実効電力制御による高性能計算機クラスタ構成手法の提案

池田佳路[†] 近藤正章[†] 中村宏[†]

計算機クラスタはその利点であるコストパフォーマンスを向上させるために高性能なプロセッサを高密度に実装することが不可欠である．従来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（TDP: Thermal Design Power）を超えない HW 構成で設計されてきた．しかし実際には TDP を消費することなく，性能の余裕を残した状態で稼動している時間が多い．本研究では，TDP 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，TDP 内で動作する最も性能の高い構成を選択し，従来の構成手法によるクラスタよりも高い性能を実現する計算機クラスタ実装方法を提案し，その初期検討を行った．

A High Performance Cluster System Design with Adaptive Power Control

Yoshimichi Ikeda,[†] Masaaki Kondo[†] and Hiroshi Nakamura[†]

Compact and dense packaging is indispensable for cluster systems to achieve high performance/cost ratio. 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 reach TDP, and thus cluster systems operate with allowance for power consumption most of the time. Therefore, in this paper, we propose a new implementation scheme of cluster systems. In the proposed scheme, although theoretical peak power consumption exceeds TDP, practical power consumption is still below TDP by adjusting supply voltage and clock frequency of each node, and thus effective performance gets higher than that in conventional implementation schemes.

1. はじめに

近年，大規模学術計算の必要性が日々増大している．計算機の処理能力の向上により，実験が困難な事象に対しても計算機シミュレーションによる解析が可能になったためである．たとえば，莫大な情報量を処理する遺伝子解析や，きわめて複雑な構造を有するタンパク質構造解析などの生物・化学分野，また素粒子物理学，気候，環境のシミュレーション，さらには産業分野などでも計算機による解析が用いられている．これらの分野では，今後もさらなる計算機の処理能力向上の需要が増加していくと考えられる．

大規模科学技術計算分野において重要な位置を占めるのが計算機クラスタである．計算機クラスタは，複数台の汎用計算機をネットワークによって接続し，並列処理を行うことで，スーパコンピュータ並みの性能を実現するシステムである．スーパコンピュータは専用のハードウェアを開発することで高性能を得られる反面，高コストである．一方，計算機クラスタは比較的安価な汎用品を用いることで，高いコストパフォーマンスを実現することができるため，現在では広く用いられている．TOP500[1] によれば，世界最高性能 500 台の計算機システムのうち，360 台が計算機クラスタにより構成されており，この点からも計算機クラスタの重要性がうかがえる．

並列処理を行う計算機クラスタシステムは，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．しかし，設置面積や電源などの制限から，必然的にノード数の上限は決定される．そこで，高性能ではあるが消費電力の高いプロセッサを搭載するノードを複数用いるよりも，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多数用い，高密度に実装することで台数効果により性能向上を図る方が，高性能なシステムを構築できる場合が多い．この点に着目し，“Green Destiny” [2] や “Mega Proto” [3] は，実際に高密度に実装したクラスタシステムを構築し，既存のクラスタシステムと比較して，設置面積（体積）

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

あたりの性能や電力あたりの性能が優れたクラスタシステムであることを示している．また，IBM で開発された BlueGene/L[4] は，PowerPC をベースとしたプロセッサを超高密度に実装し，非常に高い性能を省スペースで実現している．

このように，低消費電力プロセッサの高密度実装により，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示されているが，高密度実装にも限界がある．計算機システムを構築するうえでは，単位体積あたりの冷却能力には限界が存在するため，その冷却能力を超えるような実装は不可能なためである．たとえば，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 1 U サイズの筐体では 300 W 程度の消費電力が限界といわれている．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，それを超えないように設計する必要がある．

ここで，プロセッサを含む半導体チップには，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，一般的にはそのチップのピークの消費電力に相当する．半導体チップで構成される計算機システムは，この TDP を基に冷却システムが設計されており，したがって，システムはピークの電力消費にも耐えられるように設計されていることになる．しかし，実際に製造されたプロセッサチップが，このピーク電力を消費することはまれである．また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り，たと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[5]．そのため，ピーク電力が冷却能力の限界を超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．ここで，発熱量によりノード数が制限され，ひいては性能が制限される高密度実装クラスタシステムを考えると，この余剰の冷却能力を活用できれば，さらに高性能を達成できる可能性がある．

そこで本稿では，ピーク電力がシステムの許容電力を超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」を提案する．本提案手法は，システムの電力を動的に監視し，冷却能力により決定された許容消費電力を超えそうな場合には，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，実行させるものである．本稿では，提案手法を適用したプロトタイプのクラスタシステムを構築し，従来のクラスタ構成に比べ高性能が達成できることを示す．

本稿の構成は以下のとおりである．次章では，提案するクラスタ構成手法について述べ，3 章でその有効性に関して議論する．4 章では電源管理のアルゴリズムを述べ，5 章でその評価結果を示す．6 章で関連研究を述べ，7 章でまとめと今後の課題について述べる．

