

消費電力を考慮した「京」の運用方法の検討

宇野 篤也^{1,a)} 肥田 元² 井上 文雄¹ 池田 直樹² 塚本 俊之¹ 末安 史親³ 松下 聡¹ 庄司 文由¹

概要: 近年, 計算機システムの大規模化等から, システムの消費電力を考慮した運用を行う必要性が増してきている。「京」でも消費電力は運用上の大きな課題で, 全計算ノードを使うようなジョブの実行において契約電力を超過する事例がこれまでに何度か発生している。頻繁な契約電力の超過は電力契約の見直し等につながり, 運用への影響は無視できない。これを回避するため, ジョブを消費電力の観点で事前に調査し, 電力超過を防ぐ運用体制を構築した。また, 消費電力が上限を超えた場合にそなえて, ジョブ毎の消費電力をもとに適切にジョブを停止する手法を検討した。「京」では計算ノード毎に電力計が設置されていないため, 本手法では温度センサの情報からジョブ毎の消費電力の推定を行う。「京」上で実行されたジョブで検証を行い, ジョブ毎の消費電力をもとに停止するジョブを適切に選択できることを確認した。

Operation of the K computer Focusing on System Power Consumption

ATSUYA UNO^{1,a)} HAJIME HIDA² FUMIO INOUE¹ NAOKI IKEDA² TOSHIYUKI TSUKAMOTO¹
FUMICHIKA SUEYASU³ SATOSHI MATSUSHITA¹ FUMIYOSHI SHOJI¹

Abstract: Recently, High-Performance Computing system has become more and more large, and the power consumption has become one of the important problems on the system operation. We also have the same problem on the operation of the K computer. To prevent the power consumption from exceeding the limit, we have made the preliminary review that estimates the power consumption of each job, and have controlled the system power consumption. In case of exceeding the power consumption limit, we have investigated the emergency job stopping method based on the estimated power consumption of each job. In this process, we estimate the power consumption of the job using thermal sensors in the compute racks. This estimation enables us to select the appropriate jobs to be stopped.

1. はじめに

スーパーコンピュータ「京」は, 理化学研究所と富士通株式会社が共同開発した汎用並列スーパーコンピュータである。運用は理化学研究所 計算科学研究機構 (AICS) が行っており, 2012年9月から共用を開始している。「京」は低消費電力 CPU の採用など消費電力を抑えるように設

計されているが, その消費電力は無負荷時で約 10MW, 高負荷時には 14MW を超える場合もある。運用コストに占める電力料金の割合は非常に大きく, システム全体の消費電力を考慮した運用が求められている。

一般的に近年のアーキテクチャは CPU やメモリアクセスの負荷に応じて電力が大きく変動する傾向にあり, システムの消費電力は実行されるジョブの効率等に依存することが知られている。特に「京」は規模が大きいためジョブ実行の消費電力の変動が非常に大きい。共用開始当初は, 大規模ベンチマーク等の特殊なケースを除き, 消費電力が問題になることはなかったが, 1年を経過した頃から, 消費電力が大きく変動し契約電力の上限を超える事象が何

¹ 国立研究開発法人理化学研究所 計算科学研究機構
RIKEN Advanced Institute for Computational Science
² 株式会社富士通ソーシャルサイエンスラボラトリ
FUJITSU SOCIAL SCIENCE LABORATORY LIMITED
³ 富士通株式会社
FUJITSU LIMITED
a) uno@riken.jp

度が発生した．頻繁な契約電力の超過は電力契約の見直し等へつながり，運用への影響は非常に大きい．システム全体の消費電力を適切にコントロールするためには，実行される個々のジョブの特性を事前に把握し，システム全体の消費電力を予測することが重要となる．

電力超過が発生したケースのほとんどは後述する大規模ジョブ実行期間に発生しているため，消費電力が契約電力を超過しないようにコントロールするための対策として，まず，大規模ジョブ実行期間に実行される大規模ジョブについて消費電力の観点で事前に審査する体制（事前審査制度）を構築した．しかし，これだけでは契約電力の超過を完全に防止することはできないので，消費電力が上限値を超過した場合に速やかにジョブ毎の推定消費電力に基づいて適切なジョブを停止する手法について検討を行った．「京」では計算ノード毎に電力計は取り付けられていないため，各計算ラック（計算ノード 96 台）に取り付けられている温度センサの情報とシステム全体の消費電力の情報を組み合わせて個々のジョブの消費電力の推定を行った．

本稿では，事前審査制度の概要とジョブ毎の消費電力の推定方法とその結果について述べる．

2. 「京」の概要

「京」は，82,944 台の計算ノードと 1.27PiB のメモリ，11PB のローカルファイルシステム (LFS) と 30PB のグローバルファイルシステム (GFS) 等から構成される．図 1 に京のシステム構成概要を示す [1]．

「京」の運用に必要な電力は，商用電力（関西電力）と自家発電により供給されている．図 2 に AICS の電源設備を示す．自家発電設備として，ガスタービンによるコジェネレーションシステム (CGS) を 2 台備えている*1．CGS 1 台の定格出力は約 5MW で，通常運用時は 1 台ずつ交互に運転を行い，不足電力分を商用電力から受電している．

