Sun™ NFS™ サーバーの調整

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

Part No. 806-3908-10
2000 年 2 月
Revision A

原典　NFS Server Performance and Tuning Guide for Sun Hardware
Part No: 806-2195-10
Revision A

目次

図目次

表目次

はじめに

このマニュアルでは、NFS™ 分散型ネットワークファイルシステムに関する、以下の内容について説明します。

- NFS とネットワークの性能分析および性能調整
- NFS ネットワーク監視ツール

対象とするサーバーは、以下の条件を満たしている必要があります。

- Solaris™ 2.4、2.5、2.5.1、2.6、7、8 のいずれかのオペレーティング環境が動作していること。
- ネットワーク環境で使用していること。
- Ultra 5S、10S、SE2、250、450、SPARCserver、SPARCcenter 2000(E)、Netra™ NFS 150、Sun™ Enterprise™ 3000/4000/5000/6000、3500/4500/5500/6500 のいずれかのシステムであること。

このマニュアルは、ネットワーククライアントに NFS サービスを提供するサーバーのシステム構成、性能分析および性能調整を担当するシステム管理者とネットワーク担当者を対象にしています。このマニュアルでは、Solaris 2.4、2.5、2.5.1、2.6、7 オペレーティング環境における NFS のバージョン 2 およびバージョン 3 の性能調整について説明します。

書体と記号について

このマニュアルで使用している書体と記号について説明します。

表 P–1　このマニュアルで使用している書体と記号

書体または記号	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。	.login ファイルを編集します。 **ls -a** を実行します。 % You have mail.
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表します。	machine_name% **su** Password:
AaBbCc123 またはゴシック	コマンド行の可変部分。実際の名前や値と置き換えてください。	**rm** *filename* と入力します。 **rm** ファイル名 と入力します。
『 』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照。 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こえる場合に、継続を示します。	% **grep '^#define ** **XV_VERSION_STRING'**

シェルプロンプトについて

シェルプロンプトの例を以下に示します。

表 P–2　シェルプロンプト

シェル	プロンプト
UNIX の C シェル	machine_name%
UNIX の Bourne シェルと Korn シェル	machine_name$
スーパーユーザー (シェルの種類を問わない)	#

第1章

NFS の概要

この章では、NFS™ の特徴、性能調整の行程、NFS の性能を監視するためのサン以外のツールについて簡単に説明します。

NFS の特徴

NFS 環境では、ネットワークを経由して、ローカルファイルと同様に遠隔ファイルにアクセスすることができます。遠隔デバイスのファイルシステムは、ローカルに存在するかのように見えます。クライアントは、`mount` コマンドやオートマウンタを利用して、遠隔ファイルシステムにアクセスすることができます。

NFS プロトコルは、クライアントからの再試行と障害からの容易な回復を可能にします。クライアントは、サーバーが処理を行うために必要なすべての情報を提供します。サーバーで受信が確認されるまで、または再試行が時間切れになるまで、要求を繰り返し送信します。データが不揮発性の記憶装置にフラッシュされると、サーバーから書き込みの確認があります。

マルチスレッド方式のカーネルでは、`nfsd` プロセスや非同期ブロック入出力デーモン (`biod`) プロセスを複数管理する必要がありません。どちらのプロセスも、オペレーティングシステムのカーネルのデーモンとして実現されています。このため、クライアント側に `biod` は存在せず、サーバー側に `nfsd` プロセスが 1 つ存在します。

NFS トラフィックは、突発性を持っています。NFS 要求は突然発生して、通常、その種類も多岐にわたります。NFS サーバーは、このような NFS ファイルサービス要求に対処する必要があります。要求は広範囲にわたりますが、通常の運用では、そうした要求は比較的予測可能なものです。

ローカル、遠隔にかかわらず、アプリケーションからの要求の多くは、以下のように処理されます。

1. アプリケーションバイナリの必要部分が読み込まれ、コードページが実行され、処理するデータセットを指定するためのユーザーダイアログが表示されます。
2. アプリケーションによって、指定データセットがディスク (ほとんどの場合は遠隔ディスク) から読み取られます。
3. ユーザーとアプリケーションとの対話が可能になり、メモリー上のデータが操作されます。アプリケーションの実行の大部分は、この対話に費やされます。
4. 変更を加えられたデータセットがディスクに保存されます。

注 – 通常は、上記の処理が進むにつれて、アプリケーションバイナリのより多くの部分が読み込まれます (ページイン) 。

NFS バージョン 2 と 3 について

Solaris™ 2.5 から 8 までのソフトウェア環境では、NFS のバージョン 2 と 3 の両方が提供されます。NFS バージョン 3 は、Solaris 2.5 以降のソフトウェア環境で新たに追加されたものです。

NFS のバージョン 2 とバージョン 3 のどちらを使用するかについて、NFS クライアントとサーバーはネゴシエーションを行います。サーバーが NFS バージョン 3 をサポートする場合は、バージョン 3 がデフォルトで使用されます。デフォルトで使用される NFS バージョンを変更するには、`vers=` マウントオプションを変更します。

NFS バージョン 2 とバージョン 3 は、同じ方法で調整することができます。

NFS バージョン 3 の機能と特長

NFS バージョン 3 には、性能の改善とサーバーの負荷軽減、ネットワークトラフィック量の軽減を実現するための機能がいくつか含まれています。NFS バージョン 3 では、入出力書き込みが高速化されており、ネットワークを介した処理が減っているため、ネットワーク使用時の効率性を向上させることができます。スループットが大きくなるにしたがって、ネットワークが混雑する (ビジー状態になる) 場合があります。

NFS バージョン 3 は、バージョン 2 の状態を持たない (stateless) サーバーの設計と、単純な障害回復機能を継承しながら、協調するプロトコルによる分散型ファイルサービスを構築する手法をとっています。

NFS バージョン 2 およびバージョン 3 の主な機能と特長を、以下に示します。

表 1–1　NFS 操作

操作	バージョン 2 の機能	バージョン 3 での変更点
create	ファイルシステムノードを作成します。ファイルまたはシンボリックリンクのどちらでも作成できます。	なし
statfs	動的にファイルシステム情報を取得します。	fsstat に置き換え
getattr	ファイルタイプやサイズ、アクセス権、アクセス時間などのファイル、ディレクトリ属性を取得します。	なし
link	リモートファイルシステムにハードリンクを作成します。	なし
lookup	ディレクトリからファイルを探し、ファイルハンドルを返します。	なし
mkdir	ディレクトリを作成します。	なし
null	何もしません。サーバーからの応答の検査とタイミング調整に使用します。	なし
read	8 KB のデータブロックを読み出します (32 KB データブロック)。TCP では 64 KB まで可能です。	データブロックは最大 4 GB
readdir	ディレクトリエントリを読み出します。	なし
readlink	サーバーに作成されているシンボリックリンクに従います。	なし
rename	ファイルのディレクトリ名エントリを変更します。	なし
remove	ファイルシステムノードを削除します。	なし
rmdir	ディレクトリを削除します。	なし

表 1–1　NFS 操作

操作	バージョン 2 の機能	バージョン 3 での変更点
root	リモートファイルシステムのルートを読み出します (現在は不使用) 。	削除
setattr	ファイル、ディレクトリ属性を変更します。	なし
symlink	リモートファイルシステムにシンボリックリンクを作成します。	なし
wrcache	リモートキャッシュに 8 KB のデータブロックを書き出します (現在は不使用) 。	削除
write	8 KB のデータブロックを書き込みます (32 KB データブロック)。TCP では 64 KB まで可能です。	データブロックは最大 4 GB

NFS バージョン 3 の新機能を、以下に示します。

表 1–2　NFS バージョン 3 の新機能

バージョン 3 での操作	機能
access	アクセス権の確認
mknod	特殊デバイスの作成
readdir	ディレクトリからの読み込み
readdirplus	ディレクトリからの拡張読み込み
fsinfo	ファイルシステムの静的な情報の取得
pathconf	POSIX 情報の取得
commit	正常な記憶装置にキャッシュされたサーバー上のデータの確認

バージョン 3 での変更点

root および writecache は削除されました。mknod は、特殊ファイルの作成が許可されるように定義されました。このため、create のオーバーロードは排除されます。 クライアント上でのキャッシュは、バージョン 3 でも定義や命令はできません。バージョン 3 には、クライアントが効率的にキャッシュを管理するためのキャッシュの実装についての情報が追加されました。

ファイルやディレクトリの属性に影響を与える操作では、属性キャッシュの検証で使用された後続の `getattr` を最適化する操作が完了した後、新しい属性を返すようになります。また、目的のオブジェクトが隣接するディレクトリに変更を加える操作は、検証中にクライアントがより効率的なキャッシュを実装できるように、ディレクトリの古い属性と新しい属性を返します。

`access` は、サーバー上でのアクセス権の確認をする機能です。`fsstat` は、ファイルシステムとサーバーの静的な情報を返す機能です。`readdirplus` は、ディレクトリエントリに加え、ファイルハンドルや属性を返す機能です。`pathconf` は、ファイルに関する POSIX パス構成情報を返す機能です。

64 ビットファイルサイズ

NFS プロトコルのバージョン 3 では、64 ビットサイズのファイルを扱うことができます。バージョン 2 では、ファイルサイズは 32 ビット (4 GB 以下) である必要がありました。

大きなファイル (64 ビット) を扱うには、クライアント、サーバー、オペレーティングシステムが 64 ビットファイルに対応している必要があります。クライアントの実装状態が 32 ビットファイルのみに対応している場合は、サーバーが 64 ビットファイルに対応している場合でも、クライアントは 64 ビットファイルを扱うことはできません。また、クライアントが 64 ビットファイルに対応していて、サーバーが 32 ビットファイルのみに対応している場合においても、クライアントは 32 ビットファイルのみを扱うことができます。Solaris 7 オペレーティング環境は、このプロトコルの機能を最初に利用する Solaris リリースです。Solaris 7 より前のオペレーティング環境では、64 ビットファイルを扱うことはできません。

Solaris 2.6、Solaris 7、および Solaris 8 オペレーティング環境における UNIX® ファイルシステムの制限は、1 TB (40 ビット) です。

非同期書き込み (async write)

NFS バージョン 3 では、オプションで非同期書き込み機能を利用することができます。NFS バージョン 3 クライアントは、サーバーに非同期書き込み要求を送信し、サーバーは、データを受信したことを通知します。ただし、このときサーバーは、応答する前に安定した記憶装置にデータを書き込む必要はありません。書き込みをスケジュールするか、複数の書き込み要求がまとまるのを待つことができます。

クライアントは、サーバーが書き込みを完了できない場合に備えて、データのコピーを保持します。クライアントは、コピーを解放する場合に、COMMIT 操作によりサーバーに通知します。サーバーは、データを安定した記憶装置に書き込んだ後で、肯定応答を返します。これ以外の場合はエラーが返されるため、クライアントはデータを同期モードで再送します。

非同期書き込みによって、サーバーはデータの同期をとる最善の方法を決めることができます。データは、COMMIT が着く前に同期がとられる可能性が非常に高くなります。NFS バージョン 2 と比較して、バッファリング処理の効率が増し、並列処理の度合いが高まります。

NFS バージョン 2 では、データが安定した記憶装置に書き込まれるまで、サーバーは書き込み要求に応答しません。ただし、サーバーが要求に応答する前に、複数の書き込み要求をまとめることによって、複数の並行要求を発行することができます。

属性付きディレクトリの読み取り

NFS バージョン 3 では、READDIRPLUS と呼ばれる操作があります。たとえば、ls や ls -l などの、大部分の READDIR が READDIRPLUS コールとして発行されます。バージョン 3 で ls -l コマンドを実行すると、ディレクトリ内の名前リストと共に、ファイルハンドルと属性が返されます。バージョン 2 では、名前が最初に返され、ファイルハンドルと属性を取得するには、続いてサーバーを呼び出す必要があります。

バージョン 3 の READDIRPLUS 操作の利点は、ファイルごとに GETATTR 要求を送信する必要がないため時間が短縮され、ls と ls -l の速度が同程度になることです。

弱いキャッシュの一貫性維持

ほとんどの NFS バージョン 2 クライアントは、ファイルとディレクトリのデータをキャッシュして性能向上を図っています。ただし、このバージョン 2 の方法は、複数のクライアントが同じデータを共有してキャッシュする際に、正しく動作しない場合があります。

弱いキャッシュの一貫性維持により、クライアントは、前回アクセスしてから、次に要求を出すまでの間に、別のクライアントによってデータが変更されたかどうかを検出することができます。これは、応答と一緒にサーバーに前回の属性を送り返させることによって実現します。これにより、クライアントは、実際の前回の属性と自分が持っている属性を比較し、違いを検出することができます。

性能調整の行程

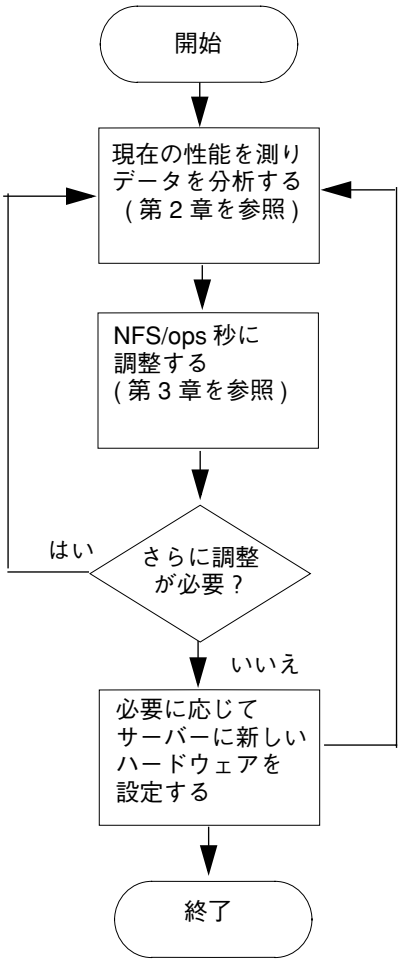

図 1–1　性能調整の流れ

NFS の性能を監視するためのサン以外のツール

NFS ネットワークに使用可能なサン以外のツールを、以下に示します。

- NetMetrix (Hewlett-Packard)
- SharpShooter (Network General Corporation, 旧 AIM Technology)

SharpShooter (Version 3) は、NFS プロトコルのバージョン 3 に対応しています。

第2章

NFS 性能の分析

この章では、NFS 性能の分析方法について説明します。システムを調整するための一般的な手順を簡単に示し、ネットワーク、サーバー、各クライアントの検査方法について説明します。

調整手順

NFS サーバーを最初に設定する際には、サーバーが最高の性能を発揮するように調整する必要があります。また、後で、特定の状況に応じてよりよい性能が得られるように、その都度調整する必要があります。

全般的な性能を向上させるための調整手順

以下の手順に従って、NFS サーバーの性能を向上させます。

1. ネットワーク、サーバー、各クライアントの現在の性能レベルを測定します。10 ページの「ネットワーク、サーバー、クライアントの性能検査」を参照してください。
2. 収集されたデータのグラフを作成し、分析します。例外、ディスクと CPU の使用状況、ディスクサービス時間を調べてください。しきい値または性能値ルールを、データに適用します。
3. サーバーを調整します。第 3 章を参照してください。

4. 満足のいく性能レベルが得られるまで手順 1 から手順 3 を繰り返します。

性能上の問題を解決するための調整手順

性能上の問題が発生した場合の調整手順は以下のとおりです。

1. 状態を観察し、ツールを使用して問題の原因を突き止めます。
2. ネットワークとサーバー、各クライアントの現在の性能レベルを測定します。
   10 ページの「ネットワーク、サーバー、クライアントの性能検査」を参照してください。
3. 収集されたデータのグラフを作成し、分析します。例外、ディスクと CPU の使用状況、ディスクサービス時間を調べてください。しきい値または性能値ルールを、データに適用します。
4. サーバーを調整します。第 3 章を参照してください。
5. 満足のいく性能レベルが得られるまで手順 1 から手順 4 を繰り返します。