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．たとえば，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では，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．例として，Intel Pentium M を搭載した PC を 8 台用いたクラスタシステムにおいて，ベクトル積を並列に計算するプログラムを，プログラム全体の演算量と通信量を変えずに，通信回数のみを変化させた場合の消費電力を**図 1** に示す．図の横軸は，1 回の浮動小数点演算あたりの通信回数を通信頻度と定義し，示している．なお，実験環境の詳細については，3.1 節で述べる．また，本 PC システム 1 ノードで HPL を実行した際の消費電力は約 57 W であった．文献 17) 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と見なせると述べられている．よって，本 PC システムのピー

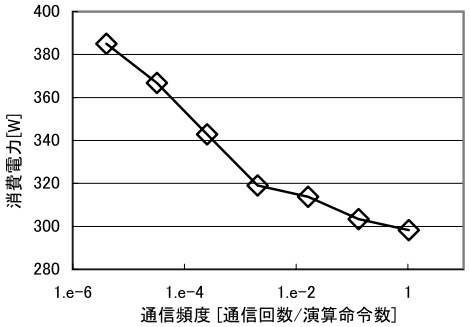

図 1 通信頻度に対する消費電力の違い
Fig. 1 Power consumption for various transfer rate.

ク消費電力は 57 W 程度であると考えられる．

図より，通信頻度に依存して，プログラム実行時の消費電力が大きく変化していることが分かる．通信があまりない状況では 8 台で 400 W 程度であり，1 台あたり 50 W と比較的高い消費電力となっている．一方通信の頻度が高く，通信にかかる時間が多い場合では，消費電力が 300 W 程度にまで低下し，1 台あたりでは 37 W 程度まで低下する．

上記の結果より，ピーク消費電力がシステムの許容電力を超えないように設計されているシステムにおいては，システムの許容電力に対して，余裕を残した電力しか実効的には消費しない場合が多いということが分かる．特に大規模科学技術計算のアプリケーションはデータセットが大きく，キャッシュミス率が高いプログラムが多く存在する．したがって，通常はその許容される電力，すなわち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でのハードウェアを用いて構成されている．すなわち，1 U 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．

これに対し本稿では，ピーク電力がシステムの許容電力を超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」を提案する．本手法は，余剰の許容電力（冷却能力）を活用し，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．

例として，前節で用いた Pentium M を搭載する PC システムを複数台用いて計算機クラスタシステムを実装する場合を考える．許容電力が 300 W であるユニットを仮定し，そのユニットに PC を複数台実装しようとする場合，ピークの消費電力が 57 W の PC であれば，5 台の PC を実装するのが限度となる．しかし，実効電力に基づくクラスタ構成では，当該 PC をたとえば 8 台実装する．当然，各 PC がピークに近い電力を消費すると許容電力を超えてしまうが，前節の図 1 の結果より，通信頻度が高い場合では，アプリケーションによっては 8 台がすべて動作した場合でも，実効消費電力が許容電力内に収まる場合が存在する．この場合，8 台で並列処理ができれば，6 台で実行する場合に比べ高性能が期待できる．

ただし，このようにアプリケーションによっては許容電力を超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超えてしまう場合は，(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．

なお，消費電力ではなく，実行中の温度に制約を与え，直接温度を測定しつつ温度が制約を超えそうな場合には，上記の (1) や (2) により実効電力を管理し，温度を制御する手法も考えられる．しかし，この場合には，システム中のコンポーネントすべてが安全な温度で動作する必要があるが，ある局所領域だけ温度が高くなるホットスポットが生じると，システムの信頼性が低下してしまう．したがって，温度制約を満たすよう実効電力を管理する場合には，システム内でホットスポットとなる可能性のあるすべての部分を測定する必要があり，実装が大変となる．一方，システムの許容電力は，冷却能力を考慮し，ホットスポットとなりうる部分の温度がある程度以上にならないよう決定されることが通常であり，許容電力を満たすように実効電力を管理すればつねにシステムが安全に動作することが保証される．この場合，実効電力はシステムに供給される電源を測定するだけでよく，実装が簡単になる．したがって，本稿では以降，実効電力の値を基にノード数や周波数/電源電圧を制御する手法を検討する．

3. 実効電力に基づくクラスタ構成手法の有効性

3.1 評価環境

提案する実効電力に基づくクラスタ構成手法の性能を評価するため，実際にクラスタシステムを構築し評価を行う．構築したクラスタシステムの各ノードの仕様を**表 1** に示す．各ノードには Pentium M 760 プロセッサを用いた PC を用い，これを 8 台 Gb Ethernet

表 1 評価環境
Table 1 Specification of a node.

