

## Webサーバ用計算機クラスタの電力効率最適化に関する初期検討

大谷 貴胤<sup>†</sup> 池田 佳路<sup>†</sup> 佐々木 広<sup>†</sup>  
近藤 正章<sup>†</sup> 中村 宏<sup>†</sup>

ウェブサーバ用計算機クラスタはその消費エネルギーと冷却にかかるコストの増加に伴い電力管理が重要になってきている。従来のウェブサーバクラスタは負荷のピーク時に備え全てのノードが常時稼働している設計であった。しかし実際には負荷がピークに達するような状況はめったになく、性能の余裕を残した状態で稼働している時間が多い。本研究では、負荷状況に応じて、ノード数と動作周波数を変化させることにより、性能を維持しつつ最も消費電力の低い構成を選択し、従来の構成手法よりも低い消費電力を実現する計算機クラスタの電力効率に関する初期検討を行った。

### A Study on Power Efficiency Improvement for Cluster-Based Web-Server

TAKATSUGU OYA,<sup>†</sup> YOSHIMICHI IKEDA,<sup>†</sup> HIROSHI SASAKI,<sup>†</sup>  
MASAAKI KONDO<sup>†</sup> and HIROSHI NAKAMURA<sup>†</sup>

With increasing costs of energy consumption and cooling, power management in Web server clusters has become an increasingly important design issue. Current Web server clusters are designed to handle peak loads, where all servers are fully utilized. In practice, peak load conditions rarely happen and servers are most of the time underutilized. In this paper, we study power-performance trend of a cluster-based Web server, which is reconfigured to reduce power consumption by adjusting the number of nodes and clock frequency of each node according to load conditions.

#### 1. はじめに

高度にインターネットが発達した現代社会では、ウェブ（WWW）ページの閲覧やその検索、電子決済システムによる商品購入など、計算機や携帯型端末を通して多くのインターネットサービスを楽しむことができるようになっている。それらのサービスを提供する上で必要不可欠であるのがサーバ計算機システムであり、増大するサービス要求を処理するために、非常に多くの計算機がサーバ用途として常時稼働状態にある。今後、インターネットを介したサービスは質、量ともにますます増加することが予想され、サーバ計算機システムの重要性はさらに高まると考えられる。

サーバ用計算機は通常、CPU や主記憶、ディスク装置、ネットワークデバイスなどで構成される。また大規模なサービスを提供する場合には、並列処理効果による高い処理スループットを目的に、計算機クラスタ型構成により構築されることが多い。このサーバ用計算機システムは一般的に処理性能が重視され、クラスタのノードとなる各サーバ計算機をより高性能なも

のに拡張する、あるいはノード台数を増やすなどで、システムの処理性能拡張も頻繁に行われる。

しかし、近年ではブレードサーバなどの高密度実装の結果、設置体積あたり熱密度が増加し冷却能力が不足しつつある、あるいは電力のコストが無視できなくなっているなどの理由により、サーバ用計算機においても消費電力を考慮したシステムの開発や運用が重要視されてきている。

ここ数年、大規模学術計算分野向けの計算機クラスタにおいては、電力効率を改善するための研究が活発に行われてきた<sup>1)~8)</sup>。また、電力あたりの性能が優れたクラスタシステムとして、BlueGene/L<sup>9)</sup> や、MegaProto<sup>10)</sup> といったシステムも開発されている。ここで、サーバアプリケーションと大規模学術計算では、例えばディスクシステムが性能上のボトルネックとなる場合があるなど、性能や電力消費の特性、さらには性能への要求が異なるため、単純にこれまでの手法をサーバ用計算機クラスタシステムに適用することはできない。

本稿では、サーバアプリケーションの一つとしてウェブサーバに着目し、計算機クラスタ上でサーバを動作させた際の、リクエストの種類、およびリクエストの負荷（リクエストレート）を変化させた場合のスループット、応答時間、および消費電力に関する初期評価を行

<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,  
The University of Tokyo

