Sun Cluster 3.0 12/01 の概念

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto,, CA 94303-4900
U.S.A. 650-960-1300

Part Number 816-3337
2001 年 12 月, Revision A

原典: *Sun Cluster 3.0 12/01 Concepts*

Part No: 816–2027

Revision A

目次

はじめに

『*Sun™ Cluster 3.0 12/01* の概念』には、SunPlex™ システムに関する概念の説明と参照情報が記載されています。 SunPlex システムでは、Sun のクラスタソリューションを構成するすべてのハードウェア/ソフトウェアコンポーネントがサポートされます。

このマニュアルは、Sun Cluster ソフトウェアについて知識のある、経験豊富なシステム管理者を対象としています。販売活動のガイドとしては使用しないでください。このマニュアルを読む前に、システムの必要条件を確認し、適切な装置とソフトウェアを用意しておく必要があります。

このマニュアルで説明されている作業手順を行うには、Solaris™ オペレーティング環境に関する知識と、SunPlex システムと共に使用するボリューム管理ソフトウェアに関する専門知識が必要です。

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P–1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% **su** password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% **grep '^#define ** **XV_VERSION_STRING'**

ただし AnswerBook2 では、ユーザーが入力する文字と画面上のコンピュータ出力は区別して表示されません。

コード例は次のように表示されます。

- C シェルプロンプト

```
system% command y|n [filename]
```

- Bourne シェルおよび Korn シェルのプロンプト

```
system$ command y|n [filename]
```

- スーパーユーザーのプロンプト

```
system# command y|n [filename]
```

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

シェルプロンプト

シェル	プロンプト
C シェル	*machine_name*%
C シェルスーパーユーザー	*machine_name*#
Bourne シェルおよび Korn シェル	$
Bourne シェルおよび Korn シェルスーパーユーザー	#

関連マニュアル

説明内容	タイトル	パート番号
インストール	『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』	816-3343
ハードウェア	『*Sun Cluster 3.0 12/01 Hardware Guide*』	816-2023

説明内容	タイトル	パート番号
データサービス	『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』	816-3347
API 開発	『*Sun Cluster 3.0 12/01* データサービスの開発』	816-3341
管理	『*Sun Cluster 3.0 12/01* システム管理』	816-3349
エラーメッセージと問題解決	『*Sun Cluster 3.0 12/01 Error Messages Guide*』	816-2028
最新情報	『*Sun Cluster 3.0 12/01* ご使用にあたって』	816-3354

Sun のオンラインマニュアル

`http://docs.sun.com` では、Sun が提供しているオンラインマニュアルを参照することができます。マニュアルのタイトルや特定の主題などをキーワードとして、検索を行うこともできます。

問い合わせについて

SunPlex システム のインストールまたは使用で問題が発生した場合は、ご購入先に連絡し、次の情報をお伝えください。

- 名前と電子メールアドレス (利用している場合)
- 会社名、住所、および電話番号
- ご使用のシステムのモデルとシリアル番号
- オペレーティング環境のバージョン番号 (例: Solaris 8)
- Sun Cluster ソフトウェアのバージョン番号 (例: Sun Cluster 3.0)

システムの各ノードに関する情報を収集するには、次のコマンドを使用してください。

コマンド	機能
`prtconf -v`	システムメモリーのサイズと周辺デバイス情報を表示する
`psrinfo -v`	プロセッサの情報を表示する
`showrev --p`	インストールされているパッチを報告する
`prtdiag -v`	システム診断情報を表示する
`scinstall -pv`	Sun Cluster ソフトウェアのリリースおよびパッケージのバージョン情報を表示する

`/var/adm/messages` ファイルの内容もご購入先にお知らせください。

第 1 章

基本知識と概要

SunPlex システムはハードウェアと Sun Cluster ソフトウェアが統合されたソリューションであり、高度な可用性とスケーラビリティを備えたサービスを提供するために使用されます。

このマニュアルでは、SunPlex のマニュアルの読者に必要な概念について説明します。次の読者を対象としています。

- クラスタハードウェアを設置して保守を行う担当者
- Sun Cluster ソフトウェアをインストール、構成、管理するシステム管理者
- 現在 Sun Cluster 製品に含まれていないアプリケーション用のフェイルオーバーサービスやスケーラブルサービスを開発するアプリケーション開発者

このマニュアルは、SunPlex の他のマニュアルと合わせて、SunPlex システムの全体を説明するものです。

この章では、次のことを説明します。

- SunPlex の基本知識と概要
- SunPlex の各ユーザーごとの役割と参照する情報
- SunPlex で作業するにあたって理解する必要がある重要な概念
- 重要な概念に関連する手順と情報を記載した SunPlex のマニュアル
- クラスタに関連する作業と、これらの作業手順が記載されたマニュアル

SunPlex システムの基本知識

SunPlex システムは、Solaris オペレーティング環境をクラスタオペレーティングシステムに拡張するものです。クラスタまたは Plex とは、緩やかに結合された処理ノードの集合のことで、データベース、Web サービス、ファイルサービスなどのネットワークサービスやアプリケーションを、1 つのクライアントとして扱います。

各クラスタノードは、それ自身のプロセスを実行するスタンドアロンサーバーです。これらのプロセスは、相互にやりとりすることによって、ユーザーに提供するアプリケーション、システムリソース、データを (ネットワーククライアントにとって) 1 つのシステムのように形成します。

クラスタは、従来の単一サーバーシステムと比較した場合、いくつかの利点があります。これらの利点には、フェイルオーバーサービスとスケーラブルサービスのサポート、モジュールの成長に対応できる容量、従来のハードウェアフォルトトレラントシステムよりも低価格の製品といったものがあります。

次に、SunPlex の導入目的を示します。

- ソフトウェアまたはハードウェアの障害が原因のシステム停止時間を減らす、または完全になくす
- 単一サーバーシステムを停止させるような障害が発生しても、エンドユーザーへのデータとアプリケーションの可用性を保証する
- クラスタにノードを追加し、追加したプロセッサに応じたサービスを提供できるようにすることで、アプリケーションのスループットを向上させる
- クラスタ全体を停止しなくても保守を実行できるようにすることで、システムの可用性を強化する

高可用性とフォルトトレランスの比較

SunPlex システムは、高可用性 *(HA)* システムとして設計されています。つまり、データとアプリケーションに対し、ほぼ連続的なアクセスを可能にするシステムです。

これに対して、フォルトトレラントのハードウェアシステムは、データとアプリケーションに対する一定したアクセスを可能にしますが、特殊なハードウェアが必要なため、コストが高くなります。また、通常はソフトウェアの障害を考慮していません。

SunPlex システムは、ハードウェアとソフトウェアの組み合わせによって高可用性を実現しています。冗長なクラスタインターコネクト、記憶装置、パブリックネットワークは、単一の障害に対する防護策となります。クラスタソフトウェアは、メンバーノードの状態を常に監視し、障害が発生したノードがクラスタに属さないようにしてデータの破壊を防止します。また、クラスタは、サービスとそれに依存するシステムリソースを監視し、障害が発生した場合にサービスの処理を継続するか再開します。

高可用性については、85ページの「高可用性に関する FAQ」 を参照してください。

SunPlex システムのフェイルオーバーとスケーラビリティ

SunPlex システムを使用すると、フェイルオーバーまたはスケーラブルのどちらかをベースにしてサービスを実装できます。通常、フェイルオーバーサービスは可用性 (冗長性) のみが高く、スケーラブルサービスは可用性が高いとともに、パフォーマンスも向上します。同じクラスタでフェイルオーバーサービスとスケーラブルサービスを両方ともサポートできます。

フェイルオーバーサービス

フェイルオーバーとは、クラスタが、障害の発生した主ノードから指定した二次ノードにサービスを自動的に再配置するプロセスのことです。フェイルオーバーによって、Sun Cluster ソフトウェアは高い可用性を実現します。

フェイルオーバーが発生すると、クライアントでは、サービスが短時間中断して、フェイルオーバーの終了後に再接続しなければならない場合があります。しかし、クライアントは、サービスの提供元である物理サーバーを認識していません。

スケーラビリティサービス

フェイルオーバーは冗長性に関係していますが、スケーラビリティは負荷に関係なく一定した応答時間とスループットを提供するものです。スケーラブルサービスは、1 つのクラスタにある複数のノードに作用し、アプリケーションを同時に実行するため、パフォーマンスは向上します。スケーラブルな構成では、クラスタ内の各ノードは、データを提供して、クライアント要求を処理することができます。

フェイルオーバーサービスとスケーラブルサービスの詳細については、60ページの「データサービス」 を参照してください。

SunPlex システムの概要

この節では、SunPlex システムのユーザーを 3 種類に分け、各ユーザーに関連する概念とマニュアルについて説明します。各ユーザーは次のとおりです。

- ハードウェア保守担当者
- システム管理者
- アプリケーションプログラマ

ハードウェア保守担当者

ハードウェア保守担当者にとって、SunPlex システムは、サーバー、ネットワーク、および記憶装置を含む市販のハードウェアの集合に見えます。これらのコンポーネントは、すべてのコンポーネントにバックアップがあり、単一の障害によってシステム全体が停止しないように配線されています。

重要な概念 (ハードウェア保守担当者)

ハードウェア保守担当者は、クラスタに関する次の概念を理解する必要があります。

- クラスタハードウェアの構成と配線
- 設置と保守 (追加、取り外し、交換)
  - ネットワークインタフェースコンポーネント (アダプタ、接続点、ケーブル)
  - ディスクインタフェースカード
  - ディスクアレイ
  - ディスクドライブ
  - 管理コンソールとコンソールアクセスデバイス
- 管理コンソールとコンソールアクセスデバイスの設定

参照箇所 (ハードウェア保守担当者)

次の項には、前述の重要な概念に関連する説明が記載されています。

- 25ページの「クラスタノード」

- 27ページの「多重ホストディスク」
- 29ページの「ローカルディスク」
- 30ページの「クラスタインターコネクト」
- 31ページの「パブリックネットワークインタフェース」
- 31ページの「クライアントシステム」
- 32ページの「管理コンソール」
- 31ページの「コンソールアクセスデバイス」
- 33ページの「クラスタペアトポロジ」
- 35ページの「N+1 (星型) トポロジ」

SunPlex の関連マニュアル (ハードウェア保守担当者)

次の SunPlex のマニュアルには、ハードウェア保守の概念に関連する手順と情報が記載されています。

- 『*Sun Cluster 3.0 12/01 Hardware Guide*』

システム管理者

システム管理者にとって、SunPlex システムは、ケーブルによって接続された、記憶装置を共有するサーバー (ノード) の集合に見えます。システム管理者は、次のソフトウェアを扱います。

- クラスタノード間のコネクティビティを監視するための、Solaris ソフトウェアに統合された専用のクラスタソフトウェア
- クラスタノードで実行されるユーザーアプリケーションプログラムの状態を監視するための専用のソフトウェア
- ディスクを設定して管理するためのボリューム管理ソフトウェア
- 直接ディスクに接続されていないものも含め、すべてのノードが、すべての記憶装置にアクセスできるようにするための専用のクラスタソフトウェア
- ファイルがすべてのノードに対してローカルに接続されているように表示するための専用のソフトウェア

重要な概念 (システム管理者)

システム管理者は、次の概念とプロセスについて理解する必要があります。

- ハードウェアとソフトウェアの間の対話
- クラスタをインストールして構成する方法の一般的な流れ
  - Solaris オペレーティング環境のインストール
  - Sun Cluster ソフトウェアのインストールと構成
  - ボリューム管理ソフトウェアのインストールと構成
  - クラスタを動作可能状態にするためのアプリケーションソフトウェアのインストールと構成
  - Sun Cluster データサービスソフトウェアのインストールと構成
- クラスタハードウェアとソフトウェアのコンポーネントを追加、削除、交換、およびサービス提供するためのクラスタ管理手順
- パフォーマンスを向上させるための構成の変更方法

参照個所 (システム管理者)

次の項には、前述の重要な概念に関連する説明が記載されています。

- 39ページの「管理インタフェース」
- 41ページの「高可用性フレームワーク」
- 44ページの「広域デバイス」
- 46ページの「ディスクデバイスグループ」
- 48ページの「広域名前空間」
- 50ページの「クラスタファイルシステム」
- 53ページの「定足数と定足数デバイス」
- 59ページの「ボリューム管理ソフトウェア」
- 60ページの「データサービス」
- 73ページの「リソース、リソースグループ、リソースタイプ」
- 76ページの「パブリックネットワーク管理 (PNM) とネットワークアダプタフェイルオーバー (NAFO)」
- 第 4 章

参照箇所 (システム管理者)

次の SunPlex のマニュアルには、システム管理者の概念に関連する手順と情報が記載されています。

- 『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』
- 『*Sun Cluster 3.0 12/01* のシステム管理』
- 『*Sun Cluster 3.0 12/01 Error Messages Manual*』

アプリケーションプログラマ

SunPlex システムでは、Oracle、NFS、DNS、iPlanet™ Web Server、Apache Web Server、Netscape™ Directory Server といったアプリケーションが「データサービス」としてサポートされます。 データサービスを作成するには、既成のアプリケーションを Sun Cluster ソフトウェアの制御下で動作するように設定する必要があります。Sun Cluster ソフトウェアには、このようなアプリケーションの起動や停止、監視を行う構成ファイルと管理メソッドが含まれています。フェイルオーバーサービスやスケーラブルサービスを新たに作成する必要がある場合には、SunPlex アプリケーションプログラミングインタフェース (API) や Data Service Enabling Technologies API (DSET API) を使用し、アプリケーションをクラスタ上のデータサービスとして実行するために必要な構成ファイルや管理メソッドを作成することができます。

重要な概念 (アプリケーションプログラマ)

アプリケーションプログラマは、次の点について理解する必要があります。

- 各アプリケーションの特性。アプリケーションをフェイルオーバーまたはスケーラブルデータサービスとして実行できるかどうかを判断する必要があります。
- Sun Cluster API、DSET API、汎用データサービス。プログラマは、各自のアプリケーションをクラスタ環境に合わせて構成するプログラムまたはスクリプトを記述するために、どのツールが最も適しているかを判断する必要があります。

参照箇所 (アプリケーションプログラマ)

次の項には、前述の重要な概念に関連する説明が記載されています。

- 60ページの「データサービス」

- 73ページの「リソース、リソースグループ、リソースタイプ」
- 第 4 章

SunPlex 関連マニュアル (アプリケーションプログラマ)

次の SunPlex のマニュアルには、アプリケーションプログラミングの概念に関連する手順と情報が記載されています。

- 『*Sun Cluster 3.0 12/01* データサービス開発ガイド』
- 『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』

SunPlex システムの作業

すべての SunPlex システムの作業は、いくつかの概念的な予備知識が必要です。次の表に、作業と作業手順を記載しているマニュアルを示します。このマニュアルの概念に関する章では、各概念がこれらの作業とどのように対応するかを説明します。

表 1–1 作業マップ: ユーザーの作業と参照するマニュアル

実行する作業	使用するマニュアル
クラスタハードウェアの設置	『*Sun Cluster 3.0 12/01 Hardware Guide*』
クラスタへの Solaris ソフトウェアのインストール	『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』
Sun™ Management Center ソフトウェアのインストール	『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』
Sun Cluster ソフトウェアのインストールと構成	『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』
ボリューム管理ソフトウェアのインストールと構成	『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』 各ボリューム管理ソフトウェアのマニュアル
Sun Cluster データサービスのインストールと構成	『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』
クラスタハードウェアの保守	『*Sun Cluster 3.0 12/01 Hardware Guide*』

表 1–1 作業マップ: ユーザーの作業と参照するマニュアル 続く

実行する作業	使用するマニュアル
Sun Cluster ソフトウェアの管理	『*Sun Cluster 3.0 12/01* のシステム管理』
ボリューム管理ソフトウェアの管理	『*Sun Cluster 3.0 12/01* のシステム管理』 および各ボリューム管理ソフトウェアのマニュアル
アプリケーションソフトウェアの管理	各アプリケーションのマニュアル
問題の識別と対処方法	『*Sun Cluster 3.0 12/01 Error Messages Manual*』
新しいデータサービスの作成	『*Sun Cluster 3.0 12/01* データサービス開発ガイド』

第 2 章

重要な概念 – ハードウェアコンポーネント

この章では、SunPlex システム構成のハードウェアコンポーネントに関連する重要な概念について説明します。次のトピックについて述べます。

SunPlex システムハードウェアコンポーネント

ここで示す情報は、主にハードウェアサービスプロバイダを対象としています。これらの概念は、サービスプロバイダが、クラスタハードウェアの設置、構成、またはサービスを提供する前に、ハードウェアコンポーネント間の関係を理解するのに役立ちます。またこれらの情報は、クラスタシステムの管理者にとっても、クラスタソフトウェアをインストール、構成、管理するための予備知識として役立ちます。

クラスタは、次のようなハードウェアコンポーネントで構成されます。

- ローカルディスクを備えたクラスタノード (非共有)
- 多重ホスト記憶装置 (ノード間で共有されるディスク)
- リムーバブルメディア (テープ、CD-ROM)
- クラスタインターコネクト
- パブリックネットワークインタフェース
- クライアントシステム
- 管理コンソール
- コンソールアクセスデバイス

SunPlex システムを使用すると、33ページの「Sun Cluster トポロジの例」で説明するように、これらのコンポーネントを各種の構成に組み合わせることができます。

次の図は、クラスタの構成例を示しています。

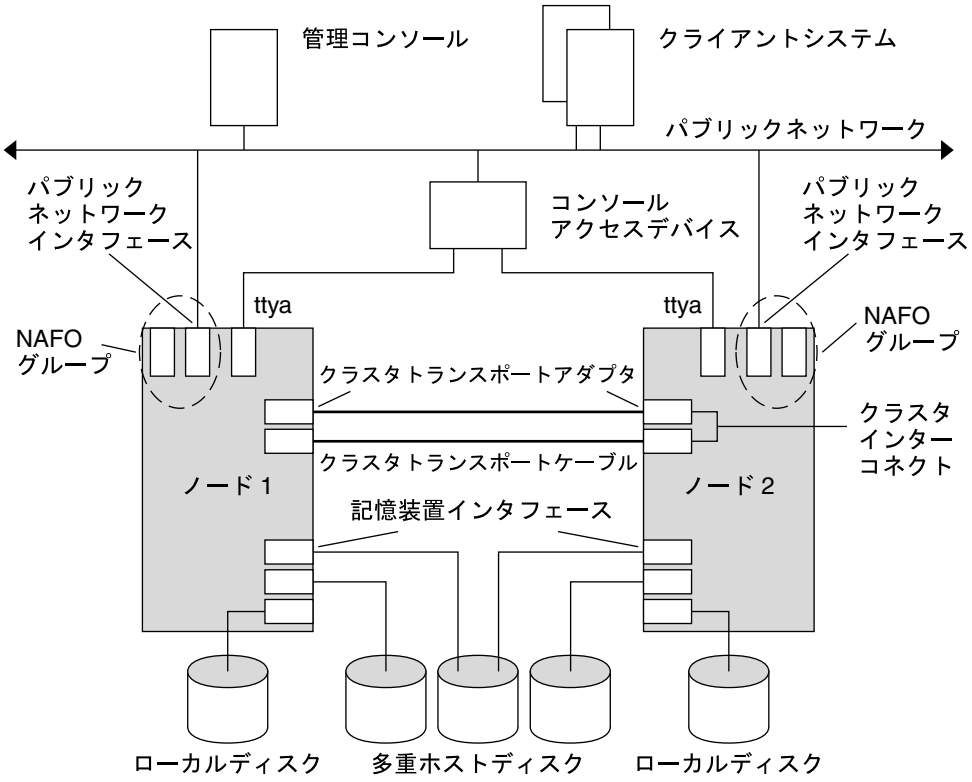

図 2–1　2 ノードクラスタ構成の例

クラスタノード

クラスタノードとは、Solaris オペレーティング環境と Sun Cluster ソフトウェアの両方を実行するマシンのことで、クラスタの現在のメンバー (クラスタメンバー) または潜在的なメンバーのどちらかです。Sun Cluster ソフトウェアを使用すると、1 つのクラスタに 2 ～ 8 台のノードを設定できます。サポートされるノード構成については、33ページの「Sun Cluster トポロジの例」を参照してください。

一般的にクラスタノードは、1 つまたは複数の多重ホストディスクに接続されます。多重ホストディスクに接続されていないノードは、クラスタファイルシステムを使用して多重ホストディスクにアクセスします。たとえば、スケーラブルサービスを 1 つ構成することで、ノードが多重ホストディスクに直接接続されていなくてもサービスを提供することができます。

さらに、パラレルデータベース構成では、複数のノードがすべてのディスクへの同時アクセスを共有します。パラレルデータベース構成については、27ページの「多重ホストディスク」と第 3 章を参照してください。

クラスタ内のノードはすべて、共通の名前 (クラスタ名) によってグループ化されます。この名前は、クラスタのアクセスと管理に使用されます。

パブリックネットワークアダプタは、ノードとパブリックネットワークを接続して、クラスタへのクライアントアクセスを可能にします。

クラスタメンバーは、1 つまたは複数の物理的に独立したネットワークを介して、クラスタ内の他のノードとやりとりします。物理的に独立したネットワークの集合は、クラスタインターコネクトと呼ばれます。

クラスタ内のすべてのノードが、別のノードがいつクラスタに結合されたか、またはクラスタから切り離されたかを認識します。さらに、クラスタ内のすべてのノードは、他のクラスタノードで実行されているリソースだけでなく、ローカルに実行されているリソースも認識します。

同じクラスタ内の各ノードの処理、メモリー、および入出力機能が同等のもので、パフォーマンスを著しく低下させることなく処理を継続できることを確認してください。フェイルオーバーの可能性があるため、すべてのノードに、バックアップまたは二次ノードとしてすべてのノードの作業負荷を引き受けるのに十分な予備容量が必要です。

各ノードは、独自のルート (/) ファイルシステムを起動します。

クラスタハードウェアメンバー用のソフトウェアコンポーネント

クラスタメンバーとして機能するには、次のソフトウェアがインストールされていなければなりません。

- Solaris オペレーティング環境
- Sun Cluster ソフトウェア
- データサービスアプリケーション
- ボリューム管理ソフトウェア (Solstice DiskSuite™ または VERITAS Volume Manager)

