

システム消費電力の上限を意識したポスト「京」向け ジョブ運用ソフトウェアの実現に向けて

秋元秀行^{†1} 三浦健一^{†1} 末安史親^{†1} 平井浩一^{†1} 住元真司^{†1}
宇野篤也^{†2} 山本啓二^{†2} 塚本俊之^{†2}

本論文ではシステム消費電力の上限を意識したポスト「京」向けのジョブ運用ソフトウェアの実現に向けて、システムのピーク電力制限とジョブ実行性能を両立した運用に必要な機能とその実現方法について述べる。ジョブ運用に必要な電力管理機能、およびソフトウェア要件を整理すると共に、2種類の電力管理方式を複数の観点から比較した。検討の結果から、ポスト「京」における電力管理方式として電力を考慮したジョブスケジューラ方式が有効であると考えている。電力を考慮したジョブスケジューラ方式は、ジョブ実行に必要な電力を予測し、加算することで全体の消費電力を制限する方式である。合計電力を制限する目的に適した、ジョブ毎の予測電力の指標についての検討と今後の方向性について併せて報告する。

1. はじめに

幅広いアプリケーションにおいて高い実行性能の実現を目指して開発されたスーパーコンピュータ「京」(以下、「京」)は2012年9月から共用が開始された[1][2]。共用開始以来、「生命科学・医療」、「エネルギー」、「防災・減災」、「次世代ものづくり」、「物質と宇宙」の戦略5分野を中心とした幅広い課題のアプリケーションの実行に有効利用され、質の高い研究成果を生み出している。「京」に限らず、スーパーコンピュータを用いたシミュレーション技術は学術利用から産業活用まで幅広い分野に広がり、より高い計算能力を持つスーパーコンピュータが求められている。

世界的にもスーパーコンピュータは先端的な研究成果を生み出す研究開発基盤、さらには、国家の経済成長を維持するために必要不可欠な共通技術として認識され、世界各地で高性能・大規模なスーパーコンピュータの開発プロジェクトが進められている。日本においても「京」の後継としてポスト「京」の設計がFLAGSHIP 2020 Projectとして2014年度から開始されている。

2020年以降に実現されるエクサスケール規模のスーパーコンピュータを実現するための共通課題として、電力性能比の改善やシステムの消費電力が着目され、様々な研究が進められている。特にシステム電力はスーパーコンピュータの規模を決定する重要な制約条件の一つとなっている。システム導入後の運用コストにおけるシステム電力費用の占める割合が増加していることが理由であり、システム電力をいかに効率よく活用するかが課題である。

システム電力を最大限に活用するための新たな設計思想として「Over Provisioning」や「電力制約適応型システム」が提案されている[3][4]。これらは、システム運用時の消費電力の制限・抑止など動的な管理を前提にしたシステム電力管理を指向している。これは現在運用中の「京」においても例外ではなく、様々な取り組みが実施されている。

本論文ではシステム消費電力の上限を意識したポスト「京」向けのジョブ運用ソフトウェアの実現に向けて、システムのピーク電力制限とジョブ実行性能を両立した運用に必要な機能とその実現方法について述べる。

2. 「京」における電力管理の取り組みと課題

2.1 「京」におけるシステム電力管理の必要性

「京」は2011年にLinpackベンチマークで実効性能10.51 PFLOPSを達成し、同年11月のスーパーコンピュータの性能ランキングTop 500で、世界で始めて10 PFLOPSを超えるシステムとして世界1位を獲得した[5]。ベンチマーク計測時の消費電力は12.66 MW、実行効率は93.17%と非常に高い値を記録している。

「京」の2012年9月の共用開始時において、Linpackベンチマークが計算ノードの最大消費電力に達すると想定し、商用電力として12 MWの契約を行い、運用を開始した。しかし、ジョブの最適化が進んだことや大規模ジョブ実行の頻度の増加などから、2013年度の運用で契約電力を超過する事象が複数回発生した[6][7]。その結果、2014年度は契約電力の見直しとして12.75 MWの契約を余儀なくされ、運用コストに大きな影響がでている。

2.2 電力管理の取り組み

前述の契約電力の引き上げの経験から、「京」においてもシステム消費電力の上限管理を開始した。「京」ではハードウェアによる電力管理・制限機構がないため、様々な検討の末、以下に示す施策による電力管理を実施している[6][7]。

- システム消費電力の常時監視と、超過時のジョブの緊急停止による削減処置(受動制限)
- 大規模ジョブに対する事前審査によりジョブ消費電力を推測し、実行可能ノード数や同時に実行するジョブ数の制限処置(能動制限)

本施策の導入後は、現在までのところ、再度の超過は発生していない。

2.3 電力管理の課題

「京」における電力管理の主体は現在のところ、商用電

^{†1} 富士通株式会社 FUJITSU LIMITED

^{†2} 理化学研究所計算科学研究機構

RIKEN Advance Institute for Computer Science

力の受電電力を契約電力以下に制限することを目的とした消費電力の上限値制限である。現在の電力管理は、ジョブの消費電力の推測を含め、手作業により実施しており、システム管理者の負担が大きい。このため、電力超過時のジョブの緊急停止等について、自動化に取り組んでいる。

3. ポスト「京」で想定されるジョブ運用機能と消費電力管理ソフトウェアの要件

本章では、ポスト「京」で想定されるジョブ運用において必要になると考えられる電力管理機能について述べ、その要件について述べる。

3.1 契約電力の超過管理機能

2章で述べたとおり、「京」においては電力管理として契約電力の超過を抑止したジョブ運用を継続して行っている。ポスト「京」においても同様にシステムの消費電力の超過を抑止したジョブ運用が不可欠であると想定しているが、その消費電力の上限値は、計算機本体の理論的な最大消費電力に対してより低い値となり、より厳しい電力超過管理が必要になることが予測される。

このためポスト「京」では、「京」のように受動的な電力管理を主体にする運用から、電力超過を事前に抑止する能動的な電力管理を主体にする運用へ移行を進める必要があると考えている。

さらに、ポスト「京」においてはシステムの消費電力の超過を抑止するのみではなく、与えられた電力を効率的に利用し、計算リソースを最大限に活用する仕組みの確立にも取り組む必要があると考えている。

3.2 システム消費電力の運用カレンダー連携管理機能

ポスト「京」の運用におけるシステム電力の上限値は、固定値だけではなく平日・休日の電力需給状況などに応じた定期的な変動や、ジョブの受付・待ち状況に応じた電力のスポット購入や自家発電の増加などの不定期な増減への対応が必要になると考えられる。このシステム電力の上限値の変動はその管理・制御方式によってはジョブ運用に影響を与える可能性がある。その一例を次に示す。

図1は未来の時刻 t_1 においてシステムの消費電力の上限値が縮小される場合について、ジョブとシステム電力の関係を模式的に示したものである。システム電力の上限値の管理として、現在値のみを管理する方式では、時刻 t_0 においてJob A, Job Bの合計電力は上限値を超過しないため、同時に実行が開始される。しかし、時刻 t_1 においてシステム電力の上限値が縮小されることに伴い、電力超過状態となる問題が生じる。

このように、ポスト「京」では、システム消費電力の上限値の計画的な変動によるジョブ運用影響を低減するために、上限値の変動計画を考慮した運用カレンダー連携機能の実現が必要である。

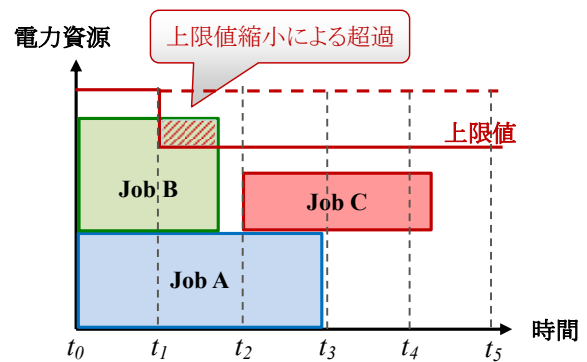


図1 システム電力の上限値縮小に伴う電力超過の模式図

3.3 高速なジョブ実行環境の安定的な提供機能

一般的に、利用者がスーパーコンピュータを使用する主目的は、高速な計算環境の利用であると考えられる。このため、スーパーコンピュータの運用においては、高速なジョブ実行環境の安定的な提供が必要不可欠である。また、システムの消費電力に制限がある場合においても、ジョブ実行性能に変動がないことが望ましいと言える。

3.4 ポスト「京」の消費電力管理ソフトウェアの要件

以上のことからポスト「京」の消費電力管理ソフトウェアの要件として以下の4つを導き出すことができる。

- システムソフトによる自動制御によりシステム電力を設定値以下に制限した運用ができること
- システム電力の制限下においても、その電力を効率的に利用し、計算リソースを最大限に活用できること
- システム電力の上限値の変動計画を設定し、それに従い運用中に適切に制御できること
- 利用者に対して高速・安定したジョブ実行環境を提供できること

4. 消費電力管理ソフトウェアの実現方式

前章で述べた要件からシステムの消費電力管理ソフトウェアにおける計算ノードの消費電力を管理する方式について検討を行う。検討に当たりポスト「京」世代に想定されるハードウェアによる電力管理・制御機構として、ノード単位のパワーキャップ機構を想定して検討を行う。本機構はノード単位の消費電力が制限値を超過した際に、ハードウェアによって自動的に性能を制限し、電力を制限値以下に保つ機構として想定する。

ポスト「京」のシステムの消費電力を制限する方法の検討として、前述のハードウェア機構を利用する方式である「1. 一律パワーキャップ方式」、およびソフトウェアのみによる方式である「2. 電力考慮スケジューラ方式」を比較する。具体的には3章に挙げた要件をそれぞれ、「電力制御の確実性」、「計算ノード、電力資源の利用効率」、「電力制限値の計画変動への追従性」、および「高速・安定したジョブ実行環境の提供」として4.1節以降で比較する。2つの

電力制限方式の概要は以下のとおりである。

1. 一律パワーキャップ方式
 計算ノード全体の上限電力を各計算ノードに均等に配分し、ノード単位のパワーキャップを一律に設定する
2. 電力考慮スケジューラ方式
 ジョブを実行する計算ノード数を制御し、計算ノード全体の消費電力を制限する

4.1 電力制御の確実性

1. 一律パワーキャップ方式
 計算ノード全体の上限電力 (P_{limit}) を以下の計算式により各計算ノードに均等に配分する方式である。

$$p_{cap} = P_{limit} / N_{total} \quad (1)$$

ここで、 p_{cap} は各計算ノードの配分電力、 N_{total} はシステム内の計算ノード数である。ノード単位のパワーキャップ値として p_{cap} を設定することにより、ハードウェアによって各ノードの消費電力が設定値以下に制限されることから、計算ノード全体の消費電力の上限值制御の確実性が期待できる。

2. 電力考慮スケジューラ方式
 本方式は計算ノードに対する電力制限や性能制限を一切行わず、ジョブ実行において同時に使用する計算ノード数 (N_{job}) を制御することにより、計算ノード全体の消費電力を制限するものである。計算ノード全体の消費電力を P_{limit} に制限する場合、 N_{job} は次式により算出できる。

$$N_{job} = (P_{limit} - p_{idle} \cdot N_{total}) / (p_{max} - p_{idle}) \quad (2)$$

ここで p_{max} は単一計算ノードの仕様上の最大消費電力、 p_{idle} はジョブが実行されていない計算ノードの消費電力である。これにより、システム電力の上限值制御の確実性が期待できる。

4.2 計算ノード、電力資源の利用効率

1. 一律パワーキャップ方式
 本方式は全ての計算ノードを同時に利用できるため、計算ノードの使用率という点では効率が良いと言える。しかしながら、ジョブの実行性能が電力制約のため制限されていることから、それぞれのジョブが長時間にわたり計算ノードを専有する点が課題である。

一方、電力の利用効率の観点では、本方式は全計算ノードに対して一律に電力の使用制限を設定するため、全ての計算ノードに対して高負荷ジョブが割り当たっている場合、利用効率は高いと言える。しかし、それ以外のジョブが混在する状況においては、利用効率が低くなる場合がありえる。

2. 電力考慮スケジューラ方式
 本方式はジョブ実行に利用可能な計算ノード数が制限されるため、計算ノードの使用率という観点では効率が良いとは言えない。一方、各ジョブの実行性能に

制限がないため、短時間で計算を終了し後続のジョブへ計算ノードを明け渡すことが可能である。

次に、電力の利用効率の観点では利用可能な計算ノード全てに高負荷ジョブが割り当たっている状態に限り高いと言える。しかし、それ以外のジョブが混在する状況においては、電力の利用効率が低くなる場合がありえる。

4.3 電力制限値の計画変動への追従性

1. 一律パワーキャップ方式
 本方式ではジョブの実行を維持したまま、システム電力の上限值を動的に変更可能なため、計画的な変動にも容易に追従可能であると言える。

2. 電力考慮スケジューラ方式
 本方式ではジョブ実行中にシステム電力の上限值を動的に変更する場合、その値によっては一部のジョブを停止するなどの処置が必要である。一方、制限値の縮小までに時間的な猶予がある場合には事前の考慮により影響を最小限にとどめることができる。

4.4 高速・安定したジョブ実行環境の提供

1. 一律パワーキャップ方式
 本方式は個別の計算ノードにおいて、その消費電力に応じてジョブ実行性能が制限される。従って、利用者に対し、高速なジョブ実行環境を安定して提供することはできない。

2. 電力考慮スケジューラ方式
 本方式は電力制限の有無によらず、計算ノードに対する性能制限を一切行わない。このため、個別ジョブの実行性能は常に高速かつ一定に保たれる。従って、利用者へ高速なジョブ実行環境を安定して提供することが可能である。

4.5 両方式の比較と電力管理方式の選定

表 1 は 2 つの電力制限方式を、前述の 4 つの観点で優劣をまとめたものである。表 1 より総合的な比較として「電力考慮スケジューラ方式」が勝っていると言える。従って、我々はポスト「京」における電力管理方式として「電力考慮スケジューラ方式」を採用する方向で検討を進める。

表 1 各方式の比較結果

	一律 パワーキャ ップ方式	電力考慮ス ケジューラ 方式
電力制御の確実性	○	○
計算ノード、電力資源の利用効率	△	△
電力制限値の計画変動への追従性	○	△
高速・安定したジョブ実行環境の提供	×	○

5. ジョブスケジューラによる電力管理方式

4章で述べた電力考慮スケジューラ方式としては、ジョブが計算ノードの仕様上の最大消費電力を常に消費すると仮定し、ジョブ実行に利用可能な計算ノード数を算出するもっとも単純な方法を想定した。しかし、一般的にジョブ毎の消費電力は、ジョブの特性によって異なる。従って、計算ノードの最大消費電力の代わりに実際のジョブの消費電力を予測して用いることができれば、電力資源の利用効率を向上することができる。

そのためには、精度の高いジョブの消費電力予測が必要となる。山本らは、ジョブの平均電力を、過去のジョブの実行実績を利用し、実行前に精度良く予測できることを示している[8]。

ポスト「京」では、ジョブ電力の予測としてこの手法を用い、図2に示すようなフローによるシステム電力管理方式を検討している。本章では図2の「電力を考慮したジョブスケジューラ」において、ジョブの予測電力としてどのような指標を扱うべきかについて検討する。

