトのストレージを管理するために作成されます。

疑似デバイス

関連付けられているハードウェアがないソフトウェアのサブシステムまたはドライバ。

共有ホストファイル

共有ファイルシステムを作成する場合、システムはホストファイルからの情報をメタデータサーバー上の共有ホストファイルへコピーします。この情報は、**samsharefs -u** コマンドを発行するときに更新します

クライアントサーバー

あるサイトのプログラムが、別のサイトのプログラムに要求を送って応答を待つ、分散システムにおける対話モデル。要求側のプログラムをクライアントと呼びます。応答を行うプログラムをサーバーと呼びます。

グローバルディレクティブ

すべてのファイルシステムに適用され、最初の **fs=** 行の前に位置する、アーカイバーディレクティブとリリーサディレクティブ。

結合ステージング

グループのいずれかのメンバーに書き込まれるときに、関連ファイルのグループが書き込まれること。ファイルが同じディレクトリにあり、一緒に使用されることがよくある場合、ファイル所有者は Oracle HSM 結合書き込みファイル属性を設定することで、これらを関連付けることができます。その後、グループ内のいずれかのファイルがアプリケーションからアクセスされるときに、グループ内にオフラインのファイルがある場合、Oracle HSM は、グループ全体をアーカイブメディアからディスクキャッシュに書き込みます。これにより、すべての必要なファイルが同時に再度使用可能になります。

高位境界値

1. アーカイブファイルシステムにおいて、Oracle HSM ファイルシステムでリリーサプロセスを開始して、以前にアーカイブされたファイルをディスクから削除するときのディスクキャッシュ利用率 (パーセント)。高位境界値が適切に構成されることで、ファイルシステムには新しいファイルや新しく書き込まれるファイル用に使用可能な領域が常に十分あります。詳細については、*sam-releaser* および *mount_samfs* のマニュアルページを参照してください。低位境界値と比較してください。
2. アーカイブファイルシステムの一部であるリムーバブルメディアライブラリにおいて、リサイクラプロセスを開始するときのメディアキャッシュ使用率 (パーセント)。リサイクルすると、現在のデータのフルボリュームの一部が空になるため、新しいメディアと交換したりラベルを付け替えたりすることができます。

シーキング

ランダムアクセス入出力操作中に、ディスクデバイスの読み取り/書き込みヘッドを、ディスク上のある場所から別の場所に移動すること。

事前割り当て

ファイルの書き込みのために、ディスクキャッシュ上の連続した領域を予約するプロセス。事前割り当ては、サイズがゼロのファイルにのみ指定できます。詳細は、*setfa* のマニュアルページを参照してください。

自動ライブラリ	オペレータが処置を必要としない、リムーバブルメディアカートリッジを自動的にロードしたりロード解除したりするように設計された、ロボット制御の装置。自動ライブラリには、1 つまたは複数のドライブと、ストレージスロットとドライブの間でカートリッジを移動するトランスポートメカニズムとが含まれています。
スーパーブロック	ファイルシステムの基本パラメータを定義する、ファイルシステム内のデータ構造。スーパーブロックは、ストレージファミリセット内のすべてのパーティションに書き込まれ、セットにおけるパーティションのメンバーシップを識別します。
ステージング	ニアラインまたはオンラインファイルをアーカイブストレージからオンラインストレージにコピーして戻すプロセス。
ストライプ化	複数のファイルをインタレース方式で論理ディスクに同時に書き込むデータアクセス方法。Oracle HSM ファイルシステムには、ストライプグループを使用する「強いストライプ化」と、*stripe=x* マウントパラメータを使用する「弱いストライプ化」の 2 種類のストライプ化があります。強いストライプ化はファイルシステムの設定時に使用可能にし、*mcf* ファイルにストライプ化グループを定義する必要があります。弱いストライプ化は *stripe=x* マウントパラメータで使用可能にし、ファイルシステムごと、またはファイルごとに変更できます。*stripe=0* を設定すると、無効にできます。強いストライプ化と弱いストライプ化はどちらも、要素数が同じ複数のストライプ化グループでファイルシステムが構成されている場合に使用できます。ラウンドロビンも参照してください。
ストライプ化グループ	*mcf* ファイルで 1 つまたは複数の *gXXX* デバイスとして定義された、ファイルシステム内のデバイスのコレクション。複数のストライプ化グループは 1 つの論理デバイスとして扱われ、常にディスク割り当て単位 (DAU) と等しいサイズでストライプ化されます。
ストライプサイズ	割り当てられたディスク割り当て単位 (DAU) の数。書き込みがこの数に達すると、ストライプの次のデバイスへ移動します。*stripe=0* マウントオプションを使用した場合、ファイルシステムはストライプ化アクセスではなくラウンドロビン式アクセスを使用します。
ストレージスロット	カートリッジがドライブで使用されていないときに保管される自動ライブラリ内の場所。
ストレージファミリセット	単一の論理デバイスとして集合的に表されるディスクのセット。
正規表現	ほかの文字列 (ファイル名、構成ファイルなど) 検索、選択、および編集用に設計された標準化パターンマッチング言語による文字列。Oracle HSM ファイルシステム操作で使用される正規表現構文の詳細については、Oracle HSM Solaris *regex* および *regcmp* のマニュアルページを参照してください。

接続

信頼性の高いストリーム配信サービスを提供する、2 つのプロトコルモジュール間のパス。TCP 接続は、1 台のマシン上の TCP モジュールと他方上の TCP モジュールをつなぎます。

タイマー

ユーザーが弱い制限値に達してから、このユーザーに強い制限値が課されるまでに経過する時間を追跡する割り当てソフトウェア。

直接アクセス

ニアラインファイルをアーカイブメディアから直接アクセスすることができるのでディスクキャッシュに取り出す必要がないことを指定する、ファイル属性 (stage never)。

直接接続ライブラリ

SCSI インタフェースを使用してサーバーに直接接続された自動ライブラリ。SCSI 接続のライブラリは、Oracle HSM ソフトウェアによって直接制御されます。

直接入出力

大型ブロック整合逐次入出力に使用される属性の 1 つ。*setfa* コマンドの *-D* オプションは、直接入出力のオプションです。このオプションは、ファイルやディレクトリの直接入出力の属性を設定します。ディレクトリに対して設定した直接入出力の属性は、継承されます。

強い制限値

割り当てにおいて、指定されたユーザー、グループ、管理セット **ID** などが消費可能なストレージリソースの最大の絶対量。弱い制限値を参照してください。

低位境界値

アーカイブファイルシステムにおいて、Oracle HSM ファイルシステムでリリーサプロセスを停止して、以前にアーカイブされたファイルをディスクから削除することを停止するときのディスクキャッシュ利用率 (パーセント)。低位境界値が適切に構成されることで、ファイルシステムでは最高のパフォーマンスを得られるようにできるだけ多くのファイルがキャッシュに保持される一方、新しいファイルや新しくステージングされるファイル用に使用可能な領域を確保します。詳細については、*sam-releaser* および *mount_samfs* のマニュアルページを参照してください。高位境界値と比較してください。

ディスクキャッシュ

オンラインディスクキャッシュとアーカイブメディアとの間でデータファイルの作成と管理に使用する、ファイルシステムソフトウェアのディスクに格納されている部分。個々のディスクパーティションまたはディスク全体で、ディスクキャッシュとして使用できます。

ディスクのストライプ化

アクセスパフォーマンスの向上と全体的なストレージ憶領域の容量の増大を図るため、1 つのファイルを複数のディスクに記録すること。ストライプ化も参照してください。

ディスクバッファー

SAM-Remote 構成において、クライアントからサーバーにデータをアーカイブするときに使用するサーバーシステム上のバッファー。

ディスク領域しきい値

管理者が定義した、ディスクキャッシュ利用率の最大レベルと最小レベル。リリーサは、これらの事前定義ディスク容量しきい値に基づいて、ディスクキャッシュ利用率を制御します。

