

モデリングに基づく Web サーバ用 計算機クラスタの低消費電力化

大谷 貴胤^{†1} 佐々木 広^{†2,*1}
近藤 正章^{†3,*2} 中村 宏^{†1,†3}

ウェブサーバ用計算機クラスタは消費エネルギーと冷却コストの増加にともない、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ずすべ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少なく、性能の余裕を残した状態で稼働している時間が多いため、我々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、応答時間制約を満たし最も消費電力の低い構成を選択し従来よりも低消費電力であるシステムを実現する手法を提案する。本論文では、その実現上必要となる処理性能と消費電力のモデルを構築し、実際のクラスタシステムに適用した際の電力削減効果について評価を行い、その有効性を検討した。

Power-performance Modeling and Optimization for Cluster-based Web Servers

TAKATSUGU OYA,^{†1} HIROSHI SASAKI,^{†2,*1}
MASAAKI KONDO^{†3,*2} and HIROSHI NAKAMURA^{†1,†3}

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. But in practice, peak load conditions rarely happen and servers are most of the time underutilized. We, so far, proposed a low power 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. In this paper, we describe power-performance modeling and optimization for cluster-based Web servers.

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きている。それらのサービスを提供するうえで必要不可欠な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、設置体積あたりの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、大規模科学技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¹⁾⁻⁸⁾。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/L⁹⁾ や、Mega Proto¹⁰⁾ といったシステムも開発されている。ここ

†1 東京大学大学院情報理工学系研究科

Graduate School of Information Science and Technology, The University of Tokyo

†2 東京大学大学院工学系研究科

School of Engineering, The University of Tokyo

†3 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

*1 現在、東京大学先端科学技術研究センター

Presently with Research Center for Advanced Science and Technology, The University of Tokyo

*2 現在、電気通信大学大学院情報システム学研究科

Presently with Graduate School of Information Systems, The University of Electro-Communications

で、サーバアプリケーションと大規模科学技術計算では、たと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。

本論文では、サーバアプリケーションの1つとしてウェブサーバに着目し、計算機クラスタベースのウェブサーバにおいて、負荷状況に応じてノード数および各ノードのCPUの動作周波数を変化させることにより、従来よりも消費電力の低いシステムを実現する手法を提案する。具体的には、リクエストされるページの負荷を入力として、システムの処理性能と消費電力をモデル化し、応答時間制約を満たしつつ最も消費電力の小さい構成となるようにシステムを最適化するものである。本論文では、計算機クラスタベースのウェブサーバについて処理性能のモデルと消費電力のモデルを構築し、実際のクラスタシステムのノード数と動作周波数を制御した際の電力削減効果について評価を行う。

本論文の構成は以下のとおりである。次章では関連研究について述べ、3章で提案手法の概要を述べる。4章で計算機クラスタベースのウェブサーバの性能および電力モデルを提案し、5章で評価を行う、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

文献 20) では、ウェブサーバの電力消費の解析を行い、DVFS 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。DVFS 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 12) で提案されている。また、ある応答時間の制限を満たしつつ DVFS 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¹³⁾。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 14) 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 15) 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ことで、消費エネルギーを削減する手法が提案されている。この手法では、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。本論文では、Disk-bound な場合も考慮しており、この点で文献 15) とは異なるものである。また、文献 15) では負荷に応

じてノード台数やプロセッサの周波数を制御する際に、あらかじめプロファイリングをとっておくことによって最適な構成を選択しているのに対して、提案手法では性能と電力のモデリングを行い、そのモデルから最適な構成を導出することが可能となっている。このように、あらかじめプロファイリングをとる必要がないことは、提案手法の優位な点としてあげられる。

3. 提案手法

3.1 クラスタ構成

本節では、本論文で仮定するクラスタサーバの構成について述べる。クラスタシステムはフロントエンドおよびバックエンドサーバから構成されている。フロントエンドサーバがクライアントからリクエストを受け取り、DNS ラウンドロビンや HTTP redirection²¹⁾ といった手法を用いてバックエンドサーバに負荷を分散し、リクエストの処理はバックエンドサーバが行う。フロントエンドサーバは処理ノードではなく、その役割は、(1) クライアントからのリクエストを受け付ける、(2) 処理をバックエンドサーバに割り振る、(3) クラスタの構成を適切に変更する (たとえば、サーバの On/Off を切り替える) というものである。また、バックエンドサーバはホモジニアスな構成を仮定している。

3.2 クラスタ型ウェブサーバの電力効率最適化

提案するクラスタ型ウェブサーバの電力効率最適化手法は、リクエストの負荷が与えられたときに、消費電力が最小となる構成を (静的または動的に) 選択する。したがって、リクエストの負荷を正確に予測することが非常に重要な要素となる。この点については、リクエストに周期性があり静的な予測で十分な場合¹⁸⁾ や、一定間隔ごとにリクエストの負荷をフィードバックし、動的に負荷を予測する手法などを用いることによって高い精度で予測ができる¹⁵⁾ ことが知られている。このような知見から、我々は正確な予測が行えるものとして、与えられた負荷に対する最適な構成を求めるための手法を提案する。

一般にウェブページは、その性質の違いにより Disk-bound なものと CPU-bound なものの2種類に大別される。Disk-bound なページはディスクなどの二次記憶装置内に存在する静的な文書ファイルであり、ディスクや I/O 性能がボトルネックとなり CPU 性能は処理のボトルネックとならない。一方、CPU-bound なページは Perl や PHP といったスクリプト言語で記述された CGI プログラムなどをウェブサーバ上で実行することによって生成されるものであり、CPU 性能が処理のボトルネックとなる。2章の関連研究で述べたように、先行研究では Disk-bound なリクエストもすべて CPU-bound であると仮定してい

る^{15),20)}。しかし、これらはボトルネックの異なる処理であるため、それぞれを別に考慮することで、より最適な構成が選択できると考えられる。そのため、本研究では Disk-bound および CPU-bound のリクエストを異なるリクエストとして扱い、さらなる消費電力の削減をねらう。