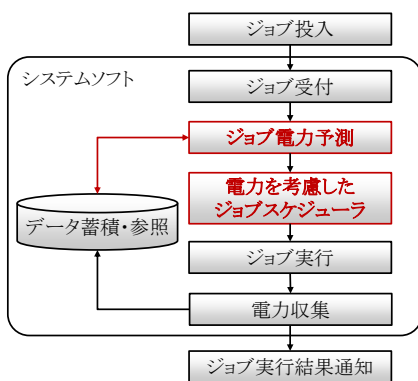


図2 システム電力管理方式の全体フロー

5.1 電力を考慮したジョブスケジューラ概要

図3にジョブスケジューラによるジョブへの資源割り当てを模式的に示した。ここで(a)は計算ノード資源の割り当てマップ、(b)は電力資源の割り当てである。

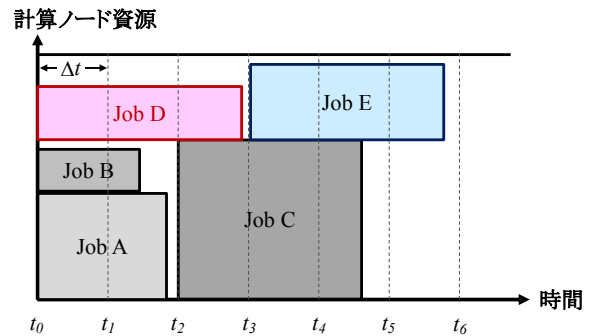
ジョブスケジューラは、計算ノード資源と電力資源を一定の時間間隔 (Δt) に区切り管理する。ジョブには Tofu インタコネクタ[9]で接続された計算ノード資源を連続した時間割り当てる。ただし、図3(a)では簡単のために計算ノード資源空間を1次元として縦軸に示した。

一方、電力資源は資源位置が固定されないため、図3(b)のように単純な合計値として管理する。なお、ジョブ実行時間中の予測電力は、開始から終了まで一定であるものとして図示した。 Δt に区切った消費電力の合計は同時に実行されるジョブ電力の合計として式(3)によって算出する。

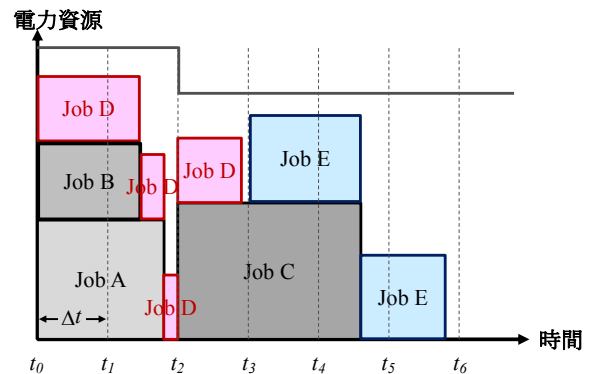
$$P_{tmap} = \sum (p_{job} \cdot n_{job}) + p_{idle} \cdot n_{idle} \quad (3)$$

ここで、 p_{job} はジョブ毎のノード当たりの平均予測電力、

n_{job} はジョブ毎の使用ノード数である。また p_{idle} , n_{idle} はそれぞれジョブが割り当たっていない計算ノードのノード当たりの平均電力、ノード数である (なお、図3(b)には第二項のアイドル電力は図示していない)。ジョブ毎の予測電力として、例えば Δt 間隔毎の平均電力を予測・使用することによって、ジョブ毎の時間変動を直接ジョブスケジューラとして扱うことも検討している。



(a) 計算ノード資源マップ



(b) 電力資源マップ

図3 計算ノード資源へのジョブ割り当て

このようにして求めた各時間の合計電力を、全ての時間帯において電力消費制限値以下にすることで、計算ノードの消費電力を制限値以下に抑止する。ここで、各時間帯の電力制限値は一定である必要はなく、図3(b)のように時刻 t_2 から縮小されるようなケースも想定される。このように計画的な電力制限値の変動を考慮することによって、3.2節に述べたシステム消費電力の運用カレンダー連携管理機能を実現する。

