

オンライントランザクション処理における Dynamic Voltage and Frequency Scaling を 用いたアプリケーション指向省電力化手法

早水 悠登^{†1} 合田 和生^{†1}
中野 美由紀^{†1} 喜連川 優^{†1}

近年、情報技術分野においても低炭素化が求められ、なかでも消費電力増加が著しいデータセンタの省電力化は急務である。電子商取引などの重要な社会インフラを支えるオンライントランザクション処理 (OLTP) はデータセンタの主たるアプリケーションであり、OLTP が稼動するサーバの省電力化は重要な課題である。しかし、OLTP はシステムの可用性や性能への要求が厳しいために省電力化が難しく、これまで有効な省電力手法はほとんど提案されてこなかった。

本論文では OLTP の省電力化をするために、プロセッサの Dynamic Voltage and Frequency Scaling (DVFS) 機能を用いたアプリケーション指向省電力手法を提案する。この手法では DVFS を利用することでシステムを停止することなく省電力化が可能である。またアプリケーションの実行時性能情報を活用して DVFS を制御するため、アプリケーションの性能要求に応えつつ省電力化を行うことができる。評価実験においては TPC-C ベンチマークを用いて、実際のサーバにおける負荷のもとで本提案手法の省電力効果を検証した。その結果、応答時間指向 DVFS ポリシーを用いた場合には 10.0W (全体の 6.4%)、スループット指向 DVFS ポリシーを用いた場合には 11.8W (全体の 7.5%) の消費電力削減効果が確認された。

Application-aware Power Saving Method for Online Transaction Processing using Dynamic Voltage and Frequency Scaling

YUTO HAYAMIZU,^{†1} KAZUO GODA,^{†1}
MIYUKI NAKANO^{†1} and MASARU KITSUREGAWA^{†1}

Recently Information Technology has been required to achieve low-carbon technology. Power consumption in data centers has been growing remarkably

recent years, and the demand for power saving of datacenters is particularly increasing. Online transaction processing (OLTP) is one of main important applications on data centers and OLTP becomes a key technology for social infrastructures such as e-commerce, e-trading and so on. Thus, for saving power of data centers, power saving of OLTP on large number of servers is an essential challenge. Because its power saving is difficult due to strong requirements on performance and availability, few solutions have been proposed so far.

In this paper, we proposed an application-aware power saving method using Dynamic Voltage and Frequency Scaling (DVFS) of processors for OLTP. Our method enables runtime power saving by using DVFS, and exploits runtime performance information of applications for saving power as much as possible under a given performance requirement. We conducted experiments to evaluate the effect of our power saving method with TPC-C and real-world workload. As a result, 10.0W (6.4% of total) was saved by using response time oriented DVFS policy, and 11.8W (7.5% of total) by throughput oriented DVFS policy.

1. はじめに

低炭素化が叫ばれる現在、情報技術分野においても広く省電力化が求められている。一方で、Amazon EC2, Microsoft Azure などのクラウド・コンピューティング・サービス、E-bay などの電子商取引、Facebook などのソーシャルコンピューティング等の大規模システムはデータセンタ上に構築されており、これらのデータセンタの消費電力は年々増加している。米国環境保護庁が 2007 年に発表した調査¹⁾によると、米国内におけるデータセンタの消費電力は 2000 年から 2006 年にかけておよそ 2 倍になり、2011 年にはさらにその 2 倍にまで増加することが予測されている。データセンタの消費電力増加は電力コストの増加を招くだけでなく、電力供給の制約により計算・記憶資源の新規導入や増強が阻害されるという問題を引き起す。そのため、データセンタの省電力化の必要性が高まっている。

EMERSON の報告 2) によると、データセンタで消費される電力のうち 44% がサーバによって消費されており、データセンタの省電力化のためにはサーバの省電力化が効果的である。サーバの消費電力を削減することにより、配電機器や無停電電源装置 (UPS) などの給電設備の消費電力もまた削減することができる。これによってデータセンタ内の総発熱量が低下するため、冷却設備の消費電力も削減することができる。つまり、サーバの省電力化は

^{†1} 東京大学 生産技術研究所

Institute of Industrial Science, the University of Tokyo

データセンタ全体の省電力化を行う足がかりとなる。

従来から、低消費電力プロセッサ、低消費電力メモリなどを組み合わせたサーバ機器の省電力化は多く試みられている。加えて、コンテナ型データセンタ、ラッカブル・クラスタなど、サーバ機器を効率よく集積し、空調効率も含めた省電力化を図るクラスタシステムが製品化されている。しかしながら、機器設備による省電力化はデータセンタ構築など新たな機器導入時には有効であるが、運用開始後のデータセンタにおいて想定以上の省電力化は望めない。また、個々のサーバ、クラスタの消費電力は低減するとしても、1)にあるようにデータセンタの成長規模がサーバ機器の省電力化を上回っていると考えられる。つまり、データセンタの省電力化においては、低消費電力な機器設備の導入に加え、運用開始後の省電力化を考慮しなくてはならない。

データセンタにおいて、大量の情報を処理し、管理するための中心的役割を果たしているのはデータベースサーバである。データベースサーバにおける主要なアプリケーションのひとつとして、オンライントランザクション処理 (OLTP) がある。OLTP アプリケーションはしばしば高スループット・低遅延を達成することが要求される。そのため、OLTP システムには潤沢な計算・記憶資源が投入され、大量の電力を消費する。それに加えて OLTP システムの停止はビジネスの機会損失に直結するため、これらのサーバは常に稼働し電力を消費する。このように消費電力の大きい OLTP システムの省電力化は、データセンタの省電力化において避けて通ることができない課題である。しかし、性能・可用性への厳しい要求のために省電力化が難しく、OLTP の省電力化手法はほとんど提案されていない。

データセンタ運用時の省電力化、特に社会活動、経済活動の基盤となるデータベース・サーバを対象とした省電力化では、アプリケーションを停めることのない実行時の省電力化手法が必要となる。アプリケーション実行時の省電力化手法として、プロセッサの動作周波数・電圧を動的に調整することにより省電力化を行う Dynamic Voltage and Frequency Scaling (DVFS) 機能が現在広く用いられている。プロセッサの消費電力は動作周波数と動作電圧の2乗に比例するため、これらを低くすることで消費電力を削減することができる。一方、プロセッサの命令スループット (単位時間あたりの命令実行数) は動作周波数に比例するため、動作周波数の低下はアプリケーション性能の低下を招く可能性がある。DVFS の動作周波数・電圧の制御方法に関して、消費電力と命令スループットのバランスをいかにとるかについては様々な研究が行われてきた。しかし、そのほとんどはプロセッサやシステムのレイヤーで閉じた制御であり、アプリケーションの性能に重点を置いた制御手法は少ない。

本論文では OLTP の省電力化を実現するために、DVFS を用いたアプリケーション指向

省電力化手法を提案する。まず、OLTP 処理モデルにおいて DVFS を利用した際の省電力化の可能性について検討し、OLTP 実行時において省電力可能であることを示す。提案する手法では、アプリケーション負荷に応じて DVFS により動作周波数制御し、アプリケーションを停止することなく省電力化を行う。その際、継続的に OLTP の実行時性能をモニタリングし、OLTP に求められる性能要求を満たしつつ省電力化を行わなければならない。DVFS を用いた OLTP 処理モデルに基づき、応答時間とスループットという二つの性能指標について、アプリケーションの性能目標を満たしながら省電力化を行うための DVFS ポリシーを設計した。さらに、提案する手法の有効性を確認するため、この二つの DVFS ポリシーを TPC-C ベンチマークを用いて評価し、実行時の省電力効果を検証した。なお、本論文においては、データベースが全て主記憶に格納されており、プロセッサおよびメモリアクセス性能がアプリケーション性能に対して支配的な影響をもつ環境を仮定して議論を行った。

本論文の構成は次のようになっている。2節では OLTP とアプリケーション指向省電力化の概要について述べると共に、DVFS を用いた OLTP 処理モデルを示し、アプリケーションの実行時性能要件を満たしつつ省電力化を行う DVFS ポリシーを提案する。3節では実験環境と当該システムの基本性能測定の結果を説明する。4節では人工的に合成した負荷と実サーバのトレースをもとにした負荷を用いて行った評価実験の結果を示す。6節では関連研究を紹介し、7節で結論を述べる。