M/B	Commell LV673 [18)] — Gb Ethernet × 2
Processor	Pentium M 760 (Max 2 GHz, FSB533 MHz)
Memory	DDR2-SDRAM 1 GB

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 を用い，cpufreq によりソフトウェア上から周波数・電源電圧が制御可能である．**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，17 cm 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 1 U のサイズになる（**図 2**）．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 57 W 程度であり，8 台分を合計すると約 460 W となり，実際に通常の 1 U で許容される 300 W 程度の消費電力を超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 の中から EP（NPB/EP），姫野ベンチ（Himeno-Bench）のプログラムを用いた．消費電力の測定には，（株）シナジェティック社製 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，取扱いが容易であるという特徴を持つ[5]．評価に用いたボードは，12 V の ATX 電源駆動であり，1 つの電源の 12 V 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 12 V の電線の電流を測定することで，消費電力を測定した．

表 2 Pentium M 760 プロセッサの周波数と電源電圧の関係
Table 2 Clock and voltage setting for Pentium M 760.

Clock (GHz)	Core Vdd (V)
2.00	1.356
1.86	1.308
1.73	1.260
1.60	1.228
1.46	1.196
1.33	1.164
1.20	1.132
1.06	1.084
0.80	0.988

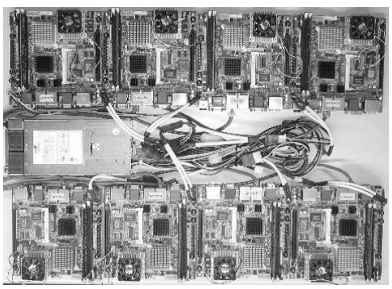

図 2 実験環境
Fig. 2 Experimental platform.

3.2 評価結果

実際に提案するクラスタ構成手法を用いた場合に，どの程度性能向上の可能性があるかを調べるために，HPL，NPB/EP，Himeno-Bench において，ノード数を 5〜8 台，周波数を表 2 の全通りで変えつつベンチマーク実行時の性能と電力を測定した．なお，処理に用いるノード数，クロック周波数は，アプリケーションの開始から終了まで固定とした．

結果を**図 3** に示す．図は，横軸に消費電力を，縦軸に性能をとり，あるノード数，およびある周波数で実行した場合の消費電力/性能の関係をプロットした

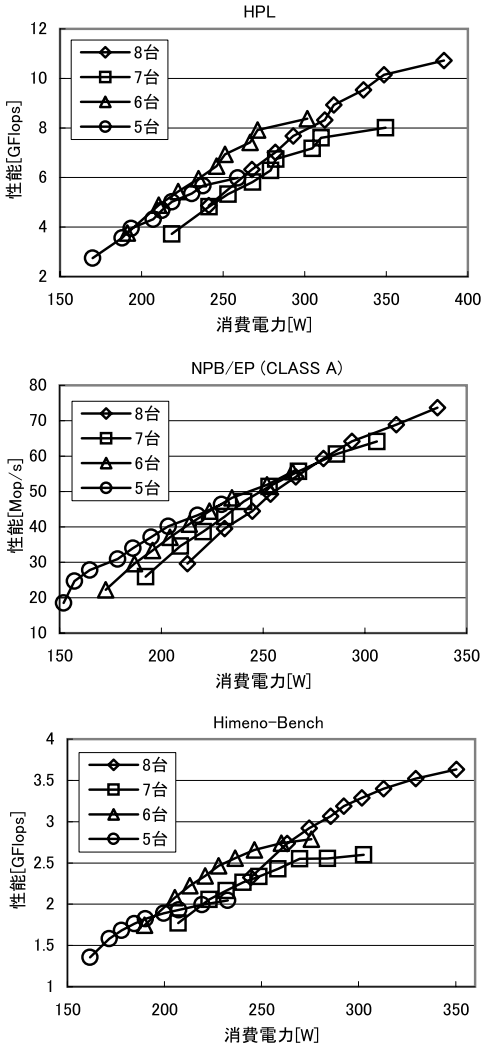

図 3 各ベンチマークにおける消費電力に対する性能
Fig. 3 Relationship between performance and power.

表 3 許容電力を 300 W とした場合の性能
Table 3 Performance on 300 W power limit case.

ベンチマーク	HPL	NPB/EP	Himeno-Bench
提案手法	8.37 [GFlops]	64.2 [Mop/s]	3.28 [GFlops]
従来手法	5.98	46.4	2.04
性能比	1.40	1.38	1.60

のもである．なお，同じ台数の場合が線で結ばれている．図 3 より，許容電力を 300 W と仮定すると，従来手法では各 PC がピークで動作した場合にも 300 W を超えないことが保証されるノード数が 5 台の構成を採用せざるをえない．したがって最高性能はノード数 5 台，周波数 2 GHz の場合となる．一方，提案手法では，消費電力 300 W 以下で最も性能の高いノード数および，周波数/電源電圧の組で各プログラムを実行する．図 3 を見ると，HPL では 5 台の 2 GHz，NPB/EP では 8 台の 1.7 GHz，Himeno-Bench では 8 台の 1.6 GHz で実行するのが，最も性能の高い構成となっている．従来型，および提案するクラスタ構成手法での最高性能を，**表 3** に示す．

表 3 より，提案手法により，従来型に比べ 1.38 倍から 1.60 倍の高い性能が得られる可能性があることが分かる．このことから，実効電力に基づくクラスタ構成手法は，クラスタの高性能化において非常に有効な手法であると考えられる．ただし，提案する構成手法で実際に高い性能を得るには，アプリケーションごとに最適なノード数，および周波数を設定する必要がある．このためには，プログラムのコンパイル時にプロファイリングなどを行い，ノード数や周波数の違いによる電力を予測したうえで実行する，などの処理が必要である．

4. 動的電力モニタリングによる実効電力制御

4.1 概　　要

2.2 節において，実効消費電力に基づくクラスタ構成手法を述べ，従来型のクラスタに比べ高い性能を得られる可能性があることを 3.2 節の評価で示した．ただし，本手法は，アプリケーションに応じて最適なノード数/周波数を決定しなければならないため，プロファイリングなどにより，あらかじめプログラム実行時の消費電力を予測したうえで最適な構成を選択し，実行する必要がある．

しかし，並列処理を行うアプリケーションでは，問題の分割などの効率から，2 のべき乗のノード数でないと実行できないなど，ノード数の選択肢にはあまり自由度がないことが多い．また，消費電力の予測がはずれた場合に，実行時に動的にノード数を変更し，実効消費電力を制御することは一般的には難しいと考えられる．

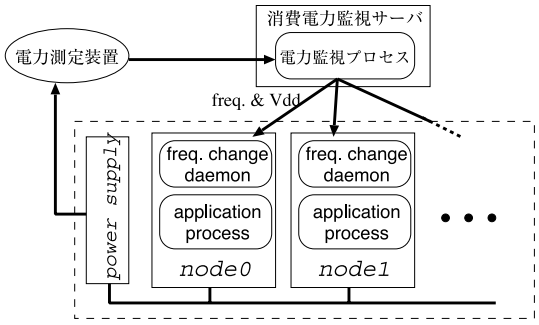

図 4 電力管理システム
Fig. 4 Runtime power control system.

そこで本稿では，ノード数はシステムに搭載された最大の構成を固定的に選択するが，実効消費電力に応じて周波数/電源電圧を動的に変更し，許容電力内で最も高い周波数で動作させることで高性能化を図る手法について検討する．ノード数を変更させないため，最適化の余地が減ることになるが，プロファイリングなどを行う必要がなく，実効消費電力に基づくクラスタ構成手法の運用方法として有用であると考えられる．

4.2 電力管理手法

通常のクラスタシステムの場合，1 U 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，その値に応じて該当ユニット内のプロセッサの周波数/電源電圧を制御することで，動的に実効消費電力を許容電力内に納めつつ，できるだけ高い周波数で動作させることを目指す．

図 4 に，この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管理サーバに送り，消費電力管理サーバの管理プロセスが，現在の電力と許容電力の値を比較し，ユニットの消費電力が許容電力を上回った場合は，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．各ノードには，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．

なお，本手法のように，動的に電力を監視しつつ周波数/電源電圧を制御した場合でも，許容電力からはずれた場合に，電力制御のための指示を送信してから実際に周波数が変更されるまで，ある程度の時間がかかる．したがって，瞬間的には許容電力以上の電力で動

作してしまう．ここで，本稿では電源に関しては許容消費電力に対し，ある程度の余裕を持った実装を行っていることを仮定する．また，許容される消費電力を決定する要因は熱密度（単位体積あたりの発熱量）であることを前提にしている．近年では，大規模な計算機クラスタを構築するうえで最も大きなコストは設置面積であり，その設置面積を抑制するために高密度実装を行うことが必要である．高密度実装のクラスタを構築するうえでの制限は，熱密度であることから，この仮定は妥当であると考えられる．

4.3 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，許容消費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どのように変更するかの戦略が重要となる．そこで，できる限り許容消費電力の範囲内でできるだけ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）と下限の閾値を与える．
(2) 全体の処理時間に対し，上限の閾値を超えてもよい時間の割合を与え，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超えており，かつ，与えられた下限の閾値を下回る消費電力を観測した場合，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 5** に示す．

```
C_wait_time = 100 / Th_overshoot
freq = Freq_min;
set_freq_all_nodes(freq);

 (invoke 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 >= C_wait_time) {
        if (W_observed <= W_min_threshold) {
            freq++;
            set_freq_all_nodes(freq);
        }
    }
    C_safe_time++;
}
```

図 5 周波数制御アルゴリズム
Fig. 5 Algorithm of runtime clock control.

