Vol.2010-OS-115 No.22
2010/8/4

マルチコア CPU の電力消費特性を考慮した仮想 CPU スケジューラ

吉田哲也[†1]　山田浩史[†1,†3]　佐々木広[†2]
河野健二[†1,†3]　中村宏[†2]

クラウド環境を提供する仮想化データセンタにおける消費エネルギーを削減するために，Dynamic Voltage and Frequency Scaling (DVFS) を利用した CPU の電力制御が行われている．ところが，既存のマルチコア CPU には，全てのコアの周波数を下げないと消費電力の削減効果が小さいという特性がある．既存の VM 環境は，この性質を考慮していないため，効率よく消費エネルギーを削減できていない．本論文では，VM 環境において，マルチコア CPU の電力消費特性を考慮した仮想 CPU (VCPU) スケジューラである Accele スケジューラを提案する．Accele スケジューラは，VCPU の周波数を考慮してスケジューリングを行い，全てのコアに低い周波数の VCPU がスケジューリングされる確率を高くする．その結果，全てのコアの周波数が下がる確率が高くなり，DVFS による消費エネルギー削減が大きくなる．SysBench と SPEC CPU2006 を用いて実験を行った結果，Xen のスケジューラである Credit スケジューラと比べ，消費エネルギーを最大 22.8% 小さくすることができた．

Scheduling Virtual CPUs with Considering a Caractaristic Power Consumption of Mulit-Core CPUs

Tetsuya Yoshida,[†1] Hiroshi Yamada,[†1]
Hiroshi Sasaki,[†2] Kenji Kono[†1]
and Hiroshi Nakamura[†2]

Dynamic voltage and frequency scaling (DVFS) is performed to reduce energy consumption in virtualized data centers used as the platform of cloud computing. However, DVFS provides inefficient energy savings in existing VM environments, because of a characteristic of multi-core CPUs that DVFS cannot save a lot of energy unless the frequencies of all cores are lowered. In this paper, we propose a energy-conscious Virtual CPU (VCPU) scheduler that takes the characteristic of multi-core CPUs into account. Our scheduler, named Accele scheduler , schedules VCPUs with considering the frequency of them to increase the probability that all cores run VCPUs having lower frequencies simultaneously. As a result, the frequencies of all cores are lowered simultaneously in high probability, and therefore we can gain higher energy savings with DVFS. We evaluate Accele scheduler with SysBench and SPEC CPU2006 by comparing Xen's Credit scheduler. The evaluation showed that Accele scheduler reduced the energy consumption of the CPU by up to 22.8% compared with Credit scheduler.

1. はじめに

近年，クラウドコンピューティングへの関心が高まっている[2]．クラウド環境で動作するアプリケーションは，必要な量だけハードウェアリソースを使うことができるため，余分なリソースを用意することによるコストの浪費や，リソース不足によるアプリケーションの性能低下を避けることができる．このようなクラウド環境は，仮想化技術を利用したコンソリデーションによって実現している．各アプリケーションを仮想マシン (VM) 上で動かすことで，ハードウェアリソース量を柔軟に変更できる．

クラウドコンピューティングへの関心が高まる一方で，クラウド環境を提供するデータセンタにおける消費電力の増加が問題となっている．その理由の 1 つとして，データセンタの運用に必要な電力コストが高いことがあげられる．ヒューレットパッカード社の調べによると，1.3 MW 規模のデータセンタにおける年間の電力コストは 1 億 2 千万円にのぼるとされている[6]．そこで，VM 環境を利用したデータセンタにおける消費電力を削減するために，様々な研究が行われている[9),13),14),17),19),20)]．

データセンタにおける電力消費量を削減する方法の一つとして，サーバマシンの CPU の消費電力削減が挙げられる．近年の CPU は，動作周波数と動作電圧を動的に変更する Dynamic Voltage and Frequency Scaling (DVFS) [1),8)] を備えており，ワークロードに合わせて DVFS を利用することで消費エネルギーを減らすことができる．

ところが，既存のマルチコア CPU の消費電力は，最速のコアの周波数に大きく依存するという特性がある[13)]．マルチコア CPU は，動作周波数を各コア毎に設定できる[1),8)]．その一方で，動作電圧は全てのコアで共通となる．動作電圧を制御するレギュレータは面積が大きく，コアごとに設置することが難しいため，全てのコアでレギュレータを共有していること

