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　1–12 (Nov. 2015)

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

黒田 明義[1,a)]　北澤 好人[1]　塚本 俊之[1]　小山 謙太郎[2]　井上 晃[3]　南 一生[1]

受付日 2015年4月14日, 採録日 2015年7月22日

概要：近年ハイパフォーマンスコンピューティングにおいて，システム性能の向上にともない，消費電力の増大が大きな課題となってきている．スーパーコンピュータ「京」では，LINPACK 測定時の電力を超える実アプリケーションはないと考えられていた．しかし共用開始以降，アプリケーションの性能チューニングが進んだことで，全系規模の超並列ジョブ実行時に契約電力を超過する事例が見られるようになった．本稿では，アプリケーション実行時の消費電力を CPU や通信，I/O などのアプリケーション固有の性能特性から評価を行った．この評価から消費電力への影響が大きい基礎性能要因は，演算密度ではなくメモリスループットであることが分かった．

キーワード：消費電力，スーパーコンピュータ「京」，アプリケーション性能，メモリスループット

Analysis of the Correlation between Application Performance and Power Consumption on the K Computer

Akiyoshi Kuroda[1,a)]　Yoshito Kitazawa[1]　Toshiyuki Tsukamoto[1]　Kentaro Koyama[2]
Hikaru Inoue[3]　Kazuo Minami[1]

Received: April 14, 2015, Accepted: July 22, 2015

Abstract: In high-performance computing, an increase of power consumption along with the advancement of system performance becomes a major concern. We had expected that there were no applications but the LINPACK benchmark program which used almost all parts of the K computer and exceeded power consumption limits. However, we found the cases that power consumption of some applications exceeded such limits by huge-size job executions occupying the entire system of the K computer due to advances of massively parallel coding. In this study, we evaluated power consumption by associating the characteristics of application performance, such as the efficiencies of floating-point calculation performance, main memory throughput, L2 cache throughput, L1 data cache throughput, and integer calculation performance. From these evaluations, we found that main memory throughput had greater influence on power consumption than floating-point calculations.

Keywords: power consumption, K computer, application performance, memory throughput

1. はじめに

近年ハイパフォーマンスコンピューティングにおいて，システム性能の向上にともない，消費電力の増大が大きな課題となってきている．当初，スーパーコンピュータ「京」（以下「京」と記す）では，全系を用いた LINPACK 測定時の本体電力 12.7 MW [1] を超える一般アプリケーションはないと考えられていた．しかし，2012 年 9 月の共用開始以降，アプリケーションの性能チューニングが進んだことで，ノード単体性能ならびに並列性能が向上し，全系規模のジョブで契約電力を超過する事例が見られるようになっ

[1] 理化学研究所
RIKEN, Kobe, Hyogo 650–0047, Japan
[2] 株式会社富士通システムズ・イースト
Fujitsu Systems East Limited., Nagano 380–0813, Japan
[3] 富士通株式会社
FUJITSU, LTD., Chiba 261–8588, Japan
[a)] kro@riken.jp

た．本件に関して運用の立場から，限られた測定環境の中で電力を見積もる方法が議論され [2]，実行するジョブをコントロールする試みがなされている [3]．ソフトウェアの立場からも，アプリケーションの消費電力解析が急務となった．

Intel では，以前から電力削減 [4] の取り組みの一環として，VTune を用いた解析，PAPI（Performance Application Programming Interface）を用いた RAPL（Running Average Power Limit）カウンタの解析 [5]，PCM（Intel Performance Counter Monitor）によるコア以外を含めた電力解析などが可能となっている．それらを用いて，実行命令性能，浮動小数点演算性能，L2 アクセス，主記憶アクセスのイベントカウンタ情報と消費電力の関係の評価 [6] や，DRAM の状態を解析することによるメモリの消費電力の見積もり [7] などが行われている．これらはカウンタ情報から稼働するモジュールを分析し，モデル式を立てて消費電力を算出している．

本稿では，アプリケーションが消費する電力について，アプリケーションの性能との因果関係に着目して解析を行ったので報告する．2 章では，背景となる事象や「京」上の測定環境について説明し，3 章では，基本ループ集を用いて消費電力を評価したので報告する．4 章では，実際のアプリケーションが消費する電力を見積もるために，実アプリケーションから抜き出したカーネルでの評価を報告する．ここまでは計算ノード内に閉じた消費電力評価であるが，計算ノード間の影響を見積もるべく，通信の影響について 5 章で評価し，I/O の影響については 6 章で評価する．7 章では，ハイパフォーマンスコンピューティングの将来の電力削減を見据えてまとめる．

2. 消費電力測定方法

2.1 研究背景

「京」では常時，システムリソースの稼働率，電力，温度など複数の計測系で監視を行い，システム異常の検知に取り組んでいる．施設全体の平均消費電力は約 15 [MW] で，アイドル状態の「京」本体およびローカルファイルシステム（以下，LFS と記す）の消費電力は約 10 [MW] である．契約電力ならびに 2 台あるコジェネレーション発電システム（以下，CGS と記す）のうち 1 台運転の条件下では，約 4.5 [MW] が計算で許容される電力変動の上限である．

しかしながら，全系を消費する大規模計算実行時に想定電力を超過する事例が見受けられるようになった．**図 1** は，2014 年 6 月に行われた全系を用いた HPCG 測定時の消費電力の時間変化であり，36 時間の受電電力，発電電力をプロットしている．HPCG 測定では，いくつかの問題規模について計算しており，**図 2** からそれぞれの電力変動を抽出することが可能である．このときは，あらかじめ実行時刻情報などが分かっていたため，CGS の吸気冷却運転に

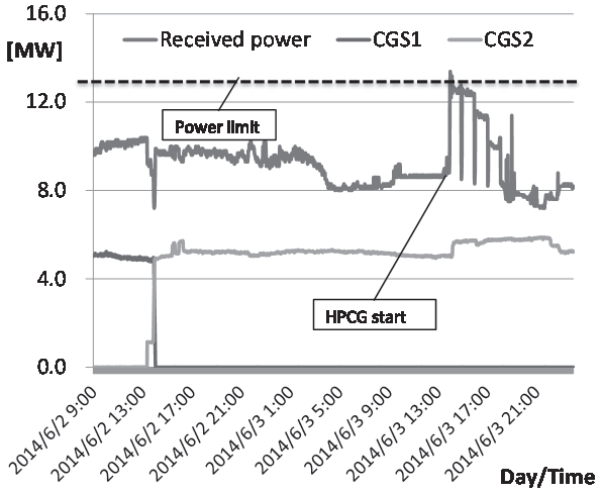


図 1　HPCG 実行時の「京」システムの電力推移
Fig. 1　Power transition at HPCG on the K computer.