ここまで述べてきたように、提案手法はリクエストの負荷に応じて最適なサーバ構成を選択するため、ある負荷のリクエストが与えられたときに、任意のサーバ構成における性能と消費電力が求められなければならない。そのために本論文では、リクエストの負荷に対するウェブサーバの性能と消費電力のモデリングを行う。本論文で提案する、モデリングに基づくクラスタ型ウェブサーバの電力効率最適化の概要は以下のとおりである：

- (1) Disk-bound および CPU-bound なリクエストに対するウェブサーバの処理性能と消費電力のモデリングを行う。
- (2) モデリングに基づき、対象とするウェブサーバシステムの処理性能と消費電力の解析を行う。
- (3) 静的または動的に得られたリクエストの負荷の予測値に対して、モデルおよび解析結果から、消費電力が最小となる構成をフロントエンドサーバが求める。
- (4) サーバの構成（ノード数、動作周波数）を動的に変更し最適化を行う。

なお、モデリングの詳細については次章で述べるが、以下の2つのモデルを構築したことが重要な点としてあげられる。

- 処理能力のモデル：まず、1台のノード上で単一の種類のリクエストを処理する場合に、リクエストの負荷からシステムの性能を表すモデル式を導出した。加えて、Disk-bound、CPU-bound なリクエストはそれぞれボトルネックが異なるため、両種類のリクエストを1台のノード上で並行処理すると性能的に有利であることを示し、それぞれの負荷からシステムの性能を表すモデル式を導出した。
- 電力のモデル：処理能力のモデルと同様に、1台のノード上で単一の種類のリクエストを処理する場合、両種類のリクエストを並行処理する場合のそれぞれにおいて、リクエストの負荷から消費電力を表すモデル式を導出した。

4. モデリング

通常、ウェブサーバの処理するリクエストは各々が独立であり、並列に処理することが可能なため、ノード数を増加させた場合の処理能力、および消費電力は単純な和になると考えられる。したがって、1ノードの性能および消費電力のモデリングを行うことによって、

サーバシステム全体の性能および消費電力を求めることが可能となる。はじめに、単一の種類のリクエストを処理する場合の処理性能と消費電力のモデリングについて述べ、次に、Disk-bound および CPU-bound 両種類のリクエストを1ノード上で並行処理する場合のモデリングについて述べる。なお、本章で示す実測値の計測には6.1節に示す評価環境1ノードを用いた。また、応答時間制約は6.1節に示す値と同一の200msとしている。

4.1 単一の種類のリクエストを処理する場合

4.1.1 処理性能のモデリング

通常、クライアントからリクエストされるページの大きさ^{*1}には様々なものが存在し、これらの異なる大きさのページに対するリクエストは混在すると考えられる。そこで本項では、そのような一般的な場合を考え、処理性能のモデリングを行う。

単位時間あたりに、ページサイズの異なる Disk-bound なページが m 種類リクエストされると仮定する。各 Disk-bound なページのページサイズを $page_size^i$ ($i = 0, 1, 2, \dots, m-1$)、リクエストレート（単位時間あたりのリクエスト数）を num_D^i (D : Disk-bound) とすると、総リクエストレート req_D は

$$req_D = \sum_{i=0}^{m-1} num_D^i \quad (1)$$

と表せる。また、平均ページサイズ x_D は

$$x_D = \frac{\sum_{i=0}^{m-1} num_D^i \cdot page_size^i}{req_D} \quad (2)$$

と表せる。また、生成時間の異なる CPU-bound なページが n 種類リクエストされると仮定する。Disk-bound の場合と同様に、各 CPU-bound なページの生成時間を run_time^j ($j = 0, 1, 2, \dots, n-1$)、リクエストレートを num_C^j (C : CPU-bound) とすると、総リクエストレート req_C は

$$req_C = \sum_{j=0}^{n-1} num_C^j \quad (3)$$

と表せる。また、平均生成時間 x_C は

*1 本論文においてページの大きさは、Disk-bound なページの場合はページサイズ（単位：KB）を、CPU-bound なページの場合は生成時間（単位：ms）を意味する。

$$x_C = \frac{\sum_{j=0}^{n-1} num_C^j \cdot run_time^j}{req_C} \quad (4)$$

と表せる．

ウェブサーバの平均処理時間 s_k ($k = D$ or C) は x_k を用いて

$$s_k(x_k, f) = \frac{x_k}{BW_k(f)} + OH_k(f) \quad (5)$$

と表せる．ここで f は CPU 動作周波数であり， OH_k はページサイズや生成時間によらないオーバーヘッド時間， BW_k は各種リクエストに対するウェブサーバシステムのバンド幅である． OH_k および BW_k は，システムに固有の値である．

なお，バンド幅については，Disk-bound の場合，ディスクからのデータ読み出しという I/O 処理が必要となることから，処理時間を決定するシステム要素は CPU 処理だけでなく，メモリバンド幅や PCI バスなどの I/O バスのバンド幅，さらにはディスク自体の読み出しバンド幅などが考えられる．これらのいずれが処理時間を決定するかはシステムに依存するが，Disk-bound の場合，それらが複合することはないと考え，以降ではこの BW_D をメモリバンド幅と呼ぶ．この場合のバンド幅は，単位時間あたりに読み出すことのできるページサイズとなるため，単位は KB/s となる．なお本論文では，最適化対象は CPU の周波数とノード数であり，たとえばディスクの回転数の調整や I/O ポートの電源制御など，ディスクアクセスの際のバンド幅を変更するような手法は対象としておらず，選んだのはメモリバンド幅である．したがって，Disk-bound の場合，式 (5) のように，CPU 周波数をパラメータとしたバンド幅を用いて処理時間を表す^{*1}．

ここで，より正確にシステムの性能をモデル化するために，ディスクキャッシュの影響を考慮する場合などは，たとえばディスクキャッシュのヒット率を導入し，ボトルネックとなるシステム要素ごとにアクセスを分割して，式 (5) のようにモデル化することで対応可能である．また，最適化対象としてディスクアクセスのバンド幅をパラメータ化する場合にも，式 (5) で周波数の関数としている $BW_k(f)$ を，最適化対象となるパラメータの関数とすることで対応可能であるため，比較的容易に拡張することができると考えられる．なお，実際にそれらの影響を考慮したモデルの拡張については，今後の課題である．

一方，CPU-bound なリクエストの場合は，演算処理がほとんどの時間を占めることから，処理時間を決定するシステム要素は CPU の演算性能がほとんどである．この場合のバンド

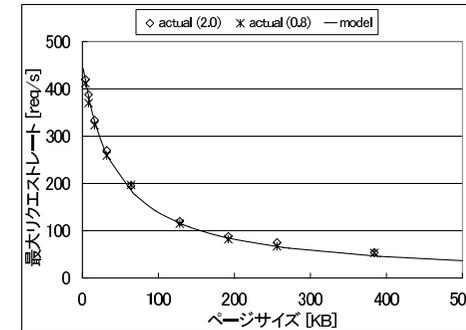


図 1 平均ページサイズと最大リクエストレートの関係 (Disk-bound)
Fig. 1 The maximum request rate to average page size (Disk-bound).

