様式 C-19

科学研究費補助金研究成果報告書

平成２１年　６月　１日現在

研究種目：若手研究（Ｂ）
研究期間：２００７～２００８
課題番号：１９７００００３８
研究課題名（和文）クラスタ型並列サーバシステムのための細粒度適応型電力制御手法

研究課題名（英文）A Study on Fine-Grain Adaptive Power Management for Clustered Sever Systems

研究代表者
近藤　正章（MASAAKI KONDO）
電気通信大学・大学院情報システム学研究科・准教授
研究者番号：３０３７６６６０

研究成果の概要：
インターネットサービスを提供するサーバ計算機システムの電力増加問題への対処として、システムの性能・電力のモデル化を行い、要求負荷に応じて各コンポーネントの速度や電源のオン・オフを細粒度に制御することでシステム構成を最適化し、サーバ計算機システムの省電力化を達成する手法を開発した。これにより、サービス品質を維持しつつサーバシステムの消費電力量を大幅に削減できるため、電力増加問題への有効な解決策になり得ると期待される。

交付額

（金額単位：円）

	直接経費	間接経費	合　計
２００７年度	2,400,000	0	2,400,000
２００８年度	900,000	270,000	1,170,000
年度			
年度			
年度			
総　計	3,300,000	270,000	3,570,000

研究分野：総合領域
科研費の分科・細目：情報学・計算機システム・ネットワーク
キーワード：並列分散システム、インターネットサーバー、低消費電力・省エネ、クラスタコンピューティング、インターネットデータセンター、グリーンＩＴ

１．研究開始当初の背景

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、電子メールの送受信、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が享受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動状態にある。また、大規模なサーバシステムを統合的に管理・運用するためのインターネットデータセンター(IDC)の数もここ数年急増しており、このことからもインターネットサービスの需要の高さがうかがえる。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、それらのサービス要求を適切に処理するために、消費電力を削減しつつ、サーバ計算機システムの処理性能を向上させることは必要不可欠である。

２．研究の目的

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、サーバ用計算機はネットワークで結合したクラスタ型並列計算機として構築されることが多い。これまでは、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ことで、システムの処理性能を向上させることが可能であった。しかし現在では、1)ブレードサーバなどの高密度実装の結果、電力密度増大により設置体積あたり熱密度が増加し、各ノードをより高性能なものに拡張することが難しい、2)データセンターの電力供給・冷却能力不足により、ノード数を追加することも限界に近づいている、といった問題により単純に性能を向上させることが難しくなっている。また、二酸化炭素排出量の削減などの環境負荷低減の要求もあり、処理性能が重視されるサーバ用計算機システムにおいても、消費電力を考慮した開発や運用が特に重要となっている。

そこで、要求負荷に応じて速度(周波数)や電源のオン・オフを制御することで電力を管理しつつ、サービス品質維持しながらシステム構成を最適化し、サーバシステムの省電力化を図る「クラスタ型並列サーバシステムのための細粒度適応型電力制御手法」を開発することが目的である。

３．研究の方法

クラスタ型サーバ計算機システムにおいて、CPU の周波数やディスク装置／ネットワークデバイス使用の有無をサーバアプリケーションやサービス要求の状況に応じて動的に制御し、システムの要求性能を維持しつつ消費電力を削減するためには、(1)細粒度に各コンポーネントの電力を制御できるサーバ用クラスタシステムの構築、(2)各コンポーネントの制御を行った際の、性能と消費電力のモデリング、(3)各種状況に応じて最も効率的にサービスの処理ができるよう各コンポーネントを制御するアルゴリズム、といった要素技術を開発する。さらに、本手法の有効性を示すために、(4)各要素技術を統合し、システムの性能・電力を評価する。

４．研究成果

(1)負荷とシステム性能・電力のモデリング

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。

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。従って、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。以下では、まず、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る。

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には様々なものが存在し、そ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、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。ここで、ページの大きさとは、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。

単位時間当たりに、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たと仮定する。各 Disk-bound なページのページサイズを $page\_size^i$ (i = 0, 1, 2, … , m － 1)、リクエストレートを num^i_D (D:Disk-bound)とすると、総リクエストレート reqD は

$$reqD = \sum_{i=0}^{m-1} num_D^i \qquad (1)$$

と表せる。また、平均ページサイズ xD は

$$xD = \frac{\sum_{i=0}^{m-1} num_D^i \bullet page\_size^i}{reqD} \qquad (2)$$

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たと仮定する。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, … , n － 1)、リクエストレートを num^j_C (C:CPU-bound)とすると、総リクエストレート reqC は

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

と表せる。また、平均生成時間 xC は

$$xC = \frac{\sum_{j=0}^{n-1} num_C^j \bullet page\_size^j}{reqC} \qquad (4)$$

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

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

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ヘッド時間、BW_k は各種類の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、単位時間当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、最適化対象は CPU の周波数とノード数であり、例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わす。

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、例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毎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。

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。この場合のバンド幅は、ある基準のシステム(例えば、対象サーバ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、単位時間当たりにどれだけの生成時間を持つページを処理可能であるかを示す量となる。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。

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

応答時間制約を L_k とすると、満たすべき条件は

$$r_k \le L_k \qquad (7)$$

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。

$$req_k \le \frac{s_k(x_k, f)}{\frac{x_k}{BW_k(f)} + OH_k(f)} - \frac{1}{L_k} \qquad (8)$$

② 負荷

前節で述べたように、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。例えば Disk-bound なページの負荷は、システムに対して「単位時間当たりに、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。また、負荷を x_k と req_k の組み合わせとしてではなく、より扱いやすくするために、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)。

前節で述べたように、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。

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

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。

$$Load_k = \frac{req_k}{max\_req_k(x_k)} \qquad (10)$$

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、例えば $Load_D$=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、これ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。

