NEC

NEC Scalable Technology File System (ScaTeFS)

運用の手引

SX-Aurora TSUBASA

輸出する際の注意事項

本製品（ソフトウェアを含む）は、外国為替および外国貿易法で規定される規制貨物（または役務）に該当することがあります。

その場合、日本国外へ輸出する場合には日本国政府の輸出許可が必要です。

なお、輸出許可申請手続きにあたり資料等が必要な場合には、お買い上げの販売店またはお近くの当社営業拠点にご相談ください。

は　し　が　き

本書は、NEC Scalable Technology File System の構築手順や運用管理、効率的な IO の出し方などについて説明したものです。

備考

(1) Linux は Linus Torvalds の国およびその他の国における登録商標あるいは商標です。

(2) Red Hatは米国およびそのほかの国において登録されたRed Hat, Inc. の登録商標です。

(3) CLUSTERPROは日本電気株式会社の登録商標です。

(4) Windowsは、米国Microsoft Corporationの米国およびその他の国における登録商標です。

(5) StoragePathSavior は日本電気株式会社の日本における登録商標です。

(6) InfiniBand は、InfiniBand Trade Association の商標またはサービスマークです。

(7) Mellanox®はメラノックステクノロジーズ社のイスラエルおよびその他の国における登録商標または商標です。

(8) Dockerはアメリカ合衆国およびその他の国におけるDocker, Inc.の商標です。

(9) 製品名などの固有名詞は、各メーカーの登録商標または商標です。

本書の読み進め方

本書は、次の構成となっています。章ごとに対象読者の範囲は異なっており、表の一番右の列にその範囲を示しています。

章	タイトル	内容	対象読者
1	NEC Scalable Technology File System の概要	ScaTeFSの概要について記載しています。	システム管理者 一般利用者
2	ネットワークの構築	ScaTeFSを利用するためのネットワークの構築手順について記載しています。	システム管理者
3	IOサーバの HW 構成	IOサーバのHW構成を記載しています。	システム管理者
4	クライアント側のHW構成	クライアントのHW構成を記載しています。	システム管理者
5	IOサーバの構築	IOサーバの構築手順を記載しています。	システム管理者
6	Linuxクライアントの設定	Linuxクライアントの構築手順を記載しています。SX-Aurora TSUBASAから利用するための手順はこちらで説明しています。また、トラブル発生時にクライアントに出力されるログの解説もしています。	システム管理者
7	SX-ACEクライアントの設定	SX-ACEクライアントの構築手順を説明しています。	システム管理者
8	DockerのコンテナからScaTeFSを利用する際の設定	Dockerコンテナを利用する際の設定について記載しています。	システム管理者
9	運用管理	QUOTAなど運用管理に必要な機能を記載しています。	システム管理者 一般利用者
10	メンテナンス	バックアップや整合性チェックと修復の手順などを記載しています。	システム管理者
11	利用者向けの利用、設定方法	ScaTeFS上のファイル管理の実装を理解し、IOを効率化するためのTIPSや、SX-Aurora TSUBASAからVEダイレクト機能を使うための手順を記載しています。	システム管理者 一般利用者
12	諸元	諸元表を記載しています。	システム管理者 一般利用者

関連説明書

- SX-Aurora TSUBASA インストレーションガイド
- HPC ソフトウェアライセンス管理説明書
- SX-Aurora TSUBASA Fortran コンパイラユーザーズガイド
- NEC Network Queuing System V (NQSV) 利用の手引 [リファレンス編]
- NEC Network Queuing System V (NQSV) 利用の手引 [管理編]
- SXクロスソフトウェア ノードロックライセンス導入ガイド (※)

SX-Aurora TSUBASA向けの各種説明書は以下のWebサイトで参照できます。

https://sxauroratsubasa.sakura.ne.jp/documentation/

※SX-Aurora TSUBASAシステムでは、ノードロックライセンスを使用しないため「SXクロスソフトウェア ノードロックライセンス導入ガイド」は参照しません。

用語定義・略語

用語	意味
ScaTeFS	「NEC Scalable Technology File System」の略称。
IOサーバ	ScaTeFSを構成するサーバ。最低２台必要。
VE	Vector Engine。従来SXアーキテクチャをベースとするNEC固有のベクトル計算用PCIeカード。VHに接続して使用する。
VH	Vector Host。VEを接続するXeon x86-64アーキテクチャのマシン。
標準モデル向けIOサーバv1	2台のサーバあたり4台のストレージで構成するIOサーバ Express製品名：Express5800/R120e-2M
小規模モデル向けIOサーバv1	2台のサーバあたり2台のストレージで構成するIOサーバ Express製品名：Express5800/R120e-2M
標準モデル向けIOサーバv3	2台のサーバあたり2台のストレージで構成するIOサーバ Express製品名：Express5800/R120g-2M
標準モデル向けIOサーバv4	2台のサーバあたり2台のストレージで構成するIOサーバ Express製品名：Express5800/R120h-2M
標準モデル向けIOサーバv4+	2台のサーバあたり2台のストレージで構成するIOサーバ Express製品名：Express5800/R120h-2M 2nd-Gen
標準モデル向けIOサーバv4++	2台のサーバあたり2台のストレージで構成するIOサーバ Express製品名：Express5800/R120h-2M 3rd-Gen
ルートIOサーバ	IOサーバの一種。mkfsを実行するサーバであり、クライアントがマウントする際のマウント先のサーバ。システム運用中においては、他のIOサーバと特に相違はなく同様な処理を行う。
IOサーバデーモン	IOサーバ上で動作するデーモン
仮想ファイル	仮想ファイルシステム上に作成されたファイル。ScaTeFS上のレギュラーファイル。
実ファイル	複数のサーバに跨った仮想ファイルの断片。実際には、実ファイルシステム上のファイルのこと。
仮想ファイルシステム	複数のIOターゲットにより構成されるクライアント見えのファイルシステム。ScaTeFSそのもの。
実ファイルシステム、IOターゲット	仮想ファイルシステムを構成する基本単位。各IOサーバ配下に作成される。実態は、Linuxで使用可能な通常のファイルシステム。
フェアシェアI/Oスケジューリング	ユーザ毎、またはノード毎にサーバ資源を公平に割り当てる機能のこと。

用語	意味
ストレージグループ	NL SAS、SSDなどアクセス速度の異なる媒体を同一ファイルシステム内のディレクトリ単位に目的別に割り当てる機能のこと。たとえば、あるディレクトリ配下は、SSDで構成されており速いためテンポラリ領域として使用。一方、他のディレクトリはNL SASにより構成されているため安価であり、大容量のファイルを格納しやすいなど。
プリマップ	指定されたファイルサイズに相当する個数の実ファイルを各実ファイルシステム上に予め生成する機能のこと。同ファイルへの並列writeを行う場合、実ファイル生成のオーバーヘッドをプリマップにより低減することが目的。scatefs_premap(1)を使用。
並列I/O	複数の計算ノードを使用して並列にデータを転送することにより、１つのファイルへの書き込み、読み込みを行うこと。巨大ファイルへのI/O効率を上げることが主たる目的。
10GbE	10Gigabit Ethernetの略。
GbE	Gigabit Ethernetの略。
IB	InfiniBandの略。
TOE	TCP Offload Engineの略。 複雑なTCP機能をハードウェア上に実装することにより、CPUの負荷を軽減する仕組み。
NIC	Network Interface Cardの略。 他ノードと通信するためのハードウェア。
HCA	Host Channel Adapterの略。 InfiniBandを使用して他ノードと通信するためのハードウェア。
IPoIB	IP over InfiniBandの略。 InfiniBandネットワーク上でIPを動作させること。
Verbs	InfiniBandのネイティブAPI。IPoIBを使った通信よりも高速な通信が可能となる。
bonding	複数のNICやHCAのポートを仮想的に束ねて使うことで冗長化や負荷分散を行う仕組み。
ib-bonding	IPoIBでbondingを行う仕組み。
サブネットマネージャ	IBのサブネットの管理、制御を行うソフトウェア。IBスイッチのベンダが提供しているものや、OpenSMがある。
QoS	Quality of Serviceの略。 本文書ではIB環境やサブネットマネージャのQoS機能を指す。
バーチャルレーン	IBにおいて１つの物理リンク上に複数の仮想的なリンク(レーン)を作って扱う仕組み。
サービスレベル	IBのパケットをバーチャルレーンにマッピングするための値。

用語	意味
標準モデル	SX-ACE(64台以上のノードで構成されるクラスタシステム)、またはLinux(RHEL)をクライアントとするシステム
小規模モデル	SX-ACE Lite(16もしくは32台のノードで構成されるクラスタシステム) 、またはLinux(RHEL)をクライアントとするシステム
ScaTeFS IBライブラリ	VHを含むスカラマシンから、InfiniBandを使用してユーザ空間上でScaTeFS IOを高速に処理するライブラリ。
ScaTeFS VEダイレクトIBライブラリ	VEからInfiniBandを使用してユーザ空間上でScaTeFS IOを高速に処理するライブラリ。
ScaTeFS InfiniBand 高速IOライブラリ	スカラマシンにおいてScaTeFS IBライブラリ、VEではScaTeFS VEダイレクトIBライブラリを使用したScaTeFS IOをユーザ空間で高速に処理する機能。
制御通信	ScaTeFSクライアントが内部的に行うIPoIBを使った通信。 IB Verbsによる通信を確立もしくは切断するために行う。
NUMA	Non-Uniform Memory Accessの略。 メモリ共有型のマルチプロセッサシステムの実装方式の１つで、プロセッサからのメモリへのアクセス速度が均一にならないような方式。
DDN社	DataDirect Networks社
SFA7990XE	DDN社が提供するストレージアプライアンス製品
VM	Virtual Machineの略。 コンピュータの動作をエミュレートするソフトウェアやフレームワーク。

目 次

表目次

図目次

第1章　NEC Scalable Technology File System の概要

1.1　NEC Scalable Technology File Systemとは

NEC Scalable Technology File System(ScaTeFS: スケートエフエス)は、HPCシステムの大規模化、データの大容量化に対応できる分散・並列ファイルシステムです。ファイルシステム全体を管理するサーバは存在せず、データ、メタデータともに複数のIOサーバへ一様に分散配置し、read/writeリクエストの処理、ファイル、ディレクトリの生成や属性の参照／更新などファイルシステムとしての基本的な機能すべてをリクエスト毎に各IOサーバへ分散して処理することで、負荷分散、スケールアウトを実現します。これにより、システム全体のスループットを向上させ、巨大ファイルへの並列I/Oを行うことができる機能を提供します。一方で、ScaTeFSはPOSIX準拠のユーザインターフェースであり、プログラムを変更することなくSX-Aurora TSUBASAへの移行が可能です。

さらに、SX-Aurora TSUBASAではアーキテクチャに最適化されたIOが可能なScaTeFS InfiniBand 高速IOライブラリが利用可能です。

また、フロントエンドマシンやPCクラスタ、SX-ACEなどからも同一ファイルシステムを共有でき、ヘテロジニアスな環境にも対応できます。

また、システム運用中のIOサーバ／ストレージの追加やIOサーバ障害時のフェイルオーバなど運用継続性の向上、さらにIBネットワークおよび10GbEネットワークを基盤とするため、大規模なFC-SAN環境の構築は不要となり、システム管理コストを低減できます。

1.2　基本コンポーネント

ScaTeFS は、図 1-1 のように大きく分けて下記の４つより成っています。

- クライアント(計算ノード)
- ネットワーク
- IOサーバ
- ストレージ

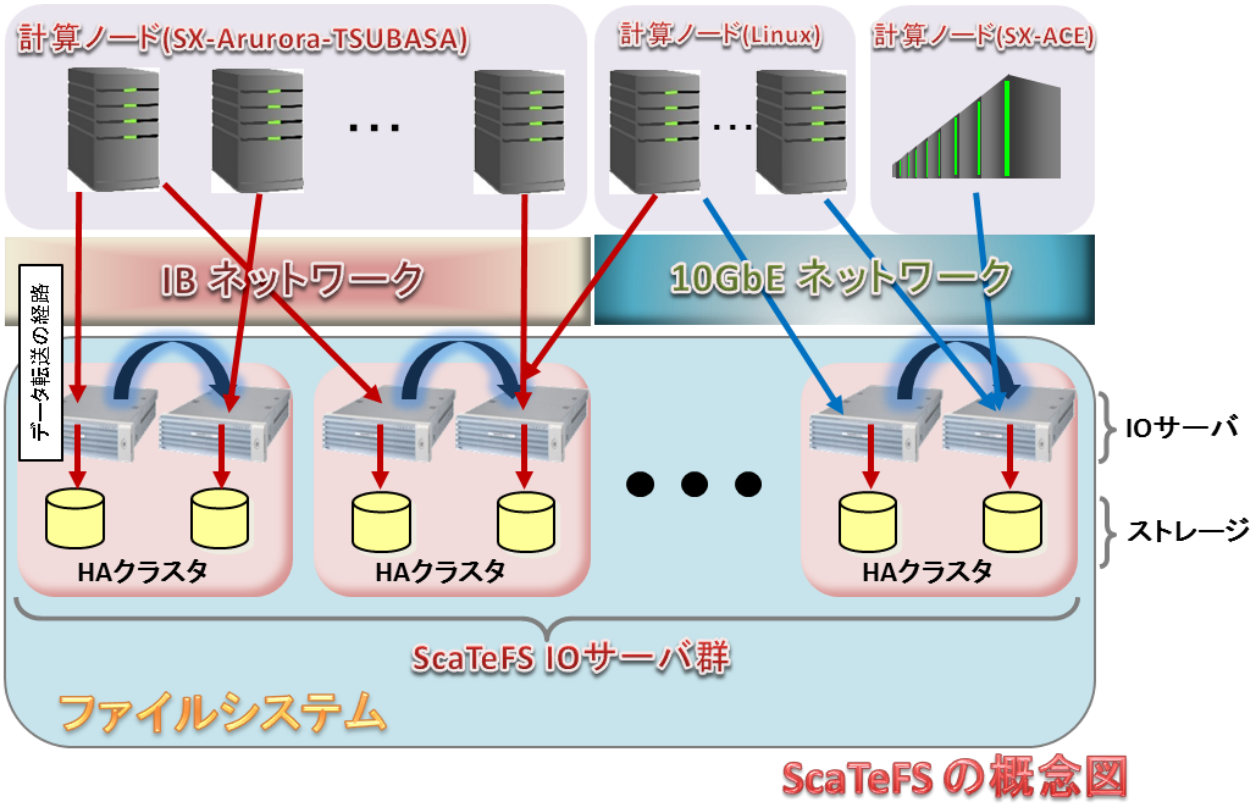


図 1-1　ScaTeFS の概念図

1.2.1　クライアント

SX-Aurora TSUBASA、Linuxマシンの計算ノードとフロントエンドマシン、SX-ACEをクライアントとすることができます。

1.2.2　ネットワーク

SX-Aurora TSUBASAの場合はIBネットワークを通して、Linuxマシンの場合は10GbEまたはIBネットワークを通して、SX-ACEの場合は10GbEを通して、クライアントとIOサーバ間においてデータをやり取りします。

SX-Aurora TSUBASAでは、IBネットワークのみが使用できます。10GbEは使用できません。

1.2.3　IO サーバ

クライアントからのリクエストに基づき、配下に接続されたストレージに格納されるファイルデータの断片、メタデータを操作します。なお、IOサーバの障害時でもフェイルオーバにより運用を継続できるよう２台のIOサーバによりHAクラスタを構成します。

1.2.4　ストレージ

各IOサーバ配下に接続され、ファイルデータの断片、メタデータを格納します。

1.3　代表的な機能

ScaTeFS の主な機能は、以下のとおりです。

(1)　大容量かつ高速な I/O 機能の提供

- 高スループット確保のための複数のIOサーバによる負荷分散
- IOサーバ数に比例した広大な容量のファイルシステムを作成可能
- 性能、容量アップのための運用を停めないIOサーバ、ストレージの追加機能
- 巨大ファイルの作成機能
- 並列I/Oのため、複数の計算ノードから同一ファイルを同時に更新できる機能
- 処理効率化のためのデータ、メタデータのキャッシュ機能
- DCB(Data Center Bridging)によるロスレス通信機能(10GbE)
- サービスレベルの指定によるQoS機能(IB)
- 大IOをユーザ空間で軽量かつ高速に処理するScaTeFS IBライブラリ(9.11参照)

(2)　可用性の確保

- IOサーバ障害時、ジャーナリングによるデータ保全のできるIOサーバのフェイルオーバ機能
- IOサーバとストレージ間のパス障害時のためのパスフェイルオーバ機能
- IOサーバのネットワークインターフェース障害への対処

(3)　運用、システム構築の容易さ

- 運用を停めないIOサーバのメンテナンスが可能
- １つのコマンド実行のみにより、複数のIOサーバに跨るファイルシステムを構築可能
- ファイルシステムの整合性チェック、修復機能

- ログ、統計情報の収集機能

(4) 多様な環境、用途への対応

- SX-Aurora TSUBASA、フロントエンドマシン、PCクラスタ、SX-ACEなどから同一ファイルシステムを利用可能
- フェアシェアI/Oスケジューリングによる多数ユーザの利用環境におけるフェアなI/O処理
- ストレージグループによる多様なストレージの使い分け
- SX-Aurora TSUBASAの各種モデルに柔軟に対応
- IBライブラリを使用したユーザ空間でのScaTeFS IOの高速処理
- 従来のSX-ACE標準モデルや、SX-ACE Liteなどの小規模モデルにまで柔軟に対応
- Linuxクライアント上のNFSサーバを使って、ファイルシステムをNFSクライアントへエクスポートすることが可能
- Linuxクライアント上にSambaサーバを構築して外部公開することにより、Windowsからのアクセスを可能とする(付録Bを参照)
- IOサーバとしてDDN社のアプライアンスSFA7990XEに対応
  ScaTeFS IBライブラリを使用することで、SFA7990XEの特徴を活かして高スループットを実現

第2章　　ネットワークの構築

2.1　はじめに

ScaTeFSを使用するためにはネットワーク環境の構築を行いますが、構築前には計算ノード、L3スイッチ、IOサーバなど各コンポーネントの配置、IPアドレッシング(IPアドレスの割り当てルール)やルーティング設定の検討が重要になります。たとえばL3スイッチの配置を減らせば管理面が楽になりますが、この場合、少ない数のスイッチで多数の相手に向かって通信を行うことになるため、性能面が落ち込むことになります。また、IPアドレッシングを検討せずにシステム全体へのIPアドレスを割り振ってしまうと管理が大変になり、どのシステムにどのIPアドレスを設定するかなどわからなくなってしまいます。このようなことから構築前の検討については入念に行う必要があります。各コンポーネントの構成や、IPアドレッシングについて検討して実際にScaTeFSを使用した際のネットワーク構成例を以下に記載します。

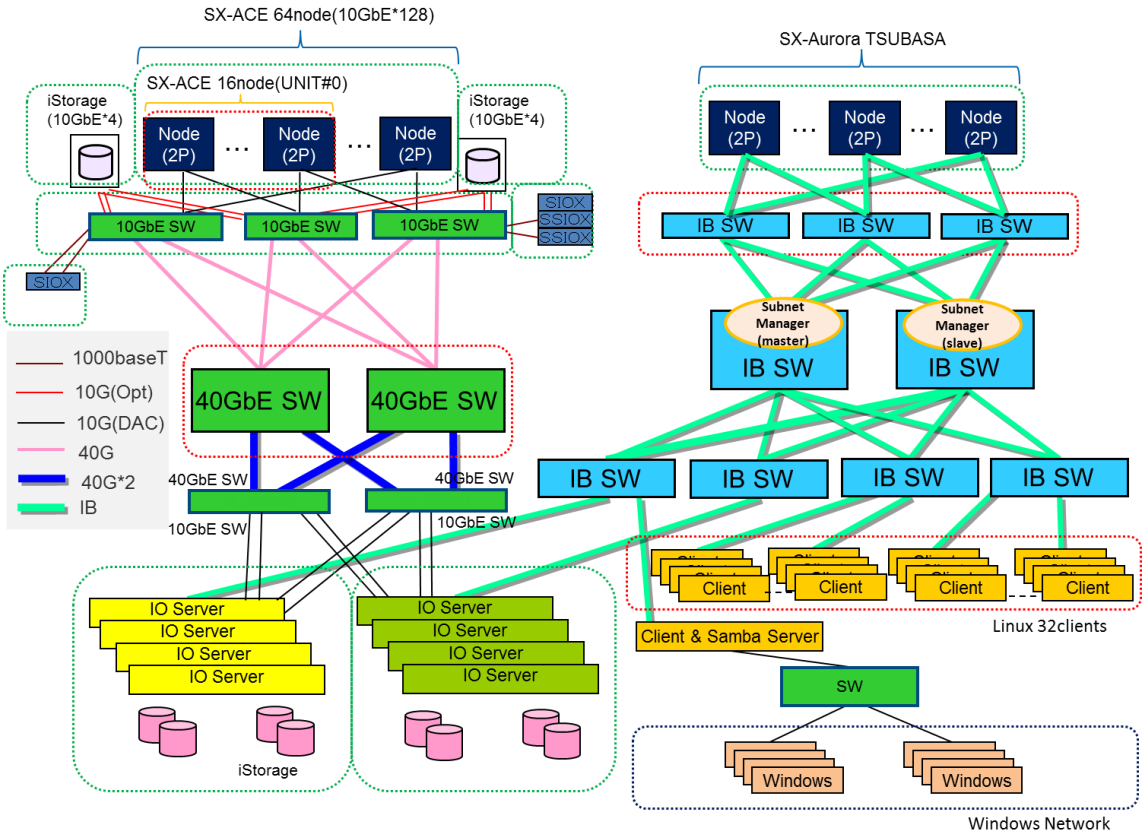


図 2-1　ネットワーク構成例

2.2　InfiniBandの利用

2.2.1　ネットワーク構成

IBのネットワークでは、ScaTeFSクライアントとIOサーバを同一のサブネットにします。クライアントとIOサーバには、IPoIBのIPv4アドレスを割り当てます。割り当てるIPoIBのIPv4アドレスはマシンあたり1つです。複数のHCAポートを利用する場合は、ib-bondingを適用できます。

2.2.2　通信方式

ScaTeFSはIB VerbsとIPoIBの二種類の通信方式を使用します。

- IB Verbs通信

  IB VerbsはIBのカーネルネイティブAPIを用いた高速な通信で、HCAのデバイス名とポート番号を使って通信を行います。ScaTeFSではファイルシステムのアクセス全般に利用します。ScaTeFSクライアントのマルチパス機能により、通信の冗長化を行うことができます。

- IPoIB通信

  IPoIBはIPv4アドレスとTCPポート番号を使った通信を行います。ScaTeFSにおいては、IB Verbsのコネクションを確立するために使用します(制御通信)。IPoIB通信はネットワークインターフェースにib-bondingを適用することで経路を冗長化できます。ib-bondingはLinuxのnmtuiコマンドやnmcliコマンドで設定します。

また、アプリケーション実行時に「ScaTeFS IBライブラリ」を使用するとライブラリ内でIB通信を行う方式となります。大IOを行うアプリケーションの性能向上が期待できます。ScaTeFS IBライブラリの詳細については「11.6 ScaTeFS InfiniBand 高速IOライブラリの使用方法」を参照してください。

2.2.3　マルチパス機能

クライアントでは、利用するHCAデバイスとポートを設定ファイルに記載します。設定ファイルに複数のHCAデバイスを記述することで、複数のHCAデバイスをActive/Active構成で利用できます。

HCAデバイスの故障や経路障害が発生した際は、残りの有効な経路を使って通信を継続します。通信できなくなった経路は監視状態となり、復旧を検知すると自動的に利用が再開されます。

マルチパス機能の設定方法や設定ファイルの詳細は「6.1.6マウント方法」および「6.1.7 クライアントのHCAデバイスの設定」を参照してください。

2.2.4　QoS (Quality of Service)

メタデータアクセスとIOのそれぞれの通信に対し、任意のサービスレベルを指定することが可能です。

サービスレベルはマウントオプションで指定します。マウントオプションの詳細については「6.1.6 マウント方法」を参照してください。

2.3　10GbEの利用

2.3.1　ネットワーク構成

表 2-1　アドレスを割り当てる際の設定値の例では、IPアドレス、VLAN-IDは以下の割り当て規則に従って設定しています。

- IPアドレス

  プライベートアドレス(class B) 172.16.0.0/25 ～ 172.31.255.255/25を使用してシステム全体のアドレスを割り当てています。IPアドレスの割り当て規則は下記のとおりです。

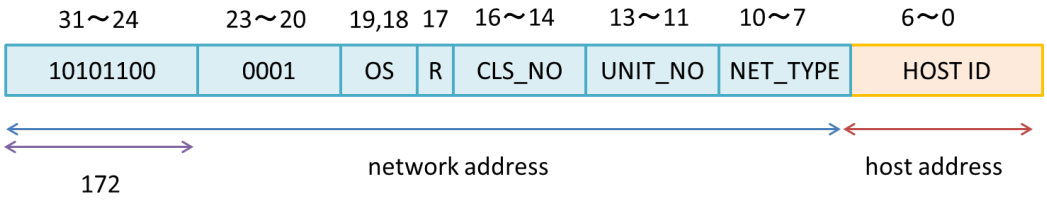


図 2-2　IP アドレッシング例

表 2-1　アドレスを割り当てる際の設定値

設定値	説明
OS	以下に従って2bitの値を設定します。 00：IPアドレスを設定するマシンがIOサーバ 01：IPアドレスを設定するマシンがSUPER-UX 10：IPアドレスを設定するマシンがSVP 11：上記に該当しないマシン
R	予約領域(MBZ)です。0を設定します。
CLS_NO	クラスタ番号に従って3bitの値を設定します。 例) クラスタ番号 2 → 010 クラスタは64nodeで構成されています。
UNIT_NO	UNIT番号に従って3bitの値を設定します。 例) UNIT番号 2 → 010 UNITは16nodeで構成されています。
NET_TYPE	NET_TYPEの4bitの内分けは下記のとおりです。 10 09 08 07 +--+--+--+--+ |type |IOC/P| +--+--+--+--+ Network Type/ IOC/ Port-No (4bit) type：Ethernetに従って値を設定します。 00：制御Ethernet(SX-ACE)またはIB(Linux-Client) 01：運用Ethernet 10：TOE 11：iSCSI IOC/P：IOCとポートに従って値を設定します。 00：IOC=0,port=0 01：IOC=0,port=1 10：IOC=1,port=0 11：IOC=1,port=1
HOST_ID	以下に従って7bitの値を設定します。 0000000：予約済 0000001～1000000：Node(0)～Node(63)に割り当てます。 1000001～1111101：iSCSI target(iStorage)に割り当てます。 1111110：ゲートウェイに割り当てます。 1111111：予約済

- VLAN-ID

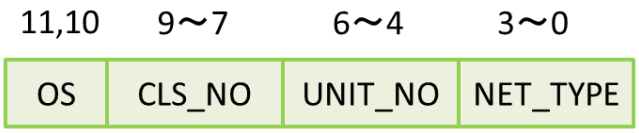


図 2-3　VLAN-ID 割り当て

各設定値については表 2-1を参照してください。

構成例のような大規模ネットワーク環境の場合、IPアドレス、VLANID-IDの割り当ては規則を定めて設定するのを推奨します。

構築方法の詳細については第5章、第6章、6.4.5を参照してください。

2.3.2　ルーティングテーブル

(1) クライアント側

クライアントがデフォルトで使用するネットワークインターフェースは、ルーティングテーブルに従って選択されるため、ルーティングテーブルを適切に設定しておく必要があります。例として、クライアントとIOサーバのネットワークインターフェースが以下であったとします。

- クライアント

  eth0:xx.xx.195.10

  eth1:xx.xx.196.10

- IOサーバ

  bond0:xx.xx.200.1

  bond1:xx.xx.201.1

※bonding(bond0,bond1)については3.2を参照してください。

※RHEL 7におけるイーサネットのインターフェース名は、デフォルトでは enXXXXX という名前です。この場合は、本マニュアル上のインターフェース名を実際の環境に合わせて読み替えてください。

eth0:xx.xx.195.10とbond0:xx.xx.200.1でコネクションを張り、eth1:xx.xx.196.10とbond1:xx.xx.201.1でコネクションを張る場合、ルーティングテーブルは以下のイメージになります。

- ipコマンドでルーティングテーブルを表示

```
# ip route
xx.xx.200.0/25  via  yy.yy.yy.yy  dev  eth0  proto  static  metric  NNN
xx.xx.201.0/25  via  zz.zz.zz.zz  dev  eth1  proto  static  metric  NNN
```

- netstatコマンドでルーティングテーブルを表示

```
# netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
xx.xx.200.0     yy.yy.yy.yy     255.255.255.128   UG     0     0       0   eth0
xx.xx.201.0     zz.zz.zz.zz     255.255.255.128   UG     0     0       0   eth1
```

(2) IO サーバ側

Linuxには外部からサブネットを跨いで接続する際、往路と復路が異なる場合接続できない制約があるため、複数のネットワークインターフェースで接続する場合はiproute2を利用する必要があります。

iproute2はルーティングをコントロールするパッケージで、往路と復路を一致させる機能があります。IOサーバのルーティングはiproute2を使用して設定してください。ルーティング設定例は5.1.18.3を参照してください。

2.3.3　DCB

Data Center Bridging (DCB)は各トラフィック種別(メタデータ、データ)に対して優先度付けを行うことができます。

ScaTeFSではDCBを利用してトラフィック種別にPriority(優先順位) を設定し、トラフィック種別毎に宛先ポートを使用することでメタデータ転送の遅延ができるだけ小さくなるよう実装しています。

メタデータ用宛先ポート番号、データ用宛先ポート番号は、/etc/scatefs/system.infoのcport(メタデータ用宛先ポート番号)、cdport(データ用宛先ポート番号)に設定します。cdportを設定しない場合、どちらもcportのポート番号を使用してコネクションを張ります。

2.3.4　Priority の設定

Priorityは各ポートに0(low)～6(high)まで設定可能で、ScaTeFSで使用するポートは以下のようにPriorityが設定されています。

表 2-2　ポートと Priority

ポート	Priority
メタデータ用ポート	6
データ用ポート	4
IOサーバ間用ポート	5

Priorityの設定は5.1.18.4を参照してください。

第3章　IO サーバの HW 構成

3.1　IOサーバ構成

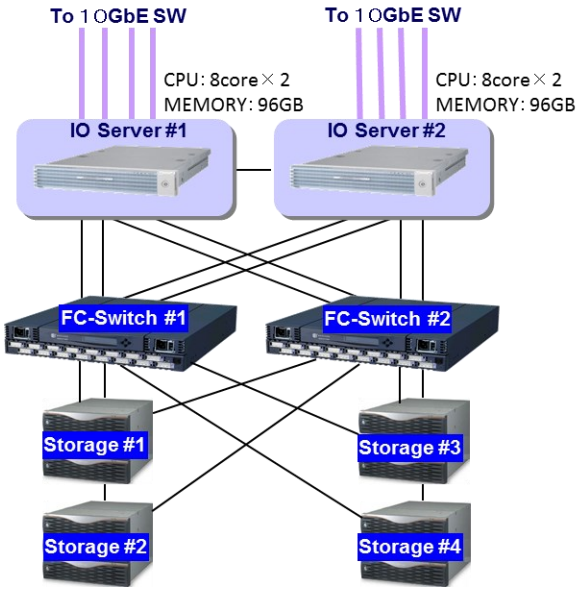
IOサーバは、2台のHAクラスタのActive-Active構成をとります。以下にモデル毎にIOサーバ構成を記載します。

3.1.1　標準モデル/小規模モデル向け IO サーバ v1

標準モデル向けIOサーバv1は、4台のストレージとは2port 8G-FC HBA x 2枚で接続します。クライアントとは2port 10GbE HBA x 2枚で接続します。

小規模モデル向けIOサーバv1は、2台のストレージとは2port 8G-FC HBA x 2枚でFC Switchを介さず直接接続します。クライアントとは2port 10GbE HBA x 1枚で接続します。

標準モデル向け IO サーバ v1



小規模モデル向け IO サーバ v1

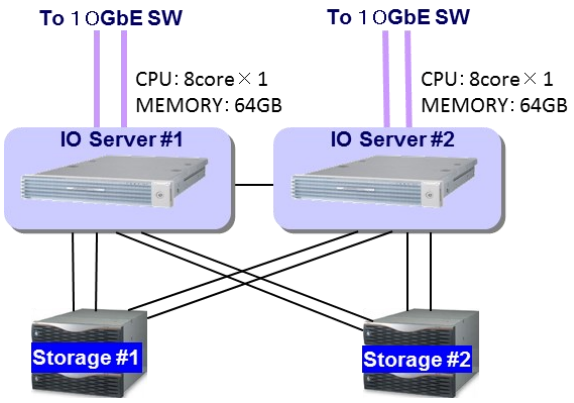


図 3-1　IOSv1 の構成例

3.1.2　標準モデル向け IO サーバ v3

標準モデル向けIOサーバv3は、2台のストレージとは2port 16G-FC HBA x 2枚でFC Switchを介さず直接接続します。クライアントとは2port 10GbE HBA x 2枚で接続します。オプションでIB HCAを装着することでIB HCAを装着しているLinuxのクライアントとIBネットワークで接続することができます。

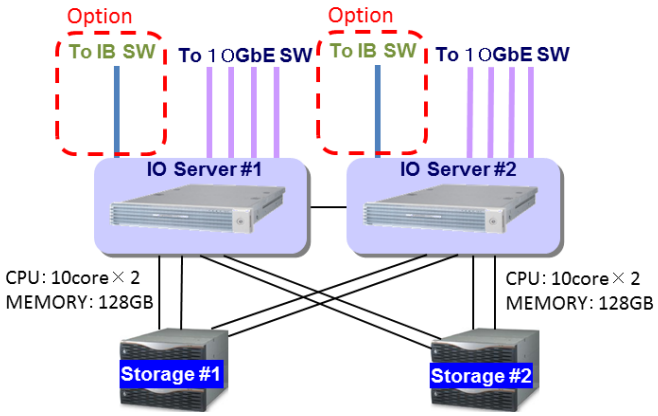


図 3-2　IOSv3 の構成例

3.1.3　標準モデル向け IO サーバ v4

標準モデル向けIOサーバv4は、2台のストレージとは2port 16G-FC/2port 32G-FC HBA x 2枚で直接接続します。クライアントとはIB HCA x 1枚、または2枚で接続します。IB HCAはEDR 1port、 2portいずれかを選択できます。オプションで10GbE HBAを装着することで、10GbE HBAを装着しているクライアントマシンや、SX-ACEなどと接続することができます。

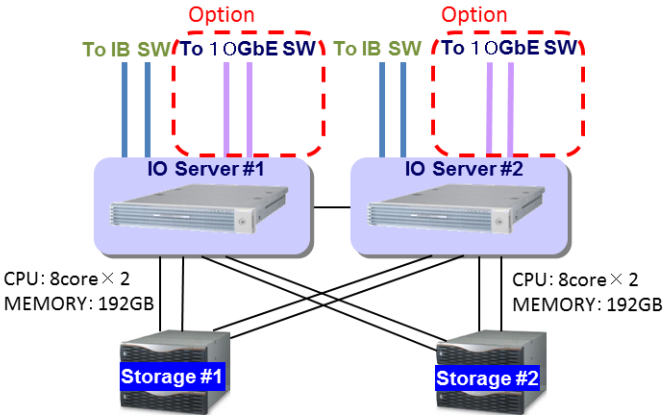


図 3-3　IOSv4の構成例

3.1.4　標準モデル向け IO サーバ v4+

標準モデル向けIOサーバv4+は、2台のストレージとは2port 16G/2port 32G-FC HBA x 2枚、またはSAS HBA x 2枚で直接接続します。クライアントとはIB HCA x 1枚、または2枚で接続します。IB HCAはEDR/HDR100 1port、 2portいずれかを選択できます。オプションで

10GbE HBAを装着することで、10GbE HBAを装着しているクライアントマシンや、SX-ACEなどと接続することができます。

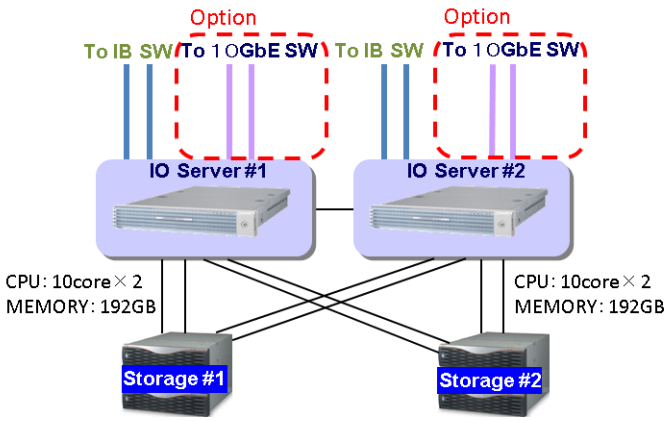


図 3-4　IOSv4+の構成例

3.1.5　標準モデル向け IO サーバ v4++

標準モデル向けIOサーバv4++は、2台のストレージとは2port 16G/2port 32G-FC HBA x 2枚、またはSAS HBA x 2枚で直接接続します。クライアントとはIB HCA x 1枚、または2枚で接続します。IB HCAはHDR100 1port、 2portいずれかを選択できます。オプションで10GbE HBAを装着することで、10GbE HBAを装着しているクライアントマシンや、SX-ACEなどと接続することができます。

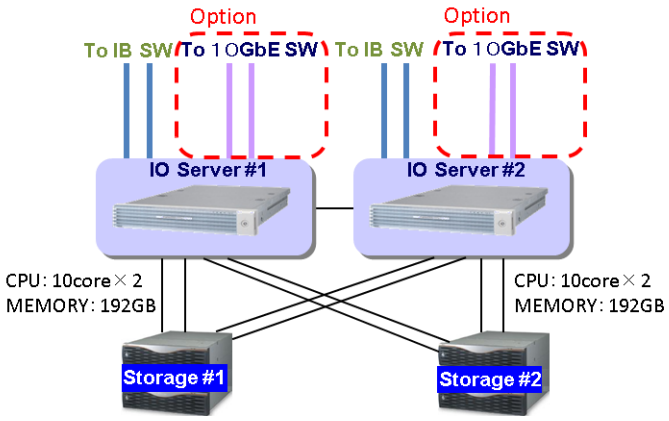


図 3-5　IOSv4++の構成例

3.1.6　DDN 社の SFA7990XE

SFA7990XEとSS9012(ディスクエンクロージャ)はSASインターフェースで直接接続されています。クライアントとはIB HCA 2枚で接続します。IB HCAはHDR100です。SFA7990XEの2つのコントローラ上のVMでIOサーバが動作します。

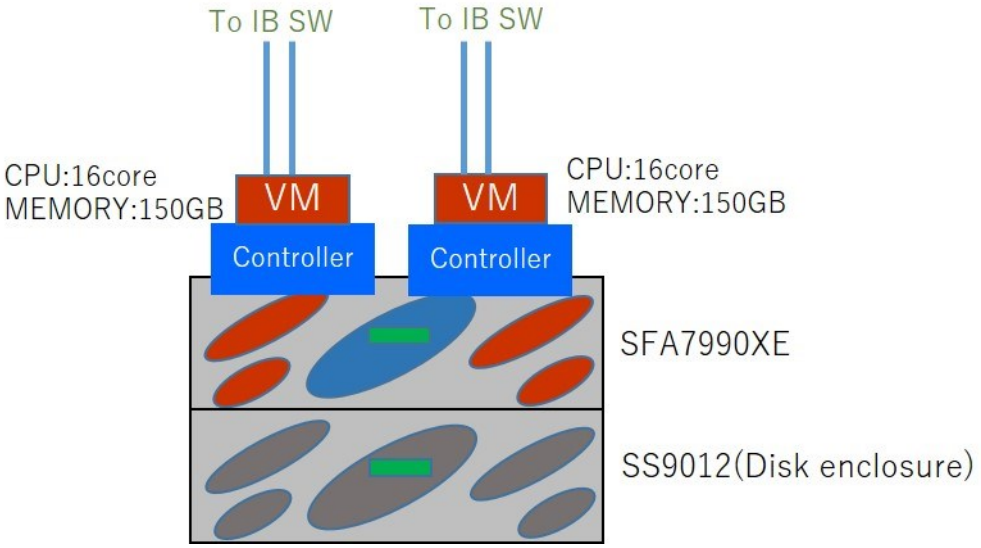


図 3-6　SFA7990XE+SS9012 の構成

3.2　10GbEのbonding

1台のマシンに複数のNICやイーサネットポートを搭載し、それらを仮想的な１つのネットワークインターフェースとして扱うことをbondingといいます。負荷分散や帯域向上、耐障害性の向上を図ることができるため、IOサーバにbondingの設定を行うことを推奨します。

bonding には複数のモードがありますが、LinuxとSX-ACEが混在する環境の場合は”802.3ad(LACP, 動的リンクアグリゲーション)”である必要があり、L3スイッチもまた”802.3ad”に対応している必要があります。また、bondingをする場合、対象のNICは同じL3スイッチ配下に接続してください。bonding設定例は5.1.18を参照してください。

第4章　　クライアント側の HW 構成

4.1　Linux（SX-Aurora TSUBASA）

クライアントとして使用できるLinuxマシンはx86-64アーキテクチャをサポートしている必要があります。他のx86等のアーキテクチャでは使用できません。

IBでIOサーバと接続する場合は、IB HCAを装備する必要があります。サポート対象のIB HCAは、NVIDIA社のConnectX-４およびConnectX-6です。

ConnectX-6はRHEL/CentOS 7.6以降でサポートしています。

4.2　SX-ACE

従来のSX-ACEで利用できるネットワークインターフェースについて説明します。

SX-ACEのネットワークインターフェースは10GbE-NICを最大2枚搭載しており、物理ポート1ポートにつき以下の4つの仮想インターフェースを利用しています。

- 制御系ネットワーク(GbE相当)
- 運用系ネットワーク(GbE相当)
- IOサーバ用ネットワーク(10GbE相当、TOEを利用しています)
- ローカルストレージ用ネットワーク(10GbE相当、iSCSI SRV/ローカルストレージ用)

VLANは仮想インターフェース毎に設定しています。
チャンネル番号の割り当てについては以下に図示しました。

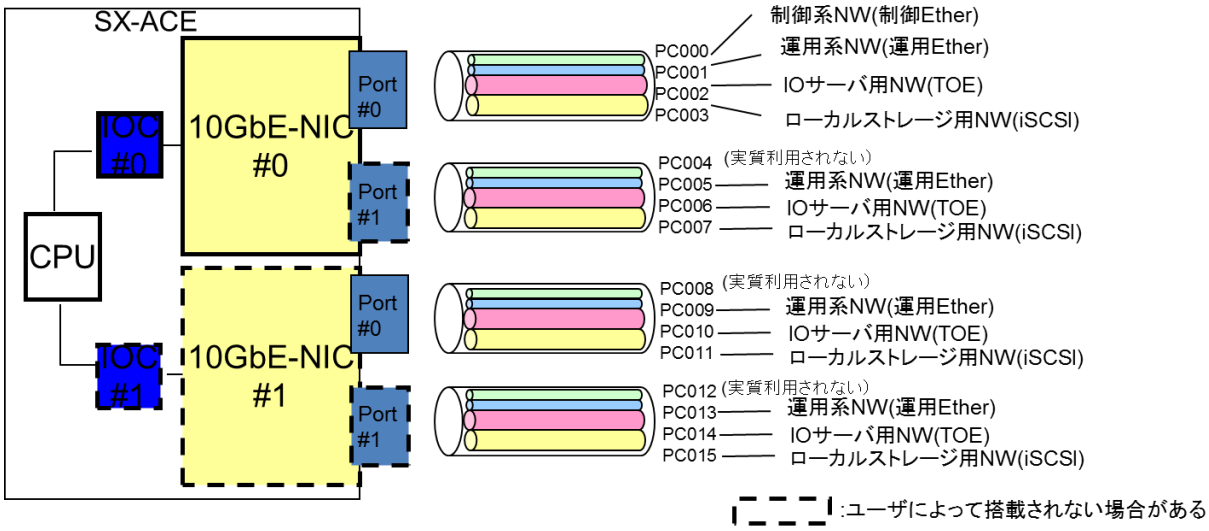


図 4-1　10GbE-NIC の構成

第5章　　IO サーバの構築

NEC 製 IO サーバ、ストレージを使用する場合は 5.1～5.4 を参照し構築してください。
DDN 製 SFA7990XE を使用する場合は 5.5 を参照し構築してください。

5.1　IOサーバの準備

IOサーバの各ノードには以下のプログラムプロダクトが必要です。

- NEC Storage Manager Agent Utility
- StoragePathSavior for Linux driver package
- CLUSTERPRO X for Linux
- NEC Scalable Technology File System/Server (ScaTeFSサーバ機能)

注：
IOサーバへOSをインストールするときは、「/ (ルート)」のデバイスタイプは 「標準パーティション」を選択してください。

以下に動作確認済みバージョンを記載します。

【標準モデル向け IO サーバ v4++】

表 5-1　標準モデル向け IO サーバ v4++ 動作確認済みバージョン

OS	kernel	MLNX_OFED	CLUSTERPRO	SPS	iSMccs
RHEL7.7	3.10.0-1062.el7.x86_64	4.7-1.0.0.1	4.3.4-1	7.3.1	-

【標準モデル向け IO サーバ v4+】

表 5-2　標準モデル向け IO サーバ v4+ 動作確認済みバージョン

OS	kernel	MLNX_OFED	CLUSTERPRO	SPS	iSMccs
RHEL7.6	3.10.0-957.el7.x86_64	4.6-4.1.2.0	4.1.1-1	7.2	10.3-005

【標準モデル向け IO サーバ v4】

表 5-3　標準モデル向け IO サーバ v4 動作確認済みバージョン

OS	kernel	MLNX_OFED	CLUSTERPRO	SPS	iSMagent
RHEL7.4	3.10.0-693.el7.x86_64	4.2-1.2.0.0	3.3.5-1	7.0 6.7	9.7-003

【標準モデル向け IO サーバ v3】

表 5-4　標準モデル向け IO サーバ v3 動作確認済みバージョン

OS	kernel	MLNX_OFED	CLUSTER PRO	SPS	iSMagent
RHEL7.4	3.10.0-693.el7.x86_64	4.2-1.2.0.0	3.3.5-1	6.7	9.7-003
RHEL7.3	3.10.0-514.26.2.el7.x86_64				

【標準モデル/小規模モデル向け IO サーバ v1】

表 5-5　標準モデル/小規模モデル向け IO サーバ v1 動作確認済みバージョン

OS	kernel	MLNX_OFED	CLUSTER PRO	SPS	iSMagent
RHEL6.4	2.6.32-358.23.2.el6.x86_64	-	3.2.0-1	6.2	8.4-002

IOサーバに以下の設定を行い、ScaTeFSに組み込む準備をします。設定には全IOサーバにrootログインして実施するものもあります。また、ディストリビューションごとに設定内容が異なるものもあります。お使いのディストリビューションに合わせて設定してください。

- IOターゲットの設計
- LVMの設計
- CLUSTERPROのクラスタ構成情報作成
- NEC Storage Manager Agent Utility(iSMagent)のインストール
- ホスト登録
- 論理ディスク割り当て
- StoragePathSavior for Linux driver package(SPS)のインストールと設定
- CLUSTERPRO X for Linuxのインストール
- DCB対応版10GbE-NICドライバのインストール
- IBドライバのインストール
- rsh関連のインストール
- ScaTeFSパッケージのインストール
- ScaTeFSのライセンス登録
- SELinux無効化

- ファイアウォール無効化
- prelink無効化
- abrtd無効化
- ネットワークの設定
- 時刻の設定(ntp)
- ファイルシステム管理アカウント(fsadmin)の設定
- 内蔵ディスク(SSD)の設定
- カーネルパラメータの設定
- syslogのログローテート設定
- updatedb.confの設定
- ScaTeFSのIOサーバとして組み込む(scatefs_addios)

本作業は、ストレージの設定(RAID作成、ゾーニングなど)の完了後に実施してください。以下の説明では 2台のIOサーバをiosv00,iosv01と記載します。

5.1.1　IO ターゲットの設計

IOターゲットは、ScaTeFSのファイルシステムの基盤となるデータストアです。クライアントノードから書き込まれたファイルデータは、IOサーバに分散され、さらにIOサーバ内のIOターゲットに分散して格納されます。

IOターゲットはファイルのデータ自体を格納するデータ領域と、ファイルタイプや更新時刻などを格納するメタデータ領域に分けられます。複数のIOターゲットを作成できますが、必ずデータ領域とメタデータ領域の個数は同数で一対一の関係です。

以下はIOサーバ2台のIOターゲットの構成例です。

表 5-6　IO ターゲットの構成例　標準モデル/小規模モデル向け IO サーバ v1

モデル	データ領域					メタデータ領域					IO ターゲット数
	HDD		プール			HDD		プール			
	容量	数	RAID	数	LD	容量	数	RAID	数	LD	
標準	1TB	36	6 (4+PQ)	6	6	600GB	6	10	1	1	6
	2TB				12						12
	3TB				18						18
	4TB				24						24
小規模	1TB	36	6 (4+PQ)	6	6	600GB	6	10	1	1	6
	2TB				6						6
	3TB				12						12
	4TB				12						12

※HDD、プールはストレージ 1 台あたりの値です。

表 5-7　IO ターゲットの構成例　標準モデル向け IO サーバ v3 と v4

モデル	データ領域					メタデータ領域					IO ターゲット数
	HDD		プール			HDD		プール			
	容量	数	RAID	数	LD	容量	数	RAID	数	LD	
標準	1TB	72	6 (4+PQ)	12	12	600GB	12	10	2	2	6
	2TB				24						12
	4TB				48						24
	6TB				72						36
	10TB		6 (8+PQ)	7	14						28
	12TB	80		8	16						32

※HDD、プールはストレージ 1 台あたりの値です。

標準モデルIOサーバv4+では、IOターゲットのファイルシステムにxfsを選択できるようにしました。ディスクタイプがNLSASの場合、xfsを推奨します。ディスクタイプがSAS、SSDの場合、ext4を推奨します。

表 5-8　IO ターゲットの構成例　標準モデル向け IO サーバ v4+以降　データ領域

データ領域							
ディスク			プール			推奨ファイルシステムタイプ	IO ターゲット数
タイプ	容量	数	RAID	数	LD		
NLSAS	4TB	72	6(4+PQ)	12	12	xfs	12
	8TB	80	6(8+PQ)	8	8		8
	12TB	80	6(8+PQ)	8	8		8
SAS	1.2TB	72	6(4+PQ)	12	12	ext4	12
SSD	1.6TB	24	6(4+PQ)	4	4		4

※ディスク、プールはストレージ 1 台あたりの値です。

表 5-9　IO ターゲットの構成例　標準モデル向け IO サーバ v4+以降　メタデータ領域

メタデータ領域							
ディスク			プール			推奨ファイルシステムタイプ	IO ターゲット数
タイプ	容量	数	RAID	数	LD		
SAS	600GB	12	10	2	2	ext4	データ領域の IOターゲット数と同じ
SSD	400GB	6	10	1	1		

※ディスク、プールはストレージ 1 台あたりの値です。

IO ターゲット ID は、scatefs_addiot コマンド(5.2.5 参照)実行時に割り当てられます。以下に IO サーバが 4 台、IO ターゲットが 12 個の場合の IO ターゲット ID の割り当て例を記載します。

表 5-10　IO ターゲット ID の割り当て例

IOサーバ	IOS#0	IOS#1	IOS#2	IOS#3
IOターゲットID	0	3	6	9
	1	4	7	10
	2	5	8	11

以降の説明では、上記の例を使用します。

5.1.2　LVM の設計

IOサーバに接続しているiStorageの構成(プール、LD(論理ディスク))からScaTeFSのメタデータ領域、データ領域のパーティション数、ストライピング数、IOターゲットを使用する順序を設計します。

以下にモデル毎にLVMの設計例を記載します。

【標準モデル向けIOサーバv1】

以下にデータ領域のHDD(1TB)の場合の設計例を記載します。

- ScaTeFSのデータ領域

  4wayストライピングのLV作成の際は、負荷分散(※)のために以下のようにします。

  (※)SPSパスのチェックについて、5.2.2で確認します。

  POOL1とPOOL2のLDで4wayストライピングのLVを作成します。

  Storage1とStorage3、Storage2とStorage4のLDは同じ番号のLDで組み合わせます。

  POOL3とPOOL4、POOL5とPOOL6も同様です。

プール構成

プール	Storage1	Storage2	Storage3	Storage4
POOL1	LD1	LD1	LD1	LD1
POOL2	LD2	LD2	LD2	LD2
POOL3	LD3	LD3	LD3	LD3
POOL4	LD4	LD4	LD4	LD4
POOL5	LD5	LD5	LD5	LD5
POOL6	LD6	LD6	LD6	LD6