†1 慶應義塾大学
†2 東京大学
†3 JST CREST

が原因である[15),22)]．この制限から，一番高い周波数のコアが必要とする動作電圧が全てのコアにかかる．CPU の消費電力 P，動作周波数 F，動作電圧 V の間には，$P \propto FV^2$ の関係があり，動作電圧の大きさは消費電力に大きく影響する．したがって，マルチコア CPU において，DVFS を使って効率よく消費電力を下げるためには，全てのコアの周波数を揃えて下げる必要がある．ところが，既存の VM 環境では，このようなマルチコア CPU の特性が考慮されていない．

本研究では，VM 環境において，マルチコア CPU の電力消費特性を考慮した仮想 CPU (VCPU) スケジューラである Accele スケジューラを提案する．Accele スケジューラは，全てのコアの周波数を揃えて下げられる確率が高くなるように VCPU をスケジューリングすることで，DVFS による消費電力の削減効果を大きくする．そのため Accele スケジューラは，各コアのランキュー内の VCPU を一定周期で周波数順にソートする．これにより，各コアで周波数が低い VCPU から順にスケジューリングされるため，全てのコアの周波数が下がる確率が高くなる．そして，低い周波数の VCPU が全てのコアに分散するようにランキューを構成することで，低い周波数の VCPU が同時にスケジューリングされる確率をさらに高くする．また，Accele スケジューラはクラウド環境での利用を想定しているため，プロポーショナルシェアスケジューリングを行う．

ここで Accele スケジューラは，各 VCPU の周波数は適切に設定されていることを前提とする．VM 環境で DVFS を行う場合，各ゲスト OS や VMM が持つ電力制御機構が，各 VCPU の周波数を決定する[13)]．本研究は，電力制御機構によって周波数が決められた VCPU を適切にスケジューリングすることで，消費エネルギーを削減する．VCPU の周波数を決める方法は本研究の対象外である．

また，本研究は CPU の消費エネルギーを削減することでエネルギー効率を上げることを目的としているため，CPU 使用率の高いワークロードを対象とする．具体的には，数値計算やインデックス計算のようなワークロードが挙げられる．そのため，CPU 使用率が低い I/O バウンドなワークロードは本研究の対象外とする．

Accele スケジューラをオープンソースの VMM である Xen[3)] に実装した．そして，Accele スケジューラの効果を調べるために，Xen のデフォルトスケジューラである Credit スケジューラと比較実験を行った．実験では，SysBench と SPEC CPU2006 を使い，消費エネルギーを測定した．その結果，実験に用いた全てのワークロードの組み合わせにおいて，Accele スケジューラの方が消費エネルギーが小さくなり，その差は最大で 22.8% となった．これにより，Accele スケジューラが CPU の消費エネルギーを大きく削減できることが示された．

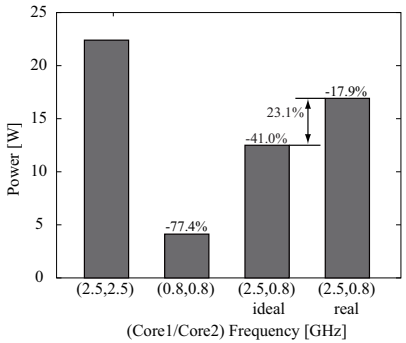


図 1　コア周波数の違いによる CPU の消費電力の変化

以降の構成は次の通りである．まず，第 2 章で本研究の背景について述べる．そして，第 3 章でマルチコア CPU の電力消費特性に関して述べる．第 4 章で本研究の提案である Accele スケジューラについて述べ，第 5 章でその実装について述べる．提案手法の有効性を示すために行った実験について第 6 章で述べ，第 7 章では本研究に関連する既存研究について述べる．最後に，第 8 章で，本論文をまとめる．

2. マルチコア CPU の電力消費特性

既存のマルチコア CPU は，全てのコアの周波数を下げないと DVFS による消費電力の削減が小さいという特性がある[13)]．これは，周波数が一番高いコアが必要とする動作電圧が，全てのコアにかかることが原因である．動作電圧を制御するレギュレータは面積が大きく，コアごとに設置することが難しい．そのため，全てのコアでレギュレータを共有しており，全てのコアで動作電圧が同一になる[15),22)]．したがって，動作電圧を下げるためには，全てのコアの周波数を下げる必要がある．