「京」の運用形態は，36,864 ノード以下の規模のジョブの実行が可能な通常運用と，36,865 ノード以上の規模のジョ

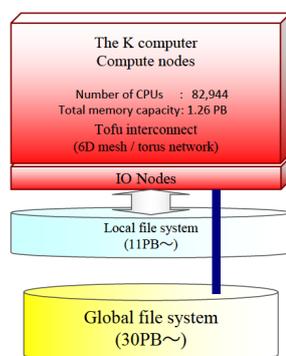


図 1 「京」のシステム構成

*1 CGS の電力は，停電時等に GFS のデータを保護するためにも使われる

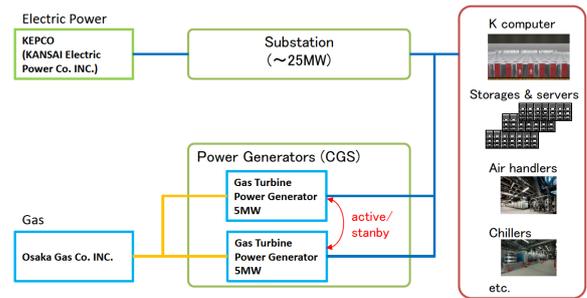


図 2 AICS の電源設備

表 1 AICS 全体の想定消費電力（共用開始時）

内訳	想定消費電力
「京」本体 (含 LFS)	10MW
ジョブ実行時の増分	~ 4MW
その他施設 (含 GFS)	~ 3MW

ブを実行できる大規模ジョブ実行運用の 2 つに大きく分けることができる．毎月第二火曜から 3 日間を大規模ジョブ実行期間として設定している [2][3]．

3. 電力消費

「京」の消費電力は無負荷時で約 10MW，高負荷時で 14MW を超える．表 1 に共用開始時に想定した AICS 全体の消費電力を示す．ジョブ実行時の増分は，実行効率が最も高いと想定した LINPACK の消費電力を参考に算出している．共用開始時には，供給電力の上限を 12MW とし電力会社と契約した．この上限を超過した場合*2，電力会社に対して違約金の支払いが発生する．この電力超過が頻繁に発生すると契約電力自体の見直しとなり，運用経費の増大という問題が発生することになる．実際，2013 年度には電力超過が 3 回発生したため，2014 年度の契約電力は 0.75MW 増の 12.75MW となった．そのため，運用側にとって電力超過を防ぐことは非常に重要な課題である [4]．

近年，データセンターや HPC システムにおける電力問題は重要な研究テーマとなっている．システム全体の消費電力に一定の制約を設定した条件下で，電力資源を最大限に利用するようなスケジューリング手法や，ジョブのスループットが最大化されるようなスケジューリング手法等が提案されている [5][6][7]．「京」の場合，そのシステム構成から，ジョブの使用ノード数の動的変更による消費電力の最適化や CPU の周波数変更による消費電力と実行時間の最適化，CPU やメモリへの電力配分の動的変更による消費電力の最適化といった手段は用いることができない．また，「京」は既に運用を開始しているため，現在の運用を大きく変えるような，特にユーザの利用が大きく制限されるような手段を導入することは難しい．そこで，現在の運用を大きく変えずに電力超過に対処する手段について検討

*2 毎時ごとの 0~30 分または，30~60 分の 30 分間における平均使用電力が契約電力を超えた場合

を行った。

3.1 電力超過対策

契約電力の超過を防ぐ方法として、(1) 自家発電量を増やす、(2) システムの一部を停止する、(3) 超過しない範囲でジョブを実行する、といった対策を検討した。

(1) 自家発電量を増やす方法

自家発電量を増やすことで、利用可能な電力上限を引き上げることができる。CGSは休止状態から発電可能になるまで2時間程度必要なため、電力超過が発生してからへの対応では間に合わない。そのため、常に2台のCGSを稼働させることが前提となる。増えた発電分を商用電力からの受電量から減らすことになる。施設設計当時は1MW当たりの単価はCGSで発電する方が安価だったが、最近のガス単価および電力単価ではその状況が逆転しており、商用電力から受電したほうがコスト的に有利である。さらに、電力単価よりもガス単価の上昇率が高い状況が続いており、当面は現在の状況が続くものと思われる。また常にCGSを2台稼働させると、CGSの故障やメンテナンス時にノードを停止させるなどの対応が必要となる。以上の理由から、自家発電量を増やす方法は採用しないこととした。

(2) システムの一部を停止する方法

システムの一部を停止することでシステム全体の消費電力を削減し、電力超過の可能性を減らすことができる。しかし、この方法では電力超過を完全に防止することはできない上、提供できる資源量が減少し事前に各課題に割り当てた計算資源を提供できなくなる。そのため、「京」では採用できないと判断した。

(3) 電力を超過しない範囲でジョブを実行する方法

ジョブ毎の消費電力を考慮してジョブを実行することができれば電力超過を防ぐことができる。この場合、実行するジョブ毎の消費電力を事前に調査および予測し、電力超過を起こさない範囲でジョブを実行することになる。しかし、全てのジョブを審査することは困難なので審査対象を絞る必要がある。電力超過を引き起こす可能性のあるジョブは、全ノードを使用するような大規模ジョブと想定されるので、大規模ジョブ実行期間で実行されるジョブに限定することで数の問題は解決できる。

以上の検討結果から、(3) 電力を超過しない範囲でジョブを実行する方法を採用することにした。しかしながら、この方法では事前の調査や予測による消費電力の制度に限界があるため、電力超過を完全に防止することは難しい。そのため、電力超過時には速やかにジョブを停止するなどの対応を別途検討する必要がある。

4. ジョブの事前審査制度

電力超過が発生させる可能性のあるジョブを事前に除外

するため、ジョブの事前審査制度を導入した。この事前審査では実行予定のジョブの種類毎の消費電力を推測し、電力超過を引き起こさない規模での実行を許可する。図3に事前審査制度のフローを示す。

