

## 高密度実装クラスタにおける実効電力制御手法の検討

池田 佳路<sup>†</sup> 近藤 正章<sup>†</sup> 中村 宏<sup>†</sup>

通常の計算機クラスタ構成手法では、最大消費電力が、冷却限界に対する許容電力を超えない HW 構成で設計されてきた。しかし実際には許容電力を消費することはなく、性能の余裕を残した状態で稼働している時間が多い。我々は、許容電力を保証しない構成で実装されたクラスタに対し、動作周波数を変化させることにより許容電力を効率的に消費することで、通常の構成手法によるクラスタよりも高い性能を実現する手法を提案している。本稿ではこれを拡張し、実行前のプロファイルによって最高性能となるノード数を予測し、実行時には実効電力を監視しつつ周波数を動的に変更することで、より高い性能を実現する手法を提案し、性能、消費エネルギーの評価を行った。

### An Adaptive Power Control Method on High Density Cluster System

YOSHIMICHI IKEDA,<sup>†</sup> MASAOKI KONDO<sup>†</sup> and HIROSHI NAKAMURA<sup>†</sup>

Packaging of cluster systems, so far, is designed to satisfy the restriction that is, peak power consumption does not exceed a given TDP (Thermal Design Power) derived from cooling limitation. However, practical power consumption seldom reaches TDP, and thus cluster systems operate with allowance for power consumption most of the time. In our research, we have proposed a new implementation scheme in which theoretical peak power consumption exceeds TDP, but practical power consumption is still below TDP by adjusting supply voltage and clock frequency of each node. In this paper, we extend the proposed technique so that the number of nodes is optimized in addition to changing frequency.

#### 1. はじめに

計算機クラスタシステムでは、ノード数を増加させることで性能を向上させることができるが、設置面積や電源容量などの制約によりその上限が存在する。近年では、それらの制約下でできる限りの高性能化を図るために、プロセッサあたりの性能はそれほど高くないが省電力なプロセッサを高密度に実装し、多数のノードを用いて高い性能を得ることを目的としたクラスタシステムが多く登場している<sup>1)~3)</sup>。

それらのシステムにより、低消費電力プロセッサを高密度実装することで、面積や消費電力効率の良いクラスタシステムを構築できることが明らかとなっているが、高密度実装にも限界がある。計算機システムを構築する上では、単位体積当たりの冷却能力には限界が存在するため、その冷却能力を越えるような実装は不可能なためである。例えば、通常の冷却システムのもとでは、19 インチラックに収納する 1U サイズの筐体では 300W 程度の消費電力が限界と言われている。したがって、クラスタシステムを設計する際には、

冷却システムが許容する発熱量に対応した許容消費電力を上限とし、それを越えないように設計する必要がある。

ここで、プロセッサを含む計算機システムにおいては、放熱面での消費電力の最大値を定めた熱設計消費電力 (TDP: Thermal Design Power) が存在し、一般的にはそのシステムのピークの消費電力に相当する。したがって、冷却システムはこのピークの電力消費にも耐えられるように設計する必要がある。しかし、実際には計算機システムがこのピーク電力を消費することはまれであり、消費電力は実行するアプリケーションに応じて大きく異なることが多い。例えばキャッシュミスが頻発する場合などは、非常に電力消費が少なくなる場合もある<sup>4)</sup>。そのため、ピーク電力が冷却能力の限界を越えないように設計されているシステムでは、通常はその冷却能力を最大限に活用しているわけではない。

そこで我々は、この余剰の冷却能力を活用して、計算機クラスタシステムのさらなる高性能化を目指す「実効電力制御による計算機クラスタ構成手法」を提案している<sup>5)</sup>。本手法は、ピーク電力がシステムの許容電力を越えてしまうような数のプロセッサを搭載する構成を用い、アプリケーションの特性にあわせて実効消費電力制御を行うことで、冷却能力を最大限に活

<sup>†</sup> 東京大学先端科学技術研究センター  
Research Center for Advanced Science and Technology  
(RCAST), The University of Tokyo