2.1 予 備 実 験

先に挙げたマルチコア CPU の特性を実証するために予備実験を行った．2 コアのマルチコア CPU の各コア上で無限ループを実行し，各コアの周波数を変えて消費電力を測定した．実験に用いた CPU は，AMD Optern Quad-Core 2384 である．この CPU は，各コアの周波数を 2.5GH，1.8GHz，1.3GHz，0.8GHz に設定できる．本実験では，最高周波数の 2.5GHz と最低周波数の 0.8 GHz を用い，全ての組み合わせ 1) 2.5GHz と 2.5GHz, 2) 0.8GHz と 0.8GHz, 3) 2.5GHz と 0.8GHz における消費電力を測定した．ここで，4 コアの内，利用しない 2 コ

アは 最低周波数の 0.8GHz に設定し，常にアイドル状態とした．

実験結果を図 1 に示す．X 軸は周波数の組み合わせ (f_1, f_2)，Y 軸は各場合における CPU の消費電力である．結果は，1) で 21.2 W，2) で 3.8 W，3) で 16.4W であり，1) の場合を基準とした消費電力の削減率は，2) で 77.4 %，3) で 17.9 % となった．ここで，消費電力が最速のコアに依存せず，各コアの周波数のみで決まると仮定した理想ケースを考えると，(2.5, 0.8) の消費電力は，1) と 3) の平均値である 12.5 W になる．この時，消費電力の削減率は 41.0% となる．これを，図 1 において，(2.5,0.8) の理想ケースとして示している．ところが実際の測定値である 2) は，この理想値より 3.9W 大きく，消費電力削減率の差に 23.1 % という大きな差がある．これより，マルチコア CPU の消費電力は最速のコアの周波数に依存することが確かめられた．

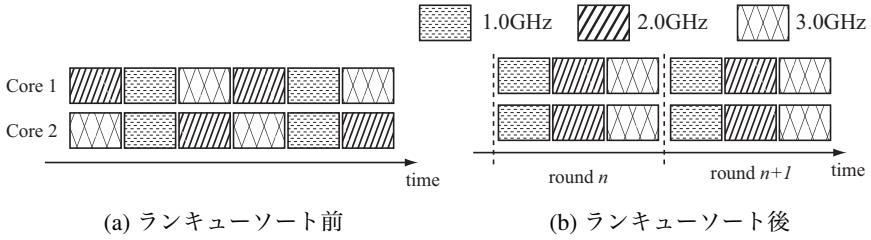
2.2 マルチコア CPU の特性を考慮した VCPU スケジューリングの必要性

前節の実験より，マルチコア CPU の消費電力は，最速のコアの周波数に依存することがわかった．このことから，マルチコア CPU において DVFS を行う場合，全てのコアの周波数を揃えて下げることで消費電力を大きく下げることができると考えられる．ここで本研究では，クラウド環境における消費電力削減を目的としているため，VM 環境における消費電力削減を考える．VM 環境では，VMM が持つ VCPU スケジューラが，物理 CPU に VCPU を割り当てることで，物理 CPU をシェアしている．そこで，VCPU をスケジューリングする際，全てのコアに低い周波数の VCPU をスケジューリングすることで，消費電力を大きく削減できると考えられる．ところが既存の VCPU スケジューラは，VCPU の周波数を考慮せずにスケジューリングを行うため，全てのコアに低い周波数の VCPU がスケジューリングされる確率が小さい．

ここで，簡単な例として，コア数 c，設定可能な周波数が n であるマルチコア CPU 上におけるスケジューリングを考える．そして，各周波数で動作する VCPU が c 個ずつあるとする．この例の場合，同じ周波数の VCPU を組にしてスケジューリングすることで，全てのコアの周波数を揃えて下げることができる．この確率を $p(c,n)$ とおくと，

$$p(c,n) = \frac{1}{n^{c-1}} \tag{1}$$

で表される．例えば，先の実験で用いた AMD Opteron の場合，コア数が 4，設定可能な周波数が 4 種類であるため，$p(4,4) = \frac{1}{64}$ となる．したがって，VCPU の周波数を考慮せずにスケジューリングした場合，同じ周波数の VCