スーパーコンピュータ「京」の運用管理ソフトウェア

Operations Management Software for the K computer

● 平井浩一　● 井口裕次　● 宇野篤也　● 黒川原佳

あらまし

スーパーコンピュータシステムは，とどまることのない計算需要の増大に呼応する形でシステム規模（CPUコア数やノード数など）が増大の一途をたどっている。このような，超大規模システムを安定稼働させるとともに，利用者に対して高性能な計算環境を高い稼働率を維持して提供するためには，運用管理ソフトウェアが重要な鍵を握ることとなる。富士通では，従来のスーパーコンピュータシステム向け運用管理ソフトウェアとして「Parallelnavi」を開発し，3000ノード規模のシステムの一元管理を実現してきた。スーパーコンピュータ「京」は，従来規模をはるかに凌駕（りょうが）する8万個以上のノードから構成される超大規模システムであり，この規模を安定かつ効率的に一元管理するため，更なる技術開発を行った。

本稿では，「京」向けに開発した運用管理ソフトウェアについて，超大規模システムを安定稼働させるためのシステム運用管理機能の実現方式，および高性能な計算環境を提供するジョブスケジューラの特徴について説明する。

Abstract

Supercomputer systems have been increasing steadily in scale (number of CPU cores and number of nodes) in response to an ever increasing demand for computing power. Operations management software is the key to operating such ultra-large-scale systems in a stable manner and providing users with a high-performance computing environment having a high utilization rate. Fujitsu previously developed software called “Parallelnavi” to provide uniform operations management for its 3000-node supercomputer systems, but to uniformly manage an ultra-large-scale system like the K computer on a scale of more than 80 000 nodes, it expanded its development of operations management technologies. This paper introduces operations management software developed for the K computer, focusing on functions for achieving stable operation of an ultra-large-scale system and a job scheduler for providing a high-performance computing environment.

ま え が き

スーパーコンピュータ「京」(注1) のような超大規模システムの運用管理においては，以下の3点がシステムの安定稼働と高性能な計算環境を提供するための課題となる。

・運用管理ソフトウェアのオーバヘッド削減
・膨大な情報のサマライズ表示
・バッチジョブ間の相互干渉による実行性能低下の抑止

著者らは，これらの課題へ対応するとともに，ハードウェアの持つ高い演算性能を効果的に引き出すことを目標に運用管理ソフトウェアの更なる技術開発を行った。

本稿では，「京」向けの運用管理ソフトウェアにおける上記の課題解決への取組み，および特徴について紹介する。

運用管理ソフトウェアの構成

「京」向けの運用管理ソフトウェアは，大きく以下の二つの機能から構成される。

(1) システム管理機能

システム管理者に対し，シングルシステムイメージで運用管理ビューを提供する。システムの構成情報を管理し，構成要素である各ノードの稼働状況を監視するとともに，異常発生に対し，高い可用性を提供する。また，ソフトウェアのインストールや，パッチ適用などのソフトウェア構成の管理も行う。

(2) ジョブスケジューラ

複数の利用者に対し，バッチジョブの実行環境を提供することにより，スーパーコンピュータシステムの共同利用を実現する。

次章以降でこれら二つの機能の特徴を紹介する。

システム管理機能

● システム管理機能における大規模対応

システム管理機能は，異常発生などに備え，定常的にシステムの運用状況を監視している。システムが超大規模化するに伴い，このような定期監視のためのネットワーク負荷や計算ノード上での監視処理が，無視できないほど大きなものとなってしまう。このためシステム管理機能では，以下のような取組みを行い，今後更にシステム規模が拡大しても，それに追従できることを目指した。

・階層化構造による監視処理などの負荷分散
・Tofuインターコネクトを活用したシステムノイズ削減

上記のほか，各ノードの稼働状態をサマライズして表示するなど，超大規模システムに対する工夫を実施している。

● 階層化構造による負荷分散

システム管理者にシングルシステムイメージでの運用操作環境を提供するには，システム管理者が操作するPC（管理用PC）やサーバへの負荷集中を回避する必要がある。8万個以上のノード規模の環境で単純に各ノードが管理用PCへ監視データをネットワーク経由で送信すると，途中経路のネットワークや管理用PCの処理能力を超えてしまうこととなる。また，今後もますますノード数は増大するものと考えられ，拡張性のある負荷分散が必要となる。