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。リクエストが無い待機中の場合でも、電源が On の状態のサーバは常に電力を消費する。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、$Load_k$ 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。

$$\begin{aligned} &Power_k(f_k, Load_k) \\ &= base\_power(f) + \Delta Power_k(f_k, Load_k) \\ &= base\_power(f) + grad_k(f) \bullet Load_k \end{aligned} \quad (11)$$

(2) 1 ノード上で両種類を並行処理する場合

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 $Load_D$、$Load_C$ の処理に必要となる 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。この時、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 $Load_D$、$Load_C$ の処理に必要となる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。この時、式(14)が成り立つ必要がある。

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

両辺を δ で割って

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

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。

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、

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

と表せる。

(3) 最適なクラスタ構成の導出

ここでは、前節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を述べる。

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、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。この時、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。従って、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、従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。従って、負荷を均等に分散する場合が電力最小となる。

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、上述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 ($1 \leqq i \leqq$ *最大ノード数*) とした時の 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類の負荷を、式(13)と式(15)に代入し、両方の式を共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。

(4) 評価

上述のモデル、およびクラスタ構成導出のアルゴリズムを用いた場合のウェブサーバシステムの電力削減効果について評価を行った。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た。

また、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った。

- *Always-Max*: ノード数、周波数ともに常に最大の構成で動作
- *Only-DVFS*: ノード数は常に最大であるが、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*: 周波数は常に最大であるが、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*: 本研究での提案手法であるモデルに基きノード数と周波数を最適化

図 1 負荷変動と消費電力の関係

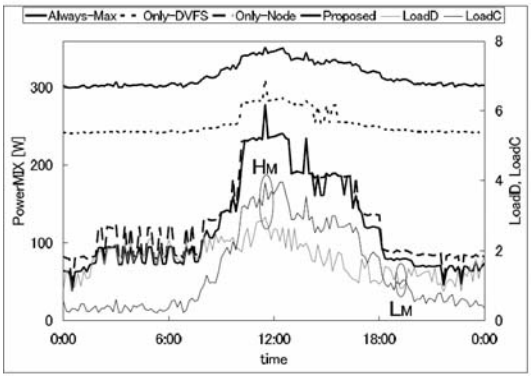

図 1 に Disk-bound および CPU-bound の両リクエストを 1 ノードで並行処理する場合の、あるリクエスト状況における一日の消費電力の変動を示す。横軸は時刻を表し、縦軸（左）が消費電力、縦軸（右）が負荷を表している。

図を見ると、Always-Max の場合が最も消費電力が大きく、次に Only-DVFS の電力が大きい。ノード数を変更可能な Only-Node では、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。提案手法である Proposed の場合は、他の比較手法に比べて常に低い電力消費であり、ノード数と周波数を共に最適化することの重要性が伺える。

表 1 各種法の 1 日の消費電力量

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

表 1 に、図 1 の状況における 1 日を通しての電力量の値を示す。提案手法を用いることで、常に最大の構成で動作させる Always-Max に比べて 62%消費電力量を削減できることがわかる。この結果より、将来的な課題であるインターネットサービスを提供するための消費エネルギー増大の問題に対し、本研究課題で得られた技術、および知見は一つの有効な解決策となり得ると考えられる。

５．主な発表論文等

〔雑誌論文〕（計４件）

1. 近藤正章, 佐々木広, 中村宏, “トラクションコントロール実行：CMP 向けプロセス実行制御方式の提案”, 情報処理学会論文誌：コンピューティングシステム, Vol.1, No.2, pp.111-123, 2008.（査読有り）
2. 大谷貴胤, 佐々木広, 近藤正章, 中村宏,

“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”，情報処理学会論文誌：コンピューティングシステム，Vol.1，No.1，pp.120-132，2008.（査読有り）
3. 金井遵，佐々木広，近藤正章，中村宏，天野英晴，宇佐美公良，並木美太郎，“性能予測モデルの学習と実行時性能最適化機構を有する省電力化スケジューラ”，情報処理学会論文誌, Vol. 49 (SIG2), pp.20-36, 2008.（査読有り）
4. 近藤正章，中村宏，“CMP 向け動的電源電圧・周波数制御手法”，情報処理学会論文誌，Vol.48 (SIG13)，pp.80-91，2007.（査読有り）

〔学会発表〕（計４件）
1. 大谷貴胤，佐々木広，近藤正章，中村宏，“ヘテロ構成を考慮した Web サーバ用クラスタシステムの性能と電力のモデリング”，情報処理学会計算機アーキテクチャ研究会，2008 年 8 月 7 日，佐賀市.
2. 佐々木広，近藤正章，中村宏，“CMP の統計的モデリングによる実行時最適化手法”，情報処理学会計算機アーキテクチャ研究会，2008 年 8 月 5 日，佐賀市.
3. Ryo Watanabe, Masaaki Kondo, Hiroshi Nakamura, and Takashi Nanya, “Power Reduction of Chip Multi-Processors using Shared Resource Control Cooperating with DVFS”, XXV International Conference on Computer Design (ICCD-2007), 2007 年 10 月 10 日, Lake Tahoe, CA.
4. 大谷貴胤，佐々木広，近藤正章，中村宏，“Web サーバ用計算機クラスタの性能と電力のモデリングに関する研究”，情報処理学会計算機アーキテクチャ研究会，2007 年 8 月 2 日，旭川市.

６．研究組織
(1)研究代表者
近藤　正章（MASAAKI KONDO）
電気通信大学・大学院情報システム学研究科・准教授
研究者番号：３０３７６６６０

(2)研究分担者
（　　）

研究者番号：

(3)連携研究者
（　　）

研究者番号：