  例外として、複数のディスクの冗長配列 (RAID) を使用する Oracle Parallel Sever(OPS) 構成があります。この構成には、Oracle データを管理するために Solstice DiskSuite や VERITAS Volume Manager などのボリューム管理ソフトウェアは必要ありません。

Solaris オペレーティング環境、Sun Cluster、およびボリューム管理ソフトウェアをインストールする方法については、『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』を参照してください。

データサービスをインストールして構成する方法については、『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』を参照してください。

前述のソフトウェアコンポーネントの概念については、第 3 章を参照してください。

次の図は、Sun Cluster ソフトウェア環境を構成するソフトウェアコンポーネントとその関係を示しています。

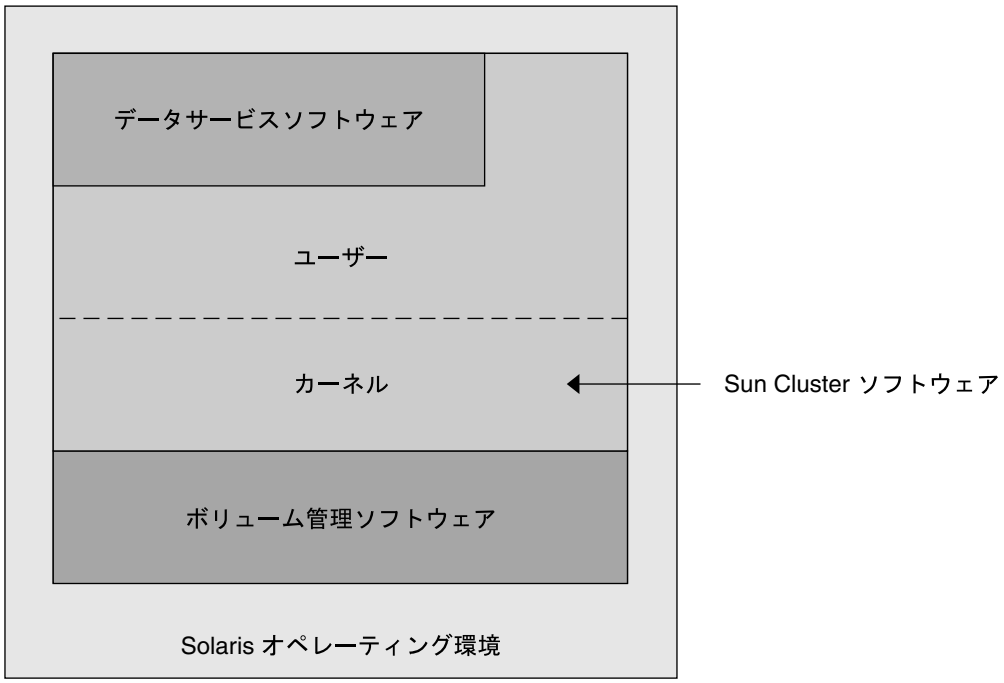

図 2–2　Sun Cluster ソフトウェアコンポーネントの相互の関係

クラスタメンバーについては、第 4 章を参照してください。

多重ホストディスク

Sun Cluster には、一度に複数のノードに接続可能なディスクである、多重ホストディスク記憶装置が必要です。Sun Cluster 環境では、多重ホスト記憶装置によってディスクデバイスの可用性を強化できます。

多重ホストディスクには、次の特徴があります。

- 多重ホストディスクは、単一ノードの障害に耐えられる。
- 多重ホストディスクはアプリケーションデータを保存するだけでなく、アプリケーションバイナリと構成ファイルも保存できる。
- 多重ホストディスクはノード障害を防止する。クライアント要求があるノードを介してデータにアクセスしていて失敗した場合、これらの要求は、同じディスクへの直接接続を持つ別のノードを使用するようにスイッチオーバーされます。

- 多重ホストディスクは、ディスクのマスターとなる主ノードによって広域的にアクセスされるか、ローカルパスによって同時に直接アクセスされる。現在、直接同時アクセスを使用するアプリケーションはOPSだけです。

ボリューム管理ソフトウェアは、多重ホストディスクのデータ冗長性に対して、ミラー化された構成またはRAID-5構成を提供します。現在、Sun Clusterは、ボリューム管理ソフトウェアとしてSolstice DiskSuite™ とVERITAS Volume Managerをサポートしています。さらに、Sun StorEdge™ A3x00ストレージユニットのRDAC RAID-5ハードウェアコントローラをサポートしています。

多重ホストディスクをミラー化およびストライプ化したディスクと組み合わせると、ノードの障害および個々のディスクの障害の両方に対する防御策となります。

多重ホスト記憶装置については、第4章を参照してください。

多重イニシエータ SCSI

この項は、多重ホストディスクに使用されるファイバチャネル記憶装置ではなく、SCSI記憶装置にのみ適用されます。

スタンドアロンサーバーでは、サーバーノードが、このサーバーを特定のSCSIバスに接続するSCSIホストアダプタ回路によって、SCSIバスのアクティビティを制御します。このSCSIホストアダプタ回路は、*SCSI*イニシエータと呼ばれます。この回路は、このSCSIバスに対するすべてのバスアクティビティを開始します。Sunシステムの SCSIホストアダプタのデフォルトSCSIアドレスは7です。

クラスタ構成では、多重ホストディスクを使用し、複数のサーバーノード間で記憶装置を共有します。クラスタ記憶装置がSCSIデバイスまたはDifferential SCSIデバイスからなる場合、この構成は多重イニシエータSCSIと呼ばれます。この用語が示すように、複数のSCSIイニシエータがSCSIバスに存在します。

SCSI仕様では、SCSIバスの各デバイスに一意のSCSIアドレスが必要です。(ホストアダプタも、SCSIバス上のデバイスの1つです。) 多重イニシエータ環境でデフォルトのハードウェア構成を行うと、すべてのSCSIホストアダプタがデフォルトで7に設定されるため衝突が生じます。

この衝突を解決するには、各SCSIバスで、SCSIアドレスが7のSCSIホストアダプタを1つ残し、他のホストアダプタには、未使用のSCSIアドレスを設定します。これらの未使用のSCSIアドレスには、現在未使用のアドレスと最終的に未使用となるアドレスの両方を含めるべきです。将来未使用となるアドレスの例としては、新しいドライブを空のドライブスロットに設置することによる記憶装置の追加

があります。ほとんどの構成では、二次ホストアダプタに使用できる SCSI アドレスは 6 です。

これらのホストアダプタに指定された SCSI アドレスは、`scsi-initiator-id` Open Boot PROM (OBP) プロパティを設定することにより変更できます。このプロパティは 1 つのノードに対して、またはホストアダプタごとに広域的に設定できます。一意の `scsi-initiator-id` を各 SCSI ホストアダプタに設定するための手順は、『*Sun Cluster 3.0 12/01 Hardware Guide*』の各ディスク格納装置に関する章に記載されています。

ローカルディスク

ローカルディスクとは、単一ノードにのみ接続されたディスクをいいます。したがって、これらはノードの障害に対して保護されていません (可用性が高くありません)。ただし、ローカルディスクを含むすべてのディスクが広域的名前空間に含まれ、広域デバイスとして構成されています。したがって、ディスク自体をすべてのクラスタノードから参照できます。

ローカルディスク上のファイルシステムは、広域マウントポイントに置くことによって、他のノードで使用できるようにすることができます。これらの広域ファイルシステムのいずれかがマウントされているノードに障害が生じると、すべてのノードがそのファイルシステムにアクセスできなくなります。ボリューム管理ソフトウェアを使用すると、これらのディスクがミラー化されるため、障害が発生してもこれらのファイルシステムがアクセス不能になることはありません。ただし、ノード障害をボリューム管理ソフトウェアで保護することはできません。

広域デバイスについては、44ページの「広域デバイス」を参照してください。

リムーバブルメディア

クラスタでは、テープドライブや CD-ROM ドライブなどのリムーバブルメディアがサポートされています。通常、これらのデバイスは、非クラスタ化環境と同じ方法で設置して構成することで使用できます。これらのデバイスは、Sun Cluster で広域デバイスとして構成されるため、クラスタ内の任意のノードから各デバイスにアクセスできます。リムーバブルメディアの設置方法と構成については、『*Sun Cluster 3.0 12/01 Hardware Guide*』を参照してください。

広域デバイスについては、44ページの「広域デバイス」を参照してください。

クラスタインターコネクト

クラスタインターコネクトとは、クラスタプライベート通信とデータサービス通信をクラスタノード間で転送するために使用される、デバイスの物理構成のことです。インターコネクトは、クラスタプライベート通信で拡張使用されるため、パフォーマンスが制限される可能性があります。

クラスタノードだけがプライベートインターコネクトに接続できます。Sun Cluster セキュリティモデルは、クラスタノードだけがプライベートインターコネクトに物理的にアクセスできるものと想定しています。

少なくとも 2 つの冗長な物理的に独立したネットワーク、またはパスを使用して、すべてのノードをクラスタインターコネクトによって接続し、単一地点による障害を回避する必要があります。任意の 2 つのノード間で、複数の物理的に独立したネットワーク (2 ～ 6) を設定できます。クラスタインターコネクトは、アダプタ、接続点、およびケーブルの 3 つのハードウェアコンポーネントからなります。

次に、これらの各ハードウェアコンポーネントを説明します。

- アダプタ – 個々のクラスタノードに存在するネットワークアダプタカード。この名前は、デバイス名と物理ユニット番号から構成されています (qfe2 など)。一部のアダプタには物理ネットワーク接続が 1 つしかありませんが、qfe カードのように複数の物理接続を持つものもあります。ネットワークインタフェースと記憶装置インタフェースの両方を持つものもあります。

  複数のインタフェースを持つネットワークカードは、カード全体に障害が生じると、単一地点による障害の原因となる可能性があります。可用性を最適にするには、2 つのノード間の唯一のパスが、単一のネットワークカードに依存しないように、クラスタを設定してください。

- 接続点 – クラスタノードの外部に常駐するスイッチ。これらは、パススルーおよび切り換え機能を実行して、3 つ以上のノードに同時に接続できるようにします。2 ノードクラスタでは、各ノードの冗長アダプタに接続された冗長物理ケーブルによって、ノードを相互に直接接続できるため、接続点は必要ありません。3 ノード以上の構成では、通常は接続点が必要です。

- ケーブル – 2 つのネットワークアダプタまたはアダプタと接続点の間をつなぐ物理接続。

クラスタインターコネクトについては、第 4 章を参照してください。

パブリックネットワークインタフェース

クライアントは、パブリックネットワークインタフェースを介してクラスタに接続します。各ネットワークアダプタカードは、カードに複数のハードウェアインタフェースがあるかどうかによって、1 つまたは複数のパブリックネットワークに接続できます。ノードは、複数のパブリックネットワークインタフェースカードを構成して、1 つのカードがアクティブで、もう 1 つのカードがバックアップとして動作するように設定できます。Sun Cluster ソフトウェアのサブシステムであるパブリックネットワーク管理 (PNM)は、アクティブインタフェースを監視します。アクティブアダプタに障害が生じると、ネットワークアダプタフェイルオーバー (NAFO) ソフトウェアが呼び出されて、バックアップアダプタの 1 つにインタフェースの処理を継続します。

パブリックネットワークインタフェースのクラスタ化に関連する特殊なハードウェアの考慮事項はありません。

パブリックネットワークについては、第 4 章を参照してください。

クライアントシステム

クライアントシステムには、パブリックネットワークによってクラスタにアクセスするワークステーションや他のサーバーが含まれます。クライアント側プログラムは、クライアントで実行されるサーバー側アプリケーションによって提供されるデータやサービスを使用します。

クライアントシステムの可用性は高くありません。クラスタ上のデータとアプリケーションは、高い可用性を備えています。

クラスタシステムについては、第 4 章を参照してください。

コンソールアクセスデバイス

すべてのクラスタノードへのコンソールアクセスが必要です。コンソールアクセスを行うには、クラスタハードウェアとともに購入した端末集配信装置か、 Sun Enterprise E10000 サーバーのシステムサービスプロセッサ (SSP)、Sun Fire™ サーバーのシステムコントローラ、または各ノードの `ttya` にアクセスできるその他のデバイスが必要です。

サポートされている唯一の端末集配信装置は、Sun から提供されています。サポートされている Sun の端末集配信装置の使用はオプションです。端末集配信装置を使

用すると、TCP/IP ネットワークを使用して、各ノードの /dev/console にアクセスできます。この結果、ネットワークの任意の場所にあるリモートワークステーションから、各ノードにコンソールレベルでアクセスできます。

システムサービスプロセッサ (SSP) は、Sun Enterprise E10000 サーバーへのコンソールアクセスを提供します。SSP は、Ethernet ネットワーク上のマシンであり、Sun Enterprise E10000 サーバーをサポートするように構成されています。SSP は、Sun Enterprise E10000 サーバーの管理コンソールです。Sun Enterprise E10000 サーバーのネットワークコンソール機能を使用すると、ネットワーク上のすべてのワークステーションがホストコンソールセッションを開くことができます。

これ以外のコンソールアクセス方式には、他の端末集配信装置、別ノードおよびダンプ端末からの tip(1) シリアルポートアクセスが含まれます。Sun™ キーボードとモニター、または他のシリアルポートデバイスも使用できます。

管理コンソール

管理コンソールと呼ばれる専用の SPARCstation™ システムを使用し、アクティブクラスタを管理できます。通常、Cluster Control Panel (CCP) や、Sun Management Center 用の Sun Cluster モジュールなどの管理ツールソフトウェアを管理コンソールにインストールして実行します。CCP で cconsole を使用すると、一度に複数のノードコンソールに接続できます。CCP の使用法の詳細については、『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

管理コンソールはクラスタノードではありません。管理コンソールは、パブリックネットワークを介して、または任意でネットワークベースの端末集配信装置 (コンセントレータ) を介して、クラスタノードへの遠隔アクセス用に使用します。クラスタが Sun Enterprise E10000 プラットフォームによって構成されている場合は、管理コンソールからシステムサービスプロセッサ (SSP) にログインし、netcon(1M) コマンドを使用して接続できなければなりません。

通常、ノードはモニターなしで構成します。ノードのコンソールには、端末集配信装置に接続され、さらにその端末集配装置からノードのシリアルポートに接続された管理コンソールから telnet セッションによってアクセスします。Sun Enterprise E10000 サーバーの場合は、システムサービスプロセッサ (SSP) から接続します。詳細は、31ページの「コンソールアクセスデバイス」を参照してください。

Sun Cluster には、専用の管理コンソールは必要ありませんが、専用コンソールを使用すると、次の利点が得られます。

- コンソールと管理ツールを同じマシンにまとめることで、クラスタ管理を一元化できる。
- ハードウェアサービスプロバイダによる問題解決が迅速に行われる

管理コンソールについては、第 4 章を参照してください。

Sun Cluster トポロジの例

トポロジとは、クラスタノードと、クラスタで使用される記憶装置プラットフォームを接続する接続スキーマをいいます。

Sun Cluster は、次のトポロジをサポートしています。

- クラスタペア
- ペア +M
- N+1 (星型)

次の各項では、それぞれのトポロジを表す図を示しています。

クラスタペアトポロジ

クラスタペアトポロジとは、単一のクラスタ管理フレームワークのもとで動作する複数のノードペアをいいます。この構成では、ペアの間でのみフェイルオーバーが発生します。ただし、すべてのノードがクラスタインターコネクトによって接続されていて、Sun Cluster ソフトウェア制御のもとで動作します。このトポロジを使用して、1 つのペアでパラレルデータベースアプリケーションを実行し、別のペアでフェイルオーバーまたはスケーラブルなアプリケーションを実行できます。

クラスタファイルシステムを使用すると、すべてのノードがアプリケーションデータを保存するディスクに直接接続されていない場合でも、複数のノードがスケーラブルサービス、またはパラレルデータベースを実行する 2 ペア構成を設定できます。

次の図は、クラスタペア構成を示しています。

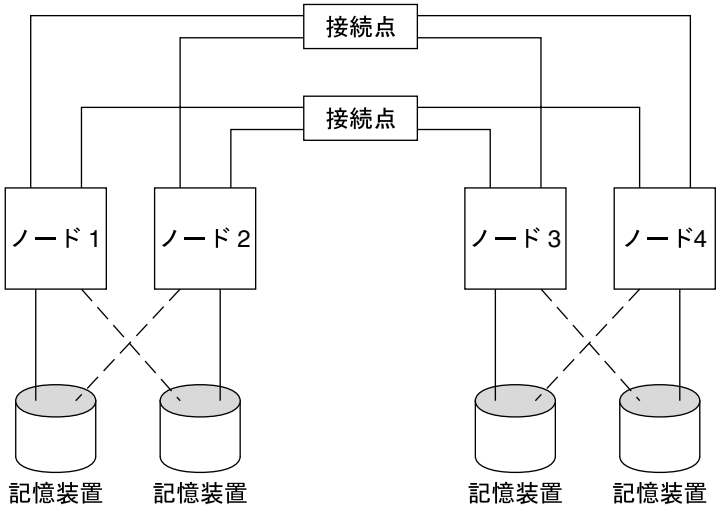

図 2–3　クラスタペアトポロジ

ペア +M トポロジ

ペア +M トポロジには、共有記憶装置に直接接続されたノードのペアと、クラスタインターコネクトを使用して共有記憶装置にアクセスするノードの追加セットが含まれます。これらのノードはそれぞれ直接には接続されていません。

次の図は、4 つのノードのうち 2 つ (ノード 3 とノード 4) がクラスタインターコネクトを使用して記憶装置にアクセスする、1 つのペア +M トポロジを示しています。この構成を拡張し、共有記憶装置には直接アクセスしない追加ノードを含めることができます。

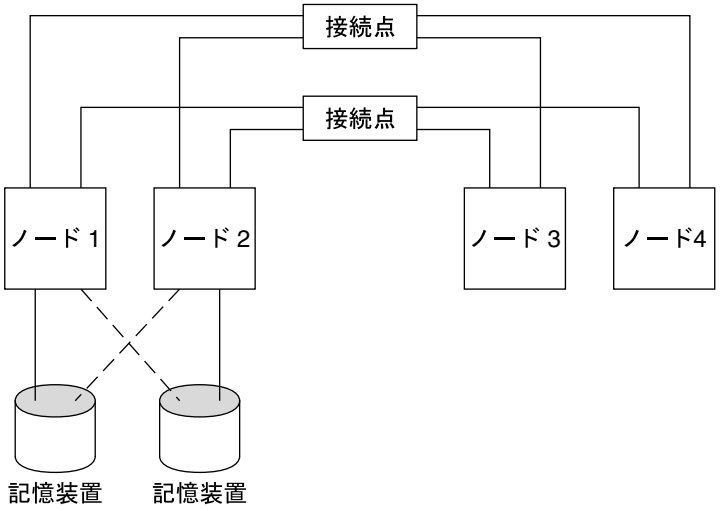

図 2–4　ペア +M トポロジ

N+1 (星型) トポロジ

N+1 トポロジには、いくつかの主ノードと 1 つの二次ノードが含まれます。主ノードと二次ノードを同等に構成する必要はありません。主ノードは、アプリケーションサービスをアクティブに提供します。二次ノードは、主ノードに障害が生じるのを待機する間、アイドル状態である必要はありません。

二次ノードは、この構成ですべての多重ホスト記憶装置に物理的に接続されている唯一のノードです。

主ノードで障害が発生すると、Sun Cluster はそのリソースの処理を二次ノードで続行し、リソースは自動または手動で主ノードに切り換えられるまで二次ノードで機能します。

二次ノードには、主ノードの 1 つに障害が発生した場合に負荷を処理できるだけの十分な予備の CPU 容量が常に必要です。

次の図は、N+1 構成を示しています。

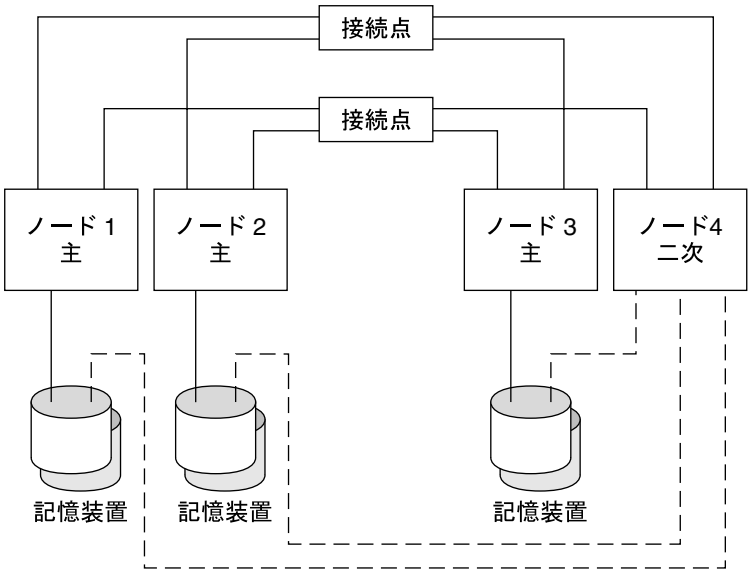

図 2–5　N+1 トポロジ

第 3 章

重要な概念 – 管理とアプリケーション開発

この章では、SunPlex システム構成のソフトウェアコンポーネントに関連する重要な概念について説明します。次のトピックについて述べます。

クラスタ管理とアプリケーション開発

この情報は、主に、SunPlex API および SDK を使用するシステム管理者とアプリケーション開発者を対象としています。クラスタシステムの管理者にとっては、この情報は、クラスタソフトウェアのインストール、構成、管理についての予備知識となります。アプリケーション開発者は、この情報を使用して、作業を行うクラスタ環境を理解できます。

次の図は、クラスタ管理の概念がクラスタの構造にどのように対応するかを示しています。

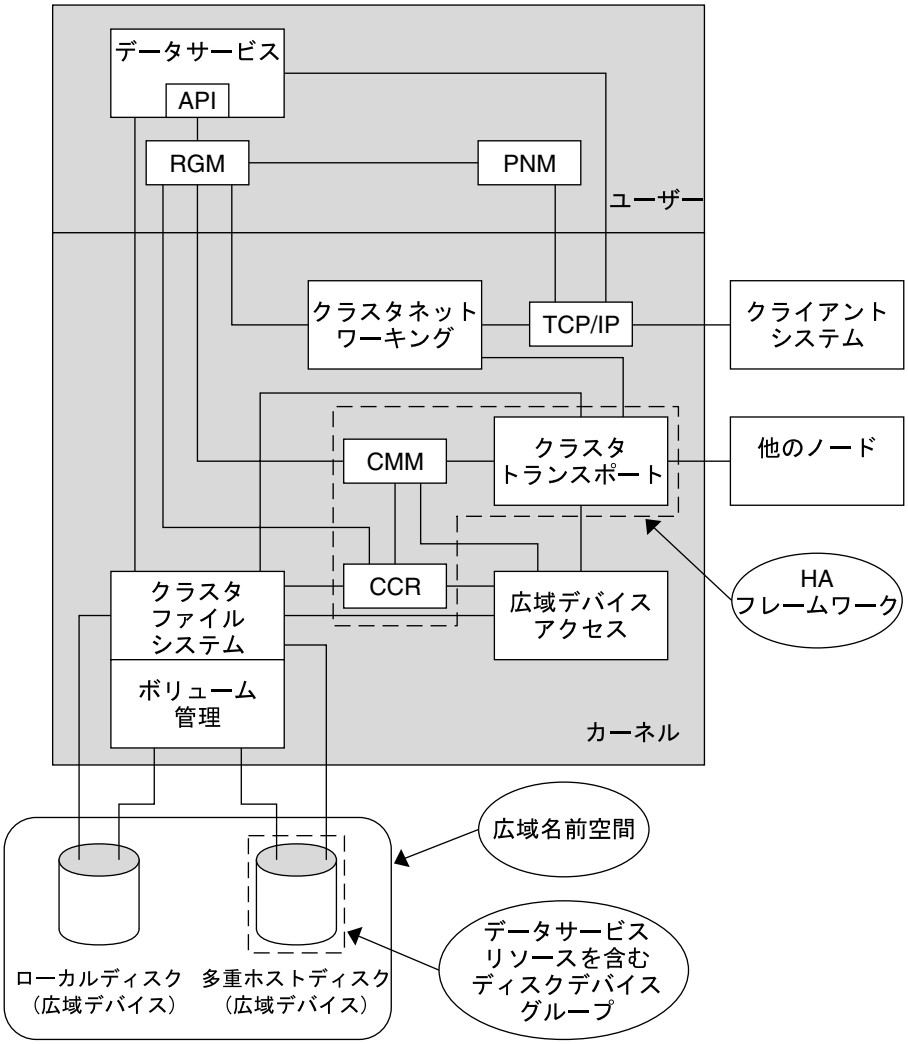

図 3–1　Sun Cluster ソフトウェアの構造

管理インタフェース

任意のユーザーインタフェースを使用し、SunPlex をインストール、構成、管理できます。システム管理作業は、コマンド行インタフェースで行えます。コマンド行インタフェースには、構成作業を簡略化するユーティリティーがあります。SunPlex システムには、Sun Management Center の一部として実行される、特定のクラスタ作業に GUI を提供するモジュールもあります。管理インタフェースの詳細

については、『*Sun Cluster 3.0 12/01* のシステム管理』の概要を説明している章を参照してください。

クラスタ内の時間

クラスタ内のすべてのノード間の時刻は同期をとる必要があります。クラスタノードの時刻と外部の時刻ソースの同期をとるかどうかは、クラスタの操作にとって重要ではありません。SunPlex システムは、Network Time Protocol (NTP) を使用し、ノード間のクロックの同期をとっています。

通常、システムクロックが数分の 1 秒程度変更されても問題は起こりません。しかし、システムクロックと時刻の起点の同期をとるため
に、`date(1)`、`rdate(1M)`、`xntpdate(1M)` を (対話形式または `cron` スクリプト内で) アクティブクラスタに対して実行すると、これよりも大幅に時刻変更が可能です。この強制的な変更は、ファイル修正時刻の表示に問題が生じたり、NTP サービスに混乱が生じる場合があります。

Solaris オペレーティング環境を各クラスタノードにインストールする場合は、ノードのデフォルトの時刻と日付の設定を変更できます。通常は、工場出荷時のデフォルト値を使用します。

`scinstall(1M)` を使用して Sun Cluster ソフトウェアをインストールする場合、インストールプロセスの手順の 1 つとして、クラスタの NTP を構成します。Sun Cluster ソフトウェアは、テンプレートファイル `ntp.cluster` を提供します (インストールされたクラスタノードの `/etc/inet/ntp.cluster` を参照)。これは、1 つのノードを優先ノードにし、すべてのクラスタノード間で対等関係を確立します。各ノードはそれぞれのプライベートホスト名で識別され、時刻の同期化がクラスタインターコネクト全体で行なわれます。NTP のクラスタを構成する方法については、『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』を参照してください。

また、クラスタの外部に 1 つまたは複数の NTP サーバーを設定し、`ntp.conf` ファイルを変更してその構成を反映させることもできます。

通常の操作では、クラスタの時刻を調整する必要はありません。ただし Solaris オペレーティング環境をインストールしたときに時刻が正しく設定されておらず、変更する必要がある場合は、『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

高可用性フレームワーク

SunPlex システムでは、ネットワークインタフェース、アプリケーションそのもの、ファイルシステム、および多重ホストディスクなど、ユーザーとデータ間のパスにおけるすべてのコンポーネントの可用性が高くなっています。一般に、クラスタコンポーネントは、システムで単一 (ソフトウェアまたはハードウェア) の障害が発生しても稼働し続けられる場合は可用性が高くなります。

次の表は、SunPlex コンポーネントの障害の種類 (ハードウェアとソフトウェアの両方) と、高可用性フレームワークに組み込まれた回復の種類を示しています。

表 3–1 SunPlex システムの障害の検出と回復のレベル

障害が発生したクラスタリソース	ソフトウェアの回復	ハードウェアの回復
データサービス	HA API、HA フレームワーク	なし
パブリックネットワークアダプタ	ネットワークアダプタのフェイルオーバー (NAFO)	複数のパブリックネットワークアダプタカード
クラスタファイルシステム	主複製と二次複製	多重ホストディスク
ミラー化された多重ホストディスク	ボリューム管理 (Solstice DiskSuite および VERITAS Volume Manager)	ハードウェア RAID-5 (Sun StorEdge A3x00 など)
広域デバイス	主複製と二次複製	デバイス、クラスタトランスポート接続点への多重パス
プライベートネットワーク	HA トランスポートソフトウェア	ハードウェアから独立した多重プライベートネットワーク
ノード	CMM、failfast ドライバ	複数ノード

Sun Cluster ソフトウェアの高可用性フレームワークは、ノードの障害を素早く検出して、クラスタ内の残りのノードにあるフレームワークリソース用に新しい同等のサーバーを作成します。すべてのフレームワークリソースが使用できなくなるときはありません。障害が発生したノードによって影響を受けないフレームワークリソースは、回復中も完全に使用できます。さらに、障害が発生したノードのフレー

ムワークリソースは、回復されると同時に使用可能になります。回復されたフレームワークリソースは、他のすべてのフレームワークリソースが回復するまで待機する必要はありません。

最も可用性の高いフレームワークリソースは、リソースを使用してアプリケーション (データサービス) に透過的に回復されます。フレームワークリソースのアクセス方式は、ノードの障害時にも完全に維持されます。アプリケーションは、フレームワークリソースサーバーが別のノードに移動したことを認識できないだけです。単一ノードの障害は、別ノードからのディスクに対する代替ハードウェアパスが存在するかぎり、ファイル、デバイス、およびディスクボリュームを使用する残りのノード上のプログラムに対して完全に透過的です。この例としては、複数ノードへのポートを持つ多重ホストディスクの使用があります。

クラスタメンバーシップモニター

クラスタメンバーシップモニター (CMM) は、エージェントの分散型セットであり、クラスタメンバーごとに 1 つあります。これらのエージェントは、クラスタインターコネクトによってメッセージを交換して、次のことを行います。

- すべてのノード (定足数) で一貫したメンバーシップの表示を行う
- メンバーシップの変更に応じて、登録されたコールバックを使用して同期化された再構成を行う
- クラスタパーティション分割を処理する (split-brain と amnesia)
- すべてのクラスタメンバー間での完全な接続を保証する

Sun Cluster ソフトウェアの以前のリリースとは異なり、CMM は完全にカーネルで実行されます。

クラスタメンバーシップ

CMM の主な機能は、特定の時刻にクラスタに属するノードの集合に対して、クラスタ全体の同意を確立することです。この制約をクラスタメンバーシップと呼びます。

クラスタメンバーシップを決定して、最終的にデータの完全性を保証するために、CMM は次のことを行います。

- クラスタへのノードの結合、またはクラスタからのノードの切り離しなど、クラスタメンバーシップの変更を考慮する
- 障害のあるノードがクラスタから切り離されるように保証する

- 障害のあるノードが、修復されるまでクラスタの外部におかれるように保証する
- クラスタそのものがノードのサブセットに分割されないように防止する

クラスタが複数の独立したクラスタに分割されないようにする方法については、53ページの「定足数と定足数デバイス」を参照してください。

クラスタメンバーシップモニターの再構成

データが破壊から保護されるように保証するには、すべてのノードが、クラスタメンバーシップに対して一定の同意に達していなければなりません。必要であれば、CMM は、障害に応じてクラスタサービス (アプリケーション) のクラスタ再構成を調整します。

CMM は、クラスタのトランスポート層から、他のノードへの接続に関する情報を受け取ります。CMM は、クラスタインターコネクトを使用して、再構成中に状態情報を交換します。

CMM は、クラスタメンバーシップの変更を検出すると、クラスタの同期化構成を実行します。これにより、クラスタリソースは、クラスタの新しいメンバーシップに基づいて再分配されます。

フェイルファースト機構

CMM は、ノードに重大な問題が発生したことを検出すると、クラスタフレームワークに依頼して、そのノードを強制的に停止 (パニック) 状態にし、クラスタメンバーシップから除きます。この機構を「フェイルファースト」といいます。フェイルファーストでは、ノードは次の 2 つの方法で停止されます。

- クラスタから切り離されたノードが定足数なしに再び新しいクラスタを起動しようとすると、ノードは共有ディスクへのアクセスを制限されます。フェイルファーストのこの機能については、57ページの「障害による影響の防止」を参照してください。
- クラスタ固有の 1 つまたは複数のデーモン (`clexecd`、`rpc.pmfd`、`rgmd`、`rpc.ed`) が停止すると、CMM はそれを検出し、ノードを強制的に停止 (パニック) 状態にします。

クラスタデーモンの停止によってノードが停止すると、そのノードのコンソールに次のようなメッセージが表示されます。

```
panic[cpu0]/thread=40e60: Failfast: Aborting because "pmfd" died 35 seconds ago.
```

```
409b8 cl_runtime:__0FZsc_syslog_msg_log_no_argsPviTCPCcTB+48(70f900, 30, 70df54, 407acc, 0)
%l0-7: 1006c80 000000a 000000a 10093bc 406d3c80 7110340 0000000 4001 fbf0
```

パニックが発生したノードは、再起動を行ってクラスタに再び結合しようとするか、OpenBoot PROM (OBP) プロンプトの状態に留まることができます。どちらのアクションをとるかは、OBP の `auto-boot?` パラメータの設定に依存します。

クラスタ構成レポジトリ (CCR)

クラスタ構成レポジトリ (CCR) は、クラスタの構成と状態に関する情報を保存するための、プライベートなクラスタ全体のデータベースです。CCR は分散データベースです。各ノードは、このデータベースの完全なコピーを維持します。CCR は、すべてのノードで、クラスタ全体の一貫した表示が行われるように保証します。データの破壊を避けるために、各ノードは、クラスタリソースの現在の状態を知る必要があります。

CCR は、更新に 2 フェーズのコミットアルゴリズムを使用します。更新はすべてのクラスタメンバーで正常に終了しなければなりません。そうしないと、その更新はロールバックされます。CCR はクラスタインターコネクトを使用して、分散更新を適用します。

注意 - CCR はテキストファイルで構成されていますが、CCR ファイルを手作業で絶対に編集しないでください。各ファイルには、ノード間の一貫性を保証するための検査合計レコードが含まれます。CCR ファイルを手作業で更新すると、ノードまたはクラスタ全体の機能が停止する可能性があります。

CCR は、CMM に依存して、定足数 (quorum) が確立された場合にのみクラスタが実行されるように保証します。CCR は、クラスタ全体のデータの一貫性を確認し、必要に応じて回復を実行し、データへの更新を容易にします。

広域デバイス

SunPlex システムは、広域デバイスを使用して、デバイスが物理的に接続されている場所に関係なく、任意のノードからクラスタ内のすべてのデバイスに対して、ク

ラスタ全体の可用性の高いアクセスを可能にします。通常、広域デバイスへのアクセスを提供しているときにノードに障害が発生すると、Sun Cluster ソフトウェアはそのデバイスへの別のパスを自動的に検出して、そのパスにアクセスを切り替えます。SunPlex 広域デバイスには、ディスク、CD-ROM、テープが含まれます。ディスクは、唯一サポートされている多重ポート広域デバイスです。つまり、CD-ROM とテープは、現在可用性の高いデバイスではありません。各サーバーのローカルディスクも多重ポート化されていないため、可用性の高いデバイスではありません。

クラスタは、クラスタ内の各ディスク、CD-ROM、テープデバイスに一意の ID を自動的に割り当てます。この割り当てによって、クラスタ内の任意のノードから各デバイスに対して一貫したアクセスが可能になります。広域デバイス名前空間は、`/dev/global` ディレクトリにあります。詳細は、48ページの「広域名前空間」を参照してください。

多重ポート広域デバイスは、1 つのデバイスに対して複数のパスを提供します。多重ホストディスクの場合、ディスクは複数のノードがホストとなるディスクデバイスグループの一部であるため、多重ホストディスクの可用性は高くなります。

デバイス ID (DID)

Sun Cluster ソフトウェアは、デバイス ID (DID) 擬似ドライバと呼ばれる構造によって広域デバイスを管理します。このドライバは、多重ホストディスク、テープドライブ、CD-ROM を含むクラスタ内のすべてのデバイスに対して、一意の ID を自動的に割り当てるために使用されます。

デバイス ID (DID) 擬似ドライバは、クラスタの広域デバイスアクセス機能の重要な部分です。DID ドライバは、クラスタのすべてのノードを探索して、一意のディスクデバイスのリストを作成し、それぞれに対して、クラスタのすべてのノードで一貫した一意のメジャー番号およびマイナー番号を割り当てます。広域デバイスへのアクセスは、ディスクを示す `c0t0d0` などの従来の Solaris デバイス ID ではなく、DID ドライバによって割り当てられた一意のデバイス ID を利用して行われます。

この方法により、ディスクを利用するすべてのアプリケーション (ボリューム管理ソフトウェアまたは raw デバイスを使用するアプリケーション) が、一貫したパスを使用してクラスタ全体にアクセスできます。各デバイスのローカルメジャー番号およびマイナー番号はノードによって異なり、Solaris デバイス命名規則も変更する可能性があるため、この一貫性は、多重ホストディスクにとって特に重要です。たとえば、ノード 1 は多重ホストディスクを `c1t2d0` と表示し、同じディスクをノード 2 は `c3t2d0` と表示する場合があります。DID ドライバは、そのノードが

代わりに使用する d10 などの広域名を割り当てて、各ノードに対して多重ホストディスクとの一貫したマッピングを与えます。

デバイス ID は、`scdidadm(1M)` および `scgdevs(1M)` によって更新、管理します。詳細については、それぞれのマニュアルページを参照してください。

ディスクデバイスグループ

SunPlex システムでは、すべての多重ホストディスクが、Sun Cluster ソフトウェアフレームワークの制御のもとになければなりません。まず、ボリューム管理ソフトウェアのディスクグループ (Solstice DiskSuite ディスクセットか VERITAS Volume Manager ディスクグループ) を多重ホストディスクに作成します。次に、ボリューム管理ソフトウェアのディスクグループをディスクデバイスグループとして登録します。ディスクデバイスグループは、広域デバイスの一種です。さらに、Sun Cluster ソフトウェアは、個々のディスクデバイスやテープデバイスごとに raw ディスクデバイスグループを自動的に作成します。ただし、これらのクラスタデバイスグループは、広域デバイスとしてアクセスされるまではオフラインの状態になっています。

この登録によって、SunPlex システムは、どのノードがどのボリュームマネージャディスクグループへのパスをもっているかを知ることができます。この時点でそのボリュームマネージャデバイスグループは、クラスタ内で広域アクセスが可能になります。あるディスクデバイスグループが複数のノードから書き込みができる (制御できる) 場合は、そのディスクデバイスグループに格納されるデータは、高度な可用性を有することになります。高度な可用性を備えたディスクデバイスグループには、クラスタファイルシステムを格納できます。

注 - ディスクデバイスグループは、リソースグループとは別のものです。あるノードが 1 つのリソースグループ (データサービスプロセスのグループを表す) をマスターする一方で、別のノードが、データサービスによってアクセスされるディスクグループをマスターできます。ただし、最も良い方法は、特定のアプリケーションのデータを保存するディスクデバイスグループと、アプリケーションのリソース (アプリケーションデーモン) を同じノードに含むリソースグループを維持することです。ディスクデバイスグループとリソースグループの関連付けの詳細については、『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』の概要を示す章を参照してください。

ディスクデバイスグループでは、ボリューム管理ソフトウェアのディスクグループは実際に使用するディスクに対してマルチパスサポートを提供するため、広域にな

ります。多重ホストディスクに物理的に接続された各クラスタノードは、ディスクデバイスグループへのパスを提供します。

ディスクデバイスグループのフェイルオーバー

ディスク格納装置は複数のノードに接続されるため、現在デバイスグループをマスターしているノードに障害が生じた場合、その格納装置にあるすべてのディスクデバイスグループには、代替パスによってアクセスできます。デバイスグループをマスターするノードの障害は、回復と一貫性の検査を実行するために要する時間を除けば、デバイスグループへのアクセスに影響しません。この時間中、デバイスグループが使用可能になるまで、すべての要求は (アプリケーションには透過的に) 阻止されます。

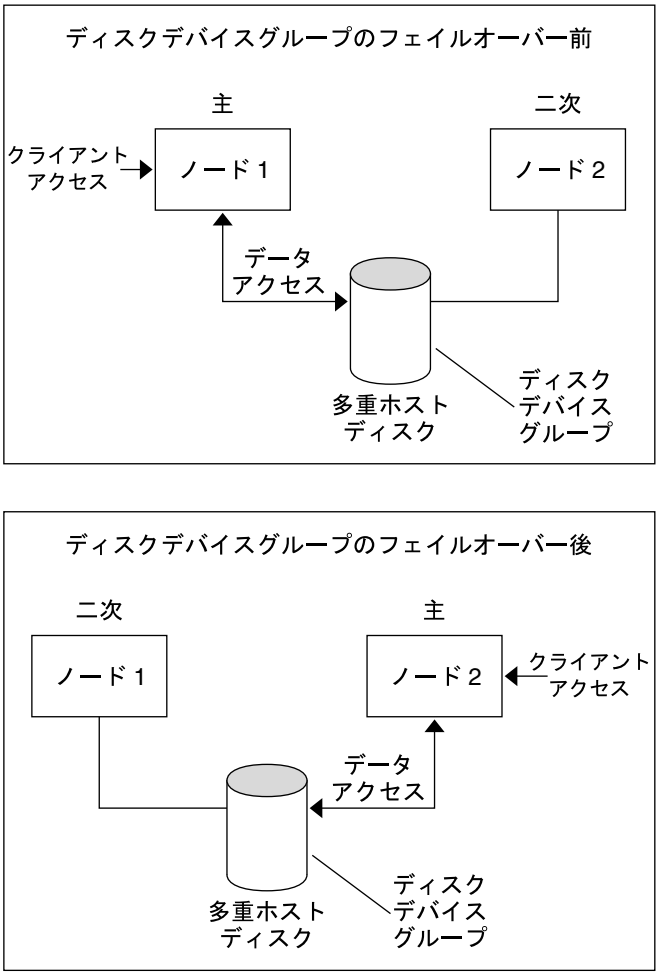

図 3–2　ディスクデバイスグループのフェイルオーバー

広域名前空間

広域名前空間は、広域デバイスを有効にする Sun Cluster ソフトウェアの機構です。広域名前空間には、ボリューム管理ソフトウェアの名前空間とともに、`/dev/global/` 階層が含まれます。広域名前空間は、多重ホストディスクとローカルディスクの両方 (および CD-ROM やテープなどの他のクラスタデバイスすべて) を反映して、多重ホストディスクへの複数のフェイルオーバーパスを提供します。多重ホストディスクに物理的に接続された各ノードは、クラスタ内のすべてのノードの記憶装置に対するパスを提供します。

通常、ボリューム管理ソフトウェアの名前空間は、Solstice DiskSuite の場合は、/dev/md/*diskset*/dsk (および rdsk) ディレクトリに、 VxVM の場合は、/dev/vx/dsk/*disk-group* ディレクトリと /dev/vx/rdsk/*disk-group* ディレクトリにそれぞれあります。 これらの名前空間は、クラスタ全体にインポートされた Solstice DiskSuite の各ディスクセットと VxVM の各ディスクグループのディレクトリからなります。これらの各ディレクトリには、そのディスクセットまたはディスクグループ内の各メタデバイスまたはボリュームのデバイスノードが入っています。

SunPlex システムでは、ボリューム管理ソフトウェアのローカルの名前空間の各デバイスノードは、/global/.devices/node@*nodeID* ファイルシステム内のデバイスノードへのシンボリックリンクとして表されます。*nodeID* は、クラスタの各ノードを表す整数です。Sun Cluster ソフトウェアは、ボリューム管理ソフトウェアデバイスをシンボリックリンクとしてそれぞれの標準的な場所にも維持します。広域名前空間と標準ボリューム管理ソフトウェア名前空間の両方を任意のクラスタノードから使用できます。