用して性能を最適化するものである。システムの電力を動的に監視することで、冷却能力により決定された許容消費電力を越えそうな場合には、動的電源電圧変更手法 (DVS: Dynamic Voltage Scaling) により消費電力を制御しつつ実行する。これまでの評価により、本手法を用いることで、従来のクラスタ構成に比べ高性能を達成できることがわかっている<sup>7)</sup>。

以前の提案手法では、アプリケーションの実行に用いるノード数は固定とし、DVS のみにより実効消費電力の制御を行っていた。しかし、電力制約下で最も高い性能を得るためには、周波数のみでなく、ノード数も実行するアプリケーション毎に最適化することが望ましい。そこで、本稿ではプロファイリングによりノード数も最適化し、従来の手法よりも高い性能を得ることができる計算機クラスタ向け実効電力制御手法について検討する。また、周波数変更のアルゴリズムに関しても、従来手法に比べ、より許容電力内でより高い周波数を選択できるよう改良を行う。さらに、これまでは消費エネルギーの評価をしていなかったため、本稿では消費エネルギーに関しても評価を行う。

## 2. 実効電力に基づくクラスタ構成手法

### 2.1 実効消費電力

多くの計算機システムにおいて、最も電力を消費する部分はプロセッサチップである。プロセッサは実際にプログラムの処理をするチップであるためプログラムの性質に依存して実効消費電力は異なる。例えば、並列処理プログラムにおいて通信と演算の比率を対象に考えると、通信が多い場合にはプロセッサはストールしている時間が長くなり実効消費電力が低下する。逆に通信頻度が少ない場合にはプロセッサは演算処理をずっと行うことができ、結果として実効消費電力が増大することが多い。

ピーク消費電力がシステムの許容電力を越えないようにに設計されているシステムでは、システムの許容電力に対して余裕を残した電力しか実際には消費しない場合が多い。したがって、通常はシステムの冷却能力を十分に活用してプログラムが実行されているわけではないと考えられる。

### 2.2 実効電力に基づくクラスタ構成

従来の構成手法で計算機クラスタを構築する場合は、ピークの消費電力が許容消費電力を超えない範囲のハードウェアを用いて構成されている。すなわち、1U などのある単位ユニットに複数のプロセッサを搭載する場合を想定すると、全プロセッサがピークで動作した場合の消費電力の総計が許容電力内に収まるように構成される。これにより、実効消費電力が許容電力を超えないことがハードウェア的に保証される。

これに対し我々は、ピーク電力がシステムの許容電力を越えてしまうような数のプロセッサを搭載し、ア

プリケーションの特性にあわせて電力制御を行う「実効電力に基づく計算機クラスタ構成手法」を提案している。本手法は、余剰の許容電力 (冷却能力) を活用してさらなる高性能を達成すべく、冷却能力を最大限に活用して性能を最適化するものである。

アプリケーションによっては許容電力を越えてしまう可能性があるため実効電力を管理する必要がある。実効電力が許容電力を越えてしまう場合は、(1) ノード数を縮小して実行する、または (2) DVFS により周波数/電源電圧を低下させて実行する、のどちらか、あるいは両者を組み合わせ、許容電力内で最高の性能が出せる条件で実行するように制御することで効率的な実行が行える。以前の提案手法では、アプリケーションの実行に用いるノード数は固定とし、(2) の DVFS のみにより実効消費電力の制御を行っていた<sup>7)</sup>。しかし、電力制約下で最も高い性能を得るためには、周波数のみでなくノード数も実行するアプリケーションごとに最適化することが望ましい。そこで、次章ではプロファイリングによりノード数も最適化し、従来の手法よりも高い性能を得ることができる計算機クラスタ向け実効電力制御手法を提案する。

## 3. 実効電力に基づく計算機クラスタ構成手法の拡張

### 3.1 概要

アプリケーションに応じて最適なノード数/周波数を決定することを考えた場合、あらかじめプロファイリングなどにより実行時の消費電力を調べた上で、最適な構成を選択する方法と、実行時に動的に消費電力を計測しつつ構成を変更する方法がある。ここで、ノード数の変更を実行時に動的に行うのは難しい、並列処理を行うアプリケーションではデータセットの分割などの効率から、2 のべき乗のノード数でないとい実行できないなどノード数選択の自由度が小さい、あるいは、実行開始時にデータセットが各ノードに分散され、ノード数変更はオーバヘッドが大きいこと、などが理由である。