2. オンライントランザクション処理とアプリケーション指向省電力化

2.1 オンライントランザクション処理と省電力化

オンライントランザクション処理 (OLTP) とは、同時に多数のトランザクションを処理するデータベースアプリケーションである。銀行取引、電子商取引、電子証券取引など社会に欠かすことのできないミッションクリティカルなサービスにおいて基盤技術として利用されており、一般に高い可用性が求められている。同時に、高いトランザクション処理性能が求められ、システムが備えるスループットや応答時間の最大性能がサービスレベル契約 (SLA) によって規定されている場合が多い。

OLTP を省電力化する際に、上述の高い可用性とアプリケーション性能に対する要求を満たしつつ、システムの消費電力の削減を行う必要がある。データベースシステムにおける省電力化のアプローチとしては、これまでシステムレベルあるいはミドルウェアレベルでのアプローチが検討されてきた。このようなアプローチは、性能の一時的な低下がある程度許容

3 オンライントランザクション処理における Dynamic Voltage and Frequency Scaling を用いたアプリケーション指向省電力化手法

される環境においては消費電力の削減に有効であるものの、アプリケーション性能が高度に規定されている OLTP においては、アプリケーション性能の達成と消費電力の削減を両立することは難しい。アプリケーション性能を直接捉え、性能規定に従ってシステムの制御を行うことにより消費電力の削減を目指すアプリケーション指向の省電力化が不可欠である。

2.2 Dynamic Voltage and Frequency Scaling によるアプリケーション指向省電力化

現在流通しているプロセッサの多くは実行時に動作周波数・電圧を変更することにより消費電力を削減する Dynamic Voltage and Frequency Scaling (DVFS) 機能が搭載されている。例えば、Intel 社の Xeon プロセッサにおいては、Intel SpeedStep Technology 機能が搭載されており、特殊なレジスタに値を書き込むことによりソフトウェアから動作周波数を指定することができる。

DVFS によりシステムの動作周波数を低下させると、プロセッサの命令実行スループットは低下するものの、消費電力は低減する。本論文では、当該特性を活用することにより、OLTP の実行時省電力化を目指す。OLTP において、システムが有するトランザクション処理性能と比較してトランザクション到着率が低い場合や、ディスク IO が多発しシステムの処理性能がディスク性能に律速している場合においては、DVFS を用いてプロセッサの命令実行スループットを低下させたとしても、これが OLTP のアプリケーション性能に与える影響は小さく、アプリケーションの性能低下を抑えつつシステムの省電力化を行うことができる可能性がある。アプリケーション性能を直接測定することにより、与えられたアプリケーション性能目標を維持しつつシステムの消費電力の削減が可能な状況を精緻に捉える実行時の DVFS 制御手法が必要である。

次節以降では、OLTP の性能指標として応答時間とスループットの 2 つに着目し、各アプリケーション性能の測定に基づき、性能目標を満たしつつシステムの消費電力を削減可能なプロセッサの周波数を実行時に決定する DVFS ポリシーを設計する。

2.3 動作周波数とオンライントランザクション処理性能の関係分析

DVFS ポリシーの設計を行うにあたり、動作周波数と OLTP 性能がどのような関係にあるのかを把握する必要がある。そこで OLTP の処理をモデル化し、その上で DVFS ポリシーの設計を行う。

図 1 にトランザクション処理のモデルを示す。OLTP アプリケーションでは、複数のユーザからデータベースサーバにトランザクションリクエストが送信され、データベース管理シ

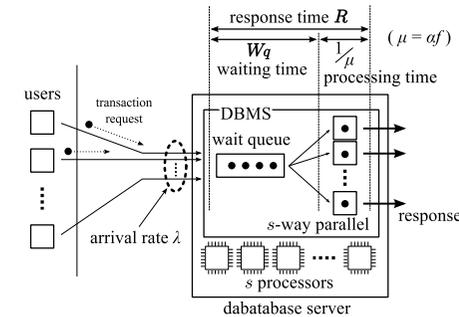


図 1 データベースサーバのトランザクション処理モデル

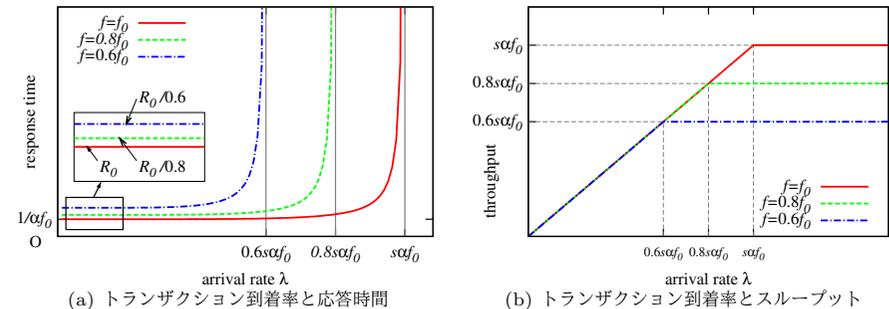


図 2 トランザクション到着率と OLTP 性能の関係

ステム (DBMS) によって処理される。データベースサーバのプロセッサ数を s^{*1} とすると、DBMS は同時に s 個のトランザクションを処理できるので、窓口数 s の待ち行列として考えることができる。ただし、トランザクション間におけるロック競合等の処理の依存関係は無いものと仮定する。ここで、1 プロセッサあたりの平均スループットを μ とすると、一般に f が高いほどスループット μ も高くなる。本論文では処理時間がプロセッサ性能に律速されるシステムを想定するため、 $\mu = \alpha f$ (α はプロセッササイクルあたりのトランザクション処理スループット) と仮定して動作周波数と OLTP 性能の関係について検討を行う。

トランザクション到着率を λ とすると、 P_0 (処理待ち及び処理中のトランザクションが系

*1 ここではマルチコアプロセッサにおける 1 コアを 1 プロセッサと数えている

内に無い確率)、 W_q (トランザクションの平均待ち時間)、 R (平均応答時間) はそれぞれ次のように表される。

$$P_0 = \frac{1}{\sum_{k=0}^s \frac{\rho^k}{k!} + \frac{\rho^{s+1}}{s!(s-\rho)}} \quad (1)$$

$$W_q = \frac{s\rho^{s+1}}{s!(s-\rho)^2} P_0 \quad (2)$$

$$R = W_q + \frac{1}{\mu} = W_q + \frac{1}{\alpha f} \quad (3)$$

ただし $\rho = \lambda/\mu = \lambda/\alpha f$ である。

プロセッサの最大動作周波数を f_0 とし、動作周波数が $f = f_0, 0.8f_0, 0.6f_0$ のそれぞれの場合について、トランザクション到着率と応答時間の関係を図 2(a) に示す。 $f = f_0$ の場合を例にとると、トランザクション到着率 λ が 0 に近くシステムが低負荷な状態にあるときは $W_q \approx 0$ であり、応答時間 R は $R \approx 1/\alpha f_0$ である。 λ が増加してシステムにかかる負荷が高くなるにつれて R は増加し、 λ が $s\alpha f_0$ に近づくにつれて急増する。この傾向は動作周波数が異なる値をとるときにも同様であり、動作周波数が $0.8f_0, 0.6f_0$ と下がるにつれて、到着率が低くても応答時間 R が急増することが図 2(a) からわかる。この応答時間 R の急激な変化を捉えることで、動作周波数切り上げの契機とすることができる。

スループットと到着率の関係を図 2(b) に示す。動作周波数 $f = f_0$ の場合を例にとると、到着率が $s\alpha f_0$ より低い場合には到着率とスループットは比例しているが、 $s\alpha f_0$ より到着率が高い場合にはスループットは上がらない。この傾向は他の動作周波数の場合にも同様であり、動作周波数 $0.8f_0$ のときにはスループットの上限は $0.8s\alpha f_0$ 、 $0.6f_0$ の場合には $0.6s\alpha f_0$ である。仮に性能目標としてトランザクション処理スループット T_{th} 以上を達成することが求められたとすると、 $s\alpha f > T_{th}$ の範囲内における最小の f を選択することで性能目標を達成しつつ省電力化を行うことができる。