広域名前空間には、次の利点があります。

- 各ノードの独立性が高く、デバイス管理モデルをほとんど変更しなくてすむ。
- デバイスを選択的に広域にできる。
- Sun の製品以外のリンクジェネレータが引き続き動作する。
- ローカルデバイス名を指定すると、その広域名を取得するために簡単なマッピングが提供される。

ローカル名前空間と広域名前空間の例

次の表は、多重ホストディスク c0t0d0s0 でのローカル名前空間と広域名前空間のマッピングを示しています。

表 3–2 ローカル名前空間と広域名前空間のマッピング

コンポーネント /パス	ローカルノード名前空間	広域名前空間
Solaris 論理名	/dev/dsk/c0t0d0s0	/global/.devices/node@ *ID*/dev/dsk/c0t0d0s0
DID 名	/dev/did/dsk/d0s0	/global/.devices/node@*ID*/dev/did/dsk/d0s0

表 3–2 ローカル名前空間と広域名前空間のマッピング 続く

コンポーネント/パス	ローカルノード名前空間	広域名前空間
Solstice DiskSuite	/dev/md/*diskset*/dsk/d0	/global/.devices/node@*ID*/dev/md/ *diskset*/dsk/d0
Solstice DiskSuite	/dev/vx/dsk/*disk-group*/v0	/global/.devices/node@*ID*/dev/vx/dsk/*disk-group*/v0

広域名前空間はインストール時に自動的に生成されて、再構成再起動のたびに更新されます。広域名前空間は、scgdevs(1M) コマンドを実行することでも生成できます。

クラスタファイルシステム

クラスタファイルシステムは、1 つのノード上のカーネル、配下のファイルシステムおよびディスクへの物理接続を持つノードで実行されるボリューム管理ソフトウェアとの間のプロキシです。

クラスタファイルシステムは、1 つまたは複数のノードへの物理接続を持つ広域デバイス (ディスク、テープ、CD-ROM) に依存しています。広域デバイスは、ノードが記憶装置への物理接続を持つかどうかに関係なく、同じファイル名 (たとえば、/dev/global/) によってクラスタ内のすべてのノードからアクセスできます。広域デバイスは通常のデバイスと同様に使用できます。つまり、newfs または mkfs、あるいはこの両方を使用し、ファイルシステムを作成できます。

広域デバイスには、mount -g を使用して広域に、または mount を使用してローカルにファイルシステムをマウントできます。

プログラムは、同じファイル名 (たとえば、/global/foo) によって、クラスタ内のすべてのノードからクラスタファイルシステムのファイルにアクセスできます。

クラスタファイルシステムは、すべてのクラスタメンバーにマウントされます。クラスタファイルシステムをクラスタメンバーのサブセットにマウントすることはできません。

クラスタファイルシステムは、特定のファイルシステムタイプではありません。つまり、クライアントは、UFS など、実際に使用するファイルシステムしか認識できません。

クラスタファイルシステムの使用法

SunPlex システムでは、すべての多重ホストディスクがディスクデバイスグループとして構成されます。このディスクグループは、Solstice DiskSuite ディスクセットか、VxVM ディスクグループ、またはソフトウェアベースのボリューム管理ソフトウェアの制御下にない個々のディスクです。

クラスタファイルシステムを高可用性にするには、使用するディスクストレージが複数のノードに接続されていなければなりません。したがって、クラスタファイルシステム内のローカルファイルシステム (ノードのローカルディスクに格納されているファイルシステム) は、高可用性ではありません。

通常のファイルシステムと同様、クラスタファイルシステムは 2 つの方法でマウントできます。

- 手作業によるマウント— `mount` コマンドと `–g` または `–o global` マウントオプションを使って、コマンド行からクラスタファイルシステムをマウントします。次はその例です。

```
# mount -g /dev/global/dsk/d0s0 /global/oracle/data
```

- 自動マウント— `global` マウントオプションで `/etc/vfstab` ファイルにエントリを作成して、ブート時にクラスタファイルシステムをマウントします。さらに、すべてのノードの `/global` ディレクトリ下にマウントポイントを作成します。ディレクトリ `/global` を推奨しますが、他の場所でも構いません。次に、`/etc/vfstab` ファイルの、クラスタファイルシステムを示す行の例を示します。

```
/dev/md/oracle/dsk/d1 /dev/md/oracle/rdsk/d1 /global/oracle/
data ufs 2 yes global,logging
```

注 - Sun Cluster ソフトウェアには、クラスタファイルシステムに対する特定の命名規則はありません。しかし、`/global/`*disk-device-group* など同じディレクトリのもとにすべてのクラスタファイルシステムのマウントポイントを作成すると、管理が容易になります。詳細は、『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』と『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

クラスタファイルシステムの機能

クラスタファイルシステムには、次の機能があります。

- ファイルのアクセス場所が透過的になります。プロセスはファイルがシステム内のどこに置かれていても開くことができ、また、すべてのノード上のプロセスが同じパス名を使用してファイルを見つけられます。

注 - クラスタファイルシステムは、ファイルを読み取る際に、ファイル上のアクセス時刻を更新しません。

- 一貫したプロトコルを使用して、ファイルが複数のノードから同時にアクセスされている場合でも、UNIX ファイルアクセス方式を維持します。
- 拡張キャッシュ機能とゼロコピーバルク入出力移動機能により、ファイルデータを効率的に移動することができます。
- クラスタファイルシステムには、`fcntl (2)` インタフェースに基づく、高度な可用性を備えたアドバイザリファイルロッキング機能があります。複数のクラスタノードで動作するアプリケーションは、クラスタファイルシステムのファイルに対してアドバイザリファイルロッキング機能を使用することによって、データへのアクセスを同期化することができます。ファイルロックを所有するノードがクラスタから切り離されたり、ファイルロックを所有するアプリケーションが異常停止すると、それらのロックはただちに解放されます。
- 障害が発生した場合でも、データへの連続したアクセスが可能です。アプリケーションは、ディスクへのパスが有効である限り、障害による影響を受けません。この保証は、raw ディスクアクセスとすべてのファイルシステム操作で維持されます。
- クラスタシステムファイルは、実際のファイルシステムおよびボリューム管理ソフトウェアに依存していません。クラスタシステムファイルは、サポートされているディスク上のファイルシステムすべてを広域にします。

syncdir マウントオプション

ファイルシステムとして UFS を使用するクラスタファイルシステムには、`syncdir` マウントオプションを使用できます。しかし、`syncdir` を指定しない方がパフォーマンスは向上します。`syncdir` を指定すると、POSIX 準拠の書き込みが保証されます。指定しないと、UFS ファイルシステムの場合と同じ動作となります。たとえば、`syncdir` を指定しないと、場合によっては、ファイルを閉じるまでスペース不足条件を検出できません。`syncdir` (および POSIX 動作) を指定すると、スペース

不足条件は書き込み動作中に検出されます。syncdir を指定しないことで生じる問題はほとんどないため、このオプションを指定しないで、パフォーマンスを向上させることを推奨します。

広域デバイスとクラスタファイルシステムについては、86ページの「ファイルシステムに関する FAQ」を参照してください。

定足数と定足数デバイス

クラスタノードではデータやリソースが共有されているため、クラスタが、同時にアクティブな複数のパーティションに分割されてはなりません。CMM では、クラスタインターコネクトがパーティション分割されていても、ある時点で動作するクラスタは常に 1 つだけです。

クラスタのパーティション分割によって起こる問題に split-brain と amnesia があります。split-brain の問題は、ノード間のインターコネクトが失われ、クラスタが複数のサブクラスタにパーティション分割されたときに発生します。この場合、個々のパーティションはそれ自体が唯一のパーティションであるとみなしますが、その原因は、クラスタノード間の通信が阻害されたことにあります。amnesia の問題は、停止したクラスタが、停止時よりも古いクラスタデータに基づいて再起動されたときに発生します。たとえば、フレームワークデータの複数のバージョンがディスクに格納されている状態で、クラスタが新たに起動されたときに最新バージョンが使用できないと、このような問題が起こる可能性があります。

split-brain と amnesia の問題は、各ノードに 1 票を与え、過半数の投票がないとクラスタが動作しないようにすることで防止できます。過半数の投票を得たパーティションは「定足数 (quorum)」を獲得し、アクティブになります。この過半数投票の仕組みは、クラスタに 3 つ以上のノードがある限り正しく機能します。しかし、2 ノードクラスタでは過半数が 2 であるため、このようなクラスタがパーティション分割されると、外部からの投票がない限り、どちらのパーティションも定足数を獲得することはできません。 この外部からの投票は、「定足数デバイス (quorum device)」によって行われます。定足数デバイスは、2 つのノード間で共有されているディスクであれば、何でもかまいません。定足数デバイスとして使用されるディスクには、ユーザーデータを格納できます。

表 3–3 に、Sun Cluster ソフトウェアが定足数を使って split-brain や amnesia の問題を防止する方法を示します。

表 3–3　クラスタ定足数、および split-brain と amnesia の問題

問題	定足数による解決策
split brain	ノード間のクラスタインターコネクトが失われてクラスタがサブクラスタに分割され、それぞれが唯一のパーティションであると誤って認識した場合に発生する
amnesia	過半数の投票があるパーティション (サブクラスタ) だけが、クラスタとして実行できるようにする (1 つのパーティションだけが、このような過半数によって存在できる)

定足数アルゴリズムは動的に動作します。クラスタイベントによってその計算が発生した場合、計算結果はクラスタの存続期間中、変化し続けます。

定足数投票数

クラスタノードと定足数デバイスはどちらも、定足数を確立するために投票します。デフォルトにより、クラスタノードは、起動してクラスタメンバーになると、定足数投票数 (quorum vote count) を 1 つ獲得します。またノードは、たとえばノードのインストール中や管理者がノードを保守状態にしたときには、投票数は 0 になります。

定足数デバイスは、デバイスへのノード接続の数に基づいて投票数を獲得します。定足数デバイスは、設定されると、最大投票数 *N*-1 を獲得します。この場合、*N* は、投票数がゼロ以外で、定足数デバイスへのポートを持つノードの数を示します。たとえば、2 つのノードに接続された、投票数がゼロ以外の定足数デバイスの投票数は 1 (2–1) になります。

定足数デバイスは、クラスタのインストール中に構成することも、『*Sun Cluster 3.0 12/01* のシステム管理』に示す手順を使って後で構成することもできます。

注 - 定足数デバイスは、現在接続されている少なくとも 1 つのノードがクラスタメンバーである場合にのみ、投票数を獲得します。また、クラスタの起動中、定足数デバイスは、現在接続されている少なくとも 1 つのノードが起動中で、その停止時に最も最近起動されたクラスタのメンバーであった場合にのみ投票数を獲得します。

定足数の構成

定足数 (quorum) の構成は、クラスタ内のノードの数によって異なります。

- **2**ノードクラスタ – 2 ノードクラスタを形成するには、定足数投票数 (quorum vote count) として 2 票が必要です。これらの 2 つの投票数は、2 つのクラスタノード、または 1 つのノードと 1 つの定足数デバイスのどちらかによるものです。ただし、2 ノードクラスタでは、一方のノードに障害が発生したときに単一ノードで処理を続行できるように、1 つの定足数デバイスが構成されなければなりません。
- **3** ノード以上のクラスタ – ディスク格納装置へのアクセスを共有するすべてのノードペア間に定足数デバイスを指定する必要があります。たとえば、図3–3 のような 3 ノードクラスタがあるとします。この場合、ノード A とノード B が同じディスク格納装置へのアクセスを共有し、ノード B とノード C は別のディスク格納装置へのアクセスを共有します。この構成では、合計 5 つの投票数があります。3 つはノードによるもの、2 つはノード間で共有される定足数デバイスによるものです。クラスタを形成するには、過半数の投票数 (この場合は 3 票) が必要です。

  Sun Cluster ソフトウェアでは、 ディスク格納装置へのアクセスを共有するすべてのノードペア間に定足数デバイスを指定することは必要でも必須でもありません。しかし、N+1 構成が 2 ノード構成になり、両方のディスク格納装置へアクセスするノードに障害が発生すると、必要な投票数を提供します。すべてのペア間で定足数デバイスを構成すると、残りのノードはクラスタとして動作を継続することができます。

  これらの構成の例については、図 3–3 を参照してください。

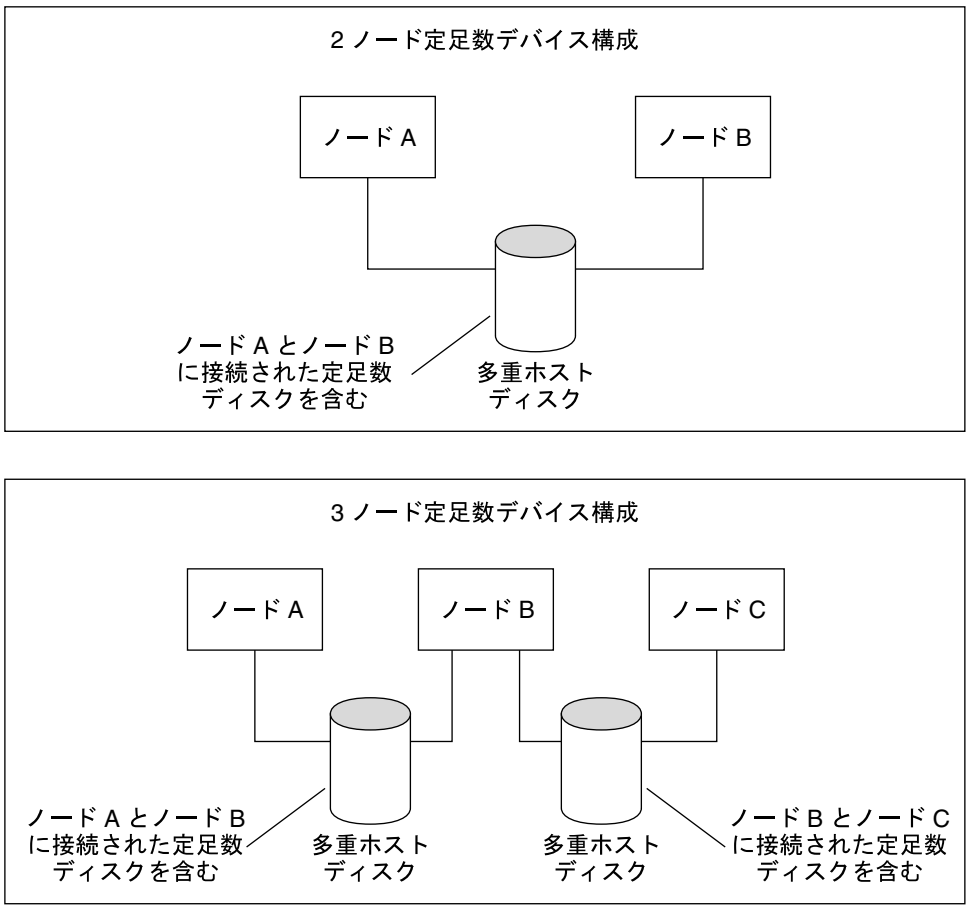

図 3–3　定足数デバイス構成の例

定足数のガイドライン

定足数デバイスを設定するときは、次のガイドラインを使用してください。

- 同じ共有ディスク格納装置に接続されたすべてのノード間で定足数デバイスを確立します。共有格納装置内に 1 つのディスクを定足数デバイスとして追加し、いずれかのノードに障害が生じたときに、他のノードが定足数を維持して、共有格納装置上のディスクデバイスグループをマスターできるようにします。
- 定足数デバイスは少なくとも 2 つのノードに接続する必要があります。
- 定足数デバイスとして、二重ポート定足数デバイスとして使用される、任意の SCSI-2 または SCSI-3 ディスクが使用できます。3 つ以上のノードに接続されたディスクは、ディスクが定足数デバイスとして使用されるかどうかに関係なく、

SCSI-3 持続的グループ予約 (PGR) をサポートしなければなりません。詳細については、『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』の計画に関する章を参照してください。

- 定足数デバイスとしてユーザーデータを含むディスクを使用できます。

ヒント - ノードの集合間で複数の定足数デバイスを構成してください。これにより、個々の定足数デバイスの障害から保護できます。異なる格納装置から複数のディスクを使用して、奇数の定足数デバイスをノードの各集合間で構成してください。

障害による影響の防止

クラスタの主要な問題は、クラスタがパーティション分割される (sprit-brain と呼ばれる) 原因となる障害です。この障害が発生すると、一部のノードが通信できなくなるため、個々のノードまたはノードの一部が、個々のクラスタまたはクラスタの一部を形成しようとします。各部分、つまりパーティションは、多重ホストディスクに対して単独のアクセスと所有権を持つものと誤って認識します。複数のノードがディスクに書き込もうとすると、データが破壊される可能性があります。

障害による影響の防止機能では、多重ホストディスクへのアクセスを物理的に防止することによって制限します。障害が発生するかパーティション分割され、ノードがクラスタから切り離されると、障害による影響の防止機能によって、ノードがディスクにアクセスできなくなります。現在のメンバーノードだけが、ディスクへのアクセス権を持つため、データの完全性が保たれます。

ディスクデバイスサービスは、多重ホストディスクを使用するサービスに対して、フェイルオーバー機能を提供します。現在、ディスクデバイスグループの主ノード (所有者) として機能しているクラスタメンバーに障害が発生するか、またはこのメンバーに到達できなくなると、新しい主ノードが選択されて、ディスクデバイスグループへのアクセスが可能になり、わずかな割り込みだけで処理が続行されます。このプロセス中、古い主ノードは、新しい主ノードが起動される前に、デバイスへのアクセスを放棄しなければなりません。ただし、あるメンバーがクラスタから切り離されて到達不能になると、クラスタはそのノードに対して、主ノードであったデバイスを解放するように通知できません。したがって、存続するメンバーが、障害の発生したメンバーから広域デバイスを制御してアクセスできるようにする手段が必要です。

SunPlex システムは、SCSI ディスク予約を使用して、障害による影響の防止機能を実装します。SCSI 予約を使用すると、障害が発生したノードは、多重ホストディスクによって阻止されて、これらのディスクへのアクセスが防止されます。

SCSI-2 ディスク予約は、ある形式の予約をサポートしています。これは、ディスクに接続されたすべてのノードへのアクセスを付与するか (予約が設定されていない場合)、または単一ノード (予約を保持するノード) へのアクセスを制限するものです。

クラスタメンバーは、別のノードがクラスタインターコネクトを介して通信していないことを検出すると、障害による影響の防止手順を開始して、そのノードが共有ディスクへアクセスするのを防止します。この障害による影響の防止機能が実行される場合、通常、阻止されるノードは、そのコンソールに「`reservation conflict`」(予約の衝突) というメッセージを表示して停止します。

予約の衝突は、ノードがクラスタメンバーではなくなったことが検出された後で、SCSI 予約がこのノードと他のノードの間で共有されるすべてのディスクに対して設定されると発生します。阻止されるノードは阻止されていることを認識しない場合があり、共有ディスクのどれかにアクセスしようとして、予約を検出して停止します。

障害の影響を防止するフェイルファースト機構

異常のあるノードが再起動され、共有ストレージに書き込むのを防ぐクラスタフレームワークの機構をフェイルファーストといいます。

クラスタのメンバーである各ノードでは、定足数ディスクを含むアクセス可能な個々のディスクに対し ioctl (MHIOCENFAILFAST) が連続的に有効にされます。この ioctl は特定のディスクドライバに対する命令です。ディスクが他のノードによって予約されているためにそのディスクにアクセスできないと、ノードは自らをパニックさせる (強制的に停止する) ことができます。

MHIOCENFAILFAST ioctl が有効になっていると、ドライバは、ノードからそのディスクに対して出されるすべての読み取りや書き込みからのエラーに、`Reservation_Conflict` エラーコードが含まれていないか検査します。ioctl はバックグラウンドでディスクに対して周期的にテスト操作を行い、`Reservation_Conflict` がないか検査します。`Reservation_Conflict` が返されると、フォアグラウンドとバックグラウンドのコントロールフローパスが両方ともパニックを発生します。

SCSI-2 ディスクの場合、予約は永続的ではないため、ノードが再起動されると無効になります。Persistent Group Reservation (PGR) の SCSI-3 ディスクでは、予約情報はそのディスクに格納されるため、ノードが再起動されても有効です。フェイルファースト機構は、SCSI-2 ディスクでも SCSI-3 ディスクでも同じように機能します。

定足数を獲得できるパーティションに属していないノードが、クラスタ内の他のノードとの接続を失うと、そのノードは別のノードによってクラスタから強制的に切り離されます。定足数を獲得できるパーティションのノードによって予約されている共有ディスクに、定足数をもたないノードからアクセスすると、ノードは予約衝突のエラーを受け取り、フェイルファースト機構に基づいてパニックを発生します。

パニックを発生したノードは、再起動を行ってクラスタに再び結合しようとするか、OpenBoot PROM (OBP) プロンプトの状態に留まることができます。どちらのアクションをとるかは、OBP の `auto-boot?` パラメータの設定に依存します。

ボリューム管理ソフトウェア

SunPlex は、ボリューム管理ソフトウェアのミラーやホットスペアディスクを使ってデータの可用性を高め、ボリューム管理ソフトウェアを使ってディスクの障害や交換に対処します。

SunPlex には独自の内部ボリューム管理ソフトウェアコンポーネントがないため、次のボリューム管理ソフトウェアが使用されます。

- Solstice DiskSuite
- VERITAS Volume Manager

クラスタ内のボリューム管理ソフトウェアは、次のものに対するサポートを提供しています。

- ノード障害のフェイルオーバー処理
- 異なるノードからの多重パスサポート
- ディスクデバイスグループへの遠隔からの透過アクセス

クラスタの制御下にあるボリューム管理オブジェクトは、ディスクデバイスグループになります。ボリューム管理ソフトウェアの詳細については、使用するボリューム管理ソフトウェアのマニュアルを参照してください。

注 - ディスクセットまたはディスクグループを計画する場合の重要な考慮事項として、その関連ディスクデバイスグループが、クラスタ内のアプリケーションリソース (データ) にどのように関連付けられているかを理解する必要があります。詳細は、『*Sun Cluster 3.0 12/01* ソフトウェアのインストール』と『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』を参照してください。

