スーパーコンピュータ「富岳」の運用系ソフトウェア

宇野　篤也　　末安　史親　　関澤　龍一

あらまし

近年，スーパーコンピュータやPCクラスタといったHPC（High Performance Computing）システムの高並列化・大規模化が進んでいる。これらの大規模かつ高性能な計算機環境を安定して運用しユーザーに提供することは重要な課題であり，システムの管理・運用を担う運用系ソフトウェアは重要なコンポーネントである。スーパーコンピュータ「富岳」（以下，「富岳」）向けの運用系ソフトウェアの開発は，8年間にわたるスーパーコンピュータ「京」（以下，「京」）の運用経験とそのソフトウェア技術をベースに，「京」からの継続性，「京」の運用課題の解決による安定した運用の実現，使い勝手の良いシステムの追求などの観点から実施された。

本稿では，「富岳」向けの運用系ソフトウェアにおいて改善・追加された機能について述べる。

1. まえがき

スーパーコンピュータ「富岳」（以下,「富岳」）[1] の運用系ソフトウェアの設計は，スーパーコンピュータ「京」（以下,「京」）[2，3] の運用経験とソフトウェア技術をベースに,「京」からの継続性,「京」の運用課題の解決による安定した運用の実現,使い勝手の良いシステムの追求などの観点から実施された。「富岳」の運用系ソフトウェアは大きく分けて，運用・システム管理系ソフトウェア，ジョブ管理系ソフトウェア，ユーザー管理系ソフトウェアで構成されており，今回の開発では全般にわたり改善を行った。

本稿では,「富岳」向けの運用系ソフトウェアにおいて改善・追加された機能について述べる。

2. 運用・システム管理系ソフトウェア

運用・システム管理系ソフトウェアは，多数のハードウェアやソフトウェアを一つのシステムとして管理・運用するためのソフトウェアである。システムの一部が故障した場合でも，全体として正常に運用を継続できるだけでなく，必要に応じてシステムを複数に分割し，それぞれを独立した環境として運用することも可能とする。

一般的に，システムの規模が大きくなると，故障の可能性が高くなる。システムの高可用性を実現し，運用停止時間を最小限にするためには，異常発生の迅速な検出と対処が必要不可欠である。「富岳」では,「京」の保守状況の分析結果を基に，以下のような改善を実施した。

2.1 システム管理機能の改善

計算ノードなどで異常が発生した場合，運用ソフトウェアが異常を検知し，当該ノードを運用から切り離すなどの処置を行う。そのため，異常検知に時間がかかると，運用への影響が大きくなる。「京」では定期的な監視のみであったため，異常を検知するまでに時間を要する場合があった。「富岳」では，サービスに異常が発生した場合に監視システムに通知を行う方式に変更し，異常検知に要する時間を短縮した。

2.2 保守時間の短縮

計算リソースの提供率を改善するためには，保守によるシステム停止時間を短くする必要がある。「京」では，システムの再起動に時間を要したため,ソフトウェア保守に最大2日程度を要することもあった。一方,「富岳」では起動処理の最適化を行い,再起動時間を「京」の半分程度まで短縮した。例えば，計算ノード群の起動処理では，突入電流や電力変動を考慮した多重起動の最適化を行い，再起動処理では保守内容に最適な方式｛ノード再起動（コールドリブート [4]），ウォームリブート [5]，サービス再起動 [6] など｝を適用するなどの改善を行った。

ハードウェア保守では，作業時間を要するストレージシステムの保守作業などの改善を行った。「富岳」のRAS（Reliability，Availability and Serviceability）は「京」の方針を踏襲しており,システム運用への影響が大きい管理系ノードは冗長化構成となっている。冗長構成になっていない計算ノードなどが故障した場合は，それらを運用から切り離したあと，故障機器を交換し再び運用に組み込む。ストレージシステムなどの故障では，保存されているソフトウェアも削除されることがあり，その場合はソフトウェアも復元する必要がある。「京」では，計算ノードのシステムディスクが故障した場合にはソフトウェアの再インストールと再設定が必要となり，保守作業が長期化する一要因であった。これを改善するために,「富岳」ではシステムディスクをディスクイメージの形で保持し，故障時にはイメージをコピーするだけで復旧できる方式とした。これにより，保守作業時間を「京」の半分程度まで短縮した。