実際のジョブ割り当てにおいては、図3(a)と(b)の資源を同時に考慮してジョブ割り当て位置を決定する。なお、先行して投入されたジョブの実行開始時刻に遅延が生じないことを条件に、実行開始時刻を追い越した割り当て (バックフィル) を行うことにより、ジョブ実行のスループットを向上させ、計算ノードと電力資源の効率的な利用が実現できることが期待される。

5.2 ジョブスケジューラにおける電力の取り扱い

本節では、電力を考慮したジョブスケジューラとして、

単純化のためにジョブの予測電力を開始から終了まで一定値として扱う。その上でジョブ毎の予測電力として、以下の値を用いて算出される計算ノードの合計消費電力について考察する。

1. 最大電力の使用
2. 平均電力の使用
3. 平均電力と標準偏差の使用

さらに、電力考慮ジョブスケジューラに適したジョブ毎の予測電力について検討する。図4はそれぞれの予測電力を用いて算出される合計電力を模式的に示した。

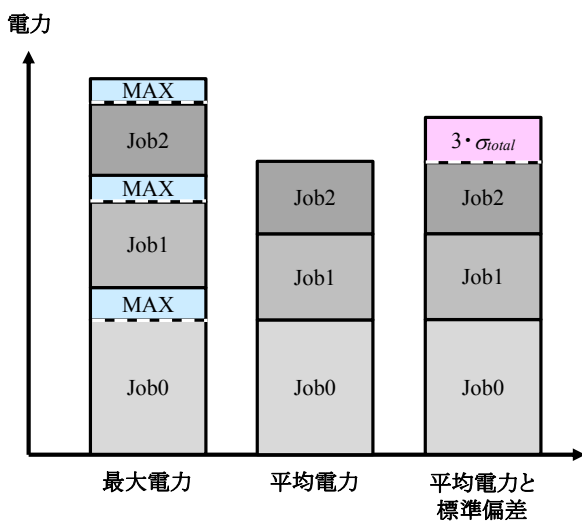


図4 ジョブ毎の予測電力として使用する値と算出される計算ノードの合計電力の関係

5.2.1 最大電力使用方式

最大電力を使用する方式は、ジョブ毎の予測電力として、ジョブの開始から終了までの最大電力を使用するものである。これは複数ジョブの同時実行時の最悪ケースにおいても計算ノードの合計電力が上限値以下になることを保証するものである。つまり、同時実行されるジョブの全てが同じタイミングで最大電力を消費した場合を想定している。

計算ノードの消費電力が、上限値を一瞬たりとも超過してはならない場合には、本方式による電力管理が確実である。一方、電力の有効活用という点では未使用電力が発生する場合があります問題である。

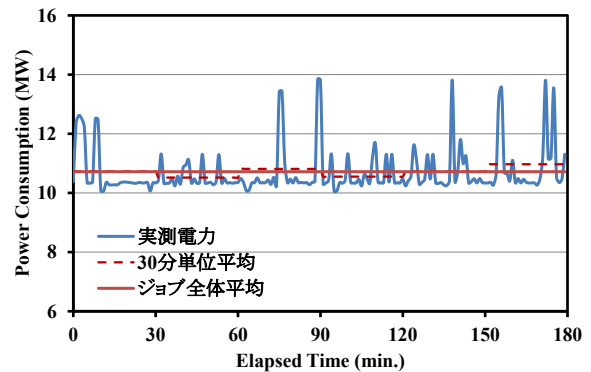
5.2.2 平均電力使用方式

平均電力を使用する方式は、ジョブ毎の予測電力として、時間に対する電力の変動をジョブ全体で平均して使用するものである。ジョブ毎の消費電力は、ジョブを構成する各計算ノードの消費電力の平均値の加算となり、システム全体の消費電力は、各ジョブのジョブ全体での平均電力の単純加算となる。