データサービス

「データサービス」という用語は、Oracle や iPlanet Web Server など、クラスタ (単一のサーバーではなく) で動作するように構成された Sun 以外のアプリケーションを意味します。データサービスは、アプリケーションや、専用の Sun Cluster 構成ファイル、および、アプリケーションの以下の操作を制御する Sun Cluster 管理メソッドからなります。

- 起動
- 停止
- 監視と訂正手段の実行

図 3–4に、単一のアプリケーションサーバーで動作するアプリケーション (単一サーバーモデル) と、クラスタで動作する同じアプリケーション (クラスタサーバーモデル) との比較を示します。ユーザーから見れば、この 2 つの構成には何の違いもありません。しかし、クラスタ化されたアプリケーションでは、処理が速くなる可能性があるだけでなく、可用性が高まります。

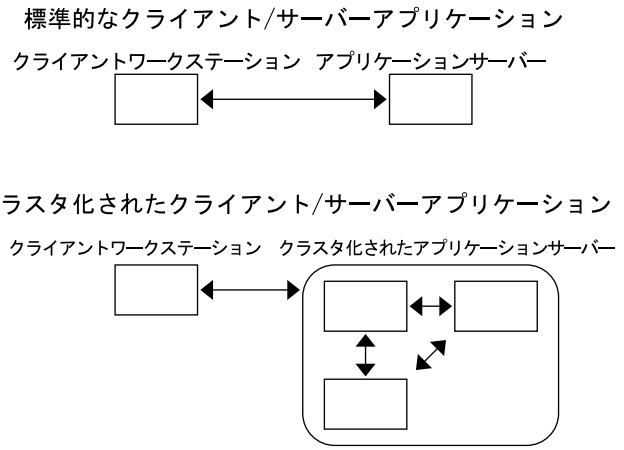

図 3–4　標準的なクライアントサーバー構成とクラスタ化されたクライアントサーバー構成

単一モデルでは、特定のパブリックネットワークインタフェース (ホスト名) を介してサーバーにアクセスするようにアプリケーションを設定します。ホスト名は、この物理サーバーに関係付けられています

クラスタサーバーモデルのパブリックネットワークインタフェースは「論理ホスト名」か「共有アドレス」です。論理ホスト名と共有アドレスを指す用語として「ネットワークリソース」が使用されます。

一部のデータサービスでは、ネットワークインタフェースとして論理ホスト名か共有アドレスのいずれか (入れ替え不可能) を指定する必要があります。しかし、別のデータサービスでは、論理ホスト名や共有アドレスをどちらでも指定することができます。どのようなタイプのインタフェースを指定する必要があるかについては、各データサービスのインストールや構成の資料を参照してください。

ネットワークリソースは、特定の物理サーバーと関連付けられているわけではありません。ネットワークリソースは、ある物理サーバーから別の物理サーバーに移すことができます。

ネットワークリソースは、当初、1 つのノード (一次ノード) に関連付けられています。しかし、一次ノードに障害が発生すると、ネットワークリソース (およびアプリケーションリソース) は、別のクラスタノード (二次ノード) にフェイルオーバーされます。ネットワークリソースがフェイルオーバーされても、アプリケーションリソースは、短時間の遅れの後に二次ノードで動作を続けます。

図 3–5 に、単一サーバーモデルとクラスタサーバーモデルとの比較を示します。クラスタサーバーモデルのネットワークリソース (この例では論理ホスト名) は、 複数のクラスタノード間を移動できます。アプリケーションは、特定のサーバーに関連付けられたホスト名として、この論理ホスト名を使用するように設定されます。

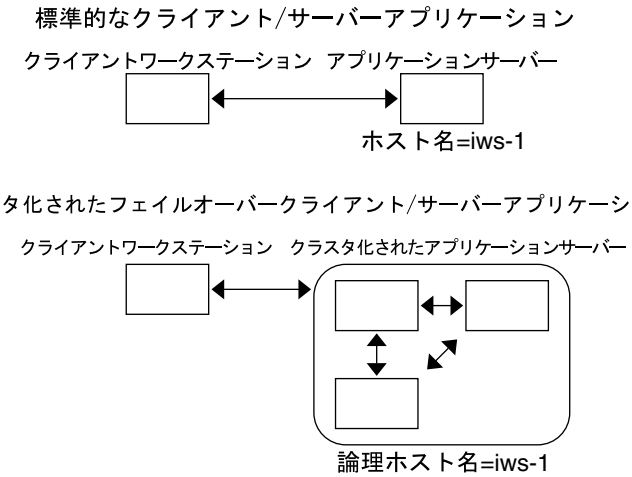

図 3–5　固定ホスト名と論理ホスト名

共有アドレスも最初は 1 つのノードに関連付けられています。このノードを広域インタフェースノード (GIN) といいます。共有アドレスは、クラスタへの唯一のネットワークインタフェースとして使用されます。これを「広域インタフェース」といいます。

論理ホスト名モデルとスケーラブルサービスモデルの違いは、スケーラブルサービスモデルでは、各ノードのループバックインタフェースにも共有アドレスがアクティブに設定される点です。この設定では、データサービスの複数のインスタンスをいくつかのノードで同時にアクティブにすることができます。「スケーラブルサービス」という用語は、クラスタノードを追加してアプリケーションの CPU パワーを強化すれば、性能が向上することを意味します。

GIN に障害が発生した場合には、共有アドレスを、同じアプリケーションのインスタンスが動作している別のノードに移すことができます (これによって、このノードが新しい GIN になる)。または、共有アドレスを、このアプリケーションを実行していない別のクラスタノードにフェイルオーバーすることができます。

図 3–6 に、単一サーバー構成とクラスタ化されたスケーラブルサービス構成との比較を示します。スケーラブルサービス構成では、共有アドレスがすべてのノードに設定されています。フェイルオーバーデータサービスに論理ホスト名が使用される場合と同じように、アプリケーションは、特定のサーバーに関連付けられたホスト名の代わりにこの共有アドレスを使用するように設定されます。

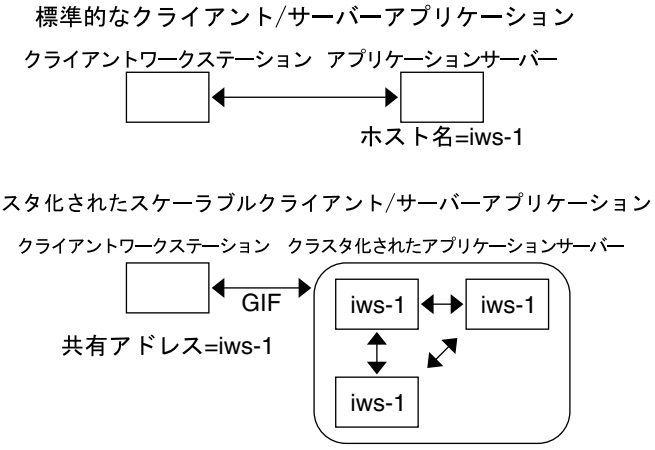

図 3–6　固定ホスト名と共有アドレス

データサービスメソッド

Sun Cluster ソフトウェアでは、Resource Group Manager (RGM) の制御下で動作する一連のサービス管理メソッドが提供されます。RGMは、これらのメソッドを使って、クラスタノードで動作するアプリケーションの起動や停止、監視を行います。これらのメソッドとクラスタフレームワークソフトウェアおよび多重ホストディス

クにより、アプリケーションは、フェイルオーバーデータサービスやスケーラブルデータサービスとして機能します。

さらに、RGM は、アプリケーションのインスタンスやネットワークリソース (論理ホスト名と共有アドレス) といったクラスタのリソースを管理します。

Sun Cluster ソフトウェアが提供するメソッドの他に、SunPlex システムからも API やいくつかのデータサービス開発ツールが提供されます。これらのツールを使用すれば、アプリケーションプログラマは、独自のデータサービスメソッドを開発することによって、他のアプリケーションを Sun Cluster ソフトウェアの下で高可用性データサービスとして実行できます。

リソースグループマネージャー (RGM)

RGM は、データサービス (アプリケーション) を、リソースタイプの実装によって管理されるリソースとして制御します。これらの実装は、Sun によって提供されるか、開発者によって、汎用データサービステンプレート、DSDL API (データサービス開発ライブラリ API)、RMAPI (リソース管理 API) のどれかを使用して作成されます。クラスタ管理者は、リソースグループと呼ばれるコンテナにリソースを作成して管理します。RGM は、クラスタメンバーシップの変更に応じて、指定ノードのリソースグループを停止および開始します。

RGM は「リソース」と「リソースグループ」に作用します。リソースやリソースグループは、RGM のアクションに従ってオンラインになったり、オフラインになったりします。リソースやリソースグループに適用される状態や設定値の詳細は、74ページの「リソースおよびリソースグループの状態と設定値」を参照してください。

フェイルオーバーデータサービス

データサービスが実行されているノード (主ノード) に障害が発生すると、サービスは、ユーザーによる介入なしで別の作業ノードに移行します。フェイルオーバーサービスは、アプリケーションインスタンスリソースとネットワークリソース (論理ホスト名) のコンテナである、フェイルオーバーリソースグループを使用します。論理ホスト名とは、1 つのノードに構成して、後で自動的に元のノードや別のノードに構成できる IP アドレスのことです。

フェイルオーバーデータサービスでは、アプリケーションインスタンスは単一ノードでのみ実行されます。フォルトモニターは、エラーを検出すると、データサービスの構成に従って、同じノードでそのインスタンスを再起動しようとするか、別のノードでそのインスタンスを起動 (フェイルオーバー) しようとします。

スケーラブルデータサービス

スケーラブルデータサービスは、複数ノードのアクティブインスタンスに対して効果があります。スケーラブルサービスは、2 つのリソースグループを使用します。スケーラブルリソースグループを使用してアプリケーションリソースを含め、フェイルオーバーリソースグループを使用してスケーラブルサービスが依存するネットワークリソース (共有アドレス) を含めます。スケーラブルリソースグループは、複数のノードでオンラインにできるため、サービスの複数のインスタンスを一度に実行できます。共有アドレスのホストとなるフェイルオーバーリソースグループは、一度に 1 つのノードでしかオンラインにできません。スケーラブルサービスのホストとなるすべてのノードは、同じ共有アドレスを使用してサービスに対するホストとなります。

サービス要求は、単一ネットワークインタフェース (広域インタフェース) を介してクラスタに入り、負荷均衡ポリシーによって設定されたいくつかの定義済みアルゴリズムの 1 つに基づいてノードに分配されます。クラスタは、負荷均衡ポリシーを使用し、いくつかのノード間でサービス負荷の均衡をとることができます。他の共有アドレスをホストしている別のノード上に、複数の広域インタフェースが存在する可能性があります。

スケーラブルサービスの場合、アプリケーションインスタンスはいくつかのノードで同時に実行されます。広域インタフェースのホストとなるノードに障害が発生すると、広域インタフェースは別のノードで処理を続行します。アプリケーションインスタンスの実行に失敗した場合、そのインスタンスは同じノードで再起動しようとします。

アプリケーションインスタンスを同じノードで再起動できず、別の未使用のノードがサービスを実行するように構成されている場合、サービスはその未使用ノードで処理を続行します。あるいは、残りのノードで実行し続けて、サービススループットを低下させることになります。

注 - 各アプリケーションインスタンスの TCP 状態は、広域インタフェースノードではなく、インスタンスを持つノードで維持されます。したがって、広域インタフェースノードに障害が発生しても接続には影響しません。

図 3–7 は、フェイルオーバーリソースグループとスケーラブルリソースグループの例と、スケーラブルサービスにとってそれらの間にどのような依存関係があるのかを示しています。この例は、3 つのリソースグループを示しています。フェイルオーバーリソースグループには、可用性の高い DNS のアプリケーションリソースと、可用性の高い DNS および可用性の高い Apache Web Server の両方によって

使用されるネットワークリソースが含まれます。スケーラブルリソースグループには、Apache Web Server のアプリケーションインスタンスだけが含まれます。リソースグループの依存関係は、スケーラブルリソースグループとフェイルオーバーリソースグループの間に存在し (実線)、Apache アプリケーションリソースはすべて、共有アドレスであるネットワークリソース `schost-2` に依存する (破線) ことに注意してください。

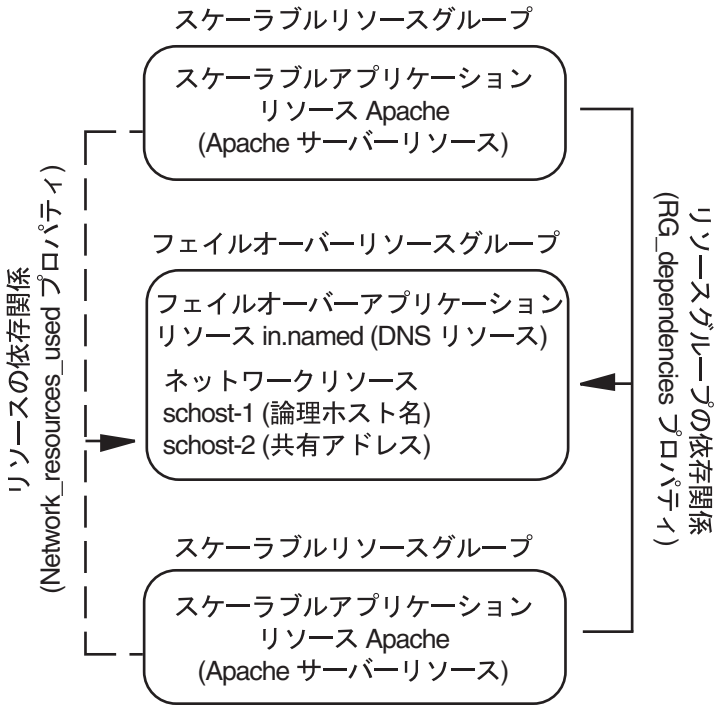

図 3–7　フェイルオーバーリソースグループとスケーラブルリソースグループの例

スケーラブルサービスの構造

クラスタネットワーキングの主な目的は、データサービスにスケーラビリティを提供することにあります。スケーラビリティとは、サービスに提供される負荷が増えたときに、新しいノードがクラスタに追加されて新しいサーバーインスタンスが実行されるために、データサービスがこの増加した負荷に対して一定の応答時間を維持できるということを示します。このようなサービスをスケーラブルデータサービスと呼びます。スケーラブルデータサービスの例としては、Web サービスがあります。通常、スケーラブルデータサービスはいくつかのインスタンスからなり、それぞれがクラスタの異なるノードで実行されます。これらのインスタンスはともに、

そのサービスの遠隔クライアントの観点から見て 1 つのサービスとして動作し、サービスの機能を実現します。たとえば、いくつかのノードで実行されるいくつかの `httpd` デーモンからなるスケーラブル Web サービスがあります。どの `httpd` デーモンもクライアント要求に対応できます。要求に対応するデーモンは、負荷均衡ポリシーによって決められます。クライアントへの応答は、その要求にサービスを提供する特定のデーモンではなく、サービスによるもののようにみえるため、単一サービスの外観が維持されます。

スケーラブルサービスは、次のものからなります。

- スケーラブルサービスに対するネットワークインフラストラクチャのサポート
- 負荷均衡
- ネットワーキングおよびデータサービスに対するサポート (リソースグループマネージャーを使用)

次の図は、スケーラブルサービスの構造を示しています。

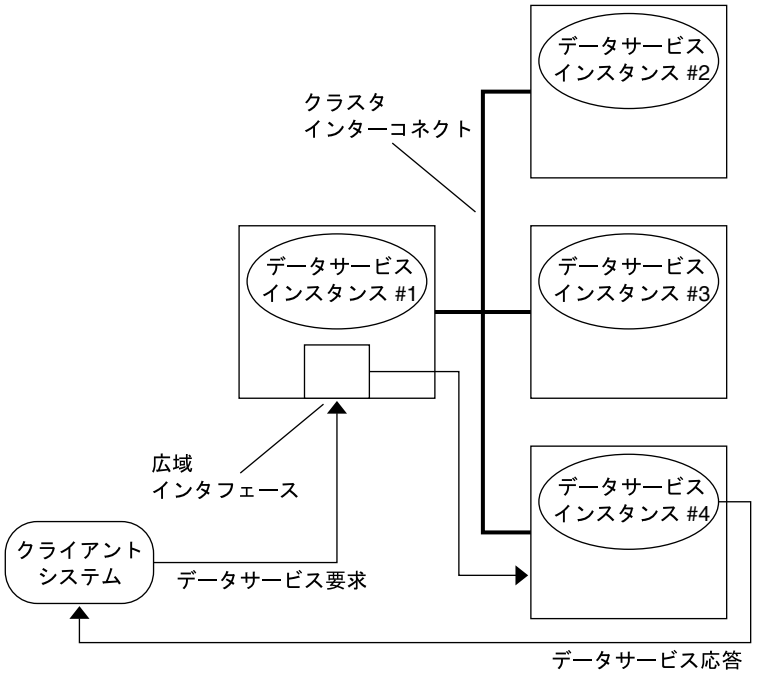

図 3–8　スケーラブルサービスの構造

広域インタフェースのホストではないノード (プロキシノード) には、そのループバックインタフェースでホストされる共有アドレスがあります。広域インタフェー

スに入るパケットは、構成可能な負荷均衡ポリシーに基づいて、他のクラスタノードに分配されます。次に、構成できる負荷均衡ポリシーについて説明します。

負荷均衡ポリシー

負荷均衡は、スケーラブルサービスのパフォーマンスを応答時間とスループットの両方の点で向上させます。

スケーラブルデータサービスには、*pure* と *sticky* の 2 つのクラスがあります。 pure サービスとは、そのいずれかのインスタンスがクライアント要求に応答できるサービスをいいます。sticky サービスとは、クライアントが同じインスタンスに要求を送るサービスをいいます。これらの要求は、別のインスタンスには変更されません。

pure サービスは、ウェイト設定した (weighted) 負荷均衡ポリシーを使用します。この負荷均衡ポリシーのもとでは、クライアント要求は、デフォルトで、クラスタ内のサーバーインスタンスに一律に分配されます。たとえば、3 ノードクラスタでは、各ノードに 1 のウェイトがあるものと想定します。各ノードは、そのサービスに代わって、クライアントからの要求の 3 分の 1 のサービスを提供します。管理者は、`scrgadm(1M)` コマンドインタフェースか SunPlex Manager GUI を使って、ウェイトをいつでも変更できます。

sticky サービスには、ordinary sticky と wildcard sticky の 2 種類があります。sticky サービスを使用すると、内部状態メモリーを共有でき (アプリケーションセッション状態)、複数の TCP 接続でアプリケーションレベルの同時セッションが可能です。

ordinary sticky サービスを使用すると、クライアントは、複数の同時 TCP 接続で状態を共有できます。単一ポートを待機しているそのサーバーインスタンスという点で、そのクライアントは *sticky* であると呼ばれます。クライアントは、インスタンスが起動していてアクセス可能であり、負荷分散ポリシーがサーバーのオンライン時に変更されていなければ、すべての要求が同じサーバーのインスタンスに送られることを保証されます。

たとえば、クライアント上の Web ブラウザは、3 つの異なる TCP 接続を使用して、ポート 80 にある 共有 IP アドレスに接続しますが、これらの接続はサービスでキャッシュされたセッション情報を交換します。

sticky ポリシーを一般化すると、同じインスタンスの背後でセッション情報を交換する複数のスケーラブルサービスにまで及びます。これらのサービスが同じインスタンスの背後でセッション情報を交換する場合、同じノードで異なるポートと通信する複数のサーバーインスタンスという点で、そのクライアントは *sticky* であると呼ばれます。

たとえば、電子商取引サイトの顧客は、ポート 80 の HTTP を使用して買い物をしますが、購入した製品の支払いをクレジットカードで行うためには、ポート 443 で SSL に切り替えて機密データを送ります。

wildcard sticky サービスは、動的に割り当てられたポート番号を使用しますが、クライアント要求がやはり同じノードに送られるものと想定します。クライアントは、同じ IP アドレスという点で、ポートに対して *sticky wildcard* です。

このポリシーの例としては、受動モード FTP があります。クライアントは、ポート 21 の FTP サーバーに接続して、動的ポート範囲のリスナーポートサーバーに接続するよう、そのサーバーによって通知されます。この IP アドレスに対する要求はすべて、サーバーが制御情報によってクライアントに通知した、同じノードに転送されます。

これらの各 sticky ポリシーでは、ウェイト設定した (weighted) 負荷均衡ポリシーがデフォルトで有効であるため、クライアントの最初の要求は、負荷均衡によって指定されたインスタンスにリダイレクトされます。インスタンスが実行されているノードとクライアントが関係を確立すると、そのノードがアクセス可能で、負荷分散ポリシーが変更されない限り、今後の要求はそのインスタンスに送られます。

次に、各負荷均衡ポリシーの詳細について説明します。

- weighted - 負荷は指定されたウェイト値に従って各種のノードに分配されます。このポリシーは `Load_balancing_weights` プロパティに設定された `LB_WEIGHTED` の値を使用して設定されます。ウェイトがノードについて明示的に設定されていない場合は、デフォルトで 1 が設定されます。

  このポリシーは、ラウンドロビンではないことに注意してください。ラウンドロビンポリシーでは、クライアントからの各要求が、最初の要求はノード 1、2 番目の要求はノード 2 といったように常に異なるノードに送られます。 ウェイトを設定したポリシーは、一定の割合のクライアントトラフィックが、特定ノードに送られるようにするものです。このポリシーは、個々の要求には対応しません。

- sticky - このポリシーでは、ポートの集合が、アプリケーションリソースの構成時に認識されます。 このポリシーは、`Load_balancing_policy` リソースプロパティの `LB_STICKY` の値を使用して設定されます。

- sticky-wild - このポリシーは、通常の “sticky” ポリシーの上位セットです。 IP アドレスによって識別されるスケーラブルサービスでは、ポートはサーバーによって割り当てられます (したがって前もって認識されません)。ポートは変更されることがあります。このポリシーは、`Load_balancing_policy` リソースプロパティの `LB_STICKY_WILD` の値を使用して設定されます。