ネットワーク、サーバー、クライアントの性能検査

NFS サーバーの調整は、ネットワーク、NFS サーバー、各クライアントの性能を検査してから行います。最初に検査するのはネットワークです。ディスクが正常に動作している場合に、NFS クライアントから見てサーバーが遅いということは、ネットワークが遅いということと同じです。ネットワークの使用状況を調べてください。

▼ ネットワークを調べる

1. 各ネットワークのパケット数と衝突/エラー発生回数を調べます。

```
server% netstat -i 15
      input   le0       output    input       (Total)    output
packets errs  packets errs  colls  packets errs  packets errs  colls
10798731 533   4868520 0     1078   24818184 555   14049209 157   894937
51       0     43       0     0      238      0     139      0     0
85       0     69       0     0      218      0     131      0     2
44       0     29       0     0      168      0     94       0     0
```

他のインタフェースを調べるには `-I` を使用します。

上記の `netstat` コマンドで使用している引数の意味は以下のとおりです。

表 2–1 `netstat -i 15` コマンドの引数

引数	説明
`-i`	TCP/IP ネットワークに使われている全インタフェースの状態を表示します。
`15`	15 秒ごとに情報を収集します。

`netstat -i 15` を入力した画面では、ネットワークトラフィックが使用可能なマシンによって、入力パケットと出力パケットの両方が連続的に増加していることが示されます。

2. **出力側衝突回数 (`Output Colls - le`) を出力パケット数 (`le`) で割ることによってネットワーク衝突率を求めます。**

   たいていの場合、ネットワーク全体の衝突率が 10 %より高いということは、ネットワークが過負荷である場合や、ネットワークの構成に問題がある場合、あるいはハードウェアに問題がある、などを意味します。

3. **入力エラー発生回数 (`le`) を入力パケット数の合計 (`le`) で割ることによって入力パケット誤り率を求めます。**

   入力誤り率が 25 %を超える場合は、ホストによってパケットが正しく送信されていない可能性があります。

   ネットワークのハードウェア、またはトラフィックの混雑、ローレベルのハードウェア上の問題によっても、伝送上の問題が発生することがあります。たとえば、ブリッジやルーターによってパケットが正しく送信されないと、強制的に再送信が行われ、性能低下の原因になります。

   ブリッジがパケットヘッダーの Ethernet アドレスを調べることによって、遅延が生じます。アドレスの検出中、ブリッジネットワークインタフェースによって、パケットの一部が正しく送信されないことがあります。

   ネットワークハードウェアに帯域幅の制限がある場合は、以下の作業を行ってください。

   - パケットサイズを小さくします。

- mount を使用するか、または /etc/vfstab ファイルに、読み取りバッファーサイズ (rsize) と書き込みバッファーサイズ (wrsize) を指定します。ブリッジを通るデータの方向によって値は異なりますが、これらの変数を 2048 に設定してください。データがブリッジなどの装置を介して双方向に伝送される場合は、両方の変数の値をさらに小さくしてください。

```
server:/home  /home/server nfs rw,rsize=2048,wsize=2048 0 0
```

クライアントがユーザーデータグラムプロトコル (UDP) と通信していて、多数の読み取りおよび書き込み要求が正しく送信されない場合は、送信されなかったパケットの代わりにパケット全体を再送信します。

4. **クライアントから「 ping -sRv サーバー名」と入力して、パケットの経路を表示することによって、パケットがネットワークを往復 (エコー) するために必要な時間を調べます。**

ネットワークが遅いか、ネットワークに遅いルータがあるか、あるいはネットワークが非常に混み合っている場合は、往復に数ミリ秒 (ms) を要します。最初に入力した ping コマンドから返された結果は、無視してください。また、ping -sRv コマンドは、パケットロス数も表示します。

以下は、ping -sRv コマンドの画面出力です。

```
client% ping -sRv サーバー名
PING server: 56 data bytes
64 bytes from server (129.145.72.15): icmp_seq=0. time=5. ms
  IP options:  <record route> router (129.145.72.1), server
(129.145.72.15), client (129.145.70.114),  (End of record)
64 bytes from server (129.145.72.15): icmp_seq=1. time=2. ms
  IP options:  <record route> router (129.145.72.1), server
(129.145.72.15), client (129.145.70.114),  (End of record)
```

上記の ping コマンドで使用している引数の意味は以下のとおりです。

表 2–2 ping コマンドの引数

引数	説明
s	1 秒にデータグラムを 1 個送信し、エコー応答を受信するたびに 1 行の結果を表示します。応答がない場合、結果は表示されません。
R	経路を記録します。インターネットプロトコル (IP) 記録オプションが設定され、IP ヘッダー内のパケットの経路が保存されます。
v	詳細情報オプションです。受信したエコー応答以外のすべての ICMP パケットを一覧表示します。

ハードウェアに問題があると思われる場合は、ping -sRv を使って、ネットワークに接続されている複数台のホストの応答時間を調べてください。予測していた応答時間が得られない場合は、そのホストに問題がある可能性があります。

ping コマンドは ICMP プロトコルのエコー要求データグラムを使って、指定ホストまたはネットワークゲートウェイから ICMP エコー要求を引き出します。時分割方式の NFS サーバーでは、ICMP エコーを取得するのにかなり時間がかかることがあります。サーバーから ICMP エコーを取得するのに要する時間は、クライアントから NFS サーバーまでの距離によっても変化します。

`ping -sRv` コマンドに対する応答が得られた場合と、得られなかった場合の処理の流れを以下に示します。

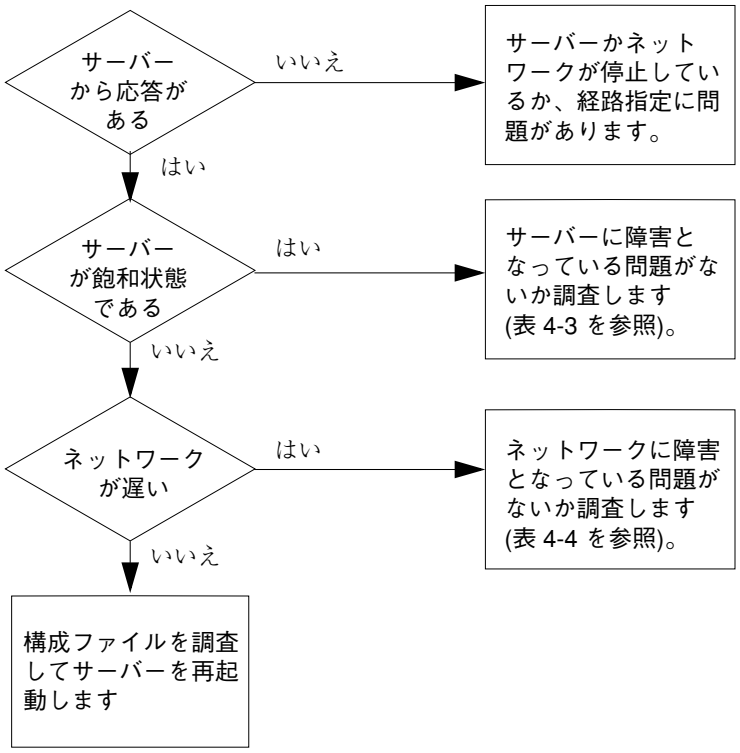

図 2–1 `ping -sRv` コマンドに対する応答の流れ

NFS サーバーの検査

注 – 以下では、大規模な構成の SPARCserver 690 システムを例にとって説明しています。

▼ NFS サーバーを検査する

1. 何がエクスポートされているのかを調べます。

```
server% share
-               /export/home    rw=netgroup    ""
-               /var/mail    rw=netgroup    ""
-               /cdrom/solaris_2_3_ab    ro    ""
```

2. マウントされているファイルシステムと、そのファイルシステムが実際にマウントされているディスクドライブを表示します。

```
server% df -k
Filesystem             kbytes    used   avail capacity  Mounted on
/dev/dsk/c1t0d0s0      73097     36739  29058     56%    /
/dev/dsk/c1t0d0s3      214638    159948 33230     83%    /usr
/proc                  0         0      0         0%    /proc
fd                     0         0      0         0%    /dev/fd
swap                   501684    32     501652    0%    /tmp
/dev/dsk/c1t0d0s4      582128    302556 267930    53%    /var/mail
/vol/dev/dsk/c0t6/solaris_2_3_ab
/dev/md/dsk/d100       7299223   687386 279377    96%    /export/home
                       113512    113514 0         100%   /cdrom/solaris_2_3_ab
```

注 – この例では、`/var/mail` と `/export/home` ファイルシステムを使用しています。

`df -k` コマンドを使用してファイルサーバーが存在するディスク番号を調べます。

ファイルシステムの使用率が 100 % になると、通常、クライアント側で NFS 書き込みエラーが発生します。

上記の例の `/var/mail` は `/dev/dsk/clt0d0s4` に、`/export/home` は Solstice DiskSuite メタディスクの `/dev/md/dsk/d100` に存在しています。

3. df -k コマンドを使用して Solstice DiskSuite メタディスクが返された場合は、ディスク番号を確認します。

```
server% /usr/opt/SUNWmd/sbin/metastat ディスク番号
```

上記の例の /usr/opt/SUNWmd/sbin/metastate d100 によって、物理ディスク /dev/md/dsk/d100 が使用しているディスクを確認することができます。

d100 ディスクはミラー化されています。各ミラーは、3 つのストライプディスクと、4 つのストライプディスクを連結したものから構成されます。それぞれのセット内のディスクは同じ大きさですが、セット間では大きさが異なります。ミラーの他に、ホットスペアディスクもあります。このシステムでは、IPI ディスク (idX) を使用しており、 SCSI ディスク (sdX) は同じディスクとして扱われます。

```
server% /usr/opt/SUNWmd/sbin/metastat d100
d100: metamirror
    Submirror 0: d10
      State: Okay
    Submirror 1: d20
      State: Okay
    Regions which are dirty: 0%
d10: Submirror of d100
    State: Okay
    Hot spare pool: hsp001
    Size: 15536742 blocks
    Stripe 0: (interlace : 96 blocks)
    Device              Start Block  Dbase State        Hot Spare
    /dev/dsk/c1t1d0s7          0     No    Okay
    /dev/dsk/c2t2d0s7          0     No    Okay
    /dev/dsk/c1t3d0s7          0     No    Okay
    Stripe 1: (interlace : 64 blocks)
    Device              Start Block  Dbase State        Hot Spare
    /dev/dsk/c3t1d0s7          0     No    Okay
    /dev/dsk/c4t2d0s7          0     No    Okay
    /dev/dsk/c3t3d0s7          0     No    Okay
    /dev/dsk/c4t4d0s7          0     No    Okay
d20: Submirror of d100
    State: Okay
    Hot spare pool: hsp001
Size: 15536742 blocks
    Stripe 0: (interlace : 96 blocks)
    Device              Start Block  Dbase State        Hot Spare
    /dev/dsk/c2t1d0s7          0     No    Okay
    /dev/dsk/c1t2d0s7          0     No    Okay
    /dev/dsk/c2t3d0s7          0     No    Okay
    Stripe 1: (interlace : 64 blocks)
    Device              Start Block  Dbase State        Hot Spare
    /dev/dsk/c4t1d0s7          0     No    Okay
    /dev/dsk/c3t2d0s7          0     No    Okay
    /dev/dsk/c4t3d0s7          0     No    Okay
   /dev/dsk/c3t4d0s7        0    No    Okay     /dev/dsk/c2t4d0s7
```

4. エクスポートされている各ファイルシステムについて `/dev/dsk` エントリを調べます。以下のいずれかの方法を使用します。

   - `whatdev` スクリプトを使用してドライブのインスタンスかニックネームを調べます(手順 5 に進みます)。
   - `ls -lL` コマンドを使用して `/dev/dsk` エントリを調べます (手順 6 に進みます)。

5. whatdev スクリプトを使用して /dev/dsk エントリを確認する場合は、以下のようにします。

   a. テキストエディタを使用して、以下の whatdev スクリプトを作成します。

```
#!/bin/csh
# print out the drive name - st0 or sd0 - given the /dev entry
# first get something like "/iommu/.../.../sd@0,0"
set dev = '/bin/ls -l $1 | nawk '{ n = split($11, a, "/"); split(a[n],b,"”:");
for(i = 4; i < n; i++) printf("/%s",a[i]); printf("/%s\n", b[1]) }''
if ( $dev == "" ) exit
# then get the instance number and concatenate with the "sd"
nawk -v dev=$dev '$1 ~ dev { n = split(dev, a, "/"); split(a[n], \
b, "@"); printf("%s%s\n", b[1], $2) }' /etc/path_to_inst
```

   b. 「df -k /ファイルシステム名」と入力して、ファイルシステムの /dev/dsk エントリを調べます。

      この例では、df -k /var/mail と入力します。

```
furious% df -k /var/mail
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c1t0d0s4     582128  302556  267930    53%    /var/mail
```

   c. 「whatdev ディスク名」と入力して、ディスク番号を求めます (ディスク名は、df -k /ファイルシステム名 コマンドによって返されたディスク名)。

      この例では、whatdev /dev/dsk/c1t0d0s4 と入力します。この場合のディスク番号は id8 (IPI ディスク 8) です。

```
server% whatdev /dev/dsk/c1t0d0s4
id8
```

   d. メタディスク (/dev/md/dsk) 以外のディスクに存在する各ファイルシステムについて手順 b と 手順 c を繰り返します。

e. ファイルシステムにメタディスク (/dev/md/dsk) が存在する場合は、metastat の出力に応じて、メタディスクを構成する全ドライブに対して whatdev スクリプトを実行します。

この例では、whatdev /dev/dsk/c2t1d0s7 と入力します。

/export/home ファイルシステムは、14 のディスクから構成されています。このディスクの 1 つである /dev/dsk/c2t1d0s7 ディスクに、whatdev スクリプトを実行すると、次の結果が得られます。

```
server% whatdev /dev/dsk/c2t1d0s7
id17
```

この場合の /dev/dsk/c2t1d0s7 はディスク id17 (IPI ディスク 17) です。

f. 手順 7 に進みます。

6. whatdev スクリプトではなく、ls -1L を使用して /dev/dsk エントリを確認する場合は、以下のようにします。

a. 「ls -1L ディスク番号」と入力し、ドライブとそのドライブの主および副デバイス番号を表示します。

/var/mail ファイルシステムの場合の入力例を以下に示します。

ls -lL /dev/dsk/c1t0d0s4

```
ls -lL /dev/dsk/c1t0d0s4
brw-r-----  1 root      66,  68 Dec 22 21:51 /dev/dsk/c1t0d0s4
```

b. ls -1L の出力から副デバイス番号を探します。

この例では、ファイル所有権 (root) の直後の 66 が主デバイス番号、次の 68 は副デバイス番号です。

c. ディスク番号を調べます。

- 上記の例では、副デバイス番号 (68) を 8 で割ります (68÷8 = 8.5)。
- 端数を切り捨てます。8 がディスク番号です。

d. スライス (パーティション) 番号を求めます。

ディスク番号の s (slice の s) の後にある数字を確認します。
たとえば `/dev/dsk/c1t0d0s4` の場合は、s の後の 4 がスライス 4 を示します。