2.4 アプリケーション指向省電力化手法における DVFS 制御方式

我々の提案するアプリケーション指向省電力化手法における DVFS 制御のアーキテクチャを図 3 に示す。この手法では、データベースサーバにおいて動作する DVFS コントローラがアプリケーションから一定時間間隔で実行時性能情報を取得し、DVFS ポリシーにしたがいプロセッサの動作周波数制御を行う。DVFS コントローラは OLTP 以外のアプリケーションへの対応を考慮し、データベースサーバにも非依存となるよう独立したモジュールとして設計している。

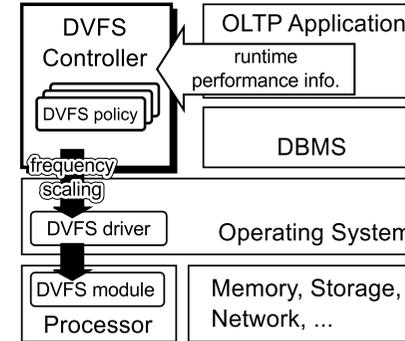


図 3 アプリケーション指向省電力化の DVFS 制御アーキテクチャ

OLTP 実行時の DVFS のコントローラにおける動作周波数の決定方法として、本研究では OLTP における SLA を定める際に用いられることの多い応答時間とスループットという 2 つの性能指標に着目する。まず、それぞれについて次のような性能目標を設定した。

- 応答時間: 平均応答時間が R_{th} 以下であること
- スループット: システムのトランザクション処理性能がスループット性能目標 T_{th} 以上である

これら性能目標のそれぞれについて、目標を達成しつつ省電力化を行う DVFS ポリシーを設計した。以下、応答時間を指標として用いる応答時間指向 DVFS ポリシーと、スループットを指標とするスループット指向 DVFS ポリシーを順に説明する。

2.4.1 応答時間指向 DVFS ポリシー

応答時間指向 DVFS ポリシーでは、平均応答時間が R_{th} 以下であることが性能目標である。この目標を達成しつつ、できるだけ動作周波数を低くして消費電力を削減するようポリシー設計を行った。

このポリシーでは、一定の時間間隔で取得される平均応答時間の値をもとに、図 4 の疑似コードに示す *RT_based_policy* 関数を用いて動作周波数の増減を決定する。この関数の中では、応答時間の測定値が $R_{th} \times R_{up_ratio}$ よりも大きければ動作周波数を上げ、 $R_{th} \times R_{down_ratio}$ よりも小さければ動作周波数を下げるという処理を行う。ただし

$$0 < R_{down_ratio} < R_{up_ratio} < 1$$

である。 R_{up_ratio} を導入することにより応答時間が実際に R_{th} を上回る前に動作周波数を上げることができると期待される。 R_{up_ratio} は、その値が 1 に近くなるほど動作周波数

5 オンライントランザクション処理における Dynamic Voltage and Frequency Scaling を用いたアプリケーション指向省電力化手法

```

1  $R_{th}$            // response time constraint
2  $R_{up\_ratio}$     // threshold for raising frequency.  $R_{up\_ratio} < 1$ 
3  $R_{down\_ratio}$   // threshold for dropping frequency.  $R_{down\_ratio} < R_{up\_ratio} < 1$ 
4
5 function RT_based_policy(response_time){
6   if (response_time >  $R_{th} * R_{up\_ratio}$ )
7     raise_frequency();
8   else if (response_time <  $R_{th} * R_{down\_ratio}$ )
9     drop_frequency();
10 }

```

図 4 応答時間指向 DVFS ポリシーの疑似コード

の上昇が抑制される。実際のシステムにおいては個々のトランザクションに対する応答時間にはばらつきがあるため、高負荷時には R_{up_ratio} が 1 に近いほど応答時間が R_{th} を超えるトランザクション数が増加すると予想される。消費電力に関しては、 R_{up_ratio} の 1 に近いほど動作周波数の上昇が抑制されるため、消費電力は低くなる傾向にあると考えられる。

また、 R_{up_ratio} と R_{down_ratio} の値に差を設けることで、動作周波数の振動を抑制することができる。 R_{down_ratio} の値が R_{up_ratio} の値と近くなるほど、平均応答時間の値が R_{down_ratio} 付近の場合に動作周波数の上昇・下降が繰り返される可能性が高くなる。性能目標が達成できない恐れがあるために動作周波数を引き上げるべきと判断される状況において、このような動作周波数の上昇・下降が繰り返されることは応答時間の性能目標違反が発生する確率を高くする。消費電力に関しては、 R_{down_ratio} の値が 0 に近くなるほど、動作周波数の下降が抑制されるため消費電力は高くなる傾向にあると考えられる。

2.4.2 スループット指向 DVFS ポリシー

スループット指向 DVFS ポリシーでは、システムのトランザクション処理性能がスループット性能目標 T_{th} 以上であることが性能目標である。そのもとの、できるだけ動作周波数を低くして消費電力が削減できるようポリシー設計を行った。

このポリシーでは、一定の時間間隔でモニタリングされるスループット、トランザクション到着率、CPU 使用率の平均値をもとに、図 5 に示す *TP_based_policy* 関数によって動作周波数の増減を決定する。

```

1  $T_{th}$            // throughput constraint
2 overspeed_ratio // threshold for detecting overspeed
3 cpu_busy0       // < 1
4 cpu_busy1       // < cpu_busy2 < 1
5 cpu_busy2       // < 1
6
7 function TP_based_policy(throughput, arrival_rate, cpu_usage){
8   if (throughput >  $T_{th} * overspeed\_ratio$ )
9     drop_frequency();           // (A)
10  else if (throughput >  $T_{th}$ )
11    if (cpu_usage < cpu_busy0)
12      drop_frequency();         // (B)
13  else
14    if (arrival_rate >  $T_{th}$ )
15      raise_frequency();         // (C)
16  else
17    if (cpu_usage < cpu_busy1)
18      drop_frequency();         // (D)
19    else if (cpu_usage > cpu_busy2)
20      raise_frequency();         // (E)
21 }

```

図 5 スループット指向 DVFS ポリシーの疑似コード

スループットの平均値 T が目標値 T_{th} よりも高い場合には、必要以上の性能が達成されているので、動作周波数を下げることで省電力化を行うことができる (図 5(A))。ただし、 T と T_{th} の値が近い場合、動作周波数を下げることで $T < T_{th}$ となってしまう可能性があるため、この領域を $overspeed_ratio (> 1)$ というパラメータを導入することで、 $T_{th} < T < T_{th} \times overspeed_ratio$ と定める。 $overspeed_ratio$ の値が大きいほど動作周波数の下降を抑制が抑制されるため、性能目標違反の確率を低く抑えることができるが消費電力は高くなる。

$T_{th} < T < T_{th} \times overspeed_ratio$ の場合には、スループット性能に加えて CPU 利用率を利用して動作周波数の制御を行う。実際のシステムではスループット性能の測定値にばらつきがあるため、性能目標に近いスループットが出ている場合には、システムのスループット性能が目標達成のために十分であるかをスループットの測定値のみから判断することが難しい。このような場合には、CPU 利用率が比較的 low プロセッサの処理性能に余裕があると判断できれば動作周波数を下げてもスループットへの影響は少ないと判断できる (図 5(B))。この判断に用いる CPU 利用率の閾値が $cpu_busy_0 (< 1)$ である。 $cpu_busy_0 (< 1)$ の値が小さいほど動作周波数の下降が抑制されるため、性能目標違反の確率は低下するが、消費電力は高くなる。

スループットの平均値 T が目標値 T_{th} よりも低く、トランザクション到着率 λ が T_{th} よりも高ければ性能目標を達成できていないということなので動作周波数を上げる (図 5(C))。

スループットの平均値 T が目標値 T_{th} よりも低く、トランザクション到着率 λ が T_{th} 以下の場合には、CPU 利用率をもとにして動作周波数の増減を行う。CPU 利用率が比較的低い場合には、プロセッサの処理性能に余裕があり、動作周波数を下げてもスループットが低下することはないと考えられる (図 5(D))。このときの CPU 利用率の閾値を $cpu_busy_1 (< 1)$ とする。逆に CPU 利用率が高い場合には、動作周波数を上げることでスループットが目標値 T_{th} を下回ることを防止する (図 5(E))。この CPU 利用率の閾値を $cpu_busy_2 (< 1)$ とする。ただし $cpu_busy_1 < cpu_busy_2$ である。 cpu_busy_1 を低くすると動作周波数の下降が抑制されるため、性能目標違反を抑える方向に作用するが、その分消費電力は高くなると考えられる。一方 cpu_busy_2 を低くすると動作周波数の上昇が抑制されるため消費電力が抑えられるが、性能目標違反の確率は高くなる。

