情報処理学会論文誌　コンピューティングシステム　Vol.8　No.4　13–25 (Nov. 2015)

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]

受付日 2015年4月21日, 採録日 2015年7月23日

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，ジョブごと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」では計算ノードごと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ごと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，ジョブごと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．

キーワード：スーパーコンピュータ「京」，消費電力推定，システム運用

Operation of the K computer Focusing on System Power Consumption

Atsuya Uno[1,a)]　Hajime Hida[2]　Fumio Inoue[1]　Naoki Ikeda[2]　Toshiyuki Tsukamoto[1]
Fumichika Sueyasu[3]　Satoshi Matsushita[1]　Fumiyoshi Shoji[1]

Received: April 21, 2015, Accepted: July 23, 2015

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.

Keywords: K computer, power consumption estimation, system operation

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 の採用等，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 10 MW，高負荷時には 14 MW 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

[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, Kobe, Hyogo 650–0047, Japan
[2] 株式会社富士通ソーシアルサイエンスラボラトリ
FUJITSU SOCIAL SCIENCE LABORATORY LIMITED, Kawasaki, Kanagawa 211–0063, Japan
[3] 富士通株式会社
FUJITSU LIMITED, Minato, Tokyo 105–7123, Japan
[a)] uno@riken.jp

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に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度か発生した．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ごと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」では計算ノードごと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ごと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 1.27 PiB のメモリ，11 PB のローカルファイルシステム（LFS）と 30 PB 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 台の出力は約 5 MW 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]．

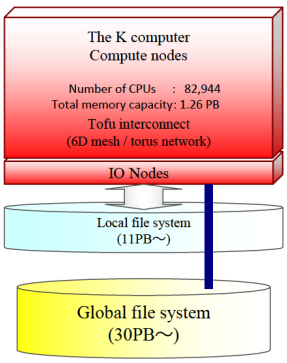

図 1　「京」のシステム構成
Fig. 1　System configuration of K computer.

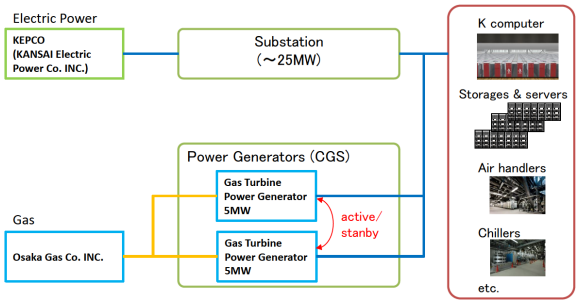

図 2　AICS の電源設備
Fig. 2　Power supply facilities of AICS.

表 1　AICS 全体の想定消費電力（共用開始時）
Table 1　Details of estimated power consumption of AICS (at start of shared operation).

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

3. 電力消費

「京」の消費電力は無負荷時で約 10 MW，高負荷時で 14 MW 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 12 MW として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 0.75 MW 増の 12.75 MW 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]．

電力超過を防ぐ運用に関する研究はこれまでにも多く行われているが [5], [6], [7]，その多くは電力制御用のハード

[*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．

[*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．

ウェアを前提としており，こういった機能を持たない「京」では適用は難しい．また，「京」はすで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の対応では間に合わない．そのため，つねに 2 台の CGS を稼働させることが前提となり，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 1 MW あたりの単価は CGS で発電する方が安価であったが，その後のガス発電単価および電力単価の状況は逆転しており，商用電力から受電したほうがコスト的に有利となっている．**図 3** に，2011 年 4 月から 2015 年 3 月までの 4 半期ごとの電力単価に対するガス発電単価比の平均値の推移を示す．図 3 からも分かるように，電力単価よりもガス発電単価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，つねに CGS を 2 台稼働させる場合には，CGS の故障やメンテナンス時にノードを停止させる等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うえ，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

ジョブごと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，実行するジョブごと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，すべ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等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

電力超過を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ごと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 4** に事前審査制度のフローを示す．

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，そのとき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．たとえば，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 8 MW 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．

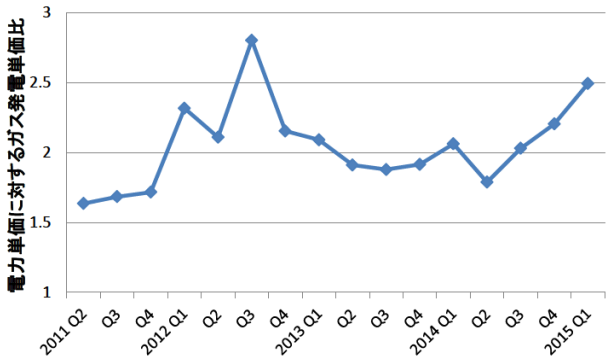

図 3　電力単価に対するガス発電単価比

Fig. 3　Comparison of gas power generation costs and commercial electricity costs.

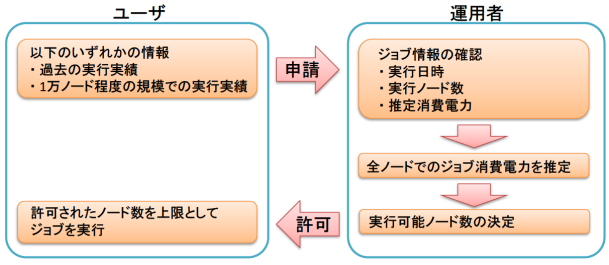

図 4　事前審査制度のフロー

Fig. 4　Flowchart of preliminary review.