う。また、評価結果を考察することで、リクエストの状況に応じて、応答時間の制約を満しつつ消費電力を削減するための指針を示す。

本稿の構成は以下のとおりである。次節では関連研究について述べ、3章ではウェブサーバの性能および電力に関する初期評価を示す。4章では、ウェブサーバの電力最適化の指針および手法を示し、5章でその評価を行う。6章で本論文のまとめと、今後の課題について述べる。

## 2. 関連研究

近年、大規模学術計算分野向けの計算機クラスタにおいて、電力効率を改善するための研究が活発に行われている。文献<sup>1),2)</sup>ではDVSが可能なクラスタにおいて、DVSを用いる、あるいはクラスタの実行ノード数を変更することによる消費電力削減効果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。また、文献<sup>3)~5)</sup>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上で消費電力/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<sup>6)</sup>において *Jitter* と呼ばれる、ノード毎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。文献<sup>7)</sup>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、制約を満たした上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、ユーザー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<sup>8)</sup>において提案されている。本稿での対象アプリケーションは、大規模科学技術計算ではなく、サーバ用途のアプリケーションであり、この点でこれらの研究とは異なるものである。

文献<sup>11)</sup>では、ウェブサーバの電力消費の解析を行い、DVSを用いた消費エネルギー削減について評価が行われている。DVSに加え、request batchingによりさらにエネルギーを削減する手法が文献<sup>12)</sup>で提案されている。また、ある応答時間の制限を満しつつDVSによりウェブサーバの消費電力を削減する手法も提案されている<sup>13)</sup>。これらの研究は、クラスタ構成のサーバシステムを対象にしたものではない。なお、クラスタ構成のサーバシステムの消費エネルギー削減手法については文献<sup>14)</sup>で提案されており、これは負荷に応じて稼働させるノード台数を制御するものである。

また、文献<sup>15)</sup>では、不均一なクラスタ構成のウェブサーバにおいて、応答時間の制限を満たす範囲で、負荷に応じてノード台数やプロセッサの周波数を制御するで、消費エネルギー削減する手法が提案されている。

本手法は、すべてのリクエストはCPU処理がボトルネック(CPU-bound)であると仮定している。しかし、多様なサービスを提供する必要のあるサーバシステムにおいては、CPUだけでなく、特にディスクシステムがボトルネック(Disk-bound)であることも多いと考えられる。本稿では、Disk-boundな場合も考慮しており、この点で文献<sup>15)</sup>とは異なるものである。

## 3. クラスタ型ウェブサーバ

1台のサーバでは処理しきれないような、多数のリクエストがクライアントから送信された場合には、複数のサーバにより負荷を分散して処理を行うことが考えられる。ウェブサーバシステムにおいて、複数のサーバ計算機によりリクエストの処理をする手法がいくつか用いられている<sup>16)</sup>。その代表的な手法について以下に説明する。

- プロキシによる負荷分散：サーバとクライアント(インターネット)の間に存在し、サーバに対してはクライアントの、クライアントに対してはサーバの役割を担うプロキシを用い、クライアントからのアクセスをプロキシが割り振ることで、負荷分散を実現する技術。柔軟な負荷分散が実現できる反面、プロキシに障害が発生した場合には全サーバにアクセスできなくなってしまうり、負荷の高いリクエストに対してはプロキシ自体がボトルネックになってしまうという欠点がある。
- DNS ラウンドロビン：ドメイン名とIPアドレスを対応づけるDNSを利用して、1つのドメイン名に複数のIPアドレスを割り当てる技術。導入が容易であり、特定のマシンに重い負荷がかかるといけないという利点がある反面、分散先のサーバの情報を用いることができないため、サーバの負荷状況を観察しながら負荷分散を行うことができなかったり、障害を起こしているサーバのIPアドレスでもクライアントに返してしまうという欠点がある。ただし、比較的新しいブラウザは接続できなかった場合には返された他のIPアドレスにアクセスするため、後者については問題ではなくなくなってきている。
- HTTP redirection：インターネットでデータの送受信に使われるHTTPにおけるサーバからの応答の種類のひとつであり、アクセス先のURIが変更されたことを通達する技術であるHTTPリダイレクトを用いて、サーバが自身の負荷が高いときには他のサーバにアクセスをリダイレクトすることによって負荷分散を実現する技術。サーバの負荷に対して最も感度が良いという利点がある反面、クライアント側からしたら新たにサーバとのコネクションを張り直す必要があるため、レイテンシが伸びてしまうという欠点がある。