3. 実験環境と TPC-C を用いた基本性能測定

3.1 実験環境

データベースサーバとして Dell PowerEdge R510 を 1 台用いた。このサーバには、Intel Xeon X5550 2.67GHz を 2 ソケット、メモリは 48GB(DDR3 8GB RDIMM \times 6) 搭載されている。Intel Xeon X5550 では、DVFS 機能である Intel SpeedStep によって実行時の動作周波数を 1.60GHz から 2.66GHz まで 9 段階で設定することができる。データロガー (Hioki 2332-20 Power Meter Module) とクランプ電流計を用いてストレージも含めたデータベースサーバ全体の消費電力を 1 秒間隔で測定した。OLTP アプリケーションのフロントエンドサーバとしては、Dell PowerEdge R900 を 1 台用いた。このサーバには、Intel Xeon X7460 2.66GHz が 4 ソケット、メモリは 128GB(DDR2 4GB FB-FIMM \times 32) 搭載されている。フロントエンドサーバとデータベースサーバは 1Gb Ethernet で接続されている。

実験に用いたソフトウェア構成を以下に述べる。データベース管理システムには MySQL 5.1.41、ストレージエンジンとして InnoDB を利用し、Linux カーネル 2.6.18 上で実行した。DVFS の効果を確認するために、InnoDB のデータファイルは全て主記憶上のファイルシステムである `tmpfs` に配置し、データファイルが全て収まるよう 20GB のパッファプールを割り当てた。DVFS の制御には、Linux カーネルのモジュールである `cpufreq` を用いた。

オンライントランザクション処理のアプリケーションとして、業界標準ベンチマークである TPC-C³⁾ を用いた。TPC-C はフロントエンドサーバ上で実行され、データベースサーバへトランザクションリクエストを送信する。TPC-C のデータサイズの尺度である warehouse 数は 100 (データサイズはおおよそ 10GB) とした。また TPC-C のターミナル数は warehouse 数と同数の 100 とし、各 warehouse に一つずつターミナルを割り当てた。TPC-C のその他のパラメータは version 5.11 に準拠するよう設定を行った。

3.2 TPC-C を用いた基本性能測定

アプリケーション指向省電力化手法の評価実験に先立ち、本研究で用いた実験環境における TPC-C の基本的な性能測定を行った。この実験では、データベースサーバの動作周波数は最大の 2.66GHz、最小の 1.60GHz、その中間 2.13GHz の 3 通り、目標トランザクション到着率^{*1}は 0~50000tpmC まで 5000tpmC 刻みの 11 通りとし、それぞれの組み合わせ

*1 実験におけるトランザクション到着率は TPC-C の keying time および think time の値により調整を行った。このときの制御目標値をここで目標トランザクション到着率と呼ぶ。

7 オンライントランザクション処理における Dynamic Voltage and Frequency Scaling を用いたアプリケーション指向省電力化手法

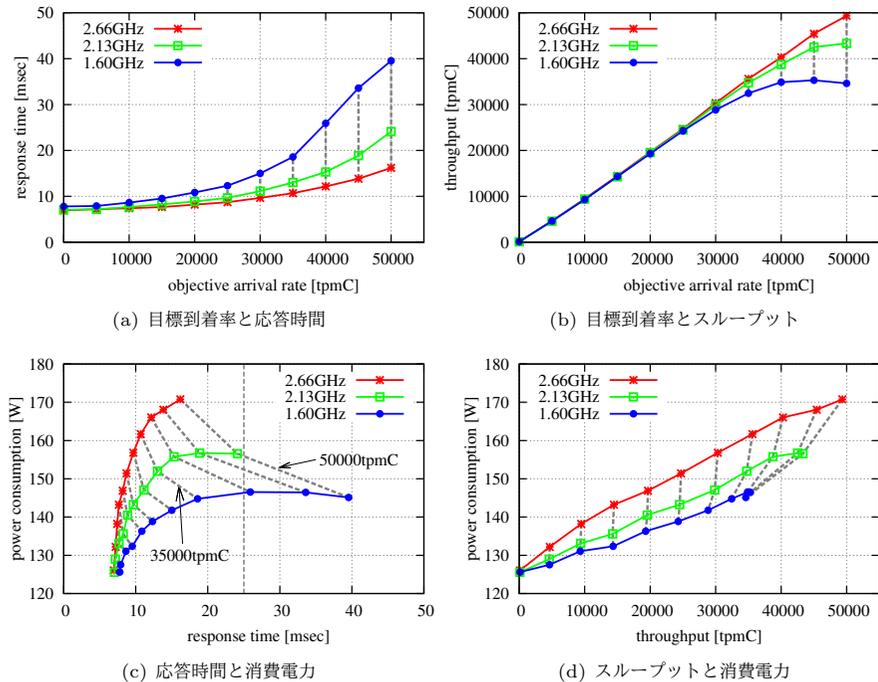


図 6 基本性能測定の実験結果

について、スループット、応答時間、消費電力を測定した。各設定で TPC-C を 20 分間実行し測定を行った。測定結果の集計においては、測定開始直後・終了直前の各 1 分間の過渡的なデータを除いた平均値を測定値として採用した。

実験結果を図 6 に示す。図 6(a) は目標トランザクション到着率と応答時間の関係を表しており、2.3 節で示したモデルから導かれるグラフ (図 2(a)) と同じ傾向が確認された。

図 6(b) は目標到着率とスループットの関係を表している。このグラフから、目標到着率が低い場合には動作周波数に関わらず到着率と同じスループットが達成できているが、それぞれの動作周波数にはスループット性能上限がありそこで頭打ちとなっていることが確認できる。これについても、2.3 節のモデルと同じ傾向である。

図 6(c) は応答時間と消費電力の関係を表している。同じ目標トランザクション到着率の

もとで行われた測定値は点線で結ばれている。このグラフからは、動作周波数を下げると応答時間は増加するが、消費電力が削減されることが読み取れる。つまり、応答時間が性能目標を満たす範囲で動作周波数を下げることで消費電力を削減できることを示している。たとえば、応答時間の性能目標が 25msec であったとすると、目標到着率 50000tpmC の場合には 2.13GHz で動作させることで性能目標を達成することが可能であり、2.66GHz で動作させるよりも消費電力を削減できる。また 35000tpmC の場合には最低の動作周波数 1.60GHz で性能目標を達成することができる。

図 6(d) はスループットと消費電力の関係を示したグラフである。目標到着率が 25000tpmC 以下の領域では動作周波数に関わらずほぼ同じスループットを達成できており、この領域では性能をほとんど下げることなく消費電力を削減できることを示している。

4. TPC-C を用いたアプリケーション指向省電力化手法の評価

提案手法による省電力効果を確認するために、3 節と同じ環境で TPC-C を実行し、人工的な負荷変動および実サーバのトレースを利用した負荷変動のもとで評価実験を行った。またそれに加えて、OLTP においてアプリケーション性能を省電力制御に利用することの有効性を示すため、CPU 使用率のみを用いる OS 組み込みの DVFS 制御手法との比較実験を行った。これらの実験では 3 節の実験と異なり動作周波数の 9 段階全てを用いた。評価を行った DVFS ポリシーは、1. 動作周波数を最大で固定、2. 応答時間指向 DVFS ポリシー、3. スループット指向 DVFS ポリシーの 3 つである。動作周波数を最大に固定した場合の性能がその時々でのシステムの最大性能を表し、評価の基準となる。

