FUJITSU

FUJITSU Software
Technical Computing Suite V4.0L20

ジョブ運用ソフトウェア
概説書

J2UL-2533-01Z0(01)
2021年8月

まえがき

本書の目的

本書では、Technical Computing Suiteのジョブ運用ソフトウェアについて、機能概要や用語を解説します。

本書の読者

本書は、ジョブ運用ソフトウェアを利用するすべての利用者が対象です。

本書の構成

本書は、次の構成になっています。

第1章 概要

Technical Computing Suiteの概要を説明します。

第2章 ジョブ運用ソフトウェア

ジョブ運用ソフトウェアについて説明します。

第3章 関連ソフトウェア

ジョブ運用ソフトウェアの関連ソフトウェアについて説明します。

付録A マニュアル一覧

ジョブ運用ソフトウェアのマニュアルの一覧です。

付録B FXサーバ固有の管理構造

富士通製CPU A64FXを搭載した計算機(FXサーバ)のハードウェアの構造について概要を説明します。

本書の表記について

機種名の表現

本書では 富士通製CPU A64FXを搭載した計算機を「FXサーバ」、FUJITSU server PRIMERGYを「PRIMERGYサーバ」(または単に「PRIMERGY」)と表記します。
また、本書で説明する機能の一部には、対象機種によって仕様に差があります。このような機能の説明では、以下のように対象機種を略称で表記します。
[FX] : FXサーバを対象にした機能です。
[PG] : PRIMERGYサーバを対象にした機能です。

マニュアル内のアイコンについて

本書では、以下のアイコンを使用しています。

注意

特に注意が必要な事項を説明しています。必ずお読みください。

参照

詳細な情報が書かれている参照先を示しています。

参考

ジョブ運用ソフトウェアに関連した参考記事を説明しています。

輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

商標

- Linux®は米国及びその他の国におけるLinus Torvaldsの登録商標です。
- そのほか、本マニュアルに記載されている会社名および製品名は、それぞれ各社の商標または登録商標です。

出版年月および版数

版数	マニュアルコード
2021年8月 第1.1版	J2UL-2533-01Z0(01)
2020年2月 初版	J2UL-2533-01Z0(00)

著作権表示

変更履歴

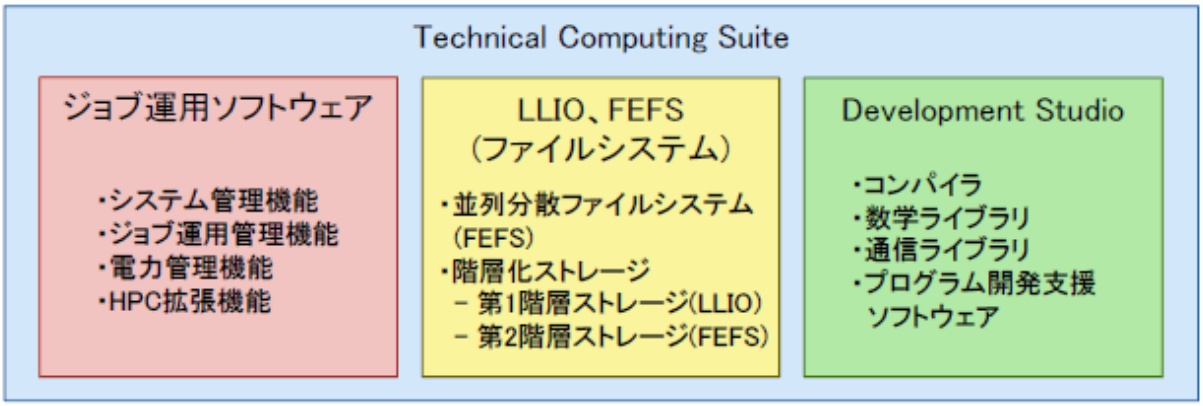
変更内容	変更箇所	版数
McKernelに関する情報の参照先URLを変更しました。	2.1.2	第1.1版

目　次

第1章 概要

Technical Computing Suiteは、スーパーコンピュータをはじめとする大規模な計算機システムの運用機能とアプリケーションの利用環境を提供するHPCミドルウェア製品です。
Technical Computing Suiteは以下のソフトウェアで構成されています。

図1.1 Technical Computing Suite のソフトウェア構成



ジョブ運用ソフトウェア

大規模な計算機システムの管理やアプリケーションの実行の管理と制御をする基盤ソフトウェア群です。これらをまとめて「ジョブ運用ソフトウェア」と呼びます。ジョブ運用ソフトウェアには以下の機能があります。

- システム管理機能
  システム内の計算機(ノード群)を階層化した管理形態や一元的な操作ビューを提供します。
- ジョブ運用管理機能
  アプリケーションをジョブと呼ぶ単位で実行するための管理や制御をします。
- 電力管理機能
  システムの消費電力の制限や、不要な消費電力を削減した省電力な運用を可能にします。
- HPC拡張機能
  Technical Computing Suiteの各機能がFXサーバを利用するためのドライバやライブラリを提供します。

LLIO、FEFS

LLIOとFEFSは、それぞれ以下の2つのファイルシステムを提供します。

- LLIO

  LLIO(Lightweight Layered IO-Accelerator)は、高速なフラッシュメモリを使用した高性能なファイルシステムです。LLIOはアプリケーションを実行する計算ノードからアクセスできます。

- FEFS

  FEFS(Fujitsu Exabyte File System)は、オープンソースのファイルシステムであるLustreの技術に基づいた、高速並列分散処理を可能にするスケーラブルなネットワークファイルシステムです。FEFSは計算機システム内での共有ファイルシステムとして使用されます。

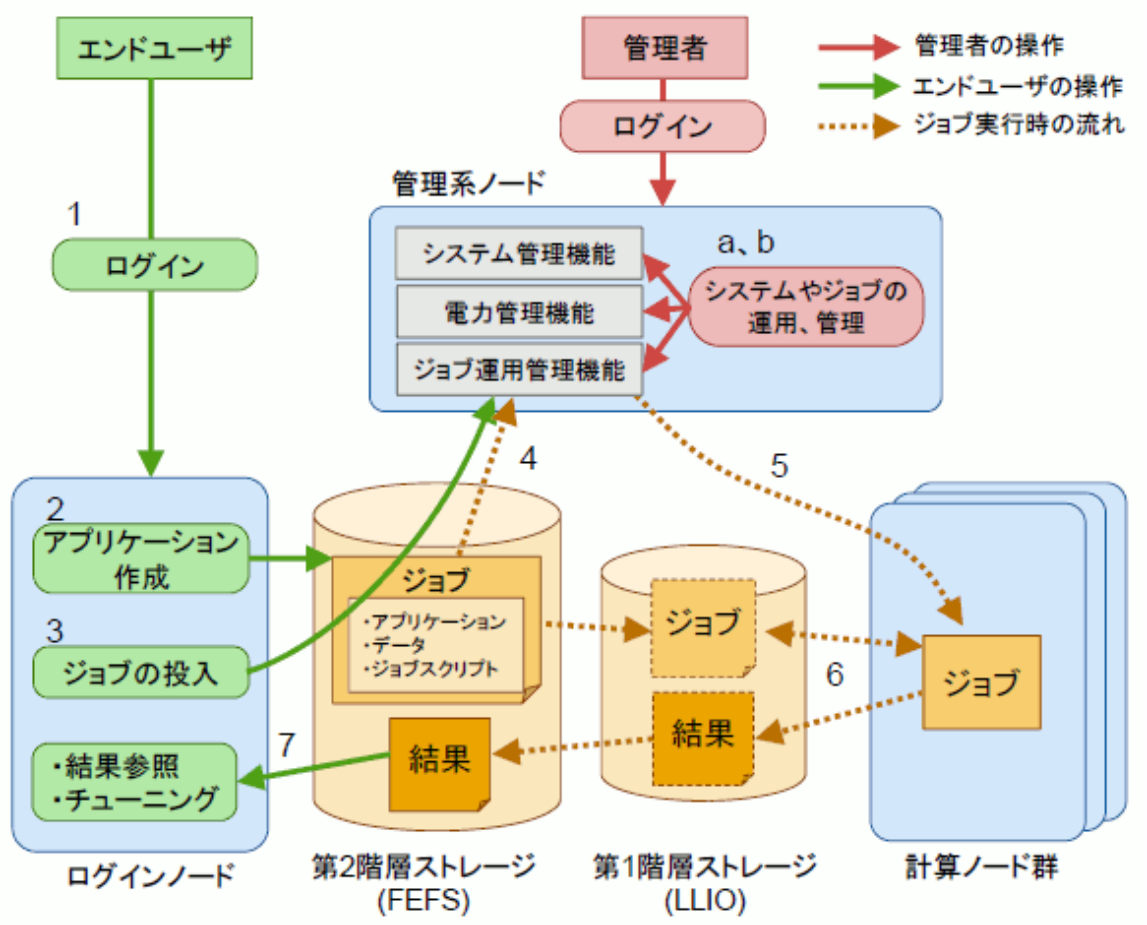
計算ノードから見たビューで、LLIOを上位層、FEFSを下位層として組み合わせることで、高速かつ大容量の階層化ストレージを実現します。階層化ストレージでは、LLIOを第1階層ストレージ、FEFSを第2階層ストレージと呼びます。

Development Studio

Fortran、C、C++ 言語で記述された科学技術計算向けプログラムの開発(コンパイル、デバッグ、チューニングなど)および実行を支援する統合的なソフトウェア群です。
自動並列化、OpenMP、およびMPI(Message Passing Interface)といった並列化技術をサポートしています。

Technical Computing Suiteの利用イメージを以下に示します。

図1.2 利用イメージ



[エンドユーザ]

エンドユーザは、アプリケーションを作成し、それをシステム上で実行します。

1. アプリケーションの作成や実行するために、エンドユーザはシステムを利用するための入り口のノード(ログインノード)にログインします。
2. Development Studioが提供するコンパイラやデバッガなどの開発環境でアプリケーションを作成します。
   作成したアプリケーション、その実行に必要なデータ、およびアプリケーションの実行手順を記述したシェルスクリプト(ジョブスクリプト)を、ジョブと呼ぶ実行単位として第2階層ストレージ(FEFS)に配置します。
3. エンドユーザは、ジョブ運用管理機能のコマンドを使ってジョブの実行を依頼します。これをジョブの投入と呼びます。エンドユーザが計算ノードへ直接ログインして、ジョブを実行することはできません。
4. ジョブ運用管理機能のコマンドで投入されたジョブの情報は、ジョブ運用管理機能へ送られ、バッチ処理されます。
5. ジョブ運用管理機能はジョブに割り当てる計算機資源(ノードや、ノード内のメモリ、CPU時間など)の量や実行の優先度などから、複数のジョブの実行順序をスケジューリングし、ジョブを実行します。
6. 第2階層ストレージ(FEFS)上のファイル(ジョブスクリプト、アプリケーション、およびアプリケーションの実行に必要なファイル)は、第1階層ストレージ(LLIO)を経由して計算ノードからアクセスされます。
   ジョブが出力した結果(ファイル)は第1階層ストレージを経由して第2階層ストレージに出力され、ログインノードで参照できます。
7. ジョブが終了すると、ジョブの標準出力と標準エラー出力の内容が出力されたファイルが、ログインノード上に作成されます。ジョブの実行中に異常が起きた場合は、メールで内容が通知されます。
   必要に応じて、エンドユーザはジョブの実行結果を参考にして、アプリケーションのチューニングをします。

参考

プログラム開発、ジョブの投入、およびジョブの実行結果の確認は、Development Studioが提供するプログラミング開発支援ソフトウェアを利用して、GUIでもできます。

[管理者]

管理者はシステムの管理をするためのノード(システム管理ノードや計算クラスタ管理ノード)にログインします。このノードでは、システムの運用に関する以下の作業が一元的にできます。

a. システムの運用、管理

- ノードの構築(ソフトウェアのインストール)
- ノードの起動・停止
- システムの稼働状況の監視
- ソフトウェアの保守(バックアップ・リストア、修正適用)
- トラブル時の調査資料採取

b. ジョブの運用、管理

- ジョブ運用の設定
- ジョブ運用の監視と操作
- ジョブの実行制御

第2章 ジョブ運用ソフトウェア

本章では、Technical Computing Suiteのジョブ運用ソフトウェアについて説明します。

科学技術計算の分野では、高い性能を達成するために、多数の計算機を使い、並列処理をする手法が利用されます。

このようなシステムでは、以下が求められます。

- 大勢の利用者からの要求に対し、膨大な数の計算機資源を効率よく割り当て、稼働率を上げること。
- 一部分の故障が起きても、システム全体としては運用を継続できること。
- 計算機の数の増加や複雑な構成になる場合でも、管理者にとっては容易にシステム全体を管理できること。
- システム全体の消費電力の状況を把握し、省電力化を図れること。
- 多数のエンドユーザがプログラムを実行するためにシステムを容易に利用できること。

このように大規模な計算機システムの管理やそのシステムでのプログラム実行管理・制御をする基盤ソフトウェアが「ジョブ運用ソフトウェア」です。

2.1 ジョブ運用ソフトウェアの機能

ジョブ運用ソフトウェアは、以下の機能で構成されています。

- システム管理機能
- ジョブ運用管理機能
- 電力管理機能
- HPC拡張機能

以降では、それぞれの機能について紹介します。

2.1.1 システム管理機能

システム管理機能は管理者向けの機能で、大規模システムでも効率的にシステム運用を行えるように、システム内の計算機(ノード群)を階層化した管理形態や一元的な操作ビューを提供します。

構成管理機能

構成管理機能は、システム内のノードやネットワークを管理するための機能です。システムを構成するノード群をクラスタやノードグループと呼ぶ単位でグルーピングします("2.2 ジョブ運用ソフトウェアの管理構造"参照)。また、ディスク装置やネットワークスイッチのような機器もシステムの構成要素として管理できます。

システム制御機能

システム制御機能は、ノードの電源制御(起動、停止)をするための機能です。ノードの電源制御は、ノード単位だけではなく、システム全体、クラスタごとなどのグループ単位での一括操作もできます。また、起動や停止の順序を考慮した制御もできます。

システム監視機能

システム監視機能は、ハードウェアやソフトウェアの稼働状況を監視するための機能です。異常を検出すると管理者へ通知したり、異常ノードを自動的にジョブ運用から切り離したりできます。ソフトウェアの稼働状況の監視では、ジョブ運用ソフトウェアのサービス以外に、管理者が指定したサービスの管理もできます。

システム保守機能

システム保守機能は、ハードウェアやソフトウェアの保守をするための支援機能です。この機能を利用することで、保守対象のノードを運用から切り離し、保守をしている間はジョブを割り当てないようにできます。また、ノードを冗長化している場合は、運用系と待機系を切り替えるフェイルオーバによって、運用に影響を与えずに保守作業ができます。システム監視機能と連携して、自動的にノードのフェイルオーバができます。

運用支援機能

運用支援機能は、多数のノードに対する操作や管理を支援するための機能です。この機能を利用することで、複数のノードに対するコマンドの実行やファイルの配送、収集を一括して操作したり、システム内の各ノードのコンソールへ1つのノードから接続したりできます。また、トラブルの調査で必要になる各ノードのダンプファイルを管理できます。

ログ管理機能

ログ管理機能は、トラブルの調査で必要になるログなどの資料を一括して収集したり、ログの内容を監視したりするための機能です。また、運用では各ノードで設定ファイルを作成する場合がありますが、この作成を容易にする支援機能も提供します。

ソフトウェア環境チェック機能

ソフトウェア環境チェック機能は、複数のノードに対するジョブ運用ソフトウェアの設定やソフトウェアパッケージの適用状況を確認するための機能です。この機能を利用することで、ノードの新規追加や保守時のソフトウェアパッケージの適用時に、複数のノードに対して期待した設定やソフトウェアパッケージが適用されているかどうかを確認できます。

インストール機能

インストール機能は、システムを構成するノードに対してOSを効率的にインストールするための機能です。この機能は、OSやパッケージをリポジトリで管理します。また、インストール機能はPRIMERGYサーバのシステム統合管理ツールServerView Suiteと連携しています。管理者はServerView Suiteを意識しなくても、PRIMERGYサーバへのOSのインストールができます。

バックアップ・リストア機能

バックアップ・リストア機能は、ノードのディスク装置の内容をディスクイメージとしてバックアップしたり、リストアしたりするための機能です。1つのノードのディスクイメージをほかのノードに複製して構築できます。また、ハードディスク故障時または修正パッケージの適用時などのトラブル発生時に、バックアップしてあるディスクイメージをリストアしてノードを以前の状態に復元できます。

参照

システム管理機能の詳細は、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド システム管理編」を参照してください。また、導入や保守についてはそれぞれマニュアル「ジョブ運用ソフトウェア 導入ガイド」、「ジョブ運用ソフトウェア 管理者向けガイド 保守編」を参照してください。

2.1.2 ジョブ運用管理機能

ジョブ運用管理機能は、アプリケーションの実行で、計算機資源を効率的に利用し、システムの最大性能を引き出します。

ジョブマネージャー機能

ジョブ運用ソフトウェアでは、アプリケーションを「ジョブ」という単位で管理し、実行を制御します。エンドユーザはジョブを計算ノードで直接実行するのではなく、ジョブ運用ソフトウェアに対して実行を依頼します。ジョブマネージャー機能は、ジョブの受付、状態管理、および実行の制御をします。