5. 動的実効電力制御手法の評価

本章では，4.2 節の電力管理システム，および 4.3 節のアルゴリズムを実際のクラスタシステムに実装し，アプリケーション実行中に動的に電圧・周波数を変更することで，提案手法の有効性を示す．

5.1 評価環境および評価条件

評価環境は 3.1 節で述べたクラスタシステムに対し，新たに電力管理サーバを追加し評価を行う．また，ベンチマークプログラムは HPL，Himeno-Bench，および NAS Parallel Benchmark（NPB）中のカーネルベンチマークすべてを用いる．

表 4 比較対象として選択された周波数（GHz）
Table 4 Selected clock frequency for the conventional cluster.

閾値	CG	EP	FT	IS
350 W	1.86	1.86	1.86	1.86
300 W	1.33	1.73	1.33	1.60
250 W	0.80	1.06	0.80	1.06
閾値	LU	MG	Himeno-Bench	HPL
350 W	1.86	1.73	1.86	1.86
300 W	1.46	1.33	1.46	1.20
250 W	0.80	0.80	0.80	0.80

また，アルゴリズム中の閾値の値は以下の 3 通りの場合について評価を行う．

- 上限：350 W，下限：340 W
- 上限：300 W，下限：290 W
- 上限：250 W，下限：240 W

従来手法との比較を行うため，アプリケーションごとに，8 ノード使用時で周波数を固定して全周波数で実行を行い，実効電力が許容電力を超えない範囲で最高の周波数の場合を選択し，比較対象として用いる．なお，各アプリケーションにおいて選択された周波数を**表 4** に示す．

なお，4.3 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システムにおいて，消費電力管理サーバ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，低い周波数で処理が行われ，次の周期の平均電力が十分

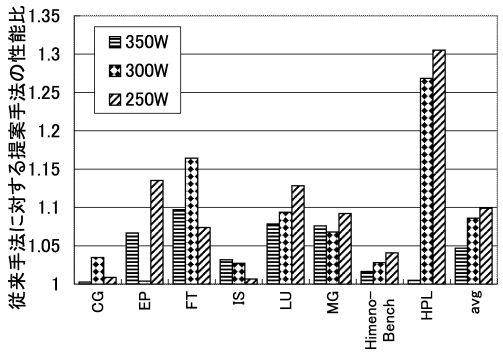

図 6 従来手法と提案手法の性能比
Fig. 6 Relative performance of the proposed cluster.

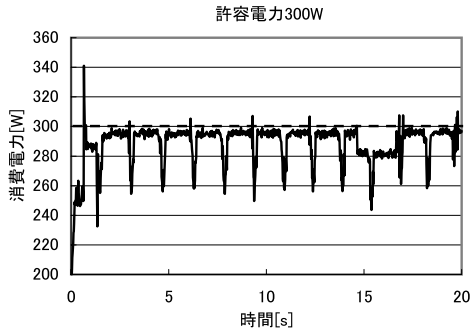

図 7 HPL での電力変化の様子（許容電力 300 W）
Fig. 7 Power profile for HPL (300 W limit).

低くなるまでの時間を計測したところ，16 ms 程度必要であることが分かった．そこで，消費電力管理サーバが対象ユニットの電力を監視する周期を 20 ms に設定して評価を行う．また，閾値逸脱の許容値は 1%として評価を行う．閾値逸脱の許容値を変化させると性能および装置の温度に与える影響は変化することが予想されるため，この値を変化させた場合について 5.3 節において評価し，議論する．

5.2 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 4.3 節で提案した電力管理手法を適用して評価した場合の結果を**図 6** に示す．図は，表 4 に示す比較対象として選択された周波数で実行した場合に対する相対性能比を表している．

結果から，提案する手法により，今回実験に用いたすべてのベンチマークにおいて性能が向上していることが分かる．平均で見ると，許容消費電力 350 W の場合で 4.6%，300 W の場合で 8.6%，250 W の場合で 9.9%の性能向上を達成している．このように上限の閾値（許容消費電力）が低い場合のほうが，提案手法による性能向上率が高いことが分かる．これは，従来型のクラスタ構成では消費電力が閾値を超えてしまうような周波数は選ばれないことに起因する．閾値が高い場合には従来型の構成では高い周波数が選択されるため，提案手法で動的に周波数/電圧を変更する場合に従来手法よりも高い周波数を選択する余地がほとんど存在しない．しかし，閾値に低い値を設定した場合，従来型の構成では低い周波数が選択されるため，機会があれば提案手法では非常に高い周波数で駆動可能であり，その効果が大きな性能向上として現れたということが分かる．

一方，我々の提案する実効電力に基づくクラスタ構成では，動的に周波数/電源電圧を制御するため，実効電力が増大した場合は許容消費電力を上回らないように周波数を低下させ，実効電力が低くなった場合には周波数を高くして実行する．このため，消費電力的に効率の良い実行ができた結果が性能向上として現れる．参考として，**図 7** に HPL の場合で許容電力 300 W の場合の電力変化の様子を示す．図から，許容電力付近で実際に動作しており，閾値逸脱の許容値が 1%以内となるように一定周期ごとに周波数が上げられていることが分かる．