表 1 Pentium M 760 プロセッサの周波数と電源電圧の関係

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

表 2 評価環境

M/B	Commeil LV673 - i915GM + ICH6M chip-set - Gb Ethernet x 2 12V CPU 電源駆動
Processor	Pentium M 760 (Max 2GHz, FSB533MHz)
Memory	DDR2-SDRAM 1GB
HDD	80GB, 7200rpm SATA 3.0GB/s Seek Time 8.8ms

本稿では、比較的リクエストの負荷が高い状況下でのサーバを想定するため、1台のマシンがボトルネックになってしまうプロキシによる負荷分散手法ではなく、HTTP redirection や、DNS ラウンドロビンなどの手法を前提として、評価や議論を行う。

#### 4. 初期評価

##### 4.1 評価環境

評価には、Pentium M 760 プロセッサを搭載した PC を各ノードとする計算機クラスタシステムを用いる。表 1 に、設定可能な周波数および電圧のセットを示す。また、各ノードの仕様を表 2 に示す。このノードを、4 台 Gb Ethernet で接続したクラスタシステムにより評価を行う。OS は Linux kernel-2.6.11 を使い、cpufreq インタフェース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。また、ウェブサーバソフトには apache2.2.3 を用いた。クライアントにはクラスタの各ノードと同様の PC を複数台使い、2 章で述べたように DNS ラウンドロビン、あるいは HTTP redirect で負荷分散を行うことを仮定する。本稿での評価では、DNS サーバやフロントエンドサーバのオーバーヘッドを含めずに評価するために、各クライアントは予め決められたサーバに直接リクエストを送信し、サーバはそのクライアントのリクエストのみを処理することとした。なお、各クライアントからは、各サーバに対して同数の同じ負荷（ページのサイズ、あるいは CPU で行わせる処理）となるようなリクエストを送信する。なお、クライアントには HTTP リクエストを送信するベンチマークソフトである httperf 0.8 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、取り扱いが容易であるという特徴を持つ。評価に用いたボードは 12V の ATX 電源駆動であり、各ノードの

表 3 動的ページの応答時間

Frequency (GHz)	0.80	1.20	1.60	2.00
Reply Time (ms)	27.0	18.3	13.8	11.3

12V 電線に流れる電流を測定し、電圧 12V を乗じることで電力としている。測定対象はサーバのみであり、電力測定装置、サーバ間を接続しているネットワークスイッチで消費する電力は考慮しない。

一般にウェブページは、その性質の違いにより「静的なページ」と「動的なページ」の 2 種類に大別される。静的なページは、ほとんど CPU 処理を必要としないものであり、ディスクなどの二次記憶装置内に存在することが多い。また、動的ページは、Perl や PHP などの言語で書かれた CGI プログラムなどをウェブサーバ上で実行して生成されるものである。従って、通常動的ページは静的ページより多くの CPU 処理を必要とする。本評価では、ウェブページの特性的違いによるサーバの性能や消費電力を評価するために、静的なページと動的なページの両者を留意して評価を行う。

まず、静的なページはページサイズが 1KB の HTML 文書とする。これは、現在のウェブページの多くが 1KB 以下といった比較的小さなサイズの静的ページを複数組み合わせて 1 つのページを形成しているという状況を踏まえたものであり、文献<sup>15)</sup>において 1KB 以下のページへのアクセス率が全体の約 4 割を占めると報告されていることから妥当な仮定であると考えられる。なお、文献<sup>15)</sup>では全ての静的なページはメモリ上にあり、ディスクアクセスを必要としないため、動的なページと同等に扱えることされているが、本稿ではさらに多様なウェブサーバ環境での評価としても適用できるよう、全ての静的なページのリクエストに対して、ディスクアクセスが必要となるような Disk-bound な状況を想定して評価を行う。

本稿で用いる動的なページは、perl で記述された乱数生成プログラムをウェブサーバ上で実行し、その最初と最後の乱数を結果としてウェブページを構築するものである。動的なページの応答時間は、URL で指定する乱数の個数によって変化可能である。参考として、周波数の違いによるこの動的ページへのアクセス時の応答時間を表 3 に示す。表より、周波数の速度にほぼ比例して応答時間が長くなっており、この動的ページが CPU-bound なものであることがわかる。

##### 4.2 評価結果

図 1、および図 2 に、リクエストが動的なページ、および静的なページの場合のそれぞれについて、リクエストレートを変化させた場合の実際のスループット

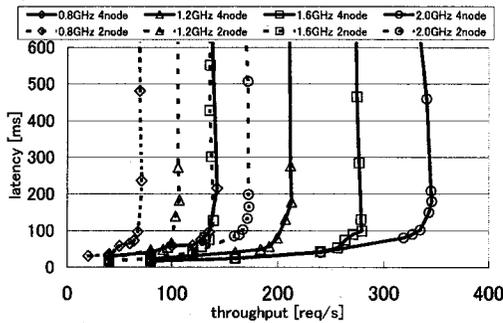


図1 動的ページリクエストに対するスループットと応答時間の関係

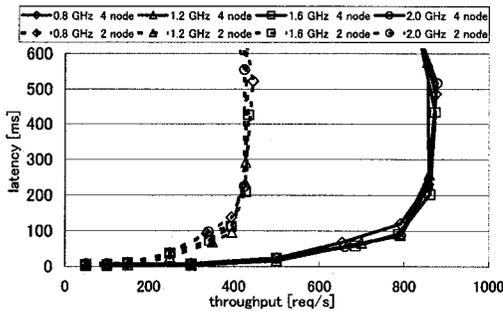


図2 静的ページリクエストに対するスループットと応答時間の関係

と応答時間（レーテンシ）の関係を示す。なお、図は2ノードおよび4ノードを用い、CPUの周波数が0.8GHz、1.8GHz、1.6GHz、2.0GHzの各場合<sup>\*</sup>における評価結果を示している。

今回の評価では全てのノードにおいて同一の性質（動的）または静的を持つリクエストを処理するため、どのノードにおいても周波数を上げることによる処理能力の向上は同一である。またプロセッサの消費電力は、周波数に比例し、電源電圧の2乗に比例する。よって最も電力効率の良い組み合わせは、全てのノードの周波数を同一にした場合となる。よって、本評価においても周波数を全てのノードで一定とした。

まず、動的なページへのリクエストにおける結果について議論する。図1より、すべての構成において、クライアントからのリクエストレートを増やすとレーテンシはあまり変わらずにスループットが増加していくが、あるところでサーバの処理能力が飽和し、スループットが向上せずにレーテンシが非常に長くなることわかる。この時、ノード数やCPUの周波数に依存して性能が飽和する点が異なり、ノード数と周波数にほぼ比例して最大処理能力が向上していることがわかる。これは、前節で述べたように、動的なページ