ジョブスケジューラー機能

ジョブマネージャー機能が受け付けたジョブは、すぐに実行されるのではなく、一旦キューイングされます。その後、ジョブスケジューラー機能によって実行順序がスケジューリングされます。ジョブスケジューラー機能は、限られた計算機資源に対して、複数のジョブを効率よく実行させるために、ジョブの優先度や計算機資源(ノードおよびノード内のメモリやCPU)の空き状況などから、ジョブの実行順序を決定します。

ジョブ資源管理機能

ジョブ資源管理機能は、ジョブが計算機資源(メモリ、CPU)を一定期間専有できるように確保します。これにより、ジョブとジョブ以外のプロセス(OSのデーモンなど)が使用する計算機資源の競合をなくし、計算機の最大性能を引き出します。

並列実行環境

並列実行環境は、複数のノードを使う並列プログラムのプロセス制御をする仕組みです。

ジョブ実行環境

ジョブ運用管理機能は、ジョブの実行環境を切り替えることができます。ホストLinux環境でのジョブ実行に加え、Dockerコンテナ上でのジョブ実行をサポートします。Dockerコンテナ上のジョブ実行環境では、ジョブの実行をシステムのソフトウェア環境に依存せず、ジョブに応じて適切なソフトウェア環境(特定のOS版数など)でのジョブ実行が可能です。
また、ジョブ運用管理機能は、HPCアプリケーションの性能向上を目的とした軽量OSのMcKernel(※)を利用したジョブ実行が可能で

す。
※ https://ihkmckernel.readthedocs.io

ジョブ運用管理機能をカスタマイズするためのAPI

ジョブ運用管理機能は、管理者やエンドユーザが独自のコマンドインターフェースを持ったジョブ運用管理機能のコマンドを作成できるようにするコマンドAPIを提供します。また、管理者がジョブ運用のポリシーに合わせてジョブの実行を制御したり、情報を取得したりするインターフェース(フック機能など)を提供します。

参照

ジョブの実行方法やジョブ運用管理機能の使い方については、それぞれマニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド」、「ジョブ運用ソフトウェア 管理者向けガイド ジョブ管理編」を参照してください。各APIについては、それぞれAPIユーザーズガイドを用意していますので、それらを参照してください。マニュアルの一覧については"付録A マニュアル一覧"を参照してください。

2.1.3 電力管理機能

システムの消費電力の制限や、不要な消費電力を削減した省電力な運用を可能にします。

システム電力収集・可視化支援機能

システム内の計算ノードや計算ノード以外のシステム運用に必要な機器(外部機器)の消費電力の情報を収集します。また、管理者にこれらの消費電力の情報を表示するためのコマンドと、アプリケーションから情報を取得するためのインターフェース(システム電力可視化支援API)を提供します。

節電機能

ジョブ運用管理機能と連携して、ジョブ実行予定が長期間ないノードを自動的に電源停止します。また、ジョブが短期間実行されないノードをハードウェアの低消費電力モードへ遷移します。ジョブの実行開始に合わせて電源停止または低消費電力モードから復帰します。このような制御により、システムの無駄な消費電力を軽減させます。

Power API

米国サンディア国立研究所が規定・推進するPower API(http://powerapi.sandia.gov/)を提供します。管理者やエンドユーザが作成したアプリケーションからCPUやメモリ単位で消費電力の計測および制御ができます。

キャッピング機能

システムの消費電力が上限を超過しないようにジョブをスケジューリングします。また、FXサーバではシステムの消費電力が急激に増加することを抑制するために、BoB(Bunch of Blades、"B.1 FXサーバのハードウェアの構成要素"参照)を単位として計算ノードのCPU周波数と電源の停止を制御します。

参照

電力管理機能の詳細については、マニュアル「ジョブ運用ソフトウェア 管理者向けガイド 電力管理編」を参照してください。また、Power API機能については、マニュアル「ジョブ運用ソフトウェア APIユーザーズガイド Power API編」も参照してください。

2.1.4 HPC拡張機能

HPC拡張機能は、標準のLinuxに含まれる機能をFXサーバ用に拡張した各種ドライバ/ライブラリを提供します。

TofuDドライバ

FXサーバにおけるインターコネクト接続をするためのハードウェアのTofuインターコネクトD(以降、Tofuインターコネクト)を利用できるようにするためのドライバを提供します。
ジョブはTofuDドライバを意識する必要がなく、ジョブ運用管理機能を通じて、Tofuインターコネクトを利用できます。

HPCタグアドレスオーバーライド制御機能

FXサーバに搭載されたプロセッサ固有のHPCタグアドレスオーバーライド機能を制御する機能です。
この機能は、関連ソフトウェアDevelopment Studioで作成されたアプリケーションの性能チューニングやプロファイラによるハードウェアイベントの取得をサポートします。詳細については、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド HPC拡張機能編」をお読みください。

電力制御ドライバ/ライブラリ

FXサーバは、電力測定、および電力制御のためのAPIであるPower APIに準拠しています。このPower APIを利用できるようにした固有のドライバ、およびライブラリを提供します。
これらのドライバ/ライブラリは、ジョブ運用ソフトウェアの電力制御機能が使うほか、アプリケーションを作成する利用者向けにも公開します。
FXサーバにおけるPower APIの利用については、マニュアル「ジョブ運用ソフトウェア APIユーザーズガイド Power API編」をお読みください。

コア間ハードウェアバリアドライバ/ライブラリ

FXサーバでコア間ハードウェアバリア機能をサポートするためのドライバとライブラリを提供します。
コア間ハードバリア機能は、スレッド並列化されたアプリケーションプログラムのスレッド間の同期を高速に取るための機能です。この機能は、Development Studioで作成したアプリケーションで利用できます。詳細は、Development Studioのマニュアルをお読みください。

セクタキャッシュドライバ/ライブラリ

FXサーバでセクタキャッシュ機能をサポートするためのドライバとライブラリを提供します。
セクタキャッシュ機能は、再利用性の高いデータを極力キャッシュ内に保持し続けるようにすることでアプリケーションの動作速度を向上させる機能です。この機能は、Development Studioで作成したアプリケーションで利用できます。詳細は、Development Studioのマニュアルをお読みください。

ラージページライブラリ

FXサーバでは、Huge PageとしてLinuxの標準機能を利用します。LinuxのHuge Pageには「THP(Transparent Huge Page)」と「HugeTLBfs」の2つがありますが、FXサーバではメモリの使用効率、Huge Page化できるメモリ領域の対象範囲、Huge Page獲得の保証性などの観点から、HugeTLBfsを採用します。

HPC拡張機能で提供するラージページライブラリは、HugeTLBfsをFXサーバ用に、より効率よく、また拡張性を高めて利用できるように機能を拡張しています。さらに、HugePageのメモリ使用状況を集計するツールも提供します。

利用者は、Development Studioで本ライブラリをリンクしてHuge Pageを使うアプリケーションを作成できます。ジョブ実行時には、ジョブスクリプトの中で環境変数を使ってラージページライブラリの動作を変更することも可能です。詳細は、マニュアル「ジョブ運用ソフトウェア エンドユーザ向けガイド HPC拡張機能編」をお読みください。

参考

LinuxのHuge Pageは、ジョブ運用ソフトウェアでは、通常ページ(Normal Page)に対してより大きなサイズという意味で「ラージページ」と呼称します。参照先のマニュアルでは、特に断りなく「ラージページ」という言葉を使用します。

また、HPC拡張機能はジョブ運用ソフトウェアの中で以下の役割を担い、管理者に意識させることなく、大規模計算システムの効率的な運用を可能にしています。

高速再起動

FXサーバの再起動時間を短縮させることで、システムの稼働率を向上させます。

ダンプ世代管理

FXサーバのような大規模計算システムの保守用資料(メモリダンプ)の数を適切に制御することで、計算ノードの資源(ディスク容量)を効率的に使えるようにします。

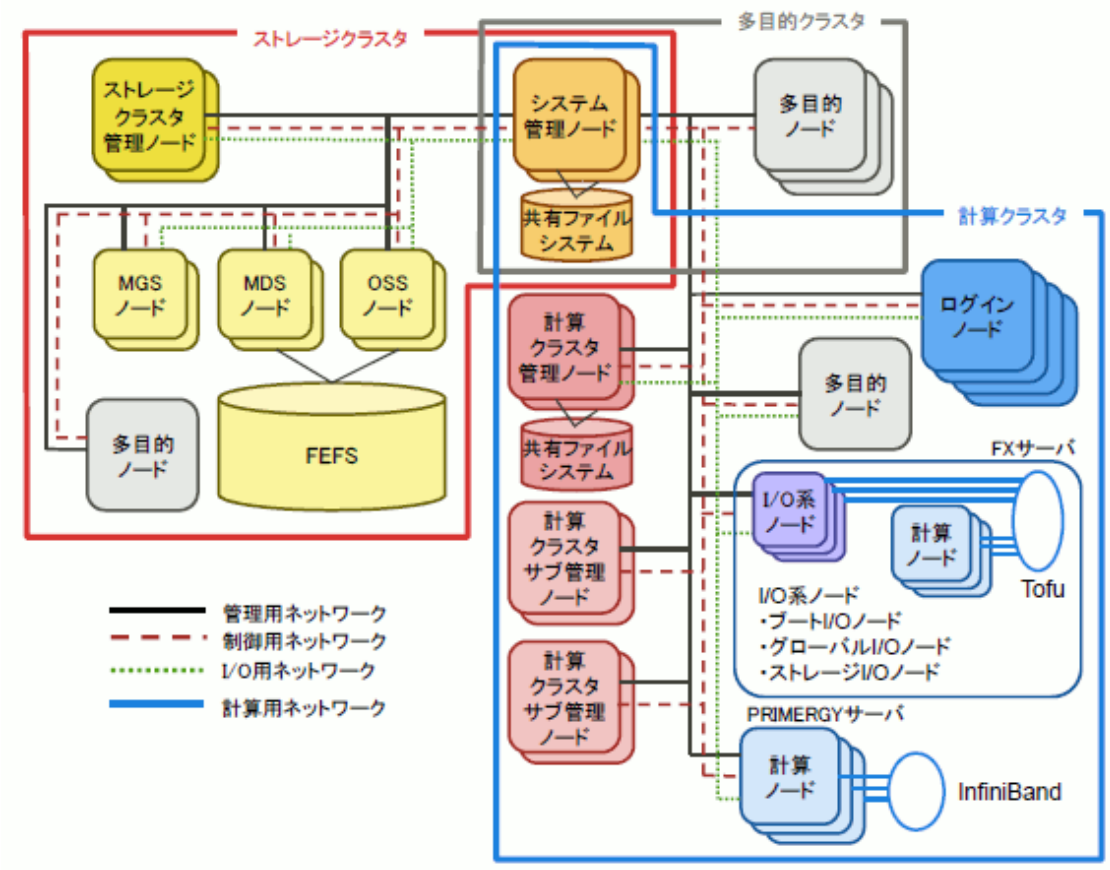
2.2 ジョブ運用ソフトウェアの管理構造

ここでは、ジョブ運用ソフトウェアがシステムを管理する構造について説明します。特に管理者は、システムの制御、管理およびジョブ運用をするために理解しておく必要があります。

2.2.1 システム構成

ジョブ運用ソフトウェアを導入するシステムは、以下のような構成になります。

図2.1 システム構成のイメージ



以降では、各構成要素について説明します。

2.2.2 ノード

ジョブ運用ソフトウェアを導入したシステムでは、ノードにはその役割に応じて以下に示す「ノード種別」が決められます。

表2.1 ノード種別

ノード種別	略称	役割
ログインノード	LN	エンドユーザがアプリケーションの作成やジョブ運用ソフトウェアにジョブの実行を依頼するためのノードです。 (LN: Login node)
システム管理ノード	SMM	クラスタの起動・停止のための電源制御や、クラスタ内のノード・サービス監視をするノードです。 (SMM: System Management node) 通常、依存関係がある計算クラスタ、ストレージクラスタ、および多目的クラスタは同じシステム管理ノードを共有します。 ジョブ運用ソフトウェアは、システム管理ノードの冗長構成をサポートします。冗長構成にする場合、運用系と待機系のシステム管理ノードの間でファイルを引き継ぐために、共有ファイルシステムが必要です。
計算クラスタ管理ノード	CCM	計算クラスタ内でのジョブ運用に関する情報を管理します。 (CCM: Compute Cluster Management node) エンドユーザがログインノードから投入したジョブは、計算クラスタ管理ノードで受付けられ、実行がスケジューリングされます。 ジョブ運用ソフトウェアは、計算クラスタ管理ノードの冗長構成をサポートします。冗長構成にする場合、運用系と待機系の計算クラスタ管理ノードの間でジョブ運用に関する情報を引き継ぐために、共有ファイルシステムが必要です。

ノード種別	略称	役割
計算クラスタサブ管理ノード	CCS	計算クラスタ管理ノードによるサービス監視の負荷を軽減するためのノードです。 (CCS: Compute Cluster Sub Management node) 計算クラスタ管理ノードが直接サービス監視をする代わりに、計算クラスタサブ管理ノードが監視します。 ジョブ運用ソフトウェアは、計算クラスタサブ管理ノードの冗長構成をサポートします。
ブートI/Oノード [FX]	BIO	FXサーバで、ノードのブートサーバになるI/Oノードです。 (BIO: Boot I/O node) FXサーバでは、一部の計算ノードがブートI/Oノードの役割を兼ね、計算ノード兼ブートI/Oノード(CN/BIO)と表記します。
グローバルI/Oノード [FX]	GIO	FXサーバで、第2階層ストレージ(FEFS)に対する入出力を中継するノードです。 (GIO: Global I/O node) FXサーバでは、一部の計算ノードがグローバルI/Oノードの役割を兼ね、計算ノード兼グローバルI/Oノード(CN/GIO)と表記します。 グローバルI/Oノードが故障した場合は、ラック内のほかのグローバルI/Oノードによる縮退運用になります。
ストレージI/Oノード [FX]	SIO	FXサーバで、第1階層ストレージに対する入出力を担う I/Oノードです。 (SIO: Storage I/O node) ストレージI/Oノードには、第1階層ストレージを構成するディスク装置(SSD)が接続されています。 FXサーバでは、一部の計算ノードがストレージI/Oノードの役割を兼ね、計算ノード兼ストレージI/Oノード(CN/SIO)と表記します。
計算ノード	CN	ジョブが動作するノードです。 (CN: Compute node)
ストレージクラスタ管理ノード	SCM	ストレージクラスタ内の構成管理やサービスを監視します。 (SCM: Storage Cluster Management node) ジョブ運用ソフトウェアは、ストレージクラスタ管理ノードの冗長構成をサポートします。また、ストレージクラスタ管理ノードはシステム管理ノードと兼用でき、システム管理ノード兼ストレージクラスタ管理ノード(SMM/SCM)と表記します。
MGSノード	MGS	ファイルシステムの構成情報を管理するためのノードです。 (MGS: Management Server) ジョブ運用ソフトウェアは、MGSノードの冗長構成をサポートします。
MDSノード	MDS	ストレージクラスタが提供するFEFSのメタデータを格納・管理するノードです。 (MDS: Meta Data Server) ジョブ運用ソフトウェアは、MDSノードの冗長構成をサポートします。
OSSノード	OSS	FEFSにおける、ファイルデータを格納・管理するノードです。 (OSS: Object Storage Server) ジョブ運用ソフトウェアは、OSSノードの冗長構成をサポートします。
多目的ノード	任意	上記以外の任意の用途に使えるノードです。ジョブ運用ソフトウェアによる状態監視や電源制御の対象になります。 略称は、多目的ノードを構築する際に管理者が定義できます。

システム管理ノードまたは計算クラスタ管理ノードを冗長構成にする場合、ログやダンプファイルの情報をそれぞれの運用系ノードだけがマウントして使用する共有ファイルシステムが必要です。

参考

- ノード種別には兼用できる組み合わせがあります。詳細は、「ジョブ運用ソフトウェア 導入ガイド」の"ノード種別の兼用について"を参照してください。
  1つのノードが複数のノード種別を兼ねる場合、マニュアルでは"CN/BIO"のようにノード種別を併記する場合があります。
- FXサーバ(ブートI/Oノード、グローバルI/Oノード、ストレージI/Oノード、および計算ノード)には、ハードウェア固有の構造があります。詳細は"付録B FXサーバ固有の管理構造"を参照してください。

2.2.3 クラスタ

クラスタとは、システムを運用機能の観点で分割する単位で、計算クラスタ、ストレージクラスタ、および多目的クラスタがあります。

2.2.3.1 計算クラスタ

計算クラスタは、アプリケーションを作成したり、作成したアプリケーションをジョブと呼ぶ単位で実行したりするノード群です。

計算クラスタ内のノードを管理する単位には以下があります。

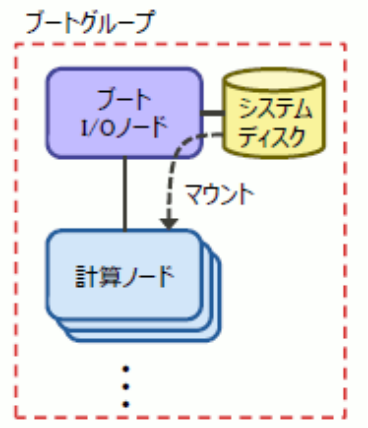
- ブートグループ [FX]
- ノードグループ
- SIOグループ [FX]
- GIOグループ [FX]
- リソースユニット
- リソースグループ

以降ではそれぞれについて説明します。

ブートグループ [FX]