フェイルバック設定

リソースグループは、ノードからノードへ処理を継続します。このようなリソースグループの移行が起こると、それまでの二次ノードが新しい主ノードになります。元の主ノードがオンラインに復帰したときにどのようなアクションを取るか (つまり、元の主ノードを再び主ノードに戻す (フェイルバックする) 、現在の主ノードをそのまま継続するか) は、フェイルバックの設定値で決まります。この選択は、リソースグループのプロパティ `Failback` で設定します。

特定のインスタンスでは、リソースグループをホストする元のノードに障害が発生して繰り返し再起動する場合、フェイルバックを設定すると、リソースグループの可用性が低下することがあります。

データサービス障害モニター

SunPlex の各データサービスには、データサービスを定期的に探索してその状態を判断する障害モニターがあります。障害モニターは、アプリケーションデーモンが実行されていて、クライアントにサービスが提供されていることを確認します。探索によって得られた情報をもとに、デーモンの再起動やフェイルオーバーの実行などの事前に定義された処置が開始されます。

新しいデータサービスの開発

サンが提供する構成ファイルや管理メソッドのテンプレートを使用することで、さまざまなアプリケーションをクラスタ内でフェイルオーバーサービスやスケーラブルサービスとして実行できます。フェイルオーバーサービスやスケーラブルサービスとして実行するアプリケーションがサンから提供されていない場合は、API や DSET API を使用して、フェイルオーバーサービスやスケーラブルサービスとして動作するようにアプリケーションを設定できます。

アプリケーションがフェイルオーバーサービスを使用できるかどうかを判断するための基準があります。個々の基準は、アプリケーションで使用できる API について説明した SunPlex のマニュアルに記載されています。

.次に、各自のサービスがスケーラブルデータサービスの構造を利用できるかどうかを知るために役立つガイドラインをいくつか示します。スケーラブルサービスの一般的な情報については、64ページの「スケーラブルデータサービス」を参照してください。

次のガイドラインを満たす新しいサービスは、スケーラブルサービスを利用できます。既存のサービスがこれらのガイドラインに従っていない場合は、そのサービスがガイドラインに準拠するように、一部を書き直さなければならない場合があります。

スケーラブルデータサービスには、以下の特性があります。まず、このようなサービスは、1 つまたは複数のサーバーインスタンスからなります。各インスタンスは、クラスタの異なるノードで実行されます。同じサービスの複数のインスタンスを、同じノードで実行することはできません。

次に、サービスが外部論理データ格納を使用する場合は、この格納に対する複数のサーバーインスタンスからの同時アクセスの同期をとって、更新が失われたり、変更中のデータを読み取ったりすることを避ける必要があります。この格納をメモリー内部の状態と区別するために「外部」と呼び、格納がそれ自体複製されている場合でも単一の実体として見えるため、「論理的」と呼んでいることに注意してください。また、この論理データ格納には、サーバーインスタンスが格納を更新するたびに、その更新がすぐに他のインスタンスで見られるという特性があります。

SunPlex システムは、このような外部記憶領域をそのクラスタファイルシステムと広域 raw パーティションを介して提供します。例として、サービスが外部ログファイルに新しいデータを書き込む場合や既存のデータを修正する場合を想定してください。このサービスの複数インスタンスが実行されている場合は、それぞれがこの外部ログへのアクセスを持ち、同時にこのログにアクセスできます。各インスタンスは、このログに対するアクセスの同期をとる必要があります。そうしないと、インスタンスは相互に妨害しあうことになります。サービスは、`fcntl(2)` および `lockf(3C)` によって通常の Solaris ファイルロックを使用して、必要な同期をとることができます。

このような格納のもう 1 つの例としては、可用性の高い Oracle や Oracle Parallel Server などのバックエンドデータベースが挙げられます。このようなバックエンドデータベースサーバーは、データベース照会または更新トランザクションを使用するのに内部組み込みの同期を使用するため、複数のサーバーインスタンスが独自の同期を実装する必要がありません。

現在の実現状態ではスケーラブルサービスではないサービスの例として、Sun の IMAP サーバーがあります。このサービスは格納を更新しますが、その格納はプライベートであり、複数の IMAP インスタンスがこの格納に書き込むと、更新の同期がとられないために相互に上書きし合うことになります。IMAP サーバーは、同時アクセスの同期をとるために書き直す必要があります。

最後に、インスタンスは、他のインスタンスのデータから切り離されたプライベートデータを持つ場合があることに注意してください。このようなケースでは、デー

タはプライベートであり、そのインスタンスだけがデータを処理するため、サービスは同時アクセスの同期をとる必要はありません。この場合、このプライベートデータが広域にアクセス可能になる可能性があるため、このデータをクラスタファイルシステムのもとで保存しないように注意する必要があります。

データサービス API と DSDL API

SunPlex システムには、アプリケーションの可用性を高めるものとして次の機能があります。

- SunPlex システムの一部として提供されるデータサービス
- データサービス API
- DSDL API (データサービス開発ライブラリ API)
- 汎用データサービス

『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』は、SunPlex システムで提供されるデータサービスをインストール、構成する方法を説明しています。『*Sun Cluster 3.0 12/01* データサービス開発ガイド』には、Sun Cluster フレームワークの下でその他のアプリケーションの可用性を高めるにはどうすればよいかが説明されています。

アプリケーションプログラマは、Sun Cluster API を使用することによって、データサービスインスタンスの起動や停止を行なう障害モニターやスクリプトを開発できます。これらのツールを使用すると、アプリケーションをフェイルオーバーまたはスケーラブルデータサービスとして設計できます。さらに、SunPlex システムの「汎用」データサービスを使用すれば、アプリケーションをフェイルオーバーサービスかスケーラブルサービスとして実行するための起動メソッドや停止メソッドを簡単に生成できます。

クラスタインターコネクトによるデータサービストラフィックの送受信

クラスタには、ノード間を結ぶ複数のネットワーク接続が必要です。クラスタインターコネクトは、これらの接続から構成されています。クラスタ化ソフトウェアは、可用性や性能を高めるために複数のインターコネクトを使用します。ファイルシステムのデータやスケーラブルサービスのデータなどの内部トラフィックでは、メッセージが、すべての使用可能なインターコネクト上にラウンドロビン方式でストライプ化されます。

クラスタインターコネクトは、ノード間の通信の可用性を高めるためにアプリケーションから使用することもできます。たとえば、分散アプリケーションでは、個々のコンポーネントが異なるノードで動作することがあり、その場合には、ノード間の通信が必要になります。パブリック伝送の代わりにクラスタインターコネクトを使用することで、個別のリンクに障害が発生しても、接続を続けることができます。

ノード間の通信にクラスタインターコネクトを使用するには、クラスタのインストール時に設定したプライベートホスト名をアプリケーションで使用する必要があります。たとえば、ノード 1 のプライベートホスト名が `clusternode1_priv` である場合、クラスタインターコネクトを経由してノード 1 と通信するときにこの名前を使用する必要があります。この名前を使用して開かれた TCP ソケットは、クラスタインターコネクトを経由するように経路指定され、ネットワークに障害が発生した場合でも、この TCP ソケットは透過的に再度経路指定されます。

複数のプライベートホスト名がインストール時に設定されているため、クラスタインターコネクトでは、その時点で選択した任意の名前を使用できます。実際の名前は、`scha_cluster_get`(3HA) に `scha_privatelink_hostname_node` 引数を指定することによって取得できます。

クラスタインターコネクトをアプリケーションレベルで使用する場合には、個々のノードペア間の通信に 1 つのインターコネクトが使用されます。ただし、可能であれば、別のノードペアには別のインターコネクトが使用されます。たとえば、3 つのノードで動作するアプリケーションが、クラスタインターコネクト経由で通信しているとします。この場合、たとえば、ノード 1 とノード 2 の通信にはインタフェース `hme0` が、ノード 1 とノード 3 の通信にはインタフェース `qfe1` がそれぞれ使用されます。つまり、アプリケーションによる 2 ノード間の通信は 1 つのインターコネクトに制限されますが、内部のクラスタ通信はすべてのインターコネクト上にストライプ化されます。

インターコネクトはアプリケーションと内部のクラスタトラフィックによって共有されるため、アプリケーションから使用できる帯域幅の量は、他のクラスタトラフィックに使用される帯域幅の量に左右されます。インターコネクトに障害が発生すると、内部トラフィックは残りのインターコネクト上にラウンドロビン方式で分散されますが、障害が発生したインターコネクト上のアプリケーションは動作しているインターコネクトに切り替えられます。

クラスタインターコネクトでは、2 つのタイプのアドレスがサポートされます。さらに、プライベートホスト名に対する `gethgethostbyname`(3N) では、通常 2 つの IP アドレスが返されます。最初のアドレスを「論理 pairwise アドレス」と呼び、2 番目のアドレスを「論理 pernode アドレス」と呼びます。

個々のノードペアには、異なる論理 pairwise アドレスが割り当てられます。この小規模な論理ネットワークでは、接続のフェイルオーバーがサポートされます。さらに、各ノードには、固定した pernode アドレスが割り当てられます。つまり、clusternode1-priv の論理 pairwise アドレスはノードごとに異なりますが、clusternode1-priv の論理 pernode アドレスは各ノードで同じです。ただし、個々のノードが pairwise アドレスを自分で持っているわけではないため、ノード 1 に gethostbyname(clusternode1-priv) を実行しても、論理 pernode アドレスだけが返されます

アプリケーションがクラスタインターコネクト経由の接続を受け入れ、セキュリティの目的で IP アドレスを検査する場合には、gethostbyname から返される最初の IP アドレスだけでなく、すべての IP アドレスを検査する必要があります。

アプリケーション全体にわたって一貫した IP アドレスが必要な場合は、クライアント側でもサーバー側でもその pernode アドレスにバインドするようにアプリケーションを設定します。これによって、すべての接続にこの pernode アドレスが使用されます。

リソース、リソースグループ、リソースタイプ

データサービスは、数種類のリソースを使用します。たとえば、Apache Web Server や iPlanet Web Server などのアプリケーションは、それらのアプリケーションが依存するネットワークアドレス (論理ホスト名と共有アドレス) を使用します。アプリケーションとネットワークリソースは、RGM によって管理される基本ユニットを形成します。

データサービスはリソースタイプです。たとえば、Sun Cluster HA for Oracle はリソースタイプ SUNW.oracle であり、Sun Cluster HA for Apache はリソースタイプ SUNW.apache です。

リソースとは、クラスタ全体で定義されたリソースタイプをインスタンス化したものです。いくつかのリソースタイプが定義されています。

ネットワークリソースは、SUNW.LogicalHostname または SUNW.SharedAddress リソースタイプです。これら 2 つのリソースタイプは、Sun Cluster ソフトウェア製品によって事前に登録されています。

SUNW.HAStorage リソースタイプは、リソースとそのリソースが依存するディスクデバイスグループの起動を同期させるのに使用します。これによって、データサービスが起動する前に、クラスタファイルシステムのマウントポイントへのパス、広域デバイス、デバイスグループ名を使用できます。

RGM 管理のリソースは、リソースグループと呼ばれるグループに入れられるため、1 つのユニットとして管理できます。リソースグループは、フェイルオーバーまたはスイッチオーバーがリソースグループで開始されたときに、ユニットで移行されます。

注 - アプリケーションリソースを含むリソースグループをオンラインにすると、アプリケーションが起動します。データサービス起動方式は、アプリケーションが起動して実行されるまで待ってから、正常に終了します。アプリケーションが起動して実行しているかどうかの判断は、データサービス障害モニターが、データサービスがクライアントにサービスを提供しているかどうかを判断するのと同じ方法で行われます。このプロセスの詳細については、『*SunCluster 3.0 12/01* データサービスのインストールと構成』を参照してください。

リソースおよびリソースグループの状態と設定値

リソースやリソースグループの値は管理者によって静的に設定されます。これらの設定値を変更するには管理上の作業が必要です。RGM では、リソースグループの「状態」が動的に変更されます。次に、このような設定値と状態について説明します。

- managed (管理) または unmanaged (非管理) - クラスタ全体に適用されるこの設定値は、リソースグループだけに使用されます。リソースグループは RGM によって管理されます。リソースグループを RGM によって管理または非管理にするには、`scrgadm(1M)` コマンドを使用します。これらの設定値は、クラスタ再構成では変更されません。

  新たに作成したリソースグループの状態は非管理になっています。このグループのいずれかのリソースをアクティブにするためには、リソースグループの状態が管理になっていなければなりません。

  スケーラブル Web サーバーなど、ある種のデータサービスでは、ネットワークリソースの起動前や停止後に、あるアクションが必要です。このアクションには、initialization (INIT) と finish (FINI) データサービスメソッドを使用します。INIT メソッドが動作するためには、リソースが置かれているリソースグループが管理状態になっていなければなりません。

  リソースグループを非管理から管理の状態に変更すると、そのグループに対して登録されている INIT メソッドがグループの各リソースに対して実行されます。

  リソースグループを管理から非管理の状態に変更すると、登録されている FINI メソッドが呼び出され、クリーンアップが行われます。

INIT や FINI メソッドは、一般にスケーラブルサービスのネットワークリソースに対して使用されますが、これらのメソッドは、アプリケーションによって行われない任意の初期設定やクリーンアップにも使用できます。

- enabled (有効) または disabled (無効) - クラスタ全体に適用されるこの設定値は、リソースだけに使用されます。リソースを有効または無効にするには、`scrgadm(1M)` コマンドを使用します。これらの設定値は、クラスタ再構成では変更されません。

  リソースの通常の設定では、リソースは有効にされ、システムでアクティブに動作しています。

  何らかの理由であるリソースをすべてのクラスタノードで使用不能にする場合は、そのリソースを無効にします。無効にされたリソースは、一般的な使用には提供されません。

- online (オンライン) または offline (オフライン) – 動的に変更可能なこの状態は、リソースとリソースグループに適用されます。

  これらの状態は、スイッチオーバーやフェイルオーバーで行われるクラスタ再構成手順でクラスタの状態が変わるのに伴って変化します。さらに、これらの状態は、管理アクションによって変更することもできます。リソースやリソースグループをオンラインまたはオフライン状態に変更するには、`scswitch(1M)` を使用します。

  フェイルオーバーリソースまたはリソースグループは、オンラインにできるのはどの時点でも 1 つのノードだけです。スケーラブルリソースまたはリソースグループは、あるノードではオンラインにし、他のノードではオフラインにすることができます。スイッチオーバーやフェイルオーバーでは、リソースグループやリソースグループに属するリソースは、あるノードでオフラインにされてから、別のノードでオンラインにされます。

  あるリソースグループがオフラインであるなら、そのすべてのリソースもオフラインです。あるリソースグループがオンラインであるなら、有効にされているそのすべてのリソースもオンラインです。

  リソースグループにはいくつかのリソースを持つことができますが、リソース間には相互依存関係があります。したがって、これらのリソースをオンラインまたはオフラインにするときには、特定の順序で行う必要があります。リソースをオンラインまたはオフラインにするためにメソッドが必要とする時間は、リソースによって異なります。リソースの相互依存関係と起動や停止時間の違いにより、クラスタの再構成では、同じリソースグループのリソースでもオンラインやオフラインの状態が異なることがあります。

リソースとリソースグループプロパティ

SunPlex データサービスのリソースやリソースグループのプロパティ値は構成できます。標準的なプロパティはすべてのデータサービスに共通です。拡張プロパティは各データサービスに特定のものです。標準プロパティおよび拡張プロパティのいくつかは、デフォルト設定によって構成されているため、これらを修正する必要はありません。それ以外のプロパティは、リソースを作成して構成するプロセスの一部として設定する必要があります。各データサービスのマニュアルでは、設定できるリソースプロパティの種類とその設定方法を指定しています。

標準プロパティは、通常特定のデータサービスに依存しないリソースおよびリソースグループプロパティを構成するために使用されます。これらの標準プロパティについては、『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』の付録を参照してください。

拡張プロパティは、アプリケーションバイナリの場所や構成ファイルなどの情報を提供するものです。拡張プロパティは、データサービスの構成に従って 修正する必要があります。拡張プロパティについては、『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』のデータサービスに関する各章を参照してください。

パブリックネットワーク管理 (PNM) とネットワークアダプタフェイルオーバー (NAFO)

クライアントは、パブリックネットワークを介してクラスタにデータ要求を行います。各クラスタノードは、パブリックネットワークアダプタを介して少なくとも 1 つのパブリックネットワークに接続されています。

Sun Cluster パブリックネットワーク管理 (PNM) ソフトウェアは、パブリックネットワークアダプタを監視し、障害の検出時に IP アドレスをアダプタからアダプタへフェイルオーバーするための基本機構を備えています。各クラスタノードには独自の PNM 構成があり、これは他のクラスタノードと異なることもあります。

パブリックネットワークアダプタは、ネットワークアダプタフェイルオーバーグループ (NAFO グループ) の中に構成されています。各 NAFO グループには、1 つまたは複数のパブリックネットワークアダプタがあります。特定の NAFO グループで一度にアクティブにできるのは 1 つのアダプタだけですが、同じグループ内の多くのアダプタは、アクティブアダプタの PNM デーモンによって障害が検出された場合のアダプタフェイルオーバーに使用されるバックアップアダプタとして機能します。フェイルオーバーでは、アクティブアダプタに関連するIP アドレスがバックアップアダプタに移動するため、ノードのパブリックネットワーク接続が維持さ

れます。フェイルオーバーは、アダプタインタフェースレベルで発生するため、フェイルオーバー中に一時的にわずかな遅延がみられる以外、TCP などのこれよりも高いレベルの接続は影響されません。

注 - TCP の構成回復特性が原因で、正常なフェイルオーバーの後、セグメントのいくつかがフェイルオーバー中に失われて、TCP の混雑制御機構をアクティブ化するために、TCP エンドポイントではさらに遅延が生じる可能性があります。

NAFO グループには、論理ホスト名と共有アドレスリソースの構築ブロックがあります。`scrgadm(1M)` コマンドは、必要に応じて NAF グループを自動的に作成します。論理ホスト名と共有アドレスリソースとは別に NAFO グループを作成して、クラスタノードのパブリックネットワーク接続を監視する必要もあります。ノード上の同じ NAFO グループが、任意の数の論理ホスト名、または共有アドレスリソースのホストとなることができます。論理ホスト名と共有アドレスリソースの詳細については、『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』を参照してください。

注 - NAFO 機構の設計は、アダプタの障害を検出してマスクすることを目的としています。この設計は、`ifconfig(1M)` を使用して論理 (または共有) IP アドレスのどれかを削除した状態から回復することを目的としていません。Sun Cluster ソフトウェアは、論理アドレスや共有 IP アドレスを RGM によって管理されるリソースとみなします。管理者が IP アドレスを追加または削除する正しい方法は、`scrgadm(1M)` を使用してリソースを含むリソースグループを修正するというものです。

PNM 障害検出とフェイルオーバープロセス

PNM は、正常なアダプタのパケットカウンタが、アダプタを介した通常のネットワークトラフィックによって変化するものと想定して、アクティブアダプタのパケットカウンタを定期的にチェックします。パケットカウンタがしばらくの間変化しない場合、PNM は `ping` シーケンスに入って、トラフィックを強制的にアクティブアダプタに送ります。PNM は、各シーケンスの最後でパケットカウンタに変化がないかを検査し、ping シーケンスが何度か繰り返された後でもパケットカウンタに変化がない場合は、アダプタの障害を宣言します。これらのイベントは、バックアップアダプタがあれば、それへのフェイルオーバーを引き起こします。

入力および出力パケットカウンタはいずれも PNM によって監視され、どちらかまたは両方にしばらくの間変化がない場合は、ping シーケンスが開始されます。

ping シーケンスは、ALL_ROUTER マルチキャストアドレス (224.0.0.2)、ALL_HOST マルチキャストアドレス (224.0.0.1)、およびローカルサブネットブロードキャストアドレスの ping からなります。

ping は、コストの低いもの順という方法で構成されているため、コストのかかる ping は、コストの低い ping が成功した場合は実行されません。また、ping はアダプタでのトラフィックを生成するための方法としてのみ使用されます。その終了状態は、アダプタが機能しているか障害が発生しているかの判断には関係ありません。

このアルゴリズムには、inactive_time、ping_timeout、repeat_test、slow_network という 4 つの調整可能なパラメータがあります。これらのパラメータによって、障害検出の速度と正確さを調整できます。パラメータの詳細とその変更方法については、『*Sun Cluster 3.0 12/01* のシステム管理』のパブリックネットワークパラメータの変更手順を参照してください。

NAFO グループのアクティブアダプタで障害が検出された後で、バックアップアダプタが使用できない場合、そのグループは停止と宣言されますが、そのバックアップアダプタのすべてのテストは続行します。また、バックアップアダプタが使用可能な場合は、そのバックアップアダプタに対してフェイルオーバーが発生します。論理アドレスとその関連フラグはバックアップアダプタに転送されて、障害の発生したアクティブアダプタは停止して切り離された状態に (unplumbed) なります。

IP アドレスのフェイルオーバーが正常に終了すると、自動的に ARP が送信されます。したがって、遠隔クライアントへの接続は維持されます。

動的再構成のサポート

Sun Cluster 3.0 による動的再構成 (DR: Dynamic Reconfiguration) ソフトウェア機能のサポートは段階的に開発されています。この節では、Sun Cluster 3.0 12/01 による DR 機能のサポートの概念と考慮事項について説明します。

Solaris 8 の DR 機能の説明で述べられているすべての必要条件、手順、制限は Sun Cluster の DR サポートにも適用されます (オペレーティング環境の休止操作を除く)。したがって、Sun Cluster ソフトウェアで DR 機能を使用する前に、必ず、Solaris 8 の DR 機能についての説明を参照してください。特に、DR Detach 操作中に、ネットワークに接続されていない入出力デバイスに影響する問題について確認してください。『*Sun Enterprise 10000 Dynamic Reconfiguration* ユーザーマニュアル』と『*Sun Enterprise 10000 Dynamic Reconfiguration* リファレンスマニュアル』が、http://docs.sun.com で参照できます。

動的再構成の概要

DR 機能では、システムハードウェアの切り離しなどの操作をシステムの稼動中に行うことができます。 DR プロセスの目的は、システムを停止したり、クラスタの可用性を中断したりせずにシステム操作を継続できるようにすることです。