LVM構成

Storage				iosv00			iosv01		
1	2	3	4	LV	IOT	順序	LV	IOT	順序
LD1	LD2	LD1	LD2	lv_data01	0	1	-	-	-
LD2	LD1	LD2	LD1	lv_data02	1	2	-	-	-
LD3	LD4	LD3	LD4	lv_data03	2	3	-	-	-
LD4	LD3	LD4	LD3	-	-	-	lv_data04	3	1
LD5	LD6	LD5	LD6	-	-	-	lv_data05	4	2
LD6	LD5	LD6	LD5	-	-	-	lv_data06	5	3

上記 LVM 構成の IO ターゲットを使用する順序を記載します。

IOサーバ	IOターゲットを使用する順序
iosv00	0 1 2
iosv01	3 4 5

5.3.1 ScaTeFS 作成の設定項目 iotid の値を記載します。

iosv00、iosv01 の IO ターゲットを使用する順序を続けて記載します。

項目	設定値
iotid	0 1 2 3 4 5

- ScaTeFSのメタデータ領域

  2wayストライピングのLV作成の際は、Storage1とStorage2、Storage3とStorage4のLDを組み合わせます。

  プール構成

プール	Storage1	Storage2	Storage3	Storage4
POOL0	LD0-2,3,4	LD0-2,3,4	LD0-2,3,4	LD0-2,3,4

  ※LD0-XのXはパーティションを指します。

  LVM構成

Storage				iosv00		iosv01	
1	2	3	4	LV	IOT	LV	IOT
LD0-2	LD0-2	-	-	lv_ctrl01	0	-	-
LD0-3	LD0-3	-	-	lv_ctrl02	1	-	-
LD0-4	LD0-4	-	-	lv_ctrl03	2	-	-
-	-	LD0-2	LD0-2	-	-	lv_ctrl04	3
-	-	LD0-3	LD0-3	-	-	lv_ctrl05	4
-	-	LD0-4	LD0-4	-	-	lv_ctrl06	5

【標準モデル向けIOサーバv3】

以下にデータ領域のHDD(4TB)の場合の設計例を記載します。

- ScaTeFSのデータ領域

  4wayストライピングのLV作成の際は、Storage1とStorage2のPOOL2、POOL3の同じ番号のLDで組み合わせます。POOL4以降も同様です。

プール構成

プール	Storage1	Storage2
POOL2	LD2,LD3,LD4,LD5	LD2,LD3,LD4,LD5
POOL3	LD6,LD7,LD8,LD9	LD6,LD7,LD8,LD9
POOL4	LDA,LDB,LDC,LDD	LDA,LDB,LDC,LDD
POOL5	LDE,LDF,LD10,LD11	LDE,LDF,LD10,LD11
POOL6	LD12,LD13,LD14,LD15	LD12,LD13,LD14,LD15
POOL7	LD16,LD17,LD18,LD19	LD16,LD17,LD18,LD19
POOL8	LD1A,LD1B,LD1C,LD1D	LD1A,LD1B,LD1C,LD1D
POOL9	LD1E,LD1F,LD20,LD21	LD1E,LD1F,LD20,LD21
POOL10	LD22,LD23,LD24,LD25	LD22,LD23,LD24,LD25
POOL11	LD26,LD27,LD28,LD29	LD26,LD27,LD28,LD29
POOL12	LD2A,LD2B,LD2C,LD2D	LD2A,LD2B,LD2C,LD2D
POOL13	LD2E,LD2F,LD30,LD31	LD2E,LD2F,LD30,LD31

LVM 構成

Storage1	Storage2	iosv00			iosv01		
		LV	IOT	順序	LV	IOT	順序
LD2,LD6	LD2,LD6	lv_data01	0	1	-		
LD3,LD7	LD3,LD7	lv_data02	1	2	-		
LD4,LD8	LD4,LD8	lv_data03	2	3	-		
LD5,LD9	LD5,LD9	lv_data04	3	4	-		
LDA,LDE	LDA,LDE	lv_data05	4	5	-		
LDB,LDF	LDB,LDF	lv_data06	5	6	-		
LDC,LD10	LDC,LD10	lv_data07	6	7	-		
LDD,LD11	LDD,LD11	lv_data08	7	8	-		
LD12,LD16	LD12,LD16	lv_data09	8	9	-		
LD13,LD17	LD13,LD17	lv_data10	9	10	-		
LD14,LD18	LD14,LD18	lv_data11	10	11	-		
LD15,LD19	LD15,LD19	lv_data12	11	12	-		
LD1A,LD1E	LD1A,LD1E	-			lv_data13	12	1
LD1B,LD1F	LD1B,LD1F	-			lv_data14	13	2

Storage1	Storage2	iosv00			iosv01		
		LV	IOT	順序	LV	IOT	順序
LD1C,LD20	LD1C,LD20	-			lv_data15	14	3
LD1D,LD21	LD1D,LD21	-			lv_data16	15	4
LD22,LD26	LD22,LD26	-			lv_data17	16	5
LD23,LD27	LD23,LD27	-			lv_data18	17	6
LD24,LD28	LD24,LD28	-			lv_data19	18	7
LD25,LD29	LD25,LD29	-			lv_data20	19	8
LD2A,LD2E	LD2A,LD2E	-			lv_data21	20	9
LD2B,LD2F	LD2B,LD2F	-			lv_data22	21	10
LD2C,LD30	LD2C,LD30	-			lv_data23	22	11
LD2D,LD31	LD2D,LD31	-			lv_data24	23	12

上記 LVM 構成の IO ターゲットを使用する順序を記載します。

IOサーバ	IOターゲットを使用する順序
iosv00	0 1 2 3 4 5 6 7 8 9 10 11
iosv01	12 13 14 15 16 17 18 19 20 21 22 23

5.3.1 ScaTeFS 作成の設定項目 iotid の値を記載します。

iosv00、iosv01 の IO ターゲットを使用する順序を続けて記載します。

項目	設定値
iotid	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

- ScaTeFSのメタデータ領域

  ストライピングなしでLVを作成します。

  プール構成

プール	Storage1	Storage2
POOL0	LD0-2,3,4,5,6,7	LD0-2,3,4,5,6,7
POOL1	LD1-2,3,4,5,6,7	LD1-2,3,4,5,6,7

  ※LD0-XのXはパーティションを指します。

  LVM構成

Storage1	Storage2	iosv00		iosv01	
		LV	IOT	LV	IOT
LD0-2		lv_ctrl01	0	-	-
LD0-3		lv_ctrl02	1	-	-
LD0-4		lv_ctrl03	2	-	-
LD0-5		lv_ctrl04	3	-	-
LD0-6		lv_ctrl05	4	-	-
LD0-7		lv_ctrl06	5	-	-
LD1-2		lv_ctrl07	6	-	-
LD1-3		lv_ctrl08	7	-	-
LD1-4		lv_ctrl09	8	-	-
LD1-5		lv_ctrl10	9	-	-
LD1-6		lv_ctrl11	10	-	-
LD1-7		lv_ctrl12	11	-	-
	LD0-2	-	-	lv_ctrl13	12
	LD0-3	-	-	lv_ctrl14	13
	LD0-4	-	-	lv_ctrl15	14
	LD0-5	-	-	lv_ctrl16	15
	LD0-6	-	-	lv_ctrl17	16
	LD0-7	-	-	lv_ctrl18	17
	LD1-2	-	-	lv_ctrl19	18
	LD1-3	-	-	lv_ctrl20	19
	LD1-4	-	-	lv_ctrl21	20
	LD1-5	-	-	lv_ctrl22	21
	LD1-6	-	-	lv_ctrl23	22
	LD1-7	-	-	lv_ctrl24	23

【データ領域の LVM ストライピングなしの設計】

データ領域の LV を LVM ストライピングなしとする場合、IO サーバ、ストレージの負荷を可能な限り分散するように IO ターゲットを使用する順序を設計してください。この順序は 5.3.1 ScaTeFS 作成の設定項目 iotid へ指定します。可能な限り以下の条件を満たす順序を設計してください。

a) IO サーバから見てストレージを交互に使用します。

b) IO サーバから見てストレージ内の奇数および偶数のプール番号を交互に使用します。

c)　1 つのプールに複数の LD が存在する場合、プールの先頭の LD を使用してから次の LD を使用します。

以下にストレージ(プール：4、LD：8)の LVM の設計例を記載します。

プール	Storage 1	Storage 2	iosv00			iosv01		
			LV	IOT	順序	LV	IOT	順序
POOL2	LD2	-	lv_data01	0	1	-	-	-
	LD3	-	lv_data02	1	5	-	-	-
POOL3	LD4	-	lv_data03	2	3	-	-	-
	LD5	-	lv_data04	3	7	-	-	-
POOL4	-	LD6	lv_data05	4	4	-	-	-
	-	LD7	lv_data06	5	8	-	-	-
POOL5	-	LD8	lv_data07	6	2	-	-	-
	-	LD9	lv_data08	7	6	-	-	-
POOL2	-	LD2	-	-		lv_data09	8	1
	-	LD3	-	-		lv_data10	9	5
POOL3	-	LD4	-	-		lv_data11	10	3
	-	LD5	-	-		lv_data12	11	7
POOL4	LD6	-	-	-		lv_data13	12	4
	LD7	-	-	-		lv_data14	13	8
POOL5	LD8	-	-	-		lv_data15	14	2
	LD9	-	-	-		lv_data16	15	6

上記設計例の IO ターゲットを使用する順序を記載します。

IOサーバ	IOターゲットを使用する順序
iosv00	0 6 2 4 1 7 3 5
iosv01	8 14 10 12 9 15 11 13

5.3.1 ScaTeFS 作成の設定項目 iotid の値を記載します。

iosv00、iosv01 の IO ターゲットを使用する順序を続けて記載します。

項目	設定値
iotid	0 6 2 4 1 7 3 5 8 14 10 12 9 15 11 13

以下にデータ領域の HDD(10TB)の場合の設計例を記載します。

- ScaTeFSのデータ領域

ストライピングなしでLVを作成します。

LVM構成

プール	Storage 1	Storage 2	iosv00			iosv01		
			LV	IOT	順序	LV	IOT	順序
POOL2	LD2	-	lv_data01	0	1	-	-	-
	LD3	-	lv_data02	1	8	-	-	-
POOL3	LD4	-	lv_data03	2	3	-	-	-
	LD5	-	lv_data04	3	10	-	-	-
POOL4	LD6	-	lv_data05	4	5	-	-	-
	LD7	-	lv_data06	5	12	-	-	-
POOL5	-	LD8	lv_data07	6	2	-	-	-
	-	LD9	lv_data08	7	9	-	-	-
POOL6	-	LDA	lv_data09	8	4	-	-	-
	-	LDB	lv_data10	9	11	-	-	-
POOL7	-	LDC	lv_data11	10	6	-	-	-
	-	LDD	lv_data12	11	13	-	-	-
POOL8	-	LDE	lv_data13	12	7	-	-	-
	-	LDF	lv_data14	13	14	-	-	-
POOL2	-	LD2	-	-	-	lv_data15	14	1
	-	LD3	-	-	-	lv_data16	15	8
POOL3	-	LD4	-	-	-	lv_data17	16	3
	-	LD5	-	-	-	lv_data18	17	10
POOL4	-	LD6	-	-	-	lv_data19	18	5
	-	LD7	-	-	-	lv_data20	19	12
POOL5	LD8	-	-	-	-	lv_data21	20	2
	LD9	-	-	-	-	lv_data22	21	9
POOL6	LDA	-	-	-	-	lv_data23	22	4
	LDB	-	-	-	-	lv_data24	23	11
POOL7	LDC	-	-	-	-	lv_data25	24	6
	LDD	-	-	-	-	lv_data26	25	13
POOL8	LDE	-	-	-	-	lv_data27	26	7
	LDF	-	-	-	-	lv_data28	27	14

上記 LVM 構成の IO ターゲットを使用する順序を記載します。

IOサーバ	IOターゲットを使用する順序
iosv00	0 6 2 8 4 10 12 1 7 3 9 5 11 13
iosv01	14 20 16 22 18 24 26 15 21 17 23 19 25 27

5.3.1 ScaTeFS 作成の設定項目 iotid の値を記載します。

iosv00、iosv01 の IO ターゲットを使用する順序を続けて記載します。

項目	設定値
iotid	0 6 2 8 4 10 12 1 7 3 9 5 11 13 14 20 16 22 18 24 26 15 21 17 23 19 25 27

- ScaTeFSのメタデータ領域

  ストライピングなしでLVを作成します。

  LVM構成

プール	Storage 1	Storage 2	iosv00		iosv01	
			LV	IOT	LV	IOT
POOL0	LD0-2	-	lv_ctrl01	0	-	-
	LD0-3	-	lv_ctrl02	1	-	-
	LD0-4	-	lv_ctrl03	2	-	-
	LD0-5	-	lv_ctrl04	3	-	-
	LD0-6	-	lv_ctrl05	4	-	-
	LD0-7	-	lv_ctrl06	5	-	-
	LD0-8	-	lv_ctrl07	6	-	-
POOL1	LD1-2	-	lv_ctrl08	7	-	-
	LD1-3	-	lv_ctrl09	8	-	-
	LD1-4	-	lv_ctrl10	9	-	-
	LD1-5	-	lv_ctrl11	10	-	-
	LD1-6	-	lv_ctrl12	11	-	-
	LD1-7	-	lv_ctrl13	12	-	-
	LD1-8	-	lv_ctrl14	13	-	-
POOL0	-	LD0-2	-	-	lv_data15	14
	-	LD0-3	-	-	lv_data16	15
	-	LD0-4	-	-	lv_data17	16

プール	Storage 1	Storage 2	iosv00		iosv01	
			LV	IOT	LV	IOT
	-	LD0-5	-	-	lv_data18	17
	-	LD0-6	-	-	lv_data19	18
	-	LD0-7	-	-	lv_data20	19
	-	LD0-8	-	-	lv_data21	20
POOL1	-	LD1-2	-	-	lv_data22	21
	-	LD1-3	-	-	lv_data23	22
	-	LD1-4	-	-	lv_data24	23
	-	LD1-5	-	-	lv_data25	24
	-	LD1-6	-	-	lv_data26	25
	-	LD1-7	-	-	lv_data27	26
	-	LD1-8	-	-	lv_data28	27

※LD0-XのXはパーティションを指します。

【標準モデル向けIOサーバv4】

以下にデータ領域のHDD(12TB)の場合の設計例を記載します。

- ScaTeFSのデータ領域

  ストライピングなしでLVを作成します。

LVM構成

プール	Storage 1	Storage 2	iosv00			iosv01		
			LV	IOT	順序	LV	IOT	順序
POOL2	LD2	-	lv_data01	0	1	-	-	-
	LD3	-	lv_data02	1	9	-	-	-
POOL3	LD4	-	lv_data03	2	3	-	-	-
	LD5	-	lv_data04	3	11	-	-	-
POOL4	LD6	-	lv_data05	4	5	-	-	-
	LD7	-	lv_data06	5	13	-	-	-
POOL5	LD8	-	lv_data07	6	7	-	-	-
	LD9	-	lv_data08	7	15	-	-	-
POOL6	-	LDA	lv_data09	8	8	-	-	-

プール	Storage 1	Storage 2	iosv00			iosv01		
			LV	IOT	順序	LV	IOT	順序
	-	LDB	lv_data10	9	16	-	-	-
POOL7	-	LDC	lv_data11	10	6	-	-	-
	-	LDD	lv_data12	11	14	-	-	-
POOL8	-	LDE	lv_data13	12	4	-	-	-
	-	LDF	lv_data14	13	12	-	-	-
POOL9	-	LD10	lv_data15	14	2	-	-	-
	-	LD11	lv_data16	15	10	-	-	-
POOL2	-	LD2	-	-	-	lv_data17	16	1
	-	LD3	-	-	-	lv_data18	17	9
POOL3	-	LD4	-	-	-	lv_data19	18	3
	-	LD5	-	-	-	lv_data20	19	11
POOL4	-	LD6	-	-	-	lv_data21	20	5
	-	LD7	-	-	-	lv_data22	21	13
POOL5	-	LD8	-	-	-	lv_data23	22	7
	-	LD9	-	-	-	lv_data24	23	15
POOL6	LDA	-	-	-	-	lv_data25	24	8
	LDB	-	-	-	-	lv_data26	25	16
POOL7	LDC	-	-	-	-	lv_data27	26	6
	LDD	-	-	-	-	lv_data28	27	14
POOL8	LDE	-	-	-	-	lv_data29	28	4
	LDF	-	-	-	-	lv_data30	29	12
POOL9	LD10	-	-	-	-	lv_data31	30	2
	LD11	-	-	-	-	lv_data32	31	10

上記 LVM 構成の IO ターゲットを使用する順序を記載します。

IOサーバ	IOターゲットを使用する順序
iosv00	0 14 2 12 4 10 6 8 1 15 3 13 5 11 7 9
iosv01	16 30 18 28 20 26 22 24 17 31 19 29 21 27 23 25

5.3.1 ScaTeFS 作成の設定項目 iotid の値を記載します。

iosv00、iosv01 の IO ターゲットを使用する順序を続けて記載します。

項目	設定値
iotid	0 14 2 12 4 10 6 8 1 15 3 13 5 11 7 9 16 30 18 28 20 26 22 24 17 31 19 29 21 27 23 25

- ScaTeFSのメタデータ領域

  ストライピングなしでLVを作成します。

LVM構成

プール	Storage 1	Storage 2	iosv00		iosv01	
			LV	IOT	LV	IOT
POOL0	LD0-2	-	lv_ctrl01	0	-	-
	LD0-3	-	lv_ctrl02	1	-	-
	LD0-4	-	lv_ctrl03	2	-	-
	LD0-5	-	lv_ctrl04	3	-	-
	LD0-6	-	lv_ctrl05	4	-	-
	LD0-7	-	lv_ctrl06	5	-	-
	LD0-8	-	lv_ctrl07	6	-	-
	LD0-9	-	lv_ctrl08	7	-	-
POOL1	LD1-2	-	lv_ctrl09	8	-	-
	LD1-3	-	lv_ctrl10	9	-	-
	LD1-4	-	lv_ctrl11	10	-	-
	LD1-5	-	lv_ctrl12	11	-	-
	LD1-6	-	lv_ctrl13	12	-	-
	LD1-7	-	lv_ctrl14	13	-	-
	LD1-8	-	lv_ctrl15	14	-	-
	LD1-9	-	lv_ctrl16	15	-	-
POOL0	-	LD0-2	-	-	lv_ctrl17	16
	-	LD0-3	-	-	lv_ctrl18	17
	-	LD0-4	-	-	lv_ctrl19	18
	-	LD0-5	-	-	lv_ctrl20	19
	-	LD0-6	-	-	lv_ctrl21	20
	-	LD0-7	-	-	lv_ctrl22	21
	-	LD0-8	-	-	lv_ctrl23	22

プール	Storage 1	Storage 2	iosv00		iosv01	
			LV	IOT	LV	IOT
	-	LD0-9	-	-	lv_ctrl24	23
POOL1	-	LD1-2	-	-	lv_ctrl25	24
	-	LD1-3	-	-	lv_ctrl26	25
	-	LD1-4	-	-	lv_ctrl27	26
	-	LD1-5	-	-	lv_ctrl28	27
	-	LD1-6	-	-	lv_ctrl29	28
	-	LD1-7	-	-	lv_ctrl30	29
	-	LD1-8	-		lv_ctrl31	30
	-	LD1-9	-		lv_ctrl32	31

※LD0-XのXはパーティションを指します。

【標準モデル向けIOサーバv4+以降】

以下にデータ領域のHDD(12TB)の場合の設計例を記載します。

- ScaTeFSのデータ領域

  ストライピングなしでLVを作成します。

LVM構成

プール	Storage 1	Storage 2	iosv00			iosv01		
			LV	IOT	順序	LV	IOT	順序
POOL2	LD2	-	lv_data01	0	1	-	-	-
POOL3	LD3	-	lv_data02	1	3	-	-	-
POOL4	LD4	-	lv_data03	2	5	-	-	-
POOL5	LD5	-	lv_data04	3	7	-	-	-
POOL6	LD6	-	lv_data05	4	8	-	-	-
POOL7	LD7	-	lv_data06	5	6	-	-	-
POOL8	LD8	-	lv_data07	6	4	-	-	-
POOL9	LD9	-	lv_data08	7	2	-	-	-
POOL2	-	LD2	-	-	-	lv_data09	8	1
POOL3	-	LD3	-	-	-	lv_data10	9	3
POOL4	-	LD4	-	-	-	lv_data11	10	5

プール	Storage 1	Storage 2	iosv00			iosv01		
			LV	IOT	順序	LV	IOT	順序
POOL5	-	LD5	-	-	-	lv_data12	11	7
POOL6	-	LD6	-	-	-	lv_data13	12	5
POOL7	-	LD7	-	-	-	lv_data14	13	6
POOL8	-	LD8	-	-	-	lv_data15	14	4
POOL9	-	LD9	-	-	-	lv_data16	15	2

上記 LVM 構成の IO ターゲットを使用する順序を記載します。

IOサーバ	IOターゲットを使用する順序
iosv00	0 7 1 6 2 5 3 4
iosv01	8 15 9 14 10 13 11 12

5.3.1 ScaTeFS 作成の設定項目 iotid の値を記載します。

iosv00、iosv01 の IO ターゲットを使用する順序を続けて記載します。

項目	設定値
iotid	0 7 1 6 2 5 3 4 8 15 9 14 10 13 11 12

- ScaTeFSのメタデータ領域

  ストライピングなしでLVを作成します。

  LVM構成

プール	Storage 1	Storage 2	iosv00		iosv01	
			LV	IOT	LV	IOT
POOL0	LD0-2	-	lv_ctrl01	0	-	-
	LD0-3	-	lv_ctrl02	1	-	-
	LD0-4	-	lv_ctrl03	2	-	-
	LD0-5	-	lv_ctrl04	3	-	-
POOL1	LD1-2	-	lv_ctrl05	4	-	-
	LD1-3	-	lv_ctrl06	5	-	-
	LD1-4	-	lv_ctrl07	6	-	-
	LD1-5	-	lv_ctrl08	7	-	-
POOL0	-	LD0-2	-	-	lv_ctrl09	8

プール	Storage 1	Storage 2	iosv00		iosv01	
			LV	IOT	LV	IOT
	-	LD0-3	-	-	lv_ctrl10	9
	-	LD0-4	-	-	lv_ctrl11	10
	-	LD0-5	-	-	lv_ctrl12	11
POOL1	-	LD1-2	-	-	lv_ctrl13	12
	-	LD1-3	-	-	lv_ctrl14	13
	-	LD1-4	-	-	lv_ctrl15	14
	-	LD1-5	-	-	lv_ctrl16	15

※LD0-XのXはパーティションを指します。

5.1.3　CLUSTERPRO のクラスタ構成情報作成

IOサーバの構築前にCLUSTERPROのクラスタ構成情報を作成します。作成したクラスタ構成情報は、「5.4 CLUSTERPROの設定」で使用します。

付録A「CLUSTERPROのクラスタ構成情報作成手順(オフラインバージョン)」を参照してください。

5.1.4　NEC Storage Manager Agent Utility(iSMagent)のインストール

【標準モデル向けIOサーバv4+以降】

IOサーバ構築手順の簡易化により、5.1.5へ進んでください。

【標準モデル向けIOサーバv1,v3,v4】

マニュアル「WebSAM iStorageManager インストールガイド」のNEC Storage Manager Agent Utility の導入 (Linux 版) を参照してください。

5.1.5　ホスト登録

ストレージ設定のアクセスコントロールを停止します。

ストレージの管理画面(iStorageManager)からアクセスコントロールの停止を実行します。

マニュアル「iStorage ソフトウェア 構成設定の手引 (GUI 編) - M シリーズ」の「10.3.3.4 アクセスコントロールの詳細設定」を参照してください。

【標準モデル向けIOサーバv4+以降】

IOサーバ構築手順の簡易化により、5.1.7へ進んでください。

【標準モデル向けIOサーバv1,v3,v4】

マニュアル「WebSAM iStorageManager インストールガイド」の「付録I ホスト情報の収

集・登録による構成設定簡易化」の「(1) –② 新規Linux サーバにおけるホスト情報のディスクアレイを経由した収集」を参照してください。

対象ファイルを確認します。※/sys/class/fc_host/配下のhost*が対象です。

```
# ls -l /sys/class/fc_host/host*/issue_lip
--w------- 1 root root 4096 Apr  8 16:36 /sys/class/fc_host/host1/issue_lip
--w------- 1 root root 4096 Apr  8 16:36 /sys/class/fc_host/host2/issue_lip
--w------- 1 root root 4096 Apr  8 16:36 /sys/class/fc_host/host3/issue_lip
--w------- 1 root root 4096 Apr  8 16:36 /sys/class/fc_host/host4/issue_lip
```

ボリュームをOSに認識させます。

```
# echo "1" > /sys/class/fc_host/host1/issue_lip
# echo "1" > /sys/class/fc_host/host2/issue_lip
# echo "1" > /sys/class/fc_host/host3/issue_lip
# echo "1" > /sys/class/fc_host/host4/issue_lip
```

ホスト情報収集コマンド(iSMcc_hostinfo コマンド)を実行します。

```
# iSMcc_hostinfo -store
iSMcc_hostinfo: Info:     iSM11700: Please wait a minute.
iSMcc_hostinfo: Info:     iSM11770: Host Information was exported successfully. (Disk
Array=iost05) (code=5ec6-5900-00a2-0000)
iSMcc_hostinfo: Info:     iSM11770: Host Information was exported successfully. (Disk
Array=iost07) (code=5ec6-5900-00a2-0000)
iSMcc_hostinfo: Info:     iSM11770: Host Information was exported successfully. (Disk
Array=iost08) (code=5ec6-5900-00a2-0000)
iSMcc_hostinfo: Info:     iSM11770: Host Information was exported successfully. (Disk
Array=iost06) (code=5ec6-5900-00a2-0000)
iSMcc_hostinfo: Info:     iSM11100: Command has completed successfully.
```

注：

構成によっては、以下のワーニングメッセージが出力される場合がありますが、特に問題ありません。

```
# iSMcc_hostinfo -store
iSMcc_hostinfo: Info:     iSM11700: Please wait a minute.
iSMcc_hostinfo: Warning:  iSM11773: Information collection was skipped. (IP Address)
(code=2fa3-5700-0001-0000)
iSMcc_hostinfo: Warning:  iSM11774: A part of Host Information was exported. (Disk
Array=iost05) (code=2fa3-5900-00a0-0000)
```

```
iSMcc_hostinfo: Warning:   iSM11774: A part of Host Information was exported. (Disk
Array=iost06) (code=2fa3-5900-00a0-0000)
iSMcc_hostinfo: Warning:   iSM11774: A part of Host Information was exported. (Disk
Array=iost08) (code=2fa3-5900-00a0-0000)
iSMcc_hostinfo: Warning:   iSM11774: A part of Host Information was exported. (Disk
Array=iost07) (code=2fa3-5900-00a0-0000)
iSMcc_hostinfo: Warning:    iSM11775: Command has completed with warning status.
(code=2fa3-2703-0004-0000)
```

5.1.6　論理ディスク割り当て

【標準モデル向けIOサーバv4+以降】

IOサーバ構築手順の簡易化により、5.1.7へ進んでください。

【標準モデル向けIOサーバv1,v3,v4】

ストレージの管理画面(iStorageManager)から接続しているIOサーバへ論理ディスクの割り当てを行います。

IOサーバへ論理ディスクを割り当てます。

ストレージに作成したすべての論理ディスクをIOサーバへ割り当てます。

マニュアル「iStorage ソフトウェア 構成設定の手引（GUI 編）- M シリーズ」の「10.1 論理ディスクの割り当て」を参照してください。

ストレージ設定のアクセスコントロールを開始します。

ストレージの管理画面(iStorageManager)からアクセスコントロールの開始を実行します。

マニュアル「iStorage ソフトウェア 構成設定の手引（GUI 編）- M シリーズ」の「10.3.3.4 アクセスコントロールの詳細設定」を参照してください。

5.1.7　StoragePathSavior for Linux driver package(SPS)のインストールと設定

下記の手順でSPSパッケージをインストールしてください。手順の詳細は、マニュアル「iStorage StoragePathSavior for Linux インストールガイド」と、「iStorage ソフトウェア StoragePathSavior 利用の手引(Linux 編)」を参照してください。

(1) インストール

a) sg3_utils、lvm2 パッケージがインストールされていない場合には、OS ディストリビューションからインストールしてください。

b) マウント先のディレクトリへ移動します。

c) インストールスクリプトを実行します。

```
# sh install.sh -i --silent
```

インストール後にOSのリブートを行うので注意してください。

(2) iStorage のデバイスの確認

sg_scan コマンドを使用し、OS に認識されている SCSI ディスクを確認します。

"NEC"、"DISK ARRAY"と表示されれば、iStorage のデバイスと判断できます。

```
# sg_scan -i /dev/sdc
/dev/sdc: scsi8 channel=0 id=0 lun=0 [em]
   NEC      DISK ARRAY       1000 [rmb=0 cmdq=1 pqual=0 pdev=0x0]
#
```

(3) LVM の設定ファイル(/etc/lvm/lvm.conf)の変更

下記の手順は「iStorage StoragePathSavior for Linux インストールガイド　付録 B LVM の設定」を参照してください。

a) デバイスのフィルタ設定

[RHEL7 の場合]

devices{} 内の「global_filter」設定を変更します。

すべての SPS デバイスを許可する場合の例を記載します。

```
global_filter = [ "a|/dev/dd.*|", "r|/dev/.*|" ]
```

[RHEL6 の場合]

devices{} 内の「filter」設定を変更します。

すべての SPS デバイスを許可する場合の例を記載します。

```
filter = [ "a|/dev/dd.*|", "r|/dev/.*|" ]
```

b) devices{} 内に types を追記

```
types = [ "dd", 16 ]
```

5.1.8 CLUSTERPRO X for Linux のインストール

下記の手順でCLUSTERPROパッケージをインストールしてください。手順の詳細は、マニュアル「CLUSTERPRO X for Linux インストール&設定ガイド」を参照してください。

CLUSTERPROの設定については、本書「5.4 CLUSTERPRO設定」を参照してください。

(1) インストール

CLUSTERPRO パッケージをインストールします

```
# rpm -ivh clusterpro-<version>.<architecture>.rpm
```

(2) ライセンス登録

ライセンスファイルを指定してライセンス登録を行います。

[CLUSTERPRO X 4.x]

```
# clplcnsc -i filepath
```

[CLUSTERPRO X 3.x]

```
[RHEL7]
# clplcnsc -i filepath -p BASE33
[RHEL6]
# clplcnsc -i filepath -p BASE32
```

(3) [RHEL7]LVM メタデータデーモンの設定

「CLUSTERPRO X for Linux スタートアップガイド　第 5 章 注意制限事項　LVM メタデータデーモンの設定」を参照し、必ず設定を変更してください。

a) LVM メタデータデーモンの停止

systemctl コマンドで LVM メタデータデーモンを停止します。

```
# systemctl stop lvm2-lvmetad.service
```

b) LVM の設定ファイル(/etc/lvm/lvm.conf)の変更

use_lvmetad の値を 0 に変更します。

```
use_lvmetad = 0
```

5.1.9　DCB 対応版 10GbE-NIC ドライバのインストール

DCB対応版10GbE-NICを使用する場合のみ実施してください。

ベンダが提供しているRPMバイナリパッケージはそのままではDCBに対応してない場合があります。サポート部門から入手したインストール手順にて10GbE-NICドライバのインストールを行ってください。

5.1.10 IB ドライバのインストール

IB HCAを使用する場合のみ実施してください。

(1) ISO ファイルのダウンロード

IO サーバがサポートする MLNX_OFED のバージョンは以下のとおりです。

OS	MLNX_OFEDのバージョン
RHEL7.3	4.2-1.2.0.0
RHEL7.4	
RHEL7.6	4.6-4.1.2.0
RHEL7.7	4.7-1.0.0.1

該当するバージョンのMLNX_OFED を、NVIDIA社の公式サイトから入手します。
https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/

※MLNX_OFED 4.6-4.1.2.0 は上記URLでは公開されていません。以下のURLからダウンロードしてください。
https://mellanox.my.salesforce.com/sfc/p/#500000007heg/a/1T000000cCrw/jAKX3brAtwtWng6sVqHpSXf2pT8UrSUL2rMKpn3c4ng
パスワード: mgIdJQfI

MLNX_OFEDが入手できない場合は、サポート部門までお問い合わせください。

(2) インストール

a) 下記のパッケージがインストールされていない場合には、OS ディストリビューションからインストールしてください。
lsof gtk2 atk cairo tcl tcsh tk pciutils

b) ダウンロードした ISO ファイルを任意のマウントポイントにマウントします。以下は/mnt/iso ディレクトリにマウントする場合の実行例です。

```
# mount -t iso9660 -o loop MLNX_OFED_LINUX-4.2-1.2.0.0-rhel7.3-x86_64.iso /mnt/iso
```

c) インストールスクリプトを実行します。

```
# /mnt/iso/mlnxofedinstall
```

IB関連のパッケージを削除してインストールを継続するか確認されるので、"y"を入力します。

```
    This program will install the MLNX_OFED_LINUX package on your machine.
Note that all other Mellanox, OEM, OFED, or Distribution IB packages will be removed.
```

```
Do you want to continue?[y/N]:
```

d) ISO ファイルをアンマウントします。

```
# umount /mnt/iso
```

(3) OS を再起動してインストールしたドライバをロード

```
# reboot
```

5.1.11 rsh 関連のインストール

リモートシェル(rsh)関連のパッケージがインストールされていない場合には、OSディストリビューションから以下をインストールしてください。

- rsh
- rsh-server
- xinetd ※RHEL6の場合

[RHEL7の場合]
rshのサーバ機能を有効にします。

```
# systemctl enable rsh.socket
```

viなどのエディタで/etc/systemd/system/sockets.target.wants/rsh.socketを開き、以下の項目を追加します。

追加
[Unit] Description=Remote Shell Facilities Activation Socket [Socket] ListenStream=514 Accept=true MaxConnections=10000 ★追加 [Install] WantedBy=sockets.target

rshのサーバ機能を起動します。

```
# systemctl start rsh.socket
```

```
# systemctl daemon-reload
```

[RHEL6の場合]

viなどのエディタで/etc/xinetd.d/rshを開き、以下の項目を追加および削除します。

追加
per_source　　= UNLIMITED instances　　= UNLIMITED cps　　= 10000 10

削除
log_on_success　　+= USERID log_on_failure　　+= USERID

rshのサーバ機能を有効にします。

```
# chkconfig rsh on
# /etc/init.d/xinetd start
```

5.1.12 ScaTeFS パッケージのインストール

すべてのIOサーバノードに、ScaTeFS/Serverパッケージをインストールしてください。

scatefs-srvパッケージは、全IOサーバでバージョンを一致させてください。

なお、バージョン3.5よりsosパッケージを事前にインストールしてください。

以下にScaTeFSのライセンスごとのインストール、アップデート手順を記載します。

5.1.12.1 HPC ソフトウェアライセンスをお使いの場合

ScaTeFS/Serverパッケージのインストールでは、以下のパッケージを使用します。

a)　ScaTeFS/Server

b)　TSUBASA-soft-release-ve1

c)　ライセンスアクセスライブラリ

a)は有償であり、ScaTeFS/ServerのPPサポートを契約しているかどうかによりパッケージの入手方法が異なります。

b)とc)は無償であり、NECの無償yumリポジトリに登録されているパッケージをyumコマンドを使用してインストールします。

c)は 5.1.13 でインストールします。

ScaTeFS/Serverのパッケージのインストール、アップデート、およびアンインストールの

手順は、ScaTeFS/ServerのPPサポートを契約しているかどうかにより異なります。以降では、PPサポートを契約している場合とPPサポートを契約していない場合に分けて説明します。

【ScaTeFS/ServerのPPサポートを契約している場合】

(1)　yum リポジトリ設定 (ScaTeFS/Server の PP サポート契約あり)

必要なソフトウェアのインストールを行うため、yumリポジトリを設定します。yumリポジトリは、インターネット上のものをオンラインで利用する、もしくはローカルに構築してオフラインで利用することができます。

yumリポジトリの設定の手順については、「SX-Aurora TSUBASA インストレーションガイド」の「3.1 インストールのための事前準備」をご参照ください。このとき、文中のVHを対象マシンに置き換えてお読みください。また、アーキテクチャについてはVE1の方をお読みください。

[RHEL7.3～RHEL7.6の場合]

有償yumリポジトリ設定ファイルのbaseurlの末尾をscatefs_el7.7 に変更してください。

(2)　ScaTeFS パッケージのインストール (ScaTeFS/Server の PP サポート契約あり)

ScaTeFS/Serverのパッケージをインストールします。

```
# /opt/nec/ve/sbin/TSUBASA-groups-remark.sh scatefs-server
# yum group install scatefs-server
```

ファイルシステムの統計情報をリアルタイムに収集しモニタリングする機能を使用する場合、モニタリングのパッケージをインストールします。

```
# /opt/nec/ve/sbin/TSUBASA-groups-remark.sh scatefs-server-monitoring
# yum group install scatefs-server-monitoring
```

(3)　ScaTeFS パッケージのアップデート (ScaTeFS/Server の PP サポート契約あり)

ScaTeFS/Serverのパッケージをアップデートします。

```
# /opt/nec/ve/sbin/TSUBASA-groups-remark.sh scatefs-server
# yum group update scatefs-server
```

ファイルシステムの統計情報をリアルタイムに収集しモニタリングする機能を使用している場合、モニタリングのパッケージをアップデートします。

```
# systemctl stop zabbix-agent
# /opt/nec/ve/sbin/TSUBASA-groups-remark.sh scatefs-server-monitoring
# yum group update scatefs-server-monitoring
```

```
# systemctl start zabbix-agent
```

【ScaTeFS/ServerのPPサポートを契約していない場合】

(1)　yum リポジトリ設定 (ScaTeFS/Server の PP サポート契約なし)

必要なソフトウェアのインストールを行うため、yumリポジトリを設定します。yumリポジトリは、インターネット上のものをオンラインで利用する、もしくはローカルに構築してオフラインで利用することができます。

yumリポジトリの設定の手順については、「SX-Aurora TSUBASA インストレーションガイド」の「3.1 インストールのための事前準備」をご参照ください。このとき、文中のVHを対象マシンに置き換えてお読みください。また、アーキテクチャについてはVE1の方をお読みください。

(2)　ScaTeFS/Server のパッケージの入手

「インターネット配信製品ダウンロードサービス」を利用してScaTeFS/Serverのパッケージを含むzipファイルをダウンロードしてください。

ダウンロードしたら、インストール対象マシンに転送し、zipファイルを展開します。

```
# unzip ScaTeFS_S-YYYYMMDD.zip
(YYYYMMDDは年月日)
```

(3)　ScaTeFS パッケージのインストール (ScaTeFS/Server の PP サポート契約なし)

ScaTeFS/Serverのパッケージをインストールします。

[RHEL/CentOS7.7]

```
# yum install scatefs-server/el7.7/scatefs-server/*.rpm
```

[RHEL7.6]

```
# yum install scatefs-server/el7.6/scatefs-server/*.rpm
```

ファイルシステムの統計情報をリアルタイムに収集しモニタリングする機能を使用する場合、モニタリングのパッケージをインストールします。

[RHEL/CentOS7.7]

```
# yum install scatefs-server/el7.7/scatefs-server-monitoring/*.rpm
```

[RHEL7.6]

```
# yum install scatefs-server/el7.6/scatefs-server-monitoring/*.rpm
```

5.1.12.2 SX クロスソフトウェア ノードロックライセンスをお使いの場合

ScaTeFS/Serverパッケージのインストールでは、以下のパッケージを使用します。

a)　ScaTeFS/Server

すべてのIOサーバノードに、以下のパッケージをインストールしてください。

```
scatefs-srv-VER.x86_64
```

また、もしインストール済みのパッケージより新しいバージョンのパッケージがある場合には、以下のようにパッケージのアップデートを行ってください。

```
# rpm -Uvh scatefs-srv-VER.x86_64.rpm
```

5.1.13 ScaTeFS のライセンス登録

ライセンス登録を行います。

手順の詳細については、「HPC ソフトウェアライセンス管理説明書」をご覧ください。

※SXクロスソフトウェア ノードロックライセンスをお使いの場合は、「HPC ソフトウェアライセンス管理説明書」ではなく「SXクロスソフトウェア ノードロックライセンス導入ガイド」をご覧ください。

5.1.14 SELinux 無効化

SELinuxの有効・無効は以下のコマンドで確認できます。

```
# /usr/sbin/getenforce
Disabled
```

もしEnabledまたはEnforcingと表示された場合には、/etc/selinux/configファイルを編集し、「SELINUX=disabled」とします。設定を有効にするにはOSの再起動が必要です。

5.1.15 ファイアウォール無効化

[RHEL7の場合]

systemctl コマンドでファイアウォールの設定を確認します。

```
# systemctl list-unit-files|grep firewalld
firewalld.service                           enabled
```

ファイアウォールが有効(enabled)の場合、以下の手順で無効化を行います。

```
# systemctl disable firewalld
# systemctl list-unit-files|grep firewalld
firewalld.service                           disabled
# systemctl stop firewalld
```

[RHEL6の場合]

chkconfigコマンドでファイアウォールの設定を確認します。

```
# /sbin/chkconfig --list iptables
iptables       0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

ファイアウォールが有効の場合、以下の手順で無効化を行います。

```
# /sbin/chkconfig iptables off
# /sbin/chkconfig --list iptables
iptables       0:off   1:off   2:off   3:off   4:off   5:off   6:off
# /etc/init.d/iptables stop
```

5.1.16 prelink 無効化

[RHEL6の場合]

/etc/sysconfig/prelinkファイルを編集し、「PRELINKING=no」とします。

```
# vi /etc/sysconfig/prelink
            ----
            PRELINKING=no
            ----
```

prelinkの無効化を行います。

```
# prelink -ua
```

注：以下のエラーメッセージが出力される場合がありますが、特に問題ありません。

```
prelink: /usr/lib64/samba/libserver-role-samba4.so: Could not find one of the
dependencies
prelink: /usr/lib64/firefox/plugin-container: Could not find one of the dependencies
```

5.1.17 abrtd 無効化

[RHEL7の場合]

systemctlコマンドでABRT関連サービスの設定を確認します。

```
# systemctl list-unit-files|grep abrt
abrt-ccpp.service                          enabled
abrt-oops.service                          enabled
abrt-pstoreoops.service                    disabled
abrt-vmcore.service                        enabled
abrt-xorg.service                          enabled
abrtd.service                              enabled
※ABRT関連サービスが表示されなければ設定不要です。
```

ABRT関連サービスが有効の場合、以下の手順で無効化を行います。

```
# yum remove abrt abrt-libs
```

[RHEL6の場合]

chkconfigコマンドでabrtdの設定を確認します。

```
# /sbin/chkconfig --list abrtd
abrtd          0:off   1:off   2:off   3:on    4:off   5:on    6:off
```

abrtdが有効の場合、以下の手順で無効化を行います。

```
# /sbin/chkconfig abrtd off
# /sbin/chkconfig --list abrtd
abrtd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
# /etc/init.d/abrtd stop
```

5.1.18 ネットワークの設定

IOサーバでは複数のネットワークポートを使用します。それぞれについて、IPアドレスの設定等を行ってください。

- 運用・管理ポート

  IOサーバにネットワーク経由でログインする際に使用します。また、ntpによるサーバ間の時刻同期や、ScaTeFSコマンドの実行でのサーバ間の通信等に使用します。

- ファイルシステムポート(10GbE)

  ScaTeFSクライアントからのファイルアクセスに使用します。同一NICのポートは、bondingにより 2 つのポートを束ねる設定をしてください。また、CLUSTERPROによりフローティングIPアドレスを設定します。

- ファイルシステムポート(IB)

  ScaTeFSクライアントからのファイルアクセスに使用します。インターフェース設定ファ

イルを作成し、設定を行ってください。また、CLUSTERPROによりフローティングIPアドレスを設定します。

- IOサーバ間インタコネクト用ポート

  IOサーバでノード間の通信に使用します。ペアとなるIOサーバ間に閉じているネットワークであり、ご使用のネットワーク環境と衝突しないネットワークアドレスを設定してください。IPアドレスは2つ必要で、サーバID(SID)が偶数のIOサーバと奇数のIOサーバで異なるIPアドレスを設定してください。

以下にファイルシステムポート(10GbE)とIOサーバ間インタコネクト用ポートのネットワークの設定例を記載します。

5.1.18.1 ファイルシステムポート(10GbE)と IO サーバ間インタコネクト用ポートのネットワークインターフェースの設定(bonding)

以下の例を元にbondingの設定方法を記載します。ファイルシステムポートとして10GbEを使用しない場合は、ファイルシステムポートのbondingは不要です。IOサーバ間インタコネクト用ポートのbonding設定のみ行ってください。

例)

対象マシン：IOサーバ

ファイルシステムポート

[RHEL7の場合]

ens28f4, ens28f4d1：bond0(172.16.6.6)

ens61f4, ens61f4d1：bond1(172.16.7.6)

[RHEL6の場合]

eth0,eth1：bond0(172.16.6.6)

eth2,eth3：bond1(172.16.7.6)

netmask：255.255.255.128

bond0のvlanid：12

bond1のvlanid：14

IOサーバ間(IOサーバ0,IOサーバ1)インタコネクト用ポート

[RHEL7の場合]

IOサーバ0 ens27f0eth4, ens27f1eth5：bond2(10.2.0.10)

IOサーバ1 ens27f0, ens27f1eth4,eth5：bond2(10.2.0.11)

[RHEL6の場合]

IOサーバ0 eth4,eth5：bond2(10.2.0.10)

IOサーバ1 eth4,eth5：bond2(10.2.0.11)

netmask：255.255.255.0

Red Hat Enterprise Linux では、bonding カーネルモジュールと、チャンネルボンディングインターフェース と呼ばれる特殊なネットワークインターフェースを使用して、管理者が複数のネットワークインターフェースを単一のチャンネルにまとめてバインドすることが可能です。

[RHEL7の場合]

チャンネルボンディングインターフェースを作成するには、nmcli、nmtuiコマンドなどでネットワークインターフェースファイルを作成します。ポートごとのnmcliコマンド実行例は以下のとおりです。

[ファイルシステムポート(10GbE)]

bond0.12

```
# nmcli connection add type bond con-name bond0 ifname bond0
# nmcli connection add type ethernet autoconnect yes ifname ens28f4 master bond0
# nmcli connection add type ethernet autoconnect yes ifname ens28f4d1 master bond0
# nmcli connection modify bond0 ipv4.never-default true
# nmcli connection modify bond0 ipv4.method disabled ipv6.method ignore
# nmcli connection modify bond0 +bond.options
mode=802.3ad,miimon=100,xmit_hash_policy=layer2+3
# nmcli connection up bond0
# nmcli connection add type vlan con-name bond0.12 dev bond0 id 12
# nmcli connection modify bond0.12 ipv4.never-default true
# nmcli connection modify bond0.12 ipv4.method disabled ipv6.method ignore
```

bond1.14

```
# nmcli connection add type bond con-name bond1 ifname bond1
# nmcli connection add type ethernet autoconnect yes ifname ens61f4 master bond1
# nmcli connection add type ethernet autoconnect yes ifname ens61f4d1 master bond1
# nmcli connection modify bond1 ipv4.never-default true
# nmcli connection modify bond1 ipv4.method disabled ipv6.method ignore
# nmcli connection modify bond1 +bond.options mode=802.3ad,miimon=100,xmit_hash_policy=layer2+3
# nmcli connection up bond1
# nmcli connection add type vlan con-name bond1.14 dev bond1 id 14
# nmcli connection modify bond1.14 ipv4.never-default true
# nmcli connection modify bond1.14 ipv4.method disabled ipv6.method ignore
```

[IOサーバ間インタコネクト用ポート]

bond2

```
# nmcli connection add type bond con-name bond2 ifname bond2
# nmcli connection add type ethernet autoconnect yes ifname ens27f0 master bond2
```

bond2

```
# nmcli connection add type ethernet autoconnect yes ifname ens27f1 master bond2
# nmcli connection modify bond2 ipv4.never-default true
# nmcli connection modify bond2 ipv6.method ignore
# nmcli connection modify bond2 +bond.options mode=802.3ad,miimon=100,xmit_hash_policy=layer2+3
・IO サーバ 0
  # nmcli connection modify bond2 ipv4.method manual ipv4.address "10.2.0.10/24"
  # nmcli connection up bond2
・IO サーバ 1
  # nmcli connection modify bond2 ipv4.method manual ipv4.address "10.2.0.11/24"
  # nmcli connection up bond2
```

[RHEL6の場合]

チャンネルボンディングインターフェースを作成するには、/etc/sysconfig/network-scripts ディレクトリに ifcfg-bondN という名前のファイルを作成し、N をそのインターフェースの番号 0 などに置き換えます。チャンネルボンディング設定ファイルの内容は以下のとおりです。

[ファイルシステムポート(10GbE)]

/etc/sysconfig/network-scripts/ifcfg-bond0(新規作成)

```
DEVICE=bond0
BOOTPROTO=none
NM_CONTROLLED=yes
ONBOOT=yes
IPV6INIT=no
USERCTL=no
BONDING_OPTS="mode=802.3ad miimon=100 xmit_hash_policy=layer2+3"
```

/etc/sysconfig/network-scripts/ifcfg-bond1 (新規作成)

```
DEVICE=bond1
BOOTPROTO=none
NM_CONTROLLED=yes
ONBOOT=yes
IPV6INIT=no
USERCTL=no
BONDING_OPTS="mode=802.3ad miimon=100 xmit_hash_policy=layer2+3"
```

/etc/sysconfig/network-scripts/ifcfg-bond0.12 (新規作成)

```
DEVICE=bond0.12
BOOTPROTO=none
ONBOOT=yes
IPV6INIT=no
USERCTL=no
BONDING_OPTS="mode=802.3ad miimon=100 xmit_hash_policy=layer2+3"
VLAN=yes
```

/etc/sysconfig/network-scripts/ifcfg-bond1.14 (新規作成)

```
DEVICE=bond1.14
BOOTPROTO=none
ONBOOT=yes
IPV6INIT=no
USERCTL=no
BONDING_OPTS="mode=802.3ad miimon=100 xmit_hash_policy=layer2+3"
```

/etc/sysconfig/network-scripts/ifcfg-bond1.14 (新規作成)
VLAN=yes

[IO サーバ間インタコネクト用ポート]

/etc/sysconfig/network-scripts/ifcfg-bond2 (新規作成) IOサーバ0
DEVICE=bond2 BOOTPROTO=none NM_CONTROLLED=yes ONBOOT=yes IPADDR=10.2.0.10 IPV6INIT=no USERCTL=no NETMASK=255.255.255.0 BONDING_OPTS="mode=802.3ad miimon=100 xmit_hash_policy=layer2+3"

/etc/sysconfig/network-scripts/ifcfg-bond2 (新規作成) IOサーバ1
DEVICE=bond2 BOOTPROTO=none NM_CONTROLLED=yes ONBOOT=yes IPADDR=10.2.0.11 IPV6INIT=no USERCTL=no NETMASK=255.255.255.0 BONDING_OPTS="mode=802.3ad miimon=100 xmit_hash_policy=layer2+3"

IOサーバ間インタコネクト用ポートの場合はCLUSTERPROの設定に関わらずIPADDR、NETMASKを設定してください。

チャンネルボンディング設定ファイルを作成した後に、バインドされるネットワークインターフェースを設定するには、その設定ファイルに MASTER 指示文と SLAVE 指示文を追加する必要があります。

チャンネルボンディングされたインターフェースの各設定ファイルは、ほぼ同一となる場合があります。

[ファイルシステムポート(10GbE)]

/etc/sysconfig/network-scripts/ifcfg-eth0(変更)
DEVICE=eth0 BOOTPROTO=none HWADDR=00:07:43:13:59:E0 NM_CONTROLLED=yes

/etc/sysconfig/network-scripts/ifcfg-eth0(変更)

```
ONBOOT=yes
TYPE=Ethernet
UUID="a328a3bb-bd19-4b46-ab89-920203554a42"
IPV6INIT=no
USERCTL=no
MASTER=bond0 ★追加
SLAVE=yes ★追加
```

/etc/sysconfig/network-scripts/ifcfg-eth1(変更)