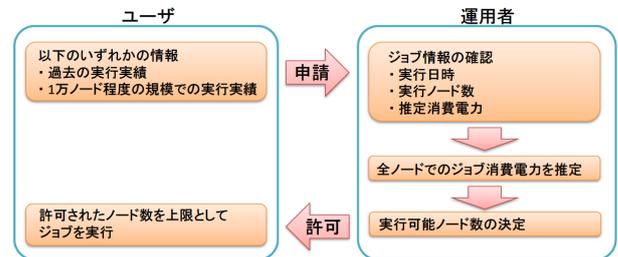


図3 事前審査制度のフロー

ジョブ実行時の消費電力推定は、大規模実行期間に実行予定のジョブと同じ種類のジョブの1万ノード程度の規模の測定値、もしくは過去の大規模実行での実績値をもとに行う。具体的な手順は以下のとおりである。まず、申請のあったジョブの実行履歴を確認し、その時のシステム全体の電力変動からそのジョブの消費電力を推定する。そして、式1から許容電力における実行許可ノード数を求める。ただし、実行許可ノード数のジョブであっても、同時に複数実行された場合には電力超過が発生する可能性がある。例えば、最大4万ノードまで許可となった場合、同時に2つ実行されると最大で8MWとなり許容電力を超えてしまう。これを防ぐため、ジョブの同時実行数も制限し、システム全体の電力が許容電力に収まるように制御する。

$$P_{node} = \frac{P_{job}}{N_{job}} \quad N = \frac{P_{max}}{P_{node}} \quad (1)$$

ここでの各パラメータは以下の通りである。 P_{node} :1ノードあたりの消費電力、 P_{job} :ジョブの消費電力、 N_{job} :計算ノード数、 N :実行許可ノード数、 P_{max} :許容電力(4MW)。

表2に2014年8月の大規模実行期間における事前審査結果を示す。表中のジョブの消費電力は、ジョブ実行中のシステム全体の電力変動から求めた最大変動値である。表2からわかるように、測定値と推測値が大きく異なる事例がいくつか発生している。特に差異が大きいジョブについて調査を行ったところ、審査対象のジョブと大規模実行

表2 事前審査結果(2014年8月分)

計算ノード数	ジョブの消費電力 (MW)		
	測定値	推測値	差異
37,544	0.44	1.18	-0.74
37,544	0.82	1.18	-0.36
65,536	1.14	0.79	0.35
80,000	0.96	0.96	0.00
80,199	3.77	0.44	3.33
82,944	1.85	2.16	-0.31
82,944	0.42	1.60	-1.18
82,944	3.32	1.00	2.32

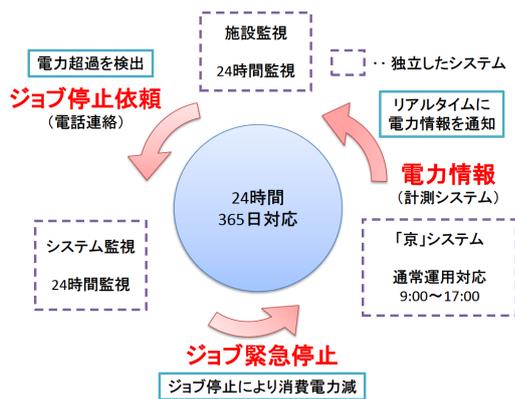


図 4 ジョブの緊急停止フローの概要

時のジョブにおいて、メモリの最大使用量に大きな差が見られたり、実経過時間の比に対して出力ファイル量に差がありすぎるなど、審査時と大規模実行時のジョブの特性が異なっている可能性が高いことがわかっている。

5. ジョブの緊急停止

電力超過が発生もしくは発生が予測された場合、実行中のジョブを停止して電力超過を防ぐ必要がある。図 4 にジョブの緊急停止フローの概要を示す。ジョブの緊急停止は、「京」システム、施設監視、システム監視の独立した 3 つのシステムを連携して行っている。リアルタイムに計測される消費電力を施設監視担当が常時監視し、電力が超過もしくはその可能性が高まった場合、システム監視担当へジョブの停止を電話で依頼する。依頼を受けたシステム監視担当は、電力超過が収まるまで実行中のジョブを順次停止する。ここで停止したジョブは電力超過が解消された後、電力超過が再発しないよう注意しながら順次再実行する。なお、これらのジョブは他のジョブより優先して再実行される。

大規模ジョブ実行期間内では、ほぼ 1 ジョブ単位でジョブが実行されるため、電力超過が発生した場合には速やかに該当ジョブを停止することができる。一方、通常運用期間では大小様々なジョブが同時に多数実行されているため、電力超過が発生した場合に超過の原因となったジョブを特定することは難しく、そのままでは手当たり次第にジョブを停止するしかない。

そこで、通常運用時においても、電力超過が発生した場合に適切にジョブを停止できるよう、ジョブ単位での消費電力の推定方法を検討した [8]。

5.1 ジョブ単位の消費電力の推定

ジョブ単位での消費電力の推定方法として、以下の方法を検討した。

- (1) ジョブが使用するノード数を用いた推定
- (2) ラックに取り付けられた温度センサを用いた推定

ジョブの規模が大きいほど実行途中の停止で無駄になる計算資源量は大きくなる。そのため、電力超過が発生した場合には、超過電力分だけ電力を減らしつつ、ジョブ停止によって失われる計算資源量を最小にするという観点から停止するジョブを選択する必要がある。失われる計算資源量として、消費電力をベースに計算する方法と、計算時間をベースに計算する方法の 2 種類が考えられる。「京」の場合、各課題はノード経過時間積で計算資源が配分されているので、失われるノード経過時間積が最小となるようにジョブを停止することになる。具体的には、電力超過時点で実行中の全てのジョブの超過時の消費電力とそれまでの実行経過時間を計算し、失われるノード経過時間積と停止するジョブ数が最小になるようにジョブを選択する。

(1) 使用ノード数による消費電力の推定

「京」で実行されている個々のジョブが使用しているノード数とシステム全体の消費電力から、ノード単位の平均消費電力を求めることができる。この場合、ジョブ単位の推定消費電力はノード数に単純に比例するので、電力超過時には削減すべき電力量をもとにジョブを順次停止すればよい。しかし、実際にはジョブ毎の消費電力は異なっているため、ジョブを停止しても予測した電力を削減できるとは限らない。また、規模の大きなジョブが複数同時に実行されているような場合には、どのジョブを停止すればよいか判断することは難しい。そのため、単純にノード数から消費電力を推定する方法は誤差が大きく効率のよい方法とは言えない。

(2) 温度センサ情報を利用した消費電力の推定

「京」の場合、ジョブの実行時の消費電力の大部分は、CPU とメモリ、Tofu インターコネクットのコントローラである ICC によって消費される。「京」の計算ラックには電力計は搭載されていないが、いくつかの温度センサは搭載されている。これらの温度センサは、各部品に異常が発生していないか監視するためのものだが、これらの情報を利用してジョブ実行時の消費電力を推定できないか検討した。

搭載されている温度センサのうち、ラック吸気温度、System Board(SB) 排気温度、水冷入力温度、CPU 温度の情報を利用して、CPU とメモリの温度変化を測定することにした。これらの温度データは、現状では 10 分毎に取得することができる。ここでは、ジョブ実行時の CPU 温度変化と SB 排気温度変化を以下のように定義した。

- CPU 温度変化 = CPU 温度 - 水冷入力温度
- SB 排気温度変化 = SB 排気温度 - ラック吸気温度

図 5 に 2014 年 4 月から 11 月までの平均 CPU 温度変化と平均 SB 排気温度変化、システム全体の消費電力変化のグラフを示す。図 5 から、システム全体の消費電力と各温度変化の変動が一致していることがわかる。このことから、各温度変化からジョブ単位の消費電力の推定は可能と判断した。

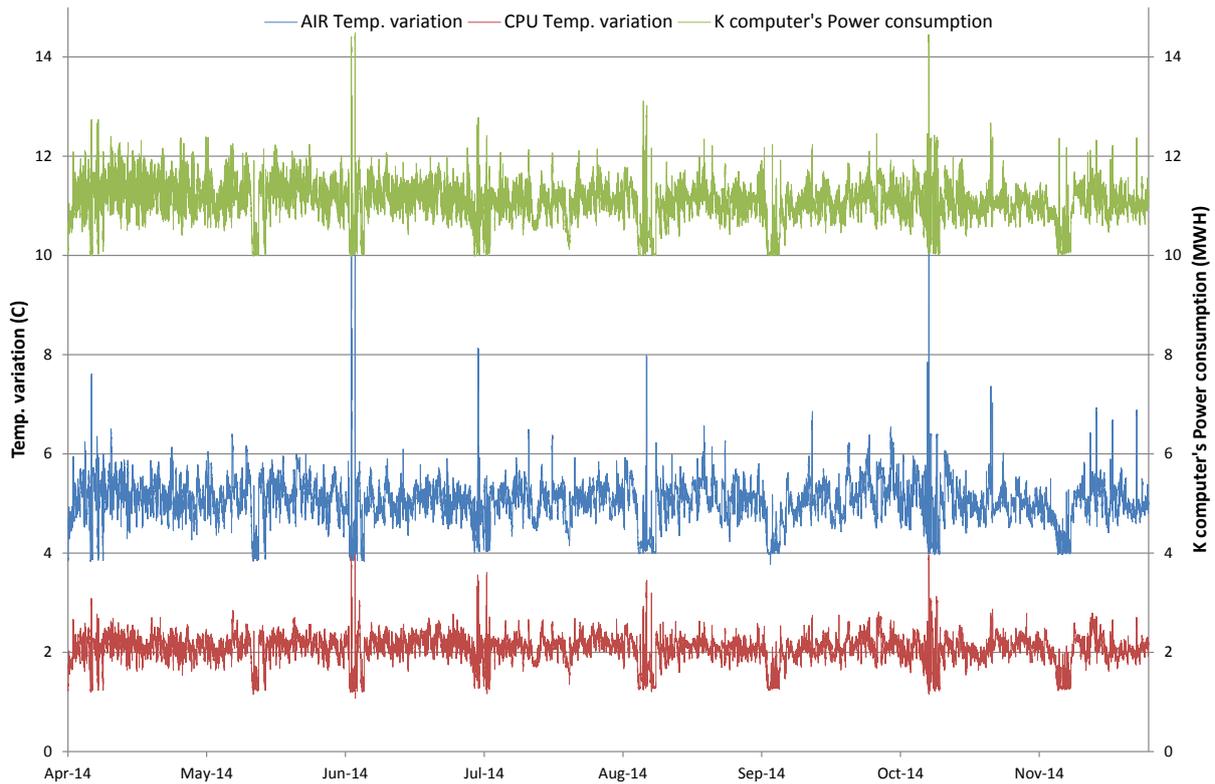


図 5 平均 CPU 温度変化，平均 SB 排気温度変化とシステム全体の消費電力変化の関係

以上の検討結果から，(2) 温度センサ情報を利用した消費電力の推定を行うことにした．

5.2 温度変化と消費電力

温度センサの情報からジョブの消費電力を推定するにあたり，「京」の一部の計算ラックに搭載されている電力計を使用して，CPU とメモリ，ICC の各温度変化と消費電力の関係について調査を行った．ファイル I/O 時の消費電力の変動についても調査を行ったが，計算ノードおよびディスクラックの消費電力にはほとんど変化がみられなかった．