<sup>\*</sup> 選択可能な周波数は9段階あるが、本評価においては簡単のため、4段階のみを使用した

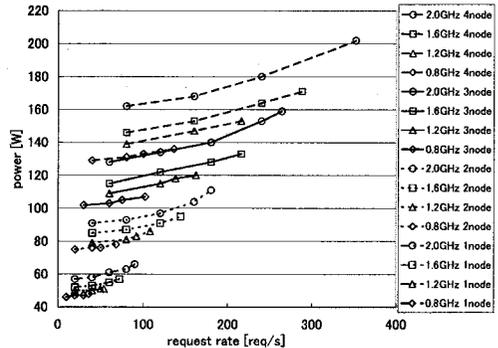


図3 動的ページのリクエストレートとwebサーバの消費電力

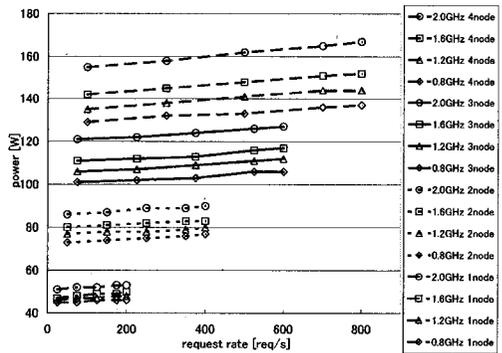


図4 静的ページのリクエストレートとwebサーバの消費電力

の処理がCPU-boundであるためである。

一方で、図2に示す静的なページへのリクエストの結果では、ノード数を増やすことで処理能力が向上するものの、周波数を変更しても処理能力にほとんど変化がないことがわかる。これは、静的なページのリクエストをサーバ上で処理する際のボトルネックがディスクアクセスであり、CPUの周波数には性能が依存しないことが原因である。

一般的に、ウェブサーバを運用する際には、応答時間がある値以下になるように、サーバの構成を設定することが多い。したがって、応答時間がそれ以下であり、その時点で処理すべきリクエストレートが満たせる構成であれば、どの構成でも良いことになる。ここで、ウェブサーバの電力効率を考えると、なるべく少ないノード数、あるいはなるべく低い周波数で動作させることが望ましく、処理すべきリクエストレートが満たせる範囲で最も消費電力の小さな構成を用いてサーバを運用するべきである。

そこで、次にリクエストレートを変化させた場合のクラスタシステムの消費電力に関して評価を行う。図3、および図4は、リクエストレートを変更させた際の、動的なページ、および静的なページへのアクセ

スの際の消費電力を、ノード数および CPU の周波数毎にプロットしたものである。なお、満たすべき応答時間の制約は、動的なページ、静的なページの場合ともに 200ms を仮定し、応答時間がそれ以上になってしまう構成はプロットしていない。

図より、周波数が上がるにつれて、またノード数を増加させるにつれて消費電力が高くなるのがわかる。特に、ノード数を増やした場合は周波数を上げる場合に比べ消費電力の増加分が大きい。これは、本評価でノードに用いた PC では、CPU の消費電力に対してチップセットやマザーボード、ハードディスクで消費される電力も大きく、またそれらの部分は CPU の周波数や稼働状態に関係なく電力が消費されることから、ノードの追加によりシステムの消費電力に大きく影響するためである。

図 3、図 4 より、動的なページ、静的なページのそれぞれについては、処理すべきリクエストレートが決まると最も低い消費電力の構成を選択することで、ウェブサーバシステムの電力効率を最適化することが可能となる。例えば、図 3 の動的なページの場合では、リクエストレートが毎秒 100 リクエスト程度であれば 2 ノードを用いて周波数を 1.2 GHz にすれば良く、毎秒 200 リクエスト程度であれば 3 ノードを用いて周波数を 1.6 GHz にすれば良い。なお、静的なページの場合は、Disk-bound なため周波数を変更しても性能が向上しないため、ノード数のみをリクエストレートに合わせて変更し、周波数は 0.8 GHz を用いることが電力効率の観点から良いことがわかる。

これまでは、動的なページと静的なページへのリクエストを独立に扱ってきた。しかし、通常はそれらのリクエストが混在することも少なくない。そこで、次節では動的なページと静的なページのリクエストが混在する場合も含め、クラスタ構成のウェブサーバシステムの電力効率最適化の指針について検討する。

## 5. 電力効率最適化の指針

本章では、動的・静的の各々のページに対するリクエストレートが与えられたときに、レーテンシの制約および処理すべきリクエストレートを満たす構成の中で、最も消費電力が少ない構成（台数・周波数）を選ぶための指針について議論する。まず、簡単のために前章で示した二種類のリクエストのみが送信されると仮定する。

### 5.1 各サーバが一種類のリクエストを扱う場合

ウェブサーバの運用形態によっては、CGI のリクエストのみを担当するサーバ、静的なファイルのみが置かれているサーバというように、各サーバが異なる役割を与えられている場合がある。本節ではまず、このような場合についての電力効率最適化について議論する。

### リクエストが一種類のみの場合

前章の初期評価より、事前にシステムのプロファイルを取ることによって、あるリクエストレートの際の最適な構成は一意に定めることができる。

### リクエストが複数種類の場合

サーバの台数に制限がない場合、各々のリクエストに対して独立に考え、それぞれの最適な構成を組み合わせることで電力効率を最適化することが可能である。

サーバの台数に制限がある（N 台）場合、各々のリクエストに対して最適な台数の和が N より大きくなる可能性がある。この場合は、台数の合計が N を超えないような組合せの中で、最適な構成を選択する必要がある。リクエストが 2 種類の場合ならば、選択可能な台数の組合せは  $N(N-1)/2$  通りである。ただし、今回の初期評価の結果からは、台数を減らすかわりに周波数を高くすることによって性能および消費電力が共に高くなるという傾向は観察されていないため、各々のリクエストに対して最適な台数の和がサーバを構成する台数を超えてしまうようなリクエストは処理できないといえる。

### 5.2 各サーバが複数種類のリクエストを扱う場合

これまでは、各サーバが処理するリクエストは一種類のみとしてきたが、さらなる電力効率最適化のためには、複数の異なるリクエストを一台で処理した方がよいことがあると考えられる。たとえば、Disk-bound なリクエストを処理中のサーバにおいて、CPU はアイドルであることが多いと思われるため、CPU-bound なリクエストの処理を並行して実行することによってサーバ利用効率を高められる可能性がある。

図 5 は、動的・静的のどちらかのページに対するリクエストレートを固定したときに、もう一方がレーテンシ制約を満たす範囲内で処理できる最大のリクエストレートをプロットしたものである（サーバ 1 台、周波数 0.8 GHz）。図中、x 軸が静的なページに対するリクエストレートを、y 軸は動的なページに対するリクエストレートを表している。図より、プロットされた点を結ぶと動的・静的各々最大のリクエストレートを結ぶ線分よりも外側に凸なグラフとなっていることが分かる。つまり、動的・静的の各々のページに対する処理可能な最大のリクエストレート値の半分よりも多くのリクエストを処理可能であることを意味している。したがって、例えば動的なページに対するリクエストレートが毎秒 60 程度、静的なページに対するリクエストが毎秒 200 程度の場合、5.1 節における最適な構成は図 3、図 4 より、動的なページを処理するのに 1 ノード 1.6 GHz、静的なページを処理するのに 1 ノード 0.8 GHz にする必要があり、このときの消費電力は 85 W であるのに対して、動的・静的なページに対するリクエストを各々均等に、共に 0.8 GHz で駆動しているサーバ上で処理すると消費電力は 73 W

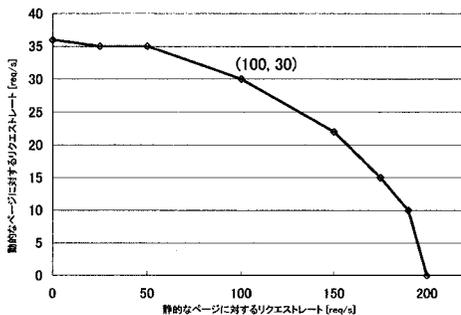


図5 静的ページのリクエストレートに対する、動的ページのリクエストレートの最大値

となり、異なるリクエストを1台で処理した方が電力効率に優れていることがわかる。

以上より、特性の異なるリクエストを同一サーバで実行した方が電力効率が向上する場合があるといえる。リクエストごとにCPUやディスク・メモリの使用率などをモデル化し、なるべく各サーバのCPUおよびディスク・メモリの使用率が最大限になるようにリクエストを割り振ることが電力効率最適化のためには重要であると考えられる。このモデル化および評価は今後の課題である。

## 6. まとめと今後の課題

本稿では計算機クラス構成によるウェブサーバにおいて、電力効率を改善するための初期検討を行った。リクエストの性質による性能、消費電力における知見を得るため、異なる動作ノード数と各ノードのCPUクロック周波数において、リクエストの種類およびリクエストレートを変化させ、性能と消費電力を評価した。また、異なる種類のリクエストを同一のサーバで処理することによって電力効率を最適化できる可能性を示した。