そこで本稿では、ノード数はプロファイリングにより実行前に最適と考えられる構成を決定し、実行時には消費電力に応じて周波数/電源電圧を動的に変更することで許容電力内で最も高い周波数で動作させ、高性能化を図る手法を提案する。

### 3.2 電力管理手法

本節ではまず、どのようにプログラム実行中の電力を監視しノード数や周波数を制御するかについて説明する。

通常のクラスタシステムの場合、1U などのあるユニット単位で電源が搭載され、また許容される電力などもそのユニット単位で定められるのが普通である。そこで、電源より定期的に消費電力の値を取得し、そ

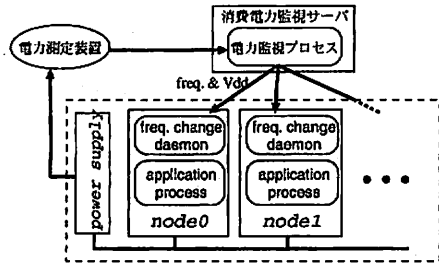


図 1 電力管理システム

の値に応じてノード数の決定や、ユニット内のプロセッサの周波数/電源電圧を制御する。

図 1 に電源管理システムの概要を示す。電力測定装置がユニットの電源の消費電力を測定しつつ、電力値を外部の消費電力サーバに送る。実行時には、消費電力管理サーバ上の管理プロセスが現在の電力を監視しつつ、周波数の変更命令をユニット内の各ノードに対し送信する。各ユニットには、消費電力管理サーバからの指示により自身の周波数/電源電圧を変更するデーモンを起動しておくことで、実際に自身の周波数/電源電圧を変更する。

### 3.3 実行時電力制御アルゴリズム

消費電力管理サーバがユニットの消費電力を監視しつつ、許容電力を超えない範囲で最も高性能となるように、周波数および電源電圧を変更する戦略は重要である。本稿では、できる限り許容電力の範囲内で可能な限り高い周波数を選択することを目的に、以下の電力制御アルゴリズムを用いる。

- (1) 消費電力の上限の閾値 (許容電力に相当) を与える
- (2) 全体の処理時間に対し、上限の閾値を超えても良い時間の割合 (閾値逸脱の許容値) を与え、周波数/電圧を上げるまでの待機時間を決定する
- (3) 各ノードの周波数を最も低い周波数に設定し、アプリケーションの処理を開始する
- (4) 実行中のユニットの消費電力を測定/監視する
- (5) 与えられた上限の閾値を上回る消費電力を観測した場合は、全ノードの周波数/電圧を 1 段階下げる
- (6) 消費電力が上限の閾値を下回った時間が、(2) で決定した待機時間を越えた場合、全ノードの周波数/電圧を 1 段階上げる
- (7) (4) にもどる

上記のアルゴリズムをまとめたものを図 2 に示す。

### 3.4 ノード数決定手法

一般的には、大規模な科学技術計算を計算機クラスター上で実行する場合実装されているノード全てを用いることで並列性が活用でき、多くの場合はそのプラットフォームにおける最高性能が達成できる。しかし、我々の提案するクラスター構成手法では、実効消費電力が許

```

Cwait_time = 100 / Thovershoot
freq = Freq_min;
set_freq_all_nodes(freq);

(involve application)

while(application_is_running) {
  /* for every power measurement cycle */
  W_observed = get_power();

  if (W_observed >= W_max_threshold) {
    freq--;
    set_freq_all_nodes(freq);
    C_safe_time = 0;
  }
  else if (C_safe_time >= Cwait_time) {
    freq++;
    set_freq_all_nodes(freq);
  }
  C_safe_time++;
}

```

図 2 周波数制御アルゴリズム

表 1 評価環境

M/B	Commell LV673 <sup>B)</sup> - Gb Ethernet x 2
Processor	Pentium M 760 (Max 2GHz, FSB533MHz)
Memory	DDR2-SDRAM 1GB