```
DEVICE=eth1
BOOTPROTO=none
HWADDR=00:07:43:13:59:E8
NM_CONTROLLED=yes
ONBOOT=yes
TYPE=Ethernet
UUID="77227c15-4565-40c0-8c73-9e04f329ac6b"
IPV6INIT=no
USERCTL=no
MASTER=bond0 ★追加
SLAVE=yes　★追加
```

/etc/sysconfig/network-scripts/ifcfg-eth2 (変更)

```
DEVICE=eth2
BOOTPROTO=none
HWADDR=00:07:43:13:56:C0
NM_CONTROLLED=yes
ONBOOT=yes
TYPE=Ethernet
UUID="f90a96a8-8ec6-4003-a22d-cccad74bb6a7"
IPV6INIT=no
USERCTL=no
MASTER=bond1 ★追加
SLAVE=yes ★追加
```

/etc/sysconfig/network-scripts/ifcfg-eth3(変更)

```
DEVICE=eth3
BOOTPROTO=none
HWADDR=00:07:43:13:56:C8
NM_CONTROLLED=yes
ONBOOT=yes
TYPE=Ethernet
```

/etc/sysconfig/network-scripts/ifcfg-eth3(変更)

```
UUID="a7d977b0-2250-42a5-a153-3228ea64d05d"
IPV6INIT=no
USERCTL=no
MASTER=bond1 ★追加
SLAVE=yes ★追加
```

[IO サーバ間インタコネクト用ポート]

/etc/sysconfig/network-scripts/ifcfg-eth4(変更)

```
DEVICE=eth4
BOOTPROTO=none
HWADDR=8C:89:A5:5F:3E:A9
NM_CONTROLLED=yes
ONBOOT=yes
TYPE=Ethernet
UUID="61e28110-46fb-4bf6-b308-e2aacf7b11e0"
IPV6INIT=no
USERCTL=no
MASTER=bond2 ★追加
SLAVE=yes ★追加
```

/etc/sysconfig/network-scripts/ifcfg-eth5(変更)

```
DEVICE=eth5
BOOTPROTO=none
HWADDR=8C:89:A5:5F:3E:AB
NM_CONTROLLED=yes
ONBOOT=yes
TYPE=Ethernet
UUID="59eab730-1ac2-4593-b988-7c9f83717a17"
IPV6INIT=no
USERCTL=no
MASTER=bond2 ★追加
SLAVE=yes ★追加
```

チャンネルボンディングインターフェースを有効にするには、カーネルモジュールがロードされている必要があります。

チャンネルボンディングインターフェースがアクティブになった時にモジュールが確実にロードされるようにするには、/etc/modprobe.d ディレクトリに bonding.conf という名前の新規ファイルを root で作成します。

ファイル名は、末尾に .conf の拡張子が付いていれば、どのような名前にすることもでき

ます。

/etc/modprobe.d/bonding.conf(新規作成)
alias netdev-bond0 bonding alias netdev-bond1 bonding alias netdev-bond2 bonding

すべての設定ファイルが用意できた後、設定を反映するためにIOサーバを再起動してください。

再起動した後はifconfigを実行して、設定したbond0、bond1、bond2が表示されているか確認してください。

5.1.18.2 ファイルシステムポート(IB)のネットワークインターフェースの設定

[RHEL7の場合]

nmcli、nmtuiコマンドなどでネットワークインターフェースファイルを作成します。ファイルシステムポート(IB)のネットワークインターフェースファイルのnmcliコマンド実行例は以下のとおりです。

[HCA 1portの場合]

ib0
nmcli connection modify ib0 connection.autoconnect yes # nmcli connection modify ib0 ipv4.never-default true # nmcli connection modify ib0 ipv4.method disabled ipv6.method ignore # nmcli connection up ib0

[HCA 2portの場合]

ibbond0
nmcli connection add type bond con-name ibbond0 ifname ibbond0 # nmcli connection add type infiniband autoconnect yes ifname ib0 master ibbond0 # nmcli connection add type infiniband autoconnect yes ifname ib1 master ibbond0 # nmcli connection modify ibbond0 ipv4.never-default true # nmcli connection modify ibbond0 ipv4.method disabled ipv6.method ignore # nmcli connection modify ibbond0 802-3-ethernet.mtu 2044 # nmcli connection modify ibbond0 +bond.options mode=active-backup,primary=ib0,miimon=100 # nmcli connection up ibbond0

すべての設定ファイルが用意できた後、設定を反映するためにIOサーバを再起動してください。

再起動した後はipコマンドを実行して、設定したネットワークインターフェースが表示されているか確認してください。

5.1.18.3 ルーティング設定

10GbEを使用する場合のみ必要な設定です。

RHEL7の場合、ダウンロードセンターから下記RPMパッケージを入手してインストールします。

NetworkManager-dispatcher-routing-rules

例としてbond0(IPアドレス：10.0.0.10、ゲートウェイ：10.0.0.100)の往路と復路を一致させる場合以下の設定を行います。

ip ruleの設定

/etc/sysconfig/network-scripts/rule-bond0の記述イメージ
table 200 from 10.0.0.0/25 ← テーブルIDは200としています。

ルーティングとip routeの設定

/opt/scatefs/script/routeadd.shの記述イメージ
#!/bin/sh # routing add script ## ip route ip route add table 200 10.0.0.0/25 dev bond0.12 proto kernel src 10.0.0.10 ip route add table 200 default via 10.0.0.100 exit 0

routeadd.shはCLUSTERPROがフローティングIPアドレスを設定した後に動作するスクリプトです。

設定が反映されているか確認する場合はIOサーバの構築が完了してから”ip rule”、” ip route show table テーブルID”を実行してください。

5.1.18.4 DCB 設定

DCB対応版の10GbEを使用する場合のみ必要な設定です。

IOサーバでDCBのPriorityを有効にするためにはvconfigコマンドを使用します。

たとえば、VLAN-ID12に属しているbond0.12に対してPriority4、5、6を設定する際は以下のスクリプトを作成してください。また、CLUSTERPRO起動時に本スクリプトが実行されるようにCLUSTERPROの設定をしてください。

下記ファイルを新規に作成します。

```
# vi /opt/scatefs/script/dcb.sh
```

下記内容を記載します。

[RHEL7の場合]

```
#!/bin/sh

ip link set bond0.12 type vlan egress 4:4
ip link set bond0.12 type vlan egress 5:5
ip link set bond0.12 type vlan egress 6:6
ip link set bond0.12 type vlan ingress 4:4
ip link set bond0.12 type vlan ingress 5:5
ip link set bond0.12 type vlan ingress 6:6

ip link set bond1.14 type vlan egress 4:4
ip link set bond1.14 type vlan egress 5:5
ip link set bond1.14 type vlan egress 6:6
ip link set bond1.14 type vlan ingress 4:4
ip link set bond1.14 type vlan ingress 5:5
ip link set bond1.14 type vlan ingress 6:6

exit 0
```

[RHEL6の場合]

```
#!/bin/sh

vconfig set_egress_map bond0.12 4 4
vconfig set_egress_map bond0.12 5 5
vconfig set_egress_map bond0.12 6 6
vconfig set_ingress_map bond0.12 4 4
vconfig set_ingress_map bond0.12 5 5
vconfig set_ingress_map bond0.12 6 6

vconfig set_egress_map bond1.14 4 4
vconfig set_egress_map bond1.14 5 5
vconfig set_egress_map bond1.14 6 6
vconfig set_ingress_map bond1.14 4 4
vconfig set_ingress_map bond1.14 5 5
vconfig set_ingress_map bond1.14 6 6
```

```
exit 0
```

dcb.shの実行権を付与します。

```
# chmod +x /opt/scatefs/script/dcb.sh
```

5.1.19 IPv6 無効化

[RHEL7の場合]

カーネルに組み込まれている ipv6 モジュールを無効にします。

/etc/default/grub を編集し、以下のように GRUB_CMDLINE_LINUX に ipv6.disable=1 を追加します。

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap      crashkernel=auto      rd.lvm.lv=rhel/root
ipv6.disable=1"
```

grub2-mkconfig コマンドを実行して grub.cfg ファイルを再生成します。

```
# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

IOサーバを再起動します。

[RHEL6の場合]

/etc/sysconfig/networkファイルを以下のように変更します。

```
NETWORKING=yes
NETWORKING_IPV6=no ★変更
HOSTNAME=iosv00
```

/etc/modprobe.d/ipv6.conf を作成して、下記のように記述します。

```
options ipv6 disable=1
```

以下のコマンドを実行して、ブート時に ip6tables サービスを無効にします。

```
# chkconfig ip6tables off
```

以下のコマンドを実行して、初期 RAM ディスクイメージを再構築します。

```
# dracut -f
```

IOサーバを再起動します。

5.1.20 時刻の設定

IOサーバ間は時刻同期させることにより、サーバ間の時間を一致させておく必要があります。

[RHEL7の場合]

chronydまたは、ntp,ntpdateの設定を行うことにより、サーバ間の時刻差を解消します。ntpを使用する場合、chronydを停止する必要があります。

[RHEL6の場合]

ntpとntpdateの設定を行うことにより、サーバ間の時刻差を解消します。

```
# vi /etc/ntp.conf
# chkconfig ntpd on
# vi /etc/ntp/step-tickers
# chkconfig ntpdate on
```

ntpdateの使用はオプションです。サーバ間の時刻の差が大きい場合に使用してください。

時刻同期の詳細はRed Hat Enterprise Linux Serverのマニュアルを参照してください。

5.1.21 ファイルシステム管理アカウント(fsadmin)の設定

IOサーバにおけるファイルシステムの管理・運用の操作は、fsadminアカウントにて行います。

fsadminアカウント自体はscatefs-srvパッケージをインストールすることによりIOサーバに作成されますが、このfsadminアカウントでIOサーバ間のリモート実行が可能なように設定を行います。

```
# su - fsadmin
-bash-4.1$ vi .rhosts
-bash-4.1$ chmod 600 .rhosts
-bash-4.1$ exit
#
```

fsadminの.rhostsファイルに全IOサーバのIPアドレスを記載してください。運用・管理ポートとファイルシステムポートの両方のアドレスを記載してください。

設定後は、IOサーバ間でfsadminによるリモート実行が可能なことを確認してください。

```
# su - fsadmin
-bash-4.1$ rsh iosv01 hostname
iosv01
```

```
-bash-4.1$
```

5.1.22 内蔵ディスク(SSD)の設定

IOサーバの内蔵SSDデバイス(/dev/sdb)を下記のように使用します。

なお、IOサーバの内蔵SSDデバイスが/dev/sdb以外で認識する場合がありますので、下記設定は認識したデバイス名に読み替えてください。

デバイス名	マウントポイント	容量	ファイルシステム	説明
/dev/sdb1	/mnt/ssd	10GB	ext4	ジャーナルログ領域
/dev/sdb2	/mnt/core	残りすべて	ext4	dump領域

SSDデバイス(/dev/sdb)を10GBの領域と残りすべての2つのパーティションに切ります。実行するコマンドは以下のとおりです。

```
# parted /dev/sdb
(parted) print
(parted) mkpart primary ext4 0% 10GB
(parted) mkpart primary ext4 10GB 100%
(parted) print
(parted) quit
```

正しくパーティションに切れた場合は、以下のように表示されます。

```
Number  Start   End     Size    Type     File system  Flags
 1      1049kB  10.0GB  9999MB  primary
 2      10.0GB  199GB   189GB   primary
```

パーティション上にファイルシステムを作成し、それぞれ/mnt/ssdと/mnt/coreにマウントします。

```
# mkfs.ext4 -E lazy_itable_init /dev/sdb1
# mkfs.ext4 -E lazy_itable_init /dev/sdb2
# mkdir -p /mnt/ssd
# mkdir -p /mnt/core
```

/etc/fstabをviなどのエディタで開き、デバイス名またはUUIDで以下の行を追加します。

例：デバイス名

```
/dev/sdb1            /mnt/ssd              ext4    defaults        0 0
```

```
/dev/sdb2            /mnt/core            ext4    defaults       0 0
```

例：UUID

```
UUID=7f879fd4-13ed-4d66-9577-e88e3abc70f8 /mnt/ssd   ext4    defaults       0 0
UUID=f2b97605-5592-46b1-a73c-ec8fe3e473c8 /mnt/core  ext4    defaults       0 0
```

※lsblkコマンドなどで内蔵SSDデバイスのUUIDを確認してください。

作成した2つのファイルシステムをmount -a により、マウントします。

5.1.23 カーネルパラメータの設定

/etc/sysctl.confファイルの末尾に以下を追加します。

```
# ScaTeFS
vm.dirty_writeback_centisecs = 2
vm.dirty_expire_centisecs = 10
vm.swappiness = 0
net.core.somaxconn = 4000
net.ipv4.ip_local_reserved_ports = 50000-50009
kernel.core_pattern = /mnt/core/core.%e
kernel.core_uses_pid = 0
kernel.unknown_nmi_panic = 1
kernel.panic_on_unrecovered_nmi = 1
```

上記設定を反映させるため、rootで以下を実行します。

```
# sysctl -p
```

注：

ディストリビューションによっては、以下のエラーメッセージが出力される場合がありますが、特に問題ありません。

```
error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key
error: "net.bridge.bridge-nf-call-iptables" is an unknown key
error: "net.bridge.bridge-nf-call-arptables" is an unknown key
```

5.1.24 syslog のログローテート設定

以下のようにsyslogの設定を行います。

項目	設定値
保存期間	30回

項目	設定値
周期	weekly
圧縮	あり

/etc/logrotate.d/syslogファイルに、以下の★箇所を追加します。

```
/var/log/cron
/var/log/maillog
/var/log/messages
/var/log/secure
/var/log/spooler
{
    rotate 30   ★追加
    weekly      ★追加
    compress    ★追加

    sharedscripts
    postrotate
        /bin/kill -HUP `cat /var/run/syslogd.pid 2> /dev/null` 2> /dev/null
|| true
    endscript
}
```

5.1.25 updatedb.conf の設定

[RHEL6の場合]

以下のように/etc/updatedb.confファイルのPRUNEPATHSに/mnt/iotを追加します。

```
RUNE_BIND_MOUNTS = "yes"
PRUNEFS = "9p afs anon_inodefs auto autofs bdev binfmt_misc cgroup"
PRUNENAMES = ".git .hg .svn"
PRUNEPATHS = "/afs /media /net  /var/tmp /mnt/iot"
```

5.1.26 ScaTeFS の IO サーバとして組み込む(scatefs_addios)

ノードをIOサーバとして動作させるため、scatefs_addiosコマンドを実行します。

このために、すべてのIOサーバのIPアドレスを定義したファイルを用意します。設定項目は以下のとおりです。

表 5-11　scatefs_addios の設定項目

設定項目	意味	IB	10GbE
ipaddr	運用・管理ポートのIPアドレス	必須	必須
fipaddr	ファイルシステムポートのIPアドレス	必須	必須
inipaddr	IOサーバ間インタコネクト用ポートのIPアドレス	必須	必須
cport	クライアント接続ポート番号。 デフォルト値 50000から変更しない場合は省略可。	必須	必須
sport	サーバ間通信接続ポート番号。 デフォルト値50001から変更しない場合は省略可。	必須	必須
cdport	データ転送用クライアント接続ポート番号。 50002を指定する。	必須	必須
iftypes	fipaddr のインターフェースタイプ。 10GbEの場合は1、IPoIBの場合は2を指定する。 iftypesはfipaddrと同じ数だけスペース区切りで記載する。iftypes は省略可能であり、省略した場合のデフォルトは10GbEとなる。	必須	—
pciid@hcaport	IBで使用するHCAデバイスを示すIDとHCAデバイスのポート番号。 0000:83:00.0@1 のようにpciid@hcaportの書式で指定する。複数指定する場合はスペース区切りで記載する。 pciidの確認方法は6.1.7を参照。	必須	—

以下は定義ファイルの記述例です。

- ファイルシステムポートとして10GbEのみを使用する場合

```
-bash-4.1$ cat datafile1
# IOS#0の設定
ipaddr   10.0.0.1
fipaddr  10.0.1.1  10.0.1.2
inipaddr 10.2.0.10
cport 50000
sport 50001
cdport 50002

# IOS#1の設定
ipaddr   10.0.0.2
fipaddr  10.0.1.3  10.0.1.4
inipaddr 10.2.0.11
```

```
cport 50000
sport 50001
cdport 50002
```

- ファイルシステムポートとしてIBのみを使用する場合

```
-bash-4.1$ cat datafile1
# IOS#0の設定
ipaddr   10.0.0.1
fipaddr  10.0.2.1
inipaddr 10.2.0.10
cport 50000
sport 50001
cdport 50002
pciid@hcaport 0000:83:00.0@1
iftypes 2

# IOS#1の設定
ipaddr   10.0.0.2
fipaddr  10.0.2.2
inipaddr 10.2.0.11
cport 50000
sport 50001
cdport 50002
pciid@hcaport 0000:83:00.0@1
iftypes 2
```

- ファイルシステムポートとして10GbEとIBを併用する場合

  下記例では IOS#0 の設定中の 10.0.1.1 と 10.0.1.2 が 10GbE の IP アドレス、10.0.2.1 が IB の IP アドレスとなります。

```
-bash-4.1$ cat datafile1
# IOS#0の設定
ipaddr   10.0.0.1
fipaddr  10.0.1.1  10.0.1.2 10.0.2.1
inipaddr 10.2.0.10
cport 50000
sport 50001
cdport 50002
pciid@hcaport 0000:83:00.0@1
```

```
iftypes 1 1 2

# IOS#1の設定
ipaddr   10.0.0.2
fipaddr  10.0.1.3 10.0.1.4 10.0.2.2
inipaddr 10.2.0.11
cport 50000
sport 50001
cdport 50002
pciid@hcaport 0000:83:00.0@1
iftypes 1 1 2
```

このファイルを引数にscatefs_addiosコマンドを実行します。

```
# su - fsadmin
-bash-4.1$ scatefs_addios -f datafile1
```

scatefs_addiosコマンドは、一台のIOサーバで実行してください。すべてのIOサーバでコマンドを実行する必要はありません。一回の実行により、すべてのIOサーバに一度に設定されます。

設定したIOサーバはscatefs_detail -s により確認可能です。

```
# su - fsadmin
-bash-4.1$ scatefs_detail -s
--------------------------------------------------------------------------------
IOSID   MATE    IP[0]          IPCNT   FIP[0]         FIPCNT  IOTCNT  FSCNT
--------------------------------------------------------------------------------
   0     1    10.0.0.1           1   10.0.1.1           2      0     0
   1     0    10.0.0.2           1   10.0.1.3           2      0     0
--------------------------------------------------------------------------------
ALL:2  CAPACITY:256
```

5.2　IOターゲットの構築

以下の手順によりIOターゲットの作成を行います。

(4)　SPS デバイス名の確認

(5)　SPS パスチェック

(6)　パーティション作成(parted)

(7)　LVM デバイス作成(pvcreate, vgcreate, lvcreate)

(8)　IO ターゲット作成(scatefs_addiot)

5.2.1　SPS デバイス名の確認

5.1.2 で設計したLVM構成のストレージの論理ディスクのSPSデバイス名を確認します。

※SPSデバイス名はLVM(PV、VG)作成時に使用します。

ストレージの管理画面(iStorageManager)から以下の情報を確認します。

- IOサーバへ接続しているストレージ筐体のシリアル番号
- LUN

  ※5.1.6論理ディスク割り当て を実施したときにLUNが設定されます。

IOサーバ(iosv00, iosv01)の/etc/sps.confファイルからSPSデバイス名を確認します。

LVMデバイスを作成するIOサーバの/etc/sps.confファイルをlessやviewコマンドでファイルを開き、シリアル番号で検索します。

以下の例ではAがストレージ筐体のシリアル番号、BがLUNです。検索で見つかったdeviceがSPSデバイス名です。

[iosv00]

```
device:/dev/dda
       disk-info:NEC     ,DISK ARRAY      ,0000000942801512,00000
                                           ^^^^^^^^^^^^^^^^ ^^^^^
                                           A                B
       LoadBalance:D2
       path-info:auto Watch:Enable
A:ストレージの筐体シリアル番号
B:LUN
```

5.2.2　SPS パスチェック

5.1.2 で設計したLVMについて、全ポートへ負荷が分散されていることを確認します。SPSでは、Status=Activeとなっているパスが均等に使用されます。

ストライピングを構成する場合、spsadminコマンドでSPSパスを確認します。

以下spsadminコマンドの出力例を参考に、ストライピングを構成する4つの論理ディスクについて、全ポートに負荷が分散されることを確認してください。

ScsiAddressの4つの数字のうち先頭がI/Oサーバ側のポート番号です。

以下に標準モデル向けIOサーバv3 ScaTeFSデータ領域 lv_data01の例を記載します。

※以下の例ではLD番号とLUNが同じケースを記載します。

LVM構成

LV		Storage1	Storage2
iosv00	iosv01		
lv_data01	-	LD2,LD6	LD2,LD6

以下の出力例で、LVM設計と異なるLUN=2,3の4つの論理ディスクでストライピングを構成すると、IOサーバ側のポートは7と8の2ポートしか使用されません。LVM設計のとおりにLUN=2,6の4つの論理ディスクでストライピングを構成すると、IOサーバ側のポートは4ポートが均等に使用されるようになります。

```
# spsadmin --lun

+++ LogicalUnit 11:0:0:2 /dev/ddc [Normal] +++
  SerialNumber=0000000J1BN00180, LUN=0x00002
  LoadBalance=LeastSectors
  2: ScsiAddress=7:0:0:2, ScsiDevice=/dev/sde, Priority=1, Status=Active
  102:      ScsiAddress=9:0:0:2,      ScsiDevice=/dev/sdda,      Priority=2,
Status=Standby

+++ LogicalUnit 11:0:0:3 /dev/ddd [Normal] +++
  SerialNumber=0000000J1BN00180, LUN=0x00003
  LoadBalance=LeastSectors
  3: ScsiAddress=7:0:0:3, ScsiDevice=/dev/sdf, Priority=1, Status=Active
  103:      ScsiAddress=9:0:0:3,      ScsiDevice=/dev/sddb,      Priority=2,
Status=Standby

+++ LogicalUnit 11:0:0:6 /dev/ddg [Normal] +++
  SerialNumber=0000000J1BN00180, LUN=0x00006
  LoadBalance=LeastSectors
  106: ScsiAddress=9:0:0:6, ScsiDevice=/dev/sdde, Priority=1, Status=Active
  6: ScsiAddress=7:0:0:6, ScsiDevice=/dev/sdi, Priority=2, Status=Standby

+++ LogicalUnit 11:0:0:9 /dev/ddj [Normal] +++
  SerialNumber=0000000J1BN00179, LUN=0x00002
  LoadBalance=LeastSectors <Path thrashing suppressed>
  38: ScsiAddress=8:0:0:2, ScsiDevice=/dev/sdao, Priority=1, Status=Active
  152:      ScsiAddress=10:0:0:2,      ScsiDevice=/dev/sdey,      Priority=2,
Status=Standby

+++ LogicalUnit 11:0:0:10 /dev/ddk [Normal] +++
```

```
  SerialNumber=0000000J1BN00179, LUN=0x00003
  LoadBalance=LeastSectors <Path thrashing suppressed>
  39: ScsiAddress=8:0:0:3, ScsiDevice=/dev/sdap, Priority=1, Status=Active
  153:      ScsiAddress=10:0:0:3,      ScsiDevice=/dev/sdez,      Priority=2,
Status=Standby

+++ LogicalUnit 11:0:0:13 /dev/ddn [Normal] +++
  SerialNumber=0000000J1BN00179, LUN=0x00006
  LoadBalance=LeastSectors
  156:      ScsiAddress=10:0:0:6,      ScsiDevice=/dev/sdfc,      Priority=1,
Status=Active
  42: ScsiAddress=8:0:0:6, ScsiDevice=/dev/sdas, Priority=2, Status=Standby
```

5.2.3　パーティション作成

LVM設計内容からIOサーバで認識したデバイスに対してパーティションを作成します。

[ScaTeFSのメタデータ領域]

以下の2種類のパーティションを作成します。

(1) CLUSTERPRO のハートビート領域用のパーティション

論理ディスクの先頭に16MB程度のパーティションを作成します。

例

```
# parted /dev/dda
GNU Parted 2.1
Using /dev/dda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel gpt
(parted) mkpart primary ext4 0% 16MB
(parted) print
Model: NEC DISK ARRAY (scsi)
Disk /dev/dda: 1700GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Number  Start   End     Size    File system  Name     Flags
1      1049kB  15.7MB  14.7MB               primary
```

(2) メタデータ領域用のパーティション

1つの論理ディスクにパーティションを作成します。

パーティションの容量が均等になるように作成します。

※パーティション数は 5.1.2 で設計したLVM構成に合わせてください。

[ScaTeFSのデータ領域]

論理ディスク全体を1パーティションとするため、パーティション作成は不要です。

作成したパーティションを認識させるため、ストレージを接続しているIOサーバでOS再起動を行います。

5.2.4　LVM デバイス作成

LVM設計内容からLVMデバイスを作成します。

LVMデバイスの作成はSPSのデバイスファイル(/dev/ddX)を使用します。

※ストライピングするデバイスの組み合わせに注意してください。

※LVのストライピング数は 5.1.2 で設計したLVM構成に合わせてください。

LVMデバイスを作成した後にLVMデバイスを認識させるため、ストレージを接続しているIOサーバでOS再起動を行います。OSが再起動したあとに、作成したLVMデバイス(デバイスファイル)が存在することを確認します

以下にモデル毎にコマンド実行例を記載します。

【標準モデル向けIOサーバv1】

- ScaTeFSのデータ領域

例：/dev/ddb, /dev/ddj, /dev/ddp, /dev/ddxのデバイスでLVMデバイス作成

PV

```
# pvcreate /dev/ddb
# pvcreate /dev/ddj
# pvcreate /dev/ddp
# pvcreate /dev/ddx
```

VG

```
# vgcreate vg_data01 /dev/ddb /dev/ddj /dev/ddp /dev/ddx
```

LV

```
# lvcreate -i 4 -I 512 -l 100%free -r none -n lv_data01 vg_data01
```

-i オプションはストライピング数を指定します。

-I オプションはストライピングサイズ(上記は512KB)を指定します。

-l 100%free は空き容量すべてを割り当てることができます。

- ScaTeFSのメタデータ領域

  例：/dev/dda2, /dev/dde2のデバイスでLVMデバイス作成

  PV

```
# pvcreate /dev/dda2
# pvcreate /dev/dde2
```

  VG

```
# vgcreate vg_ctrl01 /dev/dda2 /dev/dde2
```

  LV

```
# lvcreate -i 2 -I 512 -l 100%free -r none -n lv_ctrl01 vg_ctrl01
```

【標準モデル向け IO サーバ v3 と v4】

- ScaTeFSのデータ領域

  例：/dev/ddc, /dev/ddj, /dev/ddg, /dev/ddnのデバイスでLVMデバイス作成

  PV

```
# pvcreate /dev/ddc
# pvcreate /dev/ddj
# pvcreate /dev/ddg
# pvcreate /dev/ddn
```

  VG

```
# vgcreate vg_data01 /dev/ddc /dev/ddj /dev/ddg /dev/ddn
```

  LV

```
# lvcreate -i 4 -I 512 -l 100%free -r none -n lv_data01 vg_data01
```

  例：/dev/ddcのデバイスでLVMデバイス作成

  PV

```
# pvcreate /dev/ddc
```

  VG

```
# vgcreate vg_data01 /dev/ddc
```

LV

```
# lvcreate -l 100%free -r none -n lv_data01 vg_data01
```

※LVをストライピングなしで作成します。

- ScaTeFSのメタデータ領域

  例：/dev/dda2のデバイスでLVMデバイス作成

  PV

```
# pvcreate /dev/dda2
```

  VG

```
# vgcreate vg_ctrl01 /dev/dda2
```

  LV

```
# lvcreate -l 100%free -r none -n lv_ctrl01 vg_ctrl01
```

  ※LVをストライピングなしで作成します。

5.2.5　IO ターゲット作成(scatefs_addiot)

作成したLVM論理ボリューム(LV)をIOターゲットとしてシステムに組み込むため、scatefs_addiotコマンドを実行します。

このために、IOターゲットを定義したファイルを用意します。

作成したLVをIOターゲットとしてどのIOサーバに組み込むのか設計します。

以下に標準モデル向けIOサーバv1のデータ領域のHDD(1TB)の場合の設計例を記載します。

[ScaTeFSのデータ領域]

作成したLVの前半3つをiosv00、後半3つをiosv01へ割り当てます。

iosv00

lv_data01, lv_data02, lv_data03

iosv01

lv_data04, lv_data05, lv_data06

[ScaTeFSのメタデータ領域]

作成したLVの前半3つをiosv00、後半3つをiosv01へ割り当てます。

iosv00

lv_ctrl01, lv_ctrl02, lv_ctrl03

iosv01

lv_ctrl04, lv_ctrl05, lv_ctrl06

```
-bash-4.1$ cat datafile2
iosid 0
data   /dev/vg_data01/lv_data01
ctrl   /dev/vg_ctrl01/lv_ctrl01
data   /dev/vg_data02/lv_data02
ctrl   /dev/vg_ctrl02/lv_ctrl02
data   /dev/vg_data03/lv_data03
ctrl   /dev/vg_ctrl03/lv_ctrl03
iosid 1
data   /dev/vg_data04/lv_data04
ctrl   /dev/vg_ctrl04/lv_ctrl04
data   /dev/vg_data05/lv_data05
ctrl   /dev/vg_ctrl05/lv_ctrl05
data   /dev/vg_data06/lv_data06
ctrl   /dev/vg_ctrl06/lv_ctrl06
```

ファイル中に記載する項目の意味は以下のとおりです。

設定項目	意味
iosid	IOサーバのサーバID(SID)。 scatefs_detail -s で確認する
data	IOターゲットのデータ領域のデバイス名
ctrl	IOターゲットのメタデータ領域のデバイス名

このファイルを引数にscatefs_addiotコマンドを実行します。

```
# su - fsadmin
-bash-4.1$ scatefs_addiot -f datafile2
```

scatefs_addiotコマンドもscatefs_addiosと同様、一台のIOサーバで実行してください。す

べてのIOサーバでコマンドを実行する必要はありません。一回の実行により、すべてのIOサーバで設定が反映されます。

IOターゲットの情報はscatefs_detail -t により確認可能です。

```
# su - fsadmin
-bash-4.1$ scatefs_detail -t
----------------------------------------
IOTID   IOS     FS:SG
----------------------------------------
   0    0     none:none
   1    0     none:none
   2    0     none:none
   3    1     none:none
   4    1     none:none
   5    1     none:none
----------------------------------------
ALL:6  USED:0  CAPACITY:16384
```

この時点では、まだIOターゲット(すなわちデータ用とメタデータ用のLV)にローカルファイルシステムが作成されていません。

5.3　mkfsの準備と実行

ScaTeFSのmkfsには以下の手順を踏みます。

- ScaTeFS作成(scatefs_mkfs)

5.3.1　ScaTeFS 作成(scatefs_mkfs)

ここまでに作成したIOターゲットを元に、scatefs_mkfsによりScaTeFSのファイルシステムを作成します。

このために、作成するファイルシステムを定義したファイルを用意します。

```
-bash-4.1$ cat datafile3
name        scatefs00
iotid       0 1 2 3 4 5
```

ファイル中に記載する項目の意味は以下のとおりです。

設定項目	意味
name	ファイルシステム名 クライアントノードからマウントする際などに指定します 31文字以内
iotid	IOターゲットのID 5.1.2 で設計した設定値を指定します。 scatefs_detail -t で確認できます。
data_fstype	データ領域のファイルシステムタイプ 5.1.1で設計したファイルシステムタイプを指定します。 data_fstype は省略可能であり、省略した場合のデフォルトはext4です。

この例では、6つのIOターゲットからなるファイルシステム「scatefs00」を作成します。

このファイルを引数にscatefs_mkfsコマンドを実行します。

```
# su - fsadmin
-bash-4.1$ scatefs_mkfs -f datafile3
```

scatefs_mkfsコマンドも、一台のIOサーバで実行してください。すべてのIOサーバでコマンドを実行する必要はありません。一回の実行により、すべてのIOサーバで設定が反映されます。

scatefs_mkfsにより、各IOサーバでIOターゲットのmkfsが行われ、ローカルにマウントされます。さらにScaTeFSとしてのフォーマットが行われます。

ファイルシステムの情報はscatefs_detail -f により確認可能です。

```
# su - fsadmin
-bash-4.1$ scatefs_detail -f
--------------------------------------------------------------------------
FSID    NAME                  ROOTIOS IOSCNT  IOTCNT  SGCNT   VERSION
--------------------------------------------------------------------------
  0    scatefs00                  0     2      6     1   0x00010000
--------------------------------------------------------------------------
ALL:1  CAPACITY:32
```

5.3.2　IO サーバ設定ファイル

5.3.2.1　scatefssrv.conf

/etc/scatefsに配置するscatefssrv.confはIOサーバデーモンのチューニングパラメータの設定ファイルです。

ファイルがない場合は推奨パラメータ値で動作するため、通常は本ファイルを配置する必要はありません。

scatefssrv.confは決められたタグ[network](ネットワーク関連のチューニングパラメータ用) 、[journal](journal関連のチューニングパラメータ用)、[quota](quota関連のチューニングパラメータ用)、と [iotarget](iotarget関連のチューニングパラメータ用)のいずれかを記載してから設定値を指定する必要があります。また、以下の条件に該当する場合、対象の設定値はデフォルト値で動作します。

- /etc/scatefs/scatefssrv.confがない
- タグ名が記載されていない
- 設定値を指定していない
- 最小値、最大値の範囲外を指定

scatefssrv.conf の設定を変更した場合、scatefs_admin コマンドを使用してscatefssrv.confを各IOサーバへ転送してから各IOサーバのIOサーバデーモンを再起動してください。IOサーバデーモンの再起動とscatefs_adminについては8.9を参照してください。

下記表にパラメーター一覧を示します。通常はIBSSYNCMODE以外を設定する必要はありません。

表 5-12　network の設定値一覧

設定値	説明	最小値	最大値	デフォルト	備考
RECVTHREADNUM	クライアント用受信スレッド数	1	200	標準モデル：50 小規模モデル：32	
RECVTHREADCNNNUM	クライアント用受信スレッド1つあたりの監視ソケット数	10	512	256	
CLIWORKERTHREADNUM	クライアント用workerスレッド数	1	なし	標準モデル：64 小規模モデル：32	
SRVWORKERTHREADNUM	サーバ用workerスレッド数	10	なし	標準モデル：192 小規模モデル：96	

設定値	説明	最小値	最大値	デフォルト	備考
JNLWORKERTHREADNUM	ジャーナル用workerスレッド数	1	なし	10	
FAIRPOLICY	フェアシェアポリシー	0	2	0	0:OFF 1:UIDモード 2:ClientID モード
IBSOPTIMMWAITON	要求待ち合わせを最適化することで性能を向上させるモードのON/OFF 1:ON 0:OFF	0	1	0	
IBSIOMEMNODE	IBによるデータ転送で使用するメモリを確保するNUMAシステムのノード番号を指定する。詳細は9.12.4を参照。	0	1	1	

表 5-13　journal の設定値一覧

設定値	説明	最小値	最大値	デフォルト	備考
JMODE	journal のモード	0	3	3	0:OFF 1:メモリ 2:共有ディスク 3:メモリ+SSD
MEMSIZE	ログ領域のメモリサイズ(MB)	1	64	32	IOターゲット毎にログ領域を作成する
DDLENT	dirtyデータリストのエントリ数	1000	500000	100000	
DDLINTVAL	dirtyデータ監視周期(sec)	1	600	1	
DDLSAVING	dirtyデータ保持期間(sec)	1	600	1	

表 5-14　QUOTA 機能の設定値一覧

設定値	説明	最小値	最大値	デフォルト	備考
QUOTAMODE	QUOTA機能のモード	0	1	1	0:OFF 1:ON

表 5-15　iotarget の設定値一覧

設定値	説明	最小値	最大値	デフォルト	備考
READAHEADSIZE	先読みサイズ	1048576	2147483647	8388608	
CACHE	下記キャッシュの最大エントリ数 ・ディレクトリネーム ・iノード	1	なし	標準モデル：5242880 小規模モデル：3495253	
DECACHE	ディレクトリエントリキャッシュの最大エントリ数	1	なし	標準モデル：256000 小規模モデル：170666	
IBSSYNCMODE	write 時のディスク同期の ON/OFF。各モードの詳細については9.12.3を参照してください。 1: write(2)時ディスク同期モード 0: close(2)時ディスク同期モード	0	1	0	

5.4　CLUSTERPROの設定

これまで構築しました各リソースの管理をCLUSTERPROで行うため、CLUSTERPROの設定を行います。

CLUSTERPROの設定はWebManagerを使用するため、IOサーバへネットワーク接続できる作業用PCが必要となります。

※WebManagerの操作方法と各リソースの設定については、マニュアル「CLUSTERPRO X for Linux リファレンスガイド」を参照してください。

5.4.1　準備

5.4.1.1　クラスタ構成情報ファイルを作業用 PC へ転送

5.1.3 CLUSTERPROのクラスタ構成情報作成で作成したクラスタ構成情報ファイルを作業

用PCへ転送します。

5.4.1.2　IO サーバ間インタコネクト用ポートのネットワーク設定確認

CLUSTERPROの設定では、IOサーバ間インタコネクト用ポートを使用します。
未設定の場合はネットワーク設定を実施してください。

5.4.2　Cluster WebUI または WebManager 起動

【標準モデル向けIOサーバv4+以降（Cluster WebUI)】
Web ブラウザを起動します。
【標準モデル向けIOサーバv1,v3,v4 (WebManager)】
「管理者として実行」でWeb ブラウザを起動します。

ブラウザのアドレスバーに、IOサーバのIPアドレス(管理用)とポート番号を入力します。
※もし接続できない場合、ペアのIOサーバのIPアドレスを指定します。
http://10.0.0.1:29003

5.4.3　クラスタ構成情報ファイルのインポート

【標準モデル向けIOサーバv4+以降（Cluster WebUI)】
Cluster WebUIを起動します。ツールバーのドロップダウンメニューで[設定モード] を選択します。「設定のインポート」をクリックして、クラスタ構成情報ファイルをインポートします。
【標準モデル向けIOサーバv1,v3,v4 (WebManager)】
WebManagerが起動すると、確認画面が表示されます。
「クラスタ構成情報をインポートする」をクリックして、クラスタ構成情報ファイルをインポートします。

5.4.4　クラスタプロパティ

- インタコネクト
  ツリービューのclusterを右クリックしてプロパティを開き、インタコネクトタブをクリックします。
  以下のハートビートI/Fの設定を変更します。
  優先度2：種別(DISK)
  CLUSTERPROのハートビート領域用のパーティションのデバイス名へ変更します。

※ハートビート領域用のパーティションのデバイス名の確認方法を記載します。
ディスクハートビート用デバイス名はIOサーバから同じストレージの論理ディスク(LUN)を指定する必要があります。

ディスクハートビート用に使用するストレージのシリアル番号を0000000942801512とします。

IOサーバ(iosv00, iosv01)の/etc/sps.confファイルをlessやviewコマンドでファイルを開き、シリアル番号で検索します。

[iosv00]

```
device:/dev/dda
       disk-info:NEC     ,DISK ARRAY      ,0000000942801512,00000
                                          ^^^^^^^^^^^^^^^^ ^^^^^
                                          A                B
       LoadBalance:D2
       path-info:auto Watch:Enable
A:ストレージの筐体シリアル番号
B:LUN
```

[iosv01]

```
device:/dev/ddn
       disk-info:NEC     ,DISK ARRAY      ,0000000942801512,00000
       LoadBalance:D2
       path-info:auto Watch:Enable
```

検索で見つかったdeviceがディスクハートビート用デバイスになります。

そのデバイスにはハートビート用パーティションを作成していますので、CLUSTERPROのディスクハートビート用デバイスの設定を変更します。

5.4.5　設定反映

【標準モデル向けIOサーバv4+以降（Cluster WebUI)】

Cluster WebUIの「設定の反映」をクリックして、設定反映を行います。

【標準モデル向けIOサーバv1,v3,v4 (WebManager)】

WebManagerのメニューバーの「ファイル」の「設定の反映」をクリックして、設定反映を行います。

設定反映後にクラスタ構成のIOサーバでOS再起動を行います。OS起動後にclpstatコマンドを実行して、クラスタの状態を確認します。

5.5　IOサーバの構築（DDNストレージ編）

SFA7990XE コントローラの２つのVM上にIOサーバを構築します。

以下の説明では 2台のIOサーバをiosv00(VM1),iosv01(VM2)と記載します。

以下に動作確認済みバージョンを記載します。

【DDN製 SFA7990XE】

表 5-16　SFA7990XE 動作確認済みバージョン

OS	kernel	MLNX_OFED	CLUSTERPRO
CentOS7.7	3.10.0-1062.el7.x86_64	4.7-1.0.0.1	4.2.0-1

5.5.1　IO ターゲットの設計

IOターゲットは、ScaTeFSのファイルシステムの基盤となるデータストアです。クライアントノードから書き込まれたファイルデータは、IOサーバに分散され、さらにIOサーバ内のIOターゲットに分散して格納されます。

IOターゲットはファイルのデータ自体を格納するデータ領域と、ファイルタイプや更新時刻などを格納するメタデータ領域に分けられます。複数のIOターゲットを作成できますが、必ずデータ領域とメタデータ領域の個数は同数で一対一の関係です。

以下はIOサーバ(VM)2台のIOターゲットの構成例です。

表 5-17　IO ターゲットの構成例　SFA7990XE　データ領域

データ領域							
ディスク			プール			推奨ファイルシステムタイプ	IO ターゲット数
タイプ	容量	数	RAID	数	VD		
NLSAS	12TB	168	6(8+PQ)	4	4	xfs	4

※プールは 42 個のディスクで構成します。

表 5-18　IO ターゲットの構成例　SFA7990XE　メタデータ領域

メタデータ領域							
ディスク			プール			推奨ファイルシステムタイプ	IO ターゲット数
タイプ	容量	数	RAID	数	VD		
SSD	1.92TB	6	6(4+PQ)	1	4	ext4	データ領域のIOターゲット数と同じ

5.5.2　LVM の設計

DDN製 SFA7990XEの構成(プール、VD(仮想ディスク))からScaTeFSのメタデータ領域、

データ領域のパーティション数、ストライピング数、IOターゲットを使用する順序を設計します。

以下にモデル毎にLVMの設計例を記載します。

【DDN製 SFA7990XE】

以下にデータ領域のHDD(12TB)の場合の設計例を記載します。

- ScaTeFSのデータ領域

  ストライピングなしでLVを作成します。

LVM構成

Pool	Controller 1	Controller 2	iosv00			iosv01		
			LV	IOT	順序	LV	IOT	順序
pool-0	-	VD0	-	-	-	lv_data03	2	1
pool-1	VD1	-	lv_data01	0	1	-	-	-
pool-2	VD2	-	lv_data02	1	2	-	-	-
pool-3	-	VD3	-	-	-	lv_data04	3	2

上記LVM構成のIOターゲットを使用する順序を記載します。

IOサーバ	IOターゲットを使用する順序
vm1	0 1
vm2	2 3

5.5.22 ScaTeFS作成の設定項目iotidの値を記載します。

iosv00(VM1)、iosv01(VM2)のIOターゲットを使用する順序を続けて記載します。

項目	設定値
iotid	0 1 2 3

- ScaTeFSのメタデータ領域

  ストライピングなしでLVを作成します。

LVM構成

Pool	Controller 1	Controller 2	iosv00		iosv01	
			LV	IOT	LV	IOT
pool-4	-	VD20p2	-	-	lv_ctrl03	2

Pool	Controller 1	Controller 2	iosv00		iosv01	
			LV	IOT	LV	IOT
	VD21p2	-	lv_ctrl01	0	-	-
	VD22p2	-	lv_ctrl02	1	-	-
	-	VD23p2	-	-	lv_ctrl04	3

※VD-XのXはパーティションを指します。

5.5.3　時刻の設定

SFA7990XE コントローラの 2 つの VM 上で以下の手順で設定します。

(1)　タイムゾーンの設定

以下は Asia/Tokyo に設定する場合の例です。

```
# timedatectl set-timezone Asia/Tokyo
# timedatectl
      Local time: Fri 2020-06-19 14:51:51 JST
  Universal time: Fri 2020-06-19 05:51:51 UTC
        RTC time: Fri 2020-06-19 05:51:50
       Time zone: Asia/Tokyo (JST, +0900)
     NTP enabled: no
NTP synchronized: no
 RTC in local TZ: no
      DST active: n/a
```

(2)　時刻同期の設定

b)　/etc/chrony.conf の編集

下記行を追記します。<server-ip-addr>は chrony サーバの IP アドレスです。

```
server <server-ip-addr> iburst
```

c)　chronyd の開始

```
# systemctl enable chronyd
# systemctl start chronyd
```

5.5.4　multipath の設定

SFA7990XE コントローラの 2 つの VM 上で以下の手順で設定します。

(1) device-mapper-multipath パッケージのインストール

```
# yum install device-mapper-multipath
```

※) 既にインストール済の場合は不要です。

(2) /etc/multipath.conf の作成

```
# mpathconf –enable
# ls -l /etc/multipath.conf
-rw------- 1 root root 2415 Jun 19 02:40 /etc/multipath.conf
```

(3)　multipathd サービスの開始

```
# systemctl start multipathd
```

(4)　/etc/multipath/wwids へ WWID を追加

```
# cd /dev/disk/by-id
# ls -1 scsi-* | sed -e "s/scsi-/¥//g" ¥-e "s/$/¥//g" >> /etc/multipath/wwids
# cat /etc/multipath/wwids

# Multipath wwids, Version : 1.0
# NOTE: This file is automatically maintained by multipath and multipathd.
# You should not need to edit this file in normal circumstances.
#
# Valid WWIDs:
/360001ff0d004e000000002a589200000/
/360001ff0d004e000000002a689210001/
/360001ff0d004e000000002a789220002/
/360001ff0d004e000000002a889230003/
/360001ff0d004e000000002a989240014/
/360001ff0d004e000000002aa89250015/
/360001ff0d004e000000002ab89260016/
/360001ff0d004e000000002ac89270017/
```

(5)　alias の設定

/etc/multipath.conf へ alias を設定します。

以下に alias 名「7990xe_lun00」の設定例を記載します。

```
multipaths {
      multipath {
            wwid                  360001ff0d004e000000002a589200000
            alias                 7990xe_lun00
      }
}
```

(6)　サービス再起動後のデバイス確認

multipathd サービスを再起動後、mutlipath デバイスが表示されることを確認します。

```
# systemctl restart multipathd
# ls -l /dev/mapper/7990xe_lun*
lrwxrwxrwx 1 root root 7 Jun 19 05:18 /dev/mapper/7990xe_lun00 -> ../dm-9
lrwxrwxrwx 1 root root 7 Jun 19 05:18 /dev/mapper/7990xe_lun01 -> ../dm-6
lrwxrwxrwx 1 root root 7 Jun 19 05:18 /dev/mapper/7990xe_lun02 -> ../dm-8
lrwxrwxrwx 1 root root 7 Jun 19 05:18 /dev/mapper/7990xe_lun03 -> ../dm-7
lrwxrwxrwx 1 root root 7 Jun 19 05:18 /dev/mapper/7990xe_lun20 -> ../dm-2
lrwxrwxrwx 1 root root 7 Jun 19 05:18 /dev/mapper/7990xe_lun21 -> ../dm-5
lrwxrwxrwx 1 root root 7 Jun 19 05:18 /dev/mapper/7990xe_lun22 -> ../dm-3
lrwxrwxrwx 1 root root 7 Jun 19 05:18 /dev/mapper/7990xe_lun23 -> ../dm-4
```

5.5.5　CLUSTERPRO X for Linux のインストール

5.1.8 を参照して 2 つの VM 上でインストールしてください。

5.5.6　IB ドライバのインストール

既にインストール済の場合は不要です。

インストールされていない場合 5.1.10 を参照して 2 つの VM 上でインストールしてください。

5.5.7　rsh 関連のインストール

5.1.11 を参照して 2 つの VM 上でインストールしてください。

5.5.8　ScaTeFS パッケージのインストール

5.1.12 を参照して 2 つの VM 上でインストールしてください。

5.5.9　ScaTeFS のライセンス登録

5.1.13 を参照して 2 つの VM 上で設定してください。

5.5.10 SELinux 無効化

SELinux が無効化されていない場合は、5.1.14 を参照して 2 つの VM 上で SELinux を無効化してください。

5.5.11 ファイアウォール無効化

ファイアウォールが無効化されていない場合は、5.1.15 を参照して 2 つの VM 上でファイアウォールを無効化してください。

5.5.12 abrtd 無効化

abrtd が無効化されていない場合は、5.1.17 を参照して 2 つの VM 上で abrtd を無効化してください。

5.5.13 ファイルシステムポート（IB）のネットワークインターフェースの設定

以下の手順で 2 つの VM 上で、IB ネットワークのボンディングの設定を行ってください。

(1) ifcfg-ib0、ifcfg-ib1 の更新

以下のように MASTER, SLAVE の設定を追加してください。

/etc/sysconfig/network-scripts/ifcfg-1b0(変更)

```
CONNECTED_MODE=no
TYPE=InfiniBand
NAME=ib0
UUID=4ccd7d05-9b43-4cfc-8467-14827396a027
DEVICE=ib0
ONBOOT=yes
MASTER=ibbond0
SLAVE=yes
```

/etc/sysconfig/network-scripts/ifcfg-1b1(変更)

```
CONNECTED_MODE=no
TYPE=InfiniBand
NAME=ib1
UUID=cfbc6db7-1464-471c-bff0-2f92a432b5e8
DEVICE=ib1
ONBOOT=yes
MASTER=ibbond0
SLAVE=yes
```

(2) ボンディング設定ファイル(ifcfg-ibbond0)

以下の内容で作成してください。UUID には uuidgen コマンドで生成した UUID を設定します。

/etc/sysconfig/network-scripts/ ifcfg-ibbond0 (新規作成)

```
BONDING_OPTS="miimon=100 mode=active-backup primary=ib0"
TYPE=Bond
BONDING_MASTER=yes
PROXY_METHOD=none
BROWSER_ONLY=no
DEFROUTE=no
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ibbond0
UUID=5469860b-0057-4bcc-82e4-e0f5cd467d7c
DEVICE=ibbond0
ONBOOT=yes
MTU=2044
```

(3) ネットワークサービス再起動

ネットワークサービスを再起動して、ibbond0 のインターフェースの state が UP になることを確認してください。

```
# systemctl restart network
# ip a show dev ibbond0
7: ibbond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 2044 qdisc noqueue state UP
group default qlen 1000
    link/infiniband  20:00:11:07:fe:80:00:00:00:00:00:00:b8:59:9f:03:00:f6:89:b0  brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:ff:ff:ff:ff
    inet6 fe80::ba59:9f03:f6:89b0/64 scope link
       valid_lft forever preferred_lft forever
```

5.5.14 IPv6 無効化

5.1.19 を参照して 2 つの VM 上で IPv6 を無効化してください。SFA7990XE の場合は、設定ファイルは/boot/efi/EFI/redhat/grub.cfg ではなく、/boot/grub2/grub.cfg となりますので、読み替えて作業を実施してください。

5.5.15 ファイルシステム管理アカウント（fsadmin）の設定

5.1.21 を参照して 2 つの VM 上で設定してください。

5.5.16 カーネルパラメータの設定

下記の手順で 2 つの VM 上で設定してください。

(1) /etc/sysctl.conf の設定

以下を追記してください。

kernel.core_pattern は、/mnt/iot/<IOTID>/data/core.%e の形式で設定します。<IOTID>には VM に mount されている一番若い番号を指定してください。以下は 0 番の IOT を指定する場合の例です。

```
vm.dirty_writeback_centisecs = 2
vm.dirty_expire_centisecs = 10
vm.swappiness = 0
net.core.somaxconn = 4000
net.ipv4.ip_local_reserved_ports = 50000-50009
kernel.core_pattern = /mnt/iot/0/data/core.%e
kernel.core_uses_pid = 0
kernel.unknown_nmi_panic = 1
kernel.panic_on_unrecovered_nmi = 1
kernel.panic = 10
vm.min_free_kbytes = 4194304
```

(2) カーネルパラメータの設定

```
# sysctl -p
```

5.5.17 syslog のログローテート設定

5.1.24 を参照して 2 つの VM 上で設定してください。

5.5.18 ScaTeFS の IO サーバとして組み込む（scatefs_addios）

5.1.26 を参照して設定してください。

5.5.19 パーティション作成

以下のようにメタデータ用デバイスに対して、CLUSTERPRO のハートビート領域用のパーティション(16MB)とメタデータ領域用パーティション(残りすべて)を作成します。以下は４つのメタデータ用デバイスすべてに対して実施してください。