ブートグループは、FXサーバのノードの起動単位で、BoB(Bunch of Blades、"B.1 FXサーバのハードウェアの構成要素"参照)に相当します。ブートグループ内のノードは同じブートI/Oノードをブートサーバとして使い、そのシステムディスクをマウントします。

図2.2 ブートグループ

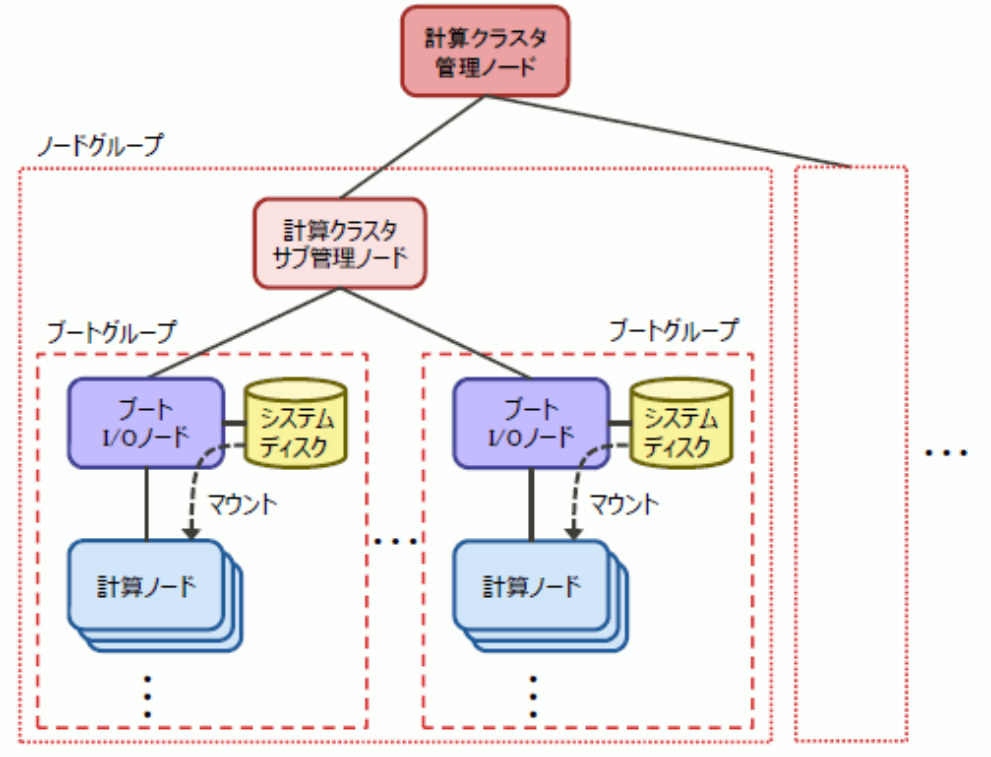


ノードグループ

計算ノード数が多い大規模システムでは、計算クラスタ管理ノードの配下に計算クラスタサブ管理ノードを設置します。計算ノードの監視を計算クラスタサブ管理ノードに分散させることで、計算クラスタ管理ノードの負荷を低減します。
1台の計算クラスタサブ管理ノードによって監視されるノード群のことをノードグループと呼びます。

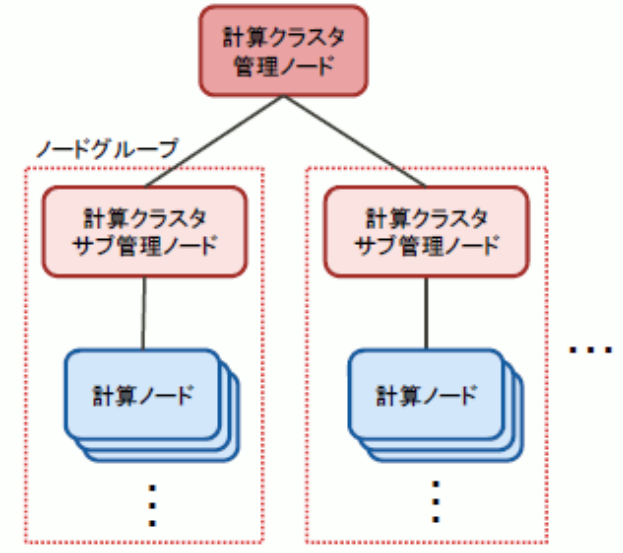
FXサーバの場合、ノードグループはブートグループを単位として構成されます。

図2.3 ノードグループの構造 (計算ノードがFXサーバの場合)



PRIMERGYサーバの場合は、ノードグループは計算クラスタサブ管理ノードと計算ノードで構成されます。

図2.4 ノードグループの構造 (計算ノードがPRIMERGYサーバの場合)



参考

計算クラスタサブ管理ノードを設置する目安は、以下のとおりです。

- FXサーバの場合
  クラスタ内のブートグループが252個を超える場合は計算クラスタサブ管理ノードが必要です。また、1ノードグループあたり、ブートグループの数が252個(4032ノード相当)を超えないようにします。

- PRIMERGYサーバの場合
  クラスタ内のPRIMERGYサーバ数が1024台を超える場合は、計算クラスタサブ管理ノードが必要です。また、1ノードグループあたり、1024台を超えないようにします。

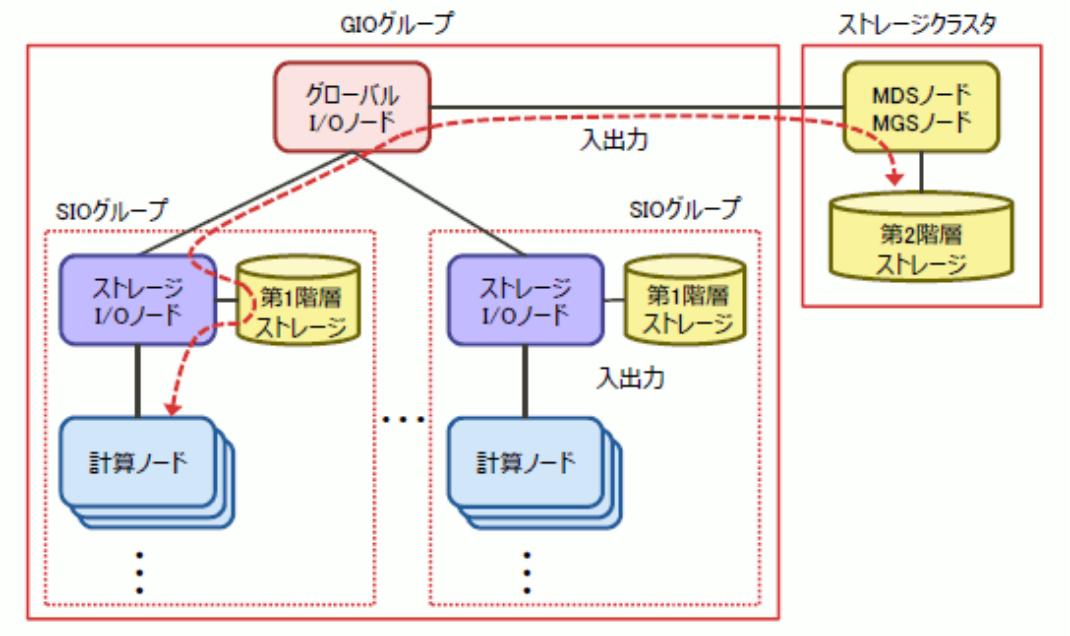
具体的な見積り方法については、マニュアル「ジョブ運用ソフトウェア 導入ガイド」の"クラスタ構成の見積り基準"を参照してください。

SIOグループとGIOグループ [FX]

FXサーバで、1つのストレージI/Oノードと、それを中継ノードとして第1階層ストレージに対して入出力をする計算ノード群をSIOグループと呼びます。

FXサーバでは、計算ノードから第2階層ストレージ(FEFS)に対する入出力は、第1階層ストレージとグローバルI/Oノードを経由します。1つの本体装置ラック内のグローバルI/Oノードと、それを使って入出力をするストレージI/Oノードおよび計算ノード群をGIOグループと呼びます。本体装置ラックについては、"B.1 FXサーバのハードウェアの構成要素"を参照してください。