容電力を越えない範囲で構成を変更することを考えるため、最高性能となるノード数と周波数は互いにトレードオフの関係にある。例えば、ノード数を減らしても、その分高い周波数で実行した方が、通信のオーバーヘッドなどを考慮した場合に高い性能が達成できる可能性がある。

そこで、取りうるノード数全てにおいて、前節で述べた実行時電力制御アルゴリズムを適用した状態で実際に各アプリケーションを一定時間実行し、ハードウェアカウンタの値から MIPS 値を算出する。得られた MIPS 値が最も高いノード数が、アプリケーション全体を通して最高性能となる処理台数であると考え、実際に処理に用いるノード数として選択する。なお、このプロファイリング時に実行時電力制御アルゴリズムを適用する理由は、提案手法では実行中に周波数/電圧を動的に変更するため、どの周波数で実行されるかをアプリケーション実行前に予測し、あるノード数での性能を予測することは困難であるためである。

## 4. 実効電力に基づくクラスター構成手法の評価

### 4.1 評価環境

提案する実効電力に基づくクラスター構成手法の性能を評価するため、実際にクラスターシステムを構築し評価を行う。構築したクラスターシステムの各ノードの仕

表 2 周波数/電圧設定と1ノードのピーク電力

Clock (GHz)	2.00	1.86	1.73	1.60	1.46	1.33	1.20	1.06	0.80
Core Vdd (V)	1.356	1.308	1.260	1.228	1.196	1.164	1.132	1.084	0.988
Peak Power (W)	56.6	52.9	50.1	48.1	45.3	43.6	42.0	41.9	39.1

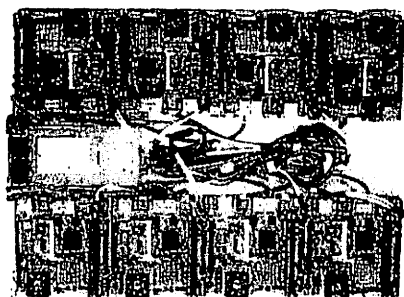


図 3 実験環境

様を表 1 に示す。各ノードには Pentium M 760 プロセッサを用いた PC を使い、これを 16 台 Gb Ethernet で接続しクラスタシステムを構成する。各ノードはディスクを持たず、外部の NFS サーバを用いたディスクレスシステムである。OS は Linux kernel-2.6.11 を使い、cpufreq によるソフトウェア上から周波数/電源電圧の制御、PAPI (Performance API) によるハードウェアカウンタの値取得が可能である。表 2 に設定可能な周波数および電圧のセットを示す。

各ノードのマザーボードのサイズは 17cm 四方であり、8 台を 2 列に並べて配置すると、ほぼ 19 インチラックの 1U のサイズになる (図 3)。ディスクを接続しない場合、マザーボードやメモリを含めたノードあたりのピーク消費電力は 57W 程度であり、8 台分を合計すると約 460W となり、実際に通常の 1U で許容される 300W 程度の消費電力を越えるシステムとなる。

ベンチマークプログラムとしては、High Performance Linpack Benchmark (HPL)、NAS Parallel Benchmark (NPB)、姫野ベンチマーク (Himeno) のプログラムを用いた。消費電力の測定には、(株)シナジェティック製 ST-30000 を用いた。この装置はホール素子、接続 BOX、A/D コンバータから構成されている。本装置は、ホール素子の間に電線を通すことで電流を測定でき、取り扱いが容易であるという特徴を持つ<sup>6)</sup>。評価に用いたボードは 12V の ATX 電源駆動であり、1つの電源の 12V 電線を分岐することで 8 台のボードに電源を供給している。今回はこの 12V の電線の電流を測定することで消費電力を測定した。

なお、本評価環境でのピーク消費電力を求めるために、1 ノードで HPL を処理し、評価環境 1 ノードにおけるピーク消費電力を求めた。文献<sup>4)</sup>においてピーク電力は実験により求められており、最適化された Linpack ベンチマークを実行するとシステムの負荷が限界に近い程度に高くなり、その際の電力がピーク電力