```
# parted /dev/mapper/7990xe_lun20
GNU Parted 3.1
Using /dev/mapper/7990xe_lun20
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel gpt
(parted) unit MB
(parted) mkpart primary ext4 0% 16MB
```

```
(parted) mkpart primary ext4 16MB  100%
(parted) print
Model: Linux device-mapper (multipath) (dm)
Disk /dev/mapper/7990xe_lun20: 927713MB
Sector size (logical/physical): 4096B/4096B
Partition Table: gpt
Disk Flags:

Number  Start   End      Size     File system  Name    Flags
 1      1.05MB  15.7MB   14.7MB               primary
 2      15.7MB  927712MB  927696MB             primary
```

5.5.20 LVM デバイス作成

LVM 設計内容から LVM デバイスを作成します。LVM デバイスの作成は multipath のデバイスファイル(/dev/mapper/7990xe_lunXX)を使用します。

LVM デバイスを作成した後に LVM デバイスを認識させるため、IO サーバ(VM)で OS 再起動を行います。OS が再起動したあとに、作成した LVM デバイス(デバイスファイル)が存在することを確認します。

以下にモデル毎にコマンド実行例を記載します。

【DDN 製 SFA7990XE】

- ScaTeFSのデータ領域

PV

```
# pvcreate /dev/mapper/7990xe_lun00
# pvcreate /dev/mapper/7990xe_lun01
# pvcreate /dev/mapper/7990xe_lun02
# pvcreate /dev/mapper/7990xe_lun03
```

VG

```
# vgcreate vg_data01 /dev/mapper/7990xe_lun01
# vgcreate vg_data02 /dev/mapper/7990xe_lun02
# vgcreate vg_data03 /dev/mapper/7990xe_lun00
# vgcreate vg_data04 /dev/mapper/7990xe_lun03
```

LV

```
# lvcreate -l 100%free -r none -n lv_data01 vg_data01
```

```
# lvcreate -l 100%free -r none -n lv_data02 vg_data02
# lvcreate -l 100%free -r none -n lv_data03 vg_data03
# lvcreate -l 100%free -r none -n lv_data04 vg_data04
```

- ScaTeFSのメタデータ領域

PV

```
# pvcreate /dev/mapper/7990xe_lun20p2
# pvcreate /dev/mapper/7990xe_lun21p2
# pvcreate /dev/mapper/7990xe_lun22p2
# pvcreate /dev/mapper/7990xe_lun23p2
```

VG

```
# vgcreate vg_ctrl01 /dev/mapper/7990xe_lun21p2
# vgcreate vg_ctrl02 /dev/mapper/7990xe_lun22p2
# vgcreate vg_ctrl03 /dev/mapper/7990xe_lun20p2
# vgcreate vg_ctrl04 /dev/mapper/7990xe_lun23p2
```

LV

```
# lvcreate -l 100%free -r none -n lv_ctrl01 vg_ctrl01
# lvcreate -l 100%free -r none -n lv_ctrl02 vg_ctrl02
# lvcreate -l 100%free -r none -n lv_ctrl03 vg_ctrl03
# lvcreate -l 100%free -r none -n lv_ctrl04 vg_ctrl04
```

5.5.21 IO ターゲット作成（scatefs_addiot）

5.2.5 を参照して設定してください。

5.5.22 ScaTeFS 作成（scatefs_mkfs）

5.3.1 を参照して設定してください。

5.5.23 CLUSTERPRO の設定

5.4 を参照して設定してください。

第6章　Linux クライアントの設定

6.1　IB利用時の設定

6.1.1　IB ドライバのインストール

SX-Aurora TSUBASAとSX-Aurora TSUBASA以外のLinuxマシンでは、インストールできるIBドライバに違いがあります。

- SX-Aurora TSUBASAの場合
- NVIDIA社が提供するMLNX_OFED をインストールします。
- SX-Aurora TSUBASA以外のLinuxマシンの場合

  OSディストリビューション付属のIBドライバと、MLNX_OFEDのいずれかを利用できます。どちらを選んでもScaTeFSの機能に違いはありません。IBを使うユーザアプリケーションの条件などに合わせて、使用するドライバを選択してください。

以下に、MLNX_OFEDのインストール方法を記載します。

※OSディストリビューション付属のIBドライバをインストールする方法については、Red Hat Enterprise Linux Serverのマニュアルを参照してください。

(1) MLNX_OFED パッケージの入手

ScaTeFS/ClientがサポートするMLNX_OFEDのバージョンは以下のとおりです。

OS	MLNX_OFEDのバージョン
RHEL/CentOS 7.3	3.4-2.1.9.0.1
RHEL/CentOS 7.4	4.2-1.2.0.0
RHEL/CentOS 7.5	4.3-3.0.2.1
RHEL/CentOS 7.6	4.6-4.1.2.0
	4.7-1.0.0.1
RHEL/CentOS 7.7	4.7-1.0.0.1
RHEL/CentOS 7.8	4.9-0.1.7.0
RHEL/CentOS 7.9	4.9-2.2.4.0
RHEL/CentOS 8.1	4.7-3.2.9.0

OS	MLNX_OFEDのバージョン
RHEL/CentOS 8.2	4.9-0.1.7.0
RHEL/CentOS 8.3	4.9-3.1.5.0
RHEL/CentOS 8.4	4.9-4.0.8.0
	5.5-1.0.3.2
RHEL/Rocky Linux 8.5	5.5-1.0.3.2 ※RHEL 8.5でカーネルバージョンが4.18.0-348.12.2.el8_5.x86_64の場合、本バージョンを使用してください。Rocky Linux 8.5の場合や、カーネルバージョンがより新しい場合は、下記の5.6-1.0.3.3を使用してください。
	5.6-1.0.3.3
RHEL/Rocky Linux 8.6	5.6-2.0.9.0 ※カーネルバージョンが4.18.0-372.26.1.el8_6.x86_64の場合、本バージョンを使用してください。カーネルバージョンがより新しい場合は、下記の5.8-1.1.2.1を使用してください。
	5.8-1.1.2.1
RHEL/Rocky Linux 8.8	23.04-1.1.3.0
RHEL/Rocky Linux 8.10	23.10-3.2.2.0

※ScaTeFS/ClientがサポートするMLNX_OFEDのバージョンは、SX-Aurora TSUBASA用InfiniBandがサポートするMLNX_OFEDのバージョンと同一になります。

該当するバージョンのMLNX_OFED を、NVIDIA社の公式サイトから入手します。
https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/

※MLNX_OFED 4.6-4.1.2.0 は上記URLでは公開されていません。以下のURLからダウンロードしてください。

https://mellanox.my.salesforce.com/sfc/p/#500000007heg/a/1T000000cCrw/jAKX3brAtwtWng6sVqHpSXf2pT8UrSUL2rMKpn3c4ng
パスワード: mgIdJQfI

MLNX_OFEDが入手できない場合は、サポート部門までお問い合わせください。

(2) 下記のパッケージがインストールされていない場合には、OS ディストリビューションからインストールしてください。

lsof gtk2 atk cairo tcl tcsh tk pciutils

```
# yum install lsof gtk2 atk cairo tcl tcsh tk pciutils
```

(3) ダウンロードした ISO ファイルを任意のマウントポイントにマウントします。以下は /mnt/iso ディレクトリにマウントする場合の実行例です。

```
# mount -t iso9660 -o loop MLNX_OFED_LINUX-xxxxx-x86_64.iso /mnt/iso
```

(4) インストールスクリプトを実行します。

```
# /mnt/iso/mlnxofedinstall
```

⚠ 注意事項

カーネルのアップデートを行っている場合、mlnxofedinstallによるインストールが失敗することがあります。この場合、以下のように --add-kernel-support と --kmp オプションを付けて実行してください。

```
# /mnt/iso/mlnxofedinstall --add-kernel-support --kmp
```

(5) IB 関連のパッケージを削除してインストールを継続するか確認されるので、"y"を入力します。

```
        This program will install the MLNX_OFED_LINUX package on your machine.
Note that all other Mellanox, OEM, OFED, or Distribution IB packages will be removed.
Do you want to continue?[y/N]:y
```

(6) インストールが完了するまで待ちます。完了したら (7) に進んでください。

(7) RHEL/CentOS 7.4 もしくは 7.3 のクライアントマシンにおいて、MLNX_OFED と 10GbE

を同時に使用する場合、以下の設定を行ってください。該当しない場合は、(8) に進んでください。

下記のように /etc/depmod.d/*-mlnx-ofa_kernel.conf の cxgb4 の行頭に"#" を追加して、設定をコメントアウトしてください。

```
# vi /etc/depmod.d/zz01-mlnx-ofa_kernel.conf
  --------------------------
  #override iw_cxgb4 * weak-updates/mlnx-ofa_kernel/drivers/infiniband/hw/cxgb4
  --------------------------
```

モジュールの依存関係を解決します。

```
# depmod -a
```

(8)　再起動を行います。

```
# reboot
```

(9)　再起動後、HCA の情報が参照できることを確認します。

```
# ibstat
CA 'mlx5_0'
       CA type: xxxxxx
       Number of ports: 1
       Firmware version: XXXXXXXXXX
       Hardware version: 0
       Node GUID: 0xXXXXXXXXXXXXXXXX
       System image GUID: 0xXXXXXXXXXXXXXXXX
       Port 1:
              State: Active
              Physical state: LinkUp
              Rate: XXX
              Base lid: XXX
              LMC: X
              SM lid: X
              Capability mask: 0xXXXXXXXX
              Port GUID: 0xXXXXXXXXXXXXXXXX
              Link layer: InfiniBand
```

インストールしたMLNX_OFEDのバージョンは、以下のコマンドで確認できます。

```
# /usr/bin/ofed_info -s
MLNX_OFED_LINUX-4.9-0.1.7.0:
```

以上で、MLNX_OFED のインストールは完了です。

> ⚠ IBドライバに関する注意事項
>
> - IBドライバのインストールは、ScaTeFS/Clientのインストールよりも先に行う必要があります。
> - MLNX_OFEDのアンインストールや再インストールを行う場合、事前にScaTeFS/Clientのアンインストールを行ってください。
>
> ScaTeFS/Clientがインストールされた環境でMLNX_OFEDをアンインストールすると、MLNX_OFEDの再インストール後にScaTeFS/Clientのサービスが起動しなくなります。この場合、MLNX_OFEDの再インストール後に、ScaTeFS/Clientのパッケージを再インストールしてください。

6.1.2　疎通確認

以下の方法でIBによる通信を確認することができます。

- IBネットワークの疎通確認

  ibpingコマンドを使用して疎通確認してください。ibpingはクライアント/サーバ型のプログラムで以下のようにクライアント/IOサーバの双方でプログラムを実行します。

  ①IOサーバでサーバプログラムとして実行

  ```
  # ibping -S
  ```

  ②クライアントでクライアントプログラムとして実行

  ```
  # ibping -L LID
  ```

  LIDにはIOサーバのHCAポートのLIDを指定します。LIDはIOサーバ上でibstatコマンドを実行すると、通信対象ポートのBase lidの項目で確認できます。

- IB上のIPネットワークの疎通確認

  ping コマンドの-I オプションを使うと送信元インターフェースを指定できます。

  ScaTeFS クライアントで使うIBのインターフェース名を指定して、IO サーバのインター

フェースと通信できるか確認してください。

```
# ping ServerAddress -I ibbondN
```

ServerAddressにはIOサーバのIPoIBアドレスを指定します。

6.1.3　パッケージのインストールとアップデート

Linuxクライアントのマシンのタイプにより、パッケージのインストールとアップデートの方法が異なります。

- SX-Aurora TSUBASAの場合

  「SX-Aurora TSUBASA インストレーションガイド」をご覧ください。

- SX-Aurora TSUBASA以外のLinuxマシンの場合

  製品に同梱されているインストレーションガイド(Installation_Guide_for_Scalar_J.txt)をご覧ください。

  ※SXクロスソフトウェア ノードロックライセンスをお使いの場合は、製品に同梱されているリリースメモをご覧ください。

6.1.4　ScaTeFS のライセンス登録

ライセンス登録を行います。

手順の詳細については、「HPC ソフトウェアライセンス管理説明書」をご覧ください。

※SXクロスソフトウェア ノードロックライセンスをお使いの場合は、「HPC ソフトウェアライセンス管理説明書」ではなく「SXクロスソフトウェア ノードロックライセンス導入ガイド」をご覧ください。

6.1.5　ScaTeFS InfiniBand 高速 IO ライブラリの設定

ScaTeFS IBライブラリまたはScaTeFS IB VEダイレクトライブラリを使用する場合に必要な共通の設定です。ScaTeFS InfiniBand 高速IOライブラリの概要については「9.12 ScaTeFS InfiniBand 高速IOライブラリ」を参照してください。

- プログラムがロックできるメモリ量の最大値設定

  ScaTeFS IBライブラリは効率的にIOを行うため、ユーザプログラムのIOで使用するメモリ領域(read(2)/write(2)の引数に指定するメモリ領域)をロックします。ユーザプログラムがIOで使用するメモリを多くロックできるほど、効率的なIOが可能となり性能向上が期待できます。デフォルトの最大値は数十KBとなっており(ulimit –aの出力中の「max locked memory」で確認可能)、MB単位以上の大きなIOを行うプログラムでは従来のカーネルに

よるIOが実行されて、期待する性能が出ません。

そのため、/etc/security/limits.confのmemlockの値を設定して、ユーザプログラムのロック可能なメモリ量の最大値を拡大してください。100MB以上に設定することをお勧めします。

ただし、ロックしたメモリはプロセス終了するまでスワップアウトされないので、多くのプロセスが最大値までIOでメモリを使用すると通常より早くシステムがメモリ不足になる可能性があります。「ライブラリを使用して実行するプログラムの想定する同時実行数」と「ロック可能メモリの最大値」の積がクライアントマシンの実装メモリの50%を超えないように設定することを推奨します。

以下は、ユーザプログラム(プロセス)のロックできる最大値を無制限に設定する場合の例です。

/etc/security/limits.confの記述イメージ
* soft memlock unlimited * hard memlock unlimited

- ライブラリが使用するHCAポートについて

  ScaTeFS IBライブラリはibdevマウントオプションに指定したラベルに属するHCAポートを自動的に検出してIO処理で使用します。ラベルに複数HCAポートを指定して使用する場合、経路障害時の自動的な経路切り替えを行います。ibdevマウントオプションに指定するラベルについては6.1.6を参照してください。

6.1.6　マウント方法

mountコマンドを使ってファイルシステムをマウントします。

以下に、ファイルシステム”scatefs00”を/mnt/scatefsにマウントする例を示します。

```
# mount -t scatefs -o ibdev=HOME,rsize=4194304,wsize=4194304 ServerAddress:scatefs00
/mnt/scatefs
```

ServerAddressには、ルートIOサーバのIPoIBのIPv4アドレスを指定します。

マウントオプションのrsizeとwsizeは、クライアントとIOサーバの間でファイルのデータを入出力する際の転送サイズを表します。ともに既定値は1MBですが、2MB、または4MBとした方が性能は向上します。

ファイルシステムに関する情報を/etc/fstabに記述し、Linuxマシンの起動時にファイルシステムを自動的にマウントする場合、マウントオプションに_netdevを記述してください。以

下に/etc/fstabの記述例を示します。

```
ServerAddress:scatefs00                    /mnt/scatefs                    scatefs
_netdev,ibdev=HOME,rsize=4194304,wsize=4194304 0 0
```

- _netdevを記述しない場合、RHEL/CentOS 7および8ではLinuxマシンの起動時にファイルシステムのマウントに失敗し、緊急モードのログインプロンプトがコンソールに表示されます。この場合、マウントオプションに_netdevを追加し再起動してください。
- マウントオプションとしてSELinuxのコンテキストが指定されなかった場合、既定値としてcontext="system_u:object_r:nfs_t:s0"が使用されます。他のコンテキストを使用したい場合は、マウントオプションでコンテキストを指定してください。

以下は、IB環境でのみ有効なマウントオプションです。

- ibdev=LABEL

  IB Verbsの通信に利用するHCAデバイスとHCAポートを指定するマウントオプションです。LABELには設定ファイルに定義したラベルを指定します。設定ファイルについては 6.1.7 を参照してください。

  ※本オプションを指定しない場合、IB VerbsではなくIPoIBを使った通信となります。必ず指定を行ってください。

- mpri=N

  メタデータアクセスの優先度を指定します。サービスレベル(0～14までの値)を指定します。指定しなかった場合はmpri=0 が適用されます。このオプションを有効にするには、サブネットマネージャにQoSの設定が必要です。

- dpri=N

  READ/WRITEの優先度を指定します。サービスレベル(0～14までの値)を指定します。指定しなかった場合はdpri=0 が適用されます。このオプションを有効にするには、サブネットマネージャにQoSの設定が必要です。

  ※サブネットマネージャのQoS設定方法は、サブネットマネージャのマニュアルを参照してください。

ibdev=LABEL を指定してマウントすると、mountコマンドは設定ファイルに記載されたLABELの定義をもとにHCAデバイス名を解決します。定義がないか誤っている場合は、mountコマンドはエラーとなります。

以下は、ScaTeFSをマウント後に、mountコマンドを引数なしで実行したときの例です。

赤字のマウントオプション(ibhca*N*)が表示される場合にはIB Verbsを利用した通信となっています。

```
# mount
(...)
172.28.71.1:scatefs00 on /mnt/scatefs type scatefs
(rw,relatime,hard,cto,ac,sync_on_close,ibhca1=mlx5_0:1,mpri=0,dpri=0,port=
50000,rsize=4194304,wsize=4194304,timeo=600,retrans=1,acregmin=3,acregmax=
60,acdirmin=30,acdirmax=60,addr=172.28.71.1)
```

IB Verbsが有効なとき、データのI/OにはInfiniBandのネイティブなプロトコルが利用されます。この通信はソケットを使用しないため、データI/O用の50002番ポート宛のコネクションは作成されません。

```
# dd if=/dev/zero of=/mnt/scatefs/testfile bs=1M count=1
# ss -n  | grep 50002
#  (表示なし)
```

なお、IB Verbsが有効であっても50000番ポート宛のコネクションは作成されます。これらのコネクションはIPoIBによる制御通信に利用されます。

ibdev=LABEL を指定しないでマウントした場合は、通信はすべてIPoIBで行います。このとき、mountコマンドの出力結果にibhca*N*のマウントオプションは表示されません。

```
# mount
(...)
172.28.71.1:scatefs00 on /mnt/scatefs type scatefs
(rw,relatime,hard,cto,ac,sync_on_close,port=50000,rsize=4194304,wsize=4194
304,timeo=600,retrans=5,acregmin=3,acregmax=60,acdirmin=30,acdirmax=60,add
r=172.28.71.1)
```

IB Verbsが無効の場合、50002番ポート宛のコネクションが作成されます。この場合、マウントオプションを確認してください。

```
# dd if=/dev/zero of=/mnt/scatefs/testfile bs=1M count=1
# ss -n  | grep 50002
tcp    ESTAB      0      0      172.28.7.1:963              172.28.71.1:50002
tcp    ESTAB      0      0      172.28.7.1:963              172.28.71.2:50002
```

各種マウントオプションの詳細については、scatefs(5)のmanデータで参照できます。

6.1.7　クライアントの HCA デバイスの設定

IBを利用する場合、マウントオプションのibdevに、HCAデバイスとポートを表すラベルを指定します。ラベルは設定ファイル "/etc/scatefs/client/ibdevice.conf" に定義します。以下のように、「HCAカードが差さっているPCI-ID」を用いてHCAデバイスを指定します。

/etc/scatefs/client/ibdevice.confの記載イメージ

```
#
# /etc/scatefs/client/ibdevice.conf
#
# This is the configuration file for ScaTeFS mount option 'ibdev'.
#
HOME1      0000:83:00.0   0000:83:00.1 … HOME1というラベルを定義、PCI情報 2つを関連付け
HOME2      0000:83:00.1   0000:83:00.0 … HOME2というラベルを定義、PCI情報 2つを関連付け

WORK1      0000:83:00.0                … WORK1というラベルを定義、PCI情報 1つを関連付け

# HCAのポート番号を明示的に指定する場合はPCI-IDの末尾に「@ポート番号」という記述をする
# (省略したときは @1 が指定されたものとみなす)
WORK2      0000:83:00.1@1
```

設定ファイルの書式ルールは以下のとおりです。

- 一行につきラベル1つを定義できます。ラベルを複数定義しておくことで、マウントポイントごとに異なる設定でマウントが可能です。
- PCI-IDは、半角スペースまたはタブで区切り、最大12個まで指定できます。
- ラベルに使用できる文字は、半角英数字およびアンダーバー(_)です。
- ラベルに指定できる最大文字数は32文字です。
- 行頭に # を書くと、コメント行とみなし無視します。

例にあるラベル HOME1 について説明します。HOME1では 0000:83:00.0 と

0000:83:00.1 という2つのPCI-IDを指定しています。PCI-IDとHCAデバイスの対応付けは以下のように確認できます。

```
# ls -l /sys/class/infiniband/
total 0
(…) mlx5_0 -> ../../devices/pci0000:80/0000:80:02.0/0000:83:00.0/infiniband/mlx5_0
(…) mlx5_1 -> ../../devices/pci0000:80/0000:80:02.0/0000:83:00.1/infiniband/mlx5_1
```

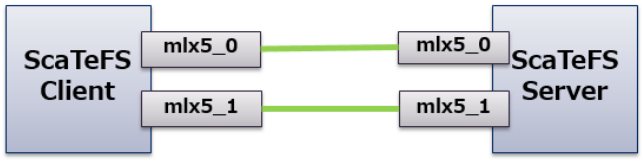
上記より、HOME1というラベルはmlx5_0 と mlx5_1 の2つのHCAデバイスを使って通信を行うことを意味しています(マルチパス通信)。

次に、HOME2のラベルについて説明します。このラベルはHOME1と同じPCI-IDを指定していますが、記述する順番が逆になっています。この定義はHOME1と等価ではなく、HOME1とは通信経路が異なるため注意が必要です。経路障害時の動作が変わったり、遠い経路を通ることで性能に影響がでたりする可能性があります。以下に詳細を記載します。

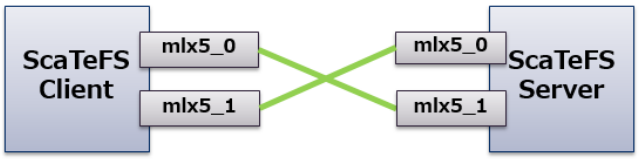
IOサーバが利用するHCAデバイスは、IOサーバの設定ファイルで定義します。このとき定義した1つ目のHCAデバイスと、クライアントの1つ目のHCAデバイスが通信を行います。同様に、IOサーバの2つ目のHCAデバイスと、クライアントの2つ目のHCAデバイスが通信を行います。

たとえば IOサーバ側で mlx5_0とmlx5_1の順番でHCAデバイスが定義されているとき、HOME1とHOME2のそれぞれがどのような通信経路になるのかを以下に示します。

【HOME1の設定でマウントしたケース】



【HOME2の設定でマウントしたケース】



いずれの設定でも通信は可能ですが、上記のようにPCI-IDを記述する順番によって通信する経路が変わるため、最適な設定を行うにはIOサーバ側のHCAデバイスの設定、および途中の経路を含むネットワーク構成を考慮して定義を行う必要があります。

6.1.8　HCA の構成とコネクション数について

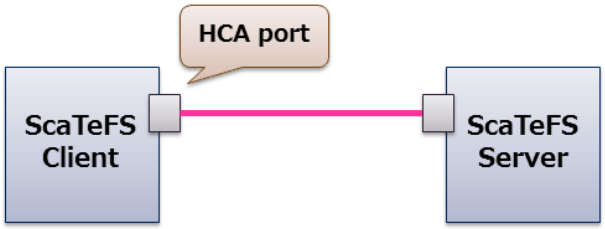
IB Verbsの通信では、クライアント2port、IOサーバ2portの構成のとき通信経路の組み合わせは4パターン存在します。ScaTeFS/Clientでは経路に対してコネクションを網羅的に張ることはせず、クライアントで指定したHCAとIOサーバのHCAのうち、多い側のノードに合わせた数のコネクションを確立します。これにより、設定ファイルに記載したHCAデバイスがすべて利用され、かつコネクション数が過剰に増えることを防いでいます。

様々な構成におけるコネクションの確立状況を以下に示します。

※ここでのコネクションは、IBにおけるQPの通信経路に該当します。

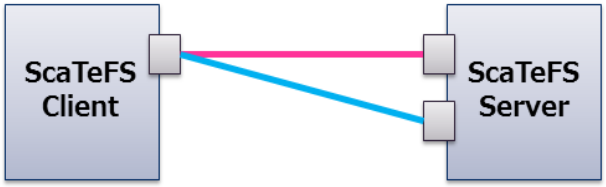
※以下の図ではクライアントに搭載したHCAポートをすべて使っていますが、一部だけを指定して使うことも可能です。

(1)　クライアントが 1 ポート、IO サーバが 1 ポートの構成

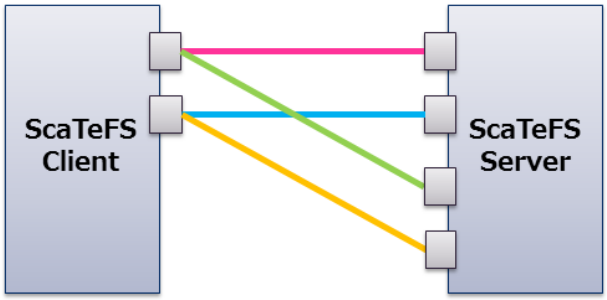


(2)　クライアントが 2 ポート、IO サーバが 2 ポートの構成

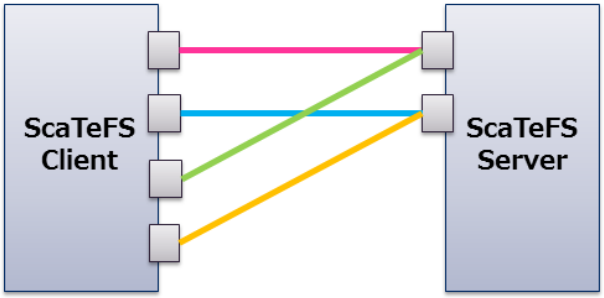
(3)　クライアントが 1 ポート、IO サーバが 2 ポートの構成



(4)　クライアントが 2 ポート、IO サーバが 4 ポートの構成



(5)　クライアントが 4 ポート、IO サーバが 2 ポートの構成



6.1.9　アンマウント方法

umountコマンドを使ってファイルシステムをアンマウントします。

以下に、/mnt/scatefsにマウントされているファイルシステムをアンマウントする例を示します。

```
# umount /mnt/scatefs
```

IOサーバとの通信が不通となった場合、-fオプションを使用することによりファイルシステムを強制的にアンマウントすることができます。以下に、/mnt/scatefsにマウントされているファイルシステムを強制的にアンマウントする例を示します。

```
# umount -f /mnt/scatefs
```

6.1.10　通信確認（ScaTeFS IB ライブラリ利用時）

ScaTeFS IB ライブラリを使用する場合は、11.6、11.6.6を参照してScaTeFS IB ライブラリを使用してIOを実行し、統計情報からScaTeFS IB ライブラリでIOが実行されたことを確認してください。

6.2　10GbE利用時の設定

6.2.1　DCB 対応版 10GbE-NIC ドライバのインストール

DCB対応版10GbEを使用する場合のみ実施してください。

ベンダが提供しているRPMバイナリパッケージはそのままではDCBに対応してない場合があります。サポート部門から入手したインストール手順にて10GbE-NICドライバのインストールを行ってください。

6.2.2　ルーティング設定

ScaTeFSクライアントは、IOサーバで設定した10GbE インターフェースの bond0, bond1 の両方と通信を行います(小規模向けIOサーバはbond0のみ)。そのため、IOサーバのbond0だけでなく bond1 とも10GbEで通信できるように静的なルーティング設定を追加してください。

現在のルーティング設定は以下で確認できます。ルーティングの設定に関する詳細はRHELの「導入ガイド」などを参照してください。

- ipコマンドでルーティングテーブルを表示

```
# ip route
```

- netstatコマンドでルーティングテーブルを表示

```
# netstat -r
```

ScaTeFSクライアントのルーティング設定が正しくない場合、以下のような現象が発生します。

- mountの応答が返って来ない

  ルートIOサーバに対するルーティング設定を確認してください。

- mountはできたが、その後のScaTeFSへのアクセスで応答が返らないことがある

  コネクションの一部が張れていない可能性があります。特に、bond1に対するルーティング設定を確認してください。

  LinuxではpingコマンドのI オプションを使うと送信元のネットワークインターフェースを指定できます。ScaTeFSクライアントで使う10GbEインターフェースを指定して、IOサーバの各10GbEインターフェースと通信できるか確認します。

```
# ping "IOサーバのbond0 IPアドレス" -I "クライアントのインターフェース名"
# ping "IOサーバのbond1 IPアドレス" -I "クライアントのインターフェース名"
```

6.2.3　パッケージのインストールとアップデート

- SX-Aurora TSUBASA以外のLinuxマシンの場合

  製品に同梱されているインストレーションガイド(Installation_Guide_for_Scalar_J.txt)をご覧ください。

  ※SXクロスソフトウェア ノードロックライセンスをお使いの場合は、製品に同梱されているリリースメモをご覧ください。

6.2.4　ScaTeFS のライセンス登録

ライセンス登録を行います。

手順の詳細については、「HPC ソフトウェアライセンス管理説明書」をご覧ください。

※SXクロスソフトウェア ノードロックライセンスをお使いの場合は、「HPC ソフトウェアライセンス管理説明書」ではなく「SXクロスソフトウェア ノードロックライセンス導入ガイド」をご覧ください。

6.2.5　マウント方法

mountコマンドを使ってファイルシステムをマウントします。

以下に、ルートIOサーバ”iosv00”のファイルシステム”scatefs00”を/mnt/scatefsにマウントする例を示します。

```
# mount -t scatefs -o rsize=4194304,wsize=4194304 iosv00:scatefs00 /mnt/scatefs
```

マウントオプションのrsizeとwsizeは、クライアントとIOサーバの間でファイルのデータを入出力する際の転送サイズを表します。ともに既定値は1MBですが、2MB、または4MBとした方が性能は向上します。

マウントオプションの詳細については、scatefs(5)のmanデータを参照してください。

ファイルシステムに関する情報を/etc/fstabに記述し、Linuxマシンの起動時にファイルシステムを自動的にマウントする場合、マウントオプションに_netdevを記述してください。本オプションを記述しない場合、RHEL/CentOS 6ではLinuxマシンの起動時に"can't mount ScaTeFS file system"のメッセージがコンソールへ出力されます。また、RHEL/CentOS 7および8ではLinuxマシンの起動時にファイルシステムのマウントに失敗し、緊急モードのログインプロンプトがコンソールに表示されます。この場合、マウントオプションに_netdevを追加し再起動してください。以下に/etc/fstabの記述例を示します。

```
iosv00:scatefs00 /mnt/scatefs scatefs _netdev,rsize=4194304,wsize=4194304 0 0
```

マウントオプションとしてSELinuxのコンテキストが指定されなかった場合、既定値としてcontext="system_u:object_r:nfs_t:s0"が使用されます。他のコンテキストを使用したい場

合は、マウントオプションでコンテキストを指定してください。

マウント後、全IOサーバとのコネクションが確立できるかどうか、IO確認を行います。

ScaTeFSは、同一ディレクトリにファイルを作成すると、各IOサーバにラウンドロビンで分散しますので、それを利用して確認します。

以下は、IOサーバ2台で実施する場合の例です。マウントポイントとループ回数は適宜変更してください。ループ回数はIOサーバ数以上とします。

```
# for N in {1..2}; do dd if=/dev/zero of=/mnt/scatefs/testfile${N} bs=10M count=1;
done;
```

- ssコマンドを用いて確認する場合

```
# ss -nt | egrep 'State|:5000'
```

- netstatコマンドを用いて確認する場合

```
# netstat -n | egrep 'Local|:5000' | sort
```

表示されるローカル側アドレスとリモート側アドレスを見て、意図したインターフェースが利用されているか確認してください。特に、ローカルアドレスが意図した10GbEのインターフェースになっていることを確認します。

以下は、bondingインターフェースが2つあるIOサーバを2台用いて ScaTeFSを構築したときの例です。

このとき、クライアントはIOサーバ1台あたり4つのコネクションを作成します。IOサーバが2台あるため、計8つのコネクションが表示されます。

```
Proto Recv-Q Send-Q Local Address       Foreign Address     State
tcp        0      0 172.28.134.43:869   172.16.6.5:50000    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.6.5:50002    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.6.6:50000    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.6.6:50002    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.7.5:50000    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.7.5:50002    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.7.6:50000    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.7.6:50002    ESTABLISHED
```

ScaTeFSでは、IOサーバのbonding インターフェース1つに対し、ポート番号50000(メタデータ通信用)と 50002(データ通信用)の2つのコネクションを作成します(IOサーバの設定にて、データポートを無効にした場合は、50000のみ)。

また、コネクションは IOサーバとの通信が必要になった時点で作成するため、IOがまだあまりされていない段階では、コネクションは一部のみ張られた状態になります。

6.2.6　アンマウント方法

umountコマンドを使ってファイルシステムをアンマウントします。

以下に、/mnt/scatefsにマウントされているファイルシステムをアンマウントする例を示します。

```
# umount /mnt/scatefs
```

IOサーバとの通信が不通となった場合、-fオプションを使用することによりファイルシステムを強制的にアンマウントすることができます。以下に、/mnt/scatefsにマウントされているファイルシステムを強制的にアンマウントする例を示します。

```
# umount -f /mnt/scatefs
```

6.3　補足事項

6.3.1　NFS サーバを使ってエクスポートする方法

Linuxクライアント上のNFSサーバを使って、ファイルシステムをNFSクライアントへエクスポートすることができます。

ファイルシステムをエクスポートする場合、/etc/exportsにfsidオプションを使ってファイルシステムを識別する整数を記述する必要があります。以下に/etc/exportsの記述例を示します。

```
/mnt/scatefs *(rw,no_root_squash,mp,fsid=1)
```

また、以下の注意事項があります。

- サポートするNFSバージョンは3のみです。また、サポートするプロトコルはTCPのみです。
- サポートするNFSクライアントはLinuxのみです。
- NFSクライアントがLinuxの場合、NFSクライアントでのマウント時にNFSバージョンとして3を明示してください。Linuxのディストリビューションによっては(たとえばRHEL 6)、NFSバージョンを明示しないとNFSバージョンとしてサポート対象外の4が使用されます。
  NFSバージョンを3とする他の方法として、NFSサーバの設定により、NFSクライアントでのNFSバージョン4の使用を防止することもできます。設定の詳細についてはお使いのRHELの「ストレージ管理ガイド」を参照してください。

- NFSクライアントがファイルをロックした場合、そのロックの影響を受けるのは、同じNFSサーバの他のNFSクライアントと、そのNFSサーバが存在するScaTeFSのクライアントのみです。NFSを介さずにファイルシステムを直接アクセスするクライアントは、NFSクライアントによるロックの影響を受けません。

6.3.2　ファイルクローズ時の同期遅延

マウントオプションでのsync_on_close(既定値)またはno_sync_on_closeの指定により、ファイルクローズ時にクライアントがファイルデータをIOサーバのスレージと同期させるかどうかを指定することができます。

sync_on_close(既定値)の場合、ファイルクローズ時に、クライアントはファイルに書き出されたデータをIOサーバへ送信し、送信したデータとIOサーバのストレージの同期を行います。データ保全性が最も高いモードです。

no_sync_on_closeの場合、ファイルクローズ時に、クライアントはファイルに書き出されたデータをIOサーバへ送信しますが、送信したデータとIOサーバのストレージの同期は行いません。送信したデータとIOサーバのストレージの同期は、ファイルクローズ後に非同期で行われます。sync_on_closeに比べデータ保全性は低下しますが、数十KB以下の小さなファイルの作成時間を短縮することができます。

no_sync_on_closeを指定した場合、ファイルに書き出されたデータをIOサーバのストレージと同期させる処理が、ファイルクローズ後に遅延されることにより、ファイルクローズの処理時間が短縮されます。これにより、数十KB以下の小さなファイルを多数作成するtarコマンドやcpコマンド等の処理時間を短縮することができます。

ただし、次の場合では、no_sync_on_closeを指定してもアプリケーションの処理時間の短縮効果は小さい、または、短縮効果はありません。

- ファイルに書き込んだデータのサイズが大きい場合(数十KB以上)
- アプリケーションで明示的に書き込んだデータの同期(fsync(2),msync(2)等)を行っている場合
- アプリケーションでレコードロックやファイルロックを行っている場合
- アプリケーションで同一ファイルに対してオープンとクローズを繰り返し行っている場合

no_sync_on_closeを指定した場合、次の注意事項があります。

- ファイルクローズ後にクライアントとIOサーバの両方が同時にダウンした場合、IOサー

バのストレージへの同期がまだ完了していない更新データは失われます。アプリケーションがデータを書き出してからIOサーバのストレージへの同期が完了するまでの時間は、クライアントでのダーティデータが書き出されるまでの時間の設定に依存しますが、概ね2分以内です。

- ファイルクローズ後に、更新データのIOサーバのストレージへの遅延同期処理でエラーが発生した場合、ファイルをクローズしたアプリケーションでそのエラーを検出することはできません。ここでのエラーとしては、たとえばストレージ障害によるIOエラーがあります。IOサーバダウンは含みません。
  このとき、クライアントの syslog にエラーメッセージ (ScaTeFS:400100, ScaTeFS:400101)が出力されます。障害にあったファイルは、障害発生日時(前述のメッセージの出力日時)、メッセージ中のファイルシステム情報やファイル情報、アプリケーションによるファイルのアクセス状況等から特定することになります。

6.4　注意事項

6.4.1　オープンしているファイルの削除について

1つのクライアント上で、あるプロセスがオープンしているファイルを削除すると、オープン中のファイルはすぐには削除されず、いったん次の形式のファイル名に自動的にリネームされます。

形式: .scatefsXXX...X(X:英数字)

例: .scatefs00000000010107640000000ab

このファイルは、オープンしていたプロセスからクローズされると自動的に削除されます。自動的に削除される前に、このファイルを手動で削除しようとすると、"Device or resource busy"のエラーになります。

6.4.2　管理ネットワークで DHCP を利用する場合の注意事項

SX-Aurora TSUBASAの運用において、DHCPにより管理ネットワークのIPアドレス設定を行っている場合、DHCPによるIPアドレス設定が遅延することにより、システム起動時のScaTeFSのファイルシステムの自動マウントが失敗することがあります。
このとき、シスログに以下のいずれかのメッセージが出力されます。

```
ScaTeFS_LS:400201 ScaTeFS client license process failed. reason=<Failed to connect to
IPAddress (port=7300): Network is unreachable>
ScaTeFS_LS:400201 ScaTeFS client license process failed. reason=<Failed to connect to
IPAddress (port=7300): No route to host>
```

この場合はシステム起動後に手動でファイルシステムをマウントしてください。

6.4.3　ScaTeFS InfiniBand 高速 IO ライブラリ使用時の注意事項

- clone(2)の直接呼び出し未サポート
  clone(2)をプログラム中で直接呼び出すことはできません。fork(2)またはvfork(2)を使用してください。
- ScaTeFS IBライブラリと他ライブラリの環境変数LD_PRELOADへの同時設定未サポート
  環境変数LD_PRELOADにScaTeFS IBライブラリと他のライブラリを同時に指定して、使用することはできません。ScaTeFS IBライブラリを使用する場合は、ScaTeFS IBライブラリのみをLD_PRELOADに指定してください。
- ScaTeFS VEダイレクトIBライブラリと他ライブラリの環境変数VE_LD_PRELOADへの同時設定未サポート
  環境変数VE_LD_PRELOADにScaTeFS VEダイレクトIBライブラリと他のライブラリを同時に指定して、使用することはできません。ScaTeFS VEダイレクトIBライブラリを使用する場合は、ScaTeFS VEダイレクトIBライブラリのみをVE_LD_PRELOADに指定してください。

6.4.4　二重マウント時の注意事項（RHEL/CentOS 8.1 以降)

ScaTeFSのファイルシステムをマウントした状態で、同じマウントポイントに他のファイルシステムをマウントした場合、はじめに他のファイルシステムをumountコマンドを使用してアンマウントし、次にScaTeFSのファイルシステムを/sbin/umount.scatefsコマンドを使用してアンマウントしてください。2番目のScaTeFSのファイルシステムのアンマウントを、umountコマンドを使用して行った場合、エラーとなりアンマウントに失敗します。

```
# umount /mnt/scatefs
# /sbin/umount.scatefs /mnt/scatefs
```

6.4.5　mlocate パッケージを使用する場合の注意事項

mlocateパッケージをインストールしている場合、既定値では updatedb が ScaTeFS のパスも毎日チェックします。各クライアントでこれが行われるとシステムへの大きな負荷になります。以下のように /etc/updatedb.conf ファイルの PRUNEFS に scatefs を追加してチェック対象外としてください。

```
# rpm -q mlocate
mlocate-XXX.x86_64
# grep PRUNEFS /etc/updatedb.conf
PRUNEFS = "9p afs anon_inodefs auto autofs bdev binfmt_misc cgroup cifs coda configfs
```

```
cpuset debugfs devpts ecryptfs exofs fuse fuse.sshfs fusectl gfs gfs2 gpfs hugetlbfs
inotifyfs iso9660 jffs2 lustre mqueue ncpfs nfs nfs4 nfsd pipefs proc ramfs rootfs
rpc_pipefs securityfs selinuxfs sfs sockfs sysfs tmpfs ubifs udf usbfs ceph fuse.ceph
scatefs"
```

第7章　SX-ACE クライアントの設定

7.1　ルーティング設定

ScaTeFSクライアントは、IOサーバで設定した10GbEインターフェースの bond0, bond1 の両方と通信を行います(小規模向けIOサーバはbond0のみ)。そのため、IOサーバのbond0 だけでなく bond1 とも10GbEで通信できるように静的なルーティング設定を追加してください。ルーティングの設定に関する詳細はSUPER-UXの「ネットワーク運用の手引」を参照してください。

```
# netstat -r
```

ScaTeFSクライアントのルーティング設定が正しくない場合、以下のような現象が発生します。

- mountの応答が返って来ない
  ルートIOサーバに対するルーティング設定を確認してください。
- mountはできたが、その後のScaTeFSへのアクセスで応答が返らないことがある
  コネクションの一部が張れていない可能性があります。特に、bond1に対するルーティング設定を確認してください。
- 以下のようなメッセージがコンソールに出る

```
WARNING:ScaTeFS: RPC: connect error 151 server XX.XX.XX.XX:5000X: Not offloaded
connection
```

ノンオフロードインターフェース(en0)を経由して通信を行おうとしたためエラーとなっています。オフロードインターフェース(ex0)を経由するようにIOサーバ "XX.XX.XX.XX" に対するルーティング設定を見直してください。

7.2　ライセンス

当該ノードにScaTeFSクライアントのロック解除コードが適用され、パッケージ ”NEC Scalable Technology File System/Client” がインストールされている必要があります。

7.3　config変数

ScaTeFSを使用する場合は、config変数SCATEFSを 1 にし、データキャッシュとして使用する容量をconfig変数SCFS_DCACHEに設定します。SCFS_DCACHEの値は、XMキャッシュ領

域から確保する割合(%)を示します。XMキャッシュ領域は、インストール時にISLパラメータファイルのCACHEDEVにより指定します。

ScaTeFS を使用しない場合は、config 変数 SCATEFS を 0 にします。その場合、SCFS_DCACHEの値は無効となります。

7.4 ScaTeFSデーモン

クライアントには、ScaTeFSデーモンscatefs_rpcd(1M) が起動されている必要があります。

本デーモンの数は、同一クライアント内において同時にIOサーバへ発行できるI/Oのリクエスト数になります。/etc/init.d/scatefsのscatefs_rpcdに与える引数を変更することにより、起動するデーモン数を変えることができます。デーモン数の既定値は４です。

7.5 ScaTeFS経路監視デーモン

ScaTeFS経路監視デーモンscatefs_pmond(1M) は、クライアントとIOサーバ間のネットワーク経路に障害が発生したときに、その経路の状態を定期的に監視するデーモンです。経路の復旧を検知すると、ScaTeFSクライアントは再びその経路を使って通信を行うようになります。

このデーモンはマルチユーザモード移行時に起動されます。

7.6 マウント方法

下記は、転送サイズ4MB、シグナル割り込み可でファイルシステム名”scatefs00”をマウントする場合のmount(1M)のコマンドラインイメージです。マウントは、ルートIOサーバのみに対して行います。

```
# mount -t scatefs -o intr,rsize=4194304,wsize=4194304 rootsrv:scatefs00 /mnt/scatefs
```

scatefs: ファイルシステムタイプ(固定)

intr: シグナルによる割り込み可。

rsize,wsize: 転送サイズ。デフォルト1MBですが、2MB、または4MBとした方が性能は向上します。

rootsrv: ルートIOサーバのホスト名、または、ルートIOサーバのIPアドレス。

scatefs00: mkfs実施時に指定したファイルシステム名。

/mnt/scatefs: ScaTeFS をマウントするクライアント上のディレクトリ。

マウントオプションの詳細は、SUPER-UXのmount(1M)を参照してください。

マウント後、全IOサーバとのコネクションが確立できるかどうか、IO確認を行ってください。

ScaTeFSは、同一ディレクトリにファイルを作成すると、各IOサーバにラウンドロビンで分散しますので、それを利用して確認します。

以下は、IOサーバ2台で実施する場合の例です。マウントポイントとループ回数は適宜変更してください。ループ回数はIOサーバ数以上とします。

```
# for N in {1..2}; do dd if=dummyfile of=/mnt/scatefs/testfile${N} bs=10240k count=1;
done;
```

{1..2} にはループ回数を指定します。たとえば、4回ループする場合には {1..4} としてください。/mnt/scatefs/ にはマウントポイントを指定します。dummyfileには、10MB程度のダミーファイルを指定してください。

IOが完了したら、以下のコマンドを実行してコネクションの状態を確認します。

```
# netstat -n | egrep 'Local|:5000' | sort
```

表示される Local Address と Foreign Address をみて、意図したインターフェースが利用されているか確認してください。特に Local Address が10GbE(オフロードex0)のインターフェースになっていることを確認します。

以下は、bonding インターフェースが2つのIOサーバ2台を用いて ScaTeFSを構築したときの例です。IOサーバあたり4本(bond0, bond1にそれぞれ50000, 50002)となるため、計8本のコネクションが表示されます。

```
Proto Recv-Q Send-Q Local Address       Foreign Address     State
tcp        0      0 172.28.134.43:869   172.16.6.5:50000    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.6.5:50002    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.6.6:50000    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.6.6:50002    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.7.5:50000    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.7.5:50002    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.7.6:50000    ESTABLISHED
tcp        0      0 172.28.134.43:869   172.16.7.6:50002    ESTABLISHED
```

ScaTeFSでは、IOサーバのbonding インターフェース1つにつき、ポート番号50000(メタデータ通信用)と 50002(データ通信用)の2つのコネクションを作成します(IOサーバの設定にて、データポートを無効にした場合は、50000のみ)。

また、コネクションは IOサーバとの通信が必要になった時点で作成するため、IOがまだあ

まりされていない段階では、コネクションは一部のみ張られた状態になります。

7.7　データキャッシュ

データキャッシュに関連するマウントオプションは下記のとおりです。

- sync, async

  データキャッシュの使用の有無を指定します。

  syncを指定した場合、データキャッシュを使用せずにIOサーバへ同期書き込み／読み込みを行います。

  asyncを指定した場合、データキャッシュを使用してI/Oの高速化を図ります。ただし、1MB以上のI/Oサイズでデータを書き込む場合は、asyncであってもsyncと同等の処理を行います。

  デフォルトはasyncです。

- csize

  データキャッシュを使用する場合に、IOサーバへリクエストをただちに送信するか否かを切りかえるしきい値を指定します。

  データキャッシュが有効であり、I/Oサイズが1MB未満のwrite系システムコールで意味を持ちます。csizeの既定値は、1MBです。

  I/Oサイズとcsizeの設定値により、以下のような処理を行います。

  csize ≦ I/Oサイズ ＜ 1MB ： データキャッシュにデータを載せると同時にデータを直ちにIOサーバへ送信し、データがディスク媒体に書き終わっていなくてもクライアントへ終了通知が返却されます。

  I/Oサイズ ＜ csize ＜ 1MB ： キャッシュに載せるのみでwrite backします。

  なお、マウントオプションの詳細は、SUPER-UXのmount(1M)を参照してください。

7.8　設定ファイル

設定ファイル(/etc/fstab, /etc/scatefs/client.conf)が各ノードへ配布されている必要があります。各ノードへの配布方法は、SUPER-UXのインストレーションガイドを参照してください。

/etc/scatefs/client.conf は将来の機能拡張のための設定ファイルであり、特に変更する必要はありません。

7.9　アンマウント方法

umount(1M)コマンドを使ってファイルシステムをアンマウントします。

以下に、/mnt/scatefsにマウントされているファイルシステムをアンマウントする例を示します。

```
# umount /mnt/scatefs
```

第8章　Docker のコンテナから ScaTeFS を利用する際の設定

8.1　ScaTeFSの設定ファイルの設定

SX-Aurora TSUBASAを使用する場合に必要な設定です。

Dockerのコンテナを起動する全てのクライアントにおいて設定ファイル(/etc/scatefs/client/libscatefsib.conf)を作成して、ファイル内に以下を記述してください。

```
RDMA_FROM_VH_ON 0
```

8.2　コンテナの起動イメージの設定

コンテナの起動イメージにScaTeFSのコンテナ用のパッケージグループをインストールする必要があります。インストールするパッケージグループはSX-Aurora TSUBASAとSX-Aurora TSUBASA以外のLinuxマシンで異なります。それぞれのケースにおいて、以下のようにDockerfileに追記してからイメージを作成してください。

(1) SX-Aurora TSUBASA の場合

```
RUN     yum -y group install scatefs-client-tsubasa-container
```

※追記する箇所は必ずパッケージグループve-container-infinibandのインストールの後になるようにしてください。

(2) SX-Aurora TSUBASA以外のLinuxマシンの場合

```
RUN     yum -y group install scatefs-client-scalar-container
```

コンテナの起動イメージに関するその他の設定については、「NEC Network Queuing System V (NQSV) 利用の手引 [管理編]」を参照ください。

8.3　コンテナ起動用スクリプトの設定

コンテナ起動用スクリプト内のdocker run実行時のオプションに以下を追加してください。

オプション	内容
-v */scatefs-dir-on-host*: */mount-dir-on-container*:rw	ジョブがアクセスするホスト上の ScaTeFS のディレクトリのパス名(*scatefs-dir-on-host*)と、コンテナ内でのマウントポイント(*mount-dir-on-container*)を指定してくだ

オプション	内容
	さい。
-v /var/run/scatefs: /var/run/scatefs:z	コンテナ内のプロセスからホスト上の ScaTeFS のデーモンにアクセスするために必要な設定です。左記の通りに変更せずに指定してください。
-v /etc/scatefs/client/: /etc/scatefs/client/:ro	SX-Aurora TSUBASA を使用する場合に必要な設定です。 コンテナ内のプロセスから ScaTeFS の設定ファイルを参照できるようにするために必要な設定です。左記の通りに変更せずに指定してください。

コンテナ起動用スクリプトに関するその他の設定については、「NEC Network Queuing System V (NQSV) 利用の手引 [管理編]」を参照ください。

8.4　注意事項

Dockerのコンテナにおいて、mountコマンドでScaTeFSをマウントすることはできません。ホスト側でScaTeFSをマウントし、docker runの-vオプションでジョブが使用するScaTeFSのディレクトリを指定してください。

第9章　　運用管理

IOサーバの運用管理ではシステム運用を停止して行う場合があります。IOサーバの運用を停止するにはIOサーバデーモンの停止を行います。IOサーバデーモンの停止および起動はCLUSTERPROのコマンドを使用しますので、ペアのIOサーバのどちらかで以下のコマンドを実行します。

- IOサーバデーモンの停止