ディスク割り当て単位 (DAU)

Oracle HSM ファイルシステムにおいて、書き込まれるデータ量とは関係なく各入出力操作で消費される連続領域の最小量。つまり、ディスク割り当て単位によって、指定サイズのファイルを転送するときに必要な入出力操作の最小回数が決まります。これはディスクデバイスのブロックサイズの倍数にする必要があります。

ディスク割り当て単位は、選択された Oracle HSM デバイスタイプおよびユーザー要件によって異なります。*md* デバイスタイプでは、デュアル割り当て単位が使用されます。DAU は、ファイルへの最初の 8 回の書き込みでは *4*K バイト、後続の書き込みではユーザー指定の *16*K、*32*K、または *64*K バイトになるため、小さいファイルは相応の小さいブロックで書き込まれ、大きいファイルは大きいブロックで書き込まれます。*mr* およびストライプ化グループのデバイスタイプでは、*[8-65528]*K バイトの範囲内で *8* の単位で調整可能な DAU が使用されます。そのため、ファイルは大きな均一ブロックで書き込まれることになり、大きな均一サイズのファイルのサイズにきわめて近くなります。

ディレクトリ

ファイルシステム内のそのほかのファイルとディレクトリを指す、ファイルデータ構造。

データデバイス

ファイルシステムで、ファイルデータが格納されるデバイスまたはデバイスグループ。

デバイススキャナ

手動でマウントされたリムーバブルデバイスの有無を定期的にモニター監視し、ユーザーやほかのプロセスによって要求されることのある、マウント済みのカートリッジの存在を検出するソフトウェア。

デバイスロギング

Oracle HSM ファイルシステムをサポートするハードウェアデバイスの特定のエラー情報を提供する、構成可能な機能。

ドライブ

リムーバブルメディアボリューム間でデータを転送するためのメカニズム。

トランスポート

ロボットを参照してください。

名前空間

ファイルおよびその属性と格納場所を示す、ファイル群のメタデータ部分。

ニアラインストレージ

アクセスする前に無人マウントが必要なリムーバブルメディアストレージ。通常、ニアラインストレージはオンラインストレージよりも安価ですが、アクセスに多少時間がかかります。

ネットワーク接続された自動ライブラリ

ベンダー提供のソフトウェアパッケージによって制御される、StorageTek、ADIC/Grau、IBM、Sony などの製品であるライブラリ。QFS のファイルシステムは、自動ライブラリ用に設計された Oracle HSM メディアチェンジャーデーモンを使用して、ベンダーソフトウェアと接続します。