幅は，ある基準のシステム（本論文では，用いた評価環境において最高周波数で動作させた場合）における，リクエストされたページの生成時間を基準として，単位時間あたりにどれだけ生成時間を持つページを処理可能であることを示す量となる．したがって，無次元数となり単位はない．

ここで，平均応答時間 r_k は，待ち行列理論 (M/M/1 型モデル) を導入すると s_k ， req_k を用いて式 (6) のように表される．

$$r_k(x_k, req_k, f) = \frac{s_k(x_k, f)}{1 - req_k \cdot s_k(x_k, f)} \quad (6)$$

応答時間制約を L_k とすると，満たすべき条件は

$$r_k \leq L_k \quad (7)$$

である．以上の式 (5)，(6)，(7) より，処理性能のモデル式である式 (8) が得られる．

$$req_k \leq \frac{1}{\frac{x_k}{BW_k(f)} + OH_k(f)} - \frac{1}{L_k} \quad (8)$$

ここで，モデルの検証を行うとともに，式 (8) の BW_k および OH_k といったシステムに固有の値を求める．その結果を図 1，図 2 に示す．図の actual (2.0) のプロットは CPU 周波数 2.0 GHz での各平均ページサイズまたは平均生成時間 (x_k) における，応答時間制約 ($L_k = 200$ ms) 内に処理可能なリクエストレート (req_k) の最大値を，actual (0.8) は CPU 周波数 0.8 GHz での値をそれぞれ示している．たとえば，図 1 においてはそれぞれの平均ページサイズに対して，その値を与える様々なページサイズの組合せを式 (2) より導

*1 後述の図 1 の結果より $BW_D(f)$ は実際には f (CPU 周波数) には依存しない．

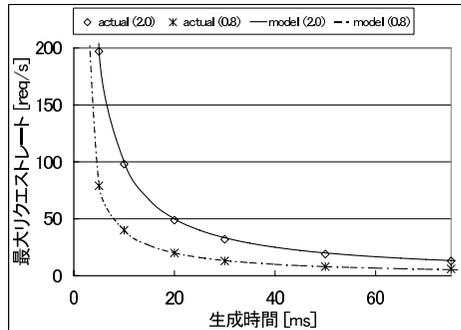


図 2 平均生成時間と最大リクエストレートの関係 (CPU-bound)

Fig. 2 The maximum request rate to average running time (CPU-bound).

出し、実験の入力とした。同じ平均ページサイズとなる組合せを入力としたときに得られる最大リクエストレートは一致したため、図にはそれぞれの平均ページサイズに対して点が 1 つしかプロットされていない。また、実線 model はこれらのプロットを、式 (8) の右辺の関数を用いて最小二乗法により近似したものであり、式 (8) で等号が成立する場合の関数を表している。図 1 および図 2 より、上記のモデルと実測値が非常によく一致しており、本モデルが有効であることが分かる。また、この近似からシステム固有の値 BW_k および OH_k を求めることができる。本論文の評価環境におけるこれらの値は、以下のとおりである。Disk-bound なページの処理の場合は、バンド幅・オーバーヘッド時間ともに周波数によらず $BW_D(f) = 2.3 \times 10^4$ [KB/s], $OH_D(f) = 2.0 \times 10^{-3}$ [s] である。一方、CPU-bound なページの処理の場合は、バンド幅が周波数に依存しており、周波数 $f = 0.8, 1.2, 1.6, 2.0$ [GHz] における $BW_C(f)$ はそれぞれ $4.0 \times 10^2, 6.0 \times 10^2, 7.9 \times 10^2, 9.9 \times 10^2$ である^{*1}。また $OH_C(f)$ は十分小さく無視可能な値である。

4.1.2 負 荷

前項で述べたように、システムにかかる負荷はページの大きさの平均値 x_k およびリクエストレート req_k によって決まる。たとえば Disk-bound なページの負荷は、システムに対して「単位時間あたりに、サイズが x_k のページを req_k 個リクエストされている」状態の処理量、と表すことができる。ここで、本論文では負荷を x_k と req_k の組合せとしてではな

く、より扱いやすくするために、図 1 および図 2 から得られたバンド幅・オーバーヘッド時間を用い、対象とするシステムにおいて、CPU-bound および Disk-bound のそれぞれに対して一次元の指標として以下のように定義する。すなわち、1 ノードにおいて応答時間の制約内で処理可能な処理量の最大値に対する、実際のリクエストの処理量の比、を負荷 $Load_k$ として定義する（よってこの値に単位はない）。

前項で述べたように、処理すべきリクエストのページの大きさの平均値 x_k が与えられたときに 1 ノードが処理可能なリクエストレートは最高周波数における式 (8) の右辺となり、これを max_req_k とすると次のように表される。

$$max_req_k(x_k) = \frac{1}{\frac{x_k}{BW_k(f_{max})} + OH_k(f_{max})} - \frac{1}{L_k} \quad (9)$$

すると、新しく定義する $Load_k$ は実際のリクエストレート req_k と式 (9) より、次式で表される。

$$Load_k = \frac{req_k}{max_req_k(x_k)} \quad (10)$$

この $Load_k$ の値を用いると、与えられた負荷が最低何台のノードを用いることによって処理可能となるかも直感的に理解できる。つまり、たとえば $Load_D = 2.0$ ならば与えられた Disk-bound な負荷は最低 2 ノードで処理可能であることが分かり、また $Load_D = 3.5$ のように小数点以下に値がある場合は、この $Load_D$ の値を切り上げた値を用い、最低 4 ノードで処理可能であることが分かる。