```
# clprsc -t exec1
# clprsc -t exec2
```

- IOサーバデーモンの起動

```
# clprsc -s exec1
# clprsc -s exec2
```

9.1　資源制限(QUOTA)

ファイルシステムおよびストレージグループごとに下記のQUOTA機能を提供します。

表 9-1　QUOTA 機能

種別	QUOTA機能	分類	
		ソフトリミット	ハードリミット
ユーザ	ファイル数	○	○
	ディスク容量	○	○
グループ	ファイル数	○	○
	ディスク容量	○	○
ディレクトリ	ファイル数	○	○
	ディスク容量	○	○

QUOTAは、ユーザやグループ、ディレクトリごとに設定が可能となります。制御対象はファイル数とディスク容量であり、それぞれハードリミットとソフトリミットによる制限が可能です。

ハードリミットとは、その値に達した場合はそれ以上アロケートすることができない制限値です。ハードリミットに達した場合、書き込み要求に対しEDQUOTを返却します。

ソフトリミットとは、一時的に超えることができる制限値です。この値を超えた状態で設定された猶予期間を経過した場合、ハードリミットに達した場合と同様に扱います。猶予期間はデフォルトでは7日間ですが、ファイルシステム・ストレージグループごとに1秒から(2^{32}-1)秒の範囲で設定可能です。設定方法については、「9.1.1.2 scatefs_edquotaコマンド」を参照してください。

ハードリミットに到達またはソフトリミットに到達後に猶予期間が経過し、書き込みができない状態となった場合、ハードリミット、ソフトリミットを下回るまでファイルの削除を行うか、scatefs_edquotaコマンドにて、ハードリミット、ソフトリミットの上限値を変更することで解消されます。

ディスク容量の計算には、各IOターゲットに配置する実ファイルのファイルサイズを使用します。このため、実ファイルのホールサイズも使用量として計算されます。

QUOTA機能は、IOサーバ構築後は有効になっています。QUOTA機能が無効の場合には、ファイル数およびディスク容量の使用量をカウントしません。

- ディレクトリQUOTA

　ユーザ/グループ QUOTA については Linux 標準の機能であるためここでは説明を省略し、ディレクトリクォータについて説明します。ディレクトリ QUOTA は、ユーザ/グループ単位の QUOTA 制限とは別にディレクトリ単位で QUOTA 制限を行う機能です。ディレクトリ QUOTA とユーザ/グループ QUOTA の使用量管理は同時に機能します。ディレクトリ QUOTA を使用することで、より柔軟な資源管理が可能となります。

　図 9-1 は、ファイルシステム（FS1）のユーザ/グループの QUOTA 制限とは別に、proj1/proj2 ディレクトリそれぞれで QUOTA 制限を行うイメージです。

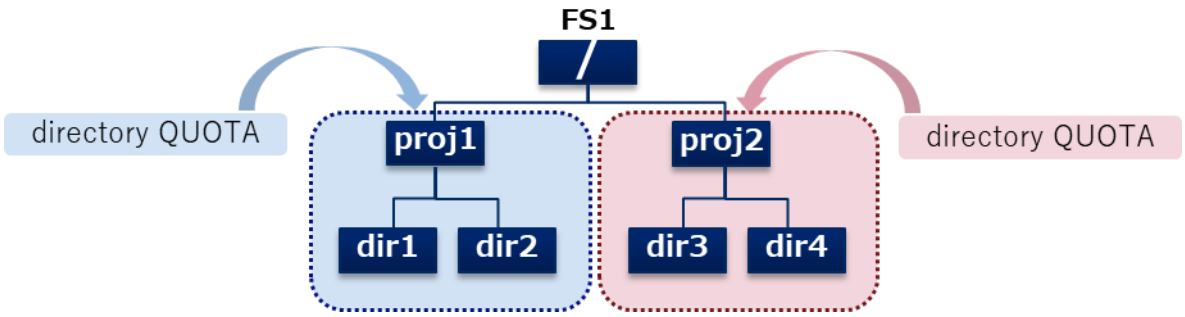


図 9-1 ディレクトリクォータイメージ図

　ディレクトリクォータの運用は以下の手順で行います。

(1) QUOTA制御ディレクトリの作成

　ディレクトリQUOTAを使用するためには、まず起点となるディレクトリを作成します。この起点となるディレクトリのことをQUOTA制御ディレクトリと呼びます。図 9-1では、proj1/proj2がQUOTA制御ディレクトリに該当します。

　QUOTA 制御ディレクトリの作成には、scatefs_mkqdirコマンドを使用します。scatefs_mkqdirコマンド については、9.1.1.5 を参照してください。

(2) QUOTA情報の編集

　ディレクトリ QUOTA 情報の編集には、scatefs_edquota コマンドを使用します。scatefs_edquota コマンドについては、9.1.1.2 を参照してください。

(3) QUOTA 設定の確認

　QUOTA 設定の確認には、scatefs_quota/scatefs_repquota コマンドを使用します。scatefs_quota コマンドについては、9.1.1.3 を、scatefs_repquota コマンドについては 9.1.1.4 を参照してください。

df コマンドの引数に QUOTA 制御ディレクトリやその配下のファイルを指定することで、ディレクトリ QUOTA としての使用状況を確認できます。この場合、df コマンドの表示は以下となります。

- 使用（Used）　　：ディレクトリ QUOTA 内の使用量
- 使用可（Available）：ハードリミットまでの残り量（※）

※ ハードリミットよりも実際のファイルシステムの空き容量が少ない場合は、ファイルシステムの空き容量が使用可能量として表示されます。

（例 1）

```
# mount -t scatefs HOST:FS1 /mnt/scatefs
# df /mnt/scatefs/proj1
Filesystem  1K-blocks  Used  Available Use%  Mounted on
HOST:FS1       200704     0     200704   0%  /mnt/scatefs
```

QUOTA制御ディレクトリをサブディレクトリマウントした場合、dfコマンドの結果に当該ディレクトリのQUOTA情報が表示されます。サブディレクトリマウントについては9.13を参照してください。

（例 2）

```
# mount -t scatefs HOST:FS1/proj1 /mnt/subdir
# df
Filesystem     1K-blocks  Used  Available Use%  Mounted on
:
HOST:FS1/proj1    200704     0     200704   0%  /mnt/subdir
```

(4) QUOTA 制御ディレクトリの削除

QUOTA制御ディレクトリの削除には、scatefs_rmqdirコマンドを使用します。scatefs_rmqdirコマンドについては、9.1.1.6 を参照してください。

9.1.1　コマンド

QUOTAの設定は、いずれかのIOサーバにログインし、ScaTeFS用QUOTAコマンドを実行する方法と、事前に登録されたLinuxクライアントマシンまたはSX-ACEクライアントマシンからリモートCLI（scatefs_rcli）によりScaTeFS用QUOTAコマンドを実行する方法を提供します。

関連するコマンドは各IOサーバデーモンが起動中かつQUOTA機能が有効な場合に限り実行可能となります。

以下、各コマンドの概要と代表的な実行イメージを記載します。詳細な利用方法については

各コマンドのmanを参照してください。

コマンド	概要
scatefs_quotacheck	QUOTA 情報の再計算と quota ファイルの修復を行う
scatefs_edquota	ユーザ、グループおよびディレクトリの QUOTA を編集する
scatefs_quota	ディスクの使用状況と使用限度を表示する
scatefs_repquota	QUOTA 情報一覧を表示する
scatefs_mkqdir	QUOTA 制御ディレクトリを作成する
scatefs_rmqdir	QUOTA 制御ディレクトリを削除する

9.1.1.1　scatefs_quotacheck コマンド

scatefs_quotacheckコマンドでは、各ファイルシステムやストレージグループのQUOTA情報の整合性を検証し、不具合があった場合に修正を行う機能を提供します。本コマンドは、IOサーバ上でのみ実行可能です。運用を停止してからscatefs_quotacheckコマンドを実行してください。

(例) IOサーバ上で、ファイルシステムscatefs00のユーザやグループ、ディレクトリに対し、QUOTA情報の整合性を検証する

```
# su fsadmin
$ scatefs_quotacheck scatefs00
```

(例) IOサーバ上で、すべてのファイルシステム、ストレージグループのユーザやグループ、ディレクトリに対し、QUOTA情報の整合性を検証する

```
# su fsadmin
$ scatefs_quotacheck -a
```

(例) IOサーバ上で、ファイルシステムscatefs00のグループに対し、設定されたハードリミット、ソフトリミットをクリアし、使用量情報を検証する

```
# su fsadmin
$ scatefs_quotacheck -c -g scatefs00
```

9.1.1.2　scatefs_edquota コマンド

scatefs_edquotaコマンドでは、ユーザやグループ、ディレクトリにQUOTA設定を行う機能を提供します。本コマンドは、rootユーザのみが実行でき、リモートCLIコマンド(9.8 リモートCLI)経由にて使用することが可能です。

(例) IOサーバ上で、ファイルシステムscatefs00のユーザ(UID 500) に対し、QUOTAを編集する(環境変数 EDITORで設定したエディタを開きます)

```
# su fsadmin
$ export EDITOR=/bin/vi
$ scatefs_edquota -u 500 scatefs00
```

(例) IOサーバ上で、ファイルシステムscatefs00のユーザ(UID 500)に対し、ディスク容量のソフトリミットを1000KB、ハードリミットを2000KBに設定する

```
# su fsadmin
$ scatefs_edquota -u 500 -b 1000:2000 scatefs00
```

(例) IOサーバ上で、ファイルシステムscatefs00のディレクトリ“/dquota”に対し、ファイル数のソフトリミットを5000ファイル、ハードリミットを10000ファイルに設定する

```
# su fsadmin
$ scatefs_edquota -d /dquota -i 5000:10000 scatefs00
```

(例) Linuxクライアント上で、IOサーバserver00のファイルシステムscatefs00のグループ(GID 500)に対し、ハードリミット、ソフトリミットを設定する

```
$# scatefs_rcli server00 edquota -g 500 -b 1000:2000 -i 5000:10000 scatefs00
```

また、edquotaコマンドでは、ソフトリミット超過にともない設定される猶予期間に関して、以下の設定を行う機能を提供します。

- 各ユーザやグループ、ディレクトリの残り猶予期間(grace time)
- 各ファイルシステムやストレージグループに属するすべてのユーザやグループ、ディレクト

リがソフトリミット超過時に初期設定される猶予期間(period time)

(例) IOサーバ上でファイルシステムscatefs00のユーザ(UID 500) に対し、残り猶予期間を編集する(環境変数 EDITORで設定したエディタを開きます)

```
# su fsadmin
$ export EDITOR=/bin/vi
$ scatefs_edquota -T -u 500 scatefs00

Times to enforce softlimit for (user 0):
Time units may be: days, hours, minutes, or seconds
            Filesystem   block grace   inode grace
             scatefs00   3550seconds        unset
```

(例) IOサーバ上で、ファイルシステムscatefs00のユーザ(UID 500)に対し、ディスク容量の残り猶予期間を7日(604800秒)に設定する

```
$ scatefs_edquota -T -u 500 -b 604800 scatefs00
```

(例) Linuxクライアント上で、IOサーバserver00のファイルシステムscatefs00のユーザ(UID 500) に対し、ファイル数の残り猶予期間を1時間(3600秒)に設定する

```
$ scatefs_rcli server00 edquota -T -u 500 -i 3600 scatefs00
```

(例) IOサーバ上で、ファイルシステムscatefs00のユーザに初期設定される猶予期間を編集する(環境変数 EDITORで設定したエディタを開きます)

```
$ export EDITOR=/bin/vi
$ scatefs_edquota -t u scatefs00

Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
            Filesystem   block grace period   inode grace period
             scatefs00              7days         3600seconds
```

(例) IOサーバ上でファイルシステムscatefs00のグループに初期設定されるディスク容量の猶予期間を1日(86400秒)に設定する

```
$ scatefs_edquota -t g -b 86400 scatefs00
```

(例) Linuxクライアント上でIOサーバserver00のファイルシステムscatefs00のディレクトリに初期設定されるファイル数の猶予期間を10000秒に設定する

```
$ scatefs_rcli server00 edquota -t d -i 10000 scatefs00
```

9.1.1.3　scatefs_quota コマンド

scatefs_ quotaコマンドは、ファイルシステムのQUOTA情報を表示する機能を提供します。本コマンドは、管理者および一般ユーザが実行でき、リモートCLIコマンド(9.8 リモートCLI)経由にて使用することが可能です。一般ユーザは、リモートCLIコマンドを使用して、自身または所属しているグループ、およびディレクトリのQUOTA情報を確認することが可能です。

正確な情報の出力が必要な場合には、事前にscatefs_quotacheckコマンド(9.1.1.1 scatefs_quotacheckコマンド)を実行してください。

(例) IOサーバ上で、ファイルシステムscatefs00のユーザ (UID 500)のQUOTA情報を表示する

```
# su fsadmin
$ scatefs_quota -s -u 500 scatefs00:sg000

ScaTeFS quotas for user  (uid 500)
       Filesystem:sgname         blocks          quota          limit  grace
files    quota    limit  grace
 ------------------------------------------------------------------------
------------------------------------
         scatefs00:ROOT          0       488.2K         9.5M     -         0
10.0K    20.0K     -
```

(例) IO サーバ上で、ファイルシステム scatefs00 のディレクトリ“qdir”(DIRID 1000)のQUOTA 情報を出力する

```
# su fsadmin
$ scatefs_quota -s -d qdir scatefs00

ScaTeFS quotas for directory /qdir (dirid 1000)
       Filesystem:sgname         blocks          quota          limit  grace
files    quota    limit  grace
 ------------------------------------------------------------------------
```

```
--------------------------------------
         scatefs00:ROOT              0         488.2K           9.5M       -
0     10.0K     20.0K      -
```

(例) Linuxクライアント上で、IOサーバserver00のファイルシステムscatefs00のグループ”group500”(GID 500)を対象としたQUOTA情報を出力する

```
$ scatefs_rcli server00 quota -s -g group500 scatefs00
ScaTeFS quotas for group group500 (gid 500)
       Filesystem:sgname         blocks          quota          limit   grace
files     quota     limit  grace
 ---------------------------------------------------------------------------
--------------------------------------
         scatefs00:ROOT              0          7.63G          9.54G       -
0     10.0K     1.00M      -
```

9.1.1.4　scatefs_repquota コマンド

scatefs_repquota コマンドは、ファイルシステムの QUOTA 情報一覧を表示する機能を提供します。本コマンドは管理者のみが実行でき、リモート CLI コマンド(9.8 リモート CLI)経由にて使用することが可能です。表示される QUOTA 情報は、未使用のユーザやグループ、ディレクトリの QUOTA 情報は出力されません。
正確な情報の出力が必要な場合には、事前に scatefs_quotacheck コマンド(9.1.1.1 scatefs_quotacheck コマンド)を実行してください。

(例) IO サーバ上で、ファイルシステム scatefs00 のユーザの QUOTA 情報一覧を出力する

```
# su fsadmin
$ scatefs_repquota -u scatefs00

*** Report for user quotas on scatefs00:ROOT
Block grace time: 7days; Inode grace time: 7days
          Block limits                                File limits
  user(id)         used         soft         hard  grace      used      soft
hard  grace
 ---------------------------------------------------------------------------
------------------------
        0            0        32768        65536      -         0     10000
```

```
10000      -
       512             0         32768         65536       -           0      20000
30000      -
      1024             0         32768         65536       -           0      50000
60000      -
      2048        225416        524288       1048576       -         729        512
1024  6days
```

(例) IOサーバ上で、ファイルシステムscatefs00のディレクトリのQUOTA情報一覧を出力する

```
# su fsadmin
$ scatefs_repquota -d scatefs00

*** Report for directory quotas on scatefs00:ROOT
Block grace time: 7days; Inode grace time: 7days
                          Block limits                                File limits
   directory(name)     used           soft           hard    grace        used
soft      hard   grace
 ----------------------------------------------------------------------------
----------------------------
  qdir00            32768       2097152       4194304       -        750       500
1000   6days
  qdir01            65536       2097152       4194304       -        256       500
1000       -
  qdir02             1048576        2097152        4194304        -         128
500        0       -
  qdir03           524288       2097152       4194304       -        300       500
0       -
```

(例) Linux クライアント上で、IO サーバ server00 にのファイルシステム scatefs00 のグループを対象とした QUOTA 情報一覧を表示する

```
# scatefs_rcli server00 repquota -g scatefs00
(出力イメージ省略)
```

また、scatefs_repquota コマンドでは、設定されているハードリミット、ソフトリミットを再設定が可能な形式でバックアップする機能を提供します。バックアップは、標準出力表示およびファイル作成にて行います。バックアップ機能は IO サーバ上でのみ実行可能です。

(例) IO サーバ上で、ファイルシステム scatefs01 のユーザのバックアップ内容を一覧で出力した後、同情報のバックアップを出力する

```
# su fsadmin
$ scatefs_repquota -u -b scatefs01
/opt/scatefs/bin/scatefs_edquota -t u      -b 604800           -i 604800
scatefs01:SG1 || echo "error: user grace scatefs01"
/opt/scatefs/bin/scatefs_edquota  -u  1024  -b  102400:204800  -i  128:256
scatefs01:SG1 || echo "error: uid 1024  scatefs01"
/opt/scatefs/bin/scatefs_edquota  -u  2048  -b  102400:204800  -i  128:256
scatefs01:SG1 || echo "error: uid 2048  scatefs01"
/opt/scatefs/bin/scatefs_edquota  -u  3072  -b  102400:204800  -i  128:256
scatefs01:SG1 || echo "error: uid 3072  scatefs01"
$ ls -l
-rw-rw-r--    1    root    fsadmin    630       9  月      18    16:58    2014
scatefs_quota.fsid1.sgid1.user
```

(例) IO サーバ上で、ファイルシステム scatefs01 のユーザのソフトリミット、ハードリミットをバックアップファイルからリストアする

```
# su fsadmin
$ ls -l
-rw-rw-r--    1    root    fsadmin    630       9  月      18    16:58    2014
scatefs_quota.fsid1.sgid1.user
$ sh ./scatefs_quota.fsid1.sgid1.user
```

9.1.1.5　scatefs_mkqdir コマンド

scatefs_mkqdir コマンドは、ファイルシステムの QUOTA 設定が可能なディレクトリを作成する機能を提供します。本コマンドは、IO サーバ上でのみ実行可能です。QUOTA 情報は作成したディレクトリ毎に管理し、ディレクトリおよび配下の使用量のカウントと、ハードリミット、ソフトリミット、残り猶予時間の設定に対応します。このコマンドで作成したディレクトリを削除する場合は、scatefs_rmqdir コマンド(9.1.1.6 scatefs_rmqdir コマンド)を使用する必要があります。

(例) IO サーバ上で、ファイルシステム scatefs00 のルートディレクトリ配下に QUOTA 制御ディレクトリ”dquota00”を作成する

```
# su fsadmin
$ scatefs_mkqdir scatefs00 /dquota00
```

(例) IO サーバ上で、ファイルシステム ID 1 のディレクトリ”work”配下に QUOTA の設定を行うディレクトリ”dquota01”を作成する

```
# su fsadmin
$ scatefs_mkqdir 1 /work/dquota01
```

9.1.1.6　scatefs_rmqdir コマンド

scatefs_rmqdir コマンドでは、ファイルシステムの QUOTA 設定が可能なディレクトリを削除する機能を提供します。本コマンドは、IO サーバ上でのみ実行可能です。

(例) IO サーバ上で、ファイルシステム scatefs00 の QUOTA 制御ディレクトリ”dquota00”を削除する

```
# su fsadmin
$ scatefs_rmqdir scatefs00 /dquota00
```

(例) IO サーバ上で、ファイルシステム ID 1 の QUOTA 制御ディレクトリ”/dquota/dquota01”を削除する

```
# su fsadmin
$ scatefs_rmqdir 1 /work/dquota01
```

9.2　レコードロック強制解除

ScaTeFSではPOSIX.1で定義されている標準的なレコードロックを行う機構を提供しています。通常は、特定の計算ノードが資源を排他利用する場合にレコードロックを行い、利用終了と同時にレコードロックを解除します。しかし、レコードロック中の計算ノードに障害が発生した場合、運用によっては当該ノードからレコードロックの解除が長期にわたり実施できない場合があります。このため、特定の計算ノードのレコードロック情報をすべて強制解除する機能をscatefs_lockreleaseコマンドとして提供します。

9.3　ファイルシステムの拡張

IOサーバやIOターゲットを追加することで、ファイルシステムを拡張することが可能です。拡張時はファイルシステムの運用を停止する必要があります。以下に拡張の手順を記載します。

(1)　すべてのクライアントからファイルシステムをアンマウントします。

(2)　IO サーバの構築」に記載されている手順で、新たに追加する IO サーバや IO ターゲットをシステムに追加します。

(3)　すべての IO サーバで IO サーバデーモンを停止します。

(4)　scatefs_extendfs コマンドで、システムに追加された IO ターゲットを拡張したいファイルシステムに追加します。
拡張するファイルシステム、追加するIOターゲットを定義したファイルを用意します。

```
-bash-4.1$ cat datafile
#拡張したいファイルシステムのファイルシステムID
fsid          0
#ファイルシステムに追加するIOターゲットID
addiotid       1
```

このファイルを引数にscatefs_extendfsコマンドを実行します。

```
# su - fsadmin
-bash-4.1$ scatefs_extendfs -f datafile
```

(5)　拡張したファイルシステム ID を指定し、ファイルシステムの整合をとります。
(例) ファイルシステムIDが0の場合

```
$ scatefs_fsck 0
```

(6)　すべての IO サーバで IO サーバデーモンを起動します。

(7)　拡張したファイルシステム名を指定し、QUOTA 情報の整合をとります。

(例) 拡張したファイルシステム名が、scatefs00の場合

```
$ scatefs_quotacheck scatefs00
```

9.4　フェアシェア

IOサーバでは、フェアシェアIOスケジューリング機能を提供します。この機能は、従来のジョブスケジューリングではなく、IOサーバ上のIOリソースのフェアシェアを実現します。

この機能を利用することによって効率的な負荷分散が行われ、特定のユーザ、または特定の計算ノードの処理負荷によるシステム全体のパフォーマンス低下を低減します。

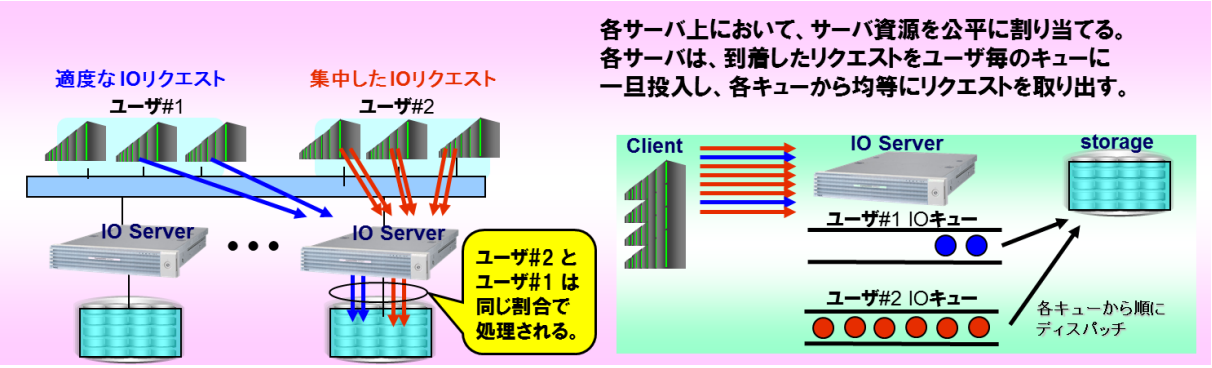


図 9-2　フェアシェアのイメージ図

IOサーバのコンフィグファイルに所定の情報を登録することで利用が可能となります。ただし、運用中の動的な変更はサポートしていません。

9.4.1　ポリシーの種類

IOスケジューリング機能は以下の3つのポリシーから選択可能とします。

- フェアシェアなし(デフォルト)
- ユーザ(UID)ごとの均等化
- ClientID(クライアント毎にユニークなID)ごとの均等化

ポリシーは全IOサーバで同一のものとする必要があり、ポリシーを変更後はIOサーバの再起動が必要となります。

9.4.2　ポリシーの変更方法

ポリシーを変更する際の手順は下記になります。

(1)　コンフィグファイル scatefssrv.conf の FAIRPOLICY を変更します。

設定可能な値は以下となります。

0：フェアシェアなし(デフォルト)

1：UIDごとの均等化

2：ClientIDごとの均等化

(2)　scatefs_admin コマンドを使用して、修正した scatefssrv.conf を全 IO サーバに配布します。

(3)　各 IO サーバを再起動します。

9.5　ストレージグループ

IOサーバでは、ファイルシステムを構成する複数のIOターゲットをグループに分けて管理する機能を提供します。このストレージグループは、ファイルシステムのディレクトリと対応づけられ、QUOTAの管理単位ともなります。このグループをストレージグループと呼びます。

この機能を利用することによって、たとえば低速なディスクと高速なディスクをグループ分けすることにより、データ特性による使い分けや課金が可能となります。

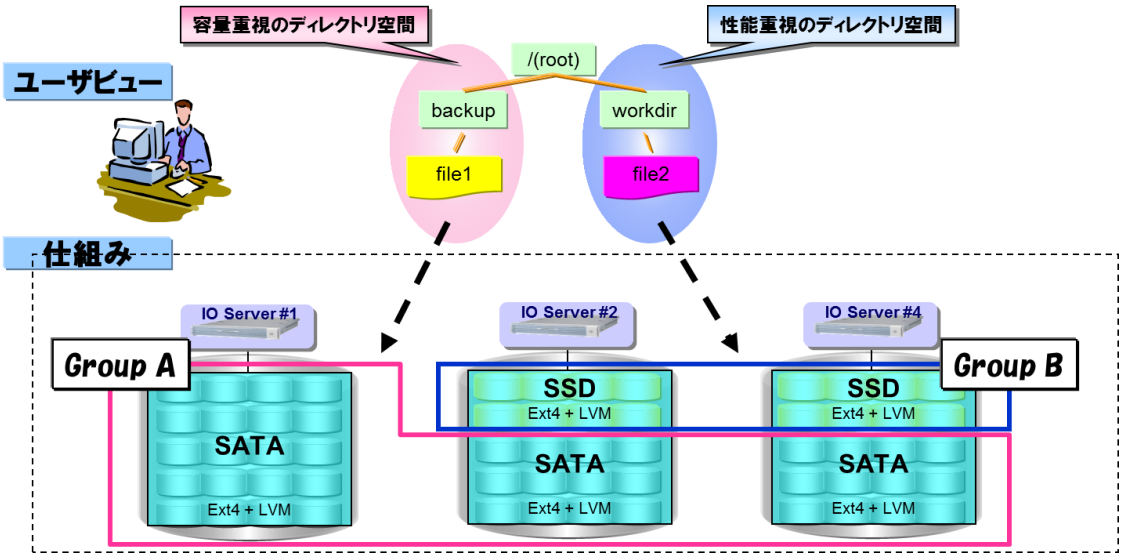


図 9-3　ストレージグループの概念図

設定は、いずれかのIOサーバにログインしScaTeFS用コマンドを実行します。初めに、scatefs_extendfsコマンドで特定のファイルシステムにストレージグループを追加します。次に、scatefs_mksgdirコマンドで先ほど登録したストレージグループと、特定のディレクトリを対応付けます。scatefs_extendfsによるストレージグループの追加は、運用を停止してから行う必要があります。scatefs_mksgdirによるストレージグループとディレクトリの対応付けは、IOサーバデーモンを起動してから行う必要があります。以下に、ストレージグループ作成の手順を記載します。

(1)　すべてのクライアントからファイルシステムをアンマウントします。

(2)　「IO サーバの構築」に記載されている手順で、新たに追加する IO サーバや IO ターゲットをシステムに追加します。

(3) すべての IO サーバで IO サーバデーモンを停止します。

(4) scatefs_extendfs コマンドで、システムに追加された IO ターゲットを指定してストレージグループを作成します。

拡張するファイルシステム、追加するIOターゲットを定義したファイルを用意します。

```
-bash-4.1$ cat datafile
#新たに追加するストレージグループをファイルシステム名：ストレージグループ名で指定する。
name            scatefs00:sgA
#ストレージグループに設定するIOターゲットID。
#ファイルシステムに登録されていない必要がある。
iotid           1
```

このファイルを引数にscatefs_extendfsコマンドを実行します。

```
# su - fsadmin
-bash-4.1$ scatefs_extendfs -f datafile -addsg
```

(5) ストレージグループを追加したファイルシステム ID を指定し、ファイルシステムの整合をとります。

(例) ファイルシステムIDが0の場合

```
   $ scatefs_fsck 0
```

(6) IO サーバの起動確認

すべての IO サーバが起動していることを確認してください。確認には、以下のコマンドを使用します。

```
$ scatefs_admin --check system
IOSID  CONFIGFILE            MD5SUM
   0  system.info           xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
   1  system.info           xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

エラーメッセージが出力されないことを確認してください。

(7) すべての IO サーバで IO サーバデーモンを起動します。

(8) scatefs_mksgdir でストレージグループとディレクトリの対応付けを行います。

(例)

```
$ scatefs_mksgdir scatefs00 sgA /sgAdir
```

マウントポイントが「/mnt/scatefs」の場合、sgAに属したディレクトリとして「/mnt/scatefs/sgAdir」
が作成されます。
※scatefs_mksgdirで作成したディレクトリを削除する場合は、scatefs_rmsgdirで削除してください。
(例) 作成したディレクトリを削除する

```
$ scatefs_rmsgdir /sgAdir
```

(9) ストレージグループを追加したファイルシステム名を指定し、scatefs_quotacheck コマンド(9.1.1.1 scatefs_quotacheck コマンド)を実行して QUOTA 情報の整合をとります。
(例) 拡張したファイルシステム名が、scatefs00の場合

```
$ scatefs_quotacheck scatefs00
```

9.6　容量管理

ScaTeFSでは、容量が閾値を超えるIOターゲットへの書き込み要求を受けた場合、容量に十分空きがある他のIOターゲットを選定し利用することでIOを継続します。しかし、本機能は通常処理コストが高いため動作しないことが望ましい状態と言えます。このような状態となった場合、システム負荷が低い状態において、ファイルシステムのリバランス実施の検討が必要です。

9.7　リバランス

ファイルシステムを拡張すると、既存ファイルと新規ファイルへのアクセスに偏りが発生することがあります。ScaTeFSでは、この偏りを解消し、全IOサーバ分の帯域を活用する、リバランス機能を提供します。リバランス機能は、ファイルシステムの運用を停止することなく実施できます。

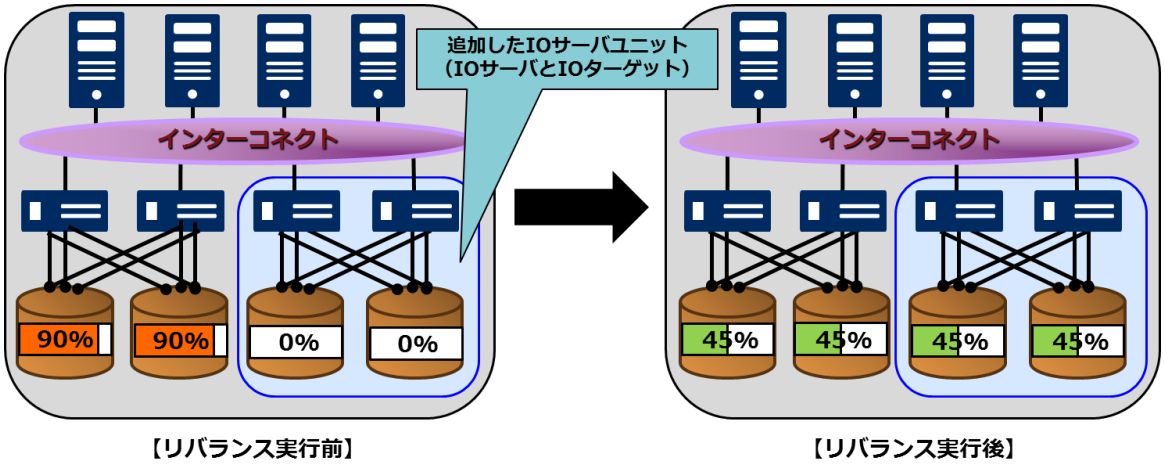


図 9-4　IO サーバユニットを追加した時のリバランスの実行例

リバランスは、以下の手順で実施します。

(1) リバランス対象ファイルの抽出
(2) リバランス対象ファイルのマイグレーション
(3) 抽出結果のクリア
(4) マイグレーション情報のクリア（メンテナンス時に実施）

(1) リバランス対象ファイルの抽出

IOサーバでscatefs_rebalanceコマンドを使用し、リバランス対象ファイルを抽出します。抽出の完了は、レポート機能でも確認できます。

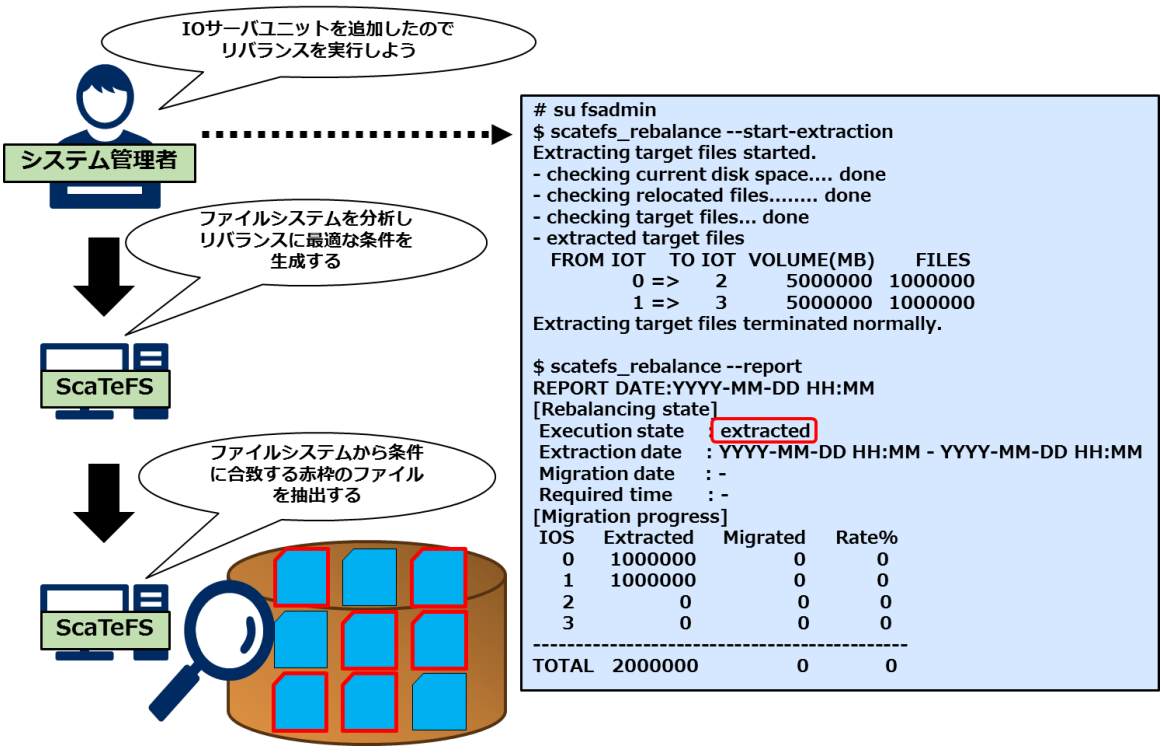


図 9-5　リバランス対象ファイル抽出の実行例

抽出し直す場合は、抽出結果をクリアしてから再度抽出を実施します。

```
# su fsadmin
$ scatefs_rebalance --clear
scatefs_rebalance: rebalance information was cleared.
$ scatefs_rebalance --start-extraction
Extracting target files started.
…
```

また、Linuxクライアントでscatefs_rebalance_importコマンドを使用して、リバランス対象ファイルを指定することもできます。

(2) リバランス対象ファイルのマイグレーション

リバランス対象ファイルの抽出が完了したら、IOサーバでscatefs_rebalanceコマンドを使用し、マイグレーションサービスを開始します。これにより、対象ファイルがマイグレーションされます。マイグレーションの状況は、レポート機能で確認します。

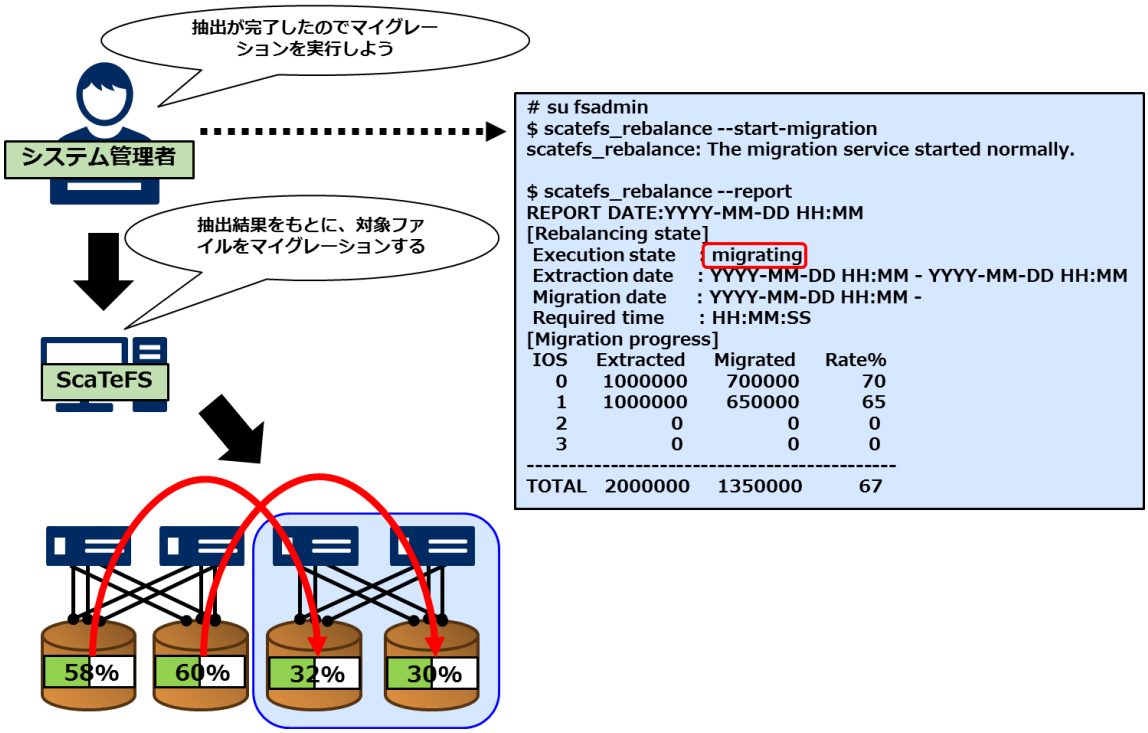


図 9-6　リバランス対象ファイルのマイグレーション実行例

マイグレーションが完了したら、マイグレーションサービスを停止します。

```
# su fsadmin
$ scatefs_rebalance --report
REPORT DATE:YYYY-MM-DD HH:MM
[Rebalancing state]
 Execution state    : migrated
 Extraction date    : YYYY-MM-DD HH:MM - YYYY-MM-DD HH:MM
 Migration date     : YYYY-MM-DD HH:MM - YYYY-MM-DD HH:MM
 Required time      : HH:MM:SS
[Migration progress]
 IOS    Extracted    Migrated    Rate%
   0    1000000      1000000       100
   1    1000000      1000000       100
   2          0            0         0
   3          0            0         0
-----------------------------------------------
TOTAL   2000000      2000000       100

$ scatefs_rebalance --stop-migration
scatefs_rebalance: The migration service stopped normally.
```

マイグレーション中でも、必要に応じマイグレーションサービスを一時停止と再開ができます。

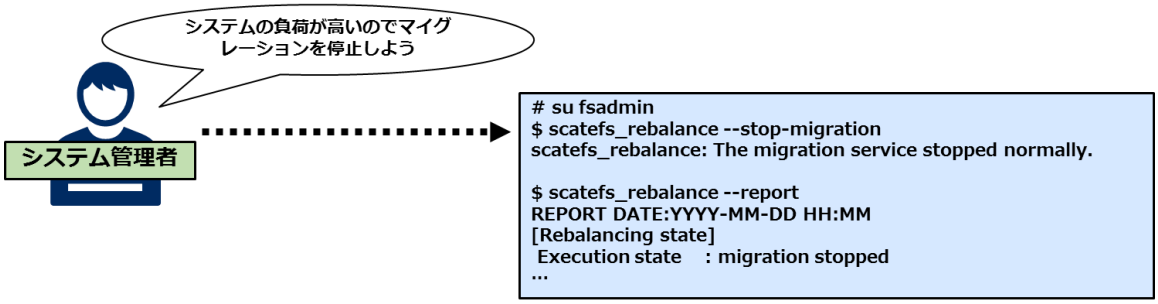


図 9-7　マイグレーションサービスの一時停止の実行例

なお、マイグレーションサービスの停止は一時的に受け付けられない場合があります。その場合は、再度コマンドを実行してください。

(3) 抽出結果のクリア

マイグレーションが完了したら、IOサーバでscatefs_rebalanceコマンドを使用し、抽出した情報をクリアします。

```
# su fsadmin
$ scatefs_rebalance --clear
scatefs_rebalance: rebalance information was cleared.
```

以上でリバランス作業は完了です。

(4) マイグレーション情報のクリア（メンテナンス時に実施）

マイグレーションが終わり、すべてのクライアントのアンマウント（※）を確認したら、IOサーバでscatefs_migrateコマンドを使用し、マイグレーション情報をクリアします。マウント中のクライアントが存在する状況下では実施しないでください。

```
# su fsadmin
$ scatefs_migrate --clear
scatefs_migrate: The migration information was cleared.
```

（※）ScaTeFSクライアント上でScaTeFSのファイルシステムをNFSでエクスポートしている場合、
そのファイルシステムをマウントしているすべてのNFSクライアントからそのファイルシステムを
アンマウントしてください。次に、ScaTeFSクライアントでnfsサービスを停止してください。

マウント中のクライアントが存在しない状況でクリアできない場合は、フォースを指定します。

```
# su fsadmin
$ scatefs_migrate --clear
scatefs_migrate: cannot clear.
$ scatefs_migrate --clear --force
scatefs_migrate: The migration information was cleared forcibly.
```

9.8　リモートCLI

IOサーバ上に配置された一部のコマンドをクライアントから実行する仕組みとして、リモートCLI(scatefs_rcli)を提供します。scatefs_rcliで実行可能なサブコマンドは以下のとおりです。

表 9-2　リモート CLI のサブコマンド

サブコマンド名	概要	実行ユーザ制限
df	ScaTeFSの使用状況表示	なし
detail	ScaTeFSの構成情報表示	特権ユーザのみ実行可能
logcollect	IOサーバのログ表示	特権ユーザのみ実行可能
quota	ScaTeFSのquota情報表示	なし
repquota	ScaTeFSのquota情報の一覧表示	特権ユーザのみ実行可能
edquota	ScaTeFSのユーザおよびグループquotaの編集	特権ユーザのみ実行可能
ifstat	IOサーバのインターフェース状態表示	特権ユーザのみ実行可能
mkqdir	ScaTeFSのQUOTA対応ディレクトリの作成	特権ユーザのみ実行可能
rmqdir	ScaTeFSのQUOTA対応ディレクトリの削除	特権ユーザのみ実行可能

9.8.1　特権ユーザ

root以外の特定ユーザに、リモートCLIを実行する上での特権を持たせる場合は、fsadminグループに所属させる必要があります。fsadminグループに所属するユーザは、リモートCLIを実行する上での特権ユーザとなります。

(例)

```
・fsadminグループを追加
# groupadd fsadmin

・fooユーザが所属するグループにfsadminを追加
# usermod foo -G xxx,yyy,fsadmin
  ※xxx,yyy は既に所属しているグループ
```

9.8.2　リモート CLI ユーザの登録

クライアントからscatefs_rcliを使用するためには、IOサーバでの登録が必要になります。

ユーザの登録はscatefs_rcliadmコマンドで行います。登録後、9.8.3の例を参照し、動作確認を実施してください。

(例)

- clientAの foo ユーザを登録

```
$ scatefs_rcliadm add clientA foo
```

- 確認

```
$ scatefs_rcliadm info
clientA foo
```

- clientAのfooユーザを削除

```
$ scatefs_rcliadm delete clientA foo
```

9.8.3　リモート CLI の実行

scatefs_rcliadmで登録されたユーザは、scatefs_rcliコマンドを実行することができます。

(例)

- clientAの foo ユーザがserverB の FSID#0を指定しdfサブコマンドを実行

```
$ scatefs_rcli serverB df 0
   IOT IOS SGID    1K-blocks         Used     Available  Use%  Mounted on
   0   0    0      11867221        305180      10974464    3%  /mnt/iot/0
   2   1    0      11867221        305180      10974464    3%  /mnt/iot/2
   1   0    0      11867213        305180      10974457    3%  /mnt/iot/1
   3   1    0      11867213        305180      10974457    3%  /mnt/iot/3
   ------------------------------------------------------------------------
---
   TOTAL          47468868       1220720      43897842    3%
```

- 登録されていないユーザで実行した場合

```
$ scatefs_rcli serverB df scatefs
Permission denied.
scatefs_rcli: df to serverB failed
```

9.9　情報表示

システムを構成する様々な情報を取得するインターフェースをIOサーバ上に配置されたコ

マンドとして提供します。

- scatefs_df

  ファイルシステム使用状況表示

  (例)ファイルシステムのディスク使用状況

```
$ scatefs_df scatefs00
 IOT IOS SGID     1K-blocks          Used      Available  Use%  Mounted on
   0   0    0   14276233588     8482274292     5492881741   60%  /mnt/iot/0
   3   1    0   14276233588     8488604028     5486868491   60%  /mnt/iot/3
   1   0    0   14276233588     8471883748     5502752757   60%  /mnt/iot/1
   4   1    0   14276233588     8461444560     5512669986   60%  /mnt/iot/4
   2   0    0   14276233588     8471705888     5502921724   60%  /mnt/iot/2
   5   1    0   14276233588     8471343548     5503265947   60%  /mnt/iot/5
 ------------------------------------------------------------------------------
 TOTAL          85657401528    50847256064    33001360646   60%
```

  (例)ファイルシステムのinode使用状況

```
$ scatefs_df scatefs00 -i
 IOT IOS SGID      Inodes       IUsed       IFree  IUse%   Mounted on
   0   0    0    32527525      816564    31710961     3%   /mnt/iot/0
   3   1    0    32527531      816625    31710906     3%   /mnt/iot/3
   1   0    0    32527531      816671    31710860     3%   /mnt/iot/1
   4   1    0    32527531      816755    31710776     3%   /mnt/iot/4
   2   0    0    32527531      816573    31710958     3%   /mnt/iot/2
   5   1    0    32527531      816734    31710797     3%   /mnt/iot/5
 ------------------------------------------------------------------------------
 TOTAL          195165180     4899922    190265258     3%
```

  (例)ストレージグループのディスク使用状況

```
$ scatefs_df -g scatefs00
 SGID     1K-blocks          Used      Available  Use%
    0      47093604        897648       43848570    2%
```

  (例)ストレージグループのinode使用状況

```
$ scatefs_df -g -i scatefs00
 SGID      Inodes       IUsed       IFree  IUse%
    0     4225772          12     4225760     0%
```

- scatefs_detail

  ファイルシステムの構成情報表示

  (例) ファイルシステム全体

```
$ scatefs_detail -f 0
display detail FS#0
 FS Name       =>     scatefs00
 Root IOS      =>     IOS#0(IOT#0)
  IP           =>     10.0.0.1
  FIP          =>     10.0.1.1 10.0.2.1
PCI-ID@PORT    =>     0000:83:00.1@1
  INIP         =>     10.0.3.1
 Number of IOS  =>     2
 Number of IOT  =>     6 / 1024
 Number of SG   =>     1 / 8
 Data FS type   =>     ext4
 Ctrl FS type   =>     ext4
 Version       =>     0x00010000
 IOTs          =>     0 3 1 4 2 5
 SG            =>     ROOT
```

(例)IOS単位で表示

```
-bash-4.1$ scatefs_detail -s 0
display detail IOS#0
 IP ADDRESS           =>     10.0.0.1
 Floating IP ADDRESS    =>     10.0.1.1 10.0.2.1
PCI-ID@PORT           =>     0000:83:00.1@1
 Inner IP ADDRESS      =>     10.0.3.1
 PORT for Client       =>     50000
 PORT for Server       =>     50001
 PORT for Client Data   =>     50002
 Defined IOTs          =>     0 1 2
 Defined FS            =>     0
```

(例)IOT単位で表示

```
$ scatefs_detail -t 0
display detail IOT#0
 defined server =>     IOS#0
 filesystem    =>     scatefs00
 storagegroup   =>     ROOT
 data device    =>     /dev/vg_data01/lv_data01
 ctrl device    =>     /dev/vg_ctrl01/lv_ctrl01
```

- scatefs_statcollect

  IOサーバの統計情報の表示

  (例) 全IOSの統計情報を表示

```
$ scatefs_statcollect -a
[IOS#0]
:
[IOS#1]
:
```

  (例)IOサーバID＃0のプロシージャの統計情報を表示

```
$ scatefs_statcollect -n 0 -p
[IOS#0]
:
```

  (例)IOサーバID＃1の関数の統計情報を表示

```
$ scatefs_statcollect -n 1 -f
[IOS#1]
:
```

- scatefs_logcollect

  IOサーバのログ表示

  ※ログをファイルに保存する場合は、リダイレクトしてください。

  (例) 全IOサーバのログを表示

```
$ scatefs_logcollect -a
  ※結果を保存する場合
$ scatefs_logcollect -a > ioserver.log
```

  (例)IOサーバの全ログを表示 (ローテートされたファイル、gz形式で圧縮されたファイルを含める)

```
  $ scatefs_logcollect -a -m
```

  (例) IOサーバID＃0のログを表示

```
  $ scatefs_logcollect -n 0
```

  (例)IOサーバID＃1と＃2のログを表示

```
$ scatefs_logcollect -n 1,2
```

9.10 システムファイルの管理

IOサーバのシステムファイルを管理するコマンドとして、scatefs_adminを提供します。scatefs_adminでは/etc/scatefs配下の各システムファイルをIOサーバ間で一致しているかのチェック、指定したIOサーバへの転送/ロールバック、チューニングパラメータファイルの作成などが可能です。コマンドの詳細はIOサーバのmanデータを参照してください。

(例)ScaTeFSの情報ファイル(system.info)がIOS間で一致しているか確認

```
$ scatefs_admin --check all system
```

(例)IOサーバデーモンのチューニングパラメータ設定ファイル(scatefssrv.conf)のデフォルトを作成

```
$ scatefs_admin --create tune
```

(例)IOサーバデーモンのチューニングパラメータ設定ファイル(scatefssrv.conf)を全IOサーバに転送

```
$ scatefs_admin --trans all tune
```

9.11 ファイルシステムの監視

ファイルシステムの統計情報をリアルタイムに収集しモニタリングする機能を提供します。必要となるソフトウェアをインストールおよび設定し、パッケージに同梱されているテンプレートをインポートすることにより、GUIベースでのファイルシステムのモニタリングが可能となります。

以下の統計情報をサポートします。

表 9-3　統計情報

ソース	統計情報
IOサーバ	ファイルシステム、IOサーバ、ユーザIDごとのread/writeのスループットやメタデータオペレーション性能データ

ソース	統計情報
	IOサーバごとのネットワークトラフィックやCPU情報 ファイルシステムごとの使用量 ファイルシステムごとのプロファイル情報（ディレクト内のファイル数やファイルサイズごとの分布など）※
IOターゲット	IOターゲットごとのread/write数 IOターゲットごとの使用量

※scatefs_profstatコマンド（引数なし）を実行することにより、ファイルシステムごとのプロファイル情報を収集します。監視間隔に合わせてコマンドを実行してください。なお、収集に要する時間はご利用になる環境により異なります。