5.2.1 CPU

CPU 温度変化と消費電力の関係について調査した．

図 6 に CPU 部分の冷却機構を示す．CPU で生じた熱は冷却水により冷やされ，冷やしきれなかった熱が CPU 温度の変化量として測定される．そのため，CPU 温度と冷却水温度の差が CPU による発熱を正確に表すことは難しい．しかし，CPU 温度の上昇は冷却性能を超えた熱が発生することにより起こると解釈すると，CPU 温度変化から消費電力をある程度推定することは可能と考えた．

図 7 に CPU の負荷を変化させた場合の CPU 温度変化と消費電力変化の関係を示す．ここでは，浮動小数点演算数と固定小数点演算の割合を変えることで flops 値を変化させながら消費電力を測定した．1 ラックの 96CPU 全てで同じプログラムを実行し，その平均値を求めている．縦

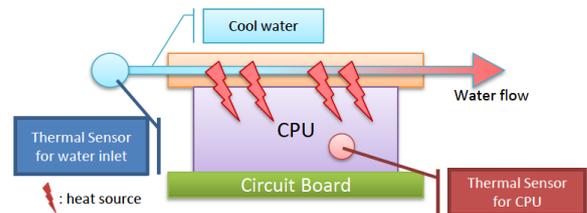


図 6 CPU の冷却機構

軸は消費電力と温度変化を，横軸は CPU の理論性能に対する flops 値の割合をそれぞれ表している．flops 値の低い領域では，CPU 温度変化が比例していない部分もあるが，全体的には CPU 温度変化も消費電力も flops 値に比例している．縦軸を消費電力，横軸を CPU 温度変化としてプロットしたグラフを図 8 に示す．この図からも CPU 温度変化と消費電力に比例関係があることが分かる．

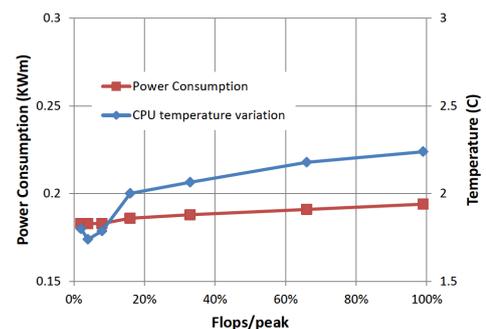


図 7 CPU 負荷と CPU 温度変化，消費電力の関係

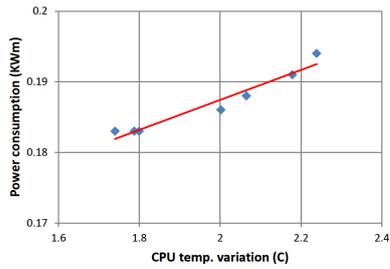


図 8 CPU 温度変化と消費電力の関係

5.2.2 メモリ

メモリ負荷と消費電力の関係について調査した。

図 9 に「京」の System Board (SB) の構成を示す。1 枚の SB には、計算ノード (CPU1 台と ICC1 台、メモリ 16GiB で構成) が 4 台載っている。CPU と ICC は主に冷却水で冷やされるので、SB 排気温度変化は、主にメモリの発熱によって生じると考えられる。

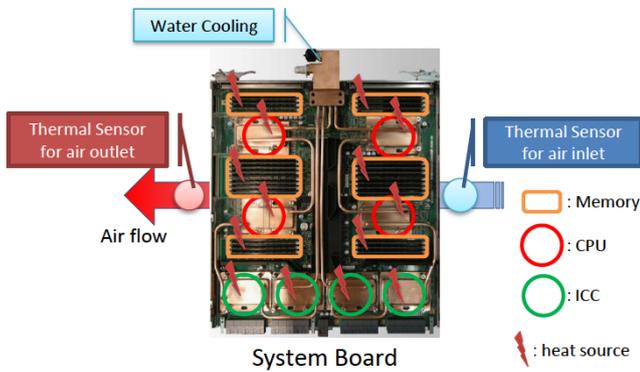


図 9 System Board の構成

図 10 にメモリ負荷 (メモリスループット) を変化した場合の SB 排気温度変化と消費電力変化の関係を示す。1 ラックの 96CPU 全てで同じプログラムを実行し、24 枚の SB の平均値を求めている。縦軸は消費電力と温度変化を、横軸はメモリスループットをそれぞれ表している。グラフから消費電力と SB 排気温度変化がメモリスループットに比例していることがわかる。縦軸を消費電力、横軸を SB 排気温度変化としてプロットしたグラフを図 11 に示す。この図からも SB 排気温度変化と消費電力に比例関係があることが分かる。

5.2.3 ICC

ICC は設計上、消費電力は一定となっている。実際に ICC の消費電力が一定かどうか、ICC の負荷を変化させた場合の消費電力について調査した。

ICC には 4 つの TNI (Tofu Network Interface) が繋がっている。リンクあたりの性能は 5GiB/sec × 2 である。今回の測定では、通信に使用する TNI の数を変えて ICC の負荷 (通信量) を変化させた。図 12 に ICC の負荷を変化させた場合の消費電力の変化を示す。縦軸は消費電力を、横

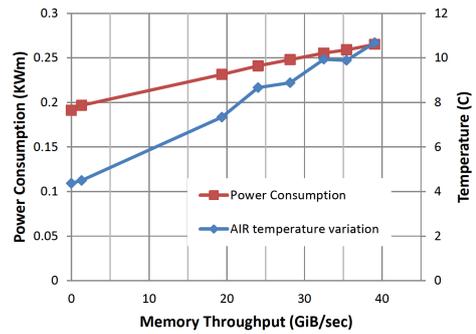


図 10 メモリスループットと消費電力, SB 排気温度変化の関係

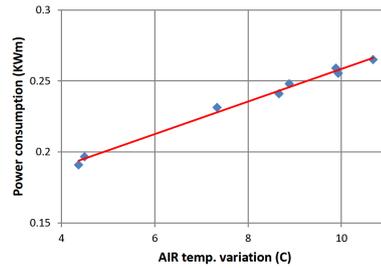


図 11 SB 排気温度変化と消費電力の関係

軸は使用する TNI の数に応じたネットワークスループットを表している。図からネットワークスループットに比例して消費電力が変動していることがわかる。通信時のデータはメモリから読みだされるため、ネットワークスループットに応じてメモリも電力を消費する。実際、測定された消費電力の変動はメモリ負荷による消費電力の変動 (図 10) と一致しており、ICC 自体の消費電力は一定と考えることができる。