表 3 従来手法において取り得るノード数/最高周波数構成

# of nodes	Possible max. clock (GHz)		
	300W limit	400W limit	500W limit
4	2.00	2.00	2.00
5	2.00	2.00	2.00
6	1.60	2.00	2.00
7	1.20	2.00	2.00
8	N/A	1.60	2.00
9	N/A	1.33	1.86
10	N/A	0.80	1.60
11	N/A	N/A	1.46
12	N/A	N/A	0.80
13 ~	N/A	N/A	N/A

とみなせると述べられている。本 PC システム 1 ノードにて周波数を固定して HPL を処理した場合のピーク消費電力を表 2 に示す。本稿ではこれを本 PC システムのピーク電力と考え、許容電力に対して各ノードで取り得る最大の周波数が決定され、提案手法を用いない場合の通常のクラスタシステムにおける性能を評価することができる。許容消費電力をある値に設定した場合に、各ノード数で取り得る最大の周波数を表 3 に示す。

#### 4.2 評価条件

評価に用いるベンチマークプログラムは HPL、Himeno、および NPB 中のカーネルベンチマークである。また、アルゴリズム中の許容電力の閾値は 300W、400W、500W の 3 通り、閾値逸脱の許容値は 1% の場合について評価を行う。なお、アプリケーション開始 15 秒後から 30 秒間 MIPS 値をパフォーマンスカウンタの値から算出し、MIPS 値が最大となるノード数を選択する。MISP 計測のためのパフォーマンスカウンタの値の取得には PAPI (Performance API) を使い、TOTINS (Total Instructions Completed) の値から算出した。通常の計算機クラスタとの比較を行うため、アプリケーションごとに表 3 に示す各構成において、実行中の周波数を固定して処理した場合の最高性能を比較対象とする。なお、EP 以外の NPB プログラムについては、問題の分割の制約から表 3 の構成のうち、2 のべき乗のノード数で処理した場合に最も高性能となったものが比較対象となる。なお、従来の提案手法であるノード数一定の場合との比較も行う。

さらに、3.4 節のノード数決定手法が、実際に最適なノード数を正しく決定できるかについても評価するために、3.3 節のアルゴリズムを適用し、プロファイリングのためのアプリケーション開始 15 秒後から何秒間実行するか ( $t$  と表す) を変化させ評価を行う。評価では、 $t$  の値は 1 秒、5 秒、10 秒、30 秒について評

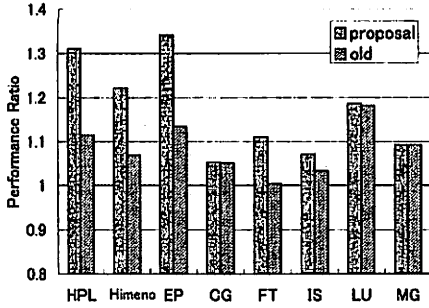


図4 文献<sup>7)</sup>と本稿の提案手法の性能比較

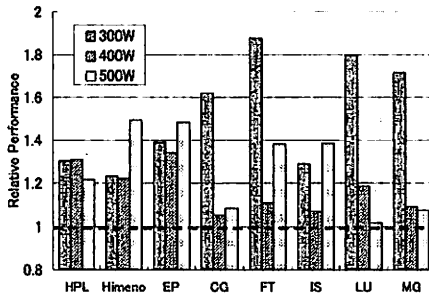


図5 通常のシステムに対する提案手法の性能比

価した。

なお、3.3節のアルゴリズムでは、ある一定周期で電力を計測し周波数変更の判断を行うため、この周期を決定する必要がある。今回実装した電力管理手法において、電力監視ノードが周波数変更命令を送信してからノードの周波数が変更された後、次の周期の平均電力が十分低くなるまでの時間を計測したところ、16ms程度必要であることがわかった。そこで、消費電力監視サーバが対象ユニットの電力を監視する周期を20msに設定して評価を行う。