表 2　事前審査結果（2014 年 8 月分）
Table 2　Results of preliminary review (Aug. 2014).

	ジョブの消費電力 (MW)		
計算ノード数	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

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

ここでの各パラメータは以下のとおりである．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 (4 MW)．

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 から分かるように，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等，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分かっている．

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 5** 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．

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]．ジョブ単位の消費電力を推定することで，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．電力超過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．このとき，停止したジョブのユーザがなるべく不利益にならないよう，これらのジョブは他のジョブより優先して再実行する．

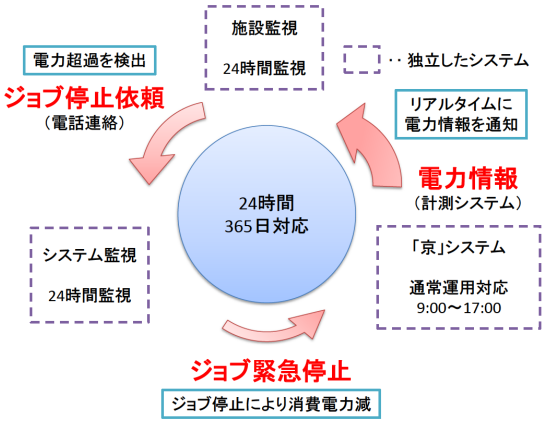

図 5　ジョブの緊急停止フローの概要
Fig. 5　Flowchart of emergency job stop.

「京」では，1 年間の計算資源量を年度始めにユーザに配分するため，ジョブの停止で計算資源量が失われることは，それ以降にユーザが使用可能な計算資源量が減ることに等しい．そのため，ジョブを停止することで失われる計算資源量を最小にすることは，ユーザ面においてもプラスとなる．

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) 計算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すべ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が最小になるようにジョブを選択する．

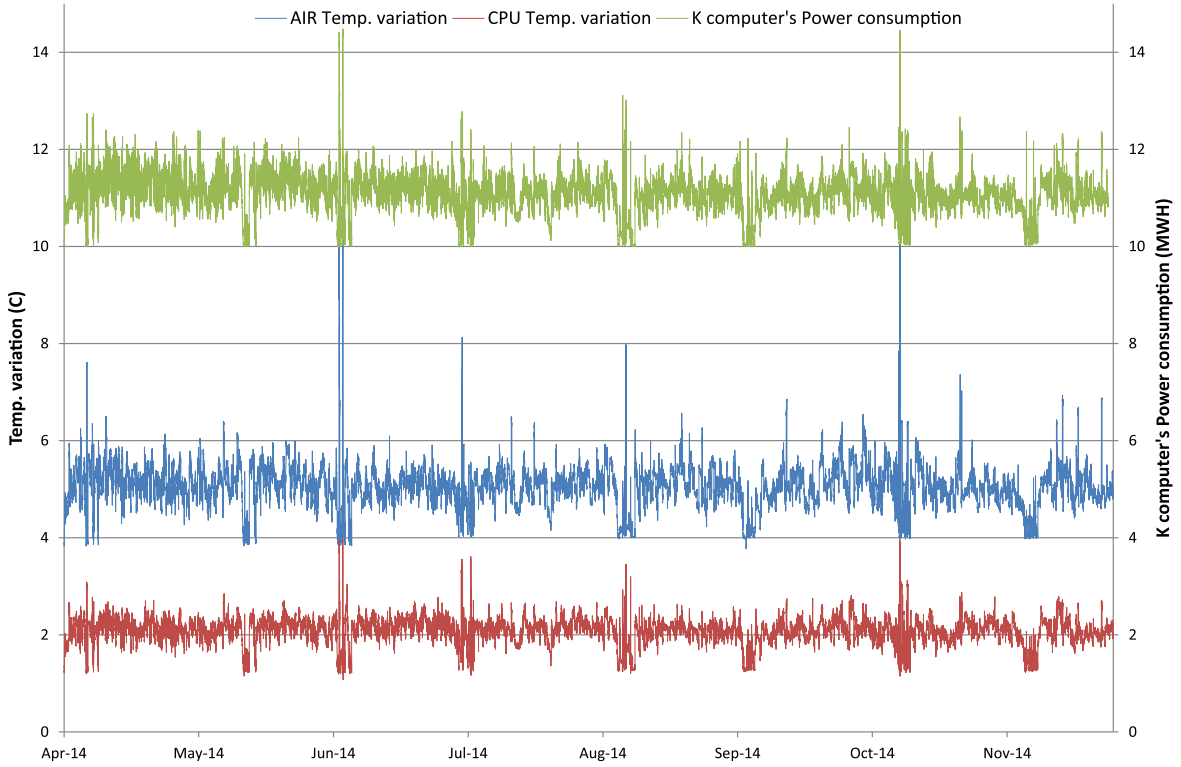

図 6　平均 CPU 温度変化，平均 SB 排気温度変化とシステム全体の消費電力変化の関係
Fig. 6　Relations among average CPU temperature variation, AIR temperature variation and whole system power consumption.

(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，実際にはジョブごと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良い方法とはいえない．

(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．これら温度センサの情報は，現状では 10 分ごと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 6 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 6 におけるシステム全体の消費電力と各温度変化の相関係数を求めたところ，それぞれ 0.90，0.88 となり，高い相関関係が認められた．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．

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見られなかった．

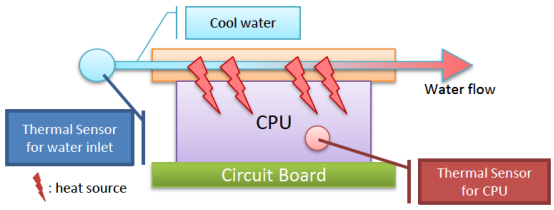

図 7　CPU の冷却機構
Fig. 7　CPU cooling system.