ディスク番号が 8、スライス番号が 4 であることを確認できました。
このディスクは `sd8` (SCSI) か `id8` (IPI) です。

7. 「`iostat -x 15`」と入力して、各ディスクのディスク統計情報を表示します。
`-x` によって、拡張ディスク統計情報を指示するオプションです。`15` は、ディスク統計情報を 15 秒おきに収集することを意味します。

```
server% iostat -x 15
extended disk statistics
disk       r/s  w/s   Kr/s    Kw/s wait actv  svc_t  %w  %b
id10       0.1  0.2    0.4     1.0  0.0  0.0   24.1   0   1
id11       0.1  0.2    0.4     0.9  0.0  0.0   24.5   0   1
id17       0.1  0.2    0.4     1.0  0.0  0.0   31.1   0   1
id18       0.1  0.2    0.4     1.0  0.0  0.0   24.6   0   1
id19       0.1  0.2    0.4     0.9  0.0  0.0   24.8   0   1
id20       0.0  0.0    0.1     0.3  0.0  0.0   25.4   0   0
id25       0.0  0.0    0.1     0.2  0.0  0.0   31.0   0   0
id26       0.0  0.0    0.1     0.2  0.0  0.0   30.9   0   0
id27       0.0  0.0    0.1     0.3  0.0  0.0   31.6   0   0
id28       0.0  0.0    0.0     0.0  0.0  0.0    5.1   0   0
id33       0.0  0.0    0.1     0.2  0.0  0.0   36.1   0   0
id34       0.0  0.2    0.1     0.3  0.0  0.0   25.3   0   1
id35       0.0  0.2    0.1     0.4  0.0  0.0   26.5   0   1
id36       0.0  0.0    0.1     0.3  0.0  0.0   35.6   0   0
id8        0.0  0.1    0.2     0.7  0.0  0.0   47.8   0   0
id9        0.1  0.2    0.4     1.0  0.0  0.0   24.8   0   1
sd15       0.1  0.1    0.3     0.5  0.0  0.0   84.4   0   0
sd16       0.1  0.1    0.3     0.5  0.0  0.0   93.0   0   0
sd17       0.1  0.1    0.3     0.5  0.0  0.0   79.7   0   0
sd18       0.1  0.1    0.3     0.5  0.0  0.0   95.3   0   0
sd6        0.0  0.0    0.0     0.0  0.0  0.0  109.1   0   0
```

`iostat -x 15` コマンドを使用し、各ディスクについて拡張ディスク統計情報を得ることができます。次の手順では、`sed` スクリプトを使用してディスク名をディスク番号に変換する方法を説明します。

拡張ディスク統計情報出力の各項目の意味は以下のとおりです。

表 2–3 `iostat -x 15` コマンドの出力 (拡張ディスク統計情報)

引数	説明
`r/s`	秒あたりの読み取り回数
`w/s`	秒あたりの書き込み回数
`Kr/s`	秒あたりの読み取り KB 数
`Kw/s`	秒あたりの書き込み KB 数
`wait`	サービス待ちの平均トランザクション数 (待ち行列の長さ)
`actv`	サービスを受けている平均トランザクション数
`svc_t`	平均サービス時間 (ミリ秒)
`%w`	待ち行列が空になっていない時間の割合
`%b`	ディスクが使用されている時間の割合

8. **ディスク名をディスク番号に変換します。**

   ここでは、`iostat` と `sar` を使用します。ディスク名をディスク番号に変換する最も簡単な方法は、`sed` スクリプトを利用することです。

   a. **以下に示す `d2fs.server sed` スクリプトを参考にし、テキストエディタを使用して `sed` スクリプトを作成します。**

      作成した `sed` スクリプトでは、ディスク番号の代わりにファイルシステム名を使用します。

この例では、id8 ディスクが /var/mail、id9 と id10、id11、id17、id18、id25、id26、id27、id28、id33、id34、id35、id36 ディスクが /export/home です。

```
sed 's/id8 /var/mail/
    s/id9 /export/home/
    s/id10 /export/home/
    s/id11 /export/home/
    s/id17 /export/home/
    s/id18 /export/home/
    s/id25 /export/home/
    s/id26 /export/home/
    s/id27 /export/home/
    s/id28 /export/home/
    s/id33 /export/home/
    s/id34 /export/home/
    s/id35 /export/home/
    s/id36 /export/home/'
```

b. iostat -xc 15 | d2fs.server と入力して、sed スクリプトを通して iostat -xc 15 コマンドを実行します。

iostat -xc 15 | d2fs.server の各オプションの意味を以下の表に示します。

表 2–4 iostat -xc 15 | d2fs.server コマンドのオプション

引数	説明
-x	拡張ディスク統計情報の指定です。
-c	システムがユーザーモード (us) 、システムモード (sy) 、入出力待ち (wt) 、アイドル (id) している時間の長さの割合を報告します。
15	15 秒ごとにディスク統計情報を収集します。

iostat -xc 15 | d2fs.server コマンドの出力例を以下に示します。

```
% iostat -xc 15 | d2fs.server
extended disk statistics           cpu
disk              r/s  w/s   Kr/s   Kw/s wait actv  svc_t  %w  %b  us sy wt id
export/home       0.1  0.2    0.4    1.0  0.0  0.0   24.1   0   1   0 11  2 86
export/home       0.1  0.2    0.4    0.9  0.0  0.0   24.5   0   1
export/home       0.1  0.2    0.4    1.0  0.0  0.0   31.1   0   1
export/home       0.1  0.2    0.4    1.0  0.0  0.0   24.6   0   1
export/home       0.1  0.2    0.4    0.9  0.0  0.0   24.8   0   1
id20              0.0  0.0    0.1    0.3  0.0  0.0   25.4   0   0
export/home       0.0  0.0    0.1    0.2  0.0  0.0   31.0   0   0
export/home       0.0  0.0    0.1    0.2  0.0  0.0   30.9   0   0
export/home       0.0  0.0    0.1    0.3  0.0  0.0   31.6   0   0
export/home       0.0  0.0    0.0    0.0  0.0  0.0    5.1   0   0
export/home       0.0  0.0    0.1    0.2  0.0  0.0   36.1   0   0
export/home       0.0  0.2    0.1    0.3  0.0  0.0   25.3   0   1
export/home       0.0  0.2    0.1    0.4  0.0  0.0   26.5   0   1
export/home       0.0  0.0    0.1    0.3  0.0  0.0   35.6   0   0
var/mail          0.0  0.1    0.2    0.7  0.0  0.0   47.8   0   0
id9               0.1  0.2    0.4    1.0  0.0  0.0   24.8   0   1
sd15              0.1  0.1    0.3    0.5  0.0  0.0   84.4   0   0
sd16              0.1  0.1    0.3    0.5  0.0  0.0   93.0   0   0
sd17              0.1  0.1    0.3    0.5  0.0  0.0   79.7   0   0
sd18              0.1  0.1    0.3    0.5  0.0  0.0   95.3   0   0
sd6               0.0  0.0    0.0    0.0  0.0  0.0  109.1   0   0
```

上記の例の各欄の用語と略語の意味は以下のとおりです。

表 2–5 iostat -xc 15 コマンドの出力

引数	説明
r/s	秒あたりの読み取り回数
w/s	秒あたりの書き込み回数
Kr/s	秒あたりの読み取り KB 数
Kw/s	秒あたりの書き込み KB 数
wait	サービス待ちの平均トランザクション数 (待ち行列の長さ)
actv	サービスを受けている平均トランザクション数
svc_t	平均サービス時間 (ミリ秒)
%w	待ち行列が空になっていない時間の割合

表 2–5 `iostat -xc 15` コマンドの出力 (続き)

引数	説明
`%b`	ディスクが使用されている時間の割合
`svc_t`	ディスク要求の処理を終えるまでの平均サービス時間 (ミリ秒)。この時間には、待ち時間とアクティブ待ち行列時間、シーク回転時間、転送待ち時間が含まれます。
`us`	CPU 時間

c. `sed` スクリプトを介して `sar -d 15 1000` コマンドを実行します。

```
server% sar -d 15 1000 | d2fs.server
12:44:17   device    %busy     avque   r+w/s  blks/s avwait  avserv
12:44:18  export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          id20          0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          var/mail      0         0.0       0       0     0.0   0.0
          export/home   0         0.0       0       0     0.0   0.0
          sd15          7         0.1       4     127     0.0  17.6
          sd16          6         0.1       3     174     0.0  21.6
          sd17          5         0.0       3     127     0.0  15.5
```

-d オプションは、ディスク装置の利用状況を報告します。15 は、15 秒ごとにデータを収集するという意味です。1000 は、データ収集を 1000 回行うという意味です。出力の各欄の用語と略語の意味は以下のとおりです。

表 2–6 sar -d 15 1000 | d2fs.server コマンドの出力

ヘッダー	説明
device	監視中のディスク装置名。
%busy	装置が転送要求の処理に費やした時間の割合 (iostat %b と同じ)。
avque	監視中に未処理の要求の平均個数 (iostat actv と同じ) 。待ち行列に要求がある場合のみ測定されます。
r+w/s	装置に対する 1 秒あたりの読み取りおよび書き込み転送回数 (iostat r/s + w/s と同じ)。
blks/s	1 秒あたりに装置に転送された 512 バイトブロック数 (iostat 2*(kr/s + kw/s)と同じ)。
avwait	転送要求が待ち行列で待たされる平均時間 (ミリ秒) (iostat wait と同じ) 待ち行列に要求がある場合のみ測定されます。
avserv	装置による転送要求の処理を終えるまでの平均時間 (ミリ秒) 。ディスクの場合、この時間には、シーク時間と回転待ち時間、データ転送時間が含まれます。

d. NFS 経由でファイルシステムがエクスポートされている場合は、%b と %busy 値を調べます。

%b 値はディスクがビジーになっている時間の割合を表し、iostat コマンドによって返されます。%busy 値は装置が転送要求の処理に費やした時間の割合を表し、sar コマンドによって返されます。%b と %busy 値が 30 % を超える場合は、手順 e に進みます。30 % を超えていない場合は、手順 9 に進みます。

e. **svc_t 値と avserv 値を調べます。**

svc_t 値は平均サービス時間 (ミリ秒) を表し、iostat コマンドによって返されます。avserv 値は装置による転送要求の処理が終えるまでの平均時間 (ミリ秒) を表し、sar コマンドによって返されます。svc_t と同じ測定値を得るには、avwait を追加します。

svc_t 値 (平均サービス時間) が 40 ミリ秒を超える場合は、ディスクの応答時間が長くなっています。NFS クライアントから見ると、ディスクの入出力を伴う NFS 要求の処理は遅くなります。NFS 応答時間は、NFS プロトコル処理とネットワーク伝送時間を考慮して、平均 50 ミリ秒以下が適当です。ディスクの応答時間は、40 ミリ秒以下が適当です。

平均サービス時間は、ディスクの関数です。高速なディスクを使用している場合、平均サービス時間は遅いディスクを使用する場合にくらべて短くなります。

9. **sys の crontab ファイルの行のコメント指定を外し、sar によって 1 か月間データを収集することによって、定期的にデータを収集します。**

```
root# crontab -l sys
#ident"@(#)sys1.592/07/14 SMI"/* SVr4.0 1.2*/
#
# The sys crontab should be used to do performance collection.
# See cron and performance manual pages for details on startup.
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

性能に関するデータが継続的に収集され、sar の実行結果の記録が作成されます。

注 – /var/adm/sa ファイルには、数百 KB の空き領域が必要です。

---

10. **負荷を分散させます。**

Solstice DiskSuite または Online: DiskSuite を使用し、ファイルシステムを複数のディスクにストライプ処理します。Prestoserve 書き込みキャッシュを使用してアクセス回数を減らし、アクセスのピーク時の負荷を分散します (47 ページの「Solstice DiskSuite または Online: DiskSuite による ディスクのアクセス負荷の分散」を参照)。

11. **読み取り専用ファイルシステムがある場合は、バッファーキャッシュを調整します**
(60 ページの「バッファーキャッシュの調整 (bufhwm)」を参照してください)。

12. %プロンプトに対して nfsstat -s と入力して、NFS の問題点を調べます。

-s オプションを指定すると、サーバーの統計情報を表示することができます。

```
server% nfsstat -s
Server rpc:
calls      badcalls   nullrecv   badlen     xdrcall
480421     0          0          0          0
Server nfs:
calls      badcalls
480421     2
null       getattr    setattr    root       lookup     readlink   read
95  0%     140354 29% 10782  2%  0  0%      110489 23% 286  0%    63095 13%
wrcache    write      create     remove     rename     link       symlink
0   0%     139865 29% 7188   1%  2140  0%   91  0%     19  0%     231  0%
mkdir      rmdir      readdir    statfs
435  0%    127  0%    2514  1%   2710  1%
```

NFS サーバーの画面には、受信された NFS コール数 (calls)、拒否されたコール数 (badcalls) 、実際に行われた各種呼び出し数とその割合が表示されます。nfsstat -s を実行して返される呼び出し数と割合については、以下の表を参照してください。

表 2–7 nfsstat -s コマンドの出力

ヘッダー	説明
calls	受信された RPC コール数の合計。
badcalls	RPC 層に拒否されたコール数の合計 (badlen と xdrcall の合計)。
nullrecv	受信したと思われるのに使用できる RPC コールがなかった回数。
badlen	最小サイズの RPC コールより短い RPC コール数。
xdrcall	ヘッダーを XDR デコードできなかった RPC コール数。

`nfsstat -s` コマンドの出力と対処方法を以下に示します。

表 2–8 `nfsstat -s` コマンドを実行して得られる出力と処置

原因	対処方法
writes 値が 5 % を超える[1]	最高の性能が得られるように、Prestoserve NFS アクセラレータ (SBus カードまたは NVRAM-NVSIMM) をインストールします (55 ページの「Prestoserve NFS アクセラレータ」を参照) 。
`badcalls` が返される	`badcalls` は、RPC レイヤーによって拒否されたコールで、`badlen` と `xdrcall` の合計です。ネットワークが過負荷になっている可能性があります。ネットワークインタフェース統計情報を利用して、過負荷のネットワークを特定してください。
`readlink` が NFS サーバーの `lookup` コール数の合計の 10 % を超える	NFS クライアントが使用しているシンボリックリンクが多すぎます。シンボリックリンクは、サーバーによってエクスポートされたファイルシステム上に存在するリンクです。シンボリックリンクをディレクトリに置き換えてください。NFS クライアントにベースのファイルシステムとシンボリックリンクのターゲットの両方をマウントします。以下の 手順 13 を参照してください。
`getattr` が 40 % を超える	`actimeo` オプションを使ってクライアントの属性キャッシュを大きくします。必ずDNLC と i ノードキャッシュは大きくしてください。`vmstat -s` を使用し DNLC のヒット率 (cache hits) を求め、必要に応じて `/etc/system` ファイルの ncsize 値を大きくします。61 ページの「ディレクトリ名ルックアップキャッシュ (DNLC)」も参照してください。

[1] writes 値 29 % は非常に高い値です。

13. シンボリックリンクを削除します。

`nfsstat -s` コマンド出力の `symlink` 値が 10 % を超える場合は、シンボリックリンクを削除してください。以下の例では、`/usr/tools/dist/sun4` が `/usr/dist/bin` のシンボリックリンク先です。

a. /usr/dist/bin のシンボリックリンクを削除します。

```
# rm /usr/dist/bin
```

b. /usr/dist/bin ディレクトリを作成します。

```
# mkdir /usr/dist/bin
```

c. ディレクトリをマウントします。

```
client# mount server: /usr/dist/bin
client# mount server: /usr/tools/dist/sun4
client# mount
```

14. vmstat -s と入力して、ディレクトリ名ルックアップキャッシュ (DNLC) ヒット率を表示します。

vmstat -s コマンドは、ヒット率 (cache hits) を返します。

```
% vmstat -s
...[略]...
79062 total name lookups (cache hits 94%)
16 toolong
```

a. ロングネーム数に問題がないにもかかわらず、ヒット率が 90 % 以下の場合は、/etc/system ファイルの ncsize 変数値を大きくします。

```
set ncsize=5000
```

30 文字より短いディレクトリ名がキャッシュされます。また、長すぎてキャッシュできないディレクトリ名も報告されます。

ncsize のデフォルト値は以下のとおりです。

ncsize (ネームキャッシュ) = 17*maxusers + 90

- NFS サーバーのベンチマークでは、16000 に設定されています。

- maxusers = 2048 の場合は、34906 に設定されます。

  ディレクトリ名ルックアップキャッシュの詳細については、61 ページの「ディレクトリ名ルックアップキャッシュ (DNLC)」を参照してください。

b. システムを再起動します。

15. Prestoserve NFS アクセラレータを使用している場合は、その状態を調べて、UP 状態になっていることを確認します。

```
server% /usr/sbin/presto
state = UP, size = 0xfff80 bytes
statistics interval: 1 day, 23:17:50  (170270 seconds)
write cache efficiency: 65%
All 2 batteries are ok
```

- DOWN 状態になっている場合は、UP 状態に設定します。

```
server% presto -u
```

- エラー状態の場合は、『Prestoserve User's Guide』を参照してください。

これでサーバーを検査する手順は終了しました。引き続き、各クライアントを検査してください。

各クライアントの検査

全体として見ると、調整にはクライアントの調整も含まれます。クライアントを調整した方が、サーバーを調整するより性能が改善されることがあります。たとえば、100 あるクライアントの 1 台ごとに 4 MB のメモリーを増設することで、非常に効果的に NFS サーバーの負荷を小さくすることができます。

▼ 各クライアントを検査する

1. %プロンプトに対して nfsstat -c と入力して、クライアント統計情報を調べ、NFS に関係する問題がないか確認します。

   エラーと再送信が発生していないかどうかを調べます。

```
client % nfsstat -c
Client rpc:
calls      badcalls   retrans    badxids    timeouts   waits      newcreds
384687     1          52         7          52         0          0
badverfs   timers     toobig     nomem      cantsend   bufulocks
0          384        0          0          0          0
Client nfs:
calls      badcalls   clgets     cltoomany
379496     0          379558     0
Version 2: (379599 calls)
null       getattr    setattr    root       lookup     readlink   read
0 0%       178150 46% 614 0%     0 0%       39852 10%  28 0%      89617 23%
wrcache    write      create     remove     rename     link       symlink
0 0%       56078 14%  1183 0%    1175 0%    71 0%      51 0%      0 0%
mkdir      rmdir      readdir    statfs
49 0%      0 0%       987 0%     11744 3%
```

この nfsstat -c コマンドの出力例では、合計で 384687 回のコールがあり、そのうち再送信 (retrans) と時間切れ (timeout) が、それぞれ 52 回発生しています。

各フィールドの意味は、以下のとおりです。

表 2–9 nfsstat -c コマンドの出力例

ヘッダー	説明
calls	コール数の合計。
badcalls	RPC によって拒否されたコール数の合計。
retrans	再送信回数の合計。
badxid	1 つの NFS 要求に対して確認が重複した回数。
timeout	タイムアウトが発生したコール数。
wait	使用できるクライアントハンドルがなかったためにコールが待たされた回数。
newcred	確証情報のリフレッシュが求められた回数。

nfsstat -c コマンドの出力の説明と対処方法を以下に示します。

表 2–10 nfsstat -c コマンドを実行して得られる出力と、それに対する処置

問題	処置
retrans 値が全コール数の 5 % を超える	サーバーに要求が届いていません。
badxid 値と badcalls 値がほぼ等しい	ネットワークの動作が遅くなっています。原因を究明してください。問題を解消するには、高速なネットワークにするかサブネットをインストールすることを検討してください。
badxid 値と timeouts 値がほぼ等しい	大部分の要求はサーバーに届いていますが、予測よりサーバーの動作が鈍いことが考えられます。nfsstat -m を使用して予測時間を調べてください。
badxid 値がゼロに近い	ネットワーク上で要求が失われています。mount オプションの rsize 値と wsize 値を小さくしてください。
null が 0 より大きい	null コール数が多いということは、オートマウンタが頻繁にマウントを再試行していることを意味します。マウント時のタイムアウト時間が短かすぎます。オートマウンタコマンド行のマウント時タイムパラメタ (timeo) の値を大きくしてください。

2. NFS マウントしている各ファイルシステムの統計情報を表示します。

   統計情報は、サーバー名とアドレス、マウントフラグ、現在の読み取り、書き込みサイズ、伝送回数、ダイナミック伝送に使われているタイマー情報から構成されます。

```
client % nfsstat -m
/export/home from server:/export/home
 Flags:
vers=2,hard,intr,dynamic,rsize=8192,wsize=8192,retrans=5
 Lookups: srtt=10 (25ms), dev=4 (20ms), cur=3 (60ms)
 Reads:   srtt=9 (22ms), dev=7 (35ms), cur=4 (80ms)
 Writes:  srtt=7 (17ms), dev=3 (15ms), cur=2 (40ms)
 All:     srtt=11 (27ms), dev=4 (20ms), cur=3 (60ms)
```

`nfsstat -m` コマンドの出力例の用語の意味は、以下のとおりです。

表 2–11 `nfsstat -m` コマンドの出力

ヘッダー	説明
srtt	正常時の往復時間。
dev	予測偏差。
cur	現在のバックオフタイムアウト値。

上記のコードの ( ) 内の数字は、ミリ秒で表した実際の時間です。その他の値は、オペレーティングシステムのカーネルによって保持されている未スケール値で、無視してかまいません。応答時間は、ルックアップと読み取り、書き込みの操作の組み合せ時について示されています。`nfsstat -m` コマンドを実行して得られる出力と対処方法を以下にまとめます。

表 2–12 nfsstat -m コマンドを実行して得られる出力と処置

問題	処置
srtt 値が 50 ミリ秒より大きい	マウントポイントの反応が鈍いことが考えられます。前述の手順を参考に、ネットワークとサーバーのマウントポイントを提供しているディスクを調べてください。
"NFS server not responding" というメッセージが表示される	メッセージが表示されないようにして、かつ性能を向上させるには、/etc/vfstab ファイルの timeo パラメタ値を大きくしてください。初期 timeo パラメタ値の 2 倍を基準にしてください。vfstab ファイルの timeo パラメタ値を変更して、nfsstat -c コマンドを実行します。badxid 値を調べて、表 2-10 の nfsstat -c コマンドの推奨処置に従ってください。
Lookups: cur 値が 80 ミリ秒より大きい	要求の処理に時間がかかりすぎています。ネットワークかサーバーのいずれかの動作が遅いことが考えられます。
Reads: cur 値が 150 ミリ秒より大きい	要求の処理に時間がかかりすぎています。ネットワークかサーバーのいずれかの動作が遅いことが考えられます。
Writes: cur 値が 250 ミリ秒より大きい	要求の処理に時間がかかりすぎています。ネットワークかサーバーのいずれかの動作が遅いことが考えられます。

第3章

最適な NFS 性能を得るためのサーバーとクライアントの設定

この章では、最適な NFS 性能を得るための推奨構成について説明します。障害追跡上のヒントについては、第 4 章を参照してください。

調整による NFS 性能の改善

この章では、以下の環境における推奨調整方法を説明します。

- 属性依存の環境

主に 100 ～ 200 バイトの小さなファイルにアクセスするアプリケーションや環境です。ソフトウェア開発環境は、属性依存の環境です。

- データを扱うことの多い環境

主に大きなファイルにアクセスするアプリケーションや環境です。大きなファイルとは、転送に 1 秒以上かかる、最低でも 1 MB ほどのファイルを指します。たとえば、CAD や CAE システムは、大きなデータを扱うことの多い環境です。

システム調整にあたっては、以下の事項を確認してください。

- 37 ページの「ネットワーク条件」
- 40 ページの「ディスクドライブ」
- 49 ページの「CPU」
- 51 ページの「メモリー」
- 54 ページの「スワップ領域の設定」
- 57 ページの「NFS スレッド数の設定 (`/etc/init.d/nfs.server`)」
- 58 ページの「`/etc/system` によるカーネル変数の変更」

NFS サーバーを設定した後、システムの調整を行ってください。NFS サーバーを調整するには、ネットワーク、ディスクドライブ、CPU、メモリーと NFS 性能の関係についての基礎的な知識が必要になります。また、システムを調整するには、どのパラメタを調整した場合にバランスが良くなるかを理解しておく必要があります。

サーバー性能の監視と調整

- 統計情報を収集します。詳細については、**第 2 章**を参照してください。
- 制限を受けている、または、過度に使用されている資源の特定と、それに基づくシステムを再構成します。
- **第 2 章**と、この章で推奨している調整方法の説明を参照してください。
- 再構成後の長期間にわたる性能測定と評価を行います。

NFS サーバーの負荷の分散

NFS 処理は、必ずユーザーレベルのタスクに優先して、オペレーティングシステムのカーネル内部で行われます。

注 – NFS の負荷が大きい場合に、それ以上 NFS サーバーがタスクを実行しようとすると、その処理は遅くなります。1 台の NFS サーバー上で複数のデータベースを実行したり、複数の時分割負荷をかけたりしないでください。

一般的に、メールの送信や印刷などの非対話型処理では、NFS の処理と NFS 以外の処理という 2 つの目的にサーバーが使用されます。これらの非対話型処理には、SPARCprinter (Solaris 2.6 以降のリリースではサポートされません) や、NeWsprint™ ソフトウェアに基づくサンのプリンタによる処理は含まれません。CPU の処理能力に余裕があり、NFS の負荷が小さい場合は、対話型の作業は問題なく行うことができます。

ネットワーク条件

十分なネットワーク帯域幅および可用性を提供することが NFS サーバーを環境設定する上で最も重要な条件となります。これを実現するには、適切な数と種類のネットワークとインタフェースを設定する必要があります。

ネットワークを設定する際は、以下のことを注意してください。

- すべてのクライアントネットワークにわたって、ネットワークトラフィックを効率よく分散し、どのネットワークにも過度に負荷がかからないようにします。
- 1 つのクライアントネットワークに過度の負荷がかかっている場合は、そのセグメントの NFS トラフィックを監視します。
- サーバーに対して最大の要求をしているホストを特定し、負荷を分散させます。
- クライアントを別のセグメントに移します。

ディスクに対する入出力の処理が、頻繁に行われるシステムではない場合は、単にシステムにディスクを追加しても、NFS の性能が向上することはありません。ファイルサーバーのサイズが大きくなるにしたがって、ネットワークが NFS の性能を制限する要素になる可能性が高くなります。システムのバランスを保つためには、ネットワークインタフェースを追加する必要があります。

1 つのネットワークでより多くのデータブロックを転送するのではなく、代表的なクライアントが使用するデータ量の特徴を把握して、複数のネットワークに NFS の読み書きを分散させてください。

データを扱うことの多い アプリケーションに対するネットワーク条件

データを扱うことの多い (Data-Intensive) アプリケーションは、ほとんどの場合はネットワークを必要としません。ただし、ネットワークを使用する場合は、大きな帯域幅が必要となります。

運用環境が以下のいずれかの特徴をもつ場合は、高速なネットワーク環境が必要になります。

- クライアントが全体として毎秒 1 MB を超えるデータ転送速度を必要とする
- 複数のクライアントが、毎秒 1 MB を同時に使用できる必要がある

ネットワークの設定

サーバーの主要アプリケーションが、データを扱うことが多い場合の推奨ネットワーク構成は、以下のとおりです。

- SunFDDI、SunATM などの高速ネットワークを構築する

光ファイバケーブルを使用できない場合は、より対線式の SunFDDI や CDDI、SunFastEthernet の導入を検討してください。SunATM は、SunFDDI と同じサイズのファイバケーブルを採用しています。SunFDDI についての詳細は、『SunFDDI/S3.0 User's Guide』を参照してください。

- 同時に動作する完全に NFS アクティブなクライアント 5 ～ 7 台につき、SunFDDI リング 1 つの割合でネットワークを構築する

データを扱うことの多いアプリケーションで、連続的に NFS 要求をするアプリケーションはほとんどありません。通常のデータを扱うことの多い EDA や地球資源アプリケーションでは、1 リング当たりのクライアント数は 24 ～ 40 になります。

一般的には、操作対象の大きなデータブロックを読み込んで、サーバーに書き戻すという使用方法です。こうした環境では、データを書き戻す処理があるため、書き込みの割合が非常に大きくなる可能性があります。

- Ethernet ケーブルを設置している場合は、アクティブなクライアント 2 台に対して Ethernet 1 つの割合でネットワークを構築し、1 ネットワークあたりのクライアント数を最高でも 4 ～ 6 にする

コミュニティを有用なものにするには多くのネットワークが必要になります。したがって、SPARCserver 1000/1000E、SPARCcenter 2000/2000E、Ultra Enterprise 3000/4000/5000/6000 といったシステムが必要になります。1 ネットワークあたりのクライアント数は、最大でも 4 ～ 6 にしてください。

属性依存のアプリケーション

データを扱うことの多いアプリケーションと比較して、大部分の属性依存 (Atribute-Intensive) のアプリケーションは、高価なネットワークを構築せずに容易に対処することができます。ただし、属性依存のアプリケーションでも、多数のネットワークが必要となります。Ethernet やトークンリングなどの低速のネットワークメディアを使用してください。

ネットワークの設定

サーバーの主要アプリケーションが、属性依存の場合の推奨ネットワーク構成は、以下のとおりです。

- Ethernet またはトークンリングを構築する
- 完全にアクティブな 8 ～ 10 のクライアントに Ethernet 1 つの割合でネットワークを構築する

1 つの Ethernet あたりのクライアント数が 20 ～ 50 を超えると、多数のクライアントがアクティブになったときに、性能が大幅に低下します。Ethernet は、衝突率は高くなりますが、SPECnfs_097 (LADDIS) ベンチマークで約 250 ～ 300 NFS ops/秒の性能を維持することができます。継続して 200 NFS ops/秒を超えることはお勧めしません。

- アクティブな 10 ～ 15 のクライアントに対して、トークンリング 1 つの割合でネットワークを構成する

Ethernet と比較して、大きな負荷に対する性能低下が少ないため、必要に応じて、1 つのトークンリングネットワークに対して、クライアント 50 ～ 80 という構成にすることができます。

複数のユーザークラスが存在するシステム

複数のユーザークラスを持つサーバー用にネットワークを設定するには、異なる種類のネットワークを混在させます。たとえば、SunFDDI とトークンリングはともに、文書画像作成アプリケーション (データを扱うことが多い) や、PC で動作する財務分析アプリケーション (ほとんどの場合は属性依存) をサポートするサーバーに適しています。

多数のネットワークインタフェースカードが必要になることがあるため、選択するプラットフォームは、ネットワークの種類と数によって決まります。

ディスクドライブ

ディスクドライブの使用状況は、NFS サーバーの性能に依存する最も重要な要素です。ファイルシステムからのデータによって、キャッシュを素早く満たすことができない場合は、十分なメモリー構成でも、性能が改善しないことがあります。

性能の低下の原因がディスクにあるかどうかを確認する

NFS 要求のストリームには相関関係がほとんどないため、生成されるディスクアクセスには、ディスクに対する大量のランダムアクセスが含まれます。ランダム入出力の最大回数は、ディスク 1 台当たり 40 ～ 90 回です。

1 台のディスクドライブが、最大ランダム入出力能力の 60 % を超えて使用されると、そのディスクが性能上の障害となります。

性能の低下の原因がディスクにあるかどうかを確認するには、`iostat` コマンドを使用して 1 秒あたりの読み取り・書き込み回数を調べます (14 ページの「NFS サーバーの検査」を参照)。

ディスクボトルネックの緩和

NFS サーバーのディスク帯域幅は、NFS クライアントの性能に最も大きな影響を与えます。最高のファイルサーバー性能を得るには、ファイルシステムのキャッシュに十分な帯域幅とメモリーを用意します。また、読み取りや書き込みの待ち時間も重要で

す。たとえば、NFSop ごとにディスクアクセスが伴うと、ディスクサービス時間が NFSop 待ちの時間に加わるため、ディスク動作が遅くなると、NFS サーバーの動作も遅くなります。

ディスクのボトルネックを緩和するには、以下のガイドラインに従ってください。

- システムのすべてのディスクに入出力負荷を均等に分散する。

特定のディスクの負荷が大きく、他のディスクが能力の低いレベルで動作している場合は、ディレクトリや頻繁にアクセスするファイルを、あまり使用されていないディスクに移動してください。

- 負荷の大きいディスクのファイルシステムを分割し、複数のディスクに分散する。

ディスクを増設することによってディスク容量が増え、ディスクの入出力帯域幅が広くなります。

- NFS クライアントが使用するファイルシステムが読み取り専用であり、変更することのないデータが含まれている場合は、ファイルシステムを複製し、クライアントに対するネットワークからディスクへの帯域幅を広くする。

41 ページの「ファイルシステムの複製の作成」を参照してください。

- 頻繁に使用するファイルシステムデータを、メモリー上でアクセスできるように、オペレーティングシステム内のキャッシュを適切な大きさに設定する。

i ノード (ファイル情報ノード) やファイルシステムのメタデータ (シリンダグループ情報など) 、名前から i ノードへの変換用のキャッシュは、十分な大きさにする必要があります。十分な大きさがない場合は、キャッシュに対するヒットミスでディスクトラフィックが増加します。たとえば 、NFS クライアントがファイルをオープンする場合に、NFS サーバーでは、名前から i ノードへの変換動作が複数回行われます。

ディレクトリ名ルックアップキャッシュ (DNLC) でヒットしなかった場合は、サーバーは、ディスク上のすべてのディレクトリエントリから適切なエントリ名を探します。通常はメモリー上の処理で行われることが、複数回のディスク動作となります。このような場合は、ファイルに関係するページもキャッシュされません。

ファイルシステムの複製の作成

通常、NFS サーバーでは、以下のファイルシステムが使用されます。

- ディスクレスクライアントの `/usr` ディレクトリ

- ローカルツールとライブラリ
- サン製品以外の一般的なパッケージ
- 読み取り専用のアーカイブ版ソースコード

上記のファイルシステムの複製を作成すると、ファイルシステムの性能を向上させることができます。特定の 1 つのファイルシステムに対する要求を処理する場合に、NFS サーバーは、ディスク帯域幅の制限を受けます。データの複製を作成すると、NFS クライアントからデータ方向へのパイプの合計サイズが大きくなります。ただし、複製は、ホームディレクトリからなるファイルシステムなどの書き込み可能なデータ性能の改善には、実用的な手段ではありません。複製は、読み取り専用のデータに使用してください。

▼ ファイルシステムを複製する

1. **複製を作成するファイルまたはファイルシステムを特定します。**
2. **複数のファイルの複製を作成する場合は、1 つのファイルシステムにまとめます。**

   頻繁に使用するファイルを 1 つのディスクにまとめることによって生じる可能性がある性能の低下は、複製を作成することによって補うことができます。

3. **一般的に入手可能なツール (`nfswatch`) を使用して、NFS サーバーグループで最も頻繁に使用されるファイルおよびファイルシステムを確認します。**

   表 A-1 に、`nfswatch` などの性能監視ツールと入手方法を紹介しています。

4. **クライアントがどのように複製を選択するかを決定します。**

   `/etc/vfstab` ファイルにサーバー名を記述し、NFS クライアントからサーバーへの対応を指定します。別の方法として、オートマウンタのマップに全サーバー名を記述することによって、完全に動的な結合を指定することもできます。ただし、この方法を使用すると、クライアントが一部の NFS サーバーに偏る可能性があります。クライアントグループが、専用の複製 NFS サーバーをもつ「ワークグループ」パーティションを実施した場合は、最も予測可能な性能を得ることができます。

5. **新しいデータを配布するための更新スケジュールと方法を選定します。**

   新しい読み取り専用データを配布する計画と方法は、そのデータを変更する頻度によって決定してください。内容が完全に変更されてしまうファイルシステム、たとえば、毎月更新される履歴データを含むファイルは、各マシン上で配付媒体のデータをコピーするか、`ufsdump` と `ufsrestore` を組み合せる方法をお勧めします。ほとんど変更のないファイルシステムは、`rdist` などの管理ツールを使用して対処することができます。

6. **複製サーバー上の古いデータをユーザーが使用したときに、どのような問題が生じるかを検討します。**

   この場合は、Solaris 2.x JumpStart™ 機構と `cron` を組み合わせる方法があります。

キャッシュファイルシステムの追加

キャッシュファイルシステムはクライアントが中心となります。クライアント上のキャッシュファイルシステムを使用して、サーバーの負荷を軽減します。キャッシュファイルシステムを使用した場合は、ファイルはブロックごとにサーバーから読み込まれます。ファイルは、クライアントのメモリーに送られ、ファイルに対する操作は直接行われます。操作されたデータは、サーバーのディスクに書き戻されます。

マウントを行うクライアントに、キャッシュファイルシステムを追加することによって、各クライアントにローカルの複製を作成することができます。キャッシュファイルシステムの `/etc/vfstab` エントリは、以下のようになります。