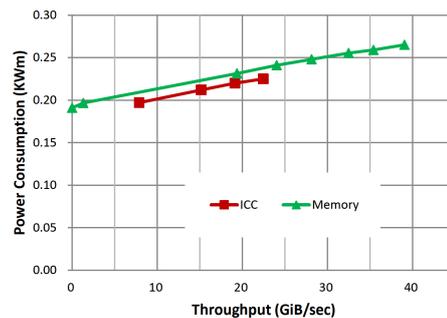


図 12 ICC の負荷と消費電力の関係

以上の結果から、CPU 温度変化と SB 排気温度変化でジョブ単位の消費電力を推測可能と判断した。

5.3 消費電力の推定

温度センサ情報をもとにジョブの消費電力の推定を行った。

ジョブの実行時に消費された電力は全て熱となると仮定し、CPU 温度変化と SB 排気温度変化から消費電力を推定する。CPU 温度変化と消費電力変化 (図 8)、SB 排気温度

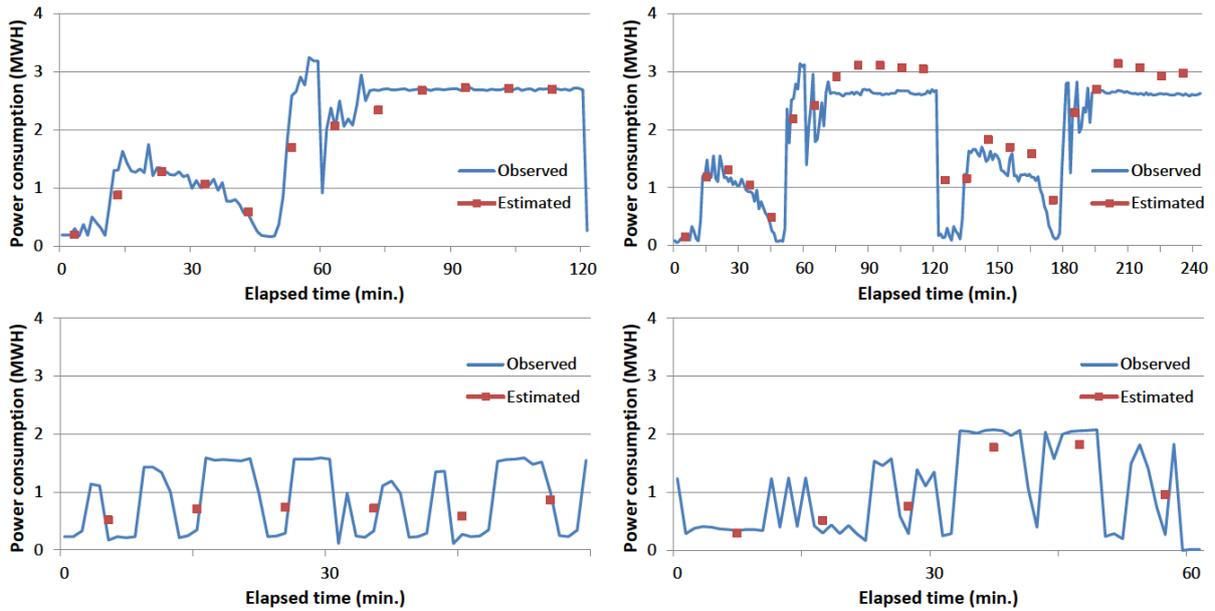


図 13 全ノードを使用したジョブの消費電力の測定値と推定値

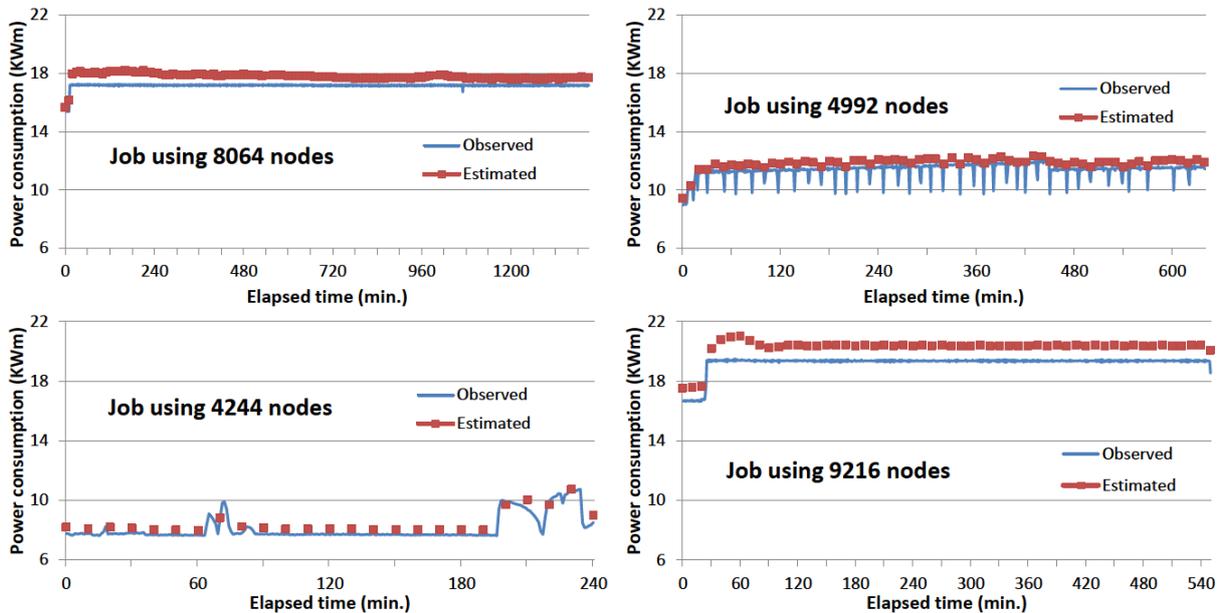


図 14 中規模ジョブの消費電力の換算値と推定値

変化と消費電力変化(図 11)の関係から、消費電力の推定式を次のように定めた。

$$P = a \cdot T_{cpu} + b \cdot T_{air} + c \quad (2)$$

P はシステム全体の消費電力を、 T_{cpu} は平均 CPU 温度変化を、 T_{air} は平均 SB 排気温度変化をそれぞれ表す。係数 a, b, c は図 5 のデータをもとに求めた。この時の標準誤差は 0.150335349919753 であった。

$$a = 0.802393382361262$$

$$b = 0.345223838880426$$

$$c = 7.67202252302052$$

温度センサの情報をもとに推定した結果と、「京」で実際

に測定した消費電力との比較を行った。

図 13 に、「京」の全計算ノードを使用したジョブの消費電力の測定値と推定値を示す。これらは月に一度実施している大規模実行期間に実行された実行時間の比較的長いジョブのデータである。現状では、システム全体の消費電力は 1 分毎に取得できるが、温度センサの情報は 10 分毎にしか取得できないため、推定値は 10 分毎にプロットしている。これらから、測定値と推定値が近似していることがわかる。しかし、推定値は 10 分毎にしか計算できないため、消費電力の変動が大きいジョブの場合、推定値と測定値が近似している場合でも推定値から予測されるジョブ全体の消費電力の変動と実際の変動が一致しない場合があ