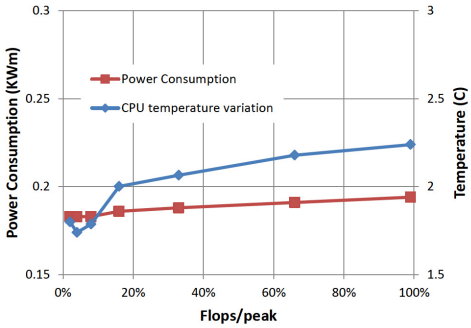

図 8　CPU 負荷と CPU 温度変化，消費電力の関係
Fig. 8　Relations among CPU load, CPU temperature variation and power consumption.

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 7 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 8 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 計算ラックの 96 CPU すべ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 9 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．

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 10 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 16 GiB 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

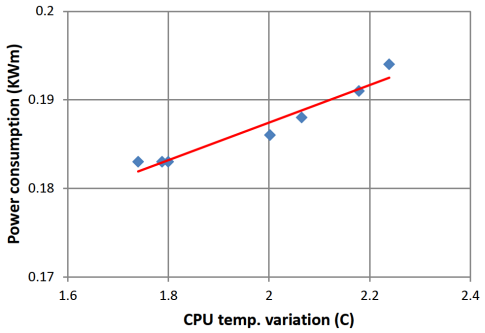

図 9　CPU 温度変化と消費電力の関係
Fig. 9　Relation between CPU temperature variation and power consumption.

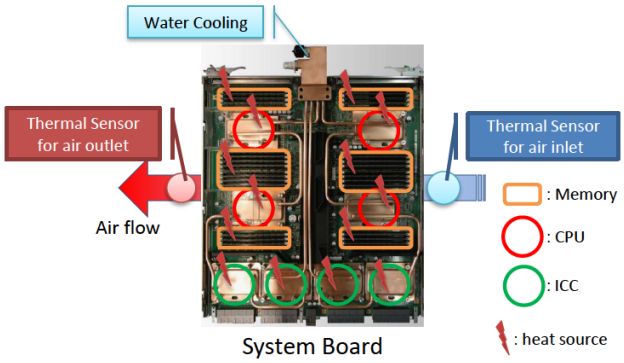

図 10　System Board の構成
Fig. 10　Configuration of System Board.

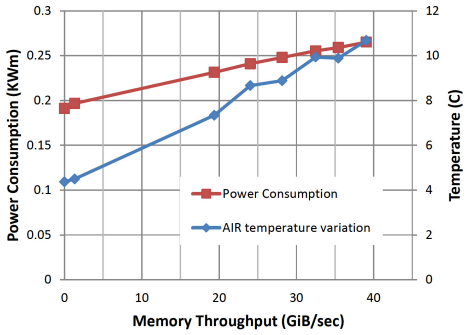

図 11　メモリスループットと消費電力，SB 排気温度変化の関係
Fig. 11　Relations among memory throughput, power consumption and AIR temperature variation.

の発熱によって生じると考えられる．

図 11 にメモリ負荷（メモリスループット）を変化させた場合の SB 排気温度変化と消費電力変化の関係を示す．1 計算ラックの 96 CPU すべ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分かる．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 12 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．

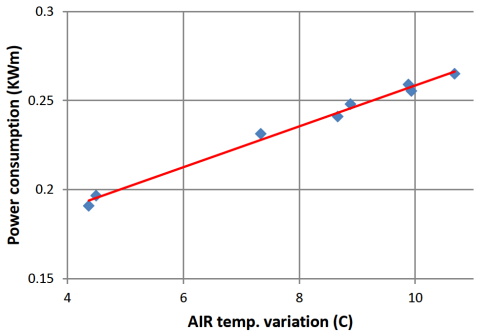

図 12　SB 排気温度変化と消費電力の関係
Fig. 12　Relation between AIR temperature variation and power consumption.

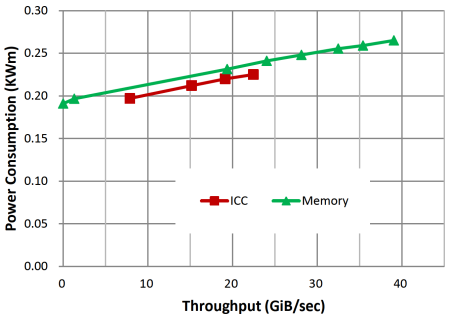

図 13　ICC の負荷と消費電力の関係
Fig. 13　Relation between ICC load and power consumption.

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）がつながっている．リンクあたりの性能は 5 GiB/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 13** 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分かる．通信時のデータはメモリから読み出されるため，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 11）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．

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

温度センサの情報をもとにジョブの消費電力の推定を行った．

表 3　推定値の評価結果
Table 3　Evaluation results of estimated power consumption.

	R^2
システム全体	0.9246
ラック単位（全体係数）	0.8940
ラック単位（ラック係数）	0.9328

5.3.1　推定式

ジョブの実行時に消費された電力はすべ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 9），SB 排気温度変化と消費電力変化（図 12）の関係から，消費電力の推定式を次のように定めた．

$$P = a \cdot T_{cpu} + b \cdot T_{air} + c \tag{2}$$

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 6 のデータ（約 33,500 件）をもとに求めた．このときの標準誤差は 0.150335349919753 であった．

$$a = 0.802393382361262$$
$$b = 0.345223838880426$$
$$c = 7.67202252302052$$

5.3.2　推定式の評価

推定式の精度を定量的に評価するため，決定係数 R^2 を用いて評価した．ここで使用した R^2 は以下の式で表される．

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - f_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{3}$$

計算結果が 1 に近いほど精度が良い．ここで y，f はそれぞれ測定値と推定値で，N は評価データ数である．

評価には 2014 年 12 月から 2015 年 2 月までの 3 カ月間のデータ（約 13,000 件）を使用した．システム全体の推定値のほか，電力計が取り付けられている計算ラックを使用してラック単位の推定値の評価も行った．ラック単位の評価では，システム全体の温度データの平均値から求めた係数（全体係数）による推定値と評価対象の計算ラックの温度データから別途求めた係数（ラック係数）による推定値の評価も行った．これらの結果を**表 3** に示す．システムの規模が違うので単純に比較できないが，サーバ内部を詳細にモデル化して推定した場合（$R^2 = 0.97$）[9] よりも精度は低いが，システム全体の推定結果は $R^2 > 0.92$ と精度良く推定できていることが分かる．また，ラック単位では全体係数よりもラック係数による推定の方が精度が良い．

図 14 に，評価対象の計算ラックにおける，全体係数 (a) とラック係数 (b) の推定値と測定値の関係を示す．図中の赤い点は全体係数による推定値で，青の直線がその近似直線を，緑の点はラック係数による推定値で，紫の直線がそ

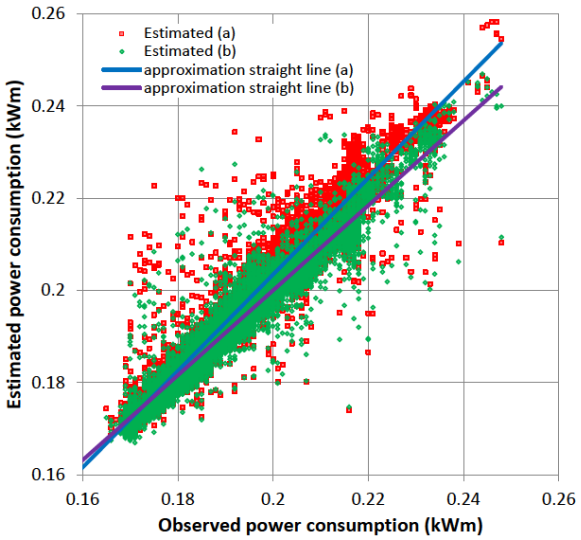

図 14　評価対象の計算ラックにおける推定値と測定値の関係
Fig. 14　Relation between estimated power consumption and observed power consumption.

の近似直線をそれぞれ示している．グラフから分かるように，消費電力に比例して全体係数の推定値の誤差が大きくなっており，この部分で推定精度に差が生じたものと考えられる．このことから，計算ラックごとに係数を求めることができれば，システム全体の推定精度を高めることができると思われる．しかし，計算ラックごとに係数を求めるためには計算ラックごとに電力計が必要となる．

5.3.3　ジョブ単位の評価

次に，実際に「京」で実行されたジョブ単位での推定値と測定値の比較を行った．比較には消費電力の測定が必要なため，全ノード（82,944 ノード）を使用したジョブと，使用したノードに電力計が設置された計算ラック上のノードがすべて含まれていたジョブを対象とした．計算ラック単位の評価には図 14 と同じ計算ラックを使い，ジョブが使用したノード数にかかわらず該当計算ラックのみ（96 ノード）を対象とした．全体係数 (a) のほかにラック係数 (b) による推定値についても評価した．評価には平均二乗誤差（RMSE: Root Mean of Square Error），平均誤差，最大誤差を用いた．RMSE は以下の式で表される．

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - f_i)^2} \qquad (4)$$

計算結果が 0 に近いほど精度が良い．ここで y，f はそれぞれ測定値と推定値で，N は評価データ数である．

2014 年 12 月から 2015 年 2 月までに実行されたジョブから，以下のような特徴のあるジョブについて評価を行った．全計算ノードを利用したジョブは数が少ないため，Job C–F については，電力計の取り付けられた計算ラックを使用したジョブを対象とした．

- 全計算ノードを利用したジョブ（Job A, B）
- RMSE が最も良かったジョブ（Job C）
- RMSE が最も悪かったジョブ（Job D）
- ジョブ実行時の測定値の平均変動量[*3]が最も大きかったジョブ（Job E）
- 全体係数とラック係数による平均推定値の差が最も大きかったジョブ（Job F）

図 15 と**図 16** にジョブごとの消費電力の測定値と推定値を，**表 4** にジョブごとの平均二乗誤差，平均誤差，最大誤差をそれぞれ示す．図 15 中の青の線は測定値を，赤の点は全体係数 (a) による推定値を，緑の点はラック係数 (b) による推定値をそれぞれ表している．現状では，システム全体の消費電力は 1 分ごとに取得できるが，温度センサの情報は 10 分ごとにしか取得できないため，推定値は 10 分ごとにプロットしている．図 16 は，図 15 のデータを横軸を測定値，縦軸を推定値としてプロットしたもので，図中の青の直線は理想的な結果を，赤の点は全体係数による推定値を，緑の点はラック係数による推定値をそれぞれ表している．

Job A の平均誤差は 2%以下で，図 16 でも理想直線上に点が集まっており，精度良く推定できていることが分かる．一方，Job B の平均誤差は 5.5%近くあり，最大誤差は約 10%であった．図 15 から分かるように，推定値が消費電力の平均値に近い値となっている．図 16 では測定値の変動にかかわらず推定値が水平方向に分散していることから，消費電力の急激な変動に温度変化が追従できず，推定値が変動時の平均値に近い値になったと推測される．また，Job B のように消費電力が頻繁に変動するジョブの場合，仮に精度良く消費電力を推定することができたとしても，推定間隔が広い場合には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考えられる．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．

Job C は精度良く推定ができており，全体係数の最大誤差でも 2%程度に収まっている．一方，Job D は全体係数の平均誤差で 3.5%に近い値となっている．図 16 から全体係数の推定値は全体的にずれていることが分かる．ラック係数の推定値は理想直線に近いことから，全体係数の誤差が大きい原因は計算ラックの個体差によるものと推測される．また，Job D は消費電力が一定にもかかわらず推定値が途中から大きく変化している．この原因を調査したところ，この時間帯に CGS の発電量を増加させたため，CGS から冷凍機に供給される水蒸気量が増えて水冷入力温度とラック吸気温度が大きく下がっていたことが分かった．CPU 温度変化と SB 排気温度変化は，水冷入力温度とラック吸気温度に対する差分なのでこれらの変化の影響を受けにくいと考えていたが，急激な温度変化の影響を受ける場合があることがこの結果から判明した．今後，これらの変

[*3] ジョブ実行開始直後と終了直前の 5 分間を除いた測定値の平均変動量．

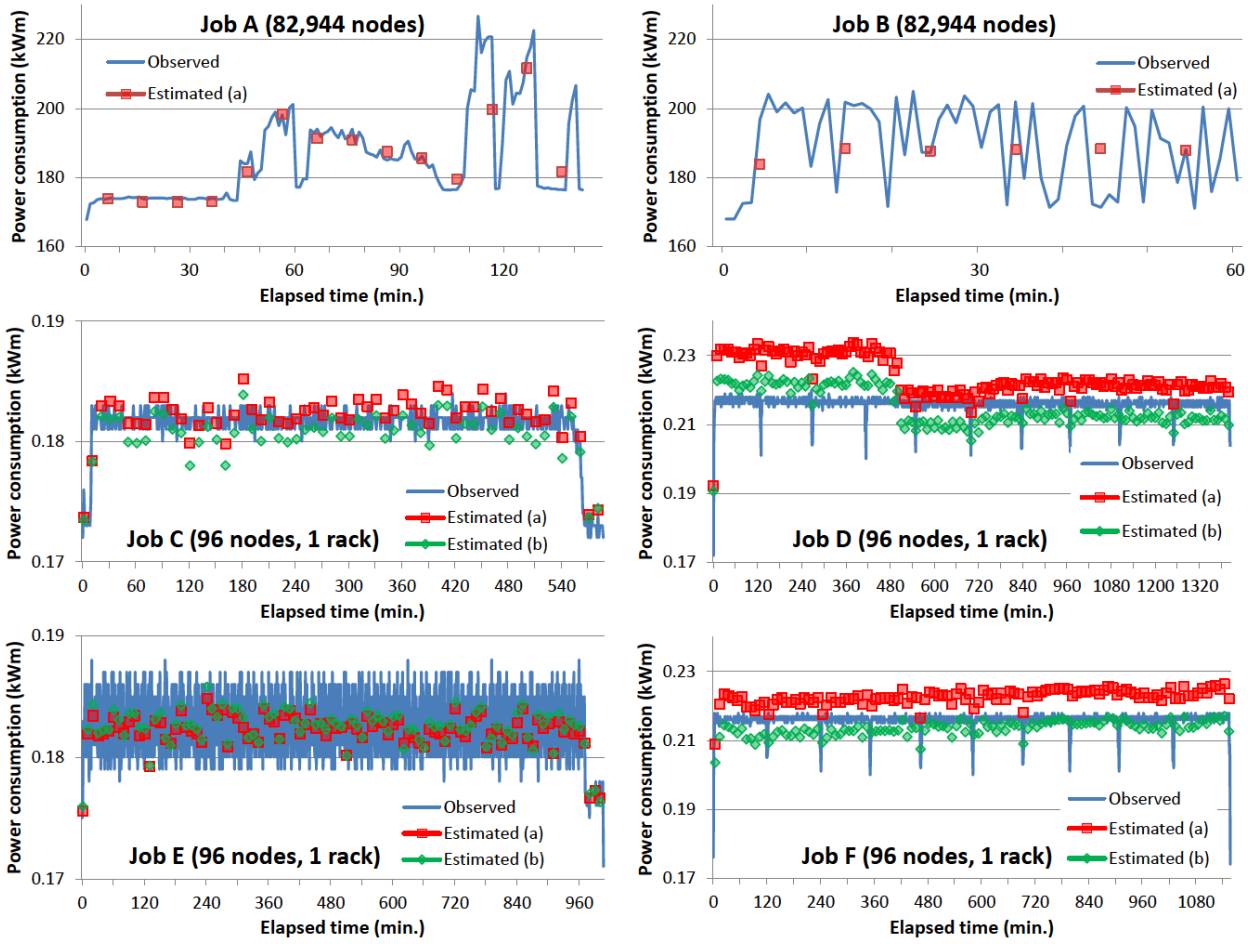

図 15　ジョブ単位の消費電力の測定値と推定値

Fig. 15　Estimated power consumption and observed power consumption.

表 4　ジョブごとの誤差

Table 4　Errors of estimated power consumption of each job.