```
# device    device    mount    FS    fsck    mount    mount
# to mount  to fsck   point    type  pass    at boot  options
server:/usr/dist      cache    /usr/dist     cachefs  3   yes
ro,backfstype=nfs,cachedir=/cache
```

アプリケーションファイルシステムなど、頻繁に読まれるファイルシステムに対してキャッシュファイルシステムを使用してください。それ以外にキャッシュファイルシステムを使用する状況として、遅いネットワーク上でデータを共有している場合があります。複製サーバーとは異なり、キャッシュファイルシステムは、書き込み可能なファイルシステムと組み合わせることができますが、書き込みの割合が高くなるにしたがって性能が低下します。書き込みの割合が高すぎる場合は、キャッシュファイルシステムによって NFS の性能が低下することがあります。

また、使用しているネットワークがルーターによって相互接続された高速のネットワークである場合にも、キャッシュファイルシステムの使用を検討する必要があります。

NFS サーバーが頻繁に更新されている場合は、キャッシュファイルシステムを使用しないでください。キャッシュファイルシステムを使用すると、NFS を介した操作よりも多くのトラフィックが発生します。

キャッシュファイルシステムを監視する

- キャッシュファイルシステムの効果を監視するには、cachefsstat コマンドを使用します (Solaris 2.5 以降で使用可能です)。

  cachefsstat コマンドの構文は以下のとおりです。

```
system# /usr/bin/cachefsstat [-z] パス名
```

  -z は統計情報を初期化します。cachefsstat を実行してキャッシュ性能の統計情報を収集する前に、cachefsstat -z (スーパーユーザーのみ) を実行する必要があります。統計情報は、統計情報が再び初期化されるまでの情報を反映します。

  パス名は、キャッシュファイルシステムがマウントされているパス名です。パス名を指定しないと、マウントされているすべてのキャッシュファイルシステムが対象となります。

  -z オプションを使用しないかぎり、このコマンドを通常の UNIX ユーザーとして使用することができます。

  cachefsstat コマンドの使用例を以下に示します。

```
system% /usr/bin/cachefsstat /home/sam
cache hit rate: 73% (1234 hits, 450 misses)
consistency checks:  700 (650 pass, 50 fail)
modifies: 321
```

  上記の例では、ファイルシステムにおけるキャッシュのヒット率は 30 % よりも高くなっている必要があります。キャッシュのヒット率が 30 % よりも低い場合は、ファイルシステムに対するアクセスが全体的に不規則であるか、キャッシュの大きさが不十分であることを意味しています。

cachefsstat コマンドを実行して得られる統計情報には、キャッシュのヒット率、一貫性の検査の実行数、および変更の数が含まれます。

表 3–1　cachefsstat コマンドを実行して得られる統計情報

出力	説明
cache hit rate	キャッシュの試行の数と成功した数の比率。成功した数と失敗した数も表示されます。
consistency checks	一貫性の検査の実行数。合格した数と、問題があった数も表示されます。
modifies	変更の数 (書き込みと作成が含まれます)。

一貫性の検査の結果は、キャッシュファイルシステムがサーバーに対してデータが有効であるかどうかを検査した結果です。失敗率が高い場合は (15 ～ 20 %)、検査対象のデータが頻繁に変更されていることを意味しています。キャッシュは、キャッシュされたファイルシステムよりも高速に更新することができる可能性があります。一貫性の検査の結果と変更の数を調べることによって、このクライアントが変更を行っているのか、他のクライアントが変更を行っているのかを調べることができます。

変更の数は、クライアントがファイルシステムに変更を行った回数です。この結果を調べることは、ヒット率が低い理由を調べるもう 1 つの方法です。変更の数が多い場合は、通常は一貫性の検査の数が多くなり、ヒット率が低くなります。

cachefswssize と cachefsstat を使用することができます。cachefswssize は、キャッシュファイルシステムで使用されるファイルのサイズの合計を表示します。cachefsstat は、キャッシュファイルシステムの統計情報が記録されている場合に情報を表示します。これらのコマンドを使用して、キャッシュファイルシステムの状態が適切かどうかを確認してください。

ディスクドライブを設定する上での規則

データを扱うことが多い環境や属性依存の環境の場合は、一般的なガイドラインの他に、それぞれの環境に応じたガイドラインがあります。

ディスクドライブを設定するときは、以下のガイドラインに従います。

- アクティブなドライブ数が SCSI の標準ガイドラインを超えない範囲で、性能を低下させることなく、各ホストアダプタのドライブを増設する。

- Solstice DiskSuite を使用し、多数のディスクにディスクアクセス負荷を分散させる。

  詳細は、47 ページの「Solstice DiskSuite または Online: DiskSuite による ディスクのアクセス負荷の分散」を参照してください。

- ディスクの最高速の領域を使用する。

  詳細は、48 ページの「最適なディスク領域の利用」を参照してください。

データを扱うことが多い環境 (Data-Intensive)

データを扱うことが多い環境でのディスクドライブの構成では、以下のガイドラインに従います。

- 逐次的な環境に設定する。
- 最も高速な転送速度のディスクを使用する (可能であればストライプ化する)。
- アクティブなバージョン 3 のクライアント 3 台に対して、1 台の RAID デバイス (論理ボリュームまたはメタディスク) を設定する。または、バージョン 2 のクライアント 4、5 台に対して、1 台のデバイスを設定する。
- Ethernet またはトークンリング上のアクティブなクライアント 1 台に対して、少なくともディスクドライブ 1 台の構成にする。

属性に依存する環境 (Attribute-Intensive)

属性に依存する環境でのディスクドライブの構成では、以下のガイドラインに従います。

- 適切な数の SCSI ホストアダプタ (ディスクアレイなど) に小型のディスクを多く接続する。
- Fast SCSI ホストアダプタ 1 基に対して、4 台から 5 台の (または 8 台から 9 台以下の) ディスクドライブの構成にする。小型のディスクドライブを複数使用することは、大型のディスクドライブを 1 台使用するよりはるかに良い結果が得られます。
- ネットワークのタイプに関係なく、クライアント 2 台に対して少なくともディスクドライブ 1 台の構成にする。
- Fast-Wide SCSI ホストアダプタ 1 基に対して、6 台から 7 台以下の 2.9 GB ディスクドライブの構成にする。

Solstice DiskSuite または Online: DiskSuite によるディスクのアクセス負荷の分散

NFS サーバーでは、ディスクドライブとディスクコントローラに、負荷を効率よく分散できない場合があります 。

負荷のバランスをとるために、以下を行ってください。

- 論理的な使用状況からではなく、物理的な使用状況から負荷のバランスをとってください。Solstice DiskSuite または Online: DiskSuite のストライプ機能とミラー機能により、透過的にディスクドライブに対するディスクアクセスが分散するようにします。

Solstice DiskSuite または Online: DiskSuite のディスクミラー化機能は、同じデータのコピー (2 つまたは 3 つ) にアクセスすることによって、ディスクのアクセス時間を短縮し、ディスクの使用回数を減らします。読み取りを主とする環境では特に有効です。一方、ミラー化によって作成されたディスクに対する書き込みは、通常遅くなります。これは、論理的な 1 つの要求に対して、2 回または 3 回の書き込みを行う必要があるためです。

- ディスクが比較的一杯になっている場合に、ディスクを連結することによって、最低レベルの負荷の分散がなされます。
- データ量の多い環境では、ディスクのスループットを改善し、サービス負荷を分散させるために飛び越し間隔の小さいストライプ処理を行ってください。ディスクのストライプ処理により、アプリケーションの連続した読み取り・書き込み速度が向上します。最初の飛び越し間隔の大きさは、ストライプを構成するディスク 1 台につき 64 KB にします。
- 属性に依存する環境 (ディスクに対するアクセスが不規則な環境) では、デフォルトの飛び越し (1 つのディスクシリンダ) でディスクをストライプ化してください。
- `iostat` と `sar` コマンドを使用して、ディスクドライブの使用状況を調べてください。

ディスクが均等に使用されるようにするには、数回にわたって監視を行い、データを再編成する必要があります。また、ディスクの使用パターンは時間とともに変化します。インストール時には最適に設定していたデータのレイアウトも、時間が経過するにしたがって、非常に効率が悪くなることがあります。ディスクドライブの使用状況を確認する方法についての詳細は、40 ページの「ディスクドライブ」を参照してください。

Solstice DiskSuite または Online: DiskSuite 3.0 によるファイルシステムのログベース化

Solaris 2.4 から Solaris 8 のソフトウェア環境と Online: DiskSuite 3.0 または Solstice DiskSuite を組み合わせることによって、標準の UNIX ファイルシステムをログベース化し、ディスクベースの Prestoserve NFS アクセラレータのように扱うことができます。

メインのファイルシステムのディスクの他に、ディスクの一部 (一般的に 10 MB の大きさ) が書き込みのシーケンシャルログ領域として使用されます。以下の 2 つの利点を持つ、このログベース化によって、Prestoserve NFS アクセラレータと同じ種類の動作が高速化されます。

- デュアルマシンの高可用性 (HA) の構成では、Prestoserve NFS アクセラレータは利用できませんが、ログは共有できます。そのため、このような環境でも使用することができます。
- オペレーティングシステムに障害が起きた場合でも、ログベースのファイルシステムの `fsck` が、ログだけを順次読み取ります。大規模なファイルシステムの場合でも、ほとんど瞬時に読み取りが行われます。

注 – 1 つのファイルシステムに Prestoserve NFS アクセラレータとログを同時に使用することはできません。

最適なディスク領域の利用

ディスク上のデータレイアウトを分析する場合は、ゾーンビット記録方式の採用を検討してください。

サンの 207 MB ディスク以外のすべてのディスクには、回転するディスクに特有な幾何特性を利用し、円盤の縁に最も近い部分に、より多くのデータを詰め込むエンコーディング方式が採用されています。この方式により、通常は、外側のシリンダに対応する下位のディスクアドレスが、内側のアドレスと比較して、性能が 50 % 向上します。

- **若い番号のシリンダにデータを書き込みます。**

  ゾーンビット記録方式のデータレイアウトでは、若い番号のシリンダが高速になります。

  この性能の差は、シリアル転送で顕著ですが、入出力時のランダムアクセスにも影響します。外側のシリンダ (ゼロ) は、読み取り・書き込みヘッドによるアクセスが速くなるだけでなく、サイズも大きくなります。データが分散するシリンダが少なくなることにより、シーク回数が少なくなり、シーク時間も短くなります。

CPU

この節では、CPU (中央演算処理装置) の使用状況を調査する方法と NFS サーバーの CPU を構成するときのガイドラインについて説明します。

▼ CPU の使用状況を調査する

- **`%` プロンプトに対して `mpstat 30` と入力して 30 秒間の平均値を得ます。**

  以下が画面に表示されます。

```
system% mpstat 30
CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0    6   0    0   114   14   25    0    6    3    0    48    1   2  25  72
  1    6   0    0    86   85   50    0    6    3    0    66    1   4  24  71
  2    7   0    0    42   42   31    0    6    3    0    54    1   3  24  72
  3    8   0    0     0    0   33    0    6    4    0    54    1   3  24  72
```

`mpstat 30` コマンドは、各プロセッサの統計情報を表示します。表の各行は、1 つのプロセッサのアクティビティー情報です。最初の行は、システムが最後に起動されてからの全アクティビティー情報で、それ以降の行が、各時間間隔のアクティビティー情報を表しています。すべての値が、1 秒当たりのイベント数に基づく割合を表します。

mpstat 出力の各欄の項目の意味は、以下のとおりです。

表 3–2 mpstat コマンドの出力

出力	説明
usr	ユーザー時間の割合
sys	システム時間の割合
wt	待ち時間の割合
idl	アイドル時間の割合

sys が 50 % を超えている場合は、CPU パワーを大きくして、NFS の性能を改善してください。

NFS サーバーの CPU を構成する場合のガイドラインを以下に示します。

表 3–3 サーバーの CPU を構成する場合のガイドライン

条件	処置
中速の Ethernet またはトークンリングネットワーク、1 ～ 3 個の構成で、主として属性依存の環境である	単一プロセッサで十分です。さらに小規模なシステムの場合は、UltraServer™ 1、SPARCserver 5、SPARCserver 4 のいずれかのシステムで、サーバーとして十分なプロセッサの処理能力が得られます。
中速の Ethernet またはトークンリングネットワーク、4 ～ 60 個の構成で、主として属性依存の環境である	UltraServer 2、SPARCserver 10、SPARCserver 20 のいずれかのシステムを使用してください。
大規模な属性依存の環境であり、SBus とディスクを拡張する十分な容量がある	UltraServer 2、SPARCserver 10、SPARCserver 20 のいずれかのシステムのマルチプロセッサモデルを使用してください。

表 3–3　サーバーの CPU を構成する場合のガイドライン (続き)

条件	処置
大規模な属性依存の環境である	以下のようなデュアルプロセッサシステムを使用してください。 – SPARCserver 10 システムモデル 512 – SPARCserver 20 システム – SPARCserver 1000/1000E システム – Sun Enterprise 3000/4000/5000/6000、3500/4500/5500/6500 システム – SPARCcenter 2000/2000E システム SPARCcenter 2000 システムの 40 MHz/1 MB または 50 MHz/2 MB モジュールのいずれも NFS の負荷に対して問題ありませんが、50 MHz/2 MB モジュールの方がより良い性能が得られます。
データを扱うことの多い環境で、高速なネットワークがある	高速なネットワーク (SunFDDI など) 1 つに対して SuperSPARC プロセッサ 1 基の構成にしてください。
データを扱うことの多い環境であるが、ケーブルに制限があり、Ethernet を使用する必要がある	Ethernet またはトークンリングネットワーク 4 つに SuperSPARC プロセッサ 1 基の構成にしてください。
純粋な NFS 環境である	推奨する台数を超えて、サーバーのプロセッサを増設する必要はありません。
NFS 処理以外の処理をサーバーで行う	プロセッサを増設して、大幅な性能向上を図ってください。

メモリー

NFS はディスク入出力処理の多いサービスのため、低速のサーバーは、入出力が問題になることがあります。メモリーを増設し、ファイルシステムキャッシュを大きくすることによって、この入出力の問題は解消します。

システムは、ファイルシステムのページ待ち状態である場合や、スワップデバイスとの間でプロセスイメージをページングしている場合もあります。NFS サービスは、完全にオペレーティングシステムのカーネル内で動作するため、ページング中にさらにサービスが供給された場合にだけ問題となります。

スワップデバイスで入出力動作が何も行われていない場合は、すべてのページング動作は、NFS 読み取りや書き込み、属性、ルックアップなどのファイル入出力操作に伴います。

NFS サーバーがメモリーを大量に使用するかどうかの調査

NFS サーバーの性能上、ディスクからメモリーへのファイルシステムデータのページングが問題になる場合があります。

▼ NFS サーバーシステムがメモリーを大量に使用するかどうかを調査する

1. `vmstat 30` コマンドを実行してスキャンレートを調べます。

   スキャンレート (`sr`、スキャンされたページ数) が毎秒 200 ページを超える場合は、メモリーが不足しています。システムは、再利用可能な未使用ページを探します。再利用可能なページをキャッシュして、NFS クライアントによる再読み取りが行えるようにします。

2. メモリーを増設します。

   メモリーを増設することによって、同じデータが繰り返し読み取られることがなくなり、サーバーのページキャッシュとのやりとりで NFS 要求を処理することができます。NFS サーバーに必要なメモリーの大きさの計算方法については、53 ページの「メモリー容量の計算」を参照してください。

   最適な性能を得るために必要なメモリー容量は、そのサーバー上で使用されるファイルの大きさの合計値によって異なります。メモリーは、最近読み取られたファイルに対してはキャッシュとして動作します。キャッシュを最も効率的に使用するには、使用するファイルの大きさの合計にできるだけ近い値にします。

   メモリーキャッシュ機能が使用されているため、サーバーが長時間アクティブな場合は、NFS サーバーの未使用メモリーが、0.5 MB ～ 1.0 MB の範囲になる場合があります。メモリーを十分に確保することで、複数の要求を問題なく処理することができます。

   実際に使用されるファイルは、時間とともに変化しますが、全体として使用されるファイルの大きさは比較的一定です。NFS は、ある一定の監視期間に取り扱うファイルに依存して、アクティブなファイルのスライドウインドウを作成します。

メモリー容量の計算

メモリー容量は、一般的なメモリー規則、または条件付きメモリー規則のいずれかの方法に従って求めることができます。

一般的なメモリー規則

以下の一般的なガイドラインに従って、必要なメモリーの容量を計算します。

- 仮想メモリー = RAM (メインメモリー) + スワップ領域
- 5 分規則—メモリーの大きさは、16 MB に、5 分間に 2 回以上アクセスされるデータをキャッシュするためのメモリー容量を加えた値になります。

条件付きのメモリー規則

以下の条件付きのガイドラインに従って、必要なメモリーの容量を計算します。

- 多くのクライアントにユーザーデータを供給するサーバーの場合は、メモリーを最低限の大きさにする。

小規模なコミュニティーでは 32 MB、大規模なコミュニティーでは 128 MB 程度にします。マルチプロセッサ構成では、1 プロセッサ当たり少なくとも 64 MB を用意します。通常、メモリーから受ける恩恵は、データを扱うことの多いアプリケーションよりも、属性依存のアプリケーションの方が大きくなります。

- ファイルを頻繁に使用するアプリケーションに、一時ファイル領域を供給することの多いサーバーの場合は、サーバー上で使用されるアクティブな一時ファイルの大きさの合計の 75 % 程度のメモリー構成にする。

たとえば、各クライアントの一時ファイルの大きさが約 5 MB で、サーバーが完全にアクティブの状態で 20 のクライアントを処理すると予測される場合は、メモリーを以下の大きさにします。

(20 クライアント×5 MB) ÷75 % = 133 MB
簡単にメモリーを構成する場合、最も適当な大きさは 128 MB です。

- 実行可能なイメージだけを供給することの多いサーバーの場合は、ライブラリを含めて、使用頻度の高いバイナリファイルの合計に等しい大きさのメモリー構成にする。

たとえば、/usr/openwin を供給するためのサーバーには、X サーバー、コマンドツール、libX11.so、libview.so、libXt をキャッシュするのに十分なメモリーをインストールします。この NFS アプリケーションは、あらゆるクライアントに同じファイルを繰り返し供給することを通常の仕事としていて、必要なデータを効果的にキャッシュすることができる、より一般的な /home や /src、あるいはデータサーバーと異なります。クライアントは、すべてのバイナリの全ページを必ず使用するわけではないため、頻繁に使用されるプログラムや、ライブラリを保持するために十分なメモリー構成にするのが妥当です。可能であれば、クライアントで Cachefs を使用し、サーバーに対する負荷と、サーバーで必要となるメモリー容量を減らしてください。

- クライアントが DOS PC または Macintosh の場合は、Sun NFS サーバー側のメモリーキャッシュを増設する。

DOS PC や Macintosh システムが行うキャッシュは、UNIX システムのクライアントが行うキャッシュよりも少なくなります。

スワップ領域の設定

NFS サーバーはユーザー処理を実行しないため、スワップ領域が必要になることはほとんどありません。

▼ スワップ領域を設定する

1. 仮想メモリー (メインメモリー + スワップ領域) を最低でも 64 MB の大きさにします (表 3–4 を参照)。
2. システムに障害が発生したときに障害ダンプを保存できるように、メインメモリーの 50 % を緊急用のスワップ領域として設定します。

表 3–4 必要なスワップ領域

RAM の容量	必要なスワップ領域
16 MB	48 MB
32 MB	32 MB
64 MB 以上	なし

Prestoserve NFS アクセラレータ

注 – NFS バージョン 3 でサポートされる機能によって、Prestoserve 機能の必要性が少なくなっています。Perstoserve NFS アクセラレータは、NFS バージョン 2 と合わせて使用した場合に大きな効果が得られますが、NFS バージョン 3 と合わせて使用した場合にはわずかな効果しか得られません。

NFS の性能を向上させる別の手段として、Prestoserve NFS アクセラレータを追加する方法があります。NFS バージョン 2 プロトコルには、あらゆる書き込みを安定した記憶領域に書き込んでから、応答をするという規定があります。この条件は、Prestoserve NFS アクセラレータを使用して、低速のディスクではなく高速の NVRAM に書き込むという形で満たすことができます。

Prestoserve NFS アクセラレータが使用する NVRAM には、以下の 2 種類があります。

- NVRAM-NVSIMM
- SBus

Prestoserve NFS アクセラレータの SBus と NVRAM-NVSIMM のどちらの場合も、以下の処理を行って NFS サーバーの処理を高速化します。

- ファイルシステムを高速に選択する
- 同期入出力時の書き込みデータのキャッシュを行う
- 同期書き込みをディスクに対して行わずに、不揮発性メモリーにデータを格納する

NVRAM-NVSIMM

SBus、NVRAM-NVSIMM のどちらの NVRAM ハードウェアも使用できる場合は、Prestoserve キャッシュとして NVRAM-NVSIMM を使用してください。NVRAM-NVSIMM と SBus ハードウェアは機能的には同じですが、効率性の面で NVRAM-NVSIMM の方が若干優れており、SBus スロットを使用しません。NVRAM-NVSIMM はメモリーに置かれ、キャッシュも SBus ハードウェアにくらべて大きくなります。

NVRAM-NVSIMM Prestoserve NFS アクセラレータは、負荷の大きい NFS クライアントや、入出力バウンドのサーバーの応答時間を大幅に改善することができます。NVRAM-NVSIMM Prestoserve NFS アクセラレータは、以下のシステムにインストールすることができます。

- SPARCserver 20 システム
- SPARCserver 1000/1000E システム
- SPARCcenter 2000/2000E システム

Sun Enterprise 3000/4000/5000/6000、3500/4500/5500/6500 システムの場合は、NFS 性能向上のもう 1 つの手段として、サーバーに接続された SPARCstorage Array の NVRAM をアップグレードします。

Sun Enterprise 3000/4000/5000/6000、3500/4500/5500/6500 システムでは、SPARCstorage Array NVRAM 高速書き込みを行うことができます。`ssaadm` コマンドを使用して、高速書き込みを有効にしてください。

NVRAM SBus

SBus Prestoserve NFS アクセラレータには、1 MB のキャッシュのみを搭載し、SBus に接続します。SPARCserver 1000(E)、SPARCcenter 2000(E)、Sun Enterprise 3000/4000/5000/6000、3500/4500/5500/6500 システム以外の SBus を備えているサーバにインストールすることができます。

SBus Prestoserve NFS アクセラレータは、以下のシステムにインストールすることができます。

- SPARCserver 5 システム
- SPARCserver 20 システム
- Sun Enterprise 1 システム
- Sun Enterprise 2 システム
- SPARCserver 600 シリーズ

パラメタの調整

この節では、NFS スレッド数の設定方法について説明します。さらに、/etc/system ファイルに含まれている、 NFS 性能に関連する主要パラメタの調整について説明します。/etc/system ファイルのパラメタを調整するときは、サーバーの物理メモリーの大きさやカーネルのアーキテクチャーに注意してください。

注 – 調整に問題があると、システムが不安定になり、最悪の場合には、起動ができなくなるなどの問題が発生することがあります。

NFS スレッド数の設定 (/etc/init.d/nfs.server)

性能の向上のために、NFS サーバーを設定する際には、必ず NFS スレッドを設定します。スレッド 1 つは、NFS 要求を 1 つ処理することができます。スレッドプールを大きくすることにより、サーバーは複数の NFS 要求を並行して処理することができます。Solaris 2.4 から Solaris 8 ソフトウェア環境では、デフォルトの設定は 16 であり、望ましい NFS 応答時間は得られません。プロセッサ数とネットワーク数に従って、このデフォルト値を大きくしてください。NFS サーバーのスレッド数は、/etc/init.d/nfs.server 内の nfsd 呼び出し行を編集することによって変更します。

```
/usr/lib/nfs/nfsd -a 64
```

上記のコード例 では、要求時 NFS スレッドの最大割当数を 64 に指定しています。

NFS スレッド数を変更する方法は 3 つあります。本書の構成上の規則に従っているかぎり、どの方法を使用してもほぼ同じ数になります。NFS スレッドの数が余分にある場合も、問題が生じることはありません。

NFS スレッドの数を設定するには、以下の 3 つの方法のうちで最大の値を使用してください。

- アクティブなクライアントプロセス 1 つに対して NFS スレッド数を 2 個にする

通常、クライアントワークステーションがもつアクティブプロセスは 1 個だけです。ただし、NFS クライアントが時分割システムの場合は、多数のアクティブプロセスをもつことがあります。

- CPU 1 個に対して NFS スレッド数を 16 ～ 32 個にする

SPARCclassic や SPARCserver 5 システムでは、NFS スレッド数を 16 個程度にします。60 MHz の SuperSPARC プロセッサを搭載したシステムでは、32 個にします。

- ネットワーク容量 10 M ビットに対して NFS スレッド数を 16 個の割合にする

たとえば、SunFDDI™ インタフェースを 1 つ使用している場合は、スレッド数を 160 に設定し、2 つ使用している場合は、スレッド数を 320 に設定します。

バッファーサイズの確認と変数の調整

カーネルの固定サイズテーブル数は、Solaris ソフトウェア環境の新しいリリースが出るたびに減少しています。現在では、ほとんどのテーブルは動的にサイズ変更されるか、`maxusers` 値とリンクしています。Solaris 2.4 から Solaris 8 のソフトウェア環境では、さらに、DNLC と i ノードキャッシュを増やすための調整が必要になります。また、Solaris 2.4 では、ページャーの調整が必要です。Solaris 2.5、2.5.1、2.6、7、8 のオペレーティング環境では、ページャーの調整は必要ありません。

`/etc/system` によるカーネル変数の変更

オペレーティングシステムのカーネルは、起動時に `/etc/system` ファイルを読み込み、ロード可能なオペレーティングシステムのカーネルモジュールの検索パスを設定して、カーネル変数を設定できるようにします。詳細は、`system(4)` のマニュアルページを参照してください。

注意 – `/etc/system` ファイルにコマンドを記述する場合は、十分に注意してください。`/etc/system` ファイル内のコマンドによって、カーネルの設定が自動的に変更されます。

使用しているマシンが起動せず、`/etc/system` に問題があると思われる場合は、`boot -a` オプションを使用してください。このオプションを指定すると、システムはデフォルトの設定で起動し、起動パラメタの指定を求めます。これには、構成ファイルの `/etc/system` も含まれます。構成ファイル `/etc/system` の指定を求めるプ

ロンプトに対しては、元の /etc/system ファイルのバックアップ用コピーの名前を入力するか、/dev/null と入力してください。ファイルを修正し、ただちにシステムを再起動して、正しく動作するかどうかを確認します。

キャッシュサイズの調整 (maxusers)

プロセステーブルなどの各種テーブルのサイズは、maxusers パラメタ値によって決定します。maxusers パラメタは、以下の形式で /etc/system ファイルに設定します。

```
set maxusers = 200
```

Solaris 2.4 から Solaris 8 のソフトウェア環境では、maxusers は、システムに搭載されているメモリー容量に基づいて動的に設定されます。設定は以下の式で表されます。