実験における応答時間指向 DVFS ポリシーのパラメータは $R_{th} = 20[\text{msec}]$ 、 $R_{up_ratio} = 0.8$ 、 $R_{down_ratio} = 0.6$ とした。またスループット指向 DVFS ポリシーのパラメータは $T_{th} = 40000[\text{tpmC}]$ 、 $overspeed_ratio = 1.1$ 、 $cpu_busy_0 = 0.75$ 、 $cpu_busy_1 = 0.7$ 、 $cpu_busy_2 = 0.8$ とした。ただしこれらパラメータの設定値は、当該実験環境において性能目標を達成しながら省電力化を行うことが可能となるよう、予備実験の結果にもとづいて経験的に設定されたものであり、必ずしも最適な結果を与えるものではない。

4.1 人工的な負荷変動を用いた評価

この実験では、トランザクション到着率が 5000tpmC から 55000tpmC へと毎分 5000tpmC ずつ増加し、その後 55000tpmC から 5000tpmC へと毎分 5000tpmC ずつ減少する負荷パターンを用いて測定を行った。測定に際しては、到着率を変動させる前後 30 秒間は到着率を 5000tpmC として測定を行っており、測定値に変化がないため実験結果からは削除して

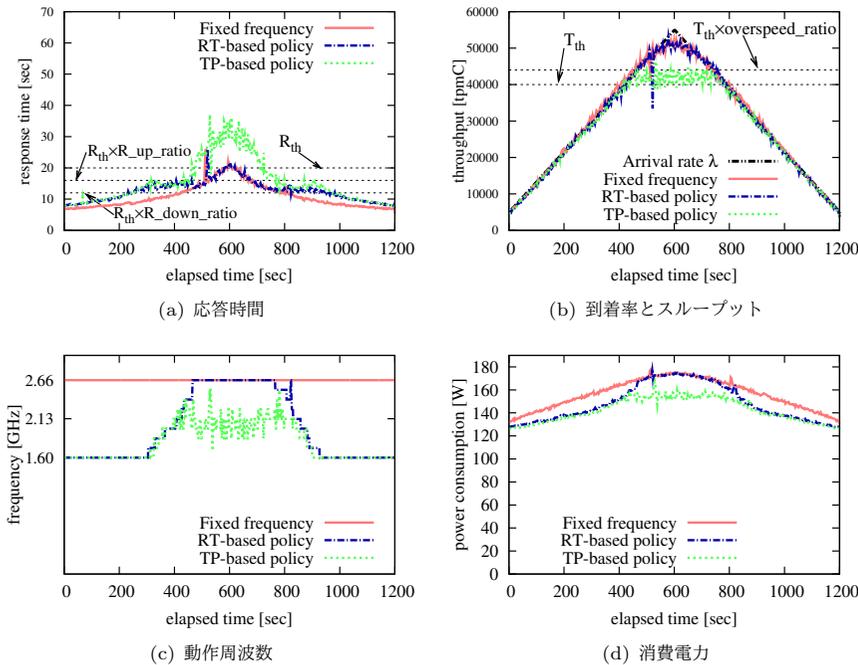


図 7 人工的な負荷変動を用いたアプリケーション指向省電力手法の評価実験結果

いる。

実験結果を図 7 に示す。凡例の“Fixed frequency”は動作周波数を最大値 2.66GHz に固定した場合、“RT-based policy”は応答時間指向ポリシー、“TP-based policy”はスループット指向ポリシーに従って DVFS 制御を行った場合を表している。それぞれのグラフは実行時間の経過にともなう測定値の変化を示しており、図 7(a) は応答時間、図 7(b) は到着率およびスループット、図 7(c) は動作周波数、図 7(d) は消費電力である。

まず応答時間指向 DVFS ポリシーの結果を説明する。図 7(a) より実行時間のほとんどで応答時間は目標値 R_{th} 以下に抑えられている。到着率が最大となる 600 sec 付近では応答時間が R_{th} を越えているが、これはトランザクション到着率がシステムの最大性能を超えているためである。スループットに関しては、図 7(b) より実行時間全体において最大動作周波数固定の場合とほぼ同じスループットであった。動作周波数は応答時間が $R_{th} \times R_{up_ratio}$

に到達した 300 sec 付近から上昇をはじめ、450 sec 付近で最大周波数に到達した。その後 770 sec 付近から動作周波数は下降しはじめ、900 sec 付近で最低動作周波数となり、以降動作周波数は一定であった。図 7(d) に示す消費電力は、450 ~ 770 sec では最大動作周波数固定の場合とほぼ同様の消費電力であったが、450 sec 以前および 770 sec 以降では最大動作周波数固定の場合に比べて消費電力を削減できたことがわかる。以上の結果より、応答時間指向 DVFS ポリシーに従って DVFS 制御を行うことで、応答時間を性能目標値 R_{th} 以下に抑えつつ、動作周波数が最大となる 450 ~ 770 sec を除いて消費電力を削減することができたことがわかった。

次にスループット指向 DVFS ポリシーの結果を説明する。図 7(b) より、400 sec 付近までは到着率の増加に伴いスループットは増加し、400 ~ 800 sec においてスループットは T_{th} と $T_{th} \times \text{overspeed_ratio}$ ($= 1.1T_{th}$) の間を推移した。800 sec 以降はふたたび到着率の低下にそってスループットは低下した。これよりスループットの性能目標を達成できたことがわかる。動作周波数は図 7(c) に示すように 300 sec 付近から上昇をはじめ、400 ~ 800 sec においては 2.13GHz 前後にとどまった。そして 800 sec 付近から下降しはじめて 900 sec 付近において最低動作周波数となった。消費電力は図 7(d) に示すように動作周波数およびスループットの値と連動しており、400 sec までは増加し、400 ~ 800 sec では 155W 前後を維持し、800 sec 付近からは減少した。到着率が最大となる 600 sec 付近においては、性能目標を達成しつつ最大動作周波数固定の場合にくらべて約 20W の消費電力削減効果が確認された。以上より、スループット指向 DVFS ポリシーに従って DVFS 制御を行うことで、システムのトランザクション処理性能を性能目標値 T_{th} 以上に維持しながら省電力化ができることがわかった。

以上の結果より、我々の提案するアプリケーション指向省電力手法の応答時間指向 DVFS ポリシーとスループット指向 DVFS ポリシーは、どちらも性能目標を達成しつつ省電力化を行うことができると確認された。また応答時間とスループットという 2 つの指標どちらを用いるかによって、同じ到着率の変動パターンのもとで OLTP 性能および消費電力に違いが生じることが確認された。

4.2 OS レベル DVFS 制御とアプリケーション性能

この実験では、OS 組込みの DVFS 制御を利用した場合に、アプリケーション性能および消費電力がどのように変化するかを、TPC-C を用いて評価を行った。

評価に用いたのは、Linux の DVFS カーネルモジュールである `cpufreq` の `ondemand` ポリシーである。`ondemand` ポリシーでは、`sampling_rate` で指定された時間間隔で CPU 使用

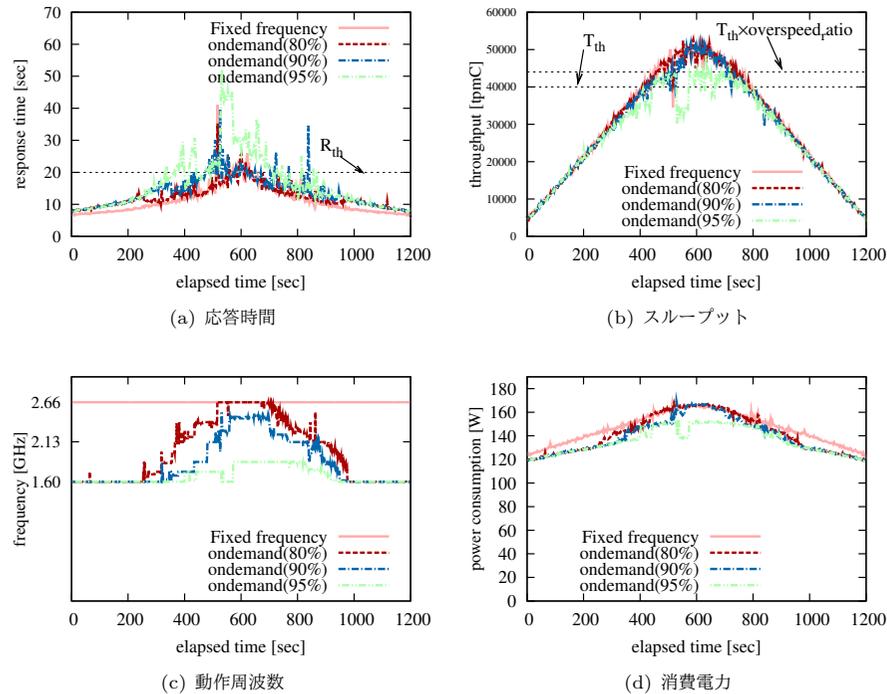


図 8 OS レベル DVFS 制御を用いた評価実験結果

率を値を測定し、その値が $up_threshold$ よりも高い場合には動作周波数を最大まで引き上げ、 $up_threshold - 10\%$ より低い場合には動作周波数を一段階下げる、という動作周波数の制御が行われる*1。同じ負荷のもとでは、 $up_threshold$ の値が高いほど動作周波数は低く抑えられるため、DVFS による性能ペナルティが大きくなり、消費電力は低く抑えられる。逆に $up_threshold$ の値が低いほど動作周波数が最大に引き上げられる機会が多くなるため、性能のペナルティは小さくなり、消費電力は高くなる。当該実験環境で用いられた Linux 2.6.18 ではこれらパラメータの初期設定値は $sampling_rate = 20[msec]$ 、 $up_threshold = 80\%$ で