ジョブ	平均二乗誤差 (RMSE)	平均誤差 (kWm)	(%)	最大誤差 (kWm)	(%)	平均変動量 (kWm)
A (a)	6.488	3.54	1.93	22.93	12.98	3.02
B (a)	12.206	10.21	5.45	20.46	10.02	12.01
C (a)	0.00154	0.00129	0.71	0.00341	1.92	0.00092
(b)	0.00160	0.00123	0.68	0.00402	2.21	
D (a)	0.00932	0.00799	3.57	0.01736	8.07	0.00140
(b)	0.00513	0.00474	2.21	0.01239	6.10	
E (a)	0.00270	0.00222	1.22	0.00744	4.07	0.00356
(b)	0.00275	0.00228	1.25	0.00707	3.86	
F (a)	0.00707	0.00673	3.02	0.01608	7.81	0.00133
(b)	0.00345	0.00279	1.31	0.01243	5.76	

化の影響を調査する予定である．

Job E は電力変動の大きいジョブであるが，Job B ほど変動幅が大きくないため，平均誤差は全体係数の場合で1.2%程度に収まっている．しかし，図 16 からも分かるように，Job B と同様に消費電力の急激な変動に温度変化が追従できていないため，推定値が変動時の平均値に近い値となっている．

Job F はジョブの消費電力が全体的に高い．図 14 に示したように，今回の評価に使用した計算ラックでは消費電力に比例して全体係数とラック係数の誤差が大きくなるため，全体係数とラック係数の推定値の差が大きくなったものと考えられる．また，経過時間に比例して推定値が徐々に高くなっているが，これは CPU 温度変化が経過時間に比例して上昇していることによるものである．CPU の高

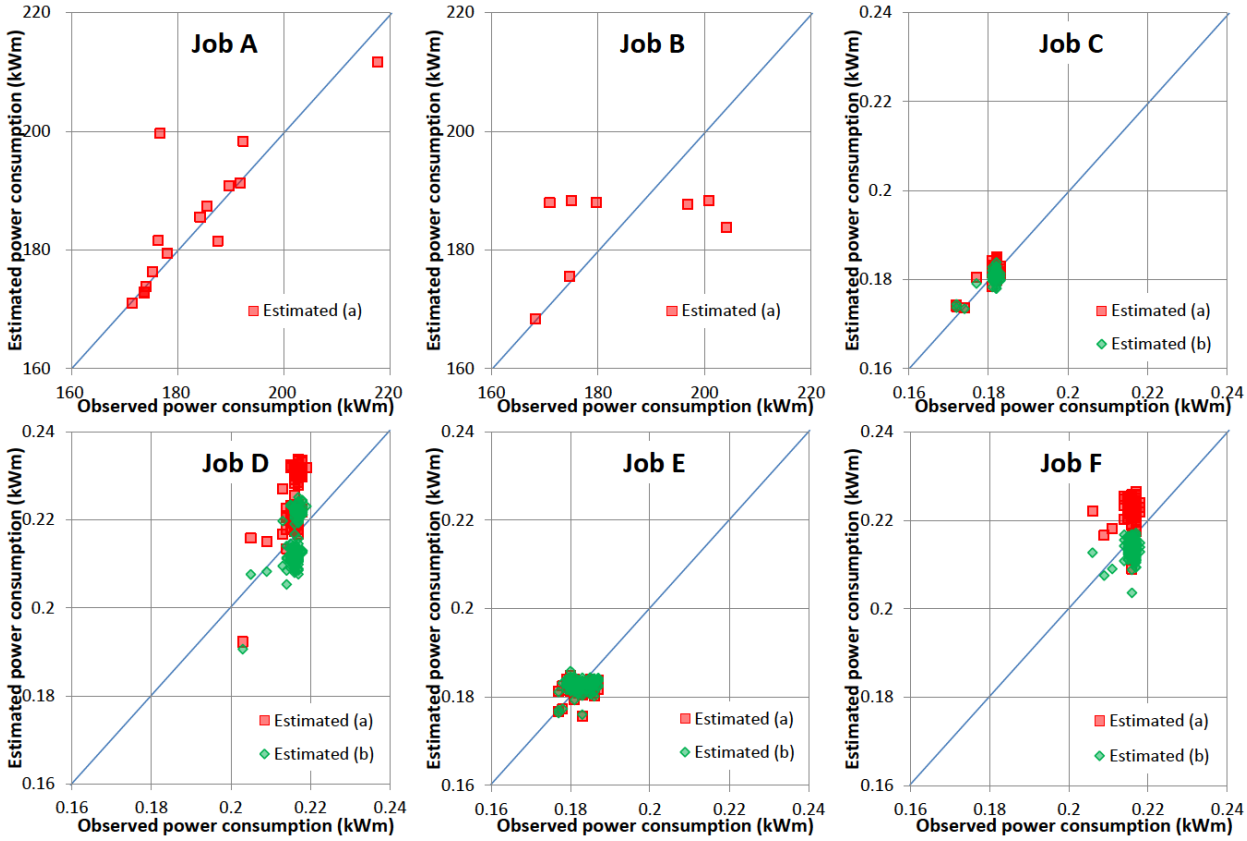

図 16　ジョブ単位の測定値と推定値の関係

Fig. 16　Relation between estimated power consumption and observed power consumption.