今後の課題としては、本稿での初期評価で得られた指針をもとに、電力効率が最大となる構成を選択するためのアルゴリズムを構築し、評価によってその有効性を示すことが挙げられる。

**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「革新的電源制御による超低電力高性能システム LSI の研究」、および東レ科学振興会科学研究助成の支援によるものである。

## 参考文献

1) C. Hsu and W. Feng, "A Feasibility Analysis of Power Awareness in Commodity-Based High-Performance Clusters", In *Proc. Cluster 2005*, Sep. 2005.

2) V. Freeh, D. Lowenthal, R. Springer, F. Pan, and N. Kappiah, "Exploring the Energy-Time Tradeoff in MPI Programs on a Power-Scalable Cluster", In *Proc. IPDPS 2005*, Apr. 2005.

3) R. Ge, X. Feng, and K. Cameron, "Improvement of Power Performance Efficiency for High-End Computing", In *Proc. Workshop on HP-PAC 2005*, Apr. 2005.

4) R. Kotla, S. Ghiasi, T. W. Keller, and F. L. Rawson, "Scheduling Processor Voltage and frequency in Server and Cluster Systems", In *Proc. IPDPS 2005*, Apr. 2005.

5) V. Freeh, D. Lowenthal, F. Pan, and N. Kappiah, "Using Multiple Energy Gears in MPI Programs on a Power-Scalable Cluster", In *Proc. PPOPP'05*, June 2005.

6) N. Kappiah, V. Freeh, D. Lowenthal, "Just-in-Time Dynamic Voltage Scaling: Exploiting Inter-node Slack to Save Energy in MPI Programs", In *Proc. SC'05*, Nov. 2005.

7) R. Springer, D. K. Lowenthal, B. Rountree, V. W. Freeh, "Minimizing Execution Time in MPI Programs on an Energy-Constrained, Power-Scalable Cluster", In *Proc. PPOPP'06*, Mar. 2006.

8) C. Hsu and W. Feng, "A Power-Aware Runtime System for High-Performance Computing", In *Proc. SC'05*, Nov. 2005.

9) IBM and Lawrence Livermore National Laboratory, "An Overview of the BlueGene/L Supercomputer", In *Proc. Supercomputing 2002*, Nov. 2002.

10) H. Nakashima et al., "MegaProto: 1TFlops/10kW Rack Is Feasible Even with Only Commodity Technology", In *Proc. Supercomputing 2005*, Nov. 2005.

11) P. Bohrer et al., "The Case for Power Management in Web Servers", In *Power-Aware Computing*, Kluwer/Plenum series in Computer Science, 2002.

12) M. Elnozahy, M. Kistler, and R. Rajamony, "Energy Conservation Policies for Web Servers", In *Proc. 4th USITS*, pp.99-112, Mar. 2004.

13) V. Sharma, A. Thomas, T. Abdelzaher, K. Skadron, L. Zhijian, "Power-aware QoS Management in Web Servers", In *Proc. 24th RTSS*, pp.63-72, Dec. 2003.

14) K. Rajamani and C. Lefurgy, "On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters", In *Proc. ISPASS 2003*, pp.111-122, Mar. 2003.

15) C. Rusu, A. Ferreira, C. Scordino, and A. Watson, "Energy-Efficient Real-Time Heterogeneous Server Clusters", In *Proc. 12th RTAS*, pp.418-428, Apr. 2006.

16) V. Cardellini, M. Colajanni, P. S. Yu, "Dynamic Load Balancing on Web-Server Systems", *IEEE Internet Computing*, Vol. 3, Issue 3, pp.28-39, 1999.

17) COMMELL, "Mini-ITX Express Motherboard LV-673 Datasheet (<http://www.commell.com.tw/Product/SBC/LV-673.HTM>)"

18) Intel, "Pentium M Processor on 90nm Process with 2-MB L2 Cache Datasheet", Jan. 2005.