*1 実際には動作周波数の変動範囲を設定できるなど更に細かな調整が可能であるが、基本的な動作は $sampling_rate$ と $up_threshold$ の 2 つによって規定される。

ある。 $up_threshold$ 本実験では、 $sampling_rate$ を提案手法と同様の 1 秒とし、 $up_threshold$ を 80%、90%、95% と変化させて評価を行った。

評価に際しては、前節の実験と同様に TPC-C トランザクション到着率が 5000tpmC から 55000tpmC へと毎分 5000tpmC ずつ増加し、その後 55000tpmC から 5000tpmC へと毎分 5000tpmC ずつ減少する負荷パターンを用いて測定を行った。測定に際しては、到着率を変動させる前後 30 秒間は到着率を 5000tpmC として測定を行っており、測定値に変化がないため実験結果からは削除している。

測定結果を図 8 に示す。凡例の“Fixed frequency”は周波数を最大値 2.66GHz に固定した場合、“ondemand(80%)”、“ondemand(90%)”、“ondemand(95%)”はそれぞれ ondemand ポリシーのパラメータ $up_threshold$ を 80%、90%、95% とした場合の測定結果を表す。各グラフは時間の経過にもなう測定値の変化を示しており、図 8(a) は応答時間、図 8(b) は到着率およびスループット、図 8(c) は動作周波数である。

まず図 8(a) に示す応答時間の変化について説明する。グラフからわかるように、 $up_threshold$ の値が大きいほど応答時間が長くなる傾向にある。 $up_threshold = 80\%$ の場合には、経過時間 300 秒から 900 秒までは動作周波数最大固定の場合とほぼ同様の応答時間となった。一方で、 $up_threshold = 90\%$ 、95% の場合には平均応答時間が大きく変動し、トランザクション到着率が最大となる場合以外においても平均応答時間が $R_{th} = 20[msec]$ を超えることがたびたび確認された。このことから、CPU 使用率をもとにした DVFS 制御では、応答時間を一定範囲内に抑えながら省電力化を行うことは難しいといえる。

次に図 8(b) に示すスループットの変動について説明する。このグラフからわかるように、 $up_threshold$ が 80% と 90% の場合については、動作周波数最大固定の場合とほぼ同じスループットが全実行時間において達成されていることがわかる。一方で、 $up_threshold = 95\%$ の場合には経過時間 400 秒から 800 秒の間には動作周波数最大固定の場合に比べてスループット性能が低くなった。つまり、スループット性能の最大値を抑えることによる省電力化を行うためには、 $up_threshold$ を 95% 程度以上に設定する必要があるといえる。しかしながら、 $up_threshold$ を高い値に設定した場合、他のプロセスが実行される等のわずかな負荷変動が動作周波数に与える擾乱が大きいと考えられる。実際に、経過時間 500 秒から 570 秒の間には擾乱によるものと思われるスループットの落ち込みが観測された。このことから、CPU 使用率を用いた DVFS 制御においては、図 7(b) に示すスループット指向 DVFS ポリシーのように高い到着率のもとでスループット性能を一定範囲内に制御することが難しいといえる。

図 8(c) に示す動作周波数の変動について説明する。当該環境において、`cpufreq` で `ondemand` ポリシーを用いた場合には動作周波数は各プロセッサコアごとに行われるため、このグラフでは動作周波数の平均値がプロットされている。このグラフより、いずれの場合についてもトランザクション到着率の上昇・下降にあわせて動作周波数が上昇・下降することが確認された。また、`up_threshold` が高いほど同じ到着率における動作周波数は低くなることも確認された。

図 8(d) は消費電力の変動を示したグラフである。いずれの場合についてもトランザクション到着率が高いほど消費電力も高い傾向がみられた。また、図 8(c) に示す動作周波数の変動と比較するとわかるように、動作周波数の差が消費電力の差に反映されていることが確認された。

これらの実験結果より、応答時間、スループットのいずれの性能指標についても、CPU 使用率を制御指標に用いる `cpufreq` の `ondemand` ポリシーではアプリケーション性能を達成しながら動作周波数を低く抑えるよう制御を行うことが難しいということが確認された。

4.3 実サーバのトレースを用いた評価

この実験では、実際のサーバの負荷パターンに対する提案手法の省電力効果を評価した。負荷のパターンとしては、WorldCup98 データセット⁴⁾を参考にしてトランザクション到着率を変動させ、測定を行った。このデータセットは 1998 年のワールドカップ期間中における Web サーバのアクセストレーズデータである。測定に際しては、到着率を変動させる前後 30 秒間は到着率を 20000tpmC として測定を行っており、測定値に変化がないため実験結果からは削除している。

実験結果を図 9 に示した。図中の凡例は“Fixed frequency”は動作周波数固定の場合、“RT-based policy”は応答時間指向 DVFS ポリシー、“TP-based policy”はスループット指向 DVFS ポリシーにしたがって DVFS 制御を行った場合を表す。それぞれのグラフは実行時間の経過にともなう測定値の変化を示しており、図 9(a) は応答時間、図 9(b) は到着率およびスループット、図 9(c) は動作周波数、図 9(d) は消費電力である。

応答時間指向 DVFS ポリシーにより DVFS 制御を行った場合の結果について説明する。図 9(a) より、応答時間は実行時間全体にわたって性能目標の閾値である R_{th} より低い値に抑えられたことがわかる。スループットは図 9(b) に示すように動作周波数最大固定の場合とほぼ同じスループットであった。動作周波数は図 9(c) に示すように 2300 sec 付近から到着率が増加するにつれて上昇し、3500 ~ 5000 sec において最大の周波数で動作した。このときには、図 9(d) に示すように消費電力は動作周波数最大固定の場合とほぼ同じ値をとり、

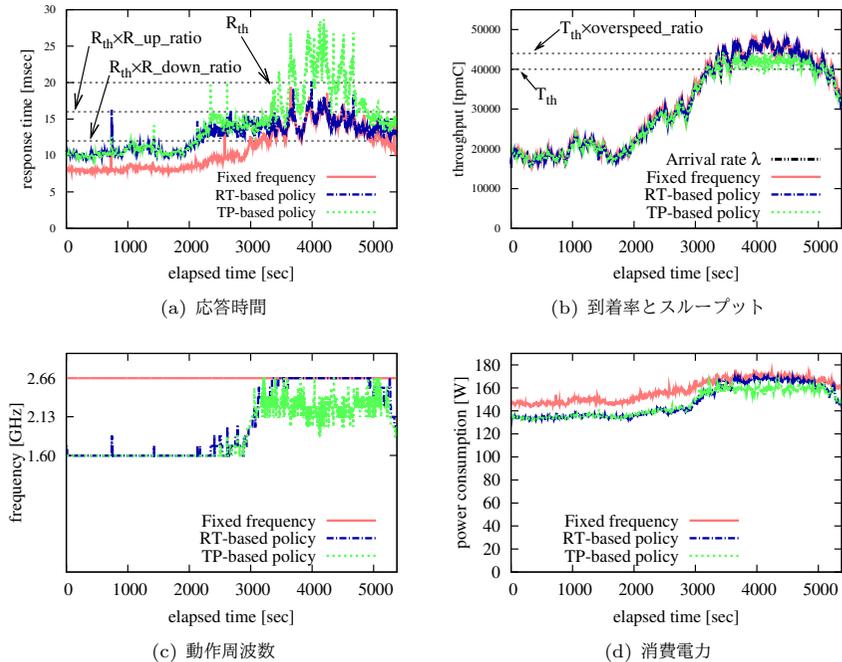


図 9 実サーバのトレースを用いたアプリケーション指向省電力手法の評価実験結果

それ以外の測定開始から 3500 sec までと 5000 sec 以降においては動作周波数最大固定の場合と比べて消費電力を削減できている。これより応答時間指向 DVFS ポリシーを用いることで、応答時間を性能目標値 R_{th} 以下に抑えつつ、省電力化を実現できることが確認された。動作周波数最大の場合と比較して、実行時間全体を通した消費電力の削減量は平均 10.0W(全体の 6.4%)であった。