## 5. 動的実効電力制御手法の評価

### 5.1 評価結果

我々の提案する実効電力に基づくクラスタ構成手法に、3.3節で提案したアルゴリズムを適用して評価した場合の結果を図4に示す。図は、通常の計算機クラスタでの性能として、表3に示す構成のうち最も高性能なものを1とした場合の、提案手法における性能比を表している。図中“proposed”は、本稿で提案したノード数をプロファイリングにより最適化する手法の結果を表し、“old”は文献<sup>7)</sup>で提案したノード数は固定(8台)で周波数制御のみを行う手法を表す。なお、許容電力を400Wとした場合の結果である。

結果から、proposedは評価に用いた全てのベンチ

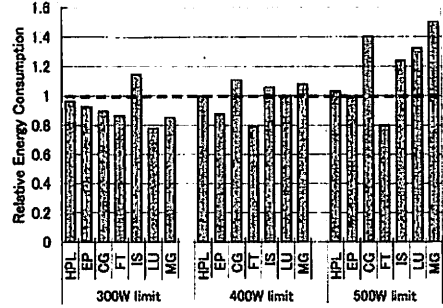


図6 通常のシステムに対する提案手法の消費エネルギー比

マークにおいて、通常の計算機クラスタ構成に比べ性能が向上していることがわかる。平均では、17%の性能向上を達成している。

また、従来の提案手法であるoldと比較した場合、ほぼ全ての場合で同じか、それ以上の性能を達成可能であることがわかる。これは、周波数変更による電力制御アルゴリズムが改良されていることと、周波数だけでなくノード数も許容電力にあわせて変更させることで、より高性能が得られることを示している。実際には、許容電力400WにおけるHPL、Himeno、EPではノード数も最適化することで、8台よりも多いノード数で処理できるようになり、大きく性能向上している。その他の場合では、ノード数はoldの場合と同じく8台で実行しており、アルゴリズムの改善が性能向上に寄与している。

次に、許容電力を300W、400W、500Wと変えてみた場合の結果を図5に示す。図5は、本稿での提案手法(proposed)のみの結果を示している。図より、400Wに設定した場合にEP以外のNPBで性能向上が小さい傾向にある。これは通常のクラスタの場合の構成が8ノード/1.60GHzであるのに対し、提案手法を適用した場合でも問題の分割の制約から2のべき乗のノード数でしか処理できず結果として8ノードが選ばれ、また周波数も2.00GHzとそれ以上の周波数の構成を取ることができず、提案手法を用いても許容電力を十分に活用できなかったためである。

また、多くのアプリケーションにおいて300Wの場合に性能向上が大きくなる傾向にある。これは、許容電力が低い場合、従来型の構成ではかなり低い周波数が選択される一方、提案手法は比較的高い周波数で動作可能であり、性能向上の機会が大きかったことが理由である。

### 5.2 消費エネルギー評価

各アプリケーションの処理に費やす消費エネルギーについて、従来手法を1とした場合の提案手法適用時の結果を図6に示す。提案手法は、性能向上を目的としたものであるにもかかわらず、多くのアプリケーションにおいて許容電力300W、400Wの場合では、

表 4 最高性能となるノード数の予測結果

benchmark	EP					Himeno					HPL				
	300		400		500	300		400		500	300		400		500
t=1	8	10	-0.7%	15	-7.3%	8	9	-3.7%	15	-6.1%	8	9	-3.6%	12	-1.3%
t=5	8	10	-0.7%	13	-2.0%	8	12	0%	12	0%	8	9	-3.6%	12	-1.3%
t=10	8	10	-0.7%	14	0%	8	12	0%	12	0%	8	10	0%	14	0%
t=30	8	11	0%	14	0%	8	12	0%	12	0%	8	10	0%	14	0%
max	8	11	-	14	-	8	12	-	12	-	8	10	-	14	-

消費エネルギーは通常のクラスタと比較してほぼ同じか、場合によってはより削減できていることがわかる。これは、性能向上により実行時間が短くなったことが原因である。

なお、許容電力 500W の場合に平均の消費エネルギーが大きく増加しているものが多い。これは、提案手法による処理時間減少よりも消費電力増加の効果が大きいことが原因である。500W の場合では、通常のクラスタシステムでは 8 台が実行するところを、提案手法により 16 台で実行する場合も多く、消費電力が著しく増加してしまうものが多い。