```
maxusers = システム内の構成されている RAM (MB)
```

メモリー容量 (MB) は、実際には、起動時にカーネルが使用する 2 MB 程度を除いた値であり、physmem で表されます。最小値は 8、自動設定される最大値は 1024 であり、この最大値は 1 GB 以上のメモリーを搭載したシステムに有効です。ユーザー自身が /etc/system に maxusers を設定することもできますが、設定された値はチェックされ、最大でも 2048 に制限されます。2048 に設定した場合、どのカーネルアーキテクチャでも安全なレベルで使用できますが、大量のオペレーティングシステムのカーネルメモリーを使用することになります。

maxusers から導出するパラメタ

性能に関係するオペレーティングシステムカーネルパラメタである、i ノードキャッシュとネームキャッシュのデフォルト値を以下に示します。

表 3–5　i ノードキャッシュとネームキャッシュのデフォルト値

カーネル資源	変数	デフォルト値
i ノードキャッシュ	ufs_ninode	17 * maxusers + 90
ネームキャッシュ	ncsize	17 * maxusers + 90

バッファーキャッシュの調整 (bufhwm)

/etc/system ファイルに設定する bufhwm 変数は、バッファーキャッシュに割り当てる最大メモリー容量を決定し、KB 単位で指定します。bufhwm のデフォルト値は 0 であり、この設定で、システムはシステムメモリーの 2 % まで使用することができます。バッファーキャッシュに使用可能な大きさは、最大でシステムメモリーの 20 % です。NFS 専用のファイルサーバーで、メモリーが比較的小容量の場合は、10 % 程度にする必要が生じることがあります。大きなシステムの場合、オペレーティングシステムのカーネルの仮想アドレス空間が不足しないように、さらに制限する必要があります。

バッファーキャッシュは、i ノードや間接ブロック、シリンダグループ関係のディスク入出力をキャッシュする目的にのみ使用されます。以下の例では、バッファーキャッシュ (bufhwm) を最大で 10 MB まで確保できるように bufhwm を設定しています。通常、これ以上の値は設定しないでください。

```
set bufhwm=10240
```

バッファーキャッシュの読み取りヒット率 (%rcache) と書き込みヒット率 (%wcache) を表示する sar -b コマンドを実行して (以下のコード例を参照)、バッファーキャッシュを監視することができます。

```
# sar -b 5 10
SunOS hostname 5.2 Generic sun4c    08/06/93
23:43:39 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
Average        0      25     100       3      22      88       0       0
```

1 秒間に 50 以上の著しい回数の読み取りと書き込みが行われる場合、読み取りヒット率 (%rcache) が 90% 以下の場合、あるいは書き込みヒット率 (%wcache) が 65% 以下の場合は、バッファーキャッシュ bufhwn の値を大きくします。

上記の sar -b 5 10 コマンドの出力例では、読み取りヒット率が 90 % 以上、書き込みヒット率が 65 % 以上になっています。

sar コマンドの引数の意味は、以下のとおりです。

表 3–6　sar コマンドの引数

引数	説明
b	バッファーの使用状況を検査します。
5	5 秒おきに検査します (最低でも 5 秒にします)。
10	統計情報を収集する回数です。

システムは、バッファーキャッシュのサイズが許容範囲を超えることを防ぎます。バッファーキャッシュサイズを大きくすると、以下の問題が発生します。

- サーバーが停止します。
- オペレーティングシステムのカーネル仮想メモリーの不足によって、デバイスドライバに障害が発生します。

ディレクトリ名ルックアップキャッシュ (DNLC)

ディレクトリ名ルックアップキャッシュ (DNLC) は、maxusers に基づいてデフォルト値が設定されます。NFS サーバーに多数のクライアントを接続している場合は、キャッシュサイズ (ncsize) を大きくしてください。大幅に性能が改善されます。

- DNLC のヒット率 (cache hits) を調べるには、vmstat -s と入力します。

```
% vmstat -s
...[略]...
79062 total name lookups (cache hits 94%)
16 toolong
```

30 文字より短いディレクトリ名はキャッシュされ、長すぎてキャッシュ不可能なディレクトリ名は報告だけされます。キャッシュされなかった場合は、ファイルを得るためにパス名の要素を検索していく際に、ディレクトリ名を読むというディスクの入出力が必要になります。ヒット率が 90 % よりはるかに低い場合は、注意が必要です。

NFS の性能が、キャッシュのヒット率によって大きな影響を受ける場合があります。getattr と setattr、lookup は、通常、全 NFS コールの 50 %より大きな値になります。要求された情報がキャッシュにない場合は、ディスクアクセスを行うので、read あるいは write 要求にともなう性能の低下が生じます。DNLC キャッシュの大きさを制限する要素は、使用可能なカーネルメモリーの容量です。

特に長い名前を多用していないにもかかわらず、ヒット率（cache hits）が 90 % より低い場合は、ncsize 変数を調整します。ncsize 変数は、DNLC キャッシュの大きさを、キャッシュすることが可能な名前変換、および v ノード変換の数で表します。DNLC エントリ 1 個に使用されるカーネルメモリーは、約 50 バイトです。

▼ ncsize を設定する

1. /etc/system ファイルを開き、ncsize を設定します。maxusers 値に基づき、デフォルトより大きな値を設定します。

   デフォルトでは、NFS 専用のサーバーには多くの RAM は必要ないため、maxusers 値および DNLC 値が小さくなっています。したがって、サイズを 2 倍にしてください。

   ```
   set ncsize=5000
   ```

   ncsize のデフォルト値は以下の式で表されます。

   ncsize (ネームキャッシュ) = 17 * maxusers + 90

2. NFS サーバーのベンチマークを 16000 に設定します。

3. maxusers を 34906 に設定します。

4. システムを再起動します。

   以下を参照してください。

i ノードキャッシュの拡張

メモリー常駐の i ノードは、ファイルシステム内の実体の操作が行われるたびに使用されます。ディスクから読み取られた i ノードは、再び必要になるときのために、キャッシュされます。ufs_ninode は、アイドル状態のノードのリストを UNIX ファイルシステムが保持しようとするサイズです。ufs_ninode を 1 に設定すること

によって、10,000 のアイドルノードが保持されます。アイドルノードの数が ufs_ninode を超えると、アイドルノードを削除することによってメモリーの空き領域が確保されます。

DNLC キャッシュのエントリは、それぞれ i ノードキャッシュのエントリに対応しています。そのため、i ノードキャッシュのサイズを変更する場合は、DNLC キャッシュのサイズも変更してください。また、i ノードキャッシュの大きさは、最低でも DNLC キャッシュと同じ大きさにする必要があります。最高の性能を得るには、Solaris 2.4 から Solaris 8 のソフトウェア環境では DNLC キャッシュと同じ大きさにすることをお勧めします。

動作中のシステムで adb を使用して ufs_ninode を操作し、ただちにその結果を確認することができます。ufs_ninode の最大値は、i ノードが使用するカーネルメモリーの容量によって制限されます。この上限は maxusers = 2048 に対応しており、ncsize では 34906 になります。

カーネルメモリに割り当てられている容量は、sar -k を実行して調べます。

- Solaris 2.4 オペレーティング環境では、i ノード 1 個は、lg_mem pool から 300 バイトのカーネルメモリーを使用します。
- Solaris 2.5.1、2.6、7、8 オペレーティング環境では、i ノード 1 個は、lg_mem pool から 320 バイトのカーネルメモリーを使用します。

Solaris 2.5.1、2.6、7、8 オペレーティング環境では、ufs_ninode は、最低でも ncsize と同じ値になるように自動的に調整されます。ヒット率を高めるために ncsize を調整し、システムがデフォルトの ufs_ninodes 値を使用できるようにしてください。

▼ Solaris 2.4 または 2.5 ソフトウェア環境において i ノードキャッシュを大きくする

i ノードキャッシュのヒット率が 90 %より低いか、DNLC からローカルディスクのファイル入出力負荷の調整要求が出された場合は、以下の操作を行います。

1. i ノードキャッシュの大きさを大きくします。

2. /etc/system ファイルの ufs_ninode を DNLC (ncsize) と同じ値に設定します。

   たとえば Solaris 2.4 ソフトウェア環境では、以下のように設定します。

```
set ufs_ninode=5000
```

   i ノードキャッシュのデフォルト値は、ncsize と同じです。

   ufs_ninode (デフォルト値) = 17 * maxusers + 90

注意 – ncsize より小さな値を ufs_ninode に設定しないでください。
ufs_ninode は、アクティブと非アクティブの i ノードの合計ではなく、非アクティブの i ノード数だけを制限します。

3. システムを再起動します。

読み取りスループットの向上

SunFDDI、SunFastEthernet、SunATM などの高速なネットワークでは、NFS クライアント側の先読み量を増やすことで、NFS の読み取りスループットが向上します。

以下の場合は、先読み値を増やさないでください。

- クライアントのメモリーが不足している
- トラフィックの多いネットワークである
- ファイルアクセスが突発的である

使用可能なメモリーの量が十分に確保できない場合は、先読みは行われません。

デフォルトでは、先読みは 1 ブロック (バージョン 2 では 8 KB、バージョン 3 では 32 KB) に設定されています。先読みを 2 ブロックに設定すると、ファイルから最初の 8 KB を読み取っている間に、次の 16 K バイトがフェッチされます。先読みでは、8 KB 単位で情報をフェッチすることによって、前もって新しい情報を確保しておくことができます。

先読み量を増やすことによって、ある点までは読み取りスループットを向上させることができます。先読み量の最大値は、構成やアプリケーションによって異なります。先読み量が最大値を超えると、スループットが低下する場合があります。通常、先読み値を 8 (8 ブロック) より大きくしても、スループットは改善されません。

注 – 以下の手順では、nfs_nra と nfs3_nra 値は別々に調整することができます。Solaris 2.5、2.5.1、2.6、7、8 のいずれかがクライアントで動作している場合は、nfs_nra の調整が必要になることがあります (NFS バージョン 2)。クライアントからサーバーに、バージョン 3 をサポートしていない旨の通知がある場合は、この調整を行ってください。

▼ 先読み値を大きくする (NFS バージョン 2)

1. NFS クライアントの /etc/system に以下の行を追加します。

```
set nfs:nfs_nra=4
```

2. システムを再起動して、新しい先読み値を有効にします。

▼ 先読み値を大きくする (NFS バージョン 3)

1. NFS クライアントの/etc/system に以下の行を追加します。

   - Solaris 2.6 より前の場合。