本機能を構成するソフトウェアとソフトウェアの要件を記載します。

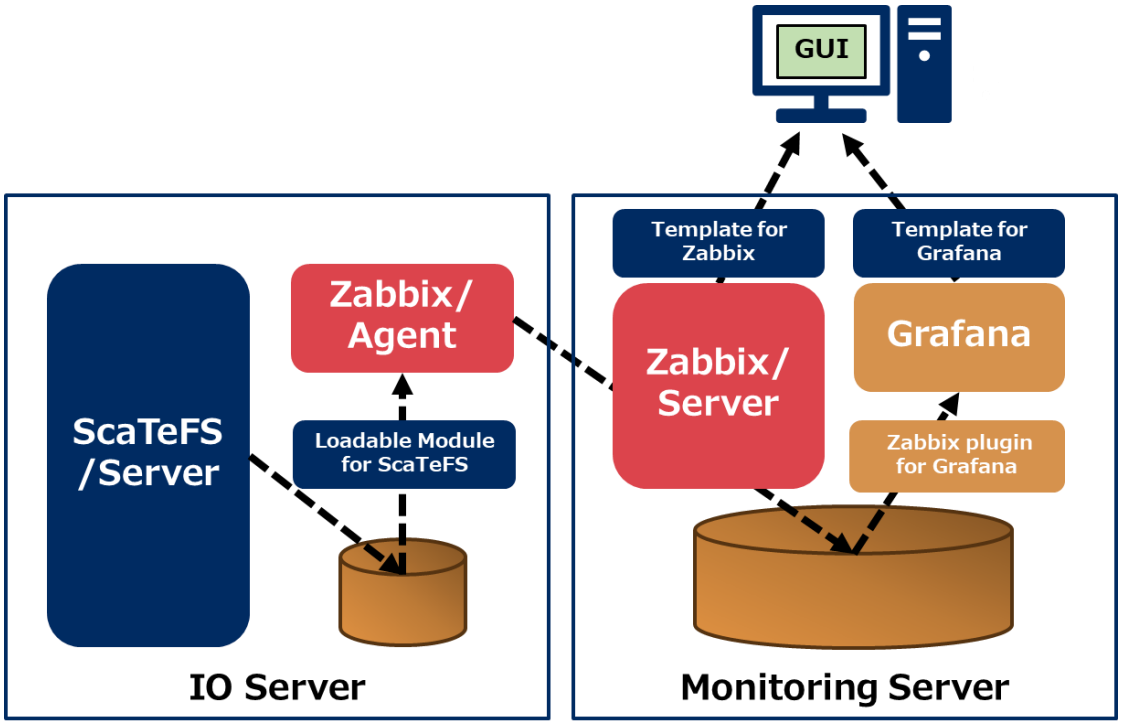


図 9-8　構成図

表 9-4　ソフトウェア

ソフトウェア	バージョン
ScaTeFS/Server	scatefs-srv 3.3以降

ソフトウェア	バージョン
	scatefs-mon 3.3以降 scatefs-monパッケージには下記が同梱されています。 Loadable Module for ScaTeFS、Template for Zabbix、 Template for Grafana
Zabbix/Server	zabbix-server 4.0 LTS以降 動作確認済み：zabbix-server-mysql-4.0.17-2.el7
Zabbix/Agent	zabbix-agent 4.0 LTS 動作確認済み：zabbix-agent-4.0.17-2.el7
Grafana	grafana-6.6以降 動作確認済み：grafana-6.6.1-1
Zabbix plugin for Grafana	v3.11.0以降 動作確認済み：alexanderzobnin-grafana-zabbix-v3.11.0-1-g52f24ec.zip

ScaTeFS/Serverをインストール後、本機能を使用するために必要な設定方法について説明します。なお、ZabbixやGrafanaを使用する上での基本的な設定はコミュニティが提供するドキュメントを参照ください。

- Loadable Module for ScaTeFS
  scatefs-srvパッケージの入手方法と同様に、scatefs-monパッケージを入手しインストールします。
- Zabbix/Agent
  Zabbixコミュニティからソフトウェアを入手しインストールします。
  Loadable Module for ScaTeFSを使用するために、zabbix_agentd.confに以下の設定を追加してください。

```
LoadModulePath=/opt/scatefs/lib/
LoadModule=libscatefszbx.so
UserParameter=scatefs.alive.daemon, pgrep scatefs_server > /dev/null 2>&1; echo $?
```

- Zabbix/Server
  Zabbixコミュニティからソフトウェアを入手しインストールします。テンプレートを使

用するために、以下の設定をしてください。

(1) scatefs-monパッケージでインストールされたZabbix用テンプレートをインポートします。

(2) ファイルシステムを構成するIOサーバを監視対象ホストに登録します。これらのIOサーバは同じホストグループに属するように設定します。

(3) 追加した監視対象ホストに(1)でインストールしたテンプレートを追加します。

(4) 追加した監視対象ホストのマクロに以下を追加します。
マクロ名：{$SCATEFS_HOSTGROUPNAME}
値：”(2)で設定したホストグループ名”（ダブルクォートで括ります）

(5) 「/etc/zabbix/zabbix_server.conf」に以下の設定を追加してください。
IOサーバ1セット（2台）につき、CacheSizeは16MB、TrendCacheSizeは8MBを指定してください。

```
CacheSize=16MB
TrendCacheSize=8MB
```

- GrafanaとZabbix plugin for Grafana
Grafanaコミュニティからソフトウェアを入手しインストールします。テンプレートを使用するために、以下の設定をしてください。
(1) Zabbix plugin for Grafanaを有効にして、データソースを追加します。
(2) scatefs-monパッケージに同梱されているGrafana用テンプレートをインポートします。

テンプレートの内容について説明します。

- Zabbixテンプレート
モニタリングに必要となる監視アイテムを定義しています。また、以下の障害監視トリガを定義しています。
  - ScaTeFS/Serverデーモンの死活監視
    ScaTeFS/Serverデーモンプロセスの有無を監視します。
  - ScaTeFSファイルシステムの使用量監視
    使用量を3つのレベルで監視します。
- Grafanaテンプレート
3つのスクリーンを定義しています。
  - Data screen of ScaTeFS

ファイルシステムやIOサーバごとの、read/writeオペレーションに関する各種統計情報を表示します。

- Metadata screen of ScaTeFS
  ファイルシステムやIOサーバごとの、メタデータオペレーションに関する各種統計情報を表示します。
- ScaTeFS throughput/IO size per UID
  ユーザIDごとの、read/wiiteやメタデータオペレーションに関する各種統計情報を表示します。

9.12 ScaTeFS InfiniBand 高速IOライブラリ

9.12.1 ScaTeFS IB ライブラリ/ScaTeFS VE ダイレクト IB ライブラリの概要

IB使用時に、IB 専用のAPIを使用してユーザ空間で大IOを軽量かつ高速に処理するライブラリです。図 9-9に示すように、プログラム１と３は、このライブラリをリンクすることにより、VHのカーネル空間をバイパスしてユーザ空間から直接IOできるようになります。これにより、プログラム２や４に比べて、大IOを行うアプリケーションの性能向上が期待できます。libcのread/write系システムコールを自動的にライブラリの処理に切り替えるので、アプリケーションの修正は不要です。

ScaTeFS IBライブラリはVHを含むスカラーマシンでプログラム実行時に使用します。ScaTeFS VEダイレクトIBライブラリはVEでプログラム実行時に使用します。

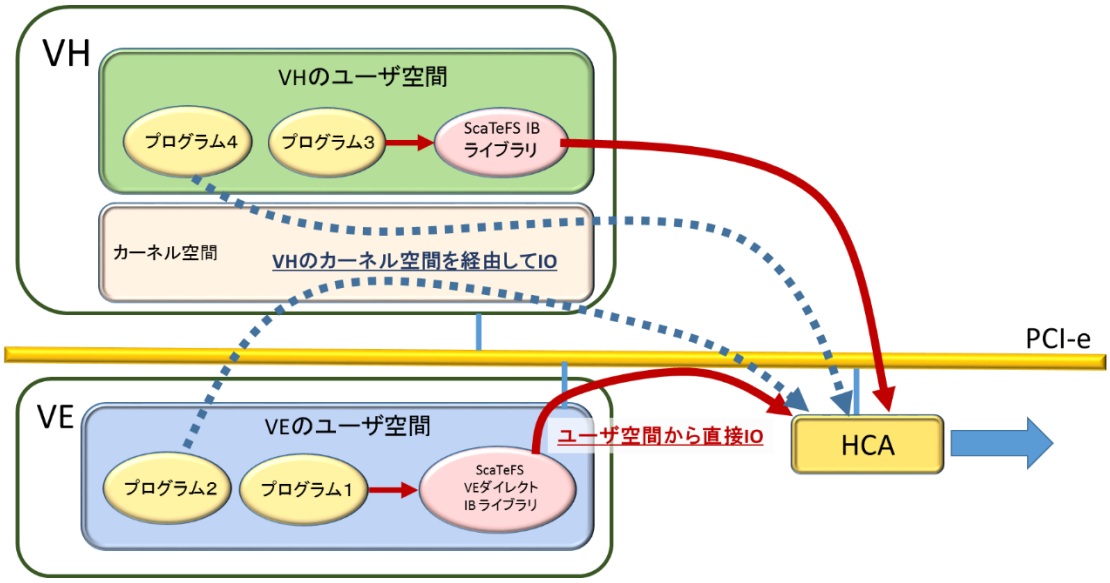


図 9-9　ScaTeFS InfiniBand 高速 IO ライブラリ

以下はScaTeFS IBライブラリとScaTeFS VEダイレクトIBライブラリで共通の閾値、設定の説明です。また、設定方法については6.1.5、使用方法については11.6を参照してください。

9.12.2 IB 専用の API による IO の閾値

read(2)/write(2)系システムコールに指定するIOサイズが1MB以上の場合に、IB 専用のAPIを使用した高速IOを実行します。1MB未満の場合は、従来どおりカーネルによるIOを実行します。

9.12.3 ディスク同期モードの設定

書き込んだデータのディスクへの同期について、以下の２つのモードがあります。各モードは性能とデータ信頼性のトレードオフの関係にあります。デフォルトはclose(2)時ディスク同期モードです。

- close(2)時ディスク同期モード(デフォルト)

  write時にディスク同期は行わずclose時に当該ファイルに対して書き込んだデータをまとめてディスクに同期します。既存のカーネルによるIOとは異なりIOサーバのフェイルオーバが発生した場合に、ジョブは継続運用せずエラーとなります。本エラー発生時は、ライブラリがread(2)/write(2)系システムコールまたはclose(2)のエラーメッセージを標準エラー出力に出力します。ユーザはエラーとなったジョブを再実行する必要があります。また、close時までディスク同期を行わないので下記の「wirte(2)時ディスク同期モード」より高性能となります。図 9-10はclose(2)時ディスク同期モードの動作イメージです。

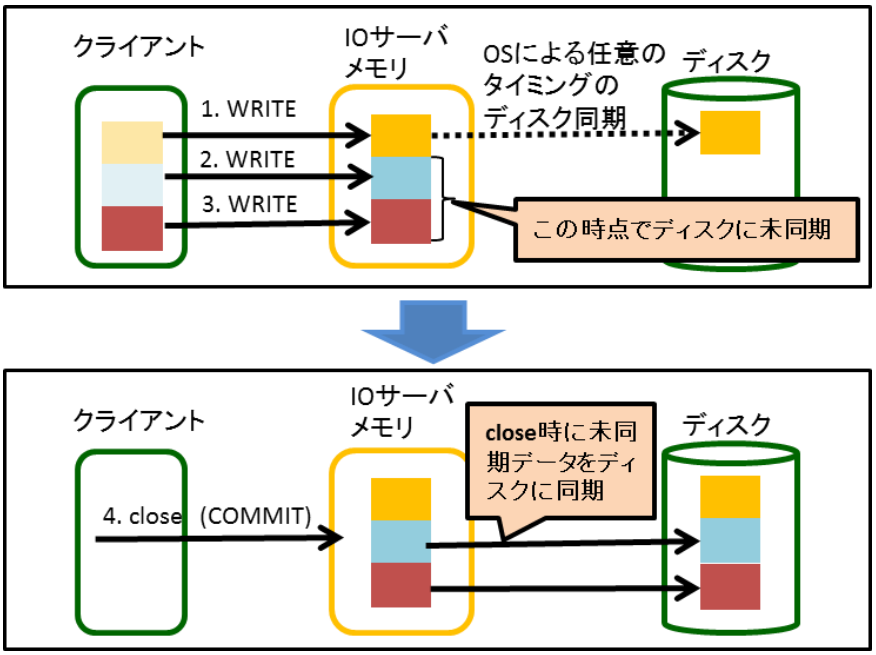


図 9-10　close(2)時ディスク同期モードのイメージ

- wirte(2)時ディスク同期モード

  write時に書き込んだデータをディスクに同期します。既存のカーネルによるIOと同様に、IOサーバのフェイルオーバ時に自動的にジョブの運用が継続します。しかし、write時に毎回ディスク同期を行うためclose(2)時ディスク同期モードに比べてwrite性能が低くなります。図 9-11はwrite(2)時ディスク同期モードの動作イメージです。

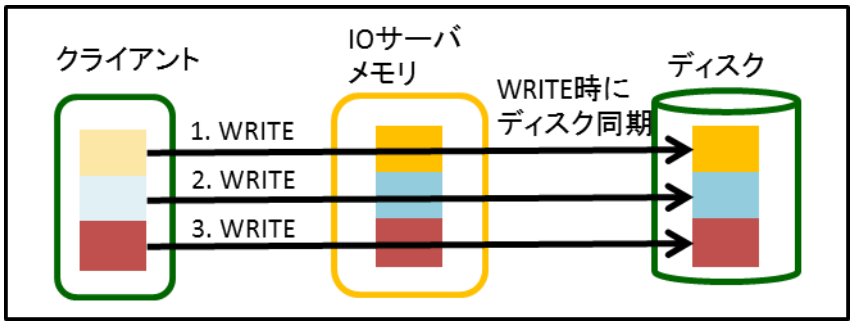


図 9-11　write(2)時ディスク同期モードのイメージ

設定はIOサーバ毎に設定します。設定方法は5.3.2.1を参照してください。

以下に上記2つのディスク同期モードの差異をまとめます。

表 9　ディスク同期モードの差異

モード	write性能	適切な運用ケース	フェイルオーバ発生時のユーザの対応
close(2)　時ディスク同期(デフォルト)	高性能	主に、比較的大きなファイル(128MB以上)に対して、4MB以上のIOサイズでwrite(2)を行うことが多い運用	read/write系システムコールまたはclose(2)でエラーとなっているジョブを特定して、当該ジョブを再実行
wirte(2)　時ディスク同期	標準	主に、比較的大きなファイル(128MB以上)に対して、128MB以上のIOサイズでwrite(2)を行うことが多い運用	自動的にジョブ運用再開するので特別な対応は不要

9.12.4 IO 用メモリ配置の設定

図 9-12のようにIOサーバは２つのCPUとメインメモリのセット(ノード)をインタコネクトで接続する構成となっています。IB 専用のAPIを使用したデータ転送方式では、HCAを接続したノードのメモリをIO用に使用することでデータ転送性能が最適になります。

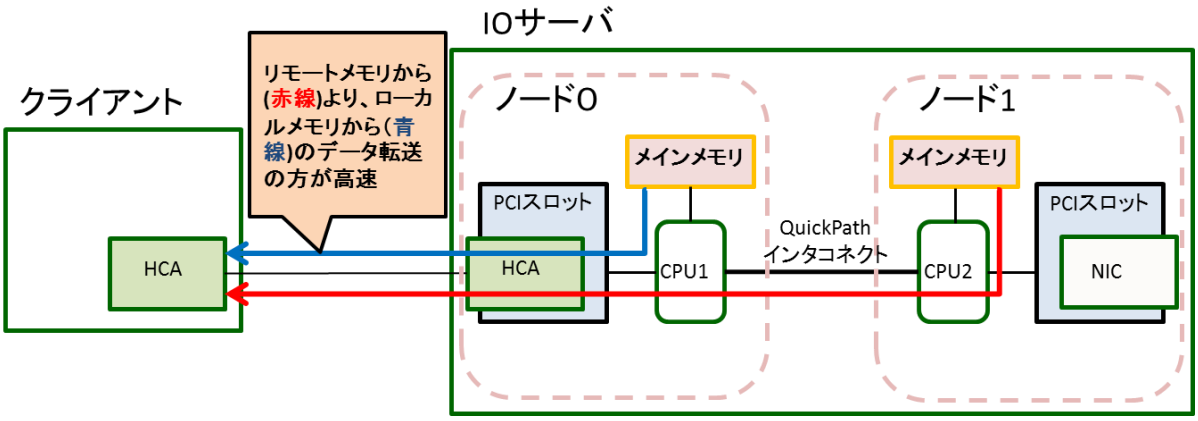


図 9-12　IO 用メモリ配置とデータ転送性能の関係

デフォルトではノード1のメモリをIO用に使用するので、HCAがノード1に接続されている「標準モデル向けIOサーバv3」を使用する場合は、IBSIOMEMNODEを設定する必要はありません。

9.13 サブディレクトリマウント

ScaTeFS のファイルシステムのうち、一部のディレクトリツリーだけを選択してクライアントからマウントする機能を提供します。本機能でマウント可能なサブディレクトリとは、ScaTeFS ファイルシステムのディレクトリ階層にある任意のディレクトリです。サブディレクトリマウントを利用することで、ファイルシステムの一部をアクセス対象とした運用が可能となります。

図 9-13 にて、サブディレクトリマウントの運用イメージを記載します。

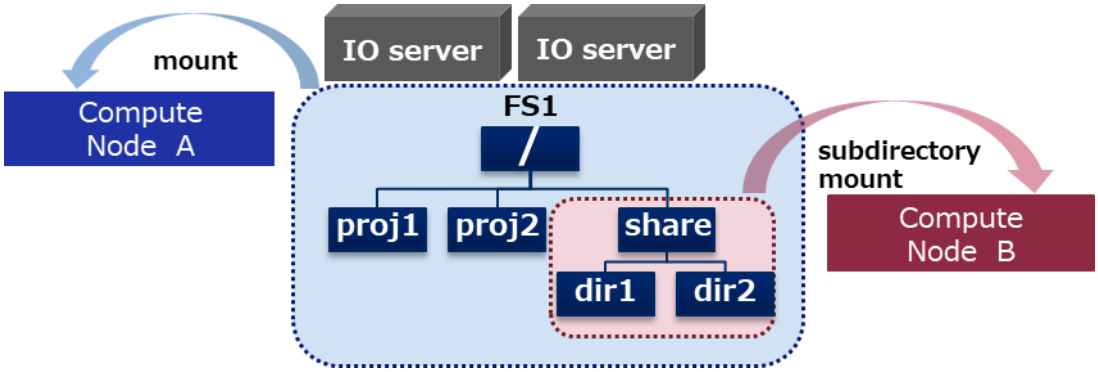


図 9-13　サブディレクトリマウントの運用イメージ

この図では、2 台の IO サーバで構成された ScaTeFS ファイルシステム（FS1）を 2 つクライアン

ト（Compute Node A/B）でマウントしています。A は FS1 全体をマウントしており、FS1 のすべてのディレクトリにアクセス可能です。B は FS1 の/share ディレクトリを部分的にマウントしており、/share/dir1、/share/dir2 はアクセス可能ですが、/proj1、/proj2 にはアクセスできません。

FS1 に対する 2 クライアント（A/B）の FS1 へのアクセス状態（〇：アクセス可能、×：アクセス不可）は以下となります。

ディレクトリ	Compute Node A	Compute Node B
/（FS1 全体）	〇	×
/proj1	〇	×
/proj2	〇	×
/share	〇	〇
/share/dir1	〇	〇
/share/dir2	〇	〇

9.13.1 マウント方法

サブディレクトリのマウントは、マウントするターゲットにサブディレクトリのパス名”/SUBDIR”を付加し、”HOST:FSNAME/SUBDIR”の形式で行います。

以下は、HOST:FS1/share を/mnt/subdir にマウントする例です。

```
# mount -t scatefs HOST:FS1/share /mnt/subdir
```

9.13.2 アンマウント方法

umount(1M)コマンドを使い、従来と同じようにファイルシステムをアンマウントします。/mnt/subdir にマウントされているファイルシステムの一部（HOST:FS1/share）をアンマウントするには、次のいずれかを実行します。

（例1）

```
# umount /mnt/subdir
```

（例2）

```
# umount HOST:FS1/share
```

第10章　メンテナンス

10.1 IOサーバの起動と停止

クラスタ構成のIOサーバの起動および停止方法を記載します。

- 起動

  2台のIOサーバの電源ボタンを続けて押し、IOサーバを起動します。

  ※2台のIOサーバの起動間隔を空けないでください。

- 停止

  どちらかのIOサーバへログインしてclpstdnコマンドを実行します。

  2台のIOサーバが停止します。

```
# clpstdn
```

- 再起動

  どちらかのIOサーバへログインしてclpstdn -rコマンドを実行します。

  2台のIOサーバが再起動します。

```
# clpstdn -r
```

- 起動確認

  どちらかのIOサーバへログインしてclpstatコマンドを実行します。クラスタ状態が表示されますので下記を確認します。

  a) すべてのリソースが Online もしくは Normal であること

  b) <group>タグの current には当該グループのサーバ名が表示されていること
  failover1 グループの current に iosv00、failover2 グループの current に iosv01 が表示されていることが正しいクラスタ状態です。
  もしフェイルオーバしている場合は、<group>タグの current に同じサーバ名が表示されます。問題がある場合は該当リソースの障害を解消してください。

  以下に実行例を記載します。

```
# clpstat
 ========================  CLUSTER STATUS  ============================
  Cluster : cluster
  <server>
   *iosv00 ..........: Online
      lankhb1       : Normal         Kernel Mode LAN Heartbeat
```

```
    diskhb1       : Normal         DISK Heartbeat
  iosv01 ..........: Online
    lankhb1       : Normal         Kernel Mode LAN Heartbeat
    diskhb1       : Normal         DISK Heartbeat
<group>
  failover1 .......: Online
    current       : iosv00
    disk_c_01     : Online
    disk_c_02     : Online
    disk_c_03     : Online
    disk_d_01     : Online
    disk_d_02     : Online
    disk_d_03     : Online
    exec1         : Online
    exec_route1   : Online
    fip_ib1       : Online
    volmgr_c_01   : Online
    volmgr_c_02   : Online
    volmgr_c_03   : Online
    volmgr_d_01   : Online
    volmgr_d_02   : Online
    volmgr_d_03   : Online
  failover2 .......: Online
    current       : iosv01
    disk_c_04     : Online
    disk_c_05     : Online
    disk_c_06     : Online
    disk_d_04     : Online
    disk_d_05     : Online
    disk_d_06     : Online
    exec2         : Online
    exec_route2   : Online
    fip_ib2       : Online
    volmgr_c_04   : Online
    volmgr_c_05   : Online
    volmgr_c_06   : Online
    volmgr_d_04   : Online
    volmgr_d_05   : Online
    volmgr_d_06   : Online
<monitor>
```

```
    diskw_c_01      : Normal
    diskw_c_04      : Normal
    fipw1           : Normal
    fipw2           : Normal
    genw1           : Normal
    genw2           : Normal
    userw           : Normal
    volmgrw1        : Normal
    volmgrw10       : Normal
    volmgrw11       : Normal
    volmgrw12       : Normal
    volmgrw2        : Normal
    volmgrw3        : Normal
    volmgrw4        : Normal
    volmgrw5        : Normal
    volmgrw6        : Normal
    volmgrw7        : Normal
    volmgrw8        : Normal
    volmgrw9        : Normal
 ========================================================================
#
```

10.2 運用中サーバのメンテナンス

IOサーバのメンテナンス作業に関して説明します。

10.2.1 バックアップ

ScaTeFSシステムとして特別なバックアップ機能はサポートしていません。このため、バックアップサーバ上でファイルシステムをマウントし、仮想ファイル単位でバックアップを実施してください。

10.2.2 ScaTeFS パッケージの無停止アップデート

scatefs-srvパッケージは、ファイルシステムの運用を継続した状態でアップデート(無停止アップデート)が可能です。ただし、ファイルシステムを構成する全IOサーバで同期が必要な場合は、無停止アップデート対象外です。パッケージが無停止アップデート可能か否かは、パッケージの指示書に記載されていますので確認してください。

作業は、各IOサーバに管理者権限(root)でログインし、以下の手順で実施してください。

なお、アップデート中は当該IOサーバへのIOが最大で3.5分間遅延します。このため、複数のIOサーバを連続してアップデートした場合、その台数分IOが遅延する可能性がありますので、運用に影響が出ないよう十分な間隔をあけてください(最低8分は間隔をあけてください)。

以下にScaTeFSのライセンスごとの無停止アップデート手順を記載します。

10.2.2.1 HPC ソフトウェアライセンスをお使いの場合

ScaTeFS/Serverのパッケージの無停止アップデート手順は、ScaTeFS/ServerのPPサポートを契約しているかどうかにより異なります。以降では、PPサポートを契約している場合とPPサポートを契約していない場合に分けて説明します。

(1) 準備

【ScaTeFS/Server の PP サポートを契約している場合】

a) yum リポジトリ設定

5.1.12.1 【ScaTeFS/Server の PP サポートを契約している場合】の(1)を参照してください。

b) ScaTeFS/Server パッケージの確認

新しいパッケージがリポジトリに存在することを確認します。

```
# yum list available scatefs-srv
```

【ScaTeFS/Server の PP サポートを契約していない場合】

a) yum リポジトリ設定

5.1.12.1 【ScaTeFS/Server の PP サポートを契約していない場合】の(1)を参照してください。

b) ScaTeFS/Server パッケージの入手

インターネット配信製品ダウンロードサービスを利用して ScaTeFS/Server のパッケージを含む zip ファイルをダウンロードしてください。

5.1.12.1 【ScaTeFS/Server の PP サポートを契約していない場合】の(2)を参照してください。

(2) 環境の確認

clpstatコマンドでクラスタ状態を確認してください。詳細は 10.1 の起動確認を参照してください。

問題がある場合、該当リソースに何らかの障害が発生していますので、無停止アップデートは実施せず障害を解消してください。

(3) アップデート

① IO サーバデーモンが実行されている状態でパッケージを適用します。

【ScaTeFS/Server の PP サポートを契約している場合】

```
# /opt/nec/ve/sbin/TSUBASA-groups-remark.sh scatefs-server
# yum group update scatefs-server
```

【ScaTeFS/Server の PP サポートを契約していない場合】

```
# yum update scatefs-srv-VER.x86_64.rpm
```

② 以下のコマンドを実行します。

```
# /opt/scatefs/sbin/scatefs_restart
```

(4) 結果確認

②のコマンドの実行結果が正常終了(0)していればアップデートは完了です。

異常終了(1)している場合、clpstat コマンドで環境を確認し、下記のそれぞれの状況に応じた対応を実施した上で、サポート部門へ連絡してください。

- フェイルオーバしている場合、以下のコマンドでアップデート前のパッケージに戻した上で、テイクバックを実施してください。

【ScaTeFS/ServerのPPサポートを契約している場合】

yumの履歴からトランザクションIDを確認して、アップデート前のパッケージへ戻します。

```
# yum history list
# yum history undo X
* X: transaction id
```

【ScaTeFS/ServerのPPサポートを契約していない場合】

以前に適用したScaTeFS/Serverパッケージファイルを指定して元に戻します。

```
# yum downgrade アップデート前のScaTeFS/Serverパッケージファイル
```

- その他

アップデート前のパッケージに戻した上で、scatefs_restartコマンドを再実行してください。

10.2.2.2 SX クロスソフトウェア ノードロックライセンスをお使いの場合

(1) 環境の確認

clpstatコマンドでクラスタ状態を確認してください。詳細は 10.1 の起動確認を参照してください。

問題がある場合、該当リソースに何らかの障害が発生していますので、無停止アップデートは実施せず障害を解消してください。

(2) アップデート

① IO サーバデーモンが実行されている状態でパッケージを適用します。

```
# rpm -Uvh アップデートするパッケージ
```

② 以下のコマンドを実行します。

```
# /opt/scatefs/sbin/scatefs_restart
```

(3) 結果確認

②のコマンドの実行結果が正常終了(0)していればアップデートは完了です。

異常終了(1)している場合、clpstat コマンドで環境を確認し、下記のそれぞれの状況に応じた対応を実施した上で、サポート部門へ連絡してください。

- フェイルオーバしている場合、以下のコマンドでアップデート前のパッケージに戻した上で、テイクバックを実施してください。

```
# rpm -Uvh -oldpackage アップデート前のパッケージ
```

- その他

アップデート前のパッケージに戻した上で、scatefs_restartコマンドを再実行してください。

10.3 運用を停止する必要のある事項

システム運用中に実施できないメンテナンス作業があります。これらメンテナンスを実施する場合、システムの運用を停止する必要があります。

- scatefs_extendfsによるファイルシステムの拡張、ストレージグループの追加、ストレージグループの拡張。
- fsckによる修復(ローカルファイルシステムおよびScaTeFSファイルシステム)。

- scatefs_quotacheckによるScaTeFS QUOTA情報の整合性チェックと修復。

10.4 ファイルシステムの整合性チェックと修復

ScaTeFSファイルシステム専用のファイルシステムの整合性チェックと修復機能を提供します。修復ではScaTeFSファイルシステムの運用停止が必要です。修復には、以下の2通りの手順があります。

■ 通常の修復手順（推奨手順）

一度の停止期間内にすべてのメンテナンスを実施します。

① ScaTeFSファイルシステムの運用を停止
② ローカルファイルシステム提供のfsckを実施（必要な場合）
③ ScaTeFSファイルシステムの整合性チェックと修復を実施
④ QUOTA情報の整合性チェックと修復を実施（実施を推奨）
⑤ ScaTeFSファイルシステムの運用を再開

■ より停止時間を短くする修復手順

時間を要するファイルシステムの整合性チェックを運用中に実施することで、運用停止時間を短くします。

① ディスク障害などローカルファイルシステムの修復が必要な場合
- ScaTeFSファイルシステムの運用を停止
- 障害の要因を取り除きローカルファイルシステム提供のfsckを実施
- ScaTeFSファイルシステムの運用を再開

② ScaTeFSファイルシステムの整合性チェックのみを実施し実行結果を任意のファイルへ退避（チェックのみの場合運用中に実施可能）
③ ScaTeFSファイルシステムの運用を停止
④ ローカルファイルシステム提供のfsckを実施（必要な場合）
⑤ ②の整合性チェック結果を入力としてScaTeFSファイルシステムの修復を実施
⑥ QUOTA情報の整合性チェックと修復を実施（実施を推奨）
⑦ ScaTeFSファイルシステムの運用を再開

なお、ディスク障害などローカルファイルシステムの修復が必要な場合、②～③の運用中に、特定のディレクトリやファイルにアクセスできないなどの事象が発生する場合があります。これらは⑤を実施することにより解消されます。

各コマンドの使用方法を以下に記載します。

- 整合性チェック

  実施対象のファイルシステムIDを指定し、ファイルシステムの整合性チェックを行います。

  (例)

```
$ scatefs_fsck -n fsid
```

- 整合性チェックと修復

  実施対象のファイルシステムIDを指定し、ファイルシステムの修復を行います。ファイルシステムの修復前にすべてのIOサーバでIOサーバデーモンの停止を行います。

  ※修復が完了しましたら正しく修復していることの検証のため、再度修復を実施してください。

  (例)

```
$ scatefs_fsck fsid
```

- 整合性チェック結果をもとにScaTeFSファイルシステムの修復

  整合性チェック結果ファイルを指定し、ファイルシステムの修復を行います。整合性チェック結果で修復対象が絞り込まれているため、高速に修復が可能になります。ファイルシステムの修復前にすべてのIOサーバでIOサーバデーモンの停止を行います。

  ※修復が完了しましたら正しく修復していることの検証のため、再度修復を実施してください。

  (例)

```
$ scatefs_f2fsck infile
```

- QUOTA情報の整合性チェックと修復

  実施対象のファイルシステム名を指定し、QUOTA情報の整合性チェックと修復を行います。

  QUOTA情報の整合性チェックと修復前にすべてのIOサーバでIOサーバデーモンの起動を行います。

  (例)

```
$ scatefs_quotacheck fsname
```

10.5 ネットワークの経路障害とパス切り替え

ScaTeFSクライアントは複数の経路を使ってIOサーバと通信を行います。

ScaTeFSクライアントとIOサーバ間の一部の経路にネットワーク障害が発生した場合、

ScaTeFSクライアントは利用できる経路に切り替えて通信を継続します(パス切り替え)。ネットワーク障害が発生した経路はScaTeFS経路監視デーモンが監視し、復旧を検知すると自動的に経路の利用が再開されます。そのため、ネットワーク障害の復旧後に必要な処置は特にありません。

10.6 10GbE-NIC

サポート部門から10GbE-NICドライバのバージョンアップの指示があった場合、ドライバのアップデートを行う必要があります。ScaTeFSはDCBを利用していますが、10GbE-NICベンダが提供しているRPMバイナリパッケージはそのままではDCBに対応しておりません。このため、アップデート手順は別途サポート部門から入手してください。

10.7 ConnectX-6 HCAカード交換後のFirmware更新

ConnectX-6 HCAカードを故障交換した場合、手動でのFirmwareアップデートが必要になる場合があります。

※対象マシンがSX-Aurora TSUBASAの場合は、本手順ではなくSX-Aurora TSUBASAのマニュアルを参照してください。本手順ではSX-Aurora TSUBASA以外のLinuxマシン（スカラーマシン）とIOサーバを対象としています。

HCAカード交換後、ibstatコマンドでFirmwareのバージョンを確認します。

```
$ /usr/sbin/ibstat | grep -i firmware
        Firmware version: 20.27.6008
```

バージョンが上記よりも古い場合は、本手順をもとにFirmwareのアップデートを行ってください。

以下にアップデート手順を記載します。

(1) NVIDIA社のサイトからFirmwareをダウンロードします。
https://network.nvidia.com/support/firmware/connectx6ib/
HDR100 1portモデルの場合、OPNは「MCX653105A-ECA」を選択します。

(2) ダウンロードしたFirmwareファイルをアップデート対象マシンに転送します。
ファイルが圧縮されている場合は展開します。

(3) アップデート対象マシン上で以下を実行します。

```
# mst start
```

```
Starting MST (Mellanox Software Tools) driver set
Loading MST PCI module - Success
Loading MST PCI configuration module - Success
Create devices
Unloading MST PCI module (unused) - Success

# mst status
MST modules:
------------
    MST PCI module is not loaded
    MST PCI configuration module loaded

MST devices:
------------
/dev/mst/mt4123_pciconf0         - PCI configuration cycles access.
                                domain:bus:dev.fn=0000:83:00.0 addr.reg=88 data.reg=92
                                Chip revision is: 00
```

上記は実行例です。実際の環境では表示が異なる場合があります。
HCAが複数搭載されている場合、デバイスのパス /dev/mst/mtXXXX_pciconfX も複数表示されます。

(4) mlxfwmanagerコマンドでファームウェアのアップデートを行います。
-dオプションには、(3)で確認したデバイスのパスを指定します。
-iオプションには、firmwareのファイルを指定します。

```
# mlxfwmanager -d /dev/mst/mt4123_pciconf0 -i fw-ConnectX6-rel-20_26_1040-MCX653105A-
ECA_Ax-UEFI-14.19.14-FlexBoot-3.5.803.bin -u
Querying Mellanox devices firmware ...

Device #1:
----------

  Device Type:      ConnectX6
  Part Number:      MCX653105A-ECA_Ax
  Description:        ConnectX-6 VPI adapter card; 100Gb/s (HDR100; EDR IB and 100GbE);
single-port QSFP56; PCIe3.0 x16; tall bracket; ROHS R6
  PSID:             MT_0000000222
  PCI Device Name:  /dev/mst/mt4123_pciconf0
```

```
  Base GUID:        xxxxxxxxxxxxxxxx
  Versions:         Current        Available
     FW             AA.AA.AAAA     BB.BB.BBBB
     PXE            x.x.xxxx       x.x.xxxx
     UEFI           xx.xx.xxxx     xx.xx.xxxx

  Status:           Forced update required

---------
Found 1 device(s) requiring firmware update...

Device #1: Updating FW ...
Initializing image partition -   OK
Writing Boot image component -   OK
Done

Restart needed for updates to take effect.
```

BB.BB.BBBB の部分にアップデート後のFirmwareバージョンが表示されます。
上記は実行例です。実際の環境では表示が異なる場合があります。

(5) (3)で複数のデバイスが表示された場合、すべてのデバイスに対して(4)を実行します。

> ⚠ **注意**
>
> - HCAカードが正常に認識されない場合、一度 HCA カードの再嵌合を実施しHCA カード故障の切り分けを実施してください。
> - 更新が反映されるのに少々時間がかかります。
>   更新失敗時はreboot後に再実行を１回施行してください。

(6) アップデート対象マシンを再起動します。

```
# reboot
```

(7) アップデートした対象マシンにPingを行い、応答を確認します。

```
$ ping <対象マシンのIPアドレス>
```

(8) アップデート対象マシンにログインし、hca_self_test.ofedコマンドをrootで実行します。

```
# hca_self_test.ofed
---- Performing Adapter Device Self Test ----
Number of CAs Detected ................. 1
PCI Device Check ....................... PASS
Kernel Arch ............................ x86_64
Host Driver Version .................... MLNX_OFED_LINUX-4.6-4.1.2.0 (OFED-4.6-4.1.2):
3.10.0-957.27.2.el7.x86_64
Host Driver RPM Check .................. PASS
Firmware on CA #0 HCA .................. VBB.BB.BBBB
Host Driver Initialization ............. PASS
Number of CA Ports Active .............. 1
Port State of Port #1 on CA #0 (HCA)..... UP 2X HDR (InfiniBand)
Error Counter Check on CA #0 (HCA)...... PASS
Kernel Syslog Check .................... PASS
Node GUID on CA #0 (HCA) ............... b8:59:9f:03:00:00:a7:04
------------------ DONE ---------------------
```

HCAカードが複数ある場合、結果も複数表示されます。すべてのHCAカードの結果に対し、以下を確認します。

- Firmware on CA #N の表示がアップデートしたFirmwareのバージョンに合致すること。
- Host Driver Initialization の表示がPASSとなっていること。
- Error Counter Check on CA #N (HCA) の表示がPASSとなっていること。
- Kernel Syslog Check の表示がPASSとなっていること。

(9) lspciコマンドで、PCIのリンク確認を行います。
指定するPCI IDは、mst statusコマンドの表示で確認した値です。

```
# lspci -s 83:00.0 -vvv | grep LnkSta:
   LnkSta: Speed 8GT/s, Width x16, TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
```

Speedが 8GT/Sであることを確認します。Widthが x16であることを確認します。

以上でFirmwareのアップデートは完了です。

10.8 syslogメッセージ

10.8.1 Linux クライアント

ファイルシステムオペレーション機能

```
ScaTeFS:400100 commit error after file close. filesystem name=<filesystem name>
dev=<device number> code=<code> data=<internal data>
```

[種別] ERROR

[説明] ファイルシステムで、ファイルクローズ後に、そのファイルに書き出されたデータのIOサーバのストレージへの同期処理でエラーが発生しました。

filesystem name: ファイルシステム名

device number: ファイルシステムのデバイス番号

code: エラーを表すコード(errnoと同じ値)

internal data: 内部データ

同じファイルシステムで継続してエラーが発生した場合、1時間間隔で本メッセージが出力されます。

[対処] 障害原因を取り除いた後に、障害にあったファイルを、ファイルを作成するジョブの再実行等により復旧してください。

障害にあったファイルは、障害発生日時(本メッセージの出力日時)、本メッセージ中のファイルシステム情報、後述のScaTeFS:400101のメッセージ中のファイル情報、アプリケーションによるファイルのアクセス状況等から特定することになります。

```
ScaTeFS:400101 commit error after file close. dev=<device number> ino=<inode number>
uid=<user id> gid=<group id> code=<code> data=<internal data>
```

[種別] ERROR

[説明] ファイルで、ファイルクローズ後に、そのファイルに書き出されたデータのIOサーバのストレージへの同期処理でエラーが発生しました。

device number: ファイルシステムのデバイス番号

inode number: ファイルのinode番号

user id: ファイルのユーザID

group id: ファイルのグループID

code: エラーを表すコード(errnoと同じ値)

internal data: 内部データ

本メッセージの出力数が5秒間で200個を超えた場合、それ以降の5秒間は、本メッセージの出力は抑止されます。抑止された場合、後述のScaTeFS:400102のメッセージが出力されます。ScaTeFS:400102のメッセージが出力された場合、本メッセージから、障害にあったすべてのファイルを特定することはできません。障害にあったが本メッセージが出力されなかったファイルが存在します。

[対処] 障害原因を取り除いた後に、障害にあったファイルを、ファイルを作成するジョブの再実行等により復旧してください。

障害にあったファイルは、障害発生日時(本メッセージの出力日時)、前述のScaTeFS:400100のメッセージ中のファイルシステム情報、本メッセージのファイル情報、アプリケーションによるファイルのアクセス状況等から特定することになります。

ScaTeFS:400102のメッセージが出力された場合、本メッセージから、障害にあったすべてのファイルを特定することはできません。障害発生直前のアプリケーションによるファイルのアクセス状況等から、障害にあったファイルを特定する必要があります。

```
ScaTeFS:400102 drop commit error messages due to rate-limiting. data=<internal data>
```

[種別] ERROR

[説明] ファイルで、ファイルクローズ後に、そのファイルに書き出されたデータのIOサーバのストレージへの同期処理でエラーが発生したこと表すメッセージ(ScaTeFS:400101)の出力が抑止されました。

internal data: 内部データ

[対処] 不要です。

データ転送機能(TCP)

```
ScaTeFS:RPC: all connections related to <ServerAddress>:<Port> are failed, still
trying
```

[種別] WARNING

[説明] IOサーバとの通信が失敗しました。全パス障害が発生しています。

[対処] ネットワーク経路に異常がないか確認してください。IOサーバの状態を確認してください。

```
ScaTeFS:RPC: all connections related to <ServerAddress>:<Port> are failed, timed out
```

[種別] WARNING

[説明] IOサーバとの通信が失敗しました。全パス障害が発生しています。ソフトマウントのため、ファイル操作はエラーとなります。

[対処] ネットワーク経路に異常がないか確認してください。IOサーバの状態を確認してください。

```
ScaTeFS:RPC: retry to server <ServerAddress>:<Port> has been cancelled by signal.
```

[種別] NOTICE

[説明] 再送を行っていましたが、要求がシグナルにより中断されました。

[対処] 不要です。

```
ScaTeFS:RPC: server <ServerAddress>:<Port> OK
```

[種別] NOTICE

[説明] 再送が発生していましたが、IOサーバと通信ができました。

[対処] 不要です。

```
ScaTeFS:RPC: server <ServerAddress>:<Port> is unavailable. Using alternative
connection path
```

[種別] WARNING

[説明] 再送オーバーが発生したため、パス切り替え処理を開始しています。

[対処] ネットワーク経路に異常がないか確認してください。IOサーバの状態を確認してください。

```
ScaTeFS:RPC: server <ServerAddress>:<Port> not responding, still trying
```

[種別] NOTICE

[説明] IOサーバとの通信がタイムアウトしたため、再送を行っています。

[対処] 頻発する場合、ネットワーク経路に異常がないか確認してください。また、IOサーバの状態を確認してください。

```
ScaTeFS:RPC: server <ServerAddress>:<Port> not responding, timed out. (pid=<PID>,
proc=<ProcedureNumber>)
```

[種別] NOTICE

[説明] RPCの応答がありません。ソフトマウントのため、RPC要求はエラーとなりました。

[対処] IOサーバの状態を確認してください。また、ネットワーク経路に異常がないか確認してください。

```
ScaTeFS:pmond: connect to server <ServerAddress>:<Port> ok
```

[種別] NOTICE

[説明] 障害状態の経路が復旧しました。

[対処] 不要です。

データ転送機能(IB Verbs)

```
ScaTeFS:verbs: all connections related to <ServerAddress> for <ConnectionType> are
failed, still trying.
```

[種別] WARNING

[説明] IOサーバとの通信が失敗しました。全パス障害が発生しています。

[対処] ネットワーク経路に異常がないか確認してください。IOサーバの状態を確認してください。

```
ScaTeFS:verbs: all connections related to <ServerAddress> for <ConnectionType> are
failed, timed out.
```

[種別] WARNING

[説明] IOサーバとの通信が失敗しました。全パス障害が発生しています。ソフトマウントのため、ファイル操作はエラーとなります。

[対処] ネットワーク経路に異常がないか確認してください。IOサーバの状態を確認してください。

```
ScaTeFS:verbs: connection to <ServerAddress>:hca<N> is marked as disconnected.
(<Internal data>)
```

[種別] NOTICE

[説明] 無効になったコネクションを検出したため、そのコネクションを切断しました。

[対処] 不要です。

```
ScaTeFS:verbs: Control request to <ServerAddress> failed. (<Internal data>)
```

[種別] NOTICE

[説明] 制御通信が失敗しました。当該IOサーバに対しIPoIBの通信ができません。

[対処] 本メッセージは補助的なメッセージです。前後に出力されるメッセージを参照してください。

```
ScaTeFS:verbs: Control request to <ServerAddress> was skipped. (<Internal data>)
```

[種別] NOTICE

[説明] 制御通信が失敗しました。当該IOサーバに対しIPoIBの通信ができません。

[対処] 本メッセージは補助的なメッセージです。前後に出力されるメッセージを参照してください。

```
ScaTeFS:verbs: Control request to <ServerAddress> was skipped. (<Internal data>)
```

[種別] NOTICE

[説明] 制御通信が失敗しました。当該IOサーバに対しIPoIBの通信ができません。

N: HCAを表す番号。IOサーバに登録したN個目のHCAに該当(1オリジン)。

[対処] 本メッセージは補助的なメッセージです。前後に出力されるメッセージを参照してください。

```
ScaTeFS:verbs: detaching device done. (device=<HCA>)
```

[種別] WARN

[説明] HCAに異常を検知しました。当該HCAを利用対象から除外しています。

[対処] クライアントのHCAの状態を確認してください。

```
ScaTeFS:verbs: pmond: could not connect to server <ServerAddress>:hca<N>, still
```

```
trying. (<Internal data>)
```

[種別] WARN

[説明] IOサーバと通信できません。状態を定期的に監視中です。

N: 通信先のHCAを表す番号。IOサーバに登録したN個目のHCAに該当(1オリジン)。

[対処] ネットワーク経路に異常がないか確認してください。IOサーバの状態を確認してください。

```
ScaTeFS:verbs: pmond: InfiniBand device is unavailable, retry after delay.
(device=<HCA>, guid=<DeviceGuid>)
```

[種別] WARN

[説明] HCAに異常を検知しました。状態を定期的に監視中です。

[対処] クライアントのHCAの状態を確認してください。

```
ScaTeFS:verbs: re-attaching device done. (device=<HCA>)
```

[種別] NOTICE

[説明] HCAの組み込みが完了しました。当該HCAの利用を再開します。

[対処] 不要です。

```
ScaTeFS:verbs: re-attaching device done. (device=<HCA>, not mounted)
```

[種別] NOTICE

[説明] HCAの組み込みが完了しました。HCAが利用可能になりました。

[対処] 不要です。

```
ScaTeFS:verbs: server <ServerAddress>:hca<N> request transmission was not
successful, still trying. (<Internal data>)
```

[種別] NOTICE

[説明] IOサーバとの通信に失敗したため、再送を行っています。

N: 通信先のHCAを表す番号。IOサーバに登録したN個目のHCAに該当(1オリジン)。

[対処] 頻発する場合、ネットワーク経路に異常がないか確認してください。IOサーバの負

荷状況を確認してください。

```
ScaTeFS:verbs: server <ServerAddress>:hca<N> is unavailable (<Internal data>). Using
alternative connection path.
```

[種別] WARNING

[説明] 再送オーバーが発生したため、パス切り替え処理を開始しています。

N: 通信先のHCAを表す番号。IOサーバに登録したN個目のHCAに該当(1オリジン)。

[対処] ネットワーク経路に異常がないか確認してください。IOサーバの状態を確認してください。

```
ScaTeFS:verbs: server <ServerAddress>:hca<N> not responding, still trying. (<Internal
data>)
```

[種別] NOTICE

[説明] IOサーバとの通信がタイムアウトしたため、再送を行っています。

N: 通信先のHCAを表す番号。IOサーバに登録したN個目のHCAに該当(1オリジン)。

[対処] 頻発する場合、ネットワーク経路に異常がないか確認してください。IOサーバの負荷状況を確認してください。

```
ScaTeFS:verbs: server <ServerAddress>:hca<N> OK. (<Internal data>)
```

[種別] NOTICE

[説明] 再送が発生していましたが、IOサーバと通信ができました。

N: 通信先のHCAを表す番号。IOサーバに登録したN個目のHCAに該当(1オリジン)。

[対処] 不要です。

```
ScaTeFS:verbs: server <ServerAddress>:hca<N> recovery OK. (<Internal data>)
```

[種別] NOTICE

[説明] 障害状態の経路が復旧しました。

N: 通信先のHCAを表す番号。IOサーバに登録したN個目のHCAに該当(1オリジン)。

[対処] 不要です。

```
ScaTeFS:verbs: start re-attaching device. (devname=<HCA>)
```

[種別] NOTICE

[説明] HCAを検出しました。当該HCAの再組み込みを開始します。

[対処] 不要です。

ライセンス管理機能

```
ScaTeFS_LS:300001 heartbeat to license server failed. continue process. errmsg=<error
message> data=<internal data>
```

[種別] WARNING

[説明] ライセンスサーバへのハートビートの送信でエラーが発生しました。ハートビート間隔の時間が経過した後にリトライします。

error message: エラーメッセージ

internal data: 内部データ

[対処] 頻発する場合、ネットワーク経路に異常がないか確認してください。ライセンスサーバの状態を確認してください。

```
ScaTeFS_LS:300002 heartbeat to license server recovered. data=<internal data>
```

[種別] WARNING

[説明] ライセンスサーバへのハートビートの送信が回復しました。

internal data: 内部データ

[対処] 不要です。

```
ScaTeFS_LS:400101 ScaTeFS client license is not valid. data=<internal data>
```

[種別] ERROR

[説明] ノードロックライセンスが有効ではありません。

internal data: 内部データ

[対処] ライセンスファイルが正しく設定されているか確認してください。

```
ScaTeFS_LS:400201 ScaTeFS client license process failed. reason=<reason>
data=<internal data>
```

[種別] ERROR

[説明] ライセンスの処理に失敗しました。

reason: 失敗の原因となったエラーメッセージ

internal data: 内部データ

[対処] 失敗の原因となったエラーメッセージの内容に応じて、必要な処置を行ってください。

10.8.2 IO サーバ

syslog を使用した IO サーバの障害監視方法を記載します。

（***は任意の文字列を示す）

ストレージ関連メッセージ

```
lpfc***Down
または
lpfc***Reset
```

[種別] ERROR

[説明] サーバ側FCポートに障害が検出されました。

[対処] ストレージとIOサーバの経路上に障害が発生した可能性があります。

サポート部門に連絡してください。

```
sps: Warning: Detect *** path fail
または
sps: Warning: *** is not redundant
```

[種別] ERROR

[説明] ディスクポートに障害が検出されました。

[対処] spsadminコマンドにてパス構成を確認してください。

SPS関連マニュアルを確認の上、サポート部門に連絡してください。

ネットワーク関連メッセージ

```
cxgb4***link down
```

[種別] ERROR

[説明] 10G NIC(T4カード)の link downが検出されました。

[対処] サポート部門に連絡してください。

CLUSTERPRO 関連メッセージ

```
There was a request to restart resource(***) from the clprm process
```

[種別] WARNING

[説明] CLUSTERPROがリソースの異常を検出し当該リソースを再起動しました。
CLUSTERPROによりフェイルオーバが実行される可能性があります。

[対処] ScaTeFSの状態確認(*1)を実施し、リソース異常の原因を取り除いてください。
メッセージ詳細は、CLUSTERPRO関連マニュアルを確認してください。

```
Detected an error in monitoring ***
```

[種別] ERROR

[説明] CLUSTERPROがモニタリソースの監視で異常を検出しました。
CLUSTERPROによりフェイルオーバが実行される可能性があります。

[対処] ScaTeFSの状態確認(*1)を実施し、リソース異常の原因を取り除いてください。
メッセージ詳細は、CLUSTERPRO関連マニュアルを確認してください。

```
Resource *** of server *** has stopped
```

[種別] ERROR

[説明] IOサーバの特定のリソースが停止しました。
CLUSTERPROによりフェイルオーバが実行されます。

[対処] フェイルオーバした状態で運用は継続可能です。ScaTeFSの状態確認(*1)を実施し、
リソース異常の原因を取り除いてください。ただし、2セット以上のIOサーバで
フェイルオーバが発生し、かつリソース異常の原因が不明の場合は、障害拡大の
可能性があるため、直ちに運用を停止してください。
メッセージ詳細は、CLUSTERPRO関連マニュアルを確認してください。

ScaTeFS 関連メッセージ

```
IOS*** server started (secondary mode)
```

[種別] ERROR

[説明] ScaTeFSサーバ機能がフェイルオーバしました。

[対処] フェイルオーバした状態で運用は継続可能です。
ScaTeFSの状態確認(*1)を実施し、リソース異常の原因を取り除いてください。
ただし、2セット以上のIOサーバでフェイルオーバが発生し、かつリソース異常の原因が不明の場合は、障害拡大の可能性があるため、直ちに運用を停止してください。

```
async event(IBV_EVENT_LID_CHANGE) at hca(***). stop the daemon.
または
async event(IBV_EVENT_CLIENT_REREGISTER) at hca(***). stop the daemon.
```

[種別] ERROR
[説明] サブネットマネージャの再起動等により、IOサーバデーモンが再起動しました。
[対処] サブネットマネージャの状態に問題がないか確認してください。
また、メンテナンスによるサブネットマネージャの再起動はScaTeFSの運用を停止してから実施するようにしてください。

```
async event(IBV_EVENT_SM_CHANGE) at hca(***). stop the daemon.
```