次に、スループット指向 DVFS ポリシーにより DVFS 制御を行った場合の結果を説明する。図 9(b) より、測定開始から 3000 sec 付近までは到着率が T_{th} より低く、スループットは到着率に追従した。3000 sec 以降は到着率が T_{th} より高くなり、3000 ~ 5000 sec においてスループットは T_{th} と $T_{th} \times \text{overspeed_ratio}$ ($= 1.1T_{th}$) の間を推移した。5000 sec 以降は到着率が T_{th} を下回り、ふたたびスループットは到着率に連動して変化した。動作周波数の変動は図 9(c) に示すように、2500 sec 付近から徐々に上昇しはじめ、3000 ~ 5000 sec

の間は到着率の変動に追従してほぼ 2.13GHz と 2.66GHz の間を変動した。図 9(d) に示す消費電力のグラフからは、実行時間全体において最大動作周波数固定の場合と比較して省電力化できたことが確認された。以上より、スループット指向 DVFS ポリシーを用いることで、トランザクション処理性能をスループットの性能目標値 T_{th} 以上に維持しながら省電力化を行うことができると確認された。この実験では、動作周波数最大固定の場合とくらべて消費電力の削減量は平均 11.8W(全体の 7.5%) であった。

以上の結果より、我々が提案するアプリケーション指向省電力手法では、応答時間指向 DVFS ポリシーとスループット指向 DVFS ポリシーのどちらかをを用いて DVFS 制御を行った場合も、実サーバのトレースをもとにした負荷変動に対して性能目標を達成しつつ消費電力を削減できることが確認された。

5. 関連研究

アーキテクチャレベルでの DVFS による省電力技術は、様々な制約条件、目的の下で提案が行われてきた。特にここ数年はマルチコアにおける DVFS の研究が主流である。Herbert らは 5) において動作周波数・電圧の制御単位である VFI(voltage/frequency island) の粒度選択のトレードオフについて検証した。また Herbert は 6) において、製造プロセスのばらつきを考慮に入れて DVFS の制御を行うことで電力効率を大幅に改善できることを示した。また DVFS 関連の研究では SPEC2000 などのベンチマークが用いられることがほとんどであったが、5) では Apache や TPC-C、TPC-H などを評価に用いて実際のサーバのワークロードでの効果を測定した。ただし 5) にて提案される DVFS 制御には CPU 使用率のみが用いられており、本研究で提案するアプリケーション性能を用いた DVFS 制御とは異なる。また評価において、TPC-C、TPC-H はそれぞれ複数あるワークロードの一つであり、省電力効果の測定のみが行われている。本論文が前提とする、OLTP における性能要求を満たすことができているか、また実行時の動的な負荷変動にどう対応するかという観点からの評価は行われていない。

アプリケーション性能を考慮した省電力化の取り組みとしては次に挙げるようなものが行われてきた。Chen らは 7) において、multitier 構成の Web アプリケーションシステムの応答時間と動作周波数の関係を予測する手法を提案した。しかし、この手法を利用した動作周波数制御に関する検討は行われていない。Lee らは 8)、9) において動画再生、動画エンコーディング、画像圧縮などのアプリケーションを対象として、DVFS 機能を有するプロセッサを搭載した環境における省電力なタスクスケジューリング手法を提案した。こ

れら動画・画像処理に特化したスケジューリング手法では、制御には CPU 使用率が指標として用いられており、実行時アプリケーション性能情報を用いる本研究とは異なっている。Elnozahy らは Web サーバをアプリケーションとした場合に、応答時間を指標として DVFS 制御を行い、それに加えてリクエストのバッチ処理を行うことによる省電力化手法を提案した¹⁰⁾。10) の提案する DVFS 制御ポリシーでは応答時間を指標として用いており、シミュレーションにより評価がなされている。それに対し、本研究では実際に稼動するシステムにおける測定で評価を行っており、スループット性能を用いた DVFS 制御ポリシーを提案している点について異なる。Rusu らは 11) において、Web アプリケーションの応答時間を用いた省電力化手法を提案した。この手法では非均一なマシン構成のクラスタを対象としており、応答時間性能の性能目標を達成できる範囲においてクラスタを再構成し、また各マシンにおいて DVFS 制御を行うことにより省電力化を実現している。この手法では応答時間と動作周波数の関係を線形なモデルで近似しているのに対して、本研究では実際の応答時間の測定値に基づいて動作周波数を決定している点、また OLTP のようにミッションクリティカルなアプリケーションで重要であるスループット性能についても検討を行っている点について異なる。

データベースシステムの省電力化技術に関する研究では、データベース分野研究の方針を決める上で重要な役割を果たしている The Claremont Report on Database Research の 2005 年版¹²⁾ にて、データベースエンジンの見直すべき項目として省電力化技術が指摘された。その他にもデータベースシステムの省電力化技術を指摘する文献は多い。Graefe は、データベースシステムのクエリ最適化の新たな指標としてエネルギー効率を導入すべきであると指摘した¹³⁾。また同著においては、クエリオプティマイザやスケジューラなどの各モジュールが解決すべき課題が提示されている。Harizopoulos らはデータベースシステムの消費電力削減はハードウェアの省電力化技術だけでは不十分であり、データベースエンジンの改良が不可欠であることを指摘した¹⁴⁾。しかし具体的な省電力化手法の提案は少ない。

データベースシステムの中でも、オンライン分析処理 (OLAP) に関しては省電力化技術への取り組みが比較的多く見られる。Meza らは OLAP に適したシステム構成のシステムにおいて、各コンポーネントの詳細な消費電力の測定を行い、最適なハードウェア構成を調べた¹⁵⁾。Poess らは業界標準ベンチマーク TPC-H のトップシステムの消費電力を見積り、電力効率の時系列変化を分析した¹⁶⁾。また彼らは様々なハードウェア構成における DBMS の消費電力と性能の関係を分析した¹⁷⁾。Tsirogiannis らはハッシュ結合、ソートマージ結合などのデータベース処理の消費電力の分析を基に、エネルギー効率を最も高めるようなシ

システム設計の指針を提案した¹⁸⁾。Lang らはクエリスケジューラを改善することでプロセスの省電力モードを活用し、消費電力を削減する手法を提案した¹⁹⁾。Xu らは消費電力を電力モデル化することによってクエリオプティマイザを構築し、データベースシステムの消費電力を削減する手法を提案した²⁰⁾。

オンライントランザクション処理に関する省電力化の取り組みとして、Poess らは TPC-C の歴代トップシステムの消費電力を見積り、消費電力あたりの性能の増加率が十分ではなく、現在の性能向上率が維持すると消費電力の絶対量は増え続けると予測した²¹⁾。しかし、²¹⁾においては省電力化手法は提案されていない。Xu らは²⁰⁾において TPC-H に加え、彼らの power-aware なクエリオプティマイザの評価実験で TPC-C を用いた。Xu らの手法では省電力化のために DVFS を用いておらず、また実行時の負荷変動を考慮していないという点において本研究とは異なる。

6. おわりに

本論文では、OLTP におけるアプリケーション指向省電力化手法を提案し、その効果を実測により評価した。我々の提案する手法では、Dynamic Voltage and Frequency Scaling を利用し、実行時のアプリケーション性能情報を活用することにより、動的な負荷変動に追従した省電力化を実現している。応答時間とスループットそれぞれを指標とした DVFS の制御ポリシーを設計して評価実験を行い、アプリケーションの性能目標を達成しつつ省電力化できていることを確認した。実サーバの負荷トレースをもとにした評価実験では、応答時間指向 DVFS ポリシーを用いた場合には 10.0W(全体の 6.4%)、スループット指向 DVFS ポリシーを用いた場合には 11.8W(全体の 7.5%) の消費電力削減効果が確認された。

今後は、データベースサーバだけでなくフロントエンドサーバまでを含めた OLTP のシステム全体の省電力化について取り組んでいきたいと考えている。

参 考 文 献

- 1) EPA: EPA Report to Congress on Server and Data Center Energy Efficiency, Technical report, U.S. Environmental Protection Agency (2007).
- 2) Network Power, E.: Energy Logic: Reducing Data Center Energy Consumption by Creating Savings that Cascade Across Systems, White paper, Emerson Electric Co. (2009).
- 3) Shanley, K.: TPC Releases New Benchmark: TPC-C, *SIGMETRICS Performance Evaluation Review*, Vol.20, No.2, pp.8-9 (1992).