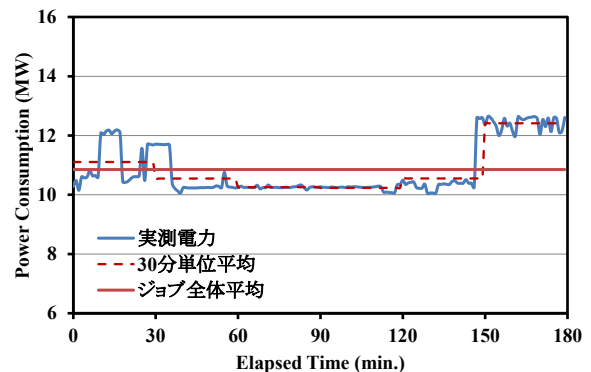
実際の計算ノードの合計消費電力は、平均値を中心に時間と共に上下に変動することになる。このため電力制限と

して一瞬の超過も許されない場合には適さない。

一方「京」のシステム電力の上限値管理では、消費電力の制限として毎時0~30分、または30~60分の30分間の平均電力が契約電力を越えないことが必要とされる。このような条件においては、全てのジョブの電力変動が、例えば図5(a)に示すように30分に比べ十分に短いものであれば、計算ノード全体の合計電力はジョブ毎の平均値の単純加算で問題とならない。一方、図5(b)に示すようにジョブの途中で消費電力の傾向が変わる場合には、計算ノード電力の上限値管理として不十分な場合がある。



(a) 電力変動周期が短いジョブの例



(b) 電力消費が途中で変わるジョブの例

図5 「京」におけるジョブの消費電力の時間変動例

5.2.3 平均電力と標準偏差使用方式

平均電力に加え標準偏差を使用する方式は、ジョブ毎の予測電力として、5.2.2で述べた平均電力に加え、ジョブ毎の標準偏差を利用するものである。

ここで、ジョブの消費電力の時間に対する変動が正規分布していると仮定すると、計算ノード全体の消費電力の標準偏差 (σ_{total}) は以下の式で算出することができる。

$$\sigma_{total} = \sqrt{\sum \sigma_{job}^2} \quad (4)$$

ここで、 σ_{job} は各ジョブの消費電力の標準偏差である。単純加算により求めた平均値および σ_{total} から確率密度分布に従い、実際の電力超過確率を制御することができる。

例えば、合計電力のマージンとして $3\sigma_{total}$ や $4\sigma_{total}$ を加算することにより、電力超過のリスクをそれぞれ 0.135%、

0.003%以下に抑える事ができる。実際の電力超過に対する許容度は、設備機器の仕様やジョブ電力の時間に対する分布傾向により異なるが、瞬間電力と30分の平均電力を制限する両方の目的において、ジョブの消費電力の時間に対する変動が正規分布しているという条件を満たすという前提で本方式は有効であると考えている。

5.3 ジョブスケジューラにおける電力管理方式の方向性

「京」のジョブ実行時間制限は基本的に6~24時間であり、ポスト「京」でも同様な運用が想定される。したがって、5.2節で述べた3つの方式を運用に適用した場合、「最大電力使用方式」では、未使用電力の有効活用が課題となる。一方、「平均電力使用方式」と「平均電力と標準偏差使用方式」については、上限値管理という点で課題が残る。

これに対し、ジョブの途中で消費電力傾向が変化する場合や大きな変動がある場合には、ジョブ全体の平均予測電力ではなく、例えば30分毎の平均値をジョブスケジューラによって直接管理する方式を採用すれば、より精度の高い上限値管理が可能になると期待できる。

平均電力と標準偏差考慮による高精度化の効果については、単位時間間隔の決定手法や、ジョブの消費電力の時間に対する変動が正規分布しているという条件が実際に満たされているのかを見極める必要がある。