「京」の運用ソフトウェアでは，以下のような階層構造のソフトウェアアーキテクチャを採用することにより，運用管理処理の負荷分散と並列化を実現している。

図-1に示すようにこの階層化構造では，システムをノードグループに分割し，各ノードグループに属すノード群を管理するためにジョブ管理サブノードを配置する。ジョブ管理サブノードが当該ノードグループ内の運用管理処理をつかさどることにより，運用管理処理負荷を分散している。また，運用開始後にノードを増やす場合もノードグループを追加するだけで容易な負荷分散が可能となり，システムの高い拡張性が実現される。

さらにこの階層構造は，定常運用中のノードの監視や操作だけでなく，システムのインストールやパッチなどの保守作業の基盤としての役割，および並列プログラムの起動処理や終了処理など，ノード間で何らかの操作を伴うような場合の基盤としての役割も果たしている。複数のジョブ管理サブノードで運用管理処理を分散・並列化したことにより，従来3000ノードの並列プログラム起動に数秒を要していたところ，1秒以内での起動を実

(注1) 理化学研究所が2010年7月に決定したスーパーコンピュータの愛称。

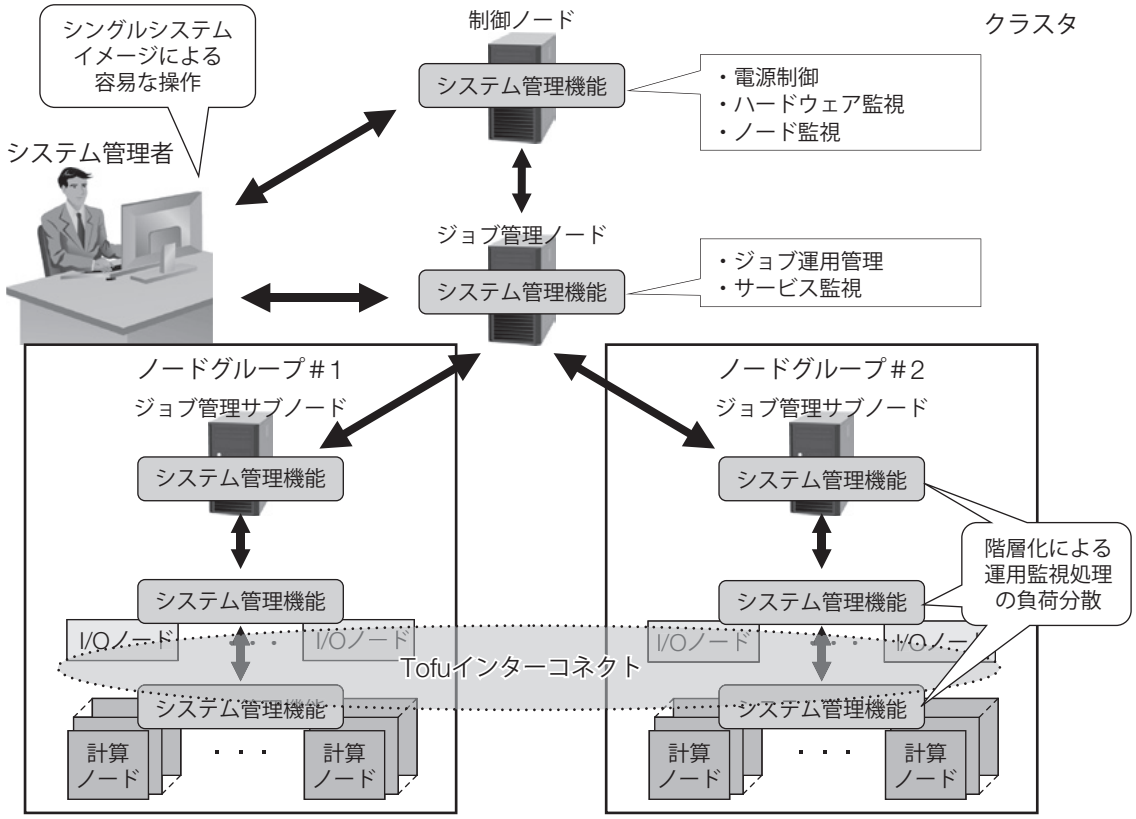

図-1　階層構造による負荷分散

現することができた。

このように運用管理の側面だけでなく超大規模並列プログラムの起動性能などの面でも，超大規模システムだからこそ無視できないシステム運用のための各種処理時間の短縮対策を実施している。