パーティション

デバイスの一部または光磁気カートリッジの片面。

バックアップ	不注意によるファイルの消去を防ぐことを目的とした、ファイル群のスナップショット。バックアップには、ファイルの属性と関連データの両方が含まれます。
ヒストリアン	Oracle HSM ヒストリアンは、*/etc/opt/SUNWsamfs/mcf* ファイルで定義されている自動メディアライブラリからエクスポートされたボリュームのカタログです。デフォルトでは、Oracle HSM ファイルシステムホストの */var/opt/SUNWsamfs/catalog/historian* にあります。詳細については、Oracle HSM *historian* のマニュアルページを参照してください。
ファイバチャネル	デバイス間の高速シリアル通信を規定する ANSI 標準。ファイバチャネルは、SCSI-3 におけるバスアーキテクチャーの 1 つとして使用されます。
ファイルシステム	階層構造によるファイルとディレクトリの集まり。
ファイルシステム固有ディレクティブ	*archiver.cmd* ファイル内のグローバルディレクティブのあとのアーカイバディレクティブとリリーサディレクティブは特定のファイルシステム専用であり、*fs =* から始まります。ファイルシステム固有ディレクティブは、次の **fs =** ディレクティブ行まで、またはファイルの終わりに到達するまで有効です。1 つのファイルシステムを対象としたディレクティブが複数存在する場合、ファイルシステム固有ディレクティブがグローバルディレクティブをオーバーライドします。
ファミリセット	ディスクの集合や、自動ライブラリ内のドライブなど、独立した物理デバイスの論理グルーピング。ストレージファミリセットも参照してください。
ファミリデバイスセット	ファミリセットを参照してください。
複数読み取りファイルシステム	複数のホストにマウント可能なファイルシステムを指定する、シングルライター、マルチリーダー機能。複数のホストがこのファイルシステムを読み込むことができますが、ファイルシステムへの書き込みを行えるのは 1 つのホストだけです。複数のリーダーは、*mount* コマンドの *-o reader* オプションによって指定します。シングルライターホストは、*mount* コマンドの *-o writer* オプションによって指定します。詳細については、*mount_samfs* のマニュアルページを参照してください。
ブロックサイズ	ブロックデバイス (ハードディスク、磁気テープカートリッジなど) 上の最小のアドレッサブルデータ単位のサイズ。ディスクデバイスでは、これはセクターサイズ (通常 512 バイト) と同等です。
ブロック割り当てマップ	ディスク上のストレージの利用可能な各ブロック。また、これらのブロックが使用中か空いているかを示す、ビットマップです。
ホストファイル	共有ファイルシステム内のすべてのホストの一覧からなるファイル。ファイルシステムを Oracle HSM 共有ファイルシステムとして初期化している場合、ファイルシステムが作成される前にホストファイル */etc/opt/SUNWsamfs/hosts.fs-name* を作成する必要があります。*sammkfs* コマンドは、ファイルシステムを作成するときにホストファイルを使用します。*samsharefs* コマンドを使用すると、あとでホストファイルの内容を置換または更新できます。

ボリューム

1. ストレージメディア上のアクセス可能な単一の論理ストレージ領域で、通常はボリュームシリアル番号 (VSN) やボリュームラベルによって操作されます。ストレージディスクおよび磁気テープカートリッジは、1 つまたは複数のボリュームを保持できます。使用する場合、ボリュームはファイルシステムの指定されたにマウントマウントポイントされます。
2. 単一の論理ボリュームを保持する磁気テープカートリッジ。
3. ランダムアクセスディスクデバイスのファイルシステム、ディレクトリ、またはファイルのことで、順次アクセスのリムーバブルメディアカートリッジ (テープなど) であるかのように構成および使用されます。

ボリュームオーバーフロー

1 つのファイルを複数のボリュームにまたがらせる機能。ボリュームオーバーフローは、個々のカートリッジの容量を超える、非常に大きなファイルを使用するサイトで、便利に利用できます。

ボリュームシリアル番号 (VSN)

1. テープまたはディスクストレージボリュームに割り当てられたシリアル番号。ボリュームシリアル番号は、最大 6 文字の大文字英数字で構成され、先頭は文字にする必要があります。また、テープライブラリやパーティションといった特定のコンテキストで、ボリュームを一意に特定する必要があります。ボリュームシリアル番号は、ボリュームラベルに書き込まれます。
2. 広義の具体的なストレージボリューム (特にリムーバブルメディアカートリッジ)。

マウントポイント

ファイルシステムがマウントされているディレクトリ。

ミラー書き込み

別々のディスク集合上で 1 つのファイルのコピーを 2 つ保管することによって、どちらかのディスクが故障してもデータを消失しないようにしてください。

メタデータ

データに関するデータ。メタデータは、ディスク上のファイルの正確なデータ位置を確認するために使用される索引情報です。ファイル、ディレクトリ、アクセス制御リスト、シンボリックリンク、リムーバブルメディア、セグメントに分割されたファイル、およびセグメントに分割されたファイルのインデックスに関する情報から構成されます。

メタデータデバイス

ファイルシステムのメタデータを保存するデバイス (ソリッドステートディスクやミラーデバイスなど)。ファイルデータとメタデータを別のデバイスに格納すると、パフォーマンスが向上します。メタデータデバイスは、*mcf* ファイルにおいて、*ma* ファイルシステム内の *mm* デバイスとして宣言されます。