さらには、大きな電力変動を含むジョブがどの程度の割合で存在しうるのか、ジョブ実行時の時間変動を含むジョブ電力の予測方法を含めた調査を継続し、時間変動を直接考慮したジョブスケジューラの実現可能性と効果について確認する必要があると考えている。

6. おわりに

本論文では、システム消費電力の上限を意識したポスト「京」向けのジョブ運用ソフトウェアの実現に向けて、システムのピーク電力制限とジョブ実行性能を両立した運用に必要な機能とその実現方法について述べた。ジョブ運用に必要な電力管理機能、およびソフトウェア要件を整理すると共に、2種類の電力制限方式を複数観点の比較、検討結果を示した。

検討の結果、総合的な評価として、電力考慮スケジューラ方式が勝っているといえる。従って、ポスト「京」における電力管理方式として、電力を考慮したジョブスケジューラ方式が有効であると考えている。

電力を考慮したジョブスケジューラ方式は、ジョブ実行に必要な電力を予測し、加算することで全体の消費電力を制限する方式である。この際にジョブ毎の予測電力として、開始から終了までを一定値として扱う方式と時間変動を直接扱う方式を検討した。前者においては、加算に際し、ジョブ毎の電力の時間に対する変動が正規分布であることを仮定し、平均に加え標準偏差を予測し、合計電力を算出、制限する方式を提案した。

今後の予定としては、個別ジョブの時間変動に対する電力分布形状の確認や大きな電力変動を含むジョブの割合などの調査を継続し、時間変動を直接考慮したジョブスケジューラの実現可能性と効果を検証し、具体化していく予定である。

謝辞 本論文の一部は、文部科学省「特定先端大型研究施設運営費等補助金（次世代超高速電子計算機システムの開発・整備等）」で実施された内容に基づくものである。

参考文献

- 1) 山本啓二, 宇野篤也, 塚本俊之, 菅田勝文, 庄司文由: スーパーコンピュータ「京」の運用状況, 情報処理, Vol.55, No.8, pp.786-793 (2014).
- 2) Keiji Yamamoto, Atsuya Uno, Hitoshi Murai, Toshiyuki Tsukamoto, Fumiyoshi Shoji, Shuji, Matsui, Ryuichi Sekizawa, Fumichika Sueyasu, Hiroshi Uchiyama, Mitsuo Okamoto, Nobuo Ohgushi, Katsutoshi Takashina, Daisuke Wakabayashi, Yuki Taguchi, Mitsuo Yokokawa: The K computer Operations: Experiences and Statistics, Proceedings of International Conference on Computational Science (ISSC), Vol.29, pp. 576-585 (2014).
- 3) 石川ほか: HPCI 技術ロードマップ白書 (2013).
- 4) 近藤正章: エクサスケールで顕在化する Power Wall 問題 ~現状と今後の打開策~, サイエントフィック・システム研究会 (科学技術計算分科会) (2013).
- 5) Super Computer TOP500, <http://www.top500.org/>
- 6) 井上文雄, 宇野篤也, 塚本俊之, 松下聡, 末安史親, 池田直樹, 肥田元, 庄司文由: 消費電力の上限を考慮した「京」の運用, 情報処理学会研究報告, Vol.2014-HPC-146, No.4 (2014).
- 7) 宇野篤也, 肥田元, 井上文雄, 池田直樹, 塚本俊之, 末安史親, 松下聡, 庄司文由: 消費電力を考慮した「京」の運用方法の検討, ハイパフォーマンスコンピューティングと計算科学シンポジウム, HPCS2015, pp. 167-175 (2015).
- 8) 山本啓二, 末安史親, 宇野篤也, 塚本俊之, 肥田元, 池田直樹, 庄司文由: 過去の実行実績を利用したジョブの消費電力予測, 情報処理学会研究報告, Vol.2015-HPC-151, No.2 (2015).
- 9) Y. Ajima, S. Sumimoto, T. Shimizu: Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers, IEEE Computer, Vol. 42. no. 11, pp. 36-40 (2009).