より一時的に約 10%出力を高めることで，契約電力の上限値までわずか 40 [kW] を残して対応することができた．

HPCG は構造計算での共役勾配法を用いた性能評価コードである．メモリアクセスが主体であり，理論演算性能比は約 4.1%である．この頃から，本件のような演算器を多く使用しないアプリケーションで電力を多く消費する事例が散見されるようになってきた．

2.2 スーパーコンピュータ「京」

本節では，評価に用いた「京」について紹介する．CPU は富士通社製の SPARC64^{TM} VIIIfx [8], [9], [10] であり，1 CPU 上に 8 コアの演算器を持つ．1 CPU，16 [GB] のメモリ，データ転送を行うインターコネクト用 LSI（ICC: Inter-Connect Controller）によりノードが構成される．システムボードに 4 個のノードを，筐体に 24 枚のシステムボードが搭載されている．この筐体が 864 台設置され，ユーザが利用可能な計算資源は，82,944 ノード，ピーク性能 10.62 [PFLOPS]，メモリ量 1.26 [PB] である．

ノードは ICC を通じて，Tofu インターコネクトと呼ばれる 6 次元メッシュ/トーラスネットワークで結合される [11], [12]．各ノードから 10 本のリンクが出ており，6 本は 3 次元メッシュ/トーラスに，残り 4 本は 12 ノードから構成される 3 次元の基本単位の結合に使用する．各リンクのバンド幅は 5 [GB/s]（双方向）で，システム全体のバイセクションバンド幅は 30 [TB/s] である．

評価に用いたコンパイラやライブラリは，「京」向け言語開発環境であり，測定には K-1.2.0-15 のバージョンを使用した．評価環境は**表 1** のとおりである．

「京」の消費電力は，様々な手法を用いて削減が試みられている．消費電力の大部分は CPU などの CMOS 特性によって決まる．CPU の消費電力は，トランジスタ回路や配線のキャパシタンスに起因するダイナミック電力なら

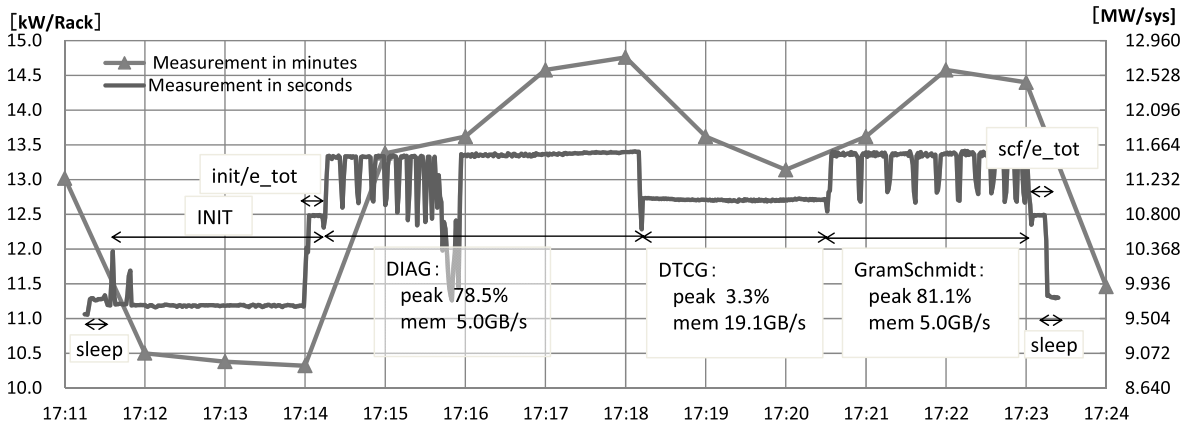


図 2　実アプリケーション RSDFT を用いた電力測定
Fig. 2　Power transition of RSDFT.

表 1「京」を用いた評価システム
Table 1　Evaluation Environment on the K computer.

ハードウェア
SPARC64^{TM} VIIIfx, 2[GHz], 8[core/CPU], 1[CPU/node]
浮動小数点演算性能: 128[GFLOPS/node]
浮動小数点レジスタ: 256[本/core]
キャッシュ: L1 - 32[KB/core] 4[B/F] , L2 – 6[MB/CPU] 2[B/F]
メモリ: 16[GB/CPU], 0.5[B/F]
ネットワーク: 6D mesh / torus network, 5[GB/s]×双方向 (6 方向)
ソフトウェア
OS: Linux (専用 OS)
言語，ライブラリ:「京」向け言語開発環境 バージョン K-1. 2. 0-15（2014 年 4 月～12 月環境） Fortran, MPI, SSL2（BLAS, LAPACK, ScaLAPACK）

びにリーク電流によるものである [13]．動作周波数は 3 乗で電力に影響を及ぼすため [13], [14]，2 [GHz] へと低く抑えることでダイナミック電力を 2.5 [GHz] で設計した場合の約 1/2 に抑えた．また水冷により温度を低く保つことでリーク電流を最大限に削減し [13], [14]，モジュールごとの消費電力の解析から不要なトランジスタを停止するなどの工夫を行った [15]．製造プロセス技術を 1 世代微細化を進めることにより，1/2 の電力削減が可能なため，SPARC64 VII に対しトータルで 1/6 の電力削減を達成した [15]．

2.3　電力測定装置

電力測定は以下 3 系統あり，一体的に管理されている．

①　受電設備に付帯する測定装置

受電設備に付帯する電力測定装置で，施設やシステム全体の消費電力を 60 [s] ごとに取得可能である．電力分解能は，「京」全システムあたりで測定され 10^{-2} [MW] である．2.1 節の HPCG での電力推移はこの測定装置によるものである．

②　一部分のラックに取り付けられた測定装置

消費電力評価のため，96 計算ラックと付帯する 24 I/O ラックに OMRON 社製の KM1-PMU2A，KE1-CTD8E が取り付けられている．装置単位の電力量評価ならびにデータ管理が可能であり，時間分解能は 60 [s]，電力量分解能はラックあたり 10^{-2} [kWh] であり，ラックで消費する電力量が自動的に採取される．

③　1 ラックに取り付けられた測定装置

電力品質を詳細に解析することを目的に，1 ラックのみ富士電機（株）社製の PowerSATELITE [16] を設置した．測定後解析処理が必要であるが，交流の電圧変動を詳細に解析することにより，100 [ms]～1 [s] の時間分解能で電力変動を評価可能である．電力分解能はラックあたり 10^{-3} [W] である．

2.4　実アプリケーションにおける電力測定

実アプリケーションを用いた消費電力の測定例を示す．2011 年に Gordon-Bell 賞を受賞した実空間密度汎関数法（RSDFT）[17] を使用し，計算規模は 13,828 バンド，288 × 288 × 48 メッシュのシリコンナノワイヤの SCF 収束計算 1 回分の計算を 1 ラック 96 ノードで実行した．DGEMM を主とする対角化，Gram-Schmidt 直交化の 2 区間と CG の区間を含み，実行効率約 59.6%の計算である．

測定には時間分解能が 60 [s] と 1 [s] の測定装置 ②，③ を用いて比較した．消費電力の時間変化は図 2 のとおりである．縦軸は RSDFT が動作している時刻におけるラックあたりの電力と全システム換算の電力をプロットしている．本 RSDFT の計算での消費電力は，全システムあたりに換算して 3.94 [MW] 増加した．電力変動を詳しく見ると前半の低い部分は，初期化処理であり，SCF ループでは，DGEMM を使用する対角化，Gram-Schmidt 直交化の箇所で電力が増加するなど処理内容ごとに消費電力が変化することが分かった．

測定器には，数%程度の測定誤差があるとされる．図 2 中の 17:18 の電力を比較すると最大 10%程度の誤差があるが，測定結果は 60 [s] の平均値ととらえると前後の時間の電力変化から定性的には同等の振舞いであると見なした．

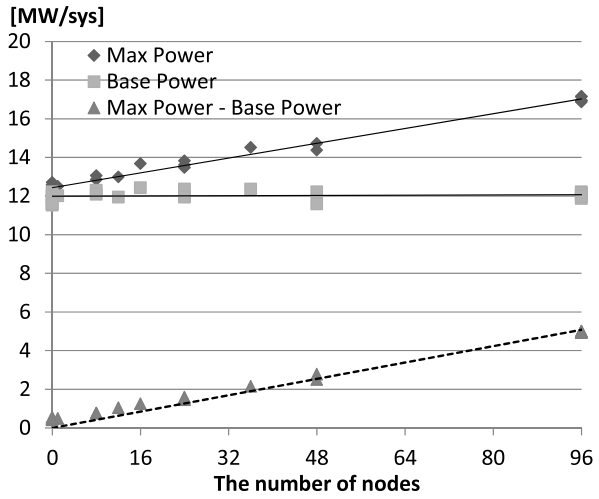


図 3　DGEMM のラック内の電力変動スケーラビリティ
Fig. 3　Scalability of power consumption in the rack for DGEMM.

以下の電力測定では，性能特性ごとに主要な箇所を抜き出して，測定装置 ② を用いて評価を行った．このとき，測定が短時間で終了する区間については，ループ長を変更することで，測定時間が時間分解能より十分大きく 600 [s] 程度になるよう配慮した．

2.5　スケーラビリティ

2.4 節の評価から，当初より消費電力が大きいと思われていた DGEMM を用いた電力のスケーラビリティ測定を行った．アプリケーションから 1 ノード分の行列サイズ (1,280 × 320) × (320 × 320) の DGEMM ループを抜き出し，ノード間は Embarrassingly Parallel とし，時間分解能の制約から，600 [s] 程度の実行時間になるようループ長を調整した．測定は，1 ラック上の 96 ノードまで行っており，ラック単位でのみ電力計測が可能なため，1 ラック占有して一部のノードを使用し計算を行った．測定電力は全システム 864 ラック分（[sys]）に換算した．ノード数と DGEMM の消費電力の関係は**図 3** のとおりである．縦軸は全システム換算の電力量で，横軸はプロセス数である．実線の最大電力と sleep 時のベース電力にラック個体差があることが分かったため，電力の変動量を算出し点線で示した．これにより計算ノード数と電力変動量は線形にスケールすることが分かる．以下の測定では，計測は 1 ラックで行い，システム全体に換算した電力変動量を用いて評価を行う．

3.　基本性能と消費電力の相関分析

3.1　基本ループ集

アプリケーション性能と消費電力の因果関係を調べるために，いくつか基本的な性能に着目してループ集を作成し，測定を行った．以下の 6 種類に分類できる．

① Change of stream
演算性能が高い状態（ピーク比 90%前後）で一定になるよう，積和演算が連続的に動作する構造を保ち，メモリスループットを 0 [GB/s]～39 [GB/s] に変化させる．サンプルコードは以下のとおりである．配列は乱数で初期化し，ストリーム数（配列 a の 2 次元目の個数）を調整することで，メモリスループットを調整する．

```
a(j,1) = ((・・・(((a(j,1)*a(j,2)+a(j,3))
                  *a(j,4)+a(j,5))
                  *a(j,6)+a(j,7))
                        ・・・
                  *a(j,7)+a(j,8))
```

② Change of stride
演算性能が低い状態（ピーク比 4%前後）で一定になるよう，ストライド幅を調整することで，メモリスループットを 0 [GB/s]～46 [GB/s] まで調整する．サンプルコードは以下のとおりである．

```
!$omp parallel private(tn, idx, i, j)
  tn = omp_get_thread_num()
  idx=1.0
  do i=1, cnt
     j = int(idx)
     a(j, tn+1) = b(j, tn+1)
     idx = idx+dx
  enddo
!$omp end parallel
```

③ STREAM Benchmark
記憶領域の性能特性依存性を調べることを目的とし，MPI のプロセス生成，消滅処理をスレッド並列版 STREAM ベンチマークに挿入し，Embarrassingly Parallel のハイブリッド並列化を行った．3 配列のトータルサイズを調整することで，L1 データキャッシュ（以下 L1D と記す），L2 キャッシュ，メモリに乗る 3 パターンのサイズを計測した．

L1D:　512 × 8 [Byte] × 3 配列 = 12 [KiByte]
L2:　24,592 × 8 [Byte] × 3 配列 = 576 [KiByte]
Memory:　4,123,000×8 [Byte]×3 配列＝94 [MiByte]

④ DGEMM
演算性能が高い DGEMM について，以下の 2 種類の行列サイズの行列行列積について計測した．配列は乱数で初期化した．

(18,144 × 6,030) × (6,030 × 6,030)
(20,000 × 20,000) × (20,000 × 20,000)

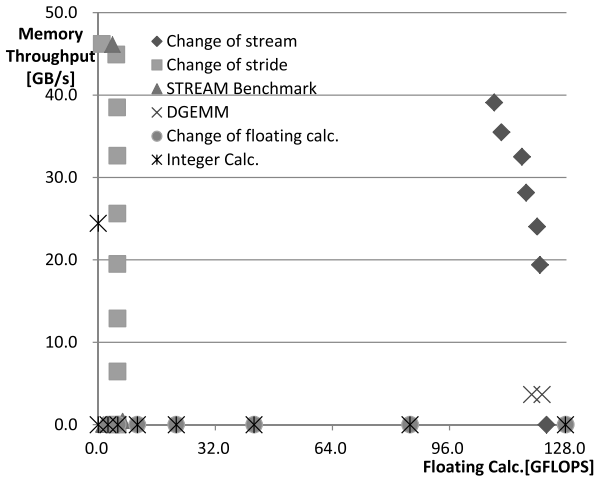


図 4　基本ループの基礎性能

Fig. 4　Basic performance of the fundamental loop.

⑤ Change of Floating Calculation
FMA 命令のみを行う命令列において，アセンブラレベルでレジスタの定義と参照の距離の差を 1 つずつ調整することで浮動小数点演算待ち時間を調整し，ピーク比を変化させた．サンプルコードは以下のとおりである．

```
sxar2
fmaddd, s   %f0,%f0,%f0,%f2
fmaddd, s   %f4,%f4,%f4,%f6
sxar2
fmaddd, s   %f8,%f8,%f8,%f10
fmaddd, s   %f12,%f12,%f12,%f14
sxar2
fmaddd, s   %f2,%f2,%f2,%f0
fmaddd, s   %f6,%f6,%f6,%f4
Sxar2
                . . .
```

⑥ Integer Calculation
整数演算の影響を調べるために，上記 ⑤ に整数演算を加えたコードならびに，UnixBench [18] の 1 つである dhry2reg，Numerical Recipes [19] に含まれる quicksort を計測した．

測定に用いたループ集の性能散布図は**図 4** のとおりである．性能軸として，演算性能とメモリスループットを用いた．基本ループ集は，幅広い性能領域がカバーされており，単一の基礎性能に特徴的なループが多い傾向にある．

3.2　基本ループ集を用いた消費電力

2.5 節のスケーラビリティの結果から，最小測定単位である 1 ラック 96 ノードにて測定し，「京」全体での電力増加量を見積もった．測定は，2.5 節同様，ノード間は Embarrassingly Parallel とし，電力測定の時間分解能から，

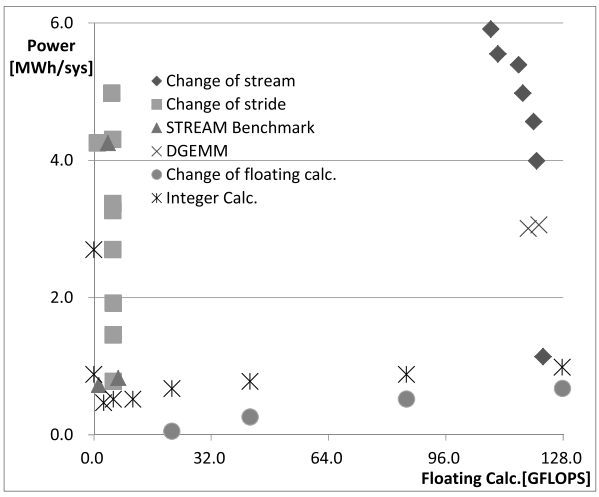


図 5　基本ループの演算性能と消費電力の関係

Fig. 5　Relation of computing performance and power consumption of the fundamental loop.

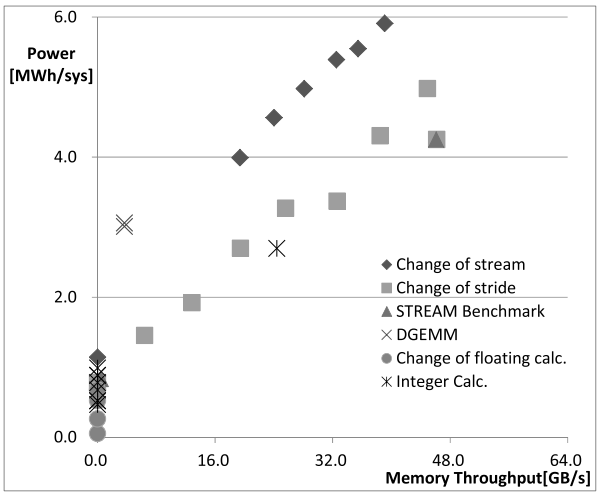


図 6　基本ループのメモリスループットと消費電力の関係

Fig. 6　Relation of memory throughput and power consumption of the fundamental loop.

600 [s] 程度繰り返した．電力増加量と演算性能の関係をプロットしたものが**図 5** である．横軸は演算性能で縦軸は全系換算の電力増加量である．今まで演算性能が高いコードの消費電力が高いと考えられてきたが明確な相関は見られない．

続いて，メモリスループットと電力増加量をプロットしたものが**図 6** である．横軸がメモリスループットで，縦軸が全系換算の電力増加量である．図中で，演算を主とする基本ループ集は縦軸上に乗っており，その他のものは電力増加量と明確な線形相関が見られる．

3.3　基本ループを用いた性能値と消費電力

基本ループの消費電力をいくつかの基礎性能値を用いてフィッティングを行った．この基本ループ集は，メモリスループットを変化させるものが多いため，図 6 からメモリ

スループットと電力の関係が顕著に確認されたが，縦軸上に演算を主とするループ集が集中していることから，少なくとも電力増加には，データアクセスの効果と，演算器の効果が含まれることが推定される．以上から，基礎性能が消費電力に及ぼす影響を調べるために用いる性能評価軸として，ハードウェアカウンタからアプリケーションに特徴的な以下の基礎性能を 5 つ算出して用いた．‘’ で囲まれる項目はハードウェアカウンタ値の差分であり，各基礎性能の影響比率を見るために，概算の実効性能を用いて規格化した．

① 浮動小数点演算性能
演算性能について，浮動小数点演算性能用いて評価した．1 ノードあたりのピーク性能 128 [GFLOPS] で規格化した．

$$(\text{‘浮動小数点命令数’} + \text{‘SIMD 浮動小数点命令数’} \times 2 + \text{‘SIMD FMA 命令数’} \times 4 + \text{‘FMA 命令数’} \times 2) \times 2\,[\text{GHz}]/\text{‘サイクル数’}/128\,[\text{GFLOPS}] \times 100[\%] \quad (1)$$

② 主記憶スループット
L2 と主記憶間のスループットの効果をメモリアクセス数から評価した．1 ノードあたりの理論ピーク性能 0.5 [B/F] に対して，実効ピーク性能を STREAM ベンチマークの結果から 46 [GB/s] = 0.36 [B/F] として規格化した．

$$(\text{‘メモリ読み出し数’} + \text{‘メモリ書き込み数’}) \times 128\,[\text{Byte}] \times 2\,[\text{GHz}]/\text{‘サイクル数’}/46\,[\text{GB/s}] \times 100[\%] \quad (2)$$

③ L2 スループット
L1D と L2 間のスループットの効果をロードストア命令数，キャッシュミス率などから評価した．1 ノードあたりの理論ピーク性能 2.0 [B/F] に対して，実効性能は約 50%とし，128 [GB/s] = 1.0 [B/F] で規格化した．

$$(\text{‘ロード/ストア数’} \times \text{‘L1D ミス率’}) \times 8\,[\text{Byte}] \times 2\,[\text{GHz}]/\text{‘サイクル数’}/128\,[\text{GB/s}] \times 100[\%] \quad (3)$$

④ L1D スループット
レジスタと L1D 間のスループットの効果を見積もるために，ロードストア命令数からデータ移動量を見積もった．1 ノードあたりの理論ピーク性能 4.0 [B/F] に対して，実効性能は約 50%とし，256 [GB/s] = 2.0 [B/F] で規格化した．

$$\text{‘ロード/ストア数’} \times 8\,[\text{Byte}] \times 2\,[\text{GHz}]/\text{‘サイクル数’}/256\,[\text{GB/s}] \times 100[\%] \quad (4)$$

⑤ 整数系演算性能
発行された命令のうち，浮動小数点命令ならびにロードストア命令などを除いた，整数演算やアドレス計算を含むその他の命令を用いて評価した．理論ピーク 32[Ginst./s] = 2 [GHz] × 2 命令 × 8 [core] で規格化した．

$$\text{‘その他の命令数’} \times 2\,[\text{GHz}]/\text{‘サイクル数’}/32\,[\text{Ginst/s}] \times 100[\%] \quad (5)$$

「京」を含めて一般的にノイマン型コンピュータは，CPU 内で命令処理の各段階において電力を消費する．代表的な処理は，命令フェッチ，命令デコード，データロード，演算，レジスタ参照，レジスタ更新，データストアなどである．今回の性能評価軸は，一連の命令処理のうち演算ならびにデータロード/ストアの処理に着目して抽出したことに相当する．データロード/データストアはいうまでもなくメモリへのアクセスであるが，昨今の CPU では通常メモリアクセスの経路に階層構造を持ち，SPARC64$^{\text{TM}}$ VIIIfx も例外ではない．したがってデータロード/データストアはメモリ，キャッシュなどいくつかの階層へのアクセスに分類できる．演算は，演算器を複数備えるスーパースカラであるため，処理よって通過するパイプライン経路は異なる．評価対象の主なアプリケーションの主要な処理は浮動小数点演算であるため，浮動小数点演算について着目したが，整数処理を主とするアプリケーションの電力への影響も考慮して，整数演算についても評価した．

3.1 節で紹介した 6 つの基本ループは，図 4 を見ると演算ならびにデータロード/データストアに関して幅広い性能領域がカバーしている．それぞれのループは，その処理内容の違いから，演算処理の多寡，および，使用するデータ量の多寡，ひいては，データロード/データストア処理の多寡と，その組合せに幅広いバリエーションがあるので，評価の対象として相応しいと判断して採用した．

また，これらの基礎性能評価軸は，互いに直交していないことに注意が必要である．たとえば，メモリアクセスをした場合，データは途中キャッシュを経由して移動する．このためメモリアクセスカウンタから算出されるメモリスループットの値にはキャッシュアクセスの効果が内包される．しかし命令処理フローの観点から分類すると，キャッシュの効果を別に考慮することにより評価軸として従属ではなくなり，問題はないと推定される．つまり，命令処理フローから見たキャッシュアクセスの効果は，アプリ特性から見たメモリアクセスの効果とキャッシュアクセスの効果の合計であると解釈できる．

これらの基礎性能評価軸を用いて，消費電力増加量を評価した．評価に用いた性能寄与を示すパラメータは以下のとおりである．

電力増加量 [MW/sys] = a × 浮動小数点演算性能 [%]

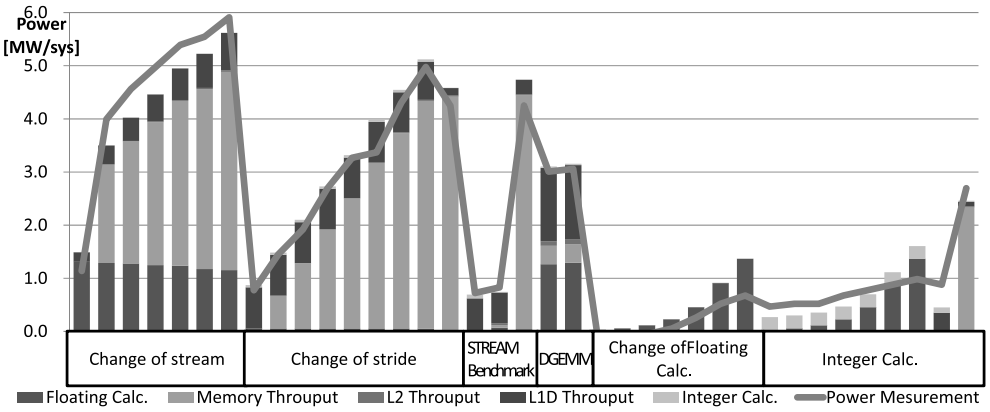


図 7　基本ループにおける消費電力の内訳
Fig. 7　Breakdown of power consumption at the fundamental loops.