負荷状態が連続したことが原因と思われるが，詳細はジョブのプロファイル情報を解析する必要がある．

以上の評価から，本稿で提案する温度センサを利用した消費電力の推定では，消費電力が短時間に頻繁に変動するような場合では推定値が大きくずれる場合があるが，それ以外では精度の良い推定が可能なことが分かる．しかし，推定式の各係数はシステム全体の温度データの平均値をもとに計算した値であるため，ジョブ単位の推定では計算ラックの個体差の影響を受けて精度が低くなる可能性がある．

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 [10]．ジョブのプロファイル情報をもとに各温度変化と消費電力の関係を分析することで，推定式を改良し精度を高めることができると考えている．

5.3.4　ジョブの緊急停止時の消費電力推定

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 17** に，実際に「京」上で実行されたジョブごと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 17 の上のグラフがジョブごとのノード数を，下のグラフがジョブごと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を示していて，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，ジョブごと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ごとの消費電力を推定する方法よりも効率良く停止ジョブを選択できることが分かる．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，前述のとおり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 1 つであると考えている．

6.　関連研究

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て，これまでに消費電力を考慮した運用方法がいくつか提案されている．

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運用を行う方法としては，整数線形計画法に基づき電力資源を最大限に利用するスケジューリング方法 [5] や，ジョ

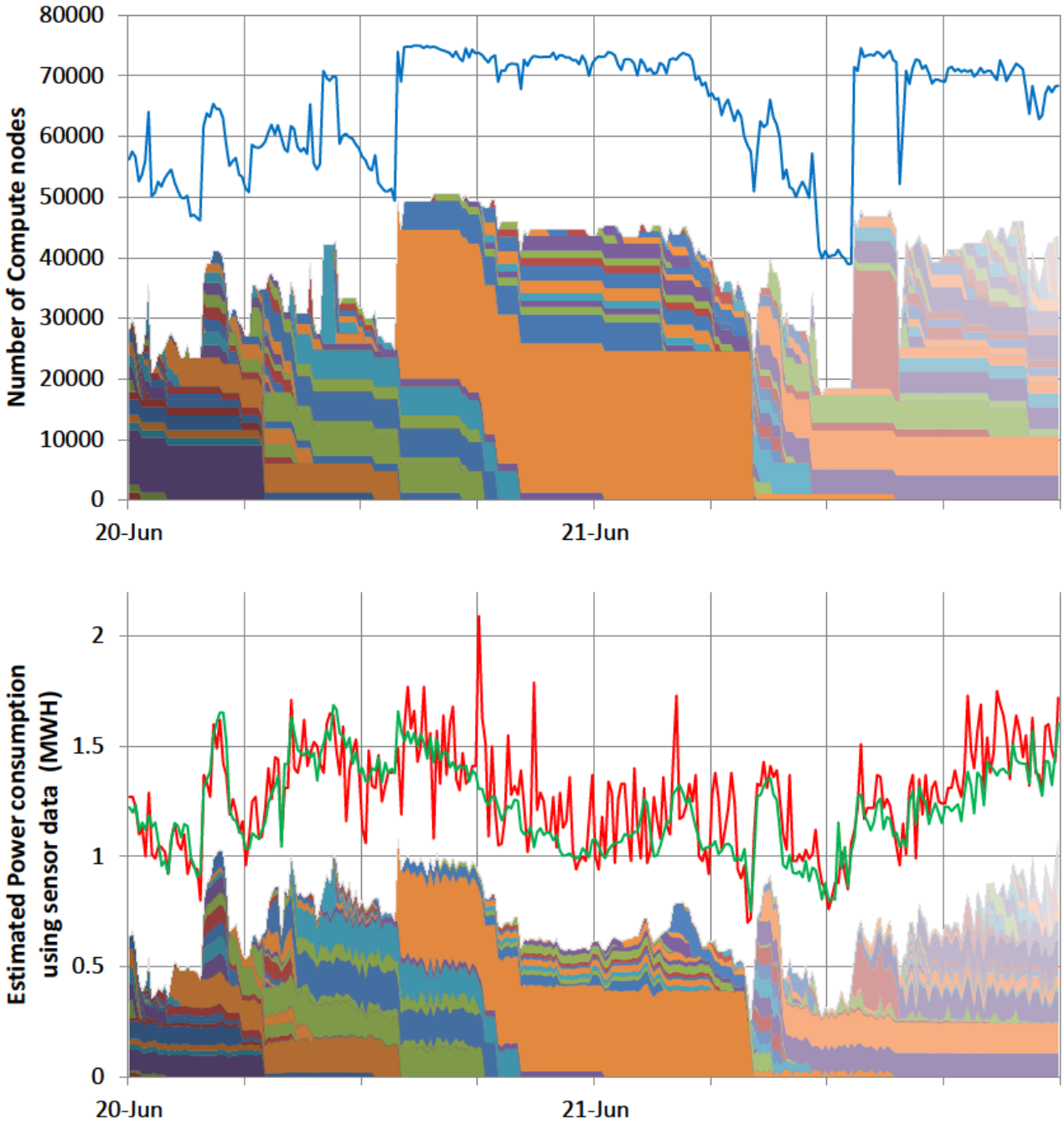

図 17　ジョブ単位のノード数（上）と温度センサ情報を使用した推定消費電力（下）

Fig. 17　Number of nodes (top) and estimated power consumption (bottom) of each job.