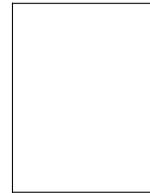
- 4) Arlitt, M. and Jin, T.: Workload Characterization of the 1998 World Cup Web Site, Technical report, Hewlett Packard Laboratories Palo Alto (1999).
- 5) Herbert, S. and Marculescu, D.: Analysis of dynamic voltage/frequency scaling in chip-multiprocessors, *ISLPED '07: Proceedings of the 2007 international symposium on Low power electronics and design*, New York, NY, USA, ACM, pp.38-43 (2007).
- 6) Herbert, S. and Marculescu, D.: Variation-aware dynamic voltage/frequency scaling, *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*, pp.301-312 (2009).
- 7) Chen, S., Joshi, K.R., Hiltunen, M.A., Schlichting, R.D. and Sanders, W.H.: Blackbox prediction of the impact of DVFS on end-to-end performance of multitier systems, *SIGMETRICS Perform. Eval. Rev.*, Vol.37, No.4, pp.59-63 (2010).
- 8) Lee, W.Y.: Energy-Saving DVFS Scheduling of Multiple Periodic Real-Time Tasks on Multi-core Processors, *DS-RT '09: Proceedings of the 2009 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, Washington, DC, USA, IEEE Computer Society, pp.216-223 (2009).
- 9) Lee, W.Y., Ko, Y.W., Lee, H. and Kim, H.: Energy-efficient scheduling of a real-time task on DVFS-enabled multi-cores, *Proceedings of the 2009 International Conference on Hybrid Information Technology*, ICHIT '09, New York, NY, USA, ACM, pp.273-277 (2009).
- 10) Elnozahy, M., Kistler, M. and Rajamony, R.: Energy conservation policies for web servers, *Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, USITS'03, Berkeley, CA, USA, USENIX Association, pp.8-8 (2003).
- 11) Rusu, C., Ferreira, A., Scordino, C. and Watson, A.: Energy-Efficient Real-Time Heterogeneous Server Clusters, *12th IEEE RealTime and Embedded Technology and Applications Symposium RTAS06*, pp.418-428 (2006).
- 12) Agrawal, R., Ailamaki, A., Bernstein, P.A., Brewer, E.A., Carey, M.J., Chaudhuri, S., Doan, A., Florescu, D., Franklin, M.J., Garcia-Molina, H., Gehrke, J., Gruenwald, L., Haas, L.M., Halevy, A.Y., Hellerstein, J.M., Ioannidis, Y.E., Korth, H.F., Kossmann, D., Madden, S., Magoulas, R., Ooi, B.C., O'Reilly, T., Ramakrishnan, R., Sarawagi, S., Stonebraker, M., Szalay, A.S. and Weikum, G.: The Claremont report on database research, *SIGMOD Rec.*, Vol.37, No.3, pp.9-19 (2008).
- 13) Graefe, G.: Database Servers Tailored to Improve Energy Efficiency, *Proceedings of EDBT'08 Workshop on Software Engineering for Tailor-made Data Management*, pp.24-28 (2008).
- 14) Harizopoulos, S., Shah, M.A., Meza, J. and Ranganathan, P.: Energy Efficiency: The New Holy Grail of Data Management Systems Research, *CIDR 2009, Fourth*

Biennial Conference on Innovative Data Systems Research (2009).

- 15) Meza, J., Shah, M.A., Ranganathan, P., Fitzner, M. and Veazey, J.: Tracking the power in an enterprise decision support system, *Proceedings of the 2009 International Symposium on Low Power Electronics and Design*, pp.261–266 (2009).
- 16) Poess, M. and Nambiar, R.O.: A power consumption analysis of decision support systems, *WOSP/SIPEW '10: Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering*, New York, NY, USA, ACM, pp. 147–152 (2010).
- 17) Poess, M. and Nambiar, R.O.: Tuning servers, storage and database for energy efficient data warehouses, *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010*, pp.1006–1017 (2010).
- 18) Tsirogiannis, D., Harizopoulos, S. and Shar, M.A.: Analyzing the Energy Efficiency of a Database Server, *SIGMOD '10: Proceedings of the 36th SIGMOD international conference on Management of data*, New York, NY, USA, ACM (2010).
- 19) Lang, W. and Patel, J.M.: Towards Eco-friendly Database Management Systems, *CIDR 2009, Fourth Biennial Conference on Innovative Data Systems Research* (2009).
- 20) Xu, Z., Tu, Y.-C. and Wang, X.: Exploring power-performance tradeoffs in database systems, *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010*, pp.485–496 (2010).
- 21) Poess, M. and Nambiar, R.O.: Energy cost, the key challenge of today's data centers: a power consumption analysis of TPC-C results, *Proc. VLDB Endow.*, Vol.1, No.2, pp.1229–1240 (2008).

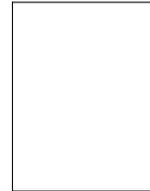
(平成 22 年 12 月 20 日受付)
(平成 23 年 4 月 7 日採録)

(担当編集委員 樋口 健)



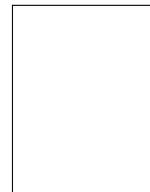
早水 悠登 (学生会員)

平成 21 年, 東京大学工学部電子情報工学科卒業. 平成 23 年, 同大学院情報理工学系研究科電子情報学専攻終了. 現在, 同専攻博士後期課程 1 年. データベースシステムの省電力化に関する研究に従事. 情報処理学会学生会員.



合田 和生 (正会員)

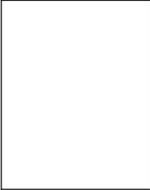
平成 12 年, 東京大学工学部電気工学科卒業. 平成 17 年, 同大学院情報理工学系研究科電子情報学専攻博士課程単位取得満期退学. 同年, 博士 (情報理工学). 現在, 東京大学生産技術研究所特任助教. 高速データベースエンジン, 高機能ストレージシステムの研究に従事. 情報処理学会, 日本データベース学会, ACM, IEEE CS, USENIX 各会員.



中野美由紀 (正会員)

東京大学理学部情報科学科卒業. 博士 (情報理工学). 富士通 (株) 勤務. 昭和 60 年年 7 月, 東京大学生産技術研究所助手 (2004 年助教). 平成 20 年 7 月, 特任准教授. データベースシステム, ストレージシステム, データ工学の研究に従事. IEEE, 電子情報通信学会, 情報処理学会, ACM, 日本データベース学会各会員.

喜連川 優 (正会員)



東京大学工学系研究科情報工学専攻博士課程修了 (1983 年), 工学博士.
東京大学生産技術研究所 講師, 助教授を経て, 現在, 同教授. 東京大学
地球観測 データ統合連携研究機構長, 東京大学生産技術研究所戦略情
報融合国際研究センター長. 文部科学官. 文部科学省「情報爆発」特定研
究領域代表 (2005- 2010), 経済産業省「情報大航海プロジェクト」戦略会
議委員長 (2007-2009), 情報処理学会フェロー, 副会長 (08-09). データベース工学の研究
に従事.