次に，それぞれのアプリケーションが実行中にどの周波数でどれだけの時間処理を行ったかの割合を**図 8** に示す．これらの結果を比較すると全体的な傾向として，許容消費電力 350 W においては我々の提案手法におけるアドバンテージは比較的小さいことが見てとれる．これは許容消費電力 350 W では比較対象で選択される固定周波数が 1.86 GHz と高いため，提案手法において 2 GHz でほとんどの時間動作しているもののあまり大きな差が生まれないことによる．一方，許容消費電力 300 W や 250 W の場合では，比較対象で選択される固定周波数が比較的低く，提案手法では高い周波数で動作する時間が長いため，許容消費電力 350 W の場合に比べて性能向上率が高くなっている．

また，アプリケーションごとの比較では，比較対象として選択された固定周波数よりも高い周波数で動作している割合が大きいものほど，提案手法の性能向上率が大きくなっていることが分かり，提案手法の有効性を示している．

5.3 閾値逸脱の許容値に関する考察

4.3 節で提案したアルゴリズムを実装するにあたって，閾値逸脱の許容値は，性能および装置の温度に影響を及ぼす．そこで，この影響を調べるため，閾値逸脱の許容値を変化させ，実際にベンチマークを処理した場合の性能と CPU の最高到達温度を CPU に搭載されている温度センサから取得し，測定した．これら

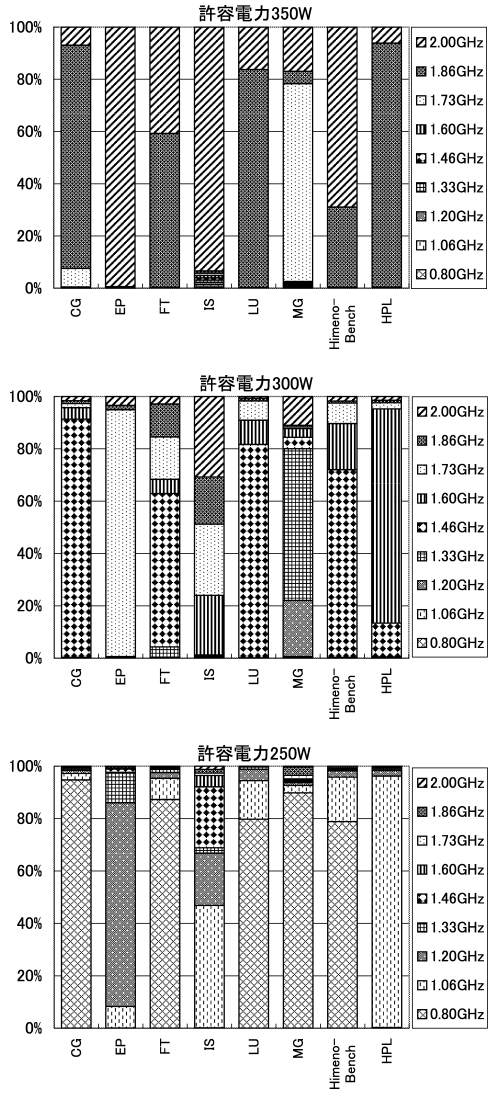

図 8 実行時間中の各周波数で動作した時間の割合
Fig. 8 Breakdown of execution time spent in each clock frequency.

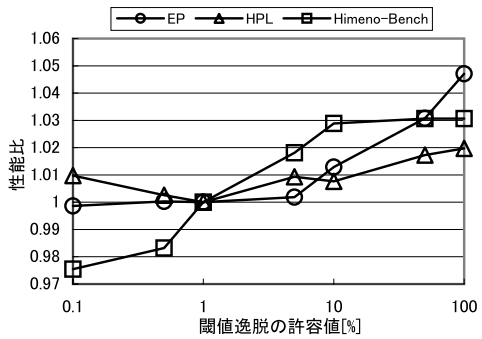

図 9 閾値逸脱の許容値と性能の関係
Fig. 9 Relative performance varying $\mathrm{Th}_{overshoot}$.

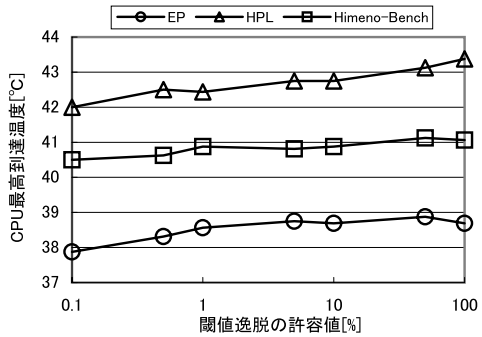

図 10 閾値逸脱の許容値と装置温度の関係
Fig. 10 Observed maximum CPU temperature varying $\mathrm{Th}_{overshoot}$.