2.3 ログ解析の効率化

システムの規模が大きくなると，各サーバなどから出力されるログも大量になる。これらのログを分析して行うシステムの運用状況の確認や障害調査などは,「京」では機能ごとのログのフォーマットなどが統一されていなかったため，効率的に分析することが難しく，分析に時間を要した。「富岳」では,運用管理系ソフトウェアのログフォーマットの統一化，ログの一元管理，および関連ログのひも付けを行い，障害調査・稼働分析を迅速に行える環境を

整備した。分析時間の短縮は，保守時間の短縮にもつながる。また，ログの収集および解析にはオープンソースソフトウェアを活用し，作業の効率化を図っている。オープンソフトウェアを活用することで，最新の技術を速やかに導入することができ，様々な事象に柔軟に対応できるなどのメリットがある。

3. ジョブ管理系ソフトウェア

ジョブ管理系ソフトウェアの主な機能は，ジョブマネージャー機能とジョブスケジューラ機能で，ジョブ管理や資源管理，ジョブスケジューリングなどを行う。以下に改善点について述べる。

3.1 ジョブマネージャー機能

「富岳」では，ジョブマネージャー機能に対して，以下のような改善を行った。

(1) 仮想マシンのサポート

「京」では物理マシンのジョブ実行環境のみをサポートしていたが，「富岳」ではKVMなどの仮想マシンのジョブ実行環境もサポートする。仮想マシンの実行環境は，システム管理者が用意する環境の他に，ユーザーが独自に用意した環境も利用することができる。仮想マシンはジョブ単位で起動され，他のジョブに影響を及ぼさずに実行することができる。

(2) カスタム資源のサポート

「富岳」では，特定計算ノード上の任意の資源（カスタム資源）を管理する機能を実装した。カスタム資源では，ハードウェア機能だけでなく，電力やソフトウェアライセンスなどの仮想的な資源も管理することができる。ユーザーがジョブ投入時に利用したいカスタム資源を要求すると，スケジューラが対象となる計算ノードを自動で割り当てる。商用アプリケーションのソフトウェアライセンスなどもカスタム資源として定義できるため，ユーザーはソフトウェアライセンス数を意識することなく，商用アプリケーションを利用することが可能である。

3.2 ジョブスケジューラ機能

「富岳」では，ジョブスケジューラ機能に対して，以下のような改善を行った。

(1) ユーザーインターフェイスの改善

全てのユーザーに対して使い勝手の良いコマンドを提供することは難しい。「富岳」では，各種コマンドのオプション体系の改善やコマンドの独自実装を可能とする，操作コマンド向けのカスタマイズAPI（Application Programming Interface）を実装した。このAPIを使うことにより，ユーザー自身がコマンドをカスタマイズすることも可能である。

(2) スケジューリングアルゴリズムのカスタマイズ機能

「富岳」では，標準で用意されているアルゴリズムの他に，センター運用の最適化を実現可能とするスケジューラAPIを実装した（図-1）。このAPIにより，ジョブスケジューラの機能のうち，(a) ジョブ選択処理部と (b) 資源選択処理部をシステム管理者がカスタマイズすることができる。(a) は，ジョブスケジューリング時の優先度制御を，(b) は計算ノードの割り当て制御を行う機能で，スケジューラAPIを用いることでこの部分を任意のアルゴリズムに置き換えることができる。

(3) ノード利用率の改善

「京」では，ジョブは最長でも指定した経過時間までしか実行できなかった。したがって，後続ジョブの実行まで間隔が空いた場合，計算ノードは待機状態となり，ノードの利用率が低下する。これを改善するために，「富岳」では後続ジョブの実行を阻害しない限り実行を継続できる，新しいジョブの実行方式を実装した（図-2）。この実行方式では，ユーザーの指定経過時間までジョブを実行したあと，後続ジョブのスケジューリング状況に応じて経過時間制限を動的に変更する。これにより，ノード利用率を改善することができる。

また，ジョブのIO性能を維持しつつ，ノード利用率を改善することを目的としたスケジューリング機能を実装した。「富岳」では，第一階層ストレージを管理するSIO（ストレージIO）ノードを48ノード単位（Shelf単位）で共有する構成になっている（図-3）。そのため，ジョブに第一階層ストレージのIO帯域を占有させるためには，SIOノードと計算ノード間を接続するTofuネットワーク [7] の構成を意識したスケジューリングが必要となる。この場合，Shelf単位（2×3×8）がスケジューリング単位と