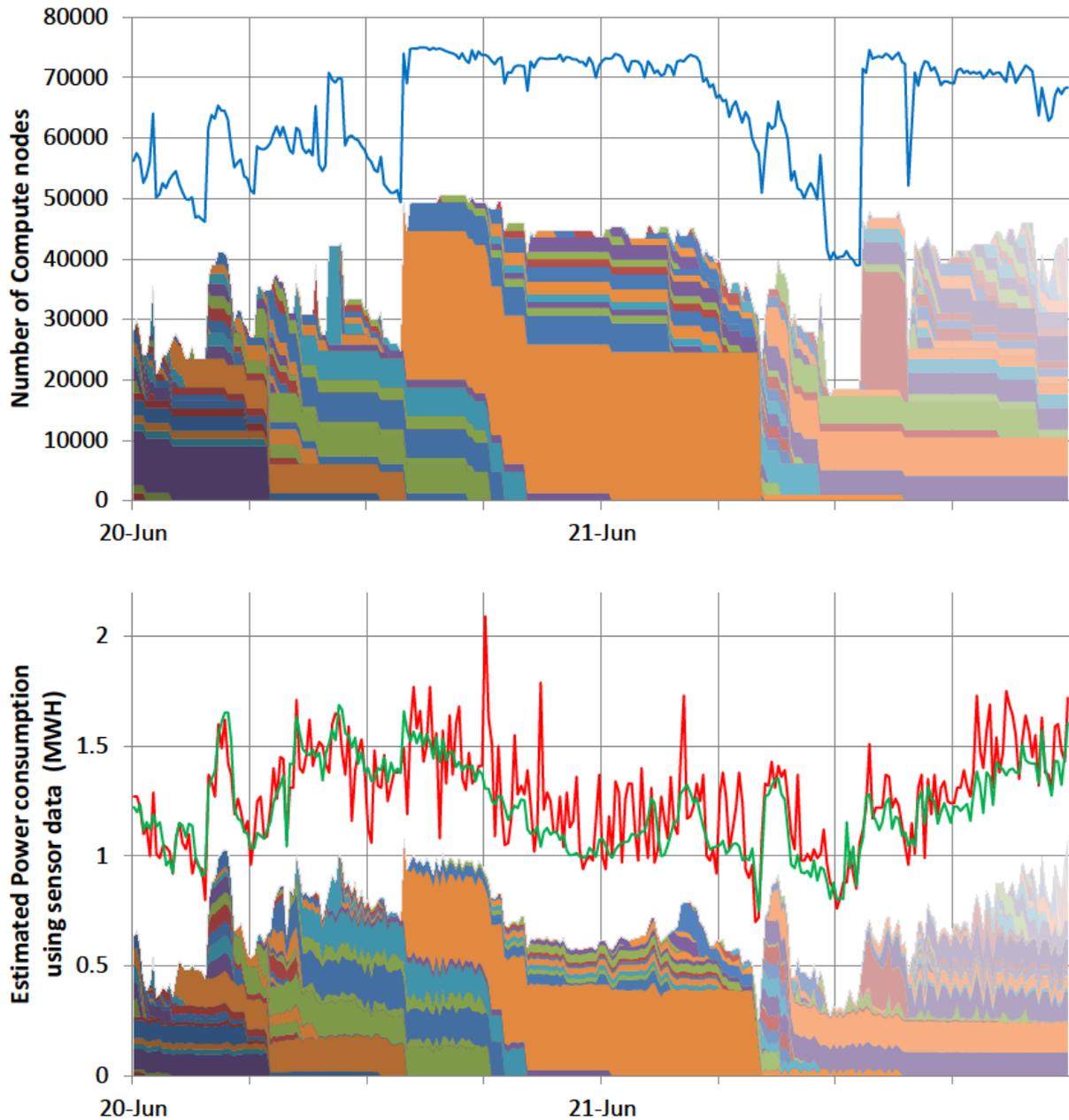


図 15 ジョブ単位のノード数（上）と温度センサ情報を使用した推定消費電力（下）

ることがわかる．これを改善するためには，温度センサ情報の取得間隔を短くする必要がある．

次に，電力計が取り付けられている計算ラックを使用して中規模のジョブについて比較を行った．「京」では一部の計算ラックにしか電力計は取り付けられていないため，通常運用で実行時に電力計のある計算ラックが含まれていたジョブを対象とした．計算ラック単位で測定された消費電力をジョブ全体の消費電力に換算して比較を行っている．図 14 に結果を示す．換算値は 1 分毎に，推定値は 10 分毎にプロットしている．なお，計算ラックの測定値にはディスクラックの消費電力は含まれないため，推定値からディスクラック相当分の消費電力を除外している．推定値と換算値がよく一致しているジョブもあるが，推定値が全体的

に高めとなったジョブもあることがわかる．本稿で提案する推定式では，消費電力は CPU 温度変化と SB 排気温度変化で決まる．ジョブには CPU を主に使う場合と，メモリを主に使う場合があることが分かっている．推定値と測定値の差が大きいジョブの特性を調査することで，推定式の精度を高めることができると考えている．

次に，実際の運用で電力超過が発生した状況を想定し，複数ジョブが実行されている状況での消費電力の推定を行った．図 15 に，実際に「京」上で実行されたジョブ毎のノード数と推定消費電力のグラフを示す．ここでは，1,000 ノード以上を使用したジョブを対象とし，消費電力はジョブ実行による変動値を表している．図 15 の上のグラフがジョブ毎のノード数を，下のグラフが温度センサ情報を基

にしたジョブ毎の推定消費電力をそれぞれ表している。同じ時間帯の同じ色は同一ジョブ示していて、グラフ中の青の折れ線はシステム全体で使用されたノード数を、赤の折れ線はシステム全体の消費電力の測定値を、緑の折れ線はシステム全体の消費電力の推定値をそれぞれ表している。白い領域は1,000ノード未満のジョブである。

温度センサ情報を使用した推定消費電力では、ジョブ毎の消費電力はノード数に比例しておらず、単純にノード数からジョブ毎の消費電力を推定する方法よりも効率よく停止ジョブを選択できることがわかる。一方、システム全体の消費電力の測定値と推定値を比較すると、推定値は測定値の急激な変動にあまり追従できていない。これは、急激な電力変動に温度変化が追従しにくいことが原因のひとつであると考えている。

6. おわりに

本稿では、消費電力を考慮した「京」の運用方法として、電力超過の発生が予測される大規模ジョブについて消費電力の観点から事前に審査する体制（事前審査制度）と、電力超過が発生した場合に適切にジョブを停止する手法について述べた。