[種別] ERROR
[説明] サブネットマネージャが予備のサブネットマネージャに切り替わり、
IOサーバデーモンが再起動しました。
[対処] サブネットマネージャの状態に問題がないか確認してください。

```
InfiniBand timeout happened on HCA#<N> (PID=*** CLIENTID=***)
```

[種別] WARNING
[説明] InfiniBandによる通信でタイムアウトが発生しました。
N: IOサーバのHCAを識別する番号。scatefs_addiosコマンドに指定する定義ファイルにおける、pciid@hcaportの項目にN番目に指定したHCAに該当します。(0オリジン)
[対処] ネットワーク経路に異常がないか確認してください。

```
NET: hca(***:<hca-id1>:<hca-port1>) is replaced with hca(***:<hca-id2>:<hca-port2>)
```

[種別] WARNING
[説明] IOサーバデーモン起動時に非ACTIVE状態のHCAを検出したので、ACTIVE状態のHCAで代替して起動しました。
<hca-idX>:<hca-portX>はscatefs_addiosコマンドに指定する定義ファイル中のpciid@hcaportに指定したHCAのIDとポート番号です。

[対処] IOサーバのHCAに異常がないか確認してください。

【注釈】

(*1)「ScaTeFS の状態確認」とは下記を指します。

- clpstat コマンドでクラスタ状態を表示し、以下を確認してください。
  異なる場合、当該リソースに何らかの問題が発生しています。
  - o 全てのリソースが Online もしくは Normal であること
  - o <group>タグの current には当該グループのサーバ名が表示されていること
    (フェイルオーバしている場合は、2 つの<group>タグの current に同じ
    サーバ名が表示されます)
- クライアントから正常にアクセスできていることを確認してください。
  - o 6.2.5 マウント方法 に記載されているマウント後の IO 確認を実施する

第11章　利用者向けの利用、設定方法

11.1 仮想ファイルシステムと実ファイルシステム

ScaTeFSは、複数のIOサーバにより構成されており、これを仮想的に１つのファイルシステムとしてScaTeFSクライアントに見せています。このため、これを「仮想ファイルシステム」と呼称します。

仮想ファイルシステムは、図 11-1のように各IOサーバ配下に接続されたストレージ上に作成される複数のLinuxのファイルシステムから成っています。これらを「実ファイルシステム」または「IOターゲット」と呼称します。実ファイルシステムは、各IOサーバ配下に最低１つ、一般に複数存在します。並列I/Oを効率よく実施するためには、仮想ファイルシステムが何台のIOサーバと実ファイルシステムにより構成されているかを把握しておく必要があります。

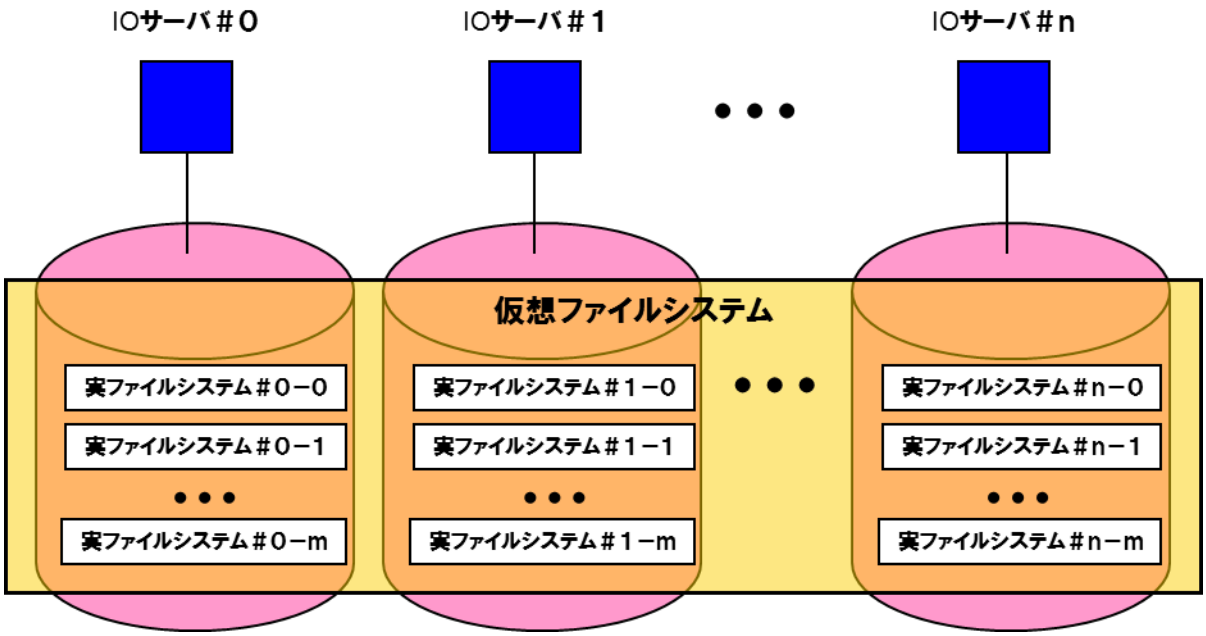


図 11-1　仮想ファイルシステムと実ファイルシステムの関係

図 11-1の例では、ファイルのデータは最大(n + 1)×(m + 1)個の実ファイルシステムに分散配置されることになります。

11.2 仮想ファイルと実ファイル

仮想ファイルの断片を各実ファイルシステムに分散配置します。この断片のことを実ファイ

ルと呼称します。この断片と各実ファイルシステムへの配置の方法の違いにより、２種類のファイルフォーマットが選択できます。

形式１：ノンストライプフォーマット

形式２：ストライプフォーマット

デフォルトは、ノンストライプフォーマットです。

11.2.1 ノンストライプフォーマット(形式１)

図 11-1の仮想ファイルのイメージのとおり、仮想ファイルは実ファイルを順に連結したものです。この連結の単位をチャンクサイズと呼称します。この値は、後述のscatefs_premap(1)により設定可能であり、チャンクサイズの既定値は256MBです。

図 11-1は、ノンストライプフォーマット(形式１)の場合の仮想ファイルのイメージとこれを構成する実ファイルが各実ファイルシステムへどのように配置されるかを例示しています。

この場合、各IOサーバ配下にそれぞれ２つの実ファイルシステム(ターゲット)を作成して１つのScaTeFSを構成しています。図 11-1の仮想ファイルは、チャンク番号#0～#10で構成されており、仮想ファイルの先頭であるチャンク番号#0は、TID＝１に配置されています。以降は、これを起点として

TID＝(１，２，３，０，５，６，７，４，１・・・)

のように最初はチャンク番号#0が配置されたターゲットと同列の各IOサーバ配下のターゲットに配置され、IOサーバを一巡すると次のターゲットに配置されます。

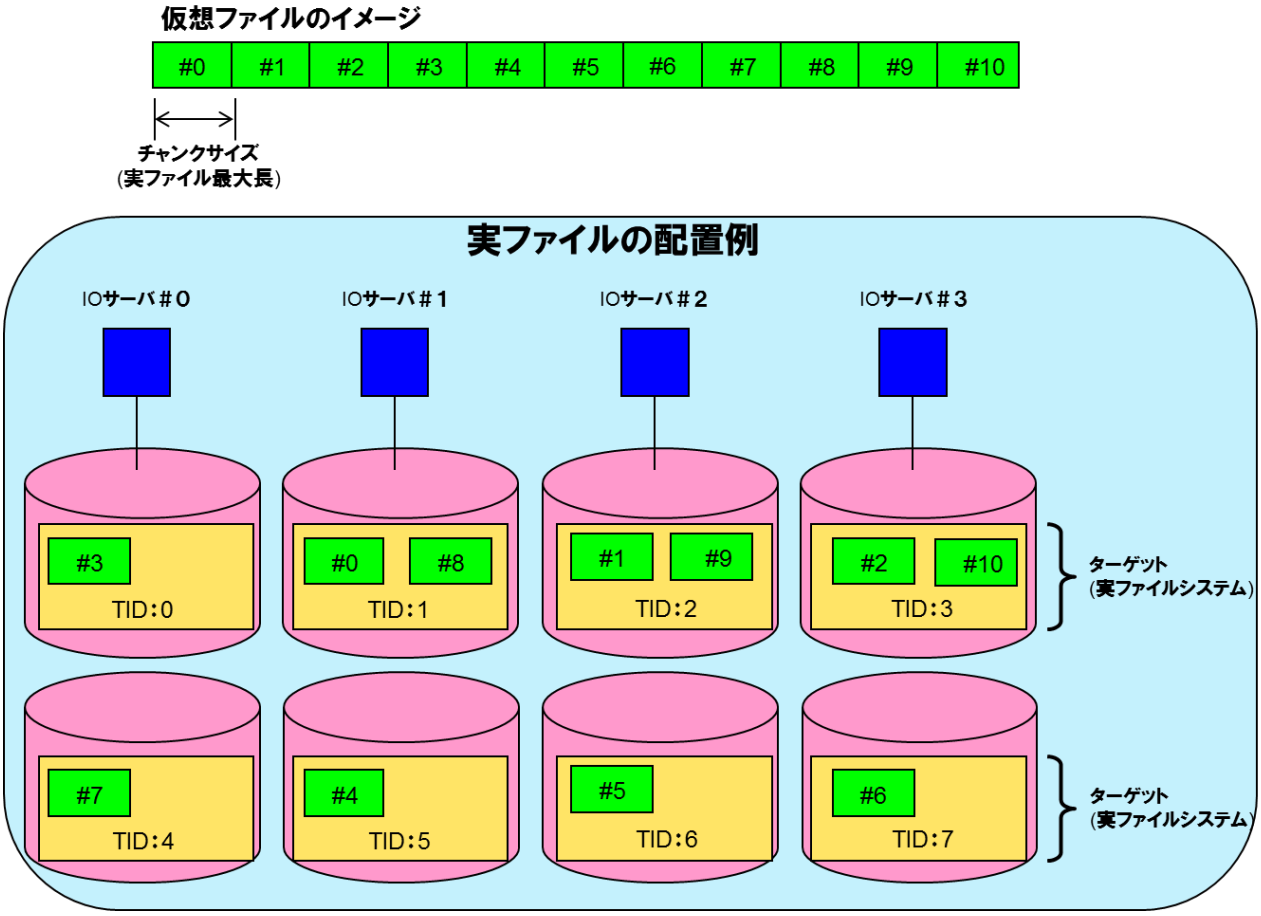


図 11-2　形式 1 の仮想ファイルと実ファイルの関係

11.2.2 ストライプフォーマット(形式 2)

特定のノードから複数のIOサーバに同時にリクエストを発行することができるので、単体I/Oの処理を効率化したい時に有用です。たとえば、図 11-3の例では、IOサーバが 2 台あり各IOサーバ配下にターゲットが 2 つあるため、ストライプサイズの 2 倍または 4 倍のI/Oサイズでread/writeシステムコールを呼び出した際に効果を期待できます。つまり、仮想ファイルの #0, #1または #0, #1, #2, #3に対してほぼ同時にread/writeができます。ただし、ScaTeFSが使用しているノード(クライアント)のネットワークインターフェースの持つ帯域に制限されることに注意してください。

なお、並列I/O(後述)の場合は、I/Oの発行の仕方によっては、ノード間で同一実ファイルの異なるオフセットを更新／参照することにより競合が発生することがあります。

図 11-3のようにストライプサイズを仮想ファイルを構成する基本単位とし、チャンクサイズはストライプサイズの倍数である必要があります。デフォルトのファイルフォーマットは形式 1 であるため、形式 2 を使用するためには後述のscatefs_premap(1)により、明示的にストライプサイズとチャンクサイズを指定する必要があります。図 11-3の例では、各IOサーバ配下にそれぞれ 2 つの実ファイルシステム(ターゲット)を作成して 1 つのScaTeFSを構成して

います。図中の仮想ファイルは、チャンク番号#0～#20で構成されており、仮想ファイルの先頭であるチャンク番号#0は、TID＝３に配置されています。以降は、これを起点とし、

TID＝(３，２，１，０，３，２，１，０・・・)

のように最初は#0が配置されたターゲットと同列の各IOサーバ配下のターゲットに配置され、IOサーバを一巡すると次のターゲットに配置されます。さらに、実ファイルのサイズがチャンクサイズに達すると、同一ターゲット内で新たに実ファイルを生成します。

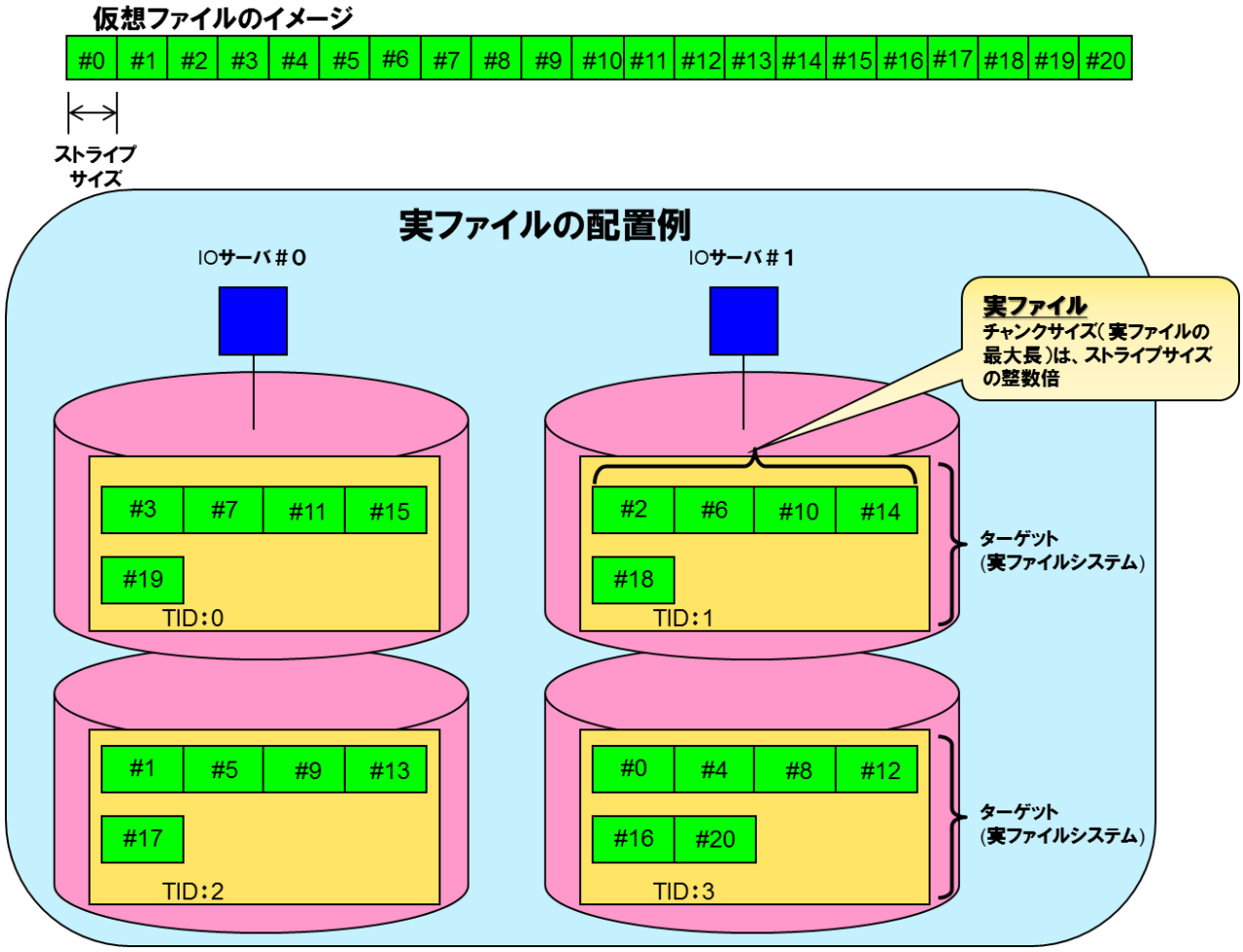


図 11-3　形式２の仮想ファイルと実ファイルの関係

11.3 並列I/O

本書においての並列I/Oとは、複数の計算ノードを使用して並列にデータを転送することにより、１つのファイルへの書き込み、読み込みを行うことを指しています。巨大ファイルへのI/O効率を上げることが主たる目的です。図 11-4は、最も簡単な並列I/Oの例です。

並列I/Oにより並列度に見合うI/O性能を得るためには、ScaTeFSを構成するIOサーバ数、IOターゲット数を考慮に入れた上で、並列I/Oの対象とする仮想ファイルのフォーマット(形

式１／形式２)、チャンクサイズなどを決める必要があります。これは、IOサーバやストレージにおいて競合を発生させないようにするためです。

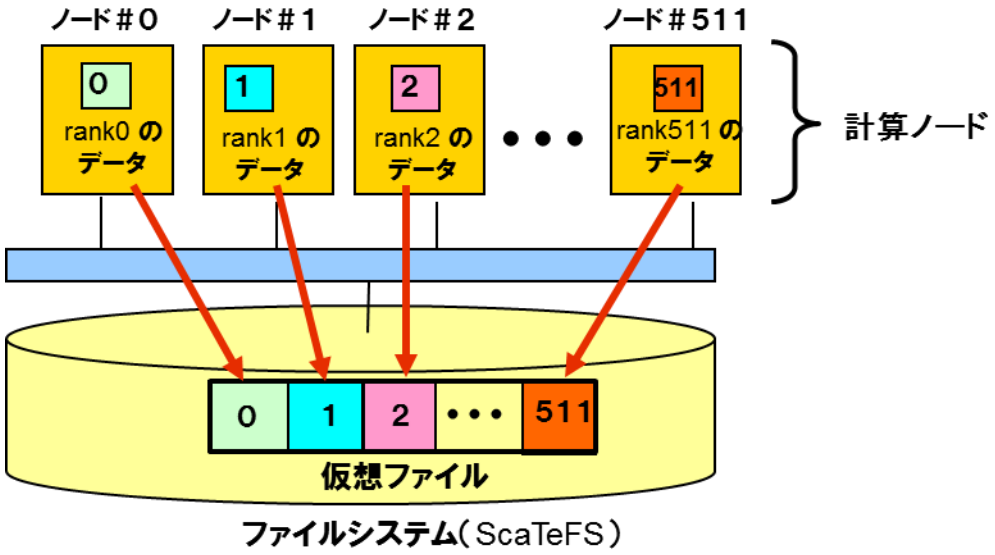


図 11-4　形式 1 を前提とした並列 I/O のイメージ

11.4 並列I/Oの効率化(ファイルのプリマップ)

図 11-4に示すように512ノードから一斉にwriteを行って、512個の実ファイルよりなる１つの仮想ファイルを作成するとします。この際、512個の実ファイルがほぼ同時に生成されることになり、仮想ファイルの管理情報の更新が若干オーバーヘッドとなる可能性が考えられます。これを軽減するために、writeに先立って予め必要数の実ファイルを生成するプリマップという機能があります。

この機能は、後述のファイルフォーマットの指定(SUPER-UXのfcntl(2)またはscatefs_premap(1))の際に同時にファイルサイズを指定することにより実行できます。詳細は、SUPER-UXのfcntl(2)およびscatefs_premap(1)を参照してください。

11.5 ファイルフォーマットの設定と表示

ファイルフォーマットの設定を行うには、対象がファイルである場合scatefs_premap(1)を、対象がディレクトリである場合scatefs_setdirattr(1)を使用します。ファイルフォーマットを確認するには、scatefs_getfinfo(1)を使用します。以下に例を記載します。

11.5.1 ノンストライプフォーマット(形式 1)の設定

- ファイル

scatefs_premap(1)に -cオプションとファイルサイズを指定することで、形式1のファイルを作成します。例では、チャンクサイズ2G、ファイルのサイズ4Gでプリマップを行っています。

(例)

```
$ scatefs_premap -c 2G 4G /mnt/scatefs/file000
```

ファイルフォーマットのみを指定したファイルを作成する場合は、ファイルサイズを0に設定します。

(例)

```
  $ scatefs_premap -c 2G 0 /mnt/scatefs/file001
```

- ディレクトリ

  scatefs_setdirattr(1)に-cオプションを指定することで、既存ディレクトリを形式1のフォーマットに変更します。変更完了後、ディレクトリ配下に新規作成されるファイル、ディレクトリに変更後の値が反映されます。既存ファイル、ディレクトリには反映されません。例では、チャンクサイズ4Gに設定を変更しています。

  (例)

```
$ scatefs_setdirattr -c 4G /mnt/scatefs/dir000
```

11.5.2 ストライプフォーマット(形式 2)の設定

- ファイル

  scatefs_premap(1)に -s オプションを指定することで、形式2のファイルを作成します。例では、ストライプサイズ4M、チャンクサイズ1G、ファイルサイズ1Gプリマップを行っています。なお、既存ファイルを指定した場合、ファイルサイズが0の場合にのみプリマップを行うことが可能です。

  (例)

```
$ scatefs_premap -s 4M -c 1G 1G /mnt/scatefs/file002
```

- ディレクトリ

  scatefs_setdirattr(1)に -s オプションを指定することで、形式2にフォーマットを変更します。変更完了後、ディレクトリ配下に新規作成されるファイル、ディレクトリに変更後の値が反映されます。既存ファイル、ディレクトリには反映されません。例では、既存ディレクトリ性を、ストライプサイズ4M、チャンクサイズ1Gに変更しています。

(例)

```
$ scatefs_setdirattr -s 4M -c 1G /mnt/scatefs/dir001
```

11.5.3 システムコールからの設定

SUPER-UXのfcntl(2)を使用してフォーマットの指定を行う例を以下に記載します。

システムコールからの設定は、SUPER-UXでのみ行えます。Linuxクライアントでは行えません。

- ファイル

  プリマップを行うファイルをopen(2)し、scfs_premap構造体のメンバにそれぞれ値を指定します。fcntl(2)の第一引数にファイルディスクリプタ、第二引数にF_SCPREMAP、第三引数にscfs_premap構造体のアドレスを指定します。

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/fcntl.h>
:
int main (int argc, char *argv[])
{
    int fd;
    char *filepath;
    struct scfs_premap p;

    fd = open(filepath, O_RDWR);

    /* 値を設定 */
    p.stripesize = stripesize;
    p.chunksize = chunksize;
    p.filesize = filesize;

    /* fcntl(2)呼び出し */
    fcntl(fd, F_SCPREMAP, &p);
    return 0;
}
```

※チャンクサイズ、ストライプサイズは4K単位で指定する必要があります。

※形式1の場合、チャンクサイズ、ストライプサイズを同じ値に指定します。

※形式2の場合、チャンクサイズはストライプサイズの倍数であり、かつストライプサイズより大きい値を指定します。
※ファイルフォーマットのみを指定したファイルを作成する場合は、ファイルサイズを0に設定します。

- ディレクトリ

  フォーマットを変更するディレクトリをopen(2)し、scfs_setdirattr構造体のメンバに値を指定します。fcntl(2)の第一引数にファイルディスクリプタ、第二引数にF_SCSETDIRATTR、第三引数にscfs_setdirattr構造体のアドレスを指定します。

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <sys/fcntl.h>
:
int main (int argc, char *argv[])
{
    int fd;
    char *dirpath;
    struct scfs_setdirattr d;

    fd = open(dirpath, O_RDONLY);

    /* 値を設定 */
    d.stripesize = stripesize;
    d.chunksize = chunksize;

    /* fcntl(2)呼び出し */
    fcntl(fd, F_SCSETDATTR, d);
    return 0;
}
```

※チャンクサイズ、ストライプサイズは4K単位で指定する必要があります。
※形式1の場合、チャンクサイズ、ストライプサイズを同じ値に設定します。
※形式2の場合、チャンクサイズはストライプサイズの倍数であり、かつストライプサイズより大きい値を設定します。

11.5.4 フォーマットの表示

scatefs_getfinfo(1)でファイル/ディレクトリのフォーマット情報を表示することが可能です。

- ファイル

(例)形式1

```
$ scatefs_getfinfo /mnt/scatefs/file001
format     : non stripe format
iot count  :               6
stripesize :       268435456
chunksize  :       268435456
filesize   :      1610612736

format      ---  ファイルフォーマット
iot count   ---  使用しているIOターゲットの数
stripesize  ---   ストライプサイズ
chunksize   ---   チャンクサイズ
filesize     ---   ファイルサイズ
```

※形式1の場合、ストライプサイズ、チャンクサイズは同じ値になります。

(例) 形式2

```
$ scatefs_getfinfo /mnt/scatefs/file002
format    :    stripe format
iot count  :               6
stripesize :        33554432
chunksize  :        67108864
filesize   :       268435456
```

-v オプションを指定することで、ファイルオフセットごとの実ファイル分布を表示することができます。形式１、形式２のファイルを例に表示情報を示します。

- ファイルの詳細表示

(例)形式1

```
$ scatefs_getfinfo -hv /mnt/scatefs/file001
format    : non stripe format
iot count  :               6
```

```
stripesize :          256.0M
chunksize  :          256.0M
filesize   :            1.5G

     offset                          no      ios     iot
-------------------------------------------------------------
              0 ...        268435455      0       0       0
      268435456 ...        536870911      1       1       3
      536870912 ...        805306367      2       0       1
      805306368 ...       1073741823      3       1       4
     1073741824 ...       1342177279      4       0       2
     1342177280 ...       1610612735      5       1       5

offset  ---  仮想ファイルのオフセットを示します。
no      ---  実ファイルのインデックスを示します。
ios     ---  実ファイルが格納されているIOサーバIDを示します。
iot     ---  実ファイルが格納されているIOターゲットIDを示します。
```

形式1の場合、オフセットと実ファイルのインデックスは一致します。以下に実ファイルの配置イメージを記載します。

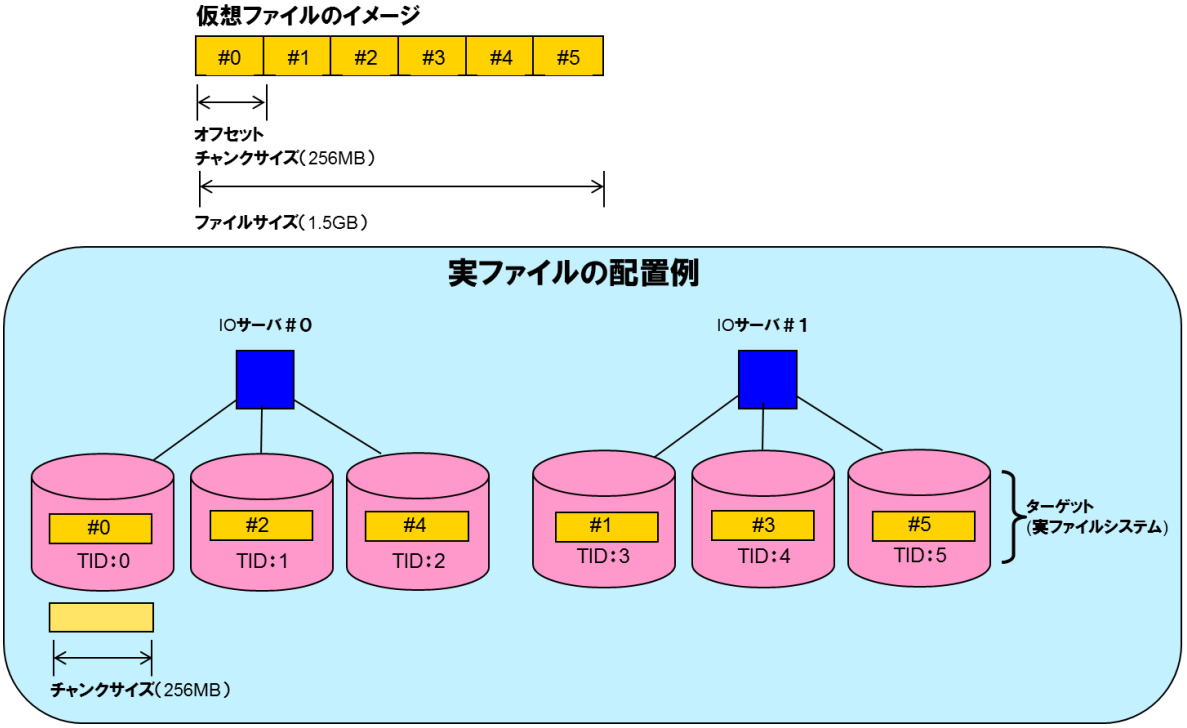


図 11-5　形式 1 における実ファイルの配置例

(例)形式2

```
$ scatefs_getfinfo -hv /mnt/scatefs/file002
format     :     stripe format
iot count  :               6
stripesize :           32.0M
chunksize  :           64.0M
filesize   :          256.0M

       offset                          no      ios     iot
-------------------------------------------------------------
              0 ...         33554431      0       0       0
       33554432 ...         67108863      1       1       3
       67108864 ...        100663295      2       0       1
      100663296 ...        134217727      3       1       4
      134217728 ...        167772159      4       0       2
      167772160 ...        201326591      5       1       5
      201326592 ...        234881023      0       0       0
      234881024 ...        268435455      1       1       3
```

形式2の場合、ストライプサイズ単位で区切ったオフセットに対応する実ファイルのインデックスを表示します。以下に実ファイルの配置イメージを記載します。

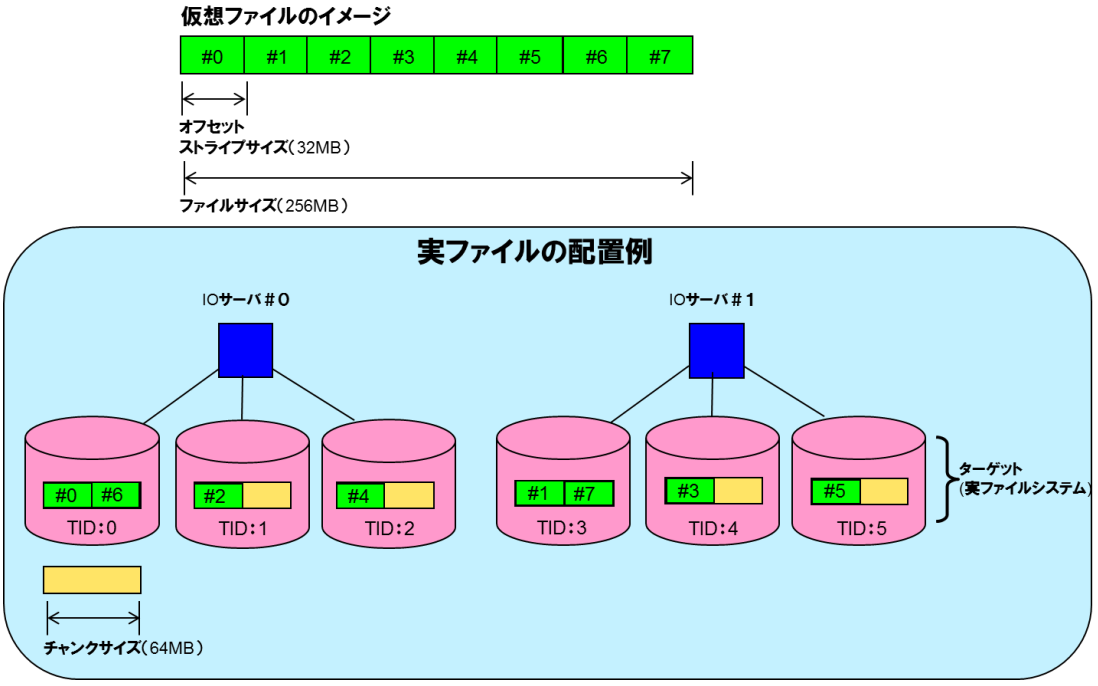


図 11-6　形式２における実ファイルの配置例

- ディレクトリ

  (例)形式1

```
$ scatefs_getfinfo -h /mnt/scatefs/dir001
format    : non stripe format
stripesize :          512.0M
chunksize  :          512.0M
```

  ※形式1の場合、ストライプサイズ、チャンクサイズは同じ値になります。

  (例)形式2

```
$ scatefs_getfinfo -h /mnt/scatefs/dir002
format    :    stripe format
stripesize :           32.0M
chunksize  :            1.0G
```

  ※ディレクトリを対象とした詳細表示オプション(-v)は無効となります。

11.6 ScaTeFS InfiniBand 高速IOライブラリの使用方法

11.6.1 ScaTeFS IB ライブラリの使用方法

下記の環境変数を設定してアプリケーションを実行することで、ScaTeFS IBライブラリを使用できます。これらの環境変数を.bashrcや.cshrcに設定することは推奨しておりません。後述の設定例のようにコマンド実行時のコマンドライン、またはジョブのスクリプト内で設定するようにしてください。

- LD_PRELOAD

  ライブラリパス(/lib64/libscatefsib.so.1)を指定してください。これを設定することで、アプリケーションを改造することなく、ユーザ空間を介した軽量で高速なIBによるIOを実行することができます。

- SCATEFS_LOG_DIR

  ライブラリのログファイルを出力するディレクトリを指定します。各ユーザのホーム配下のディレクトリ等プログラム実行者の書き込み権があるディレクトリの絶対パスを指定してください(ディレクトリは予め作成してください)。本環境変数を指定しない場合、実行プログラムのカレントディレクトリに出力されるので、後にログファイルの探索が困難になる場合があります。そのため、必ず本環境変数を指定してください。なお、ログファイルは正常運用時には出力しません。何らかのエラー等調査すべき事象発生時にのみ出力します。

ログファイルは後の解析のため消さないようにしてください。ログファイル名はlibscatefsib.<実行プロセスのPID>です。

アプリケーションの実行方法によって環境変数の設定方法が異なります。以下はcpコマンドを各実行方法で実行する場合の環境変数の設定例です。

- コマンドラインから実行する場合

  環境変数をコマンドラインから設定します。

```
$ LD_PRELOAD=/lib64/libscatefsib.so.1 SCATEFS_LOG_DIR=/home/user/log cp fileA fileB
```

- シェルスクリプトで実行する場合

  シェルスクリプト内で環境変数を設定します。NQSVのMPIジョブスクリプトとして実行する場合は設定が異なりますので、下記を参照してください。

```
#!/bin/bash
export LD_PRELOAD=/lib64/libscatefsib.so.1
export SCATEFS_LOG_DIR=/home/user/log
cp fileA fileB
```

- NQSVのMPIジョブスクリプトで実行する場合

  mpirunの-xオプションで環境変数を設定してください。設定方法は下記のスクリプトの記述例を参考にしてください。スクリプト内の記述で「export LD_PRELOAD=/lib64/libscatefsib.so.1」のように指定してもスレーブノードまでは引き継がれないのでご注意ください。

  以下はsampleプログラムを実行する場合の記述例です。#PBSの設定やmpirunに指定する環境変数${NQSII_MPIOPTS}については、NQSVのマニュアルをご参照ください。

```
#!/bin/bash
#PBS -T openmpi
#PBS -b 2
#PBS -l cpunum_job=4
#PBS -l elapstim_req=3600
mpirun ${NQSII_MPIOPTS} -npernode 1 -np 2 ¥
-x LD_PRELOAD=/lib64/libscatefsib.so.1 ¥
-x SCATEFS_LOG_DIR=/home/user/log /home/user/sample
```

11.6.2 ScaTeFS VE ダイレクト IB ライブラリの使用方法

下記の環境変数を設定してアプリケーションを実行することで、ScaTeFS VEダイレクトIBライブラリを使用できます。これらの環境変数を.bashrcや.cshrcに設定することは推奨しておりません。後述の設定例のようにコマンド実行時のコマンドライン、またはジョブのスクリプト内で設定するようにしてください。

- VE_LD_PRELOAD

  ライブラリ名(libscatefsib.so.1)を指定してください。これを設定することで、アプリケーションを改造することなく、ユーザ空間を介した軽量で高速なIBによるIOを実行することができます。

- SCATEFS_LOG_DIR

  ライブラリのログファイルを出力するディレクトリを指定します。設定上の注意や出力ファイルはScaTeFS IBライブラリと同じです。11.6.1を参照してください。

アプリケーションの実行方法によって環境変数の設定方法が異なります。以下はプログラムa.outを各実行方法で実行する場合の環境変数の設定例です。

- コマンドラインから実行する場合

  環境変数をコマンドラインから設定します。

```
$ VE_LD_PRELOAD=libscatefsib.so.1 ./a.out
```

- シェルスクリプトで実行する場合

  環境変数をシェルスクリプト内で設定します。

```
#!/bin/bash
export VE_LD_PRELOAD=libscatefsib.so.1
export SCATEFS_LOG_DIR=/home/user/logdir
./a.out
```

- NQSVのMPIジョブスクリプトで実行する場合

  上記の「シェルスクリプトで実行する場合」と同様に、環境変数をMPIジョブスクリプト内で設定します。

  また、必ず--use-hcaに必要なHCA数を指定してください。指定しない場合、IOがエラーとなり失敗するのでご注意ください。

  #PBSの設定については、NQSVのマニュアルをご参照ください。

```
#!/bin/sh
#PBS -T necmpi
#PBS -b 2
#PBS --venum-lhost=1
#PBS --use-hca=2
export VE_LD_PRELOAD=libscatefsib.so.1
export SCATEFS_LOG_DIR=/home/user/logdir

mpirun -ppn 1 mpi_prog
```

11.6.3 プログラミングのポイント

以下はScaTeFS VEダイレクト IBライブラリを使用するプログラムにおける、プログラミングのポイントです。

- 最適なIO性能を実現するためのポイント
  - 大きなサイズでのread(2)/write(2)呼び出しをお勧めします(1MB以上)

    小さなサイズで多数回read(2)/write(2)するのではなく、可能な限り1MB以上のサイズでまとめてread(2)/write(2)することで最適な性能となります。
  - 不要なstat系システムコール(stat(2)/lstat(2)/fstat(2))の呼び出しを避けることをお勧めします

    read(2)/write(2)を連続して呼び出す場合は、合間では可能な限りstat系システムコールを呼ばず、連続したread(2)/write(2)完了後に呼び出すことで最適な性能となります。
- VE-VH間や異なるVE間で動作するプロセス同士のファイルデータの整合性確保

  NFSで異なるクライアント間で同一ファイルにアクセスする場合と同様に、VE-VH間や異なるVE間で動作するプロセス間でファイルデータの整合性を確保しながら同一ファイルにアクセスする場合は、ファイルロック(flock(2), fcntl(2)のF_SETLK)を使用して同期を取りながらアクセスする必要があります。

11.6.4 性能チューニング用環境変数

ScaTeFS IBライブラリ/ScaTeFS VEダイレクトIBライブラリで共通のチューニングです。

以下の環境変数によりデータ転送サイズをチューニング可能です。read(2)/write(2)に指定するバッファサイズが大きい場合、これらの値を大きくすることによって転送処理が効率化され性能向上が期待できます。ただし、値を大きくすることで1要求に対するIOサーバの負荷が高くなり、多数プロセスが同時にREAD/WRITEする場合に性能が低下する場合があるのでご注意ください。基本的には、mountコマンドのrsize/wsizeに指定したデータ転送サイズと同

じ値に設定することを推奨します。

これらの環境変数はScaTeFS IBライブラリによるREAD/WRITEに対してのみ有効です。ライブラリを使用しない従来のカーネルによるREAD/WRITEのデータ転送サイズは、mountコマンドのrsize/wsizeオプションの設定に従います。

表 11-1　rsize/wsize オプション概要

設定値	説明	最小値	最大値	デフォルト
SCATEFS_WSIZE	IOサーバへ一度に転送するWRITEデータのサイズ(KB)。	4	4096	1024
SCATEFS_RSIZE	IOサーバから一度に転送されるREADデータのサイズ(KB)。	4	4096	1024

以下の環境変数で、IOサーバからの応答検知を高速に行うモードのON/OFFを指定します。頻繁にread(2)/write(2)を実行するプログラムにおいて性能向上が期待できます。ただし、ONの場合はOFFの場合と比較してIO中のCPU使用率が高くなります。

表 11-2　cqpollhow オプション概要

設定値	説明	最小値	最大値	デフォルト
SCATEFS_CQPOLLHOW	IOサーバからの応答検知を高速に行うモードのON/OFF。 ON:0, OFF:1	0	1	1

11.6.5 ストライプフォーマットによる性能向上

ScaTeFS IBライブラリ/ScaTeFS VEダイレクトIBライブラリで共通の設定です。

read(2)/write(2)に指定するIOサイズが複数のストライプ(またはチャンク)に跨っている場合、ライブラリはストライプ毎に並列にIOサーバにIO要求を発行します(並列数はIOサーバの数と同数)。図 11-7にIO要求が並列に発行されるイメージを記載します。これにより、同時に複数IOサーバに要求を発行することができるので、IOの処理が効率化できプロセス単体の性能向上が期待できます。IOサイズがストライプサイズより大きく(理想的にはストライプサイズの倍数)なるように、ストライプサイズ、もしくはアプリケーションのIOサイズを設定することで、複数IOサーバに並列に要求を発行します。ただし、小さすぎるストライプサイズではIOサーバ間のデータ転送サイズが小さくなることでデータ転送処理が非効率となり性能が低下する場合があります。1MB以上のストライプサイズを指定することが望ましいです。

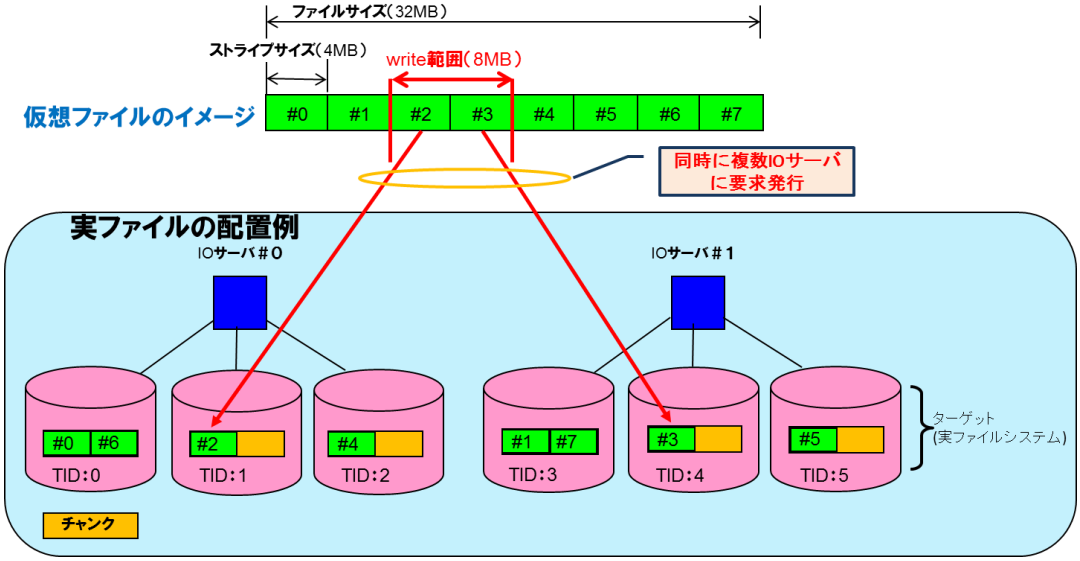


図 11-7　ストライプサイズを設定したファイルに対する IO

11.6.6 NEC Fortran のプログラムの性能チューニング

ScaTeFS VE ダイレクト IB ライブラリを使用する場合のチューニングです。

比較的小さいレコード(512KB 未満)を多数回 READ/WRITE するプログラムの場合、入出力処理が準備する I/O バッファのサイズ(VE_FORT_SETBUF) を ScaTeFS のデータ転送サイズ(SCATEFS_RSIZE, SCATEFS_WSIZE) と同じサイズに拡大することで効率的に IO が処理され性能向上が期待できます。SCATEFS_RSIZE と SCATEFS_WSIZE に異なる値を設定している場合は、大きい方の値を VE_FORT_SETBUF に設定してください。

I/O バッファのサイズは環境変数 VE_FORT_SETBUF で設定可能です。詳細は「SX-Aurora TSUBASA Fortran コンパイラユーザーズガイド」を参照ください。ScaTeFS のデータ転送サイズの設定方法、およびデフォルト値については 11.6.4 を参照ください。

HPC 領域で通常使用されるような、大きなサイズ(512KB 以上)のレコードの READ/WRITE を主に行うプログラムでは VE_FORT_SETBUF の設定を変更する必要はありません。

11.6.7 統計情報

ScaTeFS IBライブラリ/ScaTeFS VEダイレクトIBライブラリで共通の設定です。

環境変数SCATEFS_STATISTICS_ON に 1 を設定することで、実行プロセスの統計情報ファイルが出力されます。統計情報ファイルの出力場所は、環境変数SCATEFS_STATISTICS_DIRで設定できます。指定しない場合は、カレントディレクトリに出力されます。以下はコマンドラインで設定する場合のイメージです。

```
# LD_PRELOAD=/lib64/libscatefsib.so.1 SCATEFS_STATISTICS_ON=1  ¥
```

```
SCATEFS_STATISTICS_DIR=/home/user/log/   dd   if=/dev/zero   of=/mnt/scatefs/testfile
bs=1M count=1
```

SCATEFS_STATISTICS_DIRに設定したディレクトリ配下にlibscatefs_stat.<PID> という名前の統計情報ファイルが作成されます。統計情報は、統計情報ファイルをscatefs_ibstat(1)の引数に指定して実行することで確認できます。以下の出力ではScaTeFS IBライブラリにより1048576バイト(SIZE_TOTAL)のデータがWRITE処理されたことを示します。出力内容の詳細についてはscatefs_ibstat(1)のmanを参照ください。

```
# scatefs_ibstat ./stat/libscatefs_stat.9012
Pid: 9012
Time: Tue Jul 19 10:41:19 2016
REQUEST      COUNT TAT_TOTAL TAT_AVE    SIZE_TOTAL SIZE_AVE      OK      NG
WRITE            1        2       2       1048576  1048576       1       0
READ             0        0       0             0        0       0       0
COMMIT           1        8       8             0        0       1       0
write(libc)      0        0       0             0        0       0       0
read(libc)       0        0       0             0        0       0       0
```

IOサイズが1MB未満の場合、ライブラリ内で自動的にカーネルIO方式に切り替えて処理されます。

カーネルIOが行われた場合は、下記のようにwrite(libc)に計上されます。

```
# scatefs_ibstat ./stat/libscatefs_stat.9015
Pid: 9015
Time: Tue Jul 19 10:46:50 2016
REQUEST      COUNT TAT_TOTAL TAT_AVE    SIZE_TOTAL SIZE_AVE      OK      NG
WRITE            0        0       0             0        0       0       0
READ             0        0       0             0        0       0       0
COMMIT           0        0       0             0        0       0       0
write(libc)      1        1       1       1048575  1048575       1       0
read(libc)       0        0       0             0        0       0       0
```

11.6.8 ジョブ実行失敗時の対応

ScaTeFS IBライブラリ/ScaTeFS VEダイレクトIBライブラリで共通の設定です。

close時ディスク同期モードを使用している場合は、障害によりIOサーバがフェイルオーバしたときに、実行中のジョブが発行するread/write系システムコールまたはclose(2)がETIMEDOUTでエラーとなります。また、標準エラー出力、またはNQSVが出力する標準エラ

ー出力ファイルに以下のメッセージを出力します。

```
ScaTeFS failed to write: process(/bin/cp)  file(4362917)
```

※4362917はファイルのinode番号

これらのエラーを検出したジョブはファイルへの書き込みが正常に完了していない可能性があるため、当該ジョブを再実行するようにしてください。

11.6.9 メモリ使用量

ScaTeFS InfiniBand 高速 IO ライブラリ使用時は、未使用の場合と比較してプロセスあたりで下記表に示すメモリ量を追加で使用します。SFA7990XE を使用時は、SFA7990XE の特徴を活かし高速に IO を処理する仕組みにより、Express5800 の IO サーバを使用時より多くのメモリを使用します。Express5800 の IO サーバと SFA7990XE の両方に対して IO を行うプロセスでは、SFA7990XE だけを使用する場合と同じメモリ量となります。

表 11-3 ライブラリ使用時の追加のメモリ使用量

Express5800 の IO サーバを使用時	SFA7990XE を使用時
200MB	460MB

ScaTeFS IB ライブラリ使用時は、VH を含むスカラマシン上のメモリを使用します。ScaTeFS VE ダイレクト IB ライブラリ使用時は VE のメモリを使用します。

第12章　諸元

表 12-1　諸元表

項目	最大数
1ファイルシステムを構成できる最大IOサーバ数	256(128ペア)
1ファイルシステムを構成できる最大IOターゲット数(実ファイルシステム数)	1024
同一システム内に作成できるファイルシステム数	20
最大ファイルサイズ	64PB (チャンクサイズ:4GB、ファイルフォーマット:形式1を想定)
最大ファイルシステムサイズ	500PB(IOターゲット：1024を想定)
最大ファイル数	２兆ファイル(IOターゲット：1024を想定)
最大ディレクトリエントリ数	制限なし(500万ファイルまでの実績あり)
最大ファイル名長	255バイト
最大パス名長	1024バイト
1ファイルシステムに対して ScaTeFS InfiniBand 高速IOライブラリを同時に使用できるプロセス数	約35,000プロセス HCAの資源による制限です。IBを使用する他のプログラムの資源の利用状況により変わります。
1クライアントで ScaTeFS InfiniBand 高速IOライブラリを同時に使用できるプロセス数	約900プロセス ● SX-Aurora TSUBASAの場合は、1クライアント上のVHとVEで動作するプロセス数の合計の最大数となります。 ● HCAの資源による制限です。IBを使用する他のプログラムの資源の利用状況により変わります。

付録 A　CLUSTERPRO のクラスタ構成情報作成手順(オフラインバージョン)

本手順書では、以下の CLUSTERPRO のツールを使用し、IO サーバ構築前にクラスタ構成情報を事前に作成する手順について記載します。

【標準モデル向け IO サーバ v4+以降】

CLUSTERPRO X Cluster WebUI Offline

【標準モデル向け IO サーバ v1,v3,v4】

オフライン版 CLUSTERPRO builder

この手順書で実施する作業は、「NEC Scalable Technology File System(ScaTeFS)運用の手引」マニュアルの「5.4.1.1 設定ファイルを作業用 PC へ転送」で入手するよう記載がある設定ファイル自体の作成作業となります。本作業終了後は、「5.4.1.2 IO サーバ間インタコネクト用ポートのネットワーク設定確認」から CLUSTERPRO の設定を開始してください。

以下の順序で CLUSTERPRO のクラスタ構成情報を作成していきます。

1. CLUSTERPRO ツールのインストール
2. CLUSTERPRO ツールの起動
3. クラスタ構成情報作成
   - 3.1 クラスタの作成
   - 3.2 フェイルオーバグループの作成
   - 3.3 モニタリソースの作成
   - 3.4 モニタリソース異常時の回復動作設定
   - 3.5 クラスタプロパティの変更

本手順書は CLUSTERPRO のマニュアル「CLUSTERPRO X for Linux インストール&設定ガイド」を補完する位置づけですので、適宜同マニュアルを参照ください。本手順書では以下を参照ポイントとしています。

【CLUSTERPRO X Cluster WebUI Offline】

マニュアル（1）: 第 6 章クラスタ構成情報を作成する の 2 ノードクラスタ構成情報の作成手順

マニュアル（2）: 第 6 章クラスタ構成情報を作成する の クラスタ構成情報を保存する

【オフライン版 CLUSTERPRO builder】

マニュアル（1）: 第 3 章 CLUSTERPRO をインストールする のオフライン版 CLUSTERPRO

Builder をインストールするには

マニュアル（2）: 第 5 章クラスタ構成情報を作成する の 2 ノードクラスタ構成情報の作成手順

マニュアル（3）: 第 5 章クラスタ構成情報を作成する のクラスタ構成情報を保存する

また、本手順書で使用している CLUSTERPRO のツールは以下となります。

CLUSTERPRO のサイトから入手します。

【CLUSTERPRO X Cluster WebUI Offline】

4.3.2-210913-1

【オフライン版 CLUSTERPRO builder】

アプリケーションのバージョンによりデフォルト値が異なる可能性がありますので、いずれかのバージョンのアプリケーションを使用してください。

clusterprobuilder-3.2.0-1.linux.i686.exe

clusterprobuilder-3.3.5-1.linux.i686.exe

A.1　はじめに

作業を始める前に事前に決めておく情報があります。これらの情報を決めた上で作業を始めてください。

- IOサーバ名
- クラスタを構成する2台のIOサーバ間のインタコネクト用IPアドレス
- ファイルシステムポート(10GbE/IB)のフローティングIPアドレス(FIP)
- 各種リソース名

これら各種リソースとリソース間の対応表を別紙に添付していますので参照ください。本表で記載している各種リソース名は、推奨する命名規則で例示しています。特段の理由がない限り、本命名規則でリソース名を決めてください。

なお､ハートビート領域用のパーティションのデバイス名も決める必要がありますが､本名称はSPSインストール後に決定されるため、本手順書での作業完了後の、「NEC Scalable Technology File System(ScaTeFS)運用の手引」マニュアルの「5.4.5 クラスタプロパティ」で設定することになります。

A.2　CLUSTERPROツールのインストール

【CLUSTERPRO X Cluster WebUI Offline】

CLUSTERPROサイトの手順書「Cluster WebUI Offline 利用ガイド」を参照してください。

【オフライン版CLUSTERPRO builder】

マニュアル（1）を参照してください。

A.3　CLUSTERPROツールの起動

【CLUSTERPRO X Cluster WebUI Offline】

CLUSTERPROサイトの手順書「Cluster WebUI Offline 利用ガイド」を参照してください。

【オフライン版CLUSTERPRO builder】

マニュアル（1）を参照してください。

A.4　クラスタ構成情報作成

「クラスタ生成ウィザード」を使用してクラスタ構成情報を作成します。

【CLUSTERPRO X Cluster WebUI Offline】

マニュアル（1）を参照してください。

【オフライン版CLUSTERPRO builder】

マニュアル（2）を参照してください。

以降の手順では、未記載の項目はデフォルトのままとしてください。

A.5　クラスタの作成

クラスタを作成します。

A.6　クラスタの追加

【CLUSTERPRO X Cluster WebUI Offline】

「クラスタ生成ウィザード」をクリックして、ウィザードを開始します。クラスタ生成ウィザード画面のクラスタの定義はデフォルトのままで、[次へ] をクリックします。

【オフライン版CLUSTERPRO builder】

メニューバーの「ファイル」の「クラスタ生成ウィザード」をクリックして、確認画面で「標準版クラスタ生成ウィザードを開始する」をクリックします。クラスタ生成ウィザード画面のクラスタの定義はデフォルトのままで、[次へ] をクリックします。

A.7　サーバの追加

クラスタ生成ウィザード画面のサーバの定義では、クラスタを構成する2台のIOサーバを追加します。以下の説明では IOサーバをiosv00,iosv01と記載します。

サーバの定義一覧で 追加 をクリックします。

サーバ追加画面で以下の項目を設定します。マスタサーバとなります。

サーバ名

iosv00

再度、サーバの定義一覧で 追加 をクリックします。

サーバ追加画面で以下の項目を設定します。

サーバ名

iosv01

A.8　ネットワーク構成の設定

クラスタを構成するIOサーバ間のネットワーク構成を設定します。

インタコネクト一覧で 追加 をクリックします。

優先度1の行の項目を設定します。

種別

カーネルモード

iosv00

IOサーバ間インタコネクト用IPアドレス

iosv01

IOサーバ間インタコネクト用IPアドレス

インタコネクト一覧で 追加 をクリックします。

優先度2の行の項目を設定します。

種別

DISK

iosv00

CLUSTERPROのハートビート領域用のパーティションのデバイス名

iosv01

CLUSTERPROのハートビート領域用のパーティションのデバイス名

※ハートビート領域用のパーティションのデバイス名は、本手順書での作業完了後に設定してください。

A.9　ネットワークパーティション解決処理の設定(NP解決)

設定を行わないで次を進めてください。

A.10 フェイルオーバグループの作成

クラスタを構成する IO サーバで動作するフェイルオーバグループを作成します。 以下の説明では iosv00 で動作するフェイルオーバグループを failover1 、iosv01 で動作するフェイルオーバグループを failover2 と記載します。

最初に iosv00 で動作するフェイルオーバグループ failover1 を作成しますので、A.11 から A.15 を実施してください。A.11 から A.15 が完了しましたら、iosv01 で動作するフェイルオーバグループ failover2 を作成しますので、再度 A.11 から A.15 を実施してください。

なお、フェイルオーバグループ failover1 と フェイルオーバグループ failover2 で異なる設定を行う項目については、各章の中で [failover1]、[failover2]と分けて明記しています。

A.11 フェイルオーバグループの追加

グループ画面のグループ一覧で 追加 をクリックします。

グループの定義画面で以下の項目を設定します。

名前

[failover1]

failover1

[failover2]

failover2

起動可能サーバ一覧画面で すべてのサーバでフェイルオーバ可能 のチェックを外します。

利用可能なサーバで 以下の順序で IO サーバを選択して 追加 をクリックします。

※追加する順番が重要です。

[failover1]

iosv00

iosv01

[failover2]

iosv01

iosv00

グループ属性の設定画面で以下の項目をデフォルトから変更します。

フェイルバック属性

自動フェイルバック

A.12 グループリソース（フローティング IP リソース）の追加

IO サーバのネットワーク設定のファイルシステムポート(10GbE, IB)の IP アドレスを設定します。

なお、IO サーバのネットワーク設定のファイルシステムポート数で追加するリソース数が異なりますが、本手順書では 10GbE と IB を併用する場合で、10GbE の４つの FIP(fip1, fip2, fip3, fip4)と IB の２つの FIP(fip_ib1, fip_ib2)について、[failover1]へ fip1, fip2, fip_ib1 [failover2]へ fip3, fip4, fip_ib2 を追加する例を記載します。

(10GbE のみを使用する場合は下記例において fip1, fip2, fip3, fip4 のみを追加します。また、IB のみを使用する場合は fip_ib1, fip_ib2 のみを追加します)

グループリソース画面のグループリソース一覧で 追加 をクリックします。

グループリソースの定義画面で以下の項目を設定します。

タイプ

フローティング IP リソース (Floating IP resource)

名前

[failover1]

fip1

fip2

fip_ib1

[failover2]

fip3

fip4

fip_ib2

依存関係画面ではデフォルトのまま設定を行わないで次を進めてください。

復旧動作画面ではデフォルトのまま設定を行わないで次を進めてください。

詳細画面(共通タブ)で以下の項目を設定します。

IP アドレス

IO サーバのネットワーク設定のファイルシステムポート(10GbE, IB)の IP アドレス

設定例

10.0.1.1/25%bond0.12

調整ボタンをクリックします。

フローティング IP リソース調整プロパティのパラメータタブで以下の項目を設定します。

bonding インターフェースの場合にのみ必要な設定です。

NIC Link Down を異常と判定する

チェックボックスをオンにします。

A.13 グループリソース（ボリュームマネージャリソース）の追加

LVM 構成の設計で検討した VG について、グループリソースとして追加します。IO サーバで IO ターゲットのデータ/メタデータ領域で使用する VG を設定します。

IO ターゲット数分のグループリソース(データ/メタデータ領域)を作成します。

グループリソース画面のグループリソース一覧で 追加 をクリックします。

グループリソースの定義画面で以下の項目を設定します。

タイプ

ボリュームマネージャリソース (Volume manager resource)

名前

[failover1]

volmgr_d_01,volmgr_d_02,・・・,volmgr_d_[n]

volmgr_c_01,volmgr_c_02,・・・,volmgr_c_[n]

[failover2]

volmgr_d_[n+1],volmgr_d_[n+2],・・・,volmgr_d_[n+n]

volmgr_c_[n+1],volmgr_c_[n+2],・・・,volmgr_c_[n+n]

※[n]は LVM 構成で設計した IO サーバごとの VG 数です。

依存関係画面で以下の項目を設定します。

既定の依存関係に従う

チェックを外します

利用可能なリソースのフローティング IP アドレスリソースを選択して追加 をクリックします。

[failover1]

fip1,fip2, fip_ib1

[failover2]

fip3,fip4, fip_ib2

復旧動作画面ではデフォルトのまま設定を行わないで次を進めてください。

詳細画面(共通タブ)で以下の項目を設定します。

ターゲット名

[failover1]

vg_data01,vg_data02,・・・,vg_data[n]

vg_ctrl01,vg_ctrl02,・・・,vg_ctrl[n]

[failover2]

vg_data[n+1],vg_data[n+2],・・・,vg_data[n+n]

vg_ctrl[n+1],vg_ctrl[n+2],・・・,vg_ctrl[n+n]

A.14 グループリソース (ディスクリソース) の追加

IO ターゲットのデバイスのマウント/アンマウントを行うリソースを追加します。

※IO ターゲット数分のグループリソース(データ/メタデータ領域)を作成します。

グループリソース画面のグループリソース一覧で 追加 をクリックします。

グループリソースの定義画面で以下の項目を設定します。

タイプ

ディスクリソース (Disk resource)

名前

[failover1]

disk_d_01,disk_d_02,・・・,disk_d_[n]

disk_c_01,disk_c_02,・・・,disk_c_[n]

[failover2]

disk_d_[n+1],disk_d_[n+2],・・・,disk_d_[n+n]

disk_c_[n+1],disk_c_[n+2],・・・,disk_c_[n+n]

※[n]は LVM 構成で設計した IO サーバごとの VG 数です。

依存関係画面で以下の項目を設定します。

既定の依存関係に従う

チェックを外します

利用可能なリソースのフローティング IP アドレスリソース、対象のボリュームマネージャリソースを選択して 追加 をクリックします。

[failover1]

fip1,fip2,fip_ib1,対象のボリュームマネージャリソース(たとえば disk_d_01 に対しては volmgr_d_01)

[failover2]

fip3,fip4,fip_ib2,対象のボリュームマネージャリソース(たとえば disk_d_13 に対しては volmgr_d_13)

復旧動作画面では設定を行わないで次を進めてください。

詳細画面(共通タブ)で以下の項目を設定します。

ディスクのタイプ

lvm

ファイルシステム

【標準モデル向け IO サーバ v4+以降】

ext4 または xfs

※5.1.1 で設計したファイルシステムタイプを指定します。

【標準モデル向け IO サーバ v1,v3,v4】

ext4

デバイス名

LV Path

※VG 名と LV 名からデバイス名を設定可能です(/dev/VG 名/LV 名)。

[failover1]

データ領域

/dev/vg_data01/lv_data01,・・・,/dev/vg_data[n]/lv_data[n]

メタデータ領域

/dev/vg_ctrl01/lv_ctrl01,・・・,/dev/vg_ctrl[n]/lv_ctrl[n]

[failover2]

データ領域

/dev/vg_data[n+1]/lv_data[n+]1,・・・,/dev/vg_data[n+n]/lv_data[n+n]

メタデータ領域

/dev/vg_ctrl[n+1]/lv_ctrl[n+1],・・・,/dev/vg_ctrl[n+n]/lv_ctrl[n+n]

マウントポイント

IO ターゲット作成で指定したデバイスのマウントポイント

データ領域

/mnt/iot/X/data

メタデータ領域

/mnt/iot/X/ctrl

※X は IO ターゲット ID を指定します。

【標準モデル向け IO サーバ v4+以降】

ファイルシステムへext4を指定した場合、詳細画面(共通タブ)の調整ボタンをクリックします。

ディスクリソース調整プロパティの Fsck タブで以下の項目を設定します。

Mount 実行前の fsck アクション

実行しない

Mount 失敗時の fsck アクション

実行する(デフォルトのまま)

A.15 グループリソース（EXEC リソース）の追加

IO サーバで実行するリソース(ScaTeFS サーバ、ルーティング)を追加します。

- ルーティング

グループリソース画面のグループリソース一覧で 追加 をクリックします。

グループリソースの定義画面で以下の項目を設定します。

タイプ

EXEC リソース（EXEC resource)

名前

[failover1]

exec_route1

[failover2]

exec_route2

依存関係画面で以下の項目を設定します。

既定の依存関係に従う

チェックを外します

利用可能なリソースのフローティングIPアドレスリソースを選択して追加 をクリックします。

[failover1]

fip1,fip2,fip_ib1

[failover2]

fip3,fip4,fip_ib2

復旧動作画面ではデフォルトのまま設定を行わないで次を進めてください。

詳細画面で以下の項目を設定します。
ユーザアプリケーションを選択します。
スクリプト一覧で編集をクリックします。
アプリケーション・パスの入力で以下の項目を設定します。
開始
/opt/scatefs/script/exec_route.sh

- ScaTeFSサーバ

グループリソース画面のグループリソース一覧で 追加 をクリックします。
グループリソースの定義画面で以下の項目を設定します。
タイプ
EXEC リソース (EXEC resource)
名前
[failover1]
exec1
[failover2]
exec2

依存関係画面で以下の項目を設定します。
既定の依存関係に従う
チェックを外します
利用可能なリソースの各リソースを選択して 追加 をクリックします。
※表示されるすべてのリソースを追加します。
[failover1]
fip1,fip2,fip_ib1,ボリュームマネージャリソース,ディスクリソース,EXECリソース(ルーティング)
[failover2]
fip3,fip4,fip_ib2,ボリュームマネージャリソース,ディスクリソース,EXECリソース(ルーティング)

復旧動作画面ではデフォルトのまま設定を行わないで次を進めてください。

詳細画面で以下の項目を設定します。

ユーザアプリケーションを選択します。

スクリプト一覧で編集をクリックします。

アプリケーション・パスの入力で以下の項目を設定します。

開始

/opt/scatefs/script/start.sh

終了

/opt/scatefs/script/stop.sh

A.16 モニタリソースの作成

ここからは、フェイルオーバグループごとではなく IO サーバごとにモニタリソースを作成します。

A.17 モニタリソース (ディスクモニタ) 追加

ディスクリソースのモニタリソースを追加します。

IO サーバごとにメタデータ領域のディスクリソースの最初の 1 つ(のみ)を追加します。

iosv00

disk_c_01

iosv01

disk_c_[n+1]

※[n]は LVM 構成で設計した IO サーバごとの VG 数です。

モニタリソース画面のモニタリソース一覧で 追加 をクリックします。

モニタリソースの定義画面で以下の項目を設定します。

タイプ

ディスク RW 監視 (Disk monitor)

名前

iosv00

diskw_c_01

iosv01

diskw_c_[n+1]

監視(共通)画面で以下の項目を設定します。

監視タイミング

活性時を選択します。

対象リソース

iosv00

参照をクリックして、disk_c_01 を選択します。

iosv01

参照をクリックして、disk_c_[n+1]を選択します。

監視(固有)画面の共通タブで以下の項目を設定します。

監視方法

READ(O_DIRECT)

監視先

iosv00

/dev/vg_ctrl01/lv_ctrl01

iosv01

/dev/vg_ctrl[n+1]/lv_ctrl[n+1]

回復動作画面で以下の項目を設定します。

回復対象

iosv00

参照をクリックして、disk_c_01 を選択します。

iosv01

参照をクリックして、disk_c_[n+1]を選択します。

A.18 モニタリソース（カスタムモニタ）追加

EXEC リソース(ScaTeFS サーバ)のモニタリソースを追加します。

モニタリソース画面のモニタリソース一覧で 追加 をクリックします。

モニタリソースの定義画面で以下の項目を設定します。

タイプ

カスタム監視 (Custom monitor)

名前

iosv00

genw1

iosv01

genw2

監視(共通)画面で以下の項目を設定します。

インターバル

15

監視タイミング

活性時を選択します。

対象リソース

iosv00

参照をクリックして、exec1 を選択します。

iosv01

参照をクリックして、exec2 を選択します。

監視(固有)画面で以下の項目を設定します。

ユーザアプリケーションを選択します。

ファイル

iosv00

/opt/scatefs/script/is_exec1_ios_running.sh

iosv01

/opt/scatefs/script/is_exec2_ios_running.sh

監視タイプ

同期を選択します。

回復動作画面で以下の項目を設定します。

回復動作

回復対象を再起動、効果がなければフェイルオーバ実行

回復対象

iosv00

参照をクリックして、exec1 を選択します。

iosv01

参照をクリックして、exec2 を選択します。

以上でモニタリソース (カスタムモニタ) 追加は完了ですが、プロセス名モニタリソースを設定

している場合、以下の手順でプロセス名モニタリソースの設定は削除してください。

モニタリソース画面のモニタリソース一覧で psw1,psw2 をそれぞれ右クリックして、モニタリソースの削除をクリックします。 [確認]ダイアログが表示されるので、[はい]ボタンをクリックしてください。

A.19 モニタリソース (ボリュームマネージャモニタ) 設定変更

自動で作成されたすべてのボリュームマネージャモニタの設定を変更します。

モニタリソース画面のモニタリソース一覧で volmgrwX を選択して、プロパティ をクリックします。

監視(共通)画面で以下の項目を設定します。

タイムアウト

240

リトライ回数

3

A.20 モニタリソース (ユーザ空間モニタ) 設定変更

自動で作成されたユーザ空間モニタの設定を変更します。

モニタリソース画面のモニタリソース一覧で userw を選択して、
プロパティ をクリックします。

監視(固有)画面で以下の項目を設定します。

監視方法

keepalive

タイムアウト発生時動作

PANIC

監視の拡張設定

以下の項目について、チェックします。

ダミーファイルのオープン/クローズ

書き込みを行う

A.21 モニタリソース（フローティングIPモニタ）設定変更

自動で作成されたフローティング IP モニタの設定を変更します。bonding インターフェースの場合にのみ必要な設定です。

モニタリソース画面のモニタリソース一覧で fipw1～fipw6 をそれぞれ選択して、
プロパティをクリックします。

監視(固有)画面で以下の項目を設定します。
NIC Link Up/Down を監視する
チェックボックスをオンにします。

A.22 モニタリソース（IPモニタ)の追加(10GbE)

IP モニタリソースを追加します。
フローティング IP アドレスごとに IP モニタリソースを追加します。

モニタリソース画面のモニタリソース一覧で 追加 をクリックします。
モニタリソースの定義画面で以下の項目を設定します。
タイプ
IP 監視 (IP monitor)

名前
iosv00
ipw1
ipw2

iosv01
ipw3
ipw4

監視(共通)画面で以下の項目を設定します。
インターバル(I)
30 秒に設定します。
タイムアウト(T)

30 秒に設定します。

リトライ回数(R)

3 回に設定します。

監視タイミング

活性時(C)を選択します。

対象リソース

iosv00

ipw1

参照をクリックして、fip1 を選択します。

ipw2

参照をクリックして、fip2 を選択します。

iosv01

ipw3

参照をクリックして、fip3 を選択します。

ipw4

参照をクリックして、fip4 を選択します。

監視を行うサーバを選択する

サーバ(V)をクリックします。

独自に設定する(C)

チェックボックスをオンにします。

起動可能なサーバ

以下のように設定します。

ipw1、ipw2

iosv00

ipw3、ipw4

iosv01

監視(固有)画面の共通タブで以下の項目を設定します。

追加(D)をクリックします。

IP アドレス(I)に以下のように入力し、OK をクリックします。

ipw1

fip1 が接続されるネットワークのゲートウェイ

ipw2

fip2 が接続されるネットワークのゲートウェイ

ipw3

fip3 が接続されるネットワークのゲートウェイ

ipw4

fip4 が接続されるネットワークのゲートウェイ

※IO サーバのフローティング IP アドレスではないのでご注意ください.

回復動作画面で以下の項目を設定します。

回復動作

回復対象を再起動、効果がなければフェイルオーバ実行

回復対象

ipw1

参照(W)をクリックして、fip1 を選択します。

ipw2

参照(W)をクリックして、fip2 を選択します。

ipw3

参照(W)をクリックして、fip3 を選択します。

ipw4

参照(W)をクリックして、fip4 を選択します。

A.23 モニタリソース異常時の回復動作設定

モニタリソースを作成し、[完了] をクリックすると、以下のポップアップメッセージが表示されますので、はい をクリックします。

【CLUSTERPRO X Cluster WebUI Offline】

下記の動作を有効にしますか？

・グループの自動起動

・グループリソース活性・非活性異常時の復旧動作

・モニタリソース異常時の回復動作

【オフライン版 CLUSTERPRO builder】

モニタリソース異常時の回復動作を有効にしますか？

A.24 クラスタプロパティの変更

クラスタ(cluster)のプロパティを開きます。

タイムアウトタブを選択して以下の項目を設定します。

内部通信タイムアウト

300

監視タブを選択して以下の項目を設定します。※標準モデル向け IO サーバ v4 以降。

シャットダウン監視の監視方法

keepalive

以上で作業は完了となりますが、作成したクラスタ構成情報をファイルシステムへ保存してください。この後の IO サーバ構築で、クラスタ構成情報ファイルをインポートする際に使用します。

【CLUSTERPRO X Cluster WebUI Offline】

「設定のエクスポート」をクリックして、任意のディレクトリに保存してください。

マニュアル（２）を参照してください。

【オフライン版 CLUSTERPRO builder】

メニューバーの「ファイル」の「設定のエクスポート」をクリックして、任意のディレクトリに保存してください。

マニュアル（３）を参照してください。

表 12-2　各種リソースとリソース間の対応表

IOサーバ名		**iosv00**		**iosv01**	
インタコネクト用IPアドレス					
ハートビート用パーティションのデバイス名					
フェイルオーバグループ名		failover1		failover2	
フローティングIPアドレス	グループリソース名	fip1	fip2	fip3	fip4
	IPアドレス				
データ種別		**データ**	**メタデータ**	**データ**	**メタデータ**
ボリュームマネージャリソース	グループリソース名	volmgr_d_01 volmgr_d_02 ・・・ volmgr_d_[n]	volmgr_c_01 volmgr_c_02 ・・・ volmgr_c_[n]	volmgr_d_[n+1] volmgr_d_[n+2] ・・・ volmgr_d_[n+n]	volmgr_c_[n+1[volmgr_c_[n+2] ・・・ volmgr_c_[n+n]
	VG名	vg_data01 vg_data02 ・・・ vg_data[n]	vg_ctrl01 vg_ctrl02 ・・・ vg_ctrl[n]	vg_data[n+1] vg_data[n+2] ・・・ vg_data[n+n]	vg_ctrl[n+1] vg_ctrl[n+2] ・・・ vg_ctrl[n+n]
ディスクリソース	グループリソース名	disk_d_01 disk_d_02 ・・・ disk_d_[n]	disk_c_01 disk_c_02 ・・・ disk_c_[n]	disk_d_[n+1] disk_d_[n+2] ・・・ disk_d_[n+n]	disk_c_[n+1] disk_c_[n+2] ・・・ disk_c_[n+n]
	VG名	lv_data01 lv_data02 ・・・ lv_data[n]	lv_ctrl01 lv_ctrl02 ・・・ lv_ctrl[n]	lv_data[n+1] lv_data[n+2] ・・・ lv_data[n+n]	lv_ctrl[n+1] lv_ctrl[n+2] ・・・ lv_ctrl[n+n]
デバイス名		/dev/vg_data01/lv_data01 /dev/vg_data02/lv_data02 ・・・ /dev/vg_data[n]/lv_data[n]		/dev/vg_ctrl[n+1]/lv_ctrl[n+1] /dev/vg_ctrl[n+2]/lv_ctrl[n+2] ・・・ /dev/vg_ctrl[n+n]/lv_ctrl[n+n]	
マウントポイント		/mnt/iot/0/data /mnt/iot/1/data ・・・ /mnt/iot/[n-1]/data		/mnt/iot/[n]/ctrl /mnt/iot/[n+1]/ ctrl ・・・ /mnt/iot/[n+n-1]/ ctrl	

IOサーバ名	EXECリソース		ディスクモニタリソース名	プロセス名モニタリソース名
	ルーティング	ScaTeFSサーバ		
iosv00	exec_route1	exec1	diskw_c_01	psw1
iosv01	exec_route2	exec2	diskw_c_[n+1]	psw2

【マウントポイントパス中の IO ターゲット ID(/mnt/iot/X/)に関して】

IO ターゲット作成時にシステムで割り当てる IO ターゲット ID は、先頭の IO サーバ(iosv00)から順に 0 から割り当てます。

以下に IO サーバに IO ターゲットを n 個構築した場合の IO ターゲット ID の割り当てを記載します。

iosv00

IO ターゲット ID： 0 ～ n-1

iosv01

IO ターゲット ID：n ～ n+n-1

意図どおりに IO ターゲット ID が割り振られているか、IO ターゲット作成後、scatefs_detail -t コマンドを使用して確認してください。

コマンドの使用方法は、「NEC Scalable Technology File System(ScaTeFS)運用の手引」マニュアルの「 5.2.5 IO ターゲット作成(scatefs_addiot)」

を参照してください。

【命名規則】

グループリソース名と VG 名について、以下を推奨します。

データ領域

グループリソース名

volmgr_d_01,volmgr_d_02,・・・

VG 名

vg_data01,vg_data02,・・・

メタデータ領域

グループリソース名

volmgr_c_01,volmgr_c_02,・・・

VG 名

vg_ctrl01,vg_ctrl02,・・・

グループリソース名と LV 名について、以下を推奨します。

データ領域

グループリソース名

disk_d_01,disk_d_02,・・・

LV 名

lv_data01,lv_data02,・・・

メタデータ領域

グループリソース名

disk_c_01,disk_c_02,・・・

LV 名

lv_ctrl01,lv_ctrl02,・・・

付録 B　Windows から ScaTeFS 領域へ直接アクセスする

本手順書では、ScaTeFS 領域を Windows と共有し、Windows から ScaTeFS 領域へ直接アクセスするための環境構築について記載します。

Windows から ScaTeFS 領域へ直接アクセスするために Samba サーバを中継として使用します。以降の手順は、中継となる Samba サーバの構築手順と Windows からの接続方法について記載します。また、Samba サーバを冗長化構成にする場合については、「B.5 クラスタ構成」を参照してください。

B.1　ネットワーク構成

ネットワーク構成例については、「2.ネットワークの構築」の「2.1 構成例」を参照してください。

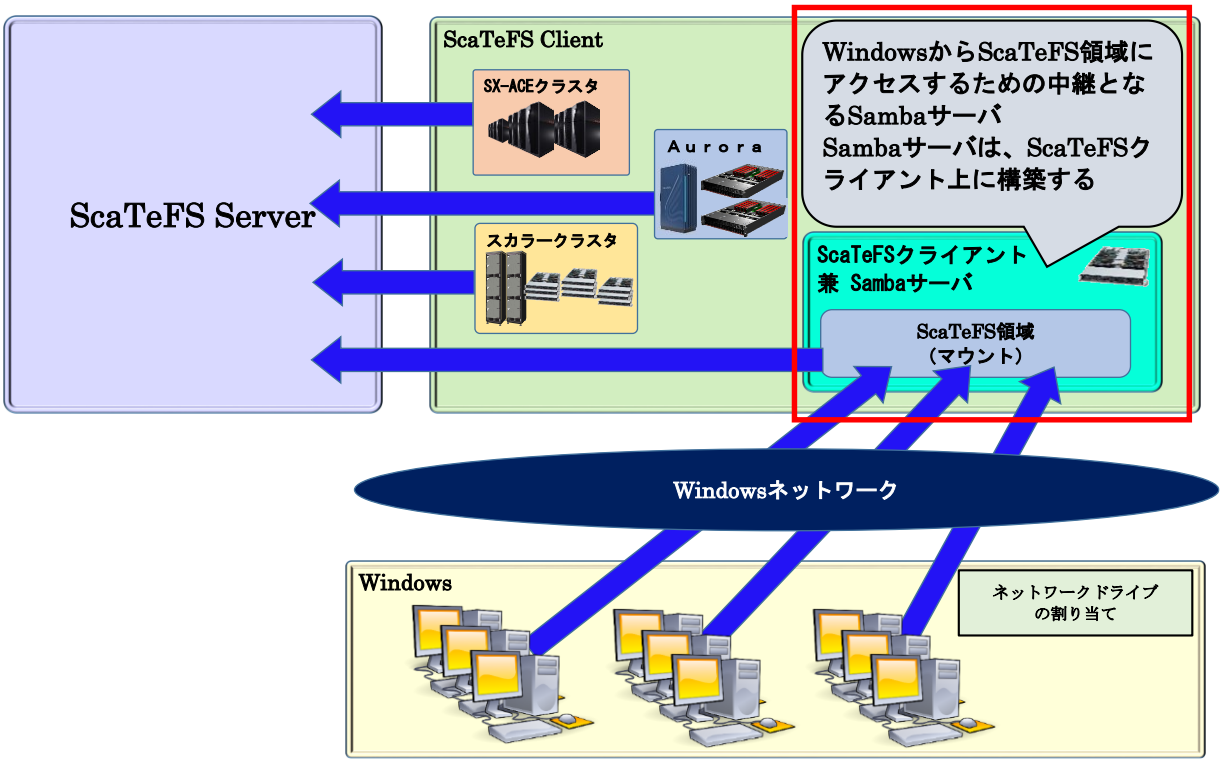


図 12-1　構成イメージ

B.2　環境構築

本作業では、ScaTeFS 領域がマウントされている ScaTeFS クライアント端末へ Samba サーバを構築し、Samba サーバのファイル共有機能により、ScaTeFS 領域を公開する設定を行います。また、Windows 端末から、Samba サーバへ接続するため設定を行います。

B.2.1　事前準備

Windows端末からアクセスできるScaTeFSクライアント端末を準備してください。

また、環境構築にあたり以下のプログラムプロダクトが必要です。

- Samba
- CLUSTERPRO X for Linux （冗長化構成とする場合に必要）

B.2.2　構築と設定

実施する設定の概要は以下のとおりです。

①Sambaのインストールと設定

B.2.1で準備したScaTeFSクライアント端末に対し、「B.3.1　Samba4インストール」から、「B.3.8　ファイアウォールの設定」を参照し、インストールと環境設定をします。

②Windows端末の設定

ScaTeFS領域へアクセスするためのWindows端末を準備してください。

「B.4　Windows端末での接続設定」を参照し、Windows端末へ設定をします。

なお、Sambaサーバを冗長化構成にする場合は、「B.5　クラスタ構成」を参照し、CLUSTERPROの設定を行ってください。

※ 使用する以下のソフトウェアについて、運用環境に合わせた設定となるように、それぞれのマニュアルを参照して設定を行ってください。

B.3　Sambaサーバの構築

以降の作業については、管理者権限 (root) でログインして実施します。

RPM 形式のパッケージを入手し、 rpm コマンドを用いてインストールと設定を行います。

※ RHELに付属するツール群からSambaのインストールを行います。

以降の手順は、RHEL 7に付属するSamba4にて構築した場合を記述しています。運用環境に合わせた詳細設定をする際には、「システム管理者のガイド - RED HAT ENTERPRISE LINUX 7 の導入、設定、管理 」 第14章 ファイルとプリントサーバを参照してください。

B.3.1　Samba4インストール

```
# rpm -ivh samba-4.v.v-x.ely.zzzzz.rpm
----------------------------------------
4.v.v : Sambaバージョン番号
x     : リリース
```

```
y     : osメジャーバージョン
zzzzz : 対応アーキテクチャ(x86_64)
----------------------------------------
```

B.3.2 Sambaの設定

Windows端末から共有ディレクトリとして参照できるようにSambaの設定ファイル/etc/samba/smb.confを編集します。

はじめに、編集前のSamba設定ファイルの名前を変更して保存し、バックアップファイルとします。

■smb.confのバックアップ

```
# cp -p /etc/samba/smb.conf /etc/samba/smb.conf_org
```

/etc/samba/smb.confを編集し、Windows端末からScaTeFSを利用するためのSambaの設定を記載します。以下に、設定を推奨する項目と推奨値を記述します。下表に示した設定項目以外については、既定値が設定されます。

■Global Settings

項番	設定項目名	既定値	推奨値	備考
1	csc policy	manual	Disable	ファイル、ディレクトリについてオフラインで使用するために設定する項目。manual、documents、 programsを設定した場合、自動または、手動による同期処理が可能となるが、同期処理中に失敗した場合、共有サーバ上のファイルが消える可能性があるため、disableを設定する
2	netbios name	マシンのDNS名	サーバ名	ネットワークコンピュータ名を設定する。設定することによりネットワークコンピュータ名一覧に当該項目で設定したサーバ名が表示される

■Share Definitions

項番	設定項目名	既定値	推奨値	備考
[ScaTeFS_share]				任意の共有名を設定する。（ScaTeFS_shareは一例）
3	Comment	なし	任意のコメント	ファイル共有で表示されるコメントを設定する。当該項目を設定することで、「ディレクトリの説明」欄に設定したコメントが表示される
4	Path	なし	ScaTeFSのマウントポイント「 /mnt/scatefsなど」	ファイル共有するパスを設定する。実際に参照されるパスを設定する
5	Writable	なし	yes	ファイル共有への書き込みを許可することを設定する

B.3.3 ScaTeFSのマウント

「NEC Scalable Technology File System(ScaTeFS)運用の手引」マニュアルの「6.1.6または6.2.5 マウント方法」を参照します。

B.3.4 公開ディレクトリの作成

Windows端末と共有するディレクトリは、ScaTeFSのマウントディレクトリ配下に作成します。

以下のコマンドを上から順に実行します。

■共有ディレクトリ作成

```
# mkdir -p /mnt/scatefs/share
# chmod -R 0777 /mnt/scatefs/share
# chown -R root:root /mnt/scatefs/share
```

B.3.5 Sambaの起動

■Sambaサービスを起動します。

```
# systemctl start smb
# systemctl start nmb
```

B.3.6 Sambaユーザの作成

Sambaで利用するLinuxユーザを作成し、Sambaのユーザとして登録します。

■Linuxユーザの作成

```
# useradd samba_user
# passwd samba_user
```

■Sambaユーザの登録

```
# pdbedit -a samba_user
```

B.3.7 SELinuxの設定

確認コマンドの結果が「Enforcing」の場合、以降の設定を行います。「Permissive」、「Disabled」の場合は、SELinuxの設定は不要となります。

■SELinuxの有効・無効を以下のコマンドで確認します。

```
# /usr/sbin/getenforce
Enforcing
```

■SELinuxの設定変更

Sambaによる共有ディレクトリのアクセスを許可するため、「samba_share_t」のタイプを

設定します。

```
chcon -t samba_share_t [共有するディレクトリ]
```

B.3.8 ファイアウォールの設定

確認コマンドの結果が「running」の場合、以降の設定を行います。「not running」の場合は、ファイアウォールの設定は不要となります。

■ファイアウォールの起動を確認します。

```
# firewall-cmd --state
```

■ファイアウォールの設定

```
# firewall-cmd --permanent --zone=public --add-service=samba
# firewall-cmd --reload
```

B.4　Windows端末での接続設定

ScaTeFS 共有領域へアクセスする接続方法として、代表的な方法を以下に示します。

B.4.1 ScaTeFS共有領域にアクセスする

Windows端末から、エクスプローラーを使用してScaTeFS共有領域を開きます。

(1) Windows 端末にてエクスプローラーを起動し、アドレスバーに Samba サーバ名と共有名を入力する。

```
例：¥¥samba_server¥ScaTeFS_share
      smba_server: netbios nameで指定したサーバ名または、IPアドレス
      ScaTeFS_share: 共有名
```

(2) エクスプローラーのライブラリウィンドウに ScaTeFS 共有領域が表示されます。

B.4.2 ネットワークドライブの割り当て

Windows端末から、ScaTeFS共有領域をネットワークドライブとして使用する。

(1) ナビゲーションウィンドウより、「コンピュータ」を右クリックします。

(2) 表示されたコンテキストメニューより、「ネットワークドライブの割り当て(N)」を選択します。

(3) ネットワークドライブの割り当て画面にて任意の「ドライブ」を選択し、「フォルダ」には Windows 端末との共有を行うサーバの IP アドレスとディレクトリ名を入力して「完了」ボタンを押下する。

(4) ナビゲーションウィンドウに上記「1」～「3」で指定したドライブが表示されます。

B.5　クラスタ構成

本章では、Samba サーバをクラスタ構成にする場合の設定例を記載します。以降に記載した設定例は、CLUSTERPRO 公式サイトより取得したリファレンスガイド、マニュアル、Samba on CLUSTERPRO for Linux HowTo※ を参照し、設定しています。環境構築する際には、運用環境に合わせた設定となるように各種マニュアルを参照して設定を行ってください。

なお、CLUSTERPRO の設定は WebManager を使用するため、対象となる Linux クライアントへネットワーク接続できる作業用 Windows 端末が必要となります。

※ Samba on CLUSTERPRO for Linux HowTo
CLUSTERPRO公式サイト「CLUSTERPRO X for Linux ソフトウェア構築ガイド」より、
「カテゴリ」がファイルサーバ、「ソフトウェア名」がSambaに記載されている設定手順書。

また、CLUSTERPRO X for Linux のオプション製品である「CLUSTERPRO X File Server Agent for Linux」を利用することで WebManager から、Samba サーバの監視に特化した設定が可能となります。こちらの製品については、CLUSTERPRO 公式サイトを参照してください。

B.5.1　クラスタの作成

クラスタを作成します。

B.5.1.1　クラスタの追加

メニューバーの「ファイル」の「クラスタ生成ウィザード」をクリックして、確認画面で「標準版クラスタ生成ウィザードを開始する」をクリックします。クラスタ生成ウィザード画面のクラスタの定義はデフォルトのままで、[次へ] をクリックします。

B.5.1.2　サーバの追加

クラスタ生成ウィザード画面のサーバの定義では、クラスタを構成する2台の Linuxクライアントを追加します。以下の説明では Linuxクライアントをlxcl00, lxcl01と記載します。

サーバの定義一覧で 追加 をクリックします。

サーバ追加画面で以下の項目を設定します。マスタサーバとなります。

サーバ名

lxcl00

再度、サーバの定義一覧で 追加 をクリックします。

サーバ追加画面で以下の項目を設定します。

サーバ名

lxcl01

B.5.1.3　ネットワーク構成の設定

クラスタを構成するLinuxクライアント間のネットワーク構成を設定します。

インタコネクト一覧で 追加 をクリックします。

優先度1の行の項目を設定します。

種別

カーネルモード

MDC

使用しない

lxcl00

Linuxクライアント間インタコネクト用IPアドレス

lxcl01

Linuxクライアント間インタコネクト用IPアドレス

※ 本手順は、インタコネクト用IPアドレスとして、クライアントのネットワーク設定ポートのIPアドレスを設定しています。

B.5.1.4　ネットワークパーティション解決処理の設定（NP解決）

設定を行わないで次に進めてください。

B.5.2　フェイルオーバグループの作成

クラスタを構成するLinuxクライアントで動作するフェイルオーバグループを作成します。

B.5.2.1　フェイルオーバグループの追加

グループ画面のグループ一覧で 追加 をクリックします。

グループの定義画面で以下の項目を設定します。

名前

[failover1]

failover1

起動可能サーバ一覧画面で すべてのサーバでフェイルオーバ可能 のチェックを外します。

利用可能なサーバで 以下の順序でLinuxクライアントを選択して 追加 をクリックします。

[failover1]

lxcl00

lxcl01

グループ属性の設定画面で以下の項目をデフォルトから変更します。

フェイルバック属性

自動フェイルバック

B.5.2.2 グループリソース (フローティング IP アドレス) の追加

Linuxクライアントのネットワーク設定のポート(10GbE)のIPアドレスを設定します。

なお、Linuxクライアントのネットワーク設定のポート数で追加するリソース数が異なりますが、本手順書では10GbEを使用する場合で、10GbEのFIP(fip1)について、[failover1]へ fip1を追加する例を記載します。

グループリソース画面のグループリソース一覧で 追加 をクリックします。

グループリソースの定義画面で以下の項目を設定します。

タイプ

floating ip resource

名前

[failover1]

fip1

依存関係画面ではデフォルトのまま設定を行わないで次を進めてください。

復旧動作画面ではデフォルトのまま設定を行わないで次を進めてください。

詳細画面(共通タブ)で以下の項目を設定します。

IPアドレス

Linuxクライアントのネットワーク設定のポート(10GbE)のIPアドレス

設定例

192.168.0.31/24%enp4s9

調整ボタンをクリックします。

フローティングIPリソース調整プロパティのパラメータタブで以下の項目を設定します。

NIC Link Downを異常と判定する

チェックボックスをオンにします。

B.5.2.3 グループリソース（EXEC リソース）の追加

Linuxクライアントで実行するリソース(Samba)を追加します。

Sambaサーバ

グループリソース画面のグループリソース一覧で 追加 をクリックします。

グループリソースの定義画面で以下の項目を設定します。

タイプ

execute resource

名前

[failover1]

exec_samba

依存関係画面で以下の項目を設定します。

既定の依存関係に従う

チェックを外します

利用可能なリソースの各リソースを選択して 追加 をクリックします。

※表示されるすべてのリソースを追加します。

[failover1]

fip1, EXECリソース(exec_samba)

復旧動作画面ではデフォルトのまま設定を行わないで次を進めてください。

詳細画面で以下の項目を設定します。

ユーザアプリケーションを選択します。

スクリプト一覧で編集をクリックします。

アプリケーション・パスの入力で以下の項目を設定します。

開始

/root/samba_ctl/start.sh

終了

/root/samba_ctl/stop.sh

※ 上記「start.sh」、「stop.sh」は、「B.5 クラスタ構成」に記載している「Samba on CLUSTERPRO for Linux HowTo」を参考に運用環境に合わせて作成してください。また、保存場所についても運用環境に合わせた保存先を指定してください。

B.5.3 モニタリソースの作成

ここからは、フェイルオーバグループごとではなくLinuxクライアントごとにモニタリソースを作成します。

B.5.3.1 モニタリソース (フローティングIPモニタ) 設定変更

自動で作成されたフローティングIPモニタの設定を変更します。

モニタリソース画面のモニタリソース一覧でfipw1を選択して、プロパティをクリックします。

監視(固有)画面で以下の項目を設定します。

NIC Link Up/Downを監視する

チェックボックスをオンにします。

B.5.3.2 モニタリソース (カスタムモニタ) 追加

EXECリソース(Sambaサーバ)のモニタリソースを追加します。

モニタリソース画面のモニタリソース一覧で 追加 をクリックします。

モニタリソースの定義画面で以下の項目を設定します。

タイプ

custom monitor

名前

lxcl00

genw_samba

監視(共通)画面で以下の項目を設定します。

インターバル

30

監視タイミング

活性時を選択します。

対象リソース

lxcl00

参照をクリックして、exec_sambaを選択します。

監視(固有)画面で以下の項目を設定します。

ユーザアプリケーションを選択します。

ファイル

lxcl00

/root/samba_ctl/is_samba_running.sh

監視タイプ

同期を選択します。

回復動作画面で以下の項目を設定します。

回復動作

回復対象に対してフェイルオーバ実行

回復対象

lxcl00

参照をクリックして、exec_sambaを選択します。

※ 上記で記載している「is_samba_running.sh」では、監視対象となるデーモンを起動させるシェルスクリプトとなります。シェルスクリプトは、運用環境に合わせて作成してください。また、保存場所についても運用環境に合わせた保存先を指定してください。
以下に例を記載します。
・is_samba_running.shの例

```
#!/bin/bash


systemctl status smb | grep Active | grep -q "active (running)"
SMB_STATUS=$(echo $?)
```

```
systemctl status nmb | grep Active | grep -q "active (running)"
NMB_STATUS=$(echo $?)

if [ ${SMB_STATUS} -eq 0 -a ${NMB_STATUS} -eq 0 ]; then
       exit 0
else
       exit 1
fi
```

B.5.4　クラスタプロパティの変更

ツリービューからクラスタ(cluster)を選択して右クリックして プロパティ を選択します。

タイムアウトタブを選択して以下の項目を設定します。

内部通信タイムアウト

300

付録 C　発行履歴

C.1　発行履歴一覧表

2018 年	2 月	初版
2018 年	8 月	2 版
2018 年	12 月	3 版
2019 年	5 月	4 版
2019 年	10 月	5 版
2019 年	11 月	6 版
2020 年	1 月	7 版
2020 年	5 月	8 版
2020 年	7 月	9 版
2020 年	10 月	10 版
2020 年	12 月	11 版
2021 年	5 月	12 版
2021 年	10 月	13 版
2021 年	12 月	14 版
2022 年	3 月	15 版
2022 年	6 月	16 版
2023 年	1 月	17 版
2023 年	3 月	18 版
2023 年	9 月	19 版
2024 年	10 月	20 版

C.2　追加・変更点詳細

- 初版

  新規作成

- 2 版

  9.1 資源制限(QUOTA)にディレクトリクォータに関する記載を追加

  9.7 リバランスを追加

- 3 版

  3.1 IOサーバ構成にIOSv4に関する記載を追加

5.1.2 LVMの設計にIOSv4に関する記載を追加

6.5 syslogメッセージを更新

10.6.2 VE_LD_PRELOADに指定するライブラリに関する記載をglibc対応に変更

10.6.6 NEC Fortranのプログラムの性能チューニングに関する記載を追加

- ４版

5.1.2 LVMの設計にIOターゲットを使用する順序に関する記載を追加

5.1.7 SPSインストールとLVM設定に関する記載を追加

5.1.8 CLUSTERPROインストールとLVM設定に関する記載を追加

5.1.13 ライセンス登録に関する記述を更新

6.1.1 IBドライバのインストール方法を更新

6.1.4 ライセンス登録に関する記述を更新

10.1 IOサーバの起動と停止を追加

表 12-1 諸元表を更新

付録A.24 シャットダウン監視の監視方法の設定を追加

- ５版

5.1.12 ScaTeFS/Serverパッケージインストール手順を更新

6.1.1 RHEL/CentOS 7.6用のMellanox OFEDの記載を追加

9.1.1 scatefs_quotacheckコマンドの記載を更新

11.6.2 ScaTeFS VEダイレクトIBライブラリの使用方法の記載を更新

表 12-1 諸元表を更新

- ６版

IOサーバIOSv4+において下記をサポート

- RHEL7.6
- CLUSTERPRO X 4.1
- IB HCA HDR100 (ConnectX-6)
- xfsをデータ領域のIOターゲットとしてサポート

3.1 IOサーバ構成にIOSv4+に関する記載を追加

4.1 クライアントの構成にConnectX-6の記述を追加

5.1.1 IOターゲットの設計にIOSv4+に関する記載を追加

5.1.2 LVMの設計にIOSv4+に関する記載を追加

5.1.4、5.1.5、5.1.6 IOサーバ構築手順の簡易化に関する記載を更新

5.1.8 CLUSTERPRO X 4.1に関する記載を追加

5.1.10 IBドライバのインストールにRHEL7.6に関する記載を追加

5.3.1 ScaTeFS作成に設定項目data_fstypeに関する記載を追加

5.4.2 WebManager起動にバージョン4.1に関する記載を追加

10.2.2 ScaTeFSライセンスごとの無停止アップデート手順の記載を更新

10.7 ConnectX-6 HCAカード交換後のFirmware更新の記載を追加

付録A.14 ディスクリソースの設定値を更新

- ７版

  6.1.1 RHEL/CentOS 7.7およびMellanox OFED 4.7の記述を追加

  ８章 DockerコンテナからScaTeFSを利用する際の設定の記載を追加

- ８版

  IOサーバIOSv4++において下記をサポート

  - RHEL7.7
  - CLUSTERPRO X 4.2
  - SPS 7.3.1

  3.1 IOサーバ構成にIOSv4++に関する記載を追加

  5.1.10 IBドライバのインストールにRHEL7.7に関する記載を追加

- ９版

  “3.1.6 DDN社のSFA7990XE” を追加

  "5.1.12 ScaTeFS パッケージのインストール" を更新

  “5.5 IOサーバの構築（DDNストレージ編）” を追加"

  "6.1.1 IB ドライバのインストール" に RHEL/CentOS 8.1 向けMellanox OFEDドライバのバージョンを記載

  "6.4.4 二重マウント時の注意事項" を追加

  "9.11 ファイルシステムの監視" を追加

- 10版

  “5.1.12 ScaTeFS パッケージのインストール” の SX-Aurora TSUBASAインストレーションガイドの参照先を更新

  “6.1.1 IB ドライバのインストール” に RHEL/CentOS 7.8 向けMellanox OFEDドライバのバージョンを記載

- 11版

  “6.1.1 IB ドライバのインストール” に RHEL/CentOS 8.2 向けMellanox OFEDドライバ

のバージョンを記載

“表 9-2 リモートCLIのサブコマンド”にmkqdir、rmqdirのサブコマンドに関する記載を追加

“9.11 ファイルシステムの監視”を更新

“10.8.1 Linuxクライアント”に追加し、6.5章の記載内容を転記

“10.8.2 IOサーバ”を追加

“11.6.9 ScaTeFS InfiniBand 高速IOライブラリ使用時のメモリ使用量”を追加

- 12版

  “5.1.12 ScaTeFSパッケージのインストール” に sosパッケージの事前インストールに関する記述を追加

  “6.1.1 IB ドライバのインストール” に RHEL/CentOS 7.9 向けMellanox OFEDドライバのバージョンを記載

  “6.1.6 マウント方法”に RHEL 8の記述とCentOSの記述を追加

  “6.2.5 マウント方法”に RHEL 8の記述とCentOSの記述を追加

  “10.8.2 IOサーバ”のScaTeFS関連メッセージを更新

  “11.6.4 性能チューニング用環境変数”に 環境変数 SCATEFS_CQPOLLHOWの説明を追加

- 13版

  “6.1.1 IB ドライバのインストール” に RHEL/CentOS 8.3 向けMellanox OFEDドライバのバージョンを記載

  “10.4 ファイルシステムの整合性チェックと修復”を更新

- 14版

  “6.1.1 IB ドライバのインストール” に RHEL/CentOS 8.4 向けMellanox OFEDドライバのバージョンを記載

  “6.4.1 オープンしているファイルの削除について”を更新

- 15版

  “1.2.1 クライアント”を更新

  “1.2.2 ネットワーク”を更新

“6.1.1 IB ドライバのインストール” に RHEL/CentOS 8.4とRHEL 8.5向けMellanox OFEDドライバのバージョンを記載

“6.4.5 mlocateパッケージを使用する場合の注意事項”を追加

- 16版

  “Mellanox OFED“を”MLNX_OFED“に変更

  "5.1.10 IB ドライバのインストール"のMLNX_OFEDドライバのダウンロードサイトのURLを変更

  "6.1.1 IB ドライバのインストール"のMLNX_OFEDドライバのダウンロードサイトのURLを変更

  "6.1.1 IB ドライバのインストール"にRHEL/Rocky Linux 8.5向けのMLNX_OFEDドライバのバージョンを追加

  "10.7 ConnectX-6 HCAカード交換後のFirmware更新"のFirmwareのダウンロードサイトのURLを変更

  "10.7 ConnectX-6 HCAカード交換後のFirmware更新"のConnectX-6のFirmwareのバージョンを更新

- 17版

  関連説明書のURLを変更

  “表 5-1 標準モデル向けIOサーバv4++ 動作確認済みバージョン”のCLUSTERPROのバージョンを更新

  "6.1.1 IB ドライバのインストール"にRHEL/Rocky Linux 8.6向けのMLNX_OFEDドライバのバージョンを追加

  “9.13 サブディレクトリマウント”を追加

  “9.13.1 マウント方法”を追加

  “9.13.2 アンマウント方法”を追加

  “9.1 資源制限(QUOTA)”にディレクトリQUOTAの記述を追加

  “10.8.2 IOサーバ”のScaTeFS関連メッセージを更新

- 18版

  IOサーバIOSv4++においてCLUSTERPRO X 4.3.4-1をサポート

  “5.1.3 CLUSTERPROのクラスタ構成情報作成”を更新

“5.1.12.1 HPCソフトウェアライセンスをお使いの場合"の「SX-Aurora TSUBASA インストレーションガイド」の参照先を更新

“5.4 CLUSTERPROの設定”にCluster WebUIに関する記載を追加

5.4.1.1 のタイトルを”クラスタ構成情報ファイルを作業用PCへ転送”に修正

5.4.2 のタイトルを”Cluster WebUIまたはWebManager起動”に修正

5.4.3 のタイトルを”クラスタ構成情報ファイルのインポート”に修正

"6.1.1 IB ドライバのインストール"にRHEL/Rocky Linux 8.6向けのMLNX_OFEDドライバのバージョンを追加

付録 A のタイトルを”CLUSTERPROのクラスタ構成情報作成手順(オフラインバージョン)”に修正し、Cluster WebUI Offlineに関する記載を追加

付録A.2 のタイトルを”CLUSTERPROツールのインストール”に修正

付録A.3 のタイトルを”CLUSTERPROツールの起動”に修正

付録A.12 のタイトルを”グループリソース (フローティング IP リソース) の追加”に修正

- 19版

  "6.1.1 IB ドライバのインストール"にRHEL/Rocky Linux 8.8向けのMLNX_OFEDドライバのバージョンを追加

- 20版

  “5.5.16 カーネルパラメータの設定”を更新

  "6.1.1 IB ドライバのインストール"にRHEL/Rocky Linux 8.10向けのMLNX_OFEDドライバのバージョンを追加

SX-Aurora TSUBASAシステムソフトウェア

NEC Scalable Technology File System (ScaTeFS)

運用の手引

2024年 10月 20版

日本電気株式会社

東京都港区芝五丁目７番１号

TEL(03)3454-1111（大代表）