$$+b\times\text{主記憶スループット}\,[\%]+c\times\text{L2 スループット}\,[\%]$$
$$+d\times\text{L1D スループット}\,[\%]+e\times\text{整数系演算性能}\,[\%] \tag{6}$$

基本ループ集の消費電力を用いて，これらのパラメータを最小二乗法によりフィッティングすると，

$$a=1.3659,\ b=4.3906,\ c=0.0857,$$
$$d=2.3299,\ e=0.2429\,[\text{MW/sys}] \tag{7}$$

となり，フィッティング平面に対する電力のばらつき具合の残差は 5.24×10^{-2} [MW/sys] であった．

この評価から，消費電力はメモリスループットからの寄与（b）が大きく，演算の効果は，L1D 周辺のアクセス寄与（d）と浮動小数点命令処理の効果（a）の和として評価できることが分かる．また L2 スループットの効果（c）ならびに整数演算の効果（e）は比較的小さい．

残差が係数に比べて 1〜2 桁小さいことから，今回使用した評価軸は全体の傾向を的確にとらえていると見なせる．これは，一連の命令処理の中で回路量や電力消費の観点から，処理に要するコストの大きいものは，データロード，演算，データストアであるためと考えられる．また電力変動の多くは，回路の充放電に起因するダイナミック電力によるものと思われる．リーク電力は無視されているが，残差が小さいことから，係数ならびに小さな残差の中にリーク電力は含まれていると考えられる．

それぞれの基本ループの消費電力と基礎性能の寄与の内訳をプロットしたものを**図 7** に示す．横軸は基本ループで，縦軸は各基礎性能評価軸の寄与を積み上げたグラフと，測定した電力増加量を折れ線でプロットした．定性的にこれらの基礎性能評価軸を用いて電力増加量が説明可能であることが分かる．タイプ ⑤ の基本ループの消費電力が高めに評価される傾向が見られた．これはその他のループ集は，演算時にメモリ周辺へのアクセスも行うため，それらの寄与を演算の寄与としてカウントしているが，タイプ⑤ではレジスタアクセスのみであるため，メモリ周辺の電力が余分に勘定されたためと考えられる．

4. アプリケーションカーネルの消費電力

3 章での基本的なループ構造の電力測定により，基礎性能と消費電力の関係が見積もられたが，実際のアプリケーションでの消費電力を評価すべく，アプリケーションから抜き出したカーネル集を用いて電力を測定した．

4.1 アプリケーションカーネル

測定に用いたアプリケーションカーネルは，各分野のアプリケーションから，主要な処理ブロックを切り出したものである．本評価に用いたカーネルは，有限要素法や有限差分法による流体・弾性体計算．密度汎関数法，分子動力学法，格子 QCD 法，Particle-in-Cell 法などのアプリケーションから抜き出している．「京」では単体チェックスイートとして整備を行い，システムソフトウェアが変わるごとに環境評価に利用しており，ハードウェアやシステムソフトウェアの問題点をスクリーニングすることを目的に作成した．このため切り出しには，シミュレーションで計算する演算特性ごとに数式を単位にして抜き出されており，主な構成単位はループもしくはその集合という，基本的な構成になるよう設計されている．並列性能と単体性能を明確に分離して評価するために，並列化されている箇所は，1 プロセスごとに抽出されている．このアプリケーションカーネルは，コンパイラ最適化の指針などシステムソフトウェア高度化の場でも用いられており，ハードウェアの評価にも利用可能である [20]．またループ単位で切り出しを行っていることから，アプリケーション本体の高度化に向けての試作が容易であり，カーネルを用いた試作過程が再びカーネルに反映され，その試作の結果が実アプリケーションにもフィードバックされている [21]．

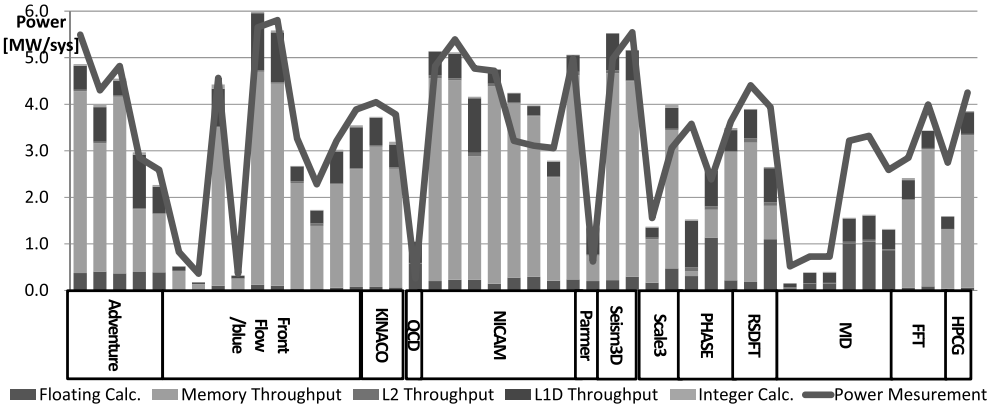


図 8　アプリケーションカーネルにおける消費電力内訳
Fig. 8　Power consumption breakdown of the application kernels.

4.2　アプリケーションカーネルを用いた消費電力

アプリケーションカーネルを用いて測定した電力は**図 8** の折れ線のとおりである．横軸はアプリケーションごとのカーネルを表し，測定した電力増加量を折れ線でプロットした．測定条件は 3 章の基本ループと同様である．基本ループでの測定と同様に，演算効率の高いアプリケーションの消費電力が多いわけではないことが分かる．また有限要素法の流体アプリケーションである FrontFlow/blue について，左から 1，3，5 番目を比較すると，次第に消費電力が増加している．これは流体の有限要素法計算における行列ベクトル積のループについてメモリの利用効率が高くなるようチューニングを施した結果であり，チューニングが進むことで，消費電力が増加した．基本ループでも使用した性能指標をもとにアプリケーションカーネルでも性能依存性を評価したところ，パラメータは，

$$a = 1.5362,\ b = 3.7143,\ c = 0.6075,$$
$$d = 2.1157,\ e = 2.0319\,[\mathrm{MW/sys}] \qquad (8)$$

となった．整数演算の効果が 8.4 倍ほど多めに算出されたが，アプリケーションカーネルの中で主要な整数演算は，間接参照によるアドレス計算であり，基本ループに含まれる陽な整数演算と性質が異なるためと予想される．陽な整数演算を特徴とするアプリケーションカーネルの作成は今後の課題であるが，「京」ではこれらの演算数を別々に測定することはできない．それ以外の効果については定性的に同等の振舞いといえる．フィッティング平面に対する電力のばらつき具合の残差は 8.91×10^{-2} [MW/sys] と若干増えている．

4.3　アプリケーションカーネルでの性能値と消費電力

基本ループで算出したパラメータを用いて，アプリケーションカーネルの電力の内訳をプロットしたものが図 8 の積み上げグラフである．このときのフィッティング平面に対する電力のばらつき具合の残差は，9.68×10^{-2} [MW/sys] であった．基礎ループで評価したパラメータが，ほぼ実際のアプリケーションカーネルでも定量的に評価可能であり，アプリケーションの性能情報から，消費電力が予測可能であるといえる．今回用いたアプリケーションの基礎性能となる評価項目は，ループレベルで解析を行えば，ハードウェアカウンタを使用しなくても Roofline モデルの拡張 [21] を用いて，コードから各基礎性能を見積もり，それぞれの消費電力への影響を算出可能であるといえる．

5.　通信における消費電力

3，4 章では，ノード単体の消費電力について解析を行ったが，ここでは，ノード間の消費電力として，通信にともなう電力の解析を行った．

5.1　隣接通信による消費電力

隣接通信の評価として，1 ラック 96 プロセスにて，3 次元の sendrecv 通信を同時に行った際の電力増加を評価した．

通信転送量とスループット/消費電力の関係は**図 9** のとおりであり，Intel® MPI Benchmarks（IMB）[22] の結果とほぼ同じ傾向である [23]．この 2 つの曲線の振舞いが似ていることから通信のスループットと消費電力の関係をプロットしたものが**図 10** である．横軸は通信のスループットで縦軸は電力増加量であり，消費電力はスループットに比例して増大した．転送長の小さい通信を何度繰り返しても電力への影響は小さく，電力増加量も，CPU の演算やメモリ転送などの影響に比べるとシステムあたり 1 [MW] 程度と少なかった．

5.2　集団通信による消費電力

集団通信と消費電力の関係を評価した．対象とした集団通信は MPI_Alltoall，MPI_Allgather であり，それぞれ，転送長が 4 [B]，1 [KiB]，256 [KiB]，1 [MiB] での通信を 600 [s] 行った．各通信における電力変動をプロセス数ごとにプロットしたものが，**図 11** である．横軸は通信プロセス数

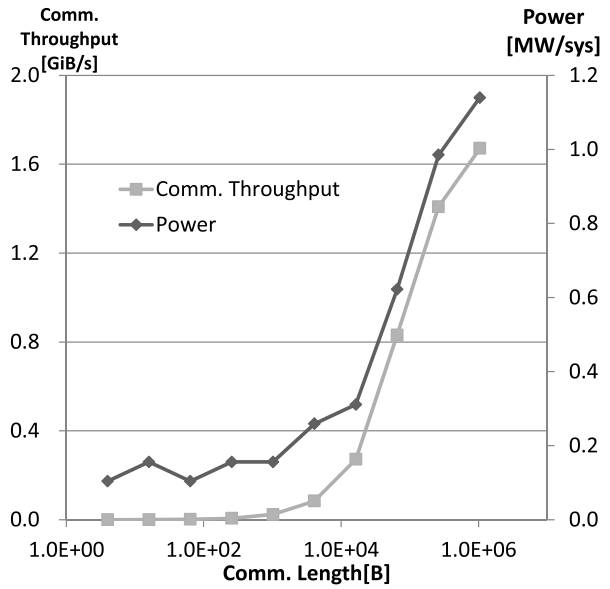


図 9　隣接通信における転送量とスループットの関係

Fig. 9 Relation of communication volume and network throughput/power consumption in neighbor communication.

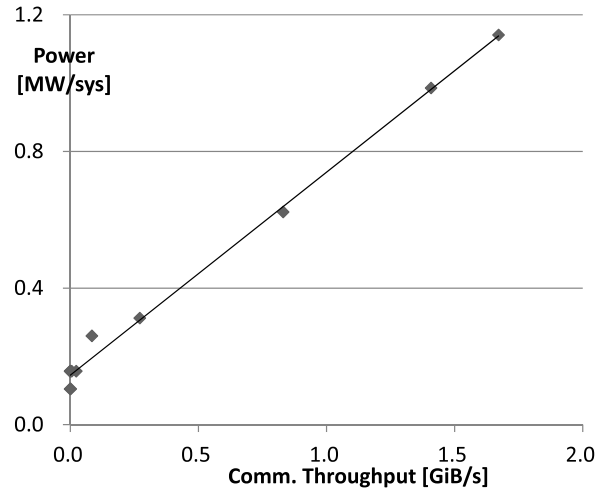


図 10　隣接通信におけるスループットと電力増加量

Fig. 10 Relation of Network throughput and power consumption in neighbor communication.

であり，16 ラック 1,536 プロセスまで変化させて消費電力を測定し，1 ラック 96 プロセスあたりの電力増加量を全システムの電力増加量に換算してプロットした．通信プロセス数が変わることによって，ラックあたりの電力増加量の変動は少なく，全体の電力増加量は通信プロセス数に比例することが分かる．また Allgather のほうが，Alltoall よりも消費電力が多いが，Allgather のアルゴリズムでは転送長が大きいとき，メッセージ分割を行うことで，複数の Tofu ネットワーク・インタフェース（TNI）を同時に使用し，Tofu ネットワークのバンド幅を使い切っている．このため 1 対 1 通信同様，転送スループットが高くなっているものと予想される．通信における転送スループットを計測

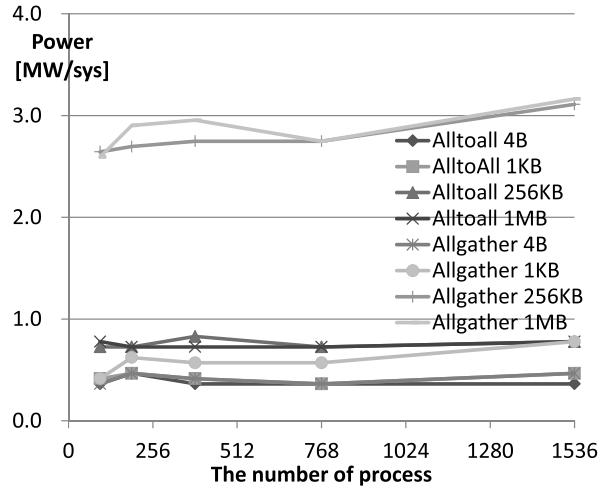


図 11　集団通信におけるプロセス数と電力増加量

Fig. 11 Relation of the number of process and power consumption in collective communication.

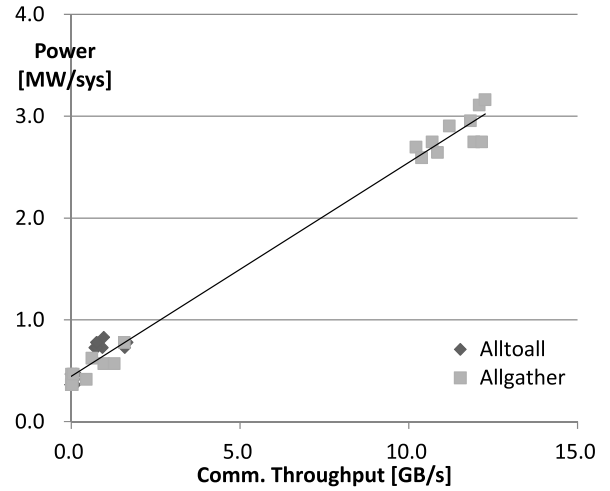


図 12　集団通信におけるスループットと電力増加量

Fig. 12 Relation of network throughput and power consumption in collective communication.