DR はボードレベルで機能します。したがって、DR 操作は、ボードのすべてのコンポーネントに影響を及ぼします。ボードには、CPU やメモリー、ディスクドライブやテープドライブ、ネットワーク接続の周辺機器インタフェースなど、複数のコンポーネントが取り付けられています。

ボードを切り離すと、ボード上のすべてのコンポーネントが使用不能になります。 DR サブシステムは、ボードを切り離す前に、ボードのコンポーネントが使用されていないかどうかを確認します。 使用されているデバイスを切り離すと、システムエラーになります。 DR サブシステムは、デバイスが使用されているのを発見すると、DR のボード切り離し操作を拒否します。したがって、DR のボード切り離し操作のタイミングによって問題が発生することはありません。

同様に、DR のボード追加操作も常に安全です。新たに追加されたボードの CPU とメモリーは、システムによって自動的にサービス状態になります。ただし、そのボードの他のコンポーネントを使用するには、クラスタを手動で構成する必要があります。

注 - DR サブシステムにはいくつかのレベルがあります。下位のレベルがエラーを報告すると、上位のレベルもエラーを報告します。 しかし、下位のレベルがエラーを特定しても、上位のレベルが「原因不明のエラー (Unknown Error)」と報告することがあります。この上位レベルのエラーは無視してください。

次の各項では、デバイスタイプごとに DR の注意事項を説明します。

CPU デバイスに対する DR クラスタリング

DR のボード切り離し操作によってボード上の CPU に影響があると、DR サブシステムは操作を受け入れた上で、これらの CPU をそのノードから使用できないようにします。

さらに、DR のボード追加操作によってボードの CPU に影響があると、DR サブシステムは、これらの CPU をそのノードから使用できるようにします。

メモリーに対する DR クラスタリング

DR では、メモリーを 2 種類に分けて考える必要があります。これらの違いはその使用方法だけであり、実際のハードウェアは同じものです。

オペレーティングシステムが使用するメモリーは、カーネルメモリーケージと呼ばれます。Sun Cluster ソフトウェアは、カーネルメモリーケージを含むボードに対するボード切り離し操作をサポートしていないため、このような操作を拒否します。DR のボード切り離し操作によってカーネルメモリーケージ以外のメモリーに影響があると、DR サブシステムはその操作を受け入れた上で、メモリーをそのノードから使用できないようにします。

DR のボード追加操作によってメモリーに影響があると、DR サブシステムは、新しいメモリーをそのノードから使用できるようにします。

ディスクドライブとテープドライブに対する DR クラスタリング

主ノードのアクティブなドライブに対し、 DR 切り離し操作を行うことはできません。DR 切り離し操作を実行できるのは、主ノードのアクティブでないドライブや二次ノードのドライブの場合だけです。 クラスタのデータアクセスは、DR 操作の前でも後でも同じように行われます。

注 - 定足数デバイスの可用性に影響を与える DR 操作を行うことはできません。 定足数デバイスの考慮事項と、定足数デバイスに対する DR 操作の実行手順については、81ページの「定足数デバイスに対する DR クラスタリング」を参照してください。

次の各手順は、ディスクドライブやテープドライブに対する DR の切り離し操作を要約したものです。詳細な手順については、『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

1. **ディスクドライブやテープドライブが、アクティブなデバイスグループに属しているかどうかを確認します。**
   - ドライブがアクティブなデバイスグループに属していない場合は、そのドライブに対して DR 切り離し操作を行うことができます。
   - DR 切り離し操作によってアクティブなディスクドライブやテープドライブに影響がある場合には、システムは操作を拒否し、操作によって影響を受ける

ドライブを特定します。そのドライブがアクティブなデバイスグループに属している場合は、81ページの手順 2 に進みます。

2. **ドライブが主ノードのコンポーネントであるか、二次ノードのコンポーネントであるかを確認します。**
   - ドライブが二次ノードのコンポーネントである場合は、そのドライブに対して DR 切り離し操作を行うことができます。
   - ドライブが主ノードのコンポーネントである場合は、主ノードと二次ノードを切り替えてから、そのデバイスに対して DR 切り離し操作を行う必要があります。

注意 - 二次ノードに対して DR 操作を行っているときに現在の主ノードに障害が発生すると、クラスタの可用性が損なわれます。 これは、新しい二次ノードが提供されるまでは、主ノードのフェイルオーバー先が存在しないためです。

定足数デバイスに対する DR クラスタリング

定足数デバイスとして構成されているデバイスに対し、 DR 切り離し操作を行うことはできません。DR のボード切り離し操作によって定足数デバイスに影響がある場合には、システムは操作を拒否し、操作によって影響を受ける定足数デバイスを特定します。定足数デバイスとしてのデバイスに対して DR 切り離し操作を行う場合は、まずそのデバイスを無効にする必要があります。

次の各手順は、定足数デバイスに対する DR 切り離し操作を要約したものです。詳細な手順については、『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

1. **DR 操作を行うデバイス以外のデバイスを定足数デバイスとして有効にします。**
2. **DR 操作を行うデバイスを定足数デバイスとして無効にします。**
3. **そのデバイスに対して DR 切り離し操作を行います。**

プライベートインターコネクトインタフェースに対する DR クラスタリング

アクティブなプライベートインターコネクトインタフェースに対し、DR 操作を行うことはできません。DR のボード切り離し操作によってアクティブなプライベートインターコネクトインタフェースに影響がある場合には、システムは操作を拒否し、操作によって影響を受けるインタフェースを特定します。アクティブなインタフェースを切り離す場合は、まずそれを無効にする必要があります (下記の注意を参照)。プライベートインターコネクトへのインタフェースが置き換えられても、その状態は変わりません。そのため、Sun Cluster の再構成を新たに行う必要はありません。

次の各手順は、プライベートインターコネクトインタフェースに対する DR 切り離し操作を要約したものです。詳細な手順については、『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

注意 - Sun Cluster の個々のクラスタノードには、他のすべてのクラスタノードに対する有効なパスが、少なくとも 1 つは存在していなければなりません。したがって、個々のクラスタノードへの最後のパスをサポートするプライベートインターコネクトインタフェースを無効にしないでください。

1. **DR 操作を行うインターコネクトインタフェースを含むトランスポートケーブルを無効にします。**

2. **物理プライベートインターコネクトインタフェースに対して DR 切り離し操作を行います。**

パブリックネットワークインタフェースに対する DR クラスタリング

DR 切り離し操作は、アクティブでないパブリックネットワークインタフェースに対して行うことができます。DR のボード切り離し操作によってアクティブなパブリックネットワークインタフェースに影響がある場合には、システムは操作を拒否し、操作によって影響を受けるインタフェースを特定します。パブリックネットワークインタフェースがアクティブである場合は、まず、ネットワークアダプタフェイルオーバー (NAFO) グループの、アクティブネットワークインスタンス というステータスから削除する必要があります。

注意 - 無効にされたネットワークアダプタに対して DR 切り離し操作を行っている間にアクティブなネットワークアダプタに障害が発生すると、システムの可用性が損なわれます。これは、DR 操作の間は、アクティブなネットワークアダプタのフェイルオーバー先が存在しないためです。

次の各手順は、パブリックネットワークインタフェースに対する DR 切り離し操作を要約したものです。詳細な手順については、『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

1. **NAFO** グループから削除するために、アクティブアダプタをバックアップアダプタに切り替えます。

2. アダプタを **NAFO** グループから削除します。

3. パブリックネットワークインタフェースに対して **DR** 操作を行います。

第 4 章

頻繁に寄せられる質問 (FAQ)

この章では、SunPlex システムに関して最も頻繁に寄せられる質問に対する回答を示します。回答は、トピックごとに構成されています。

高可用性に関する FAQ

- 可用性の高いシステムとは何ですか。

  SunPlex システムでは、高可用性 (HA) を、通常サーバーシステムを使用不能にするような障害が発生した場合でも、クラスタがアプリケーションを起動して実行できる能力として定義しています。

- クラスタが高可用性を提供するプロセスは何ですか。

  クラスタフレームワークは、フェイルオーバーとして知られるプロセスによって可用性の高い環境を提供します。フェイルオーバーとは、障害の発生したノードからクラスタ内の別の動作可能ノードにデータサービスリソースを移行するために、クラスタによって実行される一連のステップです。

- フェイルオーバーとスケーラブルデータサービスの違いは何ですか。

  高い可用性を備えたデータサービスには、フェイルオーバーデータサービスとスケーラブルデータサービスがあります。

  フェイルオーバーデータサービスとは、アプリケーションが一度に 1 つのクラスタ内の主ノードだけで実行されることを示します。他のノードは他のアプリケーションを実行できますが、各アプリケーションは単一のノードでのみ実行されま

す。主ノードに障害が発生すると、障害が発生したノードで実行されていたアプリケーションは、別のノードに処理を引き継いで実行を続けます。

スケーラブルサービスは、アプリケーションを複数のノードに広げて、単一の論理サービスを作成します。スケーラブルサービスは、実行されるクラスタ全体のノードとプロセッサの数を強化します。

クラスタへの物理インタフェースは、アプリケーションごとに1つのノードに設定されます。このノードを広域インタフェースノード (GIN) と呼びます。クラスタには、複数の GIN が存在することがあります。個々の GIN には、スケーラブルサービスから使用する 1 つまたは複数の論理インタフェースがあります。この論理インタフェースを広域インタフェースと呼びます。GIN は、特定のアプリケーションに対するすべての要求を広域インタフェースを介して受け取り、それらを、そのアプリケーションサーバーが動作している複数のノードに振り分けます。GIN に障害が発生すると、広域インタフェースは別のノードにフェイルオーバーされます。

アプリケーションが実行されているノードに障害が発生すると、アプリケーションは別のノードで実行を続けますが、障害が発生したノードがクラスタに戻るまで多少のパフォーマンス低下が生じます。

ファイルシステムに関する FAQ

- **1** つまたは複数のクラスタノードを、他のクラスタノードをクライアントとして使用して、可用性の高い **NFS** サーバーとして実行できますか。実行できません。

  ループバックマウントはしないでください。

- リソースグループマネージャー **(RGM)** によって制御されていないアプリケーションでクラスタファイルシステムを使用できますか。

  開放できます。ただし、RGM の制御下にないと、アプリケーションは、実行されているノードに障害があった場合、手動で再起動する必要があります。

- すべてのクラスタファイルシステムのマウントポイントが、`/global` ディレクトリになければなりませんか。

  いいえ。ただし、クラスタファイルシステムを `/global` などの同一のマウントポイントのもとに置くと、これらのファイルシステムの構成と管理は簡単になります。

- クラスタファイルシステムを使用した場合と **NFS** ファイルシステムをエクスポートした場合の違いは何ですか。

  次に示すいくつかの違いがあります。

  1. クラスタファイルシステムは広域デバイスをサポートします。NFS は、デバイスへの遠隔アクセスをサポートしません。
  2. クラスタファイルシステムには広域名前空間があります。したがって、必要なのは 1 つのマウントコマンドだけです。これに対し、NFS では、ファイルシステムを各ノードにマウントする必要があります。
  3. クラスタファイルシステムは、NFS よりも多くの場合でファイルをキャッシュします。たとえば、ファイルが、読み取り、書き込み、ファイルロック、非同期入出力のために複数のノードからアクセスされている場合です。
  4. クラスタファイルシステムは、1 つのサーバーに障害が生じた場合のシームレスなフェイルオーバーをサポートします。NFS は複数のサーバーをサポートしますが、フェイルオーバーは読み取り専用ファイルシステムにのみ可能です。
  5. クラスタファイルシステムは、リモート DMA とゼロコピー機能を提供する、将来の高速クラスタインターコネクトを利用するために組み込まれています。
  6. クラスタファイルシステムのファイルの属性を (`chmod(1M)` などを使用して) 変更すると、変更内容はすべてのノードでただちに反映されます。エクスポートされた NFS ファイルシステムでは、この処理に時間がかかる場合があります。

- クラスタノードに `/global/.devices/`*<node>*`@`*<node ID>* というファイルシステムがあります。高い可用性と広域属性を与えたいデータをこのファイルシステムに格納できますか。

  広域デバイス名前空間が格納されているこれらのファイルシステムは、一般的な使用を目的としたものではありません。これらのファイルシステムは広域的な属性をもっていますが、広域的にアクセスされることはありません。つまり、個々のノードは、自身の広域デバイス名前空間にしかアクセスしません。あるノードが停止しても、他のノードがこのノードに代わってこの名前空間にアクセスすることはできません。これらのファイルシステムは、高可用性を備えてはいません。したがって、高可用性や広域属性を与えたいデータをこれらのファイルシステムに格納すべきではありません。

ボリューム管理に関する FAQ

- すべてのディスクデバイスをミラー化する必要がありますか。

  ディスクデバイスの可用性を高くするには、それをミラー化するか、RAID-5 ハードウェアを使用する必要があります。すべてのデータサービスは、可用性の高いディスクデバイスか、可用性の高いディスクデバイスにマウントされたクラスタファイルシステムのどちらかを使用する必要があります。このような構成では、単一のディスク障害に耐えられます。

- ローカルディスク **(**起動ディスク**)** に対してあるボリューム管理ソフトウェアを使用し、多重ホストディスクに対して別のボリューム管理ソフトウェアを使用することはできますか。

  この構成は、ローカルディスクを管理する Solstice DiskSuite ソフトウェアと、多重ホストディスクを管理する VERITAS Volume Manager の組み合わせによってサポートされます。これ以外の組み合わせではサポートされません。

データサービスに関する FAQ

- 利用可能な **SunPlex** データサービスは何ですか。

  サポートされているデータサービスについては、『*Sun Cluster 3.0 12/01* ご使用にあたって』を参照してください。

- **SunPlex** データサービスによってサポートされているアプリケーションのバージョンは何ですか。

  サポートされているアプリケーションのバージョンについては、『*Sun Cluster 12/01* ご使用にあたって 』を参照してください。

- 独自のデータサービスを作成できますか。

  開放できます。詳細については、『*Sun Cluster 3.0 12/01* データサービス開発ガイド』と DSDL API と共に提供される文書「Data Service Enabling Technologies」を参照してください

- ネットワークリソースを作成する場合に、**IP** アドレスで指定するのですか。またはホスト名で指定するのですか。

ネットワークリソースを指定する場合には、IP アドレスではなく、UNIX のホスト名を使用することを推奨します。

- ネットワークリソースを作成する場合に、論理ホスト名 **(LogicalHostname** リソース**)** または共有アドレス **(SharedAddress** リソース**)** を使用した場合の違いは何ですか。

  Sun Cluster HA for NFS の場合を除き、`Failover` モードリソースグループの `LogicalHostname` リソースを使用するときは、`SharedAddress` リソースと `LogicalHostname` リソースは同様に使用できます。`SharedAdderss` リソースを使用すると、クラスタネットワーキングソフトウェアが `LogicalHostname` ではなく、`SharedAddress` に合わせて構成されているために、多少のオーバーヘッドが生じます。

  `SharedAddress` を使用する利点は、スケーラブルおよびファイルオーバーの両方のデータサービスを構成していて、クライアントが同じホスト名を使用して両方のサービスにアクセスできるようにする場合に得られます。この場合、`SharedAddress` リソースはフェイルオーバーアプリケーションリソースとともに 1 つのリソースグループに含まれますが、スケーラブルサービスリソースは個別のリソースグループに含まれ、`SharedAddress` を使用するように構成されます。スケーラブルおよびフェイルオーバーサービスはどちらも、`SharedAddress` リソースに構成された同じホスト名とアドレスのセットを使用します。

パブリックネットワークに関する FAQ

- **SunPlex** システムがサポートするパブリックネットワークアダプタは何ですか。

  現在、SunPlex システムは、Ethernet (10/100BASE-T および 1000BASE-SX Gb) パブリックネットワークアダプタをサポートしています。今後新しいインタフェースがサポートされる可能性があるため、最新情報については、ご購入先に確認してください。

- フェイルオーバーでの **MAC** アドレスの役割は何ですか。

  フェイルオーバーが発生すると、新しいアドレス解決プロトコル (ARP) パケットが生成されて伝送されます。これらの ARP パケットには、新しい MAC アドレス (ノードの処理が続行される新しい物理アダプタのアドレス) と古い IP アドレスが含まれます。ネットワーク上の別のマシンがこれらのパケットの 1 つを受け

取ると、その ARP キャッシュから古い MAC-IP マッピングがフラッシュされて、新しいものが使用されます。

- **SunPlex** システムでは、ホストアダプタ用の **OpenBoot™ PROM (OBP)** に **local-mac-address?=true** を設定できますか。

  設定できません。この変数はサポートされていません。

- **NAFO** がアクティブアダプタとバックアップアダプタ間のスイッチオーバーを行う際に、どのくらいの遅延がありますか。

  この遅延は数分に及ぶことがあります。これは、NAFO スイッチオーバーが行われる際に余分な ARP を送信する必要があるためです。ただし、クライアントとクラスタ間のルーターが、この余分な ARP を必ず使用するとは限りません。ルーターは、この IP アドレスの ARP キャッシュエントリがタイムアウトになるまで、停止状態の MAC アドレスを使用する可能性があります。遅延の 2 つ目の理由は、両方の NAFO アダプタが同じ Ethernet スイッチに接続されていることがあるためです。NAFO スイッチオーバーが行われると、一方の NAFO アダプタは切り離された状態に、他方の NAFO アダプタは接続された状態にそれぞれなります。したがって、Ethernet スイッチは、一方のポートを無効に、他方のポートを有効にする必要があるため、時間がかかります。さらに、Ethernet では、スイッチと新たに有効にされたアダプタの間で速度の折衝が行われるため、これにも時間がかかります。最後に、スイッチオーバーが完了すると、NAFO は、新たに有効にされたアダプタに対して最小限の検査を行い、すべてが正しく動作するか確認します。

クラスタメンバーに関する FAQ

- すべてのクラスタメンバーが同じ **root** パスワードを持つ必要がありますか。

  各クラスタメンバーに同じ root パスワードを設定する必要はありません。ただし、同じ root パスワードをすべてのノードに使用すると、クラスタの管理を簡略化できます。

- ノードが起動される順序は重要ですか。

  ほとんどの場合、重要ではありません。ただし、起動順序は、amnesia (詳細は、53ページの「定足数と定足数デバイス」を参照) を防止するために重要です。たとえば、ノード 2 が定足数デバイスの所有者であり、ノード 1 が停止してノード 2 を停止させた場合は、ノード 2 を起動してからノード 1 を起動する必要

があります。これにより、古いクラスタ構成情報を持つノードを誤って起動するのを防ぐことができます。

- クラスタノードのローカルディスクをミラー化する必要がありますか。

  開放できます。このミラー化は必要条件ではありませんが、クラスタノードのディスクをミラー化すると、ノードを停止させる非ミラー化ディスクの障害を防止できます。ただし、クラスタノードのローカルディスクをミラー化すると、システム管理の負荷が増えます。

- クラスタメンバーのバックアップの注意点は何ですか。

  クラスタには、いくつかのバックアップ方式を使用できます。1 つの方法としては、ノードをテープドライブまたはライブラリが接続されたバックアップノードとして設定します。さらに、クラスタファイルシステムを使用してデータをバックアップします。このノードは共有ディスクには接続しないでください。

  バックアップと復元の手順については、『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

クラスタ記憶装置に関する FAQ

- 多重ホスト記憶装置の可用性を高めるものは何ですか。

  多重ホスト記憶装置は、単一のディスクが失われてもミラー化(またはハードウェアベースの RAID-5 コントローラ) のために存続できるので、可用性が高くなります。多重ホスト記憶装置には複数のホスト接続があるため、接続先の単一ノードが失われても耐えることができます。

クラスタインターコネクトに関する FAQ

- **SunPlex** システムがサポートするクラスタインターコネクトは何ですか。

  現在、SunPlex システムは、Ethernet (100BASE-T Fast Ethernet および 1000BASE-SX Gb) クラスタインターコネクトをサポートしています。

- 「ケーブル」とトランスポート「パス」の違いは何ですか。

  クラスタトランスポートケーブルは、トランスポートアダプタとスイッチを使用して構成されます。ケーブルでは、アダプタやスイッチを結合します(コンポー

ネントとコンポーネント)。クラスタトポロジマネージャーは、利用可能なケーブルを使用し、ノード間にエンドツーエンドのトランスポートパスを構築します。ただし、ケーブルとトランスポートパスが 1 対 1 で対応しているわけではありません。

ケーブルは、管理者によって静的に「有効」または「無効」にされます。ケーブルには「状態」(有効または無効) はありますが、「ステータス」はありません。無効なケーブルは、構成されていない状態と同じです。無効なケーブルをトランスポートパスとして使用することはできません。ケーブルを検査することはできないため、そのステータスを知ることはできません。ケーブルの状態を見るには `scconf –p` を使用します。

トランスポートパスは、クラスタトポロジマネージャーによって動的に確立されます。トランスポートパスの「ステータス」はトポロジマネージャーによって決められますが、パスのステータスには「オンライン」と「オフライン」があります。トランスポートパスのステータスを知るには `scstat(1M)` を使用します。

次のような 2 ノードクラスタがあるとします。これには、4 つのケーブルが使用されています。

```
node1:adapter0    to switch1, port0

node1:adapter1    to switch2, port0

node2:adapter0    to switch1, port1

node2:adapter1    to switch2, port1
```

これらの 4 つのケーブルを使用して設定できるトランスポートパスには、次の 2 つがあります。

```
node1:adapter0    to node2:adapter0

node2:adapter1     to node2:adapter1
```

クライアントシステムに関する FAQ

- クラスタでの使用における特殊なクライアントの要求や制約について考慮する必要がありますか。

クライアントシステムは、他のサーバーに接続する場合と同様にクラスタに接続します。データサービスアプリケーションによっては、クライアント側ソフトウェアをインストールするか、別の構成変更を行なって、クライアントがデータサービスアプリケーションに接続できるようにしなければならないこともあります。クライアント側の構成条件の詳細については、『*Sun Cluster 3.0 12/01* データサービスのインストールと構成』を参照してください。