● Tofuインターコネクトを活用した稼働情報集約

運用管理ソフトウェアによるノードの稼働状況の監視や各種操作は，一般的にGigabitEthernetなどのネットワークからの割込みをデーモンプロセスが受け，当該ノードの情報採取を行い，再びGigabitEthernetを介して採取情報を管理用PCなどへ転送する方式を取っている。しかし，超大規模システムで同様の方式とするとネットワークからの割込みやデーモンプロセスの処理が，いわゆるシステムノイズとして発生し，並列プログラムの実行性能を低下させる大きな要因となってしまう。このため「京」向けの運用管理ソフトウェアでは，ネットワークからの割込みで情報採取・転送するのではなく，各ノードがそれぞれ独立に自ノードの状態をメモリ上に定期的に格納し，TofuインターコネクトのRDMA通信(注2) を活用し，外部から稼働情報を取得するという方式を採用した。更に各ノード上の稼働状態の監視や格納処理も，極力カーネルレイヤで実現し，システムノイズの発生を大幅に抑えた。そのほかのシステムノイズ対策も含め，PCクラスタでは1回がミリ秒オーダで発生するシステムノイズを1/100程度に抑えることができた。

● そのほかの機能

これまでに紹介した超大規模システムにおけるオーバヘッド削減対策のほか，運用管理機能としてシステムの安定稼働に必要な以下のような機能も実現している。

(1) 高可用システム

各種管理系ノードがダウンしても実行中のジョブはそのまま実行を継続する機能を提供し，異常発生時のジョブの実行継続性を保証している。これだけでは管理系ノードのダウン中に新規ジョブ受付などが行われなくなってしまうため，管理系

(注2)　Remote Direct Memory Accessの略。Tofuインターコネクトが持つ通信機能。

ノードの冗長化をサポートし，システム全体の運用継続性を保証している。

(2) サマライズ表示

管理対象となる8万個以上のノードの状態をターミナルへコマンドラインで単純に1ノード/1行で表示すると8万行以上の大量表示となる。このためシステム管理機能では，状態ごとに分類したノード数だけを表示するなどシステム全体像のサマリー情報表示を標準の表示形式とした。より詳細な情報が必要な場合は，ノードグループなどによる表示対象の絞込みや特定ノードを指定するなどのオプションを用意している。

ジョブスケジューラ

● 開発への取組み

富士通がこれまでに開発してきたスーパーコンピュータシステム向けのジョブスケジューラは，以下のような特徴がある。

- 計算資源の事前予約によるジョブ間の相互干渉抑止
- 複数ノードにまたがる並列ジョブの実現
- 共同利用センターの多種多様な要件への対応

「京」向けのジョブスケジューラでは，これらの特徴を継承しつつ，超大規模システムへ対応するための負荷分散，およびシステム稼働率の向上を大きなテーマとして開発に取り組んだ。

● ジョブスケジューラにおける超大規模対応

システム規模の拡大に伴い，ジョブスケジューラが扱うジョブ数も飛躍的に増大する。「京」では，100万を超えるジョブを扱うことが想定される。

「京」向けのジョブスケジューラは，ジョブの受付けから終了までのジョブ制御の処理ステップ（例えば受付処理，実行優先度決定処理，実行計算資源決定処理，ジョブ実行処理，終了処理など）を並列化し，処理時間を短縮した。また，投入された個々のジョブをどのような順番で，いつ，どのノードで実行するのかを決定する処理（スケジューリングと呼ぶ）もシステム規模に応じて最適解を求める計算量が増大するため，特に処理コストの高い，実行ノードを選択する処理を並列処理化することで計算時間を短縮した。これらの取組みにより，例えばジョブ投入性能においては，PCクラスタで使われる他社製品（PBSpro）が1ジョブ投入に0.4秒必要（当社調べ）であるのに対し，3ミリ秒の投入性能を実現した。

なお，システム管理機能と同様にジョブスケジューラにおいてもジョブに関する情報が同時に大量に出力されないよう，サマライズして表示するほか，**図-2**に示すように詳細情報を利用者が必要に応じて取捨選択することや表示時のソート