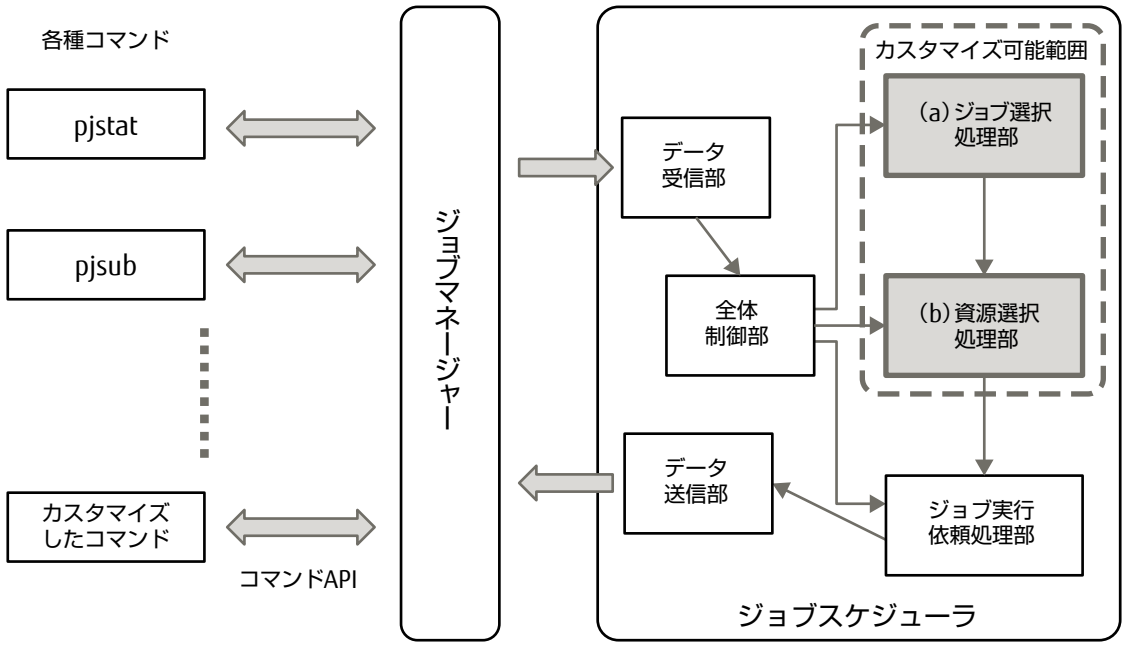

図-1　カスタマイズ可能な機能

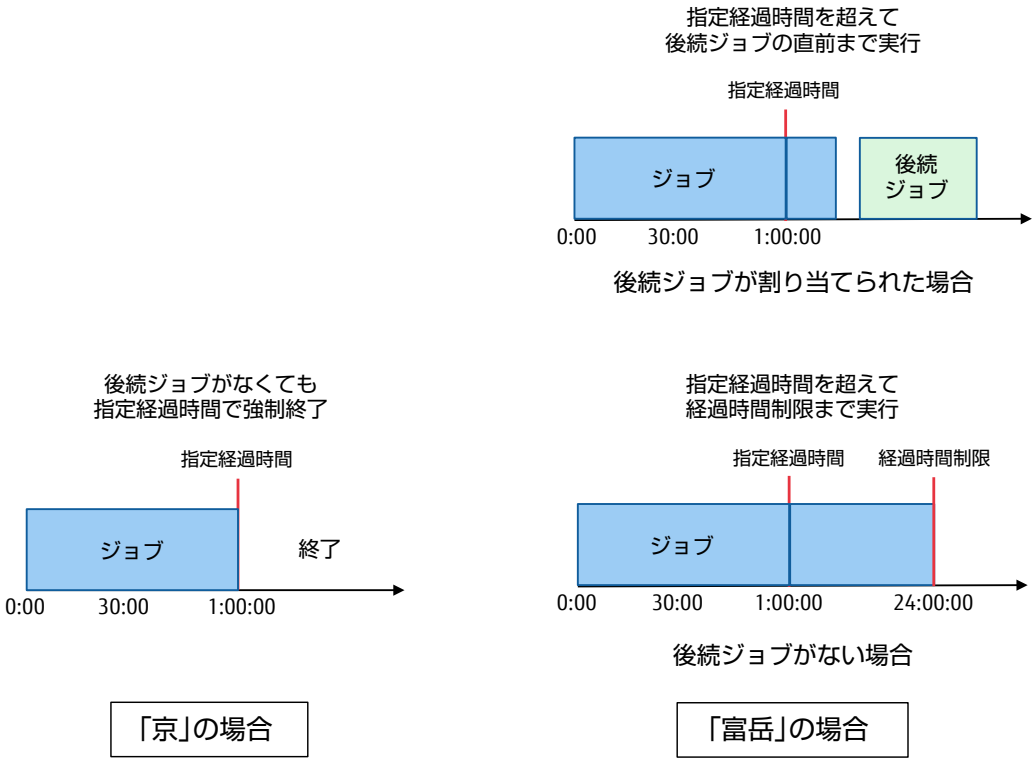

図-2　経過時間制限の動的変更

なるため，この形状を意識したスケジューリング機能を実装した。また，小さいジョブが分散してスケジューリングされると規模の大きいジョブのIOを阻害する場合がある。これを防ぐため，規模の小さいジョブをできるだけ特定のShelfにまとめる機能も実装した。これにより，大小のジョブが混在する

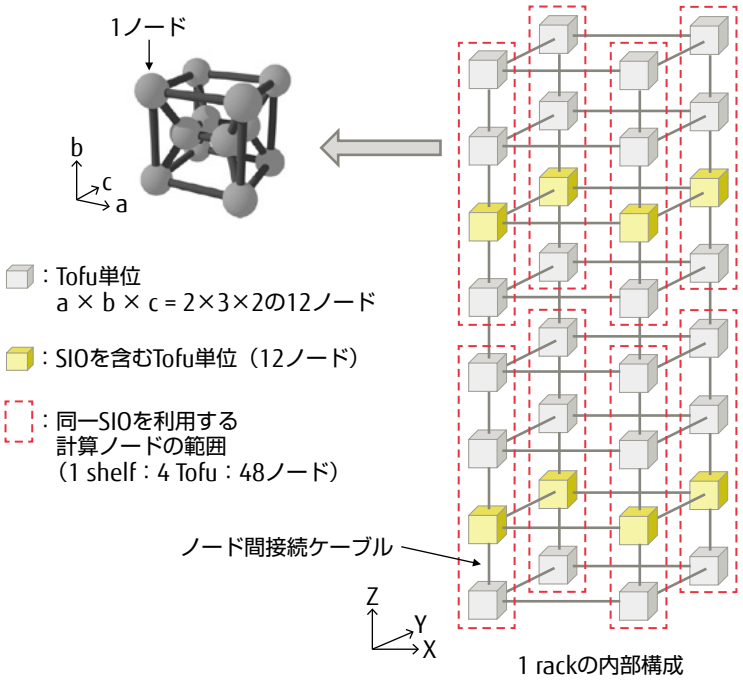

図-3　計算ラックの内部構成

環境でもIO性能を維持しつつノード利用率の改善が見込める。

(4) 電力問題への対応

「富岳」の消費電力は「京」よりも大きく，システム全体の消費電力が施設の許容範囲に納まるように運用する必要がある。「京」では，全系を使用する大規模ジョブを実行した際に，消費電力が上限値を超える場合があったため，事前に小規模ジョブの消費電力の測定結果から大規模ジョブの消費電力を推測し，実行時に上限値を超えないように制御を行っていた。「富岳」では，通常運用時にも上限値を超過する可能性がある。そのため，ジョブごとの消費電力を過去の実行実績を基に推測し，システム全体の消費電力が上限値を超えないようにスケジューリングする機能を実装した。

4. ユーザー管理系ソフトウェア

ユーザー管理系ソフトウェアは，ユーザーのアカウント管理や利用資源の管理を行うソフトウェアである。本章では，「富岳」におけるユーザー管理系ソフトウェアの改善点を述べる。

4.1 課金管理の拡張

「京」では，一般的なシステムと同様に，計算資源は課題単位で管理していた。そのため，課題が複数のサブ課題で構成されている場合に，課題内でサブ課題に割り当てられた資源量をユーザーが自由に調整することができなかった。この問題を解決するために，「富岳」では計算資源を階層管理とし，各課題の代表者が自由にサブ課題間で資源量を移動できるようにした。図-4に，資源管理の階層構造を示す。「京」ではグループに課題を割り当てていたが，「富岳」ではサブテーマに課題を割り当てる。運用者がテーマを設定したあと，各ユーザーはサブテーマ群よりも下位層の資源量を操作することができる。

「京」の資源量管理は，ジョブの実行時間とノード数の積である「ノード時間積」のみで行っていた。「富岳」では，「ノード時間積」以外の任意の資源量を管理できるように機能拡張を行った。これにより，課金計算に必要な情報が取得できる資源であれば，例えば「電力量」なども管理対象にすることが可能である。