する手段はないため，次の計算式を用いて算出した．

$$\text{転送スループット} = \text{転送長} \times \text{プロセス数/時間} \quad (9)$$

転送スループットと消費電力の関係をプロットしたものが**図 12** である．横軸は通信のスループットで，縦軸はそのときの電力増加量である．1 対 1 通信同様，転送スループットが高い方が消費電力も大きい．

5.3　通信による消費電力

図 10 ならびに図 12 の通信のスループットと電力の関係を見ると一定の相関があることが分かる．これは，通信により，ノード内でメモリアクセスが発生したことによるものと考えられる．実際，図 12 から算出される通信スループットと消費電力の関係は，傾き 0.210 [(MW/sys)/(GB/s)] で線形相関があった．通信に使用されるデータは送受信時にメモリからリード/ライトされるため，式 (7) の係数か

ら送受信にともなうメモリの寄与を見積もると，2b/46〜0.191 [(MW/sys)/(GB/s)] と，ほぼメモリアクセスに起因すると解釈できる．Tofu ネットワークの実効バンド幅は 13 [GB/s] であるため，メモリの実効バンド幅 46 [GB/s] に比べて 1/3 未満である．通信と演算が重ならない限り，演算に比べて電力増加量は少ないと予想される．

また電力増加量はメモリアクセス量を用いてほぼ見積もりが可能であることから，通信にともなう ICC などの消費する電力は，ベース電力の中に含まれることが推定される．これは，通信に関わるモジュールはつねに待機のために回路が動作し続ける必要があるからと考えられる．

6. ファイル I/O における消費電力

6.1 ファイル I/O

計算ノード間の消費電力として，計算ノードからファイル出力を行った場合のディスクラックの電力変化について評価した．計測は I/O グループを占有して，ファイル出力は /dev/zero の 1 [GB] 分のデータ出力を dd コマンドにより 600 [s] 行った．I/O グループとは，4 つの計算ラック，3 つの I/O 系統からなる 1 つの I/O ラックから構成されている．

6.2 ファイル I/O における消費電力

I/O による電力の時間変化は**図 13** のとおりである．横軸は I/O 開始からの時間であり，縦軸は電力増加量である．4 計算ラックと 3 系統の I/O ラックの電力時間変化をそれぞれ合計し全系に換算した．I/O ラックの sleep 時のベース電力は計算ラックの 1/9 ほどであるが，計算ラックの電力変動は，わずかに増加傾向が見られたものの，測定ノイズの範囲であった．I/O ラックでは有意な電力増加が観測された．しかしその増加量は約 0.1 [MW/sys] と，CPU や通信による電力変化と比べて 1 桁〜2 桁少ない．

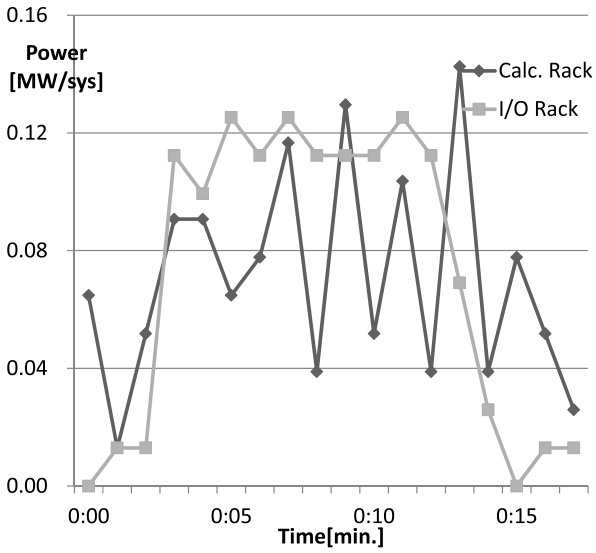


図 13　ファイル出力時の電力変化
Fig. 13　Power consumption during file output.

演算や通信同様，I/O のスループットを算出すると，384 ノードからライトした場合ファイルを書出した回数は 2,318 回 = 2,318 [GB]．実行時間は 1110 [s]．前後の sleep が 180 [s] であるため，3.09 [GB/s] 程度と，実測値から見積もられるローカルファイルシステムの実効性能の上限であると評価できる．ただし今回は，メモリからファイル書き出しではないため，計算ラックの電力変動量は I/O スループットから推定されるメモリアクセスと影響比べて，小さく無視できる．また I/O ラック自体は，ICC 同様，つねに電源が入った状態であり，消費電力とスループットとの因果関係は小さいと見られる．

7. まとめ

今回，「京」上の，アプリケーション性能と消費電力の関係について，基本的なループを用いて電力評価を行った．本稿で用いた解析指標は，CPU のモジュール単位で算出された量ではなく，アプリケーションのコードから見積もりが可能なものを選択した．この結果，アプリケーションの性能と消費電力の間に明確な相関が見られた．アプリケーションのループ特性ごとにノード内の使用する回路が異なるためで，「京」ではメモリアクセスが多いときに消費電力が増加した．

またノードの単体性能以外に，ノード間の性能について評価を行った．これら通信ならびにファイル I/O に関わる消費電力は，大部分がベース電力に含まれ，付随するメモリアクセスによる電力増加以外に大きな変化は見られなかった．通信や I/O により生ずるメモリアクセスは，演算処理を重ねるなどの手続きを行うと，CPU からのメモリアクセスにも影響を与える．また CPU 内でもメモリアクセスなどが律速になることにより，演算器の稼働率などが低下し，演算器寄与の電力が下がることもある．しかし，CPU，通信，ならびに I/O からのメモリアクセス量は，それぞれ独立に測定されており，実際のアプリケーションの消費電力は，これらの効果の単純な重みつきの足し合わせによって算出可能である．

他のアーキテクチャでは，昨今の電力削減の要請から，電力モニタリング方法や電力抑制方法について様々な報告や議論がなされているが，メモリアクセスが消費電力に与える影響の大小に関する議論は見当らない．しかし，これらの報告で使用されている実測データを見ると，Bellosa による Intel® PentiumII™ の電力算出方法に関する報告 [6] で紹介されている実測データでは，フィッティング係数を規格化していないため，算出式から単純に影響の大小を見積もることは難しいが，浮動小数点演算を変化させたときの最大電力が 39 [W] に対し，メモリスループットを変化させたときの最大電力は 44 [W] となっており，メモリによる最大消費電力へ影響が多いことが分かる．また Hong

らによる NVIDIA®の GPU GeForce™ GTX280 を用いた消費電力見積もり手法の報告 [24] で用いられている実測データでは，グラフからメモリアクセスに相当する電力変動の占める割合が大きいことが見てとれる．本来，コンピュータは，その規模や用途によってメモリ容量や性能，コントローラなどが変わり消費電力の内訳も変わると予想される．しかし，異なるアーキテクチャで同じような振舞いが見られることは興味深い．David らの Intel® Xeon™ Processor X5570 Platform での測定 [5] では，アクティブ時の消費電力は，CPU が 35%，Memory が 23%の比率になっている．同じノイマン型コンピュータでは，演算性能とデータ供給のバランスを考えて設計したとき，くしくも同様の内訳となっていることによるものと思われる．

「京」の共用が始まり 2 年が経ち，その間にアプリケーションの性能チューニングが進んだ．特に構造解析や流体解析で用いられるステンシル計算では，メモリ性能を限界まで使うことが性能を引き出すポイントであり，そうしたチューニングを推し進めることが，消費電力増加の顕在化へとつながった．

今後は，今までの性能チューニングとは別の新たな方向を検討すべきである．消費電力を抑えるチューニング，消費電力上限値を設定したうえでの最大性能を引き出す工夫などである．これらは，運用，システムソフト，アプリケーション開発の 3 つの方向で取り組むべき問題である．

「京」の CPU は不必要な電力を削減する工夫がなされている．電力削減を徹底されていないシステムでは，一度使用した回路を切ることはなく，どのアプリケーションでも性能によらず消費電力が変わらない可能性がある．「京」でも，通信や I/O 時の電力変動はほとんど見られず，ラックのベース電力は増加分と比較して，2/3 ほどである．この電力に削減の余地があるか，検討が必要である．