の結果を**図 9**，**図 10** に示す．なお，これらの結果は許容電力が 300 W の場合のものである．

図 9 では，5.2 節で用いた閾値逸脱の許容値を 1%とした場合の性能を 1 とし，それに対する性能比を示している．また，図 10 は温度の絶対値を示している．

当然であるが，閾値逸脱の許容値を大きくすれば性能が向上する傾向にある．しかし，図 10 より，閾値逸脱の許容値を大きくするにともなって，装置の最高到達温度も上昇する，すなわち発熱量が増大することが分かる．したがって，性能の向上率と発熱量の間にはトレードオフ関係が存在する．冷却能力の限界のもとで最高性能を得ようとする場合には，最も CPU 負荷が高く発熱量が大きいプログラムの発熱量が，システムが安全に動作できる範囲で実行できるよう，閾値逸脱の許容値を大きくとることが重要となる．本稿では，1%の値を用いて評価を行ったが，種々のシステムにおいて，許容電力と温度との関係を考慮し，最適な閾値逸脱の許容値を調べることは今後の課題である．

6. 関連研究

近年，消費電力/エネルギーを考慮し，性能に影響を与えない範囲で，それらを削減するための研究が活発に行われている．

文献 7) では NAS Parallel Benchmark を用い，性能と消費電力のプロファイリングの結果から，高性能クラスタシステムにおける消費電力/エネルギーの特徴に関して解析を行っている．また，文献 8) では DVS が可能なクラスタにおいて，DVS を用いることによる消費電力削減の可能性について評価が行われており，性能低下をわずかに抑えつつ，大幅な消費電力/エネルギーの削減が可能であることが示されている．文献 9) では，DVS 可能なクラスタシステムにおける消費エネルギーと性能のトレードオフ関係について調

査が行われており，DVS に加えて，クラスタの実行ノード数を変更することによる効果についても評価されている．

文献 10)〜12) では，高性能クラスタシステムにおいて，プログラムを静的な解析，あるいはプロファイリング結果を用いることで，性能の低下を極力抑えたうえで消費電力/エネルギーを削減するための種々の手法が提案されている．また，Kappiah らは文献 13) において *Jitter* と呼ばれる，ノードごとの負荷の不均衡に着目した消費電力削減手法を提案している．*Jitter* は，計算量の少ないノードに対して DVS を適用するものである．文献 14) では，消費エネルギーの上限という制約が与えられたシステムにおいて，制約を満たしたうえで実行時間を最小にするための最適なスケジューリング（ノード数，クロック周波数）手法を提案している．また，実行時に動的にプロセッサの周波数/電源電圧を，ユーザプログラムからは透過的に変更するアルゴリズムが文献 15) において提案されている．

Felter ら[16] は *power shifting*. サーバーシステムの各コンポーネント（プロセッサ，メモリサブシステム）に対して，動的に電力を割り振ることで，あらかじめ決められた電力使用量の制約を満たしつつ性能を最大化する手法を提案している．

本稿で提案する手法は，クラスタにおいて性能に影響を及ぼさずに消費電力やエネルギーを削減することを目的にしているわけではなく，科学技術計算などのスーパコンピューティング分野のアプリケーションを対象に，ピークの消費電力は許容電力を超えてしまうような数のプロセッサを搭載し，DVS を利用して消費電力の制約下で可能な限り性能向上を狙うことが目的である．この点で本質的に従来の研究とは異なるものである．

Rubio らは消費電力や冷却の制約に対して余裕のある場合にクロック周波数を名目上の最大周波数よりも高める *Dynamic Processor Overclocking*（*DPO*）という手法を提案している[17]．彼らは実機のプラットフォーム上において DPO の有効性を確認し，性能向上が見込めるアプリケーションの特徴について解析を行っている．DPO のコンセプトは本稿でのそれと類似しているが，彼らは単一のプロセッサにおける性能向上について着目しており，クラスタ計算機などの並列処理環境については考慮していない．

7. 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超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」を提案した．また，システムの電力を動的に監視し，冷却能力により決定された許容消費電力を超えそうな場合には，DVS 手法により消費電力を制御しつつ実行させる手法も提案した．また，提案手法を適用したプロトタイプのクラスタシステムを構築し，従来のクラスタ構成と比較評価を行った．

評価の結果，提案するクラスタ構成手法を用いることで，従来型のクラスタシステムに比べ高い性能を達成できることが分かった．

本稿で用いた手法では，システム内に集中的に過熱するホットスポットが存在した場合，ホットスポットが正常に作動する温度に収まるように消費電力の上限の閾値を設定しなければならない．今後の課題として，消費電力の上限の閾値を最適に設定する手法について検討を行う．