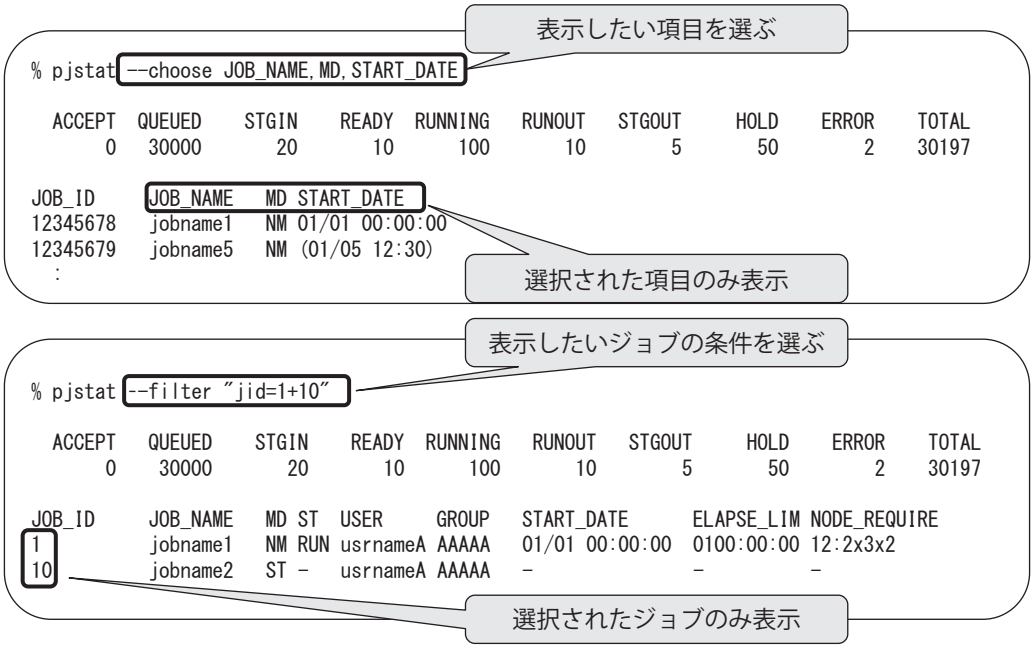

図-2　ジョブ情報の選択表示例

キーを指定可能とし，情報表示の簡素化を図っている。

● システム稼働効率向上への対応

共同利用センターにおいては，1ノードのみを必要とするジョブから数万ノードを必要とする超大規模並列ジョブまで様々な規模のジョブが混在した状態で運用される。このような環境では，例えば投入された順番にジョブを実行するだけでは，次に実行するジョブの規模が空いているノード（ジョブが実行されていないノード）数を超えると実行が待たされ，システム全体の稼働率の低下を招いてしまうことがある。

「京」のジョブスケジューラでは，バックフィルスケジューリングをサポートすることにより，この問題の解決を図っている。バックフィルスケジューリングは，大規模ジョブが実行を開始するまでに生じる計算資源の空き時間で大規模ジョブの実行予定時刻までに終了する小規模ジョブを先に実行させるものであり，稼働率を向上させることができる。

さらにジョブを実行する計算ノードの事前割当て時に以下のような工夫を行い，Tofuインターコネクトの特性を考慮し，稼働率の向上を図っている。

・隣接するノード群の事前割当てが容易となるように，未割当ての計算ノード群（空きノード）が，隣接（連続）するように制御する。

・上記制御に当たっては，Tofuインターコネクトを利用する3次元形状の様々なパターンで実行ノードの候補を探索し，連続した空きノードが最も多くなるよう制御する。空きノードを多く残すことにより，以降のノード割当てが容易となり，結果的に稼働率が向上することとなる。

● I/O競合への対応（ファイルステージング）

システムが超大規模化していくことに伴い，ジョブからのI/Oが競合することにより，ジョブの実行性能が低下する確率が高まる。

このため「京」では，ジョブの実行に必要なファイルをジョブ実行開始前にローカルファイルシステムへ事前転送し，ジョブ実行終了後に実行結果ファイルを回収する「ファイルステージング機能」をサポートした。

一般的なファイルステージング機能は，実行ジョブの一部としてファイル転送を行う（同期ステージングと呼ぶ）が，この手法ではファイル転送中もジョブ実行に必要な計算資源が予約されたままとなり，計算資源の無駄が発生する。「京」のステージング機能では，**図-3**に示すようにファイル転送とジョブ本体とを分離して実行させることにより，計算資源の無駄の発生を防いでいる。すなわち，