4.1.3 消費電力のモデリング

次に、消費電力のモデリングについて述べる。リクエストがない待機中の場合でも、電源が On の状態のサーバはつねに電力を消費する。これをベース電力 $base_power$ と表す。 $base_power$ は周波数に依存する。リクエストを処理すると、その種類、大きさ、またリクエストレートに応じて電力が増加する。電力は、4.1.2 項で定義した負荷 $Load_k$ に比例して増加すると考えることができる。すなわち増加する電力を $\Delta power_k$ と定義すると、 $\Delta power_k(f, Load_k) = grad_k(f) \cdot Load_k$ と表せる。ここで、 $grad_k$ は比例係数であり、周波数に依存すると考えられる。以上の点に留意すると、負荷に対する消費電力のモデル式は式 (11) のようになる。

$$\begin{aligned} Power_k(f, Load_k) &= base_power(f) + \Delta power_k(f, Load_k) \\ &= base_power(f) + grad_k(f) \cdot Load_k \end{aligned} \quad (11)$$

*1 近似時の誤差により、たとえば 2.0 GHz における値は 1.0×10^3 とはなっていない。

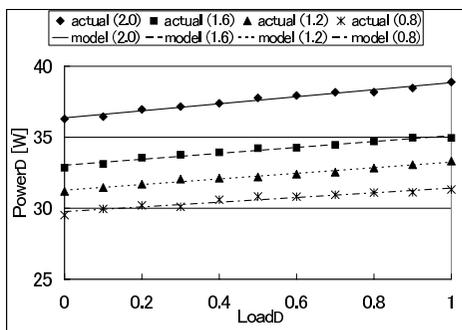


図 3 負荷と消費電力の関係 (Disk-bound)

Fig. 3 Power consumption to load (Disk-bound).

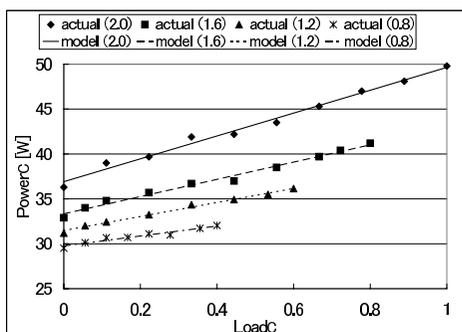


図 4 負荷と消費電力の関係 (CPU-bound)

Fig. 4 Power consumption to load (CPU-bound).

4.1.1 項と同様に、図 3、図 4 に Disk-bound, CPU-bound の場合のそれぞれにおける、実測値およびモデルから近似した関数を示す。4.1.1 項と同様に、実測値がモデルとよく一致していることから、消費電力のモデルの正しさが確認できる。また、表 1 に、本研究の評価環境における $base_power$ および $grad_k$ の値を示す。

4.2 1 ノード上で両種類を並行処理する場合

4.1 節では、単一の種類のリクエストを処理する場合を考えた。しかし、ページの種類によってボトルネックが異なるため、1 ノード上でそれぞれの種類を並行処理することによ

表 1 各周波数におけるベース電力と比例係数

Table 1 Base power consumption and coefficient at each frequency.

frequency [GHz]	base_power [W]	grad _D [W]	grad _C [W]
2.0	36.7	2.5	12.8
1.6	33.2	2.1	9.8
1.2	31.3	2.0	7.8
0.8	29.7	1.7	5.6

て電力効率が改善される場合があると考えられる。そこで本節では、そのような場合の性能と電力のモデリングについて述べる。

4.2.1 処理性能のモデリング

1 ノード上で両種類を並行処理する場合、CPU とメモリバスを共有するためリソース競合による性能低下が起こると考えられる。そこでまず、CPU の処理能力に関するモデルを立てる。CPU の処理能力を α とし、それが各種類の処理に割り当てられると考える。また、 β, γ を各々、 $Load_D = 1$ の処理、および $Load_C = 1$ の処理に必要な CPU の処理能力であるとする。 α, β, γ は周波数に依存すると考えられる。このとき、式 (12) が成り立つ必要がある。

$$\alpha(f) \geq \beta(f) \cdot Load_D + \gamma(f) \cdot Load_C \quad (12)$$

両辺を α で割って

$$1 \geq \frac{\beta(f)}{\alpha(f)} \cdot Load_D + \frac{\gamma(f)}{\alpha(f)} \cdot Load_C \quad (13)$$

次に、メモリバスのバンド幅に関するモデルを立てる。考え方は CPU の場合と同様である。すなわち、メモリバスのバンド幅を δ とし、それが各種類の処理に割り当てられると考える。また、 ϵ, ζ は各々、 $Load_D = 1$ の処理、および $Load_C = 1$ の処理に必要なメモリバンド幅である。 δ, ϵ, ζ は周波数に依存すると考えられる。このとき、式 (14) が成り立つ必要がある。

$$\delta(f) \geq \epsilon(f) \cdot Load_D + \zeta(f) \cdot Load_C \quad (14)$$

両辺を δ で割って

$$1 \geq \frac{\epsilon(f)}{\delta(f)} \cdot Load_D + \frac{\zeta(f)}{\delta(f)} \cdot Load_C \quad (15)$$

式 (13) と式 (15) の不等式を同時に満たす $Load_D, Load_C$ が 1 ノード上で並行処理可能であると考えられる。これまでと同様に、図 5 に実測値および、モデルから近似した関数

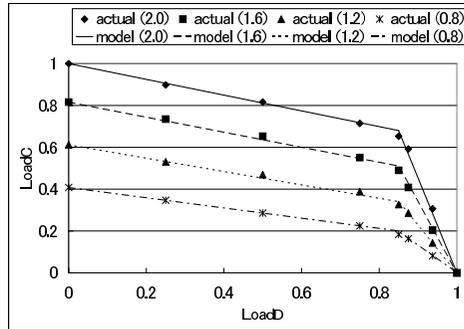


図 5 並行処理する場合の $Load_D$ に対する $Load_C$ の最大値
 Fig. 5 The maximum $Load_C$ to $Load_D$ in concurrent processing.