図2.5 SIOグループとGIOグループの構造

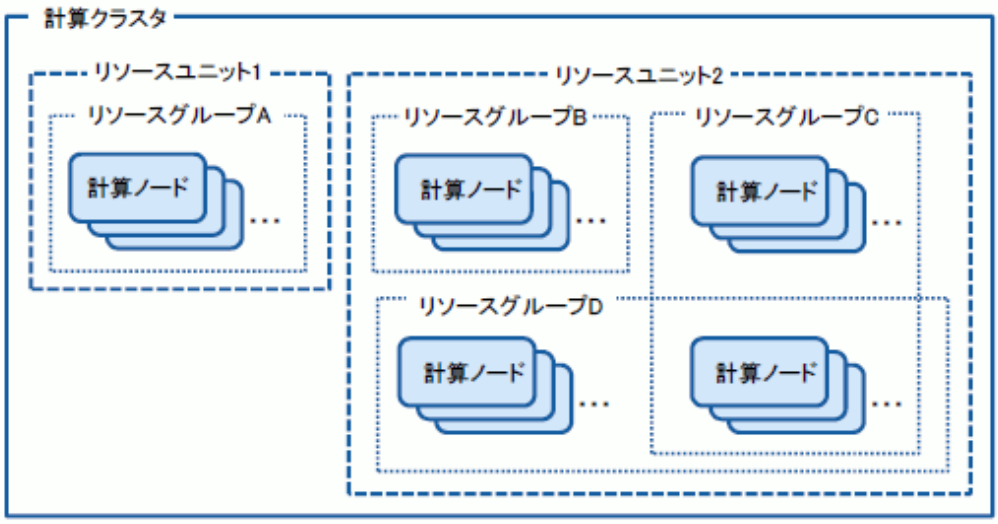


リソースユニットとリソースグループ

リソースユニットは、ジョブ運用の単位です。例えば、ジョブ運用管理機能の運用方針が異なる業務部門ごとにリソースユニットを分けて利用できます。計算クラスタの中であれば、管理者は任意の計算ノードの範囲をリソースユニットとして定義できます。ただし、1つのリソースユニットは同じ機種の計算ノードで構成する必要があります。

リソースグループは、ジョブが利用できる資源の単位です。例えば、ジョブが利用できる計算ノードの最大数やジョブを実行できる最長時間が異なるリソースグループを複数用意し、ジョブの規模や種類に応じて使い分けることで効率的なジョブ運用ができます。リソースユニット内のノードであれば、管理者は任意の計算ノードの範囲をリソースグループとして定義できます。1つの計算ノードが複数のリソースグループに属する構成もできます。

図2.6 リソースユニットとリソースグループ

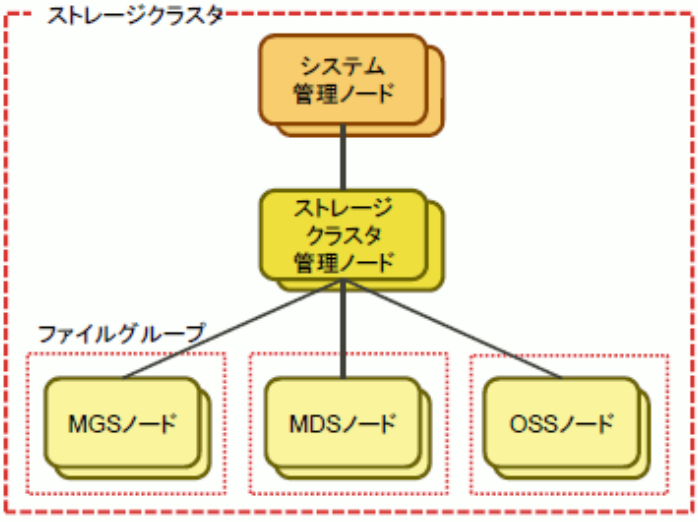


2.2.3.2 ストレージクラスタ

ストレージクラスタは、計算クラスタに対して共有ファイルシステムを提供するノード群です。共有ファイルシステムには、関連ソフトウェアのFEFSが利用できます。階層化ストレージでは、第2階層ストレージとして利用します。

ストレージクラスタでは、MGSノード、MDSノード、またはOSSノードの冗長構成のペアになるノードをファイルグループと呼びます。

図2.7 ストレージクラスタの管理構造



2.2.3.3 多目的クラスタ

多目的クラスタは、計算クラスタやストレージクラスタとは独立して電源制御や状態監視を行いたいノード群です。用途は任意で、例えば、LDAPサーバやNFSサーバを担うノード群を扱います。

2.2.3.4 クラスタとノード種別

各クラスタを構成できるノードは以下の表で"○"がついているノードです。

表2.2 クラスタを構成できるノード

クラスタを構成できるノード	計算クラスタ	ストレージクラスタ	多目的クラスタ
システム管理ノード	○(*)	○(*)	○(*)

クラスタを構成できるノード	計算クラスタ	ストレージクラスタ	多目的クラスタ
計算クラスタ管理ノード	○		
計算クラスタサブ管理ノード	○		
ブートI/Oノード [FX]	○		
グローバルI/Oノード [FX]	○		
ストレージI/Oノード [FX]	○		
計算ノード	○		
ログインノード	○		
多目的ノード	○	○	○
ストレージクラスタ管理ノード		○	
MGSノード		○	
MDSノード		○	
OSSノード		○	

(*) システム管理ノードは、各クラスタで共有できます。

2.2.4 ネットワーク

システム内のネットワークは、ノードと同様に用途別に分類されます。ネットワークを用途別に分けることで、状態監視などのシステム運用の処理が、ジョブの実行性能に対し外乱となることを避けられます。

ジョブ運用ソフトウェアのネットワークには以下の4種類があります。

表2.3 ネットワークの種類

名称	説明
制御用ネットワーク	ノードのハードウェア制御(電源制御や異常通知など)のために使うネットワークです。システム管理ノードは、このネットワークを通じて、クラスタ内の各ノードの制御装置と通信ができる必要があります。
管理用ネットワーク	ジョブ運用ソフトウェアの運用に関わるサービスの制御や情報のやり取りのために使うネットワークです。各ノードのOSと通信ができる必要があります。
I/O用ネットワーク	ストレージクラスタが提供する共有ファイルシステムを利用するための高速なネットワークです。転送の遅延が小さいインターコネクトを使って構築されます。ストレージクラスタが提供するFEFSと、ファイル入出力をするノード間で通信できる必要があります。
計算用ネットワーク	MPIプログラムのような並列プログラムがノード間通信のために使用する高速なネットワークです。このネットワークを通じて、計算ノード間で通信できる必要があります。PRIMERGYサーバでは管理用ネットワークを使用し、FXサーバではTofuネットワークを使用します。

2.2.5 構造の識別子

ジョブ運用ソフトウェアは、ここまでに述べた構造に名前や数値の識別子を付けて管理しています。これらの識別子は、管理者がシステムの電源操作やジョブ運用の管理に関する操作をする際に必要になります。各種操作で必要な識別子の値を知る方法については「ジョブ運用ソフトウェア 管理者向けガイド システム管理編」または「ジョブ運用ソフトウェア 管理者向けガイド ジョブ管理編」を参照してください。

表2.4 ジョブ運用ソフトウェアにおける構造の識別子

識別子	型	説明
クラスタ名	文字列	クラスタ(計算クラスタ、ストレージクラスタ、または多目的クラスタ)に付ける名前です。クラスタを設計する管理者が決めます。
ノードID	数値	ノードに自動で割り当てられる数値です。1つのクラスタ内で一意の値です。 ジョブ運用ソフトウェアでは、各ノードをホスト名ではなくノードIDで識別します。
ノードグループID	数値	ノードグループに自動で割り当てられる数値です。1つのクラスタ内で一意の値です。

識別子	型	説明
ブートグループID [FX]	数値	FXサーバのブートグループに自動で割り当てられる数値です。クラスタ内で一意の値です。
リソースユニット名	文字列	リソースユニットに付ける名前です。1つのシステム管理ノードが管理する範囲で一意でなければいけません。リソースユニットを設計する管理者が決めます。
リソースグループ名	文字列	リソースグループに付ける名前です。リソースユニット内で一意でなければいけません。リソースグループを設計する管理者が決めます。

2.3 管理者の分類

ジョブ運用ソフトウェアでは、管理者はOSのroot権限を持つユーザーです。

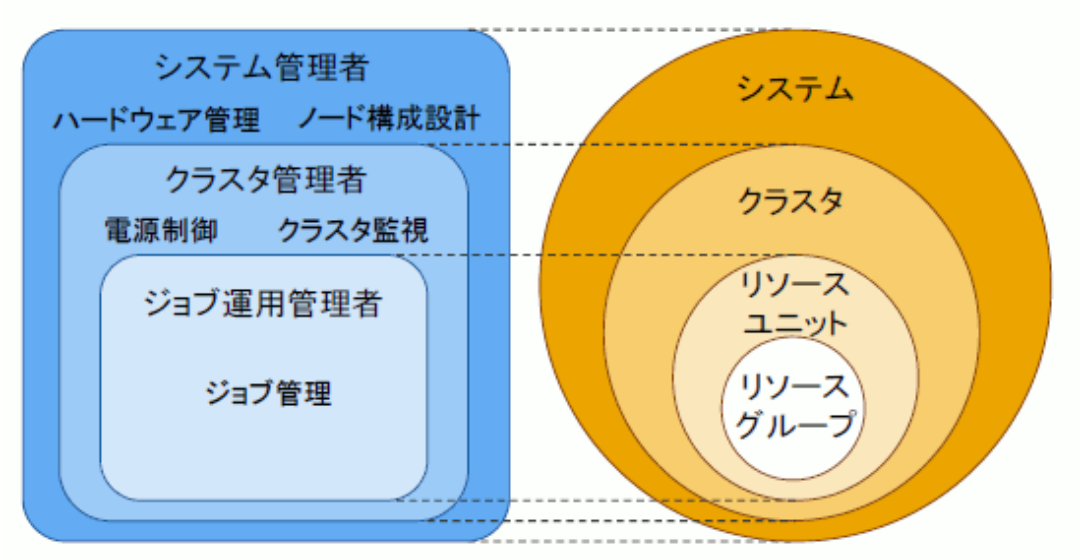
システムの規模が大きくなると、業務部門単位にシステムを分割して、利用する運用形態が考えられます。
このような場合、管理者の業務をシステムの構成やハードウェアの管理のようにシステム全体に関わる運用管理と各業務単位での運用管理のように役割で分けることもあります。