ブの消費電力と実行時間を CPU の周波数を変更して最適化するスケジューリング方法 [6]，ジョブのスループットが最大化されるようにスケジューリングする手法 [7] 等，提案されている．これらの研究では DVFS（Dynamic Voltage Frequency Scaling）のような電力制御技術を用いて実現されており，こういった機能を持たない「京」では適用は難しい．また，「京」はすでに共用利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．そこで我々は，現在の運用を大きく変えずに電力超過に対処する手段として，電力超過を引き起こす可能性のあるジョブを事前に把握し電力超過を回避する制度の導入と，消費電力が超過または超過が予測される場合には，超過の一因となるジョブを停止し超過を回避するアプローチをとることにした．本手法は，DVFS のようなハードウェアは必要とせず，実運用に比較的容易に導入することができた．

本稿で提案する手法に限らず，消費電力を考慮した運用を行うためには多くの場合，対象システムの消費電力特性を把握する必要がある．電力計を使用することで，システムの消費電力を正確に知ることができるが，ノード単位で

細かな情報を取得するには大量の電力計が必要になる．たとえば「京」の場合，ノード単位の計測のためには 82,944 台の電力計が必要となる．スケジューリング単位（12 ノード単位）で計測する場合でも 6,912 台の電力計が必要となり，そのコストは無視できないものとなる．電力計よりも精度は劣るが，ソフトウェアで消費電力を推定する研究が数多く行われている．これらの研究では，計算機の構成を詳細に分析し，パフォーマンスカウンタや各コンポーネントで消費される電力を精査してモデルを作成し，ジョブの消費電力を推定する [9], [11], [12]．これらの研究は多くても数ノードの規模で行われていて，大規模システムではまだ行われていない．本稿で提案する推定方法は，システムをブラックボックスとして扱い，温度情報だけで推定を行う．「京」に限らず消費電力の推定に必要な温度情報を随時取得できるシステムであれば，システム全体の消費電力を容易にリアルタイムに推定することが可能である．パフォーマンスカウンタ等を利用した場合よりも精度は劣るが，本稿の目的である電力超過を回避するための停止ジョブの選択という目的は十分に果たすことができると考えている．

7. 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の情報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，消費電力の推定精度の向上のほか，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等，実運用への応用についても検討を行っ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] Yamamoto, K., Uno, A., Murai, H., Tsukamoto, T., Shoji, F., Matsui, S., Sekizawa, R., Sueyasu, F., Uchiyama, H., Okamoto, M., Ohgushi, N., Takashina, K., Wakabayashi, D., Taguchi, Y. and Yokokawa, M.: The K computer Operations: Experiences and Statistics, *Proc. 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] Etinski, M., Corbalan, J., Labarta, J. and Valero, M.: Parallel job scheduling for power constrained HPC systems, *Parallel Computing*, Vol.38, Issue 12, pp.615–630 (2012).

[6] Auweter, A., Bode, A., Brehm, M., Brochard, L., Hammer, N., Huber, H., Panda, R., Thomas, F. and Wilde, T.: A Case Study of Energy Aware Scheduling on SuperMUC, *International Supercomputing conference 2014* (*ISC'14*) (2014).

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

[9] Lewis, A., Ghosh, S. and Tzeng, N.: Run-time energy consumption estimation based on workload in server systems, *Proc. 2008 Conference on Power Aware Computing and Systems*, USENIX Association (2008).

[10] 黒田明義，北澤好人，塚本俊之，小山謙太郎，井上　晃，南　一生：スーパーコンピュータ「京」を用いたアプリケーション性能特性と使用電力の相関解析，HPCS2015 (2015).

[11] Joseph, R. and Martonosi, M.: Runtime Power Estimation in HighPerformance Microprocessors, *Proc. 2001 International Symposium on Low Power Electronics and Design*, pp.135–140, ACM (2001).

[12] Witkowski, M., Oleksiak, A., Piontek, T. and Weglarz, J.: Practical power consumption estimation for real life HPC applications, *Future Generation Computer Systems*, Vol.29, No.1, pp.208–217 (2013).

宇野 篤也　（正会員）

2000 年筑波大学大学院工学研究科博士課程修了．博士（工学）．現在，理化学研究所計算科学研究機構運用技術部門システム運転技術チームチームヘッド．「京」の運用およびシステムの高度化に従事．

肥田 元

1989 年株式会社富士通ソーシアルサイエンスラボラトリ入社．HPC 系のサーバ設計・構築を経て，2012 年より「京」の運用保守業務および運用技術支援業務に従事．

井上 文雄

1990 年富士通株式会社入社．HPC システムの構築・運用保守を担当．2013 年理化学研究所計算科学研究機構運用技術部門システム運転技術チームに出向．現在，「京」の運用保守に従事．

池田 直樹

1992 年株式会社富士通ソーシアルサイエンスラボラトリ入社．ソフトウェア開発を経て，2014 年より「京」の運用保守業務に従事．

塚本 俊之

1986 年名古屋大学博士後期課程理学研究科満了．同年富士通株式会社に入社．2010 年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研究員（出向）．2014 年理化学研究所計算科学研究機構運用技術部門施設運転技術チームヘッド．2015 年同部門副部門長．設備最適運転技術の開発に従事．

末安 史親

2012 年九州大学大学院システム情報科学府修士課程修了．同年富士通株式会社入社．2013 年より「京」の運用保守業務に従事．

松下 聡

理化学研究所計算科学研究機構運用技術部門施設運転技術チーム所属．Co-generation，蒸気吸収式冷凍機，ターボ冷凍機を組み合わせた最適化運転技術の開発に従事．

庄司 文由

1998 年金沢大学大学院自然科学研究科満期退学．同年広島大学助手．博士（理学）．2005 年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研究員．2014 年理化学研究所計算科学研究機構運用技術部門・部門長．「京」の運用・高度化に従事．