表 2 各周波数における $\frac{\beta}{\alpha}$, $\frac{\gamma}{\alpha}$
 Table 2 $\frac{\beta}{\alpha}$ and $\frac{\gamma}{\alpha}$ at each frequency.

frequency [GHz]	$\frac{\beta}{\alpha}$	$\frac{\gamma}{\alpha}$
2.0	0.38	1.00
1.6	0.43	1.25
1.2	0.51	1.67
0.8	0.59	2.50

表 3 各周波数における $\frac{\epsilon}{\delta}$, $\frac{\zeta}{\delta}$
 Table 3 $\frac{\epsilon}{\delta}$ and $\frac{\zeta}{\delta}$ at each frequency.

frequency [GHz]	$\frac{\epsilon}{\delta}$	$\frac{\zeta}{\delta}$
2.0	1.00	0.22
1.6	1.00	0.29
1.2	1.00	0.44
0.8	1.00	0.75

を示す。図から、実測値およびモデルがよく一致していることが分り本モデルが有効であることが確認できるとともに、関数が外に凸になっていることから両種類のリクエストを並行処理した方が性能的に有利であることも分かる。

また、本研究の評価環境における $\frac{\beta}{\alpha}$, $\frac{\gamma}{\alpha}$ の値を表 2 に示し、 $\frac{\epsilon}{\delta}$, $\frac{\zeta}{\delta}$ の値を表 3 に示す。

4.2.2 消費電力のモデリング

1 ノードで並行処理した場合の消費電力は、ベース電力 $base_power$ と、Disk-bound な負荷の処理により増加する電力、そして CPU-bound な負荷の処理により増加する電力の和と考えられる。すなわち、この場合の消費電力を $Power_{MIX}$ と定義すると、

$$\begin{aligned}
 Power_{MIX}(f, Load_D, Load_C) &= base_power(f) \\
 &+ grad_D(f) \cdot Load_D \\
 &+ grad_C(f) \cdot Load_C
 \end{aligned} \tag{16}$$

と表せる。なお、本研究の評価環境における式 (16) の $base_power$, $grad_D$, $grad_C$ は表 1 に示した値と同一である。

5. 最適なクラスタ構成の導出

本章では、4 章で構築したモデルに基づき最適なクラスタ構成 (ノード数, 周波数) を導出するためのアルゴリズムについて述べる。

5.1 負荷分散手法

クラスタ型のウェブサーバでは、多数のリクエストがクライアントから送信された場合、3.1 節で述べたようにフロントエンドサーバが複数のバックエンドサーバにリクエストを振り分けることにより処理を行うことが一般的である。このとき、各処理ノードの最適な周波数はフロントエンドサーバから振り分けられる負荷に応じて決まるため、どのようにリクエストを振り分けるのかによって処理ノード全体での消費電力は異なる。したがって、全体の消費電力を最小化する負荷分散手法を用いることが重要である。以下において、負荷を均等に分散させる場合に電力最小となることを説明する。

はじめに、負荷を処理ノードへ均等に分散させる場合を考えると、すべての処理ノードの最適な周波数は等しく、したがって消費電力も等しい。これに対して負荷を不均等に分散させる場合を考えると、均等に分散させる場合に比べて負荷が少なくなるノードは周波数を下げる機会が生じるため消費電力が減少するが、一方で負荷が多くなるノードは周波数を上げる必要が生じるため消費電力が増加する。ここで、1 ノードの消費電力と負荷の関係について考える。CPU の消費電力は電源電圧の 2 乗および周波数に比例するが、電源電圧が周波数に関して単調増加関数であるために、CPU の消費電力は周波数に関して下に凸な関数となる。1 ノードの消費電力はこの CPU の消費電力に、周波数に依存しない定数の電力を加えるだけであり、結局周波数に関して下に凸な関数となる。一方、処理性能は一般に周波

数に関して上に凸な関数である．これは逆に考えると，周波数は負荷に関して下に凸な関数であるといえる．これらのことから，消費電力は負荷に関して下に凸な関数となるため，負荷が少なくなるノードでの電力減少量よりも，負荷が多くなるノードでの電力増加量の方が必ず大きくなる．したがって，負荷を均等に分散する場合は電力最小となる．

5.2 クラスタ構成の導出のアルゴリズム

前節で述べたように，複数ノードで処理する場合には，負荷を処理ノードに均等に振り分ける場合が最適である．これをふまえると，最適なクラスタ構成を選択するアルゴリズムは，以下のようになる．

- (1) 予想されるリクエストの状況から，Disk-bound，CPU-bound のそれぞれについて，4.1.1 項のモデルに基づき負荷の値 ($Load_D$ および $Load_C$) を求める．
- (2) ノード数を i ($1 \leq i \leq$ 最大ノード数) としたときの 1 ノードあたりの負荷を， $Load_D$ ， $Load_C$ を i で割ることで求める．
- (3) 求めた 1 ノードあたりの各種の負荷を，式 (13) と式 (15) に代入し，両方の式をもとに満す最小の周波数 f_i を求める．
- (4) 上記の (1) で仮定したノード数と，(3) で得られた周波数を式 (16) への入力とし，1 ノードあたりの電力を求める．
- (5) 1 ノードあたりの電力を i 倍することでサーバ全体の電力を求める．
- (6) 最小の消費電力を与える (i, f_i) が最適な構成となる．

なお，ノードごとに単一の種類のリクエストを処理する場合には，片方の負荷の値を 0 として上記のアルゴリズムを適用することで，最適な構成を求めることが可能である．

6. 評価

6.1 評価環境

評価には，Intel PentiumM-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 4 に，PentiumM-760 プロセッサにおいて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 5 に示す．このノードを，8 台 Gigabit 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，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 httpperf 0.8¹⁹⁾ を用いた．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，HTTP redirection²¹⁾ などを用いて負荷分散を行い複数の

表 4 Intel Pentium M 760 プロセッサの周波数と電源電圧の関係
Table 4 Clock and voltage setting for an Intel Pentium M 760 processor.

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 5 評価環境

Table 5 Specification of a node.

M/B	Commell LV673 - i915GM + ICH6M chip-set - Gb Ethernet x 2 12 V CPU 電源駆動
Processor	Intel Pentium M 760 (Max 2 GHz, FSB533 MHz)
Memory	DDR2-SDRAM 1 GB
HDD	80 GB, 7200 rpm SATA 3.0 GB/s Seek Time 8.8 ms

ノードで処理することが必要となるが，5.1 節で述べた手法すなわち処理ノードに負荷を均等に振り分ける手法により負荷分散を行うことを仮定する．なお，公平性のために，次節で述べる 4 つの手法すべてにおいてこの負荷分散手法を適用することとした．ただし，評価においては，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，クライアントはあらかじめ決められたサーバに分散してリクエストを送信することとした．また，Disk-bound なページは HTML 文書とし，CPU-bound 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生成するものである．生成時間は，URL で指定する乱数の個数によって変化可能である．なお，満たすべき応答時間の制約は，Disk-bound であるページ，CPU-bound であるページの場合ともに 200 ms を仮定する．

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．評価に用いたボードは 12 V の ATX 電源駆動であり，各ノードの 12 V 電線に流れる電流を測定し，電圧 12 V を乗じることで電力を求める．測定対象はサーバのみであり，本評価では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．

6.2 評価方法

本評価では、提案手法の効果を調べるために、以下に示す 4 つの手法について比較を行う。

- *Always-Max*: ノード数, 周波数ともにつねに最大の構成で動作
- *Only-DVFS*: ノード数はつねに最大であるが, 周波数は応答時間制約を満たす範囲で最も低い周波数をモデルに基づき選択
- *Only-Node*: 周波数はつねに最大であるが, ノード数は応答時間制約を満たす範囲で最も少ないノード数をモデルに基づき選択
- *Proposed*: 本論文の提案手法である, モデルに基づいてノード数と周波数を最適化

本論文では, *Disk-bound* と *CPU-bound* のそれぞれの負荷として $Load_D$, および $Load_C$ の値を入力として評価を行うが, それらは文献 20) で例示されているウェブサーバの *Disk-bound* と *CPU-bound* それぞれの 1 日の負荷変動を参考に, それと近い変動となるように, 10 分ごとの $Load_D$ および $Load_C$ の値をプロットしたものをを用いる. なお, 実際には文献 20) に示されている負荷変動は 1 ノードのリクエストレートであり, 1 ノードでも十分に処理できてしまう値であるため, 文献 20) の最大リクエストレートが, 本評価環境におけるクラスタ型ウェブサーバで処理可能な最大負荷となるように変換している. なお, リクエストの負荷は 10 分ごとに変化するが, リクエストの時間間隔はポアソン分布に従うものとする. また, ノード数および周波数は, 各評価手法のポリシーに従い, この 10 分間隔で構成を最適化する. なお, ノード数変更の際に, ノードの電源オフの実装としてサスペンドを用いれば, オン/オフの際にかかるオーバーヘッドは約 1 秒程度であり, 10 分間隔でノード数を変更する際のオーバーヘッドはほとんど無視できると考えられる.

6.3 評価結果

はじめに, 負荷の種類ごとに処理するノードが異なる場合について評価を行う. 8 ノード中 4 ノードを *Disk-bound* リクエストの処理に用い, 残りの 4 ノードを *CPU-bound* リクエストの処理に用いるものとする.

図 6 に *Disk-bound* リクエストの場合, 図 7 に *CPU-bound* リクエストの場合の 1 日の消費電力の変動を示す. 横軸は時刻を表し, 縦軸 (左) が消費電力, 縦軸 (右) が負荷を表している. なお, 提案手法のクラスタ構成については, 5.2 節の最後で述べたように, 片方の負荷の値を 0 としてアルゴリズムを適用する. 処理性能のモデル式は式 (13) または式 (15) であり, たとえば *Disk-bound* な負荷のみの場合は式 (15) における右辺第 2 項を 0 としたものが性能のモデル式であり, 表 3 に示す値を用いている.

図 6 の *Disk-bound* リクエストの場合を見ると, *Always-Max* の場合が最も消費電力が

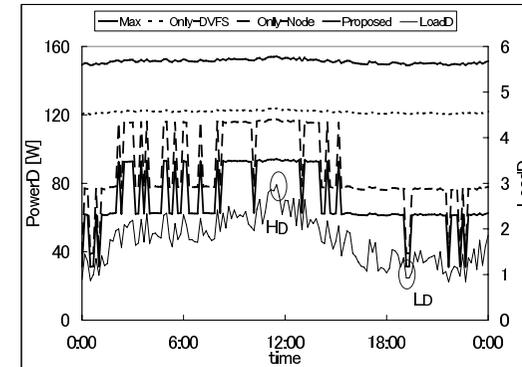


図 6 負荷変動と各手法における消費電力 (*Disk-bound*)
Fig. 6 Power consumption to load variation for each policy (*Disk-bound*).

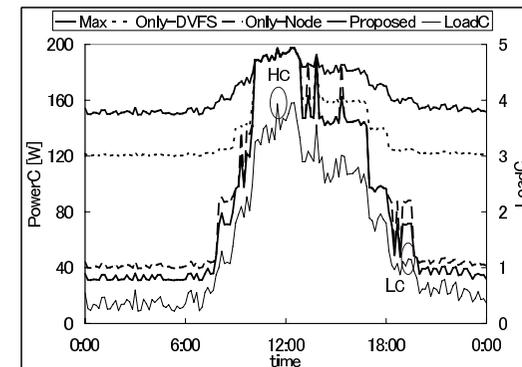


図 7 負荷変動と各手法における消費電力 (*CPU-bound*)
Fig. 7 Power consumption to load variation for each policy (*CPU-bound*).

大きく, 次に *Only-DVFS* の電力が大きい. また, それらは負荷に依存せずほぼ一定の消費電力となっていることが分かる. これは, *Disk-bound* リクエストの負荷による CPU の電力増分が少ないためである. したがって, *Always-Max* と比較した場合の *Only-DVFS* の電力削減効果は, CPU のベース電力の差によるものである. 一方, ノード数を変更可能な *Only-Node* では, さらにベース電力が削減できるため, 消費電力削減効果が大きい. 本論文

表 6 図 6 中の点 H_D および L_D における各手法のとりシステム構成と消費電力Table 6 System configuration and power consumption for each policy at H_D and L_D in Fig. 6.