メディア

テープまたは光磁気ディスクカートリッジ。

メディアリサイクリング

アクティブファイルのあまりないアーカイブメディアをリサイクルまたは再利用するプロセス。

猶予期間	割り当てにおいて、ファイルシステムで特定のユーザー、グループ、管理セット ID などに属するファイルの合計サイズが割り当てで指定された弱い制限値を超えてもかまわない時間。
弱い制限値	割り当てにおいて、指定されたユーザー、グループ、管理セット ID などが無期限で書き込み可能なストレージ領域の最大量。ファイルは、強い制限値を上限として弱い制限値で許可された領域以上を使用できますが、これは割り当てで定義される短い猶予期間の間に限られます。強い制限値を参照してください。
ライブラリ	自動ライブラリを参照してください。
ライブラリカタログ	カタログを参照してください。
ラウンドロビン	個々のファイル全体を逐次的に論理ディスクに書き込むデータアクセス方法。1 つのファイルがディスクに書き込まれるとき、そのファイル全体が第 1 論理ディスクに書き込まれます。そして、2 つめのファイルはその次の論理ディスクに書き込まれる、というふうになります。各ファイルのサイズによって、入出力のサイズが決まります。ディスクのストライプ化およびストライプ化も参照してください。
リース	特定の期間中、ファイルを操作するアクセス権をクライアントホストに与える機能。メタデータサーバーは、各クライアントホストに対してリースを発行します。ファイル操作を続行するため、必要に応じてリースが更新されます。
リサイクラ	期限切れアーカイブのコピーが格納されている空間またはカートリッジを回収する、Oracle HSM のユーティリティー。
リムーバブルメディアファイル	磁気テープや光磁気ディスクカートリッジなど、常駐場所であるリムーバブルメディアカートリッジから直接アクセスできる、特殊なタイプのユーザーファイル。アーカイブファイルデータや書き込みファイルデータの書き込みにも使用します。
リモート手続き呼び出し	**RPC**を参照してください。
リリーサ	アーカイブされたファイルを識別し、そのディスクキャッシュコピーを開放することで、利用可能なディスクキャッシュ空間を増やす Oracle HSM コンポーネント。リリーサは、オンラインディスクストレージの容量を、上限値と下限値に合わせて自動的に調整します。
ローカルファイルシステム	Solaris Cluster システムの 1 つのノードにインストールされたファイルシステム。ほかのノードからは、あまり利用されません。サーバーにインストールされたファイルシステムのことも指します。
ロボット	ストレージのスロットとドライブとの間でカートリッジを移動する自動ライブラリコンポーネント。トランスポートとも呼ばれます。
割り当て	指定されたユーザー、グループ、または 管理セット ID が消費可能なストレージリソース量。強い制限値および弱い制限値を参照してください。