ジョブ運用ソフトウェアのマニュアルでは、このような運用上の役割の違いを意識して、管理者を以下のように呼び分けています。

表2.5 ジョブ運用ソフトウェアにおける管理者の分類

管理者の種類	責任範囲	役割
システム管理者	システム全体	ジョブ運用ソフトウェアの利用における最高権限の管理者です。ほかの管理者の作業も含めて、すべての機能を使用することが許されますが、主にシステム全体に関わる構成設計やハードウェアの管理を担当します。
クラスタ管理者	クラスタ内	ジョブ運用ソフトウェアにおける一番大きな運用単位のクラスタの管理者です。例えば、部門ごとに別々のクラスタを運用するような場合に、それぞれクラスタ管理者を設定します。クラスタ管理者は、クラスタにおける運用責任者であり、クラスタやノードの起動・停止や、ノード・サービスの状態監視の設定などします。また、クラスタ全体に関するジョブ運用の設定もします。
ジョブ運用管理者	リソースユニット内	ジョブ運用の単位である、リソースユニットの管理者です。ジョブ運用の方針やジョブが必要とする計算機資源を管理します。

図2.8 管理者の責任範囲



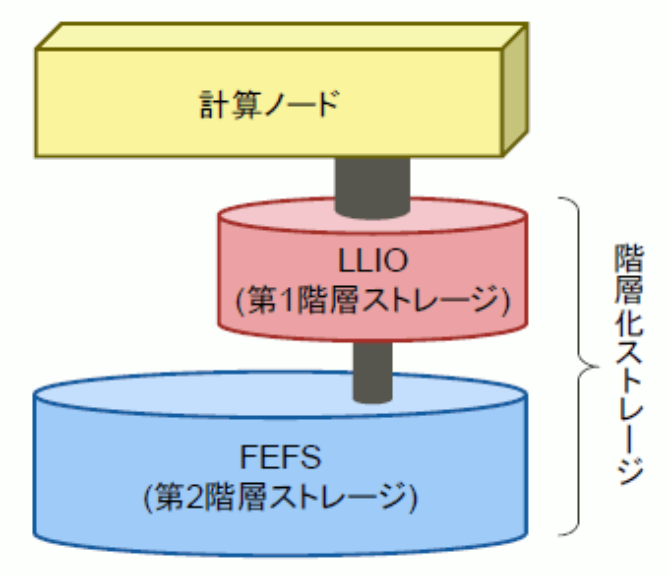
第3章 関連ソフトウェア

本章では、ジョブ運用ソフトウェアの関連ソフトウェアについて紹介します。

3.1 LLIO、FEFS

Technical Computing Suiteは、ファイルシステムのLLIOとFEFSを提供します。
LLIOを第1階層ストレージ、FEFSを第2階層ストレージとする階層化ストレージは、それぞれの特性を最大限に生かし、高速かつ大容量なファイルシステムを実現します。

図3.1 LLIOとFEFS



3.1.1 LLIO

LLIOは、FEFSと計算ノードの間に高速なフラッシュメモリを使用したストレージ階層(第1階層ストレージ)を設け、FEFSのキャッシュ領域やジョブの一時領域として使用することで高性能を実現するファイルシステムです。LLIOは以下の特長を持っています。

ジョブに最適なLLIO領域の構築

LLIOは、第2階層ストレージのキャッシュ領域、共有テンポラリ領域、およびノード内テンポラリ領域という異なる3種類の領域を持ち、エンドユーザはジョブ投入時に最適な大きさの領域を構築できます。

ファイルの高速アクセス

LLIOは、ファイルの高速アクセスを実現するため、以下の機能を実現します。

- ストライプ機能
- 共通ファイル配布機能
- 計算ノード内キャッシュ機能

統計情報

LLIOは、多くの統計情報を採取し、ジョブ実行者およびシステム管理者に提供しています。これらの情報は、ジョブのI/Oチューニングをするときや、システムトラブルの調査をするときに役立ちます。

LLIOの利用については、LLIOのマニュアルを参照してください。

3.1.2 FEFS

FEFSは、オープンソースのファイルシステムであるLustreの技術に基づいた大規模および高性能な並列分散ファイルシステムです。以下の特長を持っています。

大規模

10万ノードのクライアント、8EiB(8×2^{60})のファイルシステムサイズをサポートしています。

高性能

ストライプ、ラウンドロビンの手法を使用して、ファイルデータをストレージに分散格納することによって、I/O性能を向上させています。

使いやすさ

クライアント間のI/O優先制御/ユーザー間フェアシェア(QoS機能)などを通じて、大量のI/Oをするほかのユーザーの影響を軽減しています。

高信頼

MGS(Management Server)、MDS(Meta Data Server)、およびOSS(Object Storage Server)のフェイルオーバ機能を持っています。

拡張性

メタデータ領域/データ格納領域の動的な拡張が可能です。マルチMDS機能をサポートし、MDS/MDTの数によりスケーラブルな性能向上が可能です。

FEFSの利用については、FEFSのマニュアルを参照してください。

3.2 Development Studio

Development Studioは、Fortran、C言語、およびC++言語による、高性能な並列プログラムの開発から実行までを支援するソフトウェアです。

Development Studioには、以下の特徴があります。

- 高性能な並列プログラムの開発を支援
- 大規模並列プログラムの効率的な開発を支援
- 可搬性の高いプログラムの開発を支援

詳細は、Development Studioのマニュアルを参照してください。

3.2.1 コンパイラ

Fortranコンパイラ、Cコンパイラ、およびC++コンパイラは、各言語で記述されたプログラムを翻訳し、対象となる計算ノード向けに、CPUの実行性能を十分に引き出せるように高度に最適化された実行可能プログラムを作成できます。また、自動並列化やOpenMP仕様により、スレッドレベルで並列実行可能な実行可能プログラムを作成できます。

主要機能

各コンパイラは、翻訳時オプションを指定するだけでコンパイラが自動的にスレッドレベルでの並列処理をするプログラムを作成できる「自動並列化機能」を備えています。さらに、プログラム内に指定する指示行によりスレッドレベルでの並列処理をする「OpenMP API仕様」をサポートしています。自動並列化機能、およびOpenMP仕様による並列処理機能は、共有メモリシステムを前提としており、その機能は計算ノード内で有効です。これらの機能は、MPIライブラリと併用することで高効率なハイブリッド並列プログラミングモデル(スレッド並列＋MPIプロセス並列)を支援します。

最適化機能

各コンパイラは、以下に示すような最適化機能を備えており、計算ノード上で高速に実行できるオブジェクトプログラムを作成できます。さらに、プログラム内からこれらの最適化を促進させるための様々な最適化制御行も提供します。

- 多重ループの構成変更をする最適化
- CPUの特性に適した命令スケジューリング機能
- プリフェッチ命令によるキャッシュの効率的な利用
- SVEを活用したSIMD化による並列度の向上
- レジスタ内容の退避・復元回数の削減
- A64FXプロセッサのHPCタグアドレスオーバーライド機能を効果的に利用する最適化機能

3.2.2 数学ライブラリ

日本国内のR&Dユーザーに幅広く利用されている富士通独自の数学ライブラリ(SSL IIおよびC-SSL II)に加えて、米国で開発された線型代数分野ライブラリ(BLAS、LAPACK、ScaLAPACK)を提供します。また、4倍精度の値をdouble-double形式で表現し、演算をする高速4倍精度基本演算ライブラリを提供します。

これら数学ライブラリはどれも、各計算ノード向けに、最適な実行性能が得られるようにチューニングされています。

3.2.3 通信ライブラリ

MPIライブラリ

MPIライブラリは、MPIフォーラムで規定されている規格に準拠しています。Tofuと呼ばれる6次元メッシュ/トーラスで構成されたインターコネクトに対応し、高性能化、省メモリ化を実現しています。

uTofu

uTofuは、ユーザー空間のソフトウェアがTofuインターコネクトを使用して通信をするための、低レベルなアプリケーションプログラミングインターフェース(API)です。uTofuは、Tofuインターコネクトのワンサイド通信とバリア通信をサポートします。