手法	H_D ($Load_D = 3.0$)		L_D ($Load_D = 0.9$)	
	構成	電力	構成	電力
Always-Max	(4, 2.0)	153.6	(4, 2.0)	149.1
Only-DVFS	(4, 0.8)	123.6	(4, 0.8)	120.4
Only-Node	(3, 2.0)	117.3	(1, 2.0)	39.0
Proposed	(3, 0.8)	94.2	(1, 0.8)	31.3

表 7 図 7 中の点 H_C および L_C における各手法のとりシステム構成と消費電力Table 7 System configuration and power consumption for each policy at H_C and L_C in Fig. 7.

手法	H_C ($Load_C = 3.8$)		L_C ($Load_C = 1.2$)	
	構成	電力	構成	電力
Always-Max	(4, 2.0)	197.2	(4, 2.0)	161.6
Only-DVFS	(4, 2.0)	197.2	(4, 0.8)	125.3
Only-Node	(4, 2.0)	197.2	(2, 2.0)	88.2
Proposed	(4, 2.0)	197.2	(2, 1.2)	71.6

の提案手法である Proposed の場合は、他の比較手法に比べてつねに低い電力消費であり、ノード数と周波数をともに最適化することの重要性がうかがえる。具体的に、負荷が比較的大きい点 H_D ($Load_D = 3.0$) および小さい点 L_D ($Load_D = 0.9$) を選び、これらにおいて各手法が選択するシステム構成と消費電力を表 6 に示す。システム構成は (ノード数, 動作周波数) と表記する。動作周波数の単位は GHz であり、消費電力の単位は W である。

表 6 から、 H_D , L_D どちらの場合も提案手法が最小のノード数と最低の周波数を選択しており、最も低消費電力化を実現できていることが確認できる。

図 7 の CPU-bound リクエストの場合においても、消費電力は上記の Disk-bound なものと傾向が近い。ただし、負荷に依存した CPU の電力の増分が大きいため、Always-Max, Only-DVFS とともに、負荷につれて電力が変化することが分かる。また、負荷が非常に高い場合には、最大ノード数および周波数を選択する必要がある。どの手法を用いたとしても同じ消費電力となっている。一方、Proposed は負荷が低い場合には最も消費電力を削減できている。具体的に、 H_C ($Load_C = 3.8$) および L_C ($Load_C = 1.2$) において各手法が選択するシステム構成と消費電力を表 7 に示す。

表 7 より、 H_C においては、負荷がシステムの最大処理性能に近いために提案手法を含めすべての手法で最大ノード数、最高周波数という構成をとっているが、 L_C においては、提案手法が最も低い電力消費の構成を選択することができていることが分かる。Only-DVFS

表 8 各手法における 1 日の電力量

Table 8 Energy consumption in a day for each policy.

手法	電力量 [KWh]		
	Disk-bound	CPU-bound	MIX
Always-Max	3.65	3.99	7.65
Only-DVFS	2.94	3.35	6.12
Only-Node	2.17	2.14	3.20
Proposed	1.74	2.00	2.92

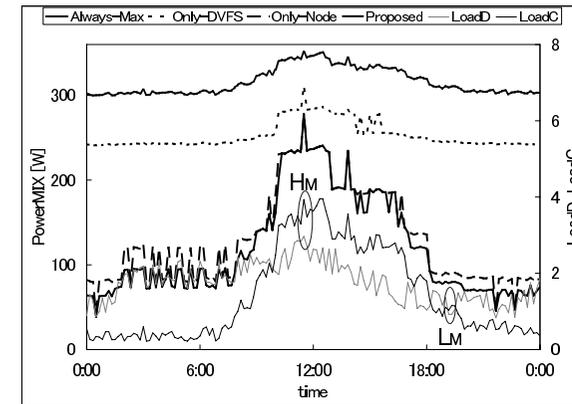


図 8 負荷変動と各手法における消費電力 (MIX)

Fig. 8 Power consumption to load variation for each policy (MIX).

は提案手法よりも低い周波数を選択しているが、ノード数を減らすことでベース電力を削減する方が電力削減効果が大きいため提案手法の方が電力消費が小さい。

表 8 に、各手法における 1 日を通しての電力量を示す。提案手法を用いることで、つねに最大の構成で動作させる場合 (Always-Max) に比べて Disk-bound リクエスト、および CPU-bound リクエストの場合において、それぞれ 52%, 50% 消費電力量を削減できていることが分かる。

次に、1 ノード上で両種類の負荷を並行処理する場合の評価結果について述べる。図 8 に各手法の消費電力を示す。横軸、縦軸は図 6、図 7 と同一である。

図 8 から、この場合も周波数とノード数の両者を最適化することで電力削減の効果が大きいことが分かる。表 8 に、1 ノード上で両種類の負荷を並行処理した場合の 1 日を通し

表 9 図 8 中の点 H_M および L_M における各手法のとりシステム構成と消費電力Table 9 System configuration and power consumption for each policy at H_M and L_M in Fig. 8.

手法	H_M		L_M	
	構成	電力	構成	電力
Always-Max	(8, 2.0)	350.8	(8, 2.0)	310.7
Only-DVFS	(8, 1.2)	310.4	(8, 0.8)	245.7
Only-Node	(6, 2.0)	278.0	(2, 2.0)	90.4
Proposed	(6, 2.0)	278.0	(2, 1.6)	79.7

での電力量も示している。つねに最大の構成で動作させる場合 (Always-Max) の消費電力量は 7.65 KWh であるのに対して、提案手法を用いた場合の消費電力量は 2.92 KWh であり、提案手法を用いることで、つねに最大の構成で動作させる場合に比べて 62%消費電力量を削減できることが分かる。具体的に、 H_M ($Load_D = 3.0$, $Load_C = 3.8$) および L_M ($Load_D = 0.9$, $Load_C = 1.2$) における各手法のとりシステム構成と消費電力を表 9 に示す。システム構成の表記方法および動作周波数や消費電力の単位は表 6 と同様である。