DAU	ディスク割り当て単位 **(DAU)**を参照してください。
FDDI	Fiber-Distributed Data Interface の略で、最大 200 km (124 マイル) まで延長可能な、ローカルエリアネットワークでのデータ転送規格。FDDI プロトコルは、トークンリングプロトコルが基礎になっています。
ftp	File Transfer Protocol の略で、2 つのホスト間でファイルを転送するためのネットワークプロトコル。よりセキュアな代替方法については、**sftp**を参照してください。
i ノード	索引ノード。ファイルシステムがファイルを記述するときに使用するデータ構造です。i ノードは、名前以外のファイル属性をすべて記述します。ファイル属性には所有権、アクセス、アクセス権、サイズ、およびディスクシステム上におけるファイルの場所などが含まれます。
i ノードファイル	ファイルシステムに常駐しているすべてのファイルの i ノード構造を含む、ファイルシステム上の特殊ファイル (*.inodes*)。i ノードは長さ 512 バイトです。i ノードファイルは、ファイルシステムのファイルデータから分離されたメタデータファイルです。
LAN	ローカルエリアネットワーク。
LUN	論理ユニット番号。
mcf	マスター構成ファイル。ファイルシステム環境内のデバイス (トポロジ) 間の関係を定義する、初期化時に読み取られるファイル。
NFS	Network File System の略で、異機種システム混在ネットワーク上で、リモートファイルシステムへの透過アクセスを提供する、ネットワークファイルシステム。
NIS	Network Information Service の略で、ネットワーク上のシステムとユーザーに関する重要な情報を含む、分散ネットワークデータベース。NIS データベースは、マスターサーバーとすべてのスレーブサーバーに保存されます。
qfsdump	**samfsdump (qfsdump)**を参照してください。
qfsrestore	**samfsrestore (qfsrestore)**を参照してください。
RAID	Redundant Array of Independent Disks。複数の独立したディスクを使用してファイル保存の信頼性を保証するディスク技術です。1 つのディスクが故障してもデータを紛失することはなく、耐障害のディスク環境を提供できます。ディスクを個別で使用した場合より、スループットを向上できます。
RPC	リモート手続き呼び出し。カスタムネットワークデータサーバーの実装時に NFS が基盤として使用するデータ交換メカニズムです。
SAM	Oracle Hierarchical Storage Manager 製品の以前の名前である Storage Archive Manager の一般的な略語。

SAM-QFS

1. Oracle Hierarchical Storage Manager 製品の古いバージョンの一般的な略語。
2. アーカイブ処理のために構成され、Oracle HSM software によって管理される **QFS** ファイルシステムを説明する形容詞。

SAM-Remote クライアント

クライアントデーモンにいくつかの擬似デバイスが含まれ、専用のライブラリデバイスも持つことがある Oracle HSM システム。クライアントは、SAM-Remote サーバーに依存して 1 つまたは複数のアーカイブのコピーに使用するアーカイブメディアを利用します。

SAM-Remote サーバー

全容量の Oracle HSM ストレージ管理サーバーと、SAM-Remote クライアントが共有するライブラリを定義する SAM-Remote サーバーデーモンの両方。

samfsdump (qfsdump)

制御構造ダンプを作成し、指定したファイル群に関する制御構造の情報をすべてコピーするプログラム。これは通常、ファイルデータのコピーは行いません。*-U* オプションが指定された場合、このコマンドはデータファイルのコピーも行います。Oracle Hierarchical Storage Manager パッケージがインストールされていない場合、このコマンドは *qfsdump* と呼ばれます。

samfsrestore (qfsrestore)

i ノードおよびディレクトリの情報を制御構造ダンプから復元するプログラム。**samfsdump (qfsdump)**も参照してください。

SAN

ストレージエリアネットワーク。

SCSI

Small Computer System Interface。ディスク、テープドライブ、自動ライブラリなどの周辺デバイスに一般的に使用される電気通信仕様。

sftp

Secure File Transfer Protocol の略で、**ftp** を基にした **ssh** のセキュアな実装。

Small Computer System Interface

SCSIを参照してください。

ssh

Secure Shell の略で、セキュアなリモートのコマンド行ログインおよびコマンド実行を可能にする暗号化ネットワークプロトコル。

Storage Archive Manager

Oracle Hierarchical Storage Manager 製品の以前の名前。

SUNW.qfs

Oracle HSM 共有ファイルシステムをサポートする Solaris Cluster リソースタイプ。*SUNW.qfs* リソースタイプは、共有ファイルシステムのメタデータサーバー (MDS) 用のフェイルオーバーリソースを定義します

tar

テープアーカイブ。イメージのアーカイブに使用される、標準のファイルおよびデータ記録フォーマット。

TCP/IP

Transmission Control Protocol/Internet Protocol。ホストツーホストのアドレッシングとルーティング、パケット配信 (IP)、および信頼性の高いアプリケーションポイント間データ配信 (TCP) を行うインターネットプロトコルです。

WORM	Write-Once-Read-Many。書き込みできるのは 1 回だけで、読み込みは何度でも行えるという、メディアの記録方式です。

索引

た

S