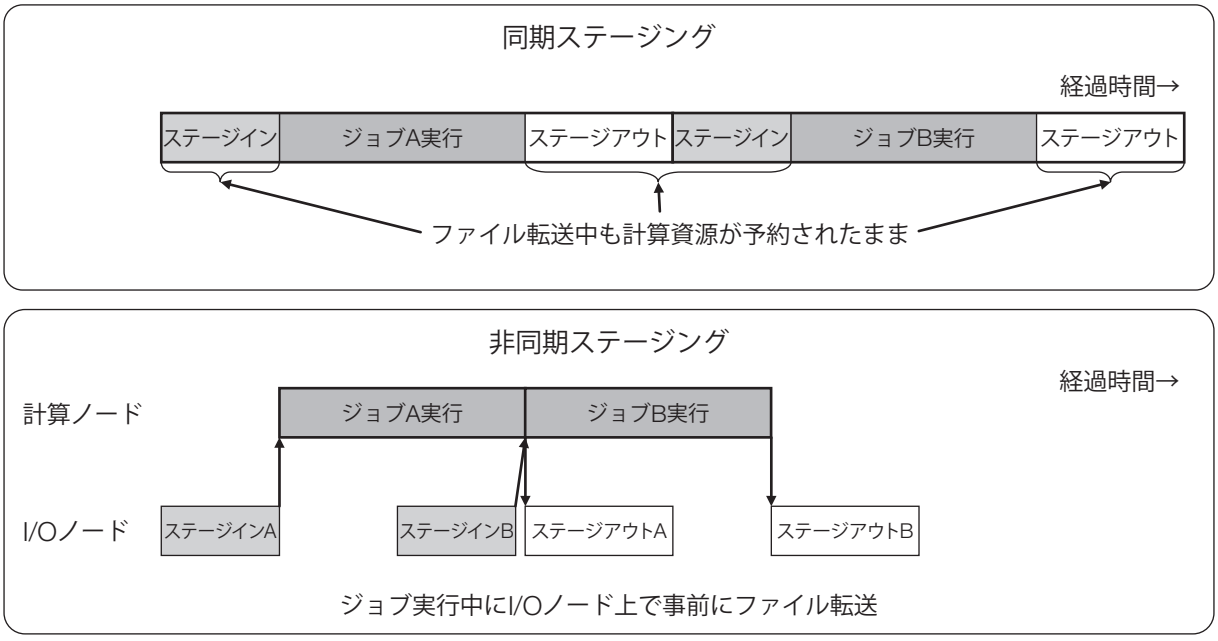

図-3　非同期ステージングによる稼働率の向上

システム構成上の特長である計算ノードと独立したI/Oノードの存在を活用し，I/Oノード上でファイル転送処理を行うことにより，ジョブ本体の実行と非同期にファイル転送を行っている（非同期ステージングと呼ぶ）。これにより，ジョブの実行開始までにファイル転送を完了させておくことで，ジョブの実行時間から，見かけ上，ファイル転送時間を控除することが可能となり，高い稼働率を実現できる。

● 運用性向上への取組み

共同利用センターにおける運用ポリシーは，センターごとに大きく異なっている。このため，様々な運用ポリシーへ柔軟に対応できるジョブスケジューラが求められる。共同利用センターにおいて，カスタマイズが必要な運用ポリシーは，大きく以下の二つに分類できる。

- ジョブ実行時に適用される制限・制約事項などを定義するジョブごとに適用されるポリシー（使用可能なメモリ量やCPU数，実行可能な経過時間など）
- どのジョブを優先的に実行するのかといったジョブ運用全体の運用上のポリシー

「京」のジョブスケジューラでは，前者のジョブごとに適用される各種制約事項を「ジョブACL」と呼ぶ機能により定義可能とし，後者のシステム全体に影響する運用ポリシーを「スケジューリングポリシー」と呼ぶ機能により定義可能とした。今回の機能開発により簡素化できたものと考えている。

む　す　び

「京」における運用管理ソフトウェアは，特に超大規模システムであるために発生する課題へ対応するとともに，今後更にシステム規模が拡大しても，それに耐え得ることを目標として開発を行った。

本稿では，超大規模システムへの取組みを中心に解説したが，今後更にシステムの運用性を向上するための様々な取組みが必要と考えている。例えばシステム管理者によるシステムチューニングやキャパシティプランニングのためには，システムパラメータを変更した場合のシステムの振る舞いをシミュレーションするようなツールが必要である。このようなツールも組み合わせ活用することでシステムを安定的に効率良く運用していくことが可能と考える。

著者紹介

平井浩一（ひらい　こういち）
次世代テクニカルコンピューティング開発本部ソフトウェア開発統括部 所属
現在，スーパーコンピュータ向け運用管理ソフトウェアの開発に従事。

宇野篤也（うの　あつや）
独立行政法人理化学研究所次世代スーパーコンピュータ開発実施本部開発グループ 所属
現在，理研においてシステムソフトウェア関連のとりまとめに従事。

井口裕次（いぐち　ゆうじ）
次世代テクニカルコンピューティング開発本部ソフトウェア開発統括部 所属
現在，スーパーコンピュータ向け運用管理ソフトウェアの開発に従事。

黒川原佳（くろかわ　もとよし）
独立行政法人理化学研究所次世代スーパーコンピュータ開発実施本部 所属
現在，理研においてシステム全般の開発に従事。