管理コンソールに関する FAQ

- **SunPlex** システムには管理コンソールが必要ですか。

  必要です。

- 管理コンソールをクラスタ専用にする必要がありますか、または別の作業に使用することができますか。

  SunPlex システムは、専用の管理コンソールを必要としません。ただし、専用のコンソールを使用すると、次の利点が得られます。

  - コンソールと管理ツールを同じマシンにまとめることで、クラスタ管理を一元化できる。
  - ハードウェアサービスプロバイダによる問題解決が迅速に行われる

- 管理コンソールをクラスタの近く、たとえば同じ部屋に配置する必要がありますか。

  ハードウェアの保守担当者に確認してください。保守作業上、コンソールをクラスタの近くに配置する必要がある場合があります。コンソールを同じ部屋に配置する必要性は、技術的にはありません。

- 距離の条件すべてが合致するかぎり、**1** 台の管理コンソールが複数のクラスタにサービスを提供できますか。

  開放できます。複数のクラスタを1 台の管理コンソールから制御できます。また、1 台の端末集配信装置 (コンセントレータ) をクラスタ間で共有することもできます。

端末集配信装置とシステムサービスプロセッサに関する FAQ

- **SunPlex** システムは端末集配信装置を必要としますか。

  Sun Cluster 3.0 以降のすべてのソフトウェアリリースを実行するために、端末集配信装置は必要はありません。障害による影響防止に端末集配信装置を必要とした Sun Cluster 2.2 とは異なり、Sun Cluster 3.0 以降の製品では端末集配信装置に依存しません。

- ほとんどの **SunPlex** サーバーは端末集配信装置を使用していますが、**E10000** は使用していないのはなぜですか。なぜですか。

  端末集配信装置は、ほとんどのサーバーで効率的なシリアル - Ethernet コンバータです。そのコンソールポートはシリアルポートです。Sun Enterprise E10000 サーバーには、シリアルコンソールがありません。システムサービスプロセッサ (SSP) は Ethernet または `jtag` ポートを介したコンソールです。Sun Enterprise E10000 サーバーの場合は、コンソールに対して常に SSP を使用します。

- 端末集配信装置を使用した場合の利点は何ですか。

  端末集配信装置を使用すると、ノードが OpenBoot PROM (OBP) にある場合でも、ネットワーク上の任意の場所にある遠隔ワークステーションから各ノードに対して、コンソールレベルのアクセスを行えます。

- **Sun** がサポートしていない端末集配信装置を使用する場合に注意する点は何ですか。

  Sun がサポートする端末集配信装置と他のコンソールデバイスの主な違いは、Sun の端末集配信装置には、端末集配信装置がコンソールに対して起動時にブレークを送信するのを防ぐ特殊なファームウェアがあるという点です。ブレーク、またはコンソールに対してブレークと解釈されることがある信号を送信するコンソールデバイスである場合、ノードが停止されることに注意してください。

- **Sun** がサポートする端末集配信装置を再起動しないで、そこにあるロックされたポートを開放できますか。

  開放できます。リセットする必要があるポート番号を書きとめて、次のコードを実行してください。

  ```
  telnet tc
  ```

(続く)

```
Enter Annex port name or number: cli
annex: su -
annex# admin
admin : reset port_number
admin : quit
annex# hangup
#
```

Sun がサポートする端末集配信装置の構成と管理の詳細については、『*Sun Cluster 3.0 12/01* のシステム管理』を参照してください。

- 端末集配信装置自体に障害が発生した場合はどうしたらいいですか。別の装置を用意しておく必要がありますか。

  ありません。端末集配信装置に障害が発生しても、クラスタの可用性はまったく失われません。端末集配信装置が再び機能するまで、ノードコンソールに接続できなくなります。

- 端末集配信装置を使用する場合に、セキュリティはどのように制御しますか。

  通常、端末集配信装置は、他のクライアントアクセスに使用されるネットワークではなく、システム管理者が使用する小規模なネットワークに接続されています。この特定のネットワークに対するアクセスを制限することでセキュリティを制御できます。

用語集

この以下の用語は、SunPlex 3.0 のマニュアルで使用されています。

amnesia
クラスタの停止後そのクラスタが無効なクラスタ構成データ (CCR) を使用して再起動する状態。たとえば、ノード 1 だけが動作可能な 2 ノードクラスタで、クラスタ構成の変更がノード 1 で発生した場合、ノード 2 の CCR は無効になる。クラスタ全体が停止してノード 2 から再起動した場合、ノード 2 の無効な CCR が原因で amnesia 状態が発生する。

DID ドライバ
Sun Cluster ソフトウェアによって実装されるドライバで、クラスタ間で一貫したデバイス名前空間を提供する。「DID 名」も参照。

DID 名
SunPlex システムの広域デバイスを識別するために使用される。Solaris 論理名と 1 対 1 または 1 対多の関係を持つクラスタ化識別子で、d*X*s*Y* の形式をとる。*X* は整数、*Y* はスライス名を示す。「Solaris 論理名」も参照。

HA データサービス
「データサービス」を参照。

Scalable Coherent Interface (SCI)
クラスタインターコネクトとして使用される高速インターコネクトハードウェア。

Solaris 物理名
Solarisのデバイスドライバによってデバイスに指定される名前。これは、Solaris マシンを、`/devices` ツリーのパスとして示す。たとえば、典型的な SCSI ディスクには、次のような Solaris 物理名がある。

```
/devices/sbus@1f,0/SUNW,fas@e,8800000/sd@6,0:c,raw
```

「Solaris 論理名」も参照。

Solaris 論理名 通常、Solaris デバイスの管理に使用される名前。ディスクの場合、これらは通常、`/dev/rdsk/c0t2d0s2` のようになる。これらの各 Solaris 論理デバイス名には、対応する Solaris 物理デバイス名がある。「DID 名」と「Solaris 物理名」も参照。

Solstice DiskSuite SunPlex システムによって使用されるボリューム管理ソフトウェア。「ボリューム管理ソフトウェア」も参照。

split brain クラスタが複数のパーティションに分割される状態。各パーティションは、相互の存在を認識しないで形成される。

Sun Cluster (ソフトウェア**)** SunPlex システムのソフトウェア部分。「SunPlex」を参照

SunPlex 高可用性サービスやスケーラブルサービスを提供するために使用される、ハードウェアと Sun Cluster ソフトウェアが統合されたシステム。

VERITAS Volume Manager SunPlex システムによって使用されるボリューム管理ソフトウェア。「ボリューム管理ソフトウェア」も参照。

イベント 管理されるオブジェクトの状態、マスター、重大度の説明における変更。

インスタンス 「リソース起動」を参照。

管理コンソール クラスタ管理ソフトウェアの実行に使用されるワークステーション。

共有アドレスリソース クラスタ内のノードで実行されるすべてのスケーラブルサービスによって結合されるネットワークアドレス。これによりノード上のサービスがスケーラブルになる。クラスタは複数の共有アドレスを持つことができ、サービスは複数の共有アドレスに結合できる。

クラスタ クラスタファイルシステムを共有し、フェイルオーバー、パラレルリソース、またはスケーラブルリソースを実行するためにともに構成された、複数の相互接続されたノードやドメイン。

クラスタインターコネクト ケーブル、クラスタトランスポート接続点、およびクラスタトランスポートアダプタから成るハードウェアのネットワークインフラス

	トラクチャ。Sun Cluster およびデータサービスソフトウェアにより、クラスタ内通信で使用される。
クラスタ構成レポジトリ **(CCR)**	Sun Cluster ソフトウェアによって、クラスタ構成情報を持続的に保存するために使用される可用性の高い複製データストア。
クラスタトランスポートアダプタ	ノードに常駐して、ノードをクラスタインターコネクトに接続するネットワークアダプタ 。「クラスタインターコネクト」も参照。
クラスタトランスポートケーブル	終端に接続するネットワーク接続。クラスタトランスポートアダプタとクラスタトランス ポート接続点の間の接続、または 2 つのクラスタトランスポートアダプタの間の接続。「クラスタインターコネクト」も参照。
クラスタトランスポート接続点	クラスタインターコネクトの一部として使用されるハードウェアスイッチ。「クラスタインターコネクト」も参照。
クラスタノード	クラスタメンバーとして構成されたノード。クラスタノードは、現在のメンバーである場合も、そうでない場合もある。「クラスタメンバー」も参照。
クラスタファイルシステム	既存のローカルファイルシステムに対し、クラスタ全体で可用性の高いアクセスを提供するクラスタサービス。
クラスタメンバー	現在のクラスタ実装のアクティブメンバー。このメンバーは、他のクラスタメンバーとリソースを共有し、他のクラスタメンバー、およびクラスタのクライアントの両方にサービスを提供できる。「クラスタノード」も参照。
クラスタメンバーシップモニター **(CMM)**	クラスタメンバーシップの一貫性を維持するソフトウェア。このメンバーシップ情報は、残りのクラスタ化されたソフトウェアによって、可用性の高いサービスを配置する場所を決定するために使用される。CCM は、非クラスタメンバーがデータを破壊したり、破壊されたデータや矛盾したデータをクライアントに送信できないようにする。
広域インタフェース	共有アドレスの物理的なホストとなる広域ネットワークインタフェース。「共有アドレス」も参照。
広域インタフェースノード	広域インタフェースのホストとなるノード。

広域デバイス	ディスク、CD-ROM、テープなど、すべてのクラスタメンバーからアクセスできるデバイス。
広域デバイス名前空間	広域デバイスを示すクラスタ全体にわたる論理的な名前空間。Solaris 環境のローカルデバイスは、`/dev/dsk`、`/dev/rdsk`、`/dev/rmt` ディレクトリに定義されている。広域デバイス名前空間は、広域デバイスを、`/dev/global/dsk`、`/dev/global/rdsk`、`/dev/global/rmt` の各ディレクトリに定義する。
広域リソース	Sun Cluster ソフトウェアのカーネルレベルで提供される可用性の高いリソース。広域リソースには、ディスク (HA デバイスグループ)、クラスタファイルシステム、広域ネットワーキングを含めることができる。
コロケーション **(collocation)**	同じノードに存在するプロパティ。この概念は、パフォーマンスを向上させるために、クラスタ構成中に使用される。
自動フェイルバック	主ノードがエラーを起こし、その後にクラスタのメンバーとして再起動された場合、リソースグループまたはデバイスグループをその主ノードに返すプロセス。
システムサービスプロセッサ **(SSP)**	Sun Enterprise 10000 構成で、特にクラスタメンバーとの通信に使用されるクラスタ外部のデバイス。
終端	クラスタトランスポートアダプタまたはクラスタトランスポート接続点上の物理ポート。
主ノード	リソースグループまたはデバイスグループが現在オンラインであるノード。つまり、主ノードとは、現在リソースに関連するサービスをホストしているか、または実装しているノードのこと。「二次ノード」も参照。
主ホスト名	主パブリックネットワーク上のノードの名前。常に、`/etc/nodename` に指定されたノード名になる。「二次ホスト名」も参照。
障害モニター	データサービスの各部分を検出して、処置をとるために使用される障害デーモンとプログラム。「リソースモニター」も参照。

スイッチオーバー	クラスタ内のあるマスター (ノード) から別のマスター (または、リソースグループが複数の主ノードに構成されている場合は複数のマスター) へのリソースグループ、またはデバイスグループの正常な転送。スイッチオーバーは、管理者によって、`scswitch(1M)` コマンドを使用して開始される。
スイッチバック	「フェイルバック」を参照。
スケーラブルサービス	複数のノードで同時に実行されるように実装されたデータサービス。
スケーラブルリソース	複数のノードで実行されるリソース (各ノードのインスタンス) で、クラスタインターコネクトを使用して、サービスの遠隔クライアントに対して単一サービスの外観を提示する。
潜在主ノード	主ノードに障害が発生した場合に、フェイルオーバーリソースをマスターできるクラスタメンバー。「デフォルトマスター」も参照。
潜在マスター	「潜在主ノード」を参照。
多重ホームホスト	複数のパブリックネットワーク上にあるホスト。
多重ホストディスク	物理的に複数のノードに接続されたディスク。
単一インタフェースリソース	クラスタ全体で最大で 1 つのリソースをアクティブにできるリソース。
端末集配信装置	Sun Enterprise 10000 以外の構成で、特にクラスタメンバーとの通信に使用されるクラスタ外部のデバイス。コンセントレータとも言う。
チェックポイント	ソフトウェアの状態の同期を取るように指示するため、主ノードから二次ノードに対して送信される通知。「主ノード」と「二次ノード」も参照。
テイクオーバー	「フェイルオーバー」を参照。
ディスクセット	「デバイスグループ」を参照。
ディスクグループ	「デバイスグループ」を参照。

ディスクデバイスグループ　「デバイスグループ」を参照。

定足数デバイス　定足数 (quorum) を確立してクラスタを実行するために使用される票を持つ、複数のノードによって共有されるディスク。クラスタは、定足数の票が利用可能な場合にのみ動作する。定足数デバイスは、クラスタが独立したノードの集合にパーティション分割されて、新しいクラスタを構成するノードの集合を設定するときに使用される。

データサービス　リソースグループマネージャー (RGM) の制御下で、可用性の高いリソースとして実行されるように実装されたアプリケーション。

デバイス **ID**　Solaris を介して使用可能になるデバイスを識別する機構。`devid_get(3DEVID)` のマニュアルページも参照。

Sun Cluster DID ドライバは、デバイス ID を使用して、異なるクラスタノードの Solaris 論理名間の相関関係を判断する。DID ドライバは、各デバイスでそのデバイス ID を検証する。そのデバイス ID が、クラスタ内の他の場所にある別のデバイスと一致する場合、両方のデバイスに同じ DID 名が指定される。そのデバイス ID がクラスタ内にない場合は、新しい DID が割り当てられる。「Solaris 論理名」と「DID ドライバ」も参照。

デバイスグループ　クラスタ HA 構成内の異なるノードからマスター可能な、ディスクなどのデバイスリソースのユーザー定義グループ。このグループには、ディスクのデバイスリソース、Solstice DiskSuite ディスクセット、および VERITAS Volume Manager ディスクグループを含めることができる。

デフォルトマスター　フェイルオーバーリソースタイプがオンラインになるデフォルトクラスタメンバー。

二次ノード　主ノードに障害が発生した場合にディスクデバイスグループおよびリソースグループをマスターするために使用できるクラスタメンバー。「主ノード」も参照。

二次ホスト名　二次パブリックネットワークのノードにアクセスするために使用される名前。「主ホスト名」も参照。

ネットワークアダプタフェイルオーバー **(NAFO)** グループ	同じノード、およびアダプタの障害時に相互にバックアップするように構成された同じサブネットにある 1 つまたは複数のネットワークアダプタのセット。
ネットワークアドレスリソース	「ネットワークリソース」を参照。
ネットワークリソース	1 つまたは複数の論理ホスト名または共有アドレスを含むリソース。「論理ホスト名リソース」と「共有アドレスリソース」も参照。
ノード	SunPlex システムの一部にできる物理的なマシンまたは (Sun Enterprise E10000 サーバーの) ドメイン。「ホスト」とも呼ばれる。
ハートビート	利用可能なクラスタインターコネクトトランスポートパスすべてに送信される定期的なメッセージ。指定の間隔および再試行回数の後にハートビートがない場合は、別のパスに対するトランスポート通信の内部フェイルオーバーが引き起こされる。クラスタメンバーへのすべてのパスに障害が発生した場合は、CMM によるクラスタ定足数 (quorum) の再評価が行われる。
バックアップグループ	「ネットワークアダプタフェイルオーバーグループ」を参照。
パブリックネットワーク管理 **(PNM)**	障害モニターとフェイルオーバーを使用し、単一のネットワークアダプタやケーブルの障害が原因でノードの可用性が失われるのを防ぐソフトウェア。PNM フェイルオーバーは、ネットワークアダプタフェイルオーバーグループと呼ばれるネットワークアダプタの集合を使用し、クラスタノードとパブリックネットワークの間の冗長接続を実現する。障害モニターとフェイルオーバー機能は共に動作し、リソースの可用性を確保する。「ネットワークアダプタフェイルオーバーグループ」も参照。
パラレルサービスインスタンス	個々のノードで実行されるパラレルリソースタイプのインスタンス。
パラレルリソースタイプ	クラスタ環境で実行されて、複数の (2 つ以上の) ノードによって同時にマスターできるように実装された、パラレルデータベースなどのリソースタイプ。

汎用リソース	汎用リソースタイプの一部としてリソースグループマネージャの制御下におかれるアプリケーションデーモンとその子プロセス。
汎用リソースタイプ	データサービスのテンプレート。汎用リソースタイプは、単純なアプリケーションをフェイルオーバーデータサービス (あるノードで停止して、別のノードで起動する) にするために使用できる。このタイプは、SunPlex API によるプログラミングを必要としない。
非クラスタモード	–x 起動オプションによってクラスタメンバーを起動した結果の状態。この状態では、ノードはクラスタメンバーではなくなるが、引き続きクラスタノードである。「クラスタメンバー」と「クラスタノード」も参照。
フェイルオーバー	障害発生後、現在の主ノードから新しい主ノードにリソースグループまたはデバイスグループを自動的に再配置すること。
フェイルオーバーリソース	リソースの 1 つで、その各リソースは、一度に 1 つのノードによって正しくマスターできる。「単一インスタンスリソース」と「スケーラブルリソース」も参照。
フェイルバック	「自動フェイルバック」を参照。
フェイルファースト **(failfast)**	無効な操作によって損傷が判明する前に、障害が発生したノードのクラスタを正常に終了して移動すること。
負荷均衡	スケーラブルサービスにのみ適用される。アプリケーション負荷をクラスタ内のノード全体に分配して、クライアント要求が適宜サービスされるようにするプロセス。詳細については、64ページの「スケーラブルデータサービス」を参照。
負荷均衡ポリシー	スケーラブルサービスにのみ適用される。アプリケーションが要求した負荷をノード間で分配するための推奨される方法。詳細については、64ページの「スケーラブルデータサービス」を参照。
プライベートホスト名	クラスタインターコネクトを介してノードと通信するために使用されるホスト名のエイリアス。
分散ロックマネージャ **(DLM)**	共有ディスク Oracle Parallel Server (OPS) 環境で使用されるロックソフトウェア。DLM を使用すると、異なる複数のノードで実行される Oracle プロセスが、データベースアクセスの同期をとること

ができる。DLM は、高い可用性を目的として設計されている。プロセスまたはノードがクラッシュしても、残りのノードを停止して再起動する必要はない。DLM の高速再構成が実行されて、このような障害から回復する。

ボリューム管理ソフトウェア　ディスクのストライプ化、連結、ミラー化、メタデバイスやボリュームの動的成長によってデータの信頼性を提供するソフトウェア製品。

マスター　「主ノード」を参照。

メタデバイス状態データベースの複製 **(**複製**)**　ディスクに保存された、すべてのメタデバイスの構成と状態およびエラー条件を記録するデータベース。この情報は、Solstice DiskSuite ディスクセットの正しい操作に重要であり、複製される。

ローカルディスク　特定のクラスタノードに物理的に接続された専用のディスク。

論理ネットワークインタフェース　インターネットアーキテクチャでは、ホストは 1 つまたは複数の IP アドレスを持つことができる。Sun Cluster ソフトウェアは、追加の論理ネットワークインタフェースを構成して、いくつかの論理ネットワークイン タフェースと単一の物理ネットワークインタフェース間のマッピングを確立する。各論理ネットワークイン タフェースは単一の IP アドレスを持つ。このマッピングを使用すると、単一物理ネットワークインタフェ ースが複数の IP アドレスに応答できる。また、このマッピングを使用すると、テイクオーバーやスイッチ オーバーのときに、ハードウェアインタフェースを追加することなく、IP アドレスを1 つのクラスタメンバーから別のメンバーに移動できる。

論理ホスト　アプリケーション、ディスクセット、またはアプリケーションデータが常駐するディスクグループと、クラスタへのアクセスに使用されるネットワークアドレスを含む、リソースのセットを表す Sun Cluster 2.0 の概念。この概念は、SunPlex システム には存在しない。この概念を現在 SunPlex システムで実装する方法については、46ページの「ディスクデバイスグループ」と 73ページの「リソース、リソースグループ、リソースタイプ」を参照してください。

論理ホスト名リソース	ネットワークアドレスを表す論理ホスト名の集合を含むリソース。論理ホスト名リソースは、一度に 1 つのノードによってのみマスターできる。「論理ホスト」も参照。
リソース	リソースタイプのインスタンス。同じタイプの多数のリソースが存在して、各リソースが独自の名前とプロパティ値を持つため、実際に使用するアプリケーションの多数のインスタンスをクラスタで実行できる。
リソース管理 **API (RMAPI)**	クラスタ環境においてアプリケーションの可用性を高める、SunPlex システムのアプリケーションプログラミングインタフェース。
リソースグループ	RGM によって 1 つの単位として管理されるリソースの集合。RGM によって管理される各リソースは、リソースグループに構成する必要がある。通常は、関連性があり相互に依存するリソースがグループ化される。
リソースグループ状態	ノードのリソースグループの状態。
リソースグループマネージャー **(RGM)**	クラスタリソースを指定のクラスタノードで自動的に起動、停止することによって、これらのリソースの可用性を高め、スケーラブルにするために使用されるソフトウェア機能。RGM は、構成済みのポリシーに従って、ハードウェアまたはソフトウェア障害、あるいは再起動時に動作する。
リソース状態	指定ノードのリソースグループマネージャーのリソースの状態。
リソースステータス	障害モニターによってレポートされるリソースの状態。
リソースタイプ	データサービス、`LogicalHostname`、`SharedAddress` クラスタオブジェクトに指定された一意の名前。データサービスリソースタイプは、フェイルオーバータイプかスケーラブルタイプのどちらかである。「データサービス」、「フェイルオーバーリソース」、「スケーラブルリソース」も参照。
リソースタイププロパティ	RGM によってリソースタイプの一部として保存され、指定タイプのリソースを記述して管理するために使用される重要な値のペア。

リソースモニター	リソースが正常に実行されているかどうか、およびどのように実行されているかを判別するための定期的な障害検出をリソースに対して実行する、リソースタイプ実装のオプション部分。
リソース呼び出し	ノードで実行されるリソースタイプのインスタンス。ノードで起動されたリソースを表す抽象的な概念。