消費電力のうち ICC については通信に関わるモジュールが常時電力を消費するため消費電力の削減は望めない．CPU については，リーク電流は電源オフや供給電圧降下で抑止し，ダイナミック電力はクロックの停止やスローダウンで消費電力削減が可能と考えられるが，全体で半分程度の電力は残る上に，最大電力の抑制にはつながらない．最大電力を抑制するためには，今までの低電力化をさらに推進するだけでなく，GPU，メモリ積層技術に次ぐ新たな技術的なブレークスルーが不可欠になると思われる．

謝辞　本報告に際し，理化学研究所計算科学研究機構運用技術部門の庄司文由氏に有用な意見をいただき，井上文雄氏には測定の協力をいただいた．同機構井上愛一郎氏には回路の節電原理について助言をいただいた．理化学研究所計算科学研究機構に常駐して「京」の運用支援に携われている佐治隆行氏には MPI 通信の電力測定ならびに解析について協力をいただいた．富士電機株式会社の湯谷浩次氏には電力測定機器の調整ならびに詳細データの提供をいただいた．これらの諸氏に感謝しますとともに，理化学研究所計算科学研究機構運用技術部門，富士通株式会社 SE，富士通株式会社次世代テクニカルコンピューティング開発本部の諸氏に感謝します．本稿の結果は，理化学研究所計算科学研究機構が保有するスーパーコンピュータ「京」によるものです．

参考文献

[1] TOP500 List, available from ⟨http://www.top500.org⟩, June 2012 Report.
[2] 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，情報処理学会研究報告，ハイパフォーマンスコンピューティング，2014-HPC-146(4), pp.1–5 (2014-09-25).
[3] 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，情報処理学会研究報告，計算機アーキテクチャ研究会報告，2014-ARC-213(20), pp.1–7 (2014-12-02).
[4] Rotem, E., Naveh, A., Ananthakrishnan, A., Rajwan, D. and Weissmann, E.: Power-Management Architecture of the Intel Microarchitecture Code-Named Sandy Bridge, *IEEE Micro*, Vol.32, No.2, pp.20–27 (2012-3/4).
[5] David, H., Gorbato, E., Hanebutte, U.R., Khanna, R. and Le, C.: RAPL: Memory Power Estimation and Capping, *International symposium on Low power electronics and design* (*ISLPED*), pp.189–194 (2010).
[6] Bellosa, F.: The benefits of event: driven energy accounting in power-sensitive systems, *Proc. 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, September 17–20, 2000, Kolding, Denmark (2000), DOI: 10.1145/566726.566736.
[7] Bircher, W.L. and John, L.K.: Complete system power estimation: A trickle-down approach based on performance events, Performance Analysis of Systems and Software (ISPASS), *IEEE International Symposium*, pp.158–168 (2007).
[8] Maruyama, T.: SPARC64 VIIIfx: Fujitsu's New Generation Octo-core Processor for Peta Scale Computing, *Hot Chips 21* (2009).
[9] Maruyama, T.: SPARC64 VIIIFX: A New-Generation Octocore Processor for Petascale Computing, *IEEE micro*, Vol.30, No.2, pp.30–40 (2010).
[10] SPARC64VIIIfx Extensions, Fujitsu Ltd., architecture manual (2008).
[11] Ajima, Y., Sumimoto, S. and Shimizu, T.: Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers, *IEEE Computer*, pp.36–40 (2009).
[12] Toyoshima, T.: ICC: An interconnect controller for the Tofu interconnect architecture, *Hot Chips 22* (2010).
[13] 井上愛一郎：SPARC64V/VI の高性能，高信頼技術，サイエンティフィック・システム研究会，2006 年度科学技術計算分科会 (2006-10-31).
[14] 井上愛一郎：コンピューティングパワー拡大に伴う技術課題，情報処理学会研究報告，計算機アーキテクチャ研究会（ARC），2007-ARC-173(7), pp.37–42 (2007-05-31).
[15] 川辺幸仁，菅　竜二，山下英男，岡野　廣：次世代スーパーコンピュータ向け SPARC64VIIIfx プロセッサの電力削減手法，FUJITSU, Vol.62, No.5, pp.594–600 (2011-09).
[16] 戸井雅則，高橋竜生，佐藤　進，大田洋充，湯谷浩次，呉為麟：電力品質広域計測解析システム（WAMS）の開発，

電気学会研究会資料．PPR，保護リレーシステム研究会，Vol.2007, No.29, pp.21–24 (2007-09-07).
[17] Hasegawa, Y., Iwata, J., Tsuji, M., Takahashi, D., Oshiyama, A., Minami, K., Boku, T., Shoji, F., Uno, A., Kurokawa, M., Inoue, H., Miyoshi, I. and Yokokawa, M.: First principles calculation of electronic states of a silicon nanowire with 100000 atoms on the K computer, *SC '11 Proc. 2011 International Conference for High Performance Computing Networking Storage and Analysis*, 2011.11.14–17, Washington State Convention Center Seattle WA, ACM (2011).
[18] UnixBench, available from ⟨https://code.google.com/p/byte-unixbench/⟩.
[19] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P.: Numerical Recipes 3rd Edition: The Art of Scientific Computing, Cambridge University Press (2007), ISBN 978-0-521-88068-8.
[20] 2013-14 Annual Report Research Division of AICS, RIKEN AICS, pp.205–214 (2014).
[21] 南　一生，井上俊介，千葉修一，横川三津夫：キャッシュの効果を加えたルーフラインモデルの拡張によるプログラムの性能見積り，情報処理学会研究報告，計算機アーキテクチャ研究会報告，2014-HPC-147(30), pp.1–9 (2014-12-02).
[22] Intel® MPI Benchmarks, available from ⟨https://software.intel.com/en-us/articles/intel-mpi-benchmarks⟩.
[23] 住元真司，川島崇裕，志田直之，岡本高幸，三浦健一，宇野篤也，黒川原佳，庄司文由，横川三津夫：「京」のためのMPI 通信機構の設計，先進的計算基盤システムシンポジウム論文集，2012, pp.237–244 (2012-05-09).
[24] Hong, S. and Kim, H.: An integrated GPU power and performance model, *ACM SIGARCH Computer Architecture News*, Vol.38, No.3, pp.280–289 (2010).

黒田 明義

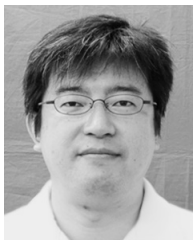
1998 年京都大学大学院人間・環境学研究科博士後期課程修了．専門は統計力学，計算物理学．2006 年から理化学研究所次世代スーパーコンピュータ開発実施本部ならびに計算科学計算機構にて，アプリケーション開発の立場から「京」コンピュータの開発ならびにソフトウェアの高度化に従事．博士（人間・環境学）．

北澤 好人

1990 年信州大学理学部物理学科卒業．2009 年から富士通長野システムエンジニアリング（現，富士通システムズ・イースト）にて，「京」コンピュータのソフトウェア高度化に従事．2014 年から理化学研究所計算科学研究機構に出向し，「京」コンピュータのソフトウェア高度化に従事．

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．



小山 謙太郎

1998 年鳥取大学大学院工学研究科博士前期課程修了．同年（株）富士通長野システムエンジニアリング（現，（株）富士通システムズ・イースト）入社．2012 年より「京」コンピュータのソフトウェアの高度化に従事．

井上 晃

1995 年慶應義塾大学大学院理工学研究科機械工学専攻修士．2006 年から富士通株式会社にて「京」コンピュータの開発に従事し，性能評価の観点からミドルウェアやシミュレーションプログラムの高度化に取り組む．

南 一生　（正会員）

1981 年日本大学理工学部物理学科卒業．同年富士通株式会社入社．2000 年財団法人高度情報科学技術研究機構入社．地球シミュレータ用ソフトウェア性能最適化研究に従事．2008 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グループアプリケーション開発チームリーダー，2012 年理化学研究所計算科学研究機構運用技術部門ソフトウェア技術チームヘッド．2011 年ゴードン・ベル賞受賞．