### 5.3 ノード数決定手法の評価

本稿で提案する手法によって選ばれたノード数と、実際の処理で最高性能が得られたノード数を表 4 に示す。表は MIPS 値取得時間  $t$  を 1 秒, 5 秒, 10 秒, 30 秒と変化させ、MIPS 値が最高となったノード数、つまりその際にプロファイルにより選択されるノード数を示している。また、最下列は各種実験の結果、実際に最高性能を達成したノード数を示している。最高性能の場合と違うノード数が選ばれた場合に関しては、そのノード数で実行した場合の、最高性能に対する性能低下を示している。この結果より、 $t=30$  秒の場合は最高性能となるノード数を正しく予測できていることが分かる。プロファイル時間  $t$  が短いものにおいては、各アプリケーションでの 1 イテレーションに満たない時間でプロファイリングが行われているため、最高性能となるノード数を正確に選択できないことが多い。ただし、 $t=10$  秒では最高性能の場合に対し性能低下はほとんどないため、10 秒程度のプロファイリングを行えば比較的正しく、最高性能のノード数を選択可能であると考えられる。

## 6. まとめと今後の課題

本稿では、システム全体のピーク電力が、システムの冷却能力の限界に対応する発熱量となる許容電力を越えてしまうような数のプロセッサを搭載する構成を利用し、アプリケーションの特性にあわせて消費電力制御を行うことで、冷却能力を最大限に活用して性能を最適化する「実効電力制御による計算機クラスタ構成手法」に対し、ノード数もプロファイリングにより最適化する手法を提案した。提案手法を実際のクラス

タシステムで評価した。

評価の結果、提案するクラスタ構成手法を用いることで、多くのアプリケーションにおいて、従来型のクラスタシステムと比較して、同程度の消費エネルギーで、より高い性能を達成できることがわかった。

性能においては、本稿で用いた全てのアプリケーションにおいて性能向上を達成したが、向上の程度にはアプリケーションの性質によって差異があり、従来手法と比較して消費エネルギーが増加する場合がある。消費エネルギー削減を求める場合、アプリケーションの性質によって電力管理の手法を再検討する必要がある。これは今後の課題である。

謝辞 本研究の一部は、科学技術振興機構・戦略的創造研究推進事業 (CREST) の研究プロジェクト「低電力化とモデリング技術によるメガスケールコンピューティング」、および文部科学省科学研究費補助金 (若手研究 (B) 17700049)、東レ科学振興会科学研究助成の支援によって行われた。

## 参考文献

- 1) M. Warren et al. "High Density Computing: A 240-Node Beowulf in One Cubic Meter", In *Proc. Supercomputing 2002*, Nov. 2002.
- 2) H. Nakashima et al. "MegaProto:1TFlops/10kW Rack Is Feasible Even with Only Commodity Technology," In *Proc. Supercomputing 2005*, Nov. 2005.
- 3) IBM and Lawrence Livermore National Laboratory, "An Overview of the BlueGene/L Supercomputer", In *Proc. Supercomputing 2002*, Nov. 2002.
- 4) J. Rubio et al. "Dynamic Processor Overclocking for Improving Performance of Power-Constrained Systems" IBM Research Report RC23666(W0507-124), July, 2005.
- 5) R. Ge et al. "Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters," In *Proc. Supercomputing 2005*, Nov. 2005.
- 6) 堀田義彦 他, "プロセッサの消費電力測定と低消費電力プロセッサによるクラスタの検討", 情報処理学会論文誌, Vol.45, No.SIG11(ACS7), pp.207-218, 2004.
- 7) 池田佳路, 近藤正章, 中村宏, "実効電力制御による高性能計算機クラスタ構成手法の検討", 先進的計算基盤システムシンポジウム SACSYS2006, pp.487-494, 2006.
- 8) <http://www.comell.com.tw/Product/SBC/LV-673.htm>