```
set nfs:nfs3_nra=6
```

   - Solaris 2.6 の場合。

```
set nfs:nfs3_nra=2
```

   - Solaris 7、8 の場合。

```
set nfs:nfs3_nra=4
```

注 – 先読み値を大きくしすぎると、読み取りのスループットが悪くなります。使用している環境における最適の値を見つけるために、nfs3_nra や nfs_nra の異なる値でベンチマークを行うことをお勧めします。

2. システムを再起動して、新しい先読み値を有効にします。

第4章

障害追跡

この章では、以下のような問題が発生した場合の対処方法について説明します。

調整に関する一般的な障害

調整で問題が発生した場合の対処方法を以下に示します。

表 4–1　一般的な障害と対処方法

コマンド／ツール	コマンドの出力	対処方法
`netstat -i`	`Collins+Ierrs+Oerrs/Ipkets + Opkets` が 2 % を超える 。	Ethernet ハードウェアに問題がないかどうかを調査してください。
`netstat -i`	`Collins/Opkets` が 10 % を超える 。	Ethernet インタフェースを増設して、クライアント負荷を分散させてください。
`netstat -i`	`Ierrs/Ipks` が 25 % を超える。	入力エラー率が高く、ホストでパケットが失われています。ネットワークのハードウェアの帯域幅制限を補償するには、パケットを小さくする、つまり、読み取りバッファーの大きさ (`rsize`)、あるいは書き込みバッファーの大きさ (`wsize`)、またはその両方を 2048 に設定します。この設定は、`mount` コマンドを使用するか、`/etc/vfstab` ファイルで指定します。10 ページの「ネットワークを調べる」を参照してください。
`nfsstat -s`	`readlink` が 10 % を超える。	シンボリックリンクをやめて、マウントをするようにしてください。
`nfsstat -s`	`writes` が 5 % を超える。	Prestoserve NFS アクセラレータ (SBus カードか NVRAM-NVSIMM) をインストールして、最高の性能が得られるようにしてください。55 ページの「Prestoserve NFS アクセラレータ」を参照してください。
`nfsstat -s`	`badcall` が発生する。	ネットワークが過負荷になっている可能性があります。ネットワークインタフェースの統計情報をもとに過負荷になっているネットワークを特定してください。

表 4–1　一般的な障害と対処方法 (続き)

コマンド／ツール	コマンドの出力	対処方法
`nfsstat -s`	`getattr` が 40 % を超える。	`actimeo` オプションを使用してクライアントの属性キャッシュを大きくしてください。また、DNLC キャッシュと i ノードキャッシュが十分な大きさであるかどうかを確認してください。`vmstat -s` を使用して DNLC のヒット率 (cache hits) を調べ、必要に応じて `/etc/system` ファイルの `ncsize` 値を大きくします。61 ページの「ディレクトリ名ルックアップキャッシュ (DNLC)」を参照してください。
`vmstat -s`	ヒット率 (cache hits) が 90 % より低い。	`/etc/system` ファイルの `ncsize` 値を大きくします。
SunNet Manager™、SharpShooter 、NetMetrix などの Ethernet モニター	`load` が 35 % を超える。	Ethernet インタフェースを増設して、クライアント負荷を分散させてください。

クライアント側の問題

クライアント側の問題と対処方法を以下に示します。

表 4–2　問題となるクライアント側の状態

状態	コマンド／ツール	原因	対処方法
NFS マウントを行っているディレクトリを使用すると、"NFS server ホスト名 not responding"というメッセージが表示される、または、コマンドに対する応答が遅い。	nfsstat	ユーザーのパス変数	ローカルファイルシステム、リモートファイルシステムの重要なディレクトリ、リモートファイルシステムのその他のディレクトリの順にディレクトリを指定してください。
NFS マウントを行っているディレクトリを使用すると、"NFS server ホスト名 not responding"というメッセージが表示される、または、コマンドに対する応答が遅い。	nfsstat	NFSにマウントを行っているファイルシステムから実行可能なファイルを実行した。	使用頻度の高いアプリケーションはローカルファイルシステムにコピーして使用してください。
"NFS server ホスト名 not responding"というメッセージが表示される、または、badxid が総コール数の 5 %を超え、badxid = timeout になる。	nfsstat -rc	サーバーが応答する前にクライアントが時間切れになる。	サーバー側で障害となっている問題がないかを調査してください。サーバーの応答時間が改善できない場合は、クライアントの /etc/vfstab ファイルの timeo パラメタ値として、25 か 50、100、200 (単位: 10 分の 1 秒) を試します。変更した場合、その都度 24 時間様子を見て、タイムアウトの発生回数が減るかどうかを確認してください。

表 4–2　問題となるクライアント側の状態 (続き)

状態	コマンド／ツール	原因	対処方法
`badxid` = 0 である。	`nfsstat -rc`	ネットワークが低速である。	`/etc/vfstab` ファイルの `rsize` 値と `wsize` 値を大きくして、相互接続デバイス (ブリッジ、ルーター、ゲートウェイ) に問題がないかどうかを調べてください。

サーバー側の問題

サーバー側の問題と対処方法について以下に示します。

表 4–3　問題となるサーバー側の状態

状態	コマンド／ツール	原因	対処方法
"NFS server ホスト名 not responding" というメッセージが表示される。	vmstat -s または iostat	キャッシュヒット率が 90 %より低い。	DNLC 関連のパラメタを推奨値に設定して、状態が改善されるかどうかを確認してください。改善されない場合は、DNLC 関連のパラメタを再設定します。同様に、バッファーキャッシュ、i ノードキャッシュについても、順に再設定してください。
"NFS server ホスト名 not responding" というメッセージが表示される。	netstat -m または nfsstat	サーバーが要求の到着速度に対応できない。	ネットワークに問題がないかどうかを調査し、ネットワークに問題がない場合は、Prestoserve NFS アクセラレータを追加するか、サーバーをアップグレードしてください。
入出力待ち時間または CPU アイドル時間が長いか、ディスクアクセスが遅い。あるいは、"NFS server ホスト名 not responding" というメッセージが表示される。	iostat -x	入出力負荷がディスクに均等に分散していない。svc_t の値が 40 ms より大きい。	長期 (最高 2 週間) にわたるサンプリングにより、ディスクにかかる負荷を分散させて、必要に応じてディスクを増設してください。同期書き込みには、Prestoserve NFS アクセラレータを追加します。ディスクとネットワークのトラフィックを減らすためには、サーバーとクライアント両方の /tmp に tmpfs を使用してください。さらに、システムキャッシュの効率性を測定し、ディスクにかかる負荷を分散させ、必要に応じてディスクを増設します。

表 4–3　問題となるサーバー側の状態 (続き)

状態	コマンド／ツール	原因	対処方法
リモートファイルにアクセスした時の応答が遅い。	`netstat -s` または `snoop`	Ethernet インタフェースでパケットが失われている。	再伝送が指示された場合は、バッファーサイズを大きくしてください。`snoop` の使用方法については、79 ページの「snoop コマンド」を参照してください。

ネットワーク関連の問題

ネットワーク関連の問題と対処方法を以下に示します。

表 4–4　問題となるネットワーク関連の状態

状態	コマンド／ツール	原因	対処方法
複数のサブネットにマウントされているディレクトリにアクセスしたときの応答が遅い、または "NFS server ホスト名 not responding"というメッセージが表示される。	netstat -rs	NFS 要求が経路指定されている。	サブネット上のクライアントを、サーバーに直接接続したままにしておいてください。
複数のサブネットにマウントされているディレクトリにアクセスしたときの応答が遅い、または "NFS server ホスト名 not responding" というメッセージが表示される。	netstat -s によって不完全または不正なヘッダー、不正なデータ長のフィールド、不正な検査合計値が示される。	ネットワークに問題がある。	ネットワークのハードウェアを調査してください。
複数のサブネットにマウントされているディレクトリにアクセスしたときの応答が遅い、または "NFS server ホスト名 not responding"というメッセージが表示される。または、インタフェースの 1 秒あたりの入出力パケットの合計値が 600 を超える。	netstat -i	ネットワークの過負荷	ネットワークセグメントが非常にビジーになっています。問題が再発する場合は、ネットワークインタフェース (le) の増設を検討してください。
ネットワークインタフェースにおける、 1 秒あたりの衝突が 120 回を超える。	netstat -i	ネットワークの過負荷	ネットワーク上のマシン数を減らすか、ネットワークハードウェアを調査してください。

表 4–4　問題となるネットワーク関連の状態 (続き)

状態	コマンド／ツール	原因	対処方法
複数のサブネットにマウントされているディレクトリにアクセスしたときの応答が遅い、または "NFS server ホスト名 not responding" というメッセージが表示される。	netstat -i	パケット衝突率が高い (Collis/Opkts が 0.10 を超える)	• パケットが壊れている場合は、MUX ボックスが壊れている可能性があります。Network General Sniffer か、他のプロトコルアナライザを使用して原因を突き止めてください。 • ネットワークに負荷がかかりすぎていないかどうかを調査し、ノードが多すぎる場合は、サブネットを作成してください。 • ネットワークのハードウェアを調査してください。10BASE-T のタップ、トランシーバ、ハブに問題がある可能性があります。ケーブルの長さと終端に問題がないかどうかを調べてください。

付録 A

NFS 性能監視ツールと
ベンチマークツールの使用方法

この付録では、サーバーの NFS とネットワークの性能を監視するためのツールについて説明します。監視ツールは、性能向上に向けた調整に役立つ情報を提供します。詳細は、第 2 章と第 3 章を参照してください。

監視ツールについての詳細は、そのツールに関するマニュアルページを参照してください。サン以外のツールについては、製品に付属しているマニュアルを参照してください。

また、この章では、SPEC SFS 2.0 という NFS ファイルサーバーのベンチマークツールについても説明します。

NFS 監視ツール

NFS の動作と性能を監視するためのツールを以下に示します。

表 A–1　NFS 動作と性能の監視ツール

ツール	機能
iostat	ディスク入出力などの入出力統計情報を提供します。
nfsstat	カーネルの NFS や RPC (遠隔手続き呼び出し) インタフェースなどの NFS 統計情報を提供します。統計情報の初期設定に使用することもできます。
nfswatch	ファイルシステム別に NFS トランザクションを提供します。nfswatch はフリーソフトウェアです。近くの ftp サイトから入手してください。
sar	CPU の使用状況やバッファーの状態、ディスクドライブやテープドライブなどの動作状況を提供します。
SharpShooter*	障害となっている問題を特定し、クライアントとサーバーに NFS 負荷を均等に分散します。アプリケーションの分散状況を示し、サーバーにネットワークトラフィックを均等に分散させます。ユーザーまたはグループ別のディスク使用状況も提供します。
vmstat	ディスクの動作状況を含む、仮想メモリーの統計情報を提供します。

* Network General Corporation (旧 ATM Technology)

他のネットワークユーティリティやネットワーク監視ユーティリティについては、購入先にお問い合わせください。

ネットワーク監視ツール

NFS 関係のネットワークの性能を監視するためのツールを以下に示します。

表 A–2　ネットワーク監視ツール

ツール	機能
snoop	Ethernet の指定パケットに関する情報を表示します。
netstat	ネットワーク関連のデータ構造体の内容を表示します。
ping	ネットワークホストに ICMP ECHO_REQUEST パケットを送信します。
NetMetrix Load Monitor	リアルタイムまたは特定の時間範囲でネットワークの負荷を監視できます。負荷情報として、時間や発信元、宛先、プロトコル、パケットの大きさを提供します。
SunNet Manager	ネットワーク装置の監視と障害追跡を行う管理・監視ツールです。
LAN アナライザ: Network General Sniffer, Novell/Excelan Lanalyzer	パケットの分析を行います。

snoop コマンド

snoop コマンドを実行すると、サンのシステムはネットワーク監視装置となります。snoop コマンドは、特定の数のネットワークパケットを確保するため、クライアントからサーバーへの呼び出しを追跡して、パケットの内容を表示することができます。パケットの内容をファイルに保存して、後で調査することもできます。

snoop コマンドは以下を行います。

- パケットを記録・表示します。
- ネットワークの NIS などの RPC 応答時間の調査に使用可能な、正確なタイムスタンプを提供します。

■ パケットとプロトコルの情報を見やすく表示します。

snoop コマンドのパケット表示形式には、1 行で表示する要約形式と拡張形式の 2 つがあります。要約形式では、最上位のプロトコルに関するデータだけが表示されます。たとえば、NFS パケットであれば NFS 情報だけが表示され、その下の RPC (遠隔手続き呼び出し) や UDP (ユーザーデータプロトコル) 、IP (インターネットプロトコル) 、ネットワークフレーム情報は表示されません。こうした情報を表示する場合は、コマンドに詳細 (-v または -V) オプションを指定します。

snoop コマンドは、データリンクプロバイダインタフェース (DLPI) のパケットフィルタとバッファーモジュールの両方を使用し、ネットワークを介してやりとりされるパケットを効率良く捕捉します。

任意の 2 台のシステム間の、全トラフィックを表示または捕捉するには、その 2 台以外のシステムで snoop を実行してください。

無制限のパケット確保では、フィルターが無効になり、使用しているシステム宛てのものかどうかに関係なく、サブネットのあらゆるパケットを監視することができます。使用しているシステム宛てではないパケットを、間接的に監視することもできます。無制限モードでは、スーパーユーザーでのみ使用することができます。

snoop はパケット解析ツールであるため、サブネットを構築する場合に特に有用となります。snoop コマンドを実行して得られる出力を使用し、負荷統計情報を蓄積するスクリプトを実行することができます。また、このコマンドには、パケットヘッダーを取り出す機能があり、この情報を利用してパケットをデバッグしたり、非互換の問題の原因を突き止めたりすることができます。

snoop コマンドの引数は以下のとおりです。

表 A–3 snoop コマンドの引数

引数	説明
`-i pkts`	pkts ファイルに確保されているパケット情報を表示します。
`-p99, 108`	確保したファイルから読み出すパケット範囲の指定です。パケット番号 99 から 108 のパケットの情報が表示されます。確保したファイルの先頭パケットのパケット番号は 1 です。
`-o pkts.nfs`	pkts.nfs ファイルに表示中のパケット情報を保存します。

表 A–3　snoop コマンドの引数 (続き)

引数	説明
rpc nfs	RPC 呼び出しのパケットまたは NFS プロトコルの応答パケットを表示します。nfs の後に、/etc/rpc の RPC プロトコル名かプログラム番号オプションが続きます。
and	2 つのブール値の論理和をとります。たとえば、sunroof boutique は、sunroof and boutique と同じです。
-v	詳細モードの指定です。パケット 101 のヘッダーの詳細情報を表示します。このオプションは、特定のパケットに関する情報が必要なときに使用します。

確保したファイル内の選択したパケット情報の表示

統計情報には、読み取り要求を出しているクライアントが示されます。左側の列に約 4 マイクロ秒の精度で秒数が表示されます。

読み取りまたは書き込み要求を発行したときに、サーバーが時間切れにならないようにしてください。サーバーが時間切れになると、クライアントは要求を再送信する必要があり、クライアントの IP コードによって、書き込みブロックがより小さな UDP ブロックに分解されます。デフォルトの書き込み時間は 0.07 秒です。時間切れの値は、mount コマンドで変更することができます。

コード例 A–1　snoop -i pkts -p99,108 コマンドの出力例

```
# snoop -i pkts -p99,108
99   0.0027   boutique -> sunroof     NFS C GETATTR FH=8E6C
100   0.0046   sunroof -> boutique     NFS R GETATTR OK
101   0.0080   boutique -> sunroof     NFS C RENAME FH=8E6C
MTra00192 to .nfs08
102   0.0102   marmot -> viper          NFS C LOOKUP FH=561E
screen.r.13.i386
103   0.0072   viper -> marmot         NFS R LOOKUP No such file
or directory
104   0.0085   bugbomb -> sunroof    RLOGIN C PORT=1023 h
105   0.0005   kandinsky -> sparky    RSTAT C Get Statistics
106   0.0004   beeblebrox -> sunroof  NFS C GETATTR FH=0307
107   0.0021   sparky -> kandinsky    RSTAT R
108   0.0073   office -> jeremiah        NFS C READ FH=2584 at
40960 for 8192
```

- パケットの詳細情報を表示するには、以下のように snoop コマンドを使用します。

```
# snoop -i pkts -v 101
```

snoop -i pkts -v 101 コマンドは、パケット 101 に関する詳細情報を表示します。

NFS パケットを表示するには、以下のように入力します。

```
# snoop -i pkts rpc nfs and sunroof and boutique
1   0.0000   boutique -> sunroof    NFS C GETATTR FH=8E6C
2   0.0046    sunroof -> boutique   NFS R GETATTR OK
3   0.0080   boutique -> sunroof    NFS C RENAME FH=8E6C MTra00192 to .nfs08
```

この例では、sunroof システムと boutique システム間の NFS パケットを表示しています。

- 確保したパケット情報を新しいファイルに保存するには、以下のように入力します。

```
# snoop -i pkts -o pkts.nfs rpc nfs sunroof boutique
```

snoop コマンドの使用法およびオプションについての詳細は、snoop のマニュアルページを参照してください。

SPEC System File Server 2.0

SPEC System File Server (SFS) 2.0 は、NFS ファイルサーバーのスループットと応答時間を計測します。これは、097.LADDIS から成るテストベンチマーク群です。異なるアプリケーション環境下の 1,000 以上の NFS サーバーの調査結果に基づいて開発された、更新された作業負荷情報が含まれます。サーバーの技術進歩により、SFS 1.1 で使用されていたときよりも、作業負荷はより大きく、応答時間のしきい値はより低くなっています。このような変更と、その他の変更により、 SPEC SFS 2.0 の結果と SFS 1.1 や SFS 1 の結果とを比較することはできません。

また、一般的なコードの改善により、SPEC SFS 2.0 には以下の機能が追加されました。

- NFS バージョン 2 およびバージョン 3 の結果の計測
- TCP サポートの追加 (TCP または UDP のいずれかがネットワークトランスポートに使用される)
- 実際の NFS の作業負荷に応用できる複雑な操作が可能
- インタフェースの向上

097.LADDIS では、以下の基準ポイントが考慮されます。

- NFS 操作スループット

  テスト対象のサーバーが、指定されたミリ秒数の間に完了することが可能な、最高 NFS 操作回数です。NFS 操作回数が多いほど、より多くのユーザーにサービスを提供することができます。

- 応答時間

  NFS クライアントが、NFS 要求に対する応答を、テスト対象のサーバーから受け取るのに要する平均時間です。クライアントが認識するサーバーの速さは、この応答時間です。

LADDIS は、テスト対象のサーバーの性能が、あるレベル以下になるまで徐々に作業負荷を大きくできるように設計されています。そのレベルは、50ms を超える平均応答時間と定義されています。この制限は、NFS 操作において、応答時間が 50ms 以下のときの 1 秒あたりの最高のスループットを導出するときに適用されます。

作業負荷とともにスループットが高くなり続けるかぎり、50ms 時のスループットが報告されます。しかし、多くの場合、スループットは、応答時間が制限の 50ms を下回ったときから低下し始め、この後に示すような形式の表によって、最高のスループット時の応答時間が報告されます。

097. LADDIS ベンチマーク

SPEC SFS 1 (097.LADDIS) ベンチマークは、アプリケーションの抽象化と NFS の操作群、NFS 操作要求率に基づく総合的な NFS の作業負荷テストです。このベンチマークによって生成される作業負荷は、NFS プロトコルレベルで集中的なソフトウェア開発環境をエミュレートします。LADDIS は、サーバーに対して直接 RPC 呼び出しを行い、実際に使用されている NFS クライアントの差を排除します。結果として、操作群や作業負荷の制御が簡単になるため、ベンダー同士の結果比較の際に有用です。ただし、この方法は同時に、キャッシュファイルシステムクライアントのように、個々のクライアントの持つ特長を隠すという側面もあります。

NFS 操作群の各操作の割合を以下にまとめます。示した値は、各操作の呼び出し数の相対値です。

表 A–4　呼び出し別の NFS 操作群

NFS 操作	割合
Lookup	34
Read	22
Write	15
GetAttr	13
ReadLink	8
ReadDir	3
Create	2
Remove	1
Statfs	1
SetAttr	1

NFS ファイルシステム用の LADDIS ベンチマークでは、書き込み操作が 15 % の操作群が使用されています。このため、実際の NFS クライアントが 1 ～ 2 % の書き込み操作しか行わない場合は、性能は低く見積もられることになります。実際の操作が、操作群の割合に近いほど、NFS 操作の最大スループットは基準としてより信頼性が高くなります。

LADDIS ベンチマークを行うには、NFS 負荷ジェネレータとして、テスト対象のサーバーに少なくとも 2 台のクライアントが、独立したネットワークで接続されている必要があります。1 つのネットワークでは、サーバーの最高の性能に達する前に飽和することがあるため、複数のネットワークをサポートすることは非常に重要です。1 台のクライアントを、LADDIS 第 1 負荷ジェネレータとして指定し、そのクライアントによって、すべての負荷生成クライアントでの LADDIS 負荷生成コードの実行を制御します。一般的に、ベンチマークも、第 1 負荷ジェネレータが制御します。また、第 1 負荷ジェネレータは、作業負荷のそれぞれのポイントでスループットと応答時間データを収集し、結果を生成することもできます。

応答時間を短くするには、NFS サーバーに NVRAM-NVSIMM Prestoserve NFS アクセラレータを追加してください。NVSIMMは、直接高速のメモリーサブシステムに記憶領域を確保しますから、待ち時間が大幅に短くなり、より少ないディスク入出力で、求めるレベルの性能を得ることができます。

NVSIMM との間では、余分なデータコピーがやりとりされるため、最高のスループットは抑えられます。しかし、NFS の負荷によって最高のスループットが維持されることはないため、NVSIMM を使用して応答時間を短くすることをお勧めします。Prestoserve NFS アクセラレータについての詳細は、55 ページの「Prestoserve NFS アクセラレータ」を参照してください。

索引

と

に

ね

は

ひ

ふ

め

も

よ

ら