事前審査制度により、事前に大規模ジョブの実行時の消費電力を予測し、それをもとにジョブの実行を調整して電力超過の可能性を減らすことができるようになった。しかし、消費電力の予測は完全ではないため、電力超過が発生した場合にそなえて、ジョブの緊急停止の仕組みを構築した。各計算ラックに取り付けられた温度センサ情報とシステム全体の消費電力から個々のジョブの消費電力を推定し、電力超過時に停止するジョブを適切に選択する方法を検討した。この推定方法では、ジョブ単位の大まかな消費電力を推定することができたが、現状では温度センサの精度やサンプリング間隔の問題等から正確な消費電力の推定は難しい。ジョブ実行時のプロファイル情報を利用することができれば、より正確な消費電力の推定が可能であると思われるが、プロファイル情報はジョブ実行が終了した後でなくては取得できない。そのため、電力超過発生時に速やかにジョブを停止することはできない。本手法の利点は、随時取得できる温度センサから消費電力をリアルタイムに推定することができる点である。得られた情報から、超過電力分だけ消費電力を減らしつつ、ジョブ停止によって失われる計算資源量を最小にするジョブを選ぶことができる。

今後は、電力超過時に人手を介さずに自動的にジョブを停止する環境の構築など、実運用への応用について検討を続けていきたいと考えている。

参考文献

[1] 黒川原佳, 庄司文由: スーパーコンピュータ「京」システム概要, 情報処理, Vol.53, No.8, pp.759-766

(2012).

[2] 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).

[3] 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 (ICCS), (2014)

[4] 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: 電力消費量の上限を考慮した「京」の運用, 情報処理学会研究会報告 Vol.2014-HPC-146 No.4 (2014).

[5] M. Etinski, J. Corbalan, J. Labarta, M. Valero: Parallel job scheduling for power constrained HPC systems, Parallel Computing, Volume 38, Issue 12, pp. 615-630, (2012).

[6] O. Sarood, A. Langer, A. Gupta, and L.V. Kale: Maximizing Throughput of Overprovisioned HPC Data Centers Under a Strict Power Budget, Super Computing 2014 (SC'14), (2014).

[7] Axel Auweter, Arndt Bode, Matthias Brehm, Luigi Brochard, Nicolay Hammer, Herbert Huber, Raj Panda, Francois Thomas, and Torsten Wilde: A Case Study of Energy Aware Scheduling on SuperMUC, International Supercomputing conference 2014 (ISC'14), (2014).

[8] 宇野篤也, 肥田元, 池田直樹, 井上文雄, 塚本俊之, 末安史親, 庄司文由: 「京」におけるジョブ単位の消費電力推定の検討, 情報処理学会研究会報告 Vol.2014-HPC-147 No.20 (2014).