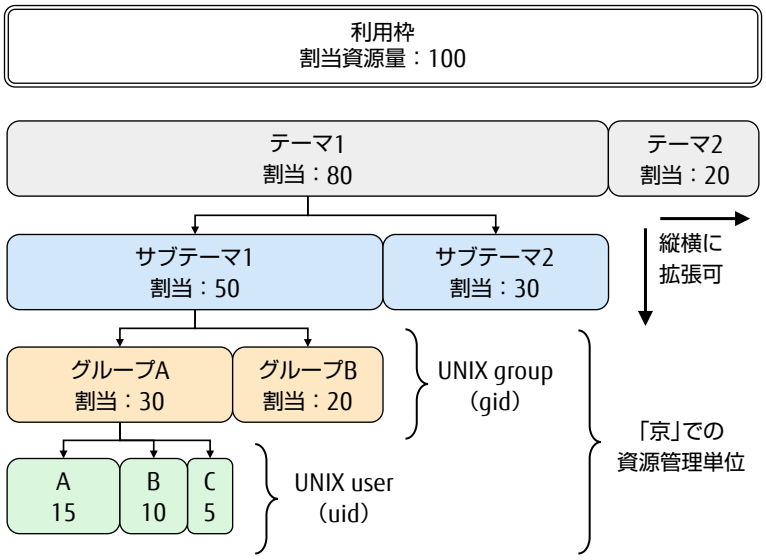

図-4　資源管理の階層構造

4.2 ファイルシステム障害時の運用改善

「富岳」では「京」と同様に，ユーザーのデータが保存される第二階層ファイルシステムは，複数ボリュームで構成されている。「京」では，ファイルシステムで障害が発生した場合，故障ボリュームのみを切り離して運用することができなかったため，障害復旧作業のために運用を停止する場合があった。この問題を解決するために，「富岳」ではシステム管理機能と連携し，障害が発生しているボリュームの利用を自動で回避し，運用を継続する機能を実装した。この機能により，ボリュームの一つが故障した場合でも，システム全体を停止することなく運用を継続することが可能となった。

5. むすび

本稿では，「富岳」向けの運用系ソフトウェアにおいて追加・改善された機能について紹介した。実際の運用ではこれらの機能を活用しつつ，運用状況に合わせた様々な取り組みが必要になると考えている。例えば，クラウド的利用やビッグデータとAIの活用などSociety 5.0 [8] の実現に必要な新たな技術への対応である。今回の開発では，「京」の運用経験を基に拡張性や柔軟性を重視した開発を行ってきたが，将来登場するであろう新規技術への対応は，今後の重要な課題の一つであると考えている。

参考文献・注記

[1] 理化学研究所 計算科学研究センター：スーパーコンピュータ「富岳」について.
https://www.r-ccs.riken.jp/jp/fugaku
[2] 特集：スーパーコンピュータ「京」. 情報処理. Vol. 53, No. 8, p. 752-807（2012）.
https://ipsj.ixsq.nii.ac.jp/ej/index.php?action=pages_view_main&active_action=repository_view_main_item_snippet&index_id=6578&pn=1&count=20&order=7&lang=japanese&page_id=13&block_id=8
[3] 理化学研究所 計算科学研究センター:「京」について.
https://www.r-ccs.riken.jp/jp/k/
[4] 電源断状態にしてからの再起動。
[5] 電源は停止しない状態からの再起動。
[6] 関連プログラムのみの再起動。
[7] 安島雄一郎：スーパーコンピュータ「京」「富岳」を実現した高次元接続技術. 富士通テクニカルレビュー.
https://www.fujitsu.com/jp/about/resources/publications/technicalreview/topics/article005.html

［8］ 内閣府：Society 5.0とは.
https://www8.cao.go.jp/cstp/society5_0/

著者紹介

宇野　篤也（うの　あつや）
国立研究開発法人理化学研究所
計算科学研究センター
運用技術部門 システム運転技術ユニット ユニットリーダー
スーパーコンピュータシステムの運用・高度化に関する研究開発に従事。

末安　史親（すえやす　ふみちか）
富士通株式会社
コンピューティング事業本部
スーパーコンピュータ「富岳」の導入構築・運用に従事。

関澤　龍一（せきざわ　りゅういち）
富士通株式会社
コンピューティング事業本部
スーパーコンピュータ「富岳」の導入構築・運用に従事。