表 9 より、 H_M , L_M いずれの場合も提案手法が最も低い電力消費である構成を選択できることが分かる。提案手法よりも Only-DVFS の方が低い周波数を選択しているが、ノード数が大きいベース電力を多く消費し、提案手法の方が消費電力が低い構成となる。

表 8 の Proposed の場合において、負荷の種類ごとに処理するノードが異なる場合の合計の消費電力量と、1 ノード上で異なる種類の負荷を並行処理する場合の電力量を比較すると、前者の電力量が 3.74 (= 1.74 + 2.00) KWh、後者の電力量が 2.92 KWh と、1 ノード上で並行処理する場合の方が消費電力量が少ない。提案するモデルにより、1 ノード上で並行処理する場合にも、任意のリクエストが来た際に最適なノード数と周波数の構成を決定することができる。この点は、従来の手法に比べ、提案手法が非常に有用な点であると考えられる。

7. まとめと今後の課題

本研究では、計算機クラスタ構成によるウェブサーバの電力効率を最適化する手法を提案した。具体的には、Disk-bound および CPU-bound の 2 種類の負荷状況に応じてノード数および各ノードの CPU の動作周波数を変化させることにより、応答時間制約を満たしつつ最も消費電力の小さい構成となるようにシステムを最適化するものである。その実現上必要となる計算機クラスタベースのウェブサーバの性能・電力のモデルを構築し、実際のクラスタシステムに適用した際の電力削減効果について評価を行った。評価の結果、1 ノード上で

両種類 (Disk-bound, CPU-bound) の負荷を並行処理する場合において、本論文の提案手法は従来手法 (ノード数と周波数がつねに最大である手法) に比べて約 62%の消費エネルギー削減効果がみられた。

今後の課題としては、ヘテロジニアス型ウェブサーバシステムの低消費電力化があげられる。ヘテロジニアスシステムでは、ノードによって性能・電力特性が異なるため、負荷を均等に処理ノードへ振り分ける場合が最適であるという本論文の負荷分散手法を適用することはできない。そこで今後は、ヘテロジニアスシステムにも適用可能な最適構成選択アルゴリズムを考案し、提案手法の効果について評価・検討していく予定である。

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「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」、および科学研究費補助金 (萌芽研究 18650010, 若手研究 (B) 19700038), 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) Hsu, C. and Feng, W.: A Feasibility Analysis of Power Awareness in Commodity-Based High-Performance Clusters, *Proc. Cluster 2005* (Sep. 2005).
- 2) 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).
- 3) Ge, R., Feng, X. and Cameron, K.: Improvement of Power Performance Efficiency for High-End Computing, *Proc. Workshop on HP-PAC 2005* (Apr. 2005).
- 4) 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).
- 5) 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).
- 6) 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).
- 7) 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).
- 8) Hsu, C. and Feng, W.: A Power-Aware Run-Time System for High-Performance Computing, *Proc. SC'05* (Nov. 2005).
- 9) IBM and Lawrence Livermore National Laboratory: An Overview of the Blue-Gene/L Supercomputer, *Proc. Supercomputing 2002* (Nov. 2002).

- 10) Nakashima, H., et al.: MegaProto: 1 TFlops/10kW Rack Is Feasible Even with Only Commodity Technology, *Proc. Supercomputing 2005* (Nov. 2005).
- 11) 大谷, 池田, 佐々木, 近藤, 中村: 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, 情報処理学会研究報告, ARC-172 HPC-109, pp.91-96 (2007).
- 12) Elnozahy, M., Kistler, M. and Rajamony, R.: Energy Conservation Policies for Web Servers, *Proc. 4th USITS*, pp.99-112 (Mar. 2004).
- 13) Sharma, V., Thomas, A., Abdelzaher, T., Skadron, K. and Zhijian, L.: Power-aware QoS Management in Web Servers, *Proc. 24th RTSS*, pp.63-72 (Dec. 2003).
- 14) Rajamani, K. and Lefurgy, C.: On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters, *Proc. ISPASS 2003*, pp.111-122 (Mar. 2003).
- 15) Rusu, C., Ferreira, A., Scordino, C. and Watson, A.: Energy-Efficient Real-Time Heterogeneous Server Clusters, *Proc. 12th RTAS*, pp.418-428 (Apr. 2006).
- 16) COMMELL: Mini-ITX Express Motherboard LV-673 Datasheet.
<http://www.commell.com.tw/Product/SBC/LV-673.HTM>
- 17) Intel: Pentium M Processor on 90nm Process with 2-MB L2 Cache Datasheet (Jan. 2005).
- 18) Crovella, M. and Bestavros, A.: Self-similarity in World Wide Web Traffic: Evidence and Possible Causes, *1996 SIGMETRICS Conference on Measurement and Modeling of Computer Systems* (May 1996).
- 19) Mosberger, D. and Jin, T.: httpperf: A Tool for Measuring Web Server Performance, *SIGMETRICS 1st Workshop on Internet Server Performance*, pp.59-67, ACM (June 1998).
- 20) Bohrer, P., Elnozahy, E.N., Keller, T., Kistler, M., Lefurgy, C., McDowell, C. and Rajamony, R.: *The case for power management in web servers*, Kluwer Academic Publishers (2002).
- 21) Cardellini, V., Colajanni, M. and Yu, P.S.: Redirection Algorithms for Load Sharing in Distributed Web-server Systmes, *Proc. 19th ICDCS*, pp.528-535 (June 1999).

(平成 19 年 10 月 10 日受付)

(平成 20 年 1 月 31 日採録)



大谷 貴胤 (学生会員)

2007 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。



佐々木 広 (正会員)

2003 年東京大学工学部計数工学科卒業。2005 年同大学大学院情報理工学系研究科修士課程修了。2008 年同大学院工学系研究科博士課程修了。博士 (工学)。2008 年より、東京大学先端科学技術研究センター特任助教。



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, 2004 年東京大学先端科学技術研究センター特任助手, 2007 年同特任准教授を経て, 現在電気通信大学情報システム学研究科准教授。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，1996年東京大学先端科学技術研究センター助教授，2008年より東京大学大学院情報理工学系研究科准教授．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，ディペンドブルコンピューティング，デ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．