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) TOP500 team: Top500 list for November 2005. http://www.top500.org/lists/2005/11
2) Warren, M., et al.: High Density Computing: A 240-Node Beowulf in One Cubic Meter, *Proc. Supercomputing 2002* (Nov. 2002).
3) Nakashima, H., et al.: MegaProto: 1 TFlops/10 kW Rack Is Feasible Even with Only Commodity Technology, *Proc. Supercomputing 2005* (Nov. 2005).
4) IBM and Lawrence Livermore National Laboratory: An Overview of the BlueGene/L Supercomputer, *Proc. Supercomputing 2002* (Nov. 2002).
5) 堀田義彦ほか：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：コンピューティングシステム，Vol.45, No.SIG11 (ACS7), pp.207–218 (2004).
6) Ge, R., et al.: Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters, *Proc. Supercomputing 2005* (Nov. 2005).

7) Feng, X., Ge, R. and Cameron, K.: Power and Energy Profiling of Scientific Applications on Distributed Systems, *Proc. IPDPS 2005* (Apr. 2005).
8) Hsu, C. and Feng, W.: A Feasibility Analysis of Power Awareness in Commodity-Based High-Performance Clusters, *Proc. Cluster 2005* (Sep. 2005).
9) Freeh, V., Lowenthal, D., Springer, R., Pan, F. and Kappiah, N.: Exploring the Energy-Time Tradeoff in MPI Programs on a Power-Scalable Cluster, *Proc. IPDPS 2005* (Apr. 2005).
10) Ge, R., Feng, X. and Cameron, K.: Improvement of Power Performance Efficiency for High-End Computing, *Proc. Workshop on HP-PAC 2005* (Apr. 2005).
11) Kotla, R., Ghiasi, S., Keller, T.W. and Rawson, F.L.: Scheduling Processor Voltage and frequency in Server and Cluster Systems, *Proc. IPDPS 2005* (Apr. 2005).
12) Freeh, V., Lowenthal, D., Pan, F. and Kappiah, N.: Using Multiple Energy Gears in MPI Programs on a Power-Scalable Cluster, *Proc. PPoPP'05* (June 2005).
13) Kappiah, N., Freeh, V. and Lowenthal, D.: Just-in-Time Dynamic Voltage Scaling: Exploiting Inter-node Slack to Save Energy in MPI Programs, *Proc. SC'05* (Nov. 2005).
14) Springer, R., Lowenthal, D.K., Rountree, B. and Freeh, V.W.: Minimizing Execution Time in MPI Programs on an Energy-Constrained, Power-Scalable Cluster, *Proc. PPoPP'06* (Mar. 2006) (to appear).
15) Hsu, C. and Feng, W.: A Power-Aware Run-Time System for High-Performance Computing, *Proc. SC'05* (Nov. 2005).
16) Felter, W., Rajamani, K., Keller, T. and Rusu, C.: A Performance-Conserving Approach for Reducing Peak Power Consumption in Server Systems, *Proc. ICS'05* (June 2005).
17) Rubio, J., Rajamani, K., Rawson, F., Hanson, H., Ghiasi, S. and Keller, T.: Dynamic Processor Overclocking for Improving Performance of Power-Constrainded Systems, IBM Research Report RC23666 (W0507-124) (July 2005).
18) http://www.commell.com.tw/Product/SBC/LV-673.HTM

(平成 18 年 1 月 30 日受付)
(平成 18 年 5 月 22 日採録)

池田 佳路（学生会員）
2005 年東京大学工学部計数工学科卒業．現在，同大学大学院情報理工学系研究科修士課程在学中．

近藤 正章（正会員）
1998 年筑波大学第三学群情報学類卒業．2000 年同大学大学院工学研究科博士前期課程修了．2003 年東京大学大学院工学系研究科先端学際工学専攻修了．博士（工学）．独立行政法人科学技術振興機構戦略的創造研究推進事業 CREST 研究員を経て，現在東京大学先端科学技術研究センター特任助手．計算機アーキテクチャ，ハイパフォーマンスコンピューティング，ディペンダブルコンピューティングの研究に従事．電子情報通信学会，IEEE，ACM 各会員．

中村 宏（正会員）
1985 年東京大学工学部電子工学科卒業．1990 年同大学大学院工学系研究科電気工学専攻博士課程修了．工学博士．同年筑波大学電子・情報工学系助手．同講師，同助教授を経て，1996 年より東京大学先端科学技術研究センター助教授．この間，1996〜1997 年カリフォルニア大学アーバイン校客員助教授．高性能・低消費電力プロセッサのアーキテクチャ，ハイパフォーマンスコンピューティング，ディペンダブルコンピューティング，ディジタルシステムの設計支援の研究に従事．情報処理学会より論文賞（平成 5 年度），山下記念研究賞（平成 6 年度），坂井記念特別賞（平成 13 年度），各受賞．IEICE，IEEE，ACM 各会員．