3.2.4 プログラム開発支援ソフトウェア

プロファイラ

プロファイラは、Fortran、C言語、またはC++言語によって作成されたアプリケーションプログラムに対して、性能分析に必要となる各種情報を取得できる性能解析ツールです。プロファイラは、スレッド並列およびMPIプロセス並列に対応したプログラムについてもプロファイラ情報を取得できます。

プロファイラは、以下の機能から構成されます。

- 基本プロファイラ
- 詳細プロファイラ
- CPU性能解析レポート

並列実行デバッガ

並列実行デバッガは、Fortran、C言語、またはC++言語によって作成されたMPIライブラリを呼び出すアプリケーションプログラムに対するデバッグツールです。

並列実行デバッガは、以下の機能から構成されます。

- 異常終了調査機能
- デッドロック調査機能
- 重複除去機能
- コマンドファイルによるデバッガ制御機能

統合開発環境

オープンソースの統合開発環境として最もメジャーで実績があるEclipseを採用しています。並列プログラム開発向けプラグインParallel Tools Platformを利用して、ジョブスケジューラーと連携し、ジョブの投入、ジョブの状態確認などができます。

付録A　マニュアル一覧

ここでは、ジョブ運用ソフトウェアのマニュアル一覧を示します。

表A.1 マニュアル一覧

マニュアル名	概要	対象読者
ジョブ運用ソフトウェア 概説書	本書。 このマニュアルは、ジョブ運用ソフトウェアとその関連ソフトウェアの概要を説明します。	エンドユーザ 管理者
ジョブ運用ソフトウェア エンドユーザ向けガイド	このマニュアルは、ジョブ運用ソフトウェアでアプリケーション(ジョブ)を実行する方法を説明します。	エンドユーザ
ジョブ運用ソフトウェア エンドユーザ向けガイド HPC拡張機能編	このマニュアルは、エンドユーザがHPC拡張機能を使う方法を説明します。	エンドユーザ
ジョブ運用ソフトウェア エンドユーザ向けガイド マスタ・ワーカ型ジョブ編	このマニュアルは、マスタ・ワーカ型ジョブの作成方法について説明します。	エンドユーザ
ジョブ運用ソフトウェア 導入ガイド	このマニュアルは、ジョブ運用ソフトウェアの導入方法を説明します。	管理者
ジョブ運用ソフトウェア 管理者向けガイド システム管理編	このマニュアルは、ジョブ運用ソフトウェアを導入したシステムの運用・管理方法を説明します。	管理者
ジョブ運用ソフトウェア 管理者向けガイド ジョブ管理編	このマニュアルは、ジョブ運用ソフトウェアを導入したシステムのジョブ運用・管理方法について説明します。	管理者
ジョブ運用ソフトウェア 管理者向けガイド ジョブ運用管理機能フック編	このマニュアルは、ジョブ運用管理機能のフックの利用方法について説明します。	管理者
ジョブ運用ソフトウェア 管理者向けガイド 電力管理編	このマニュアルは、ジョブ運用ソフトウェアを導入したシステムの電力管理について説明します。	管理者
ジョブ運用ソフトウェア 管理者向けガイド HPC拡張機能編	このマニュアルは、管理者がHPC拡張機能を使う方法を説明します。	管理者
ジョブ運用ソフトウェア 管理者向けガイド 保守編	このマニュアルは、ジョブ運用ソフトウェアを導入したシステムの保守の方法について説明します。	管理者
ジョブ運用ソフトウェア APIユーザーズガイド コマンドAPI編	このマニュアルは、ジョブ運用管理機能のコマンドAPIの利用方法について説明します。	エンドユーザ 管理者
ジョブ運用ソフトウェア APIユーザーズガイド Power API編	このマニュアルは、電力管理機能のPower APIの利用方法について説明します。	エンドユーザ
ジョブ運用ソフトウェア APIユーザーズガイド ジョブ情報通知API編	このマニュアルは、ジョブ運用管理機能のジョブ情報通知APIの利用方法について説明します。	管理者
ジョブ運用ソフトウェア APIユーザーズガイド スケジューラーAPI編	このマニュアルは、ジョブ運用管理機能のスケジューラーAPIの利用方法について説明します。	管理者
ジョブ運用ソフトウェア トラブルシューティング集	このマニュアルは、システムの導入や運用での管理者向けのトラブルシューティング集です。	管理者
ジョブ運用ソフトウェア コマンドリファレンス	このマニュアルは、ジョブ運用ソフトウェアのコマンドのリファレンスマニュアルとメッセージ集です。	エンドユーザ 管理者
ジョブ運用ソフトウェア 用語集	このマニュアルは、ジョブ運用ソフトウェアに関する用語を説明します。	エンドユーザ 管理者

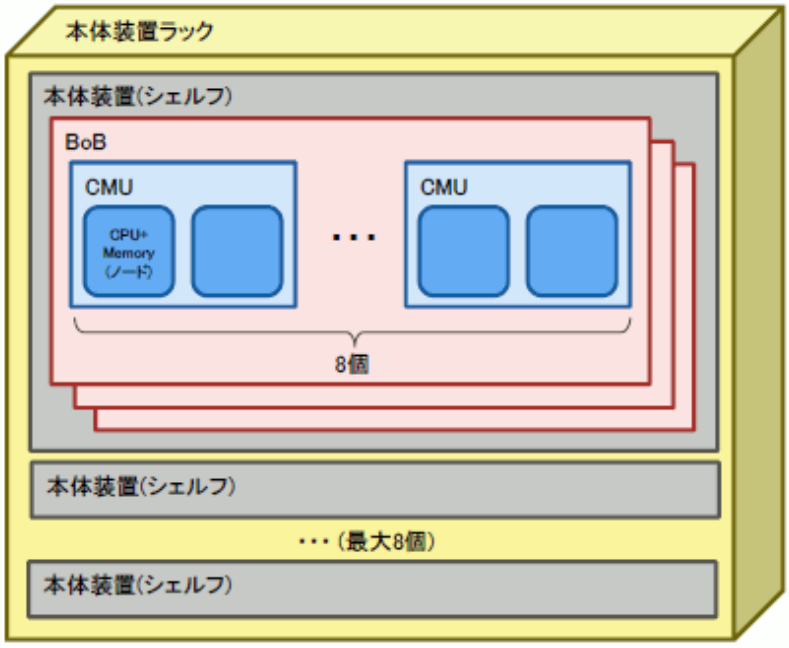
付録B　FXサーバ固有の管理構造

ここでは、FXサーバのハードウェア固有の管理構造について概要を説明します。

B.1　FXサーバのハードウェアの構成要素

FXサーバのハードウェアの構成要素は、以下のようなイメージになります。

図B.1 FXサーバのハードウェアの構成(概略)



表B.1 FXサーバのハードウェアの構成要素

構成要素	説明
CMU (CPU Memory Unit)	CPUやメモリを搭載したユニットで、2台の計算ノードに相当します。
BoB (Bunch of Blades)	BoBはFXサーバの制御単位で、8台のCMUで構成されます。すなわち、1個のBoBには16台の計算ノードが含まれます。 BoB内の計算ノードのうち、3台が入出力機能を持つノードで、それぞれブートI/Oノード、ストレージI/Oノード、およびグローバルI/Oノードを兼ねます。システムによって、BoB内のこれらのI/Oノードの数は異なります。 ブートI/Oノードには、各ノードを起動するためのシステムディスクが接続されます。ストレージI/Oノードは、ジョブ実行時に高速なテンポラリ領域として利用できるディスク装置(SSD)が接続されます。グローバルI/Oノードは、入出力用インターフェースを経由して、第2階層ストレージに接続されます。
本体装置	本体装置は3個のBoBで構成されます。なお、本体装置をシェルフと呼ぶ場合もあります。
本体装置ラック	本体装置を搭載する筐体です。本体装置ラックには本体装置を最大8個搭載できます。

B.2　Tofu単位とTofu座標

FXサーバの各ノードは、Tofuインターコネクトと呼ぶ高速なネットワークで相互に接続されています。Tofuインターコネクトは計算用ネットワークとして、並列ジョブにおけるノード間通信路として使用されます。
FXサーバではTofuインターコネクトで接続された12ノードを1つの単位として扱い、これを「Tofu単位」と呼びます。Tofu 単位は、2x3x2の直方体としてみなすことができ、各軸をA、B、C 軸と呼びます。Tofu単位はX、Y、Z軸からなる3次元の座標に配置されるものとして管理されます。したがって、FXサーバのノードは、X、Y、Z、A、B、およびC軸で表現される6次元座標で位置が定義されます。この6次元座標をTofu座標と呼びます。Tofu座標X、Y、Zの上限は、システムの規模や構成によって異なります。

図B.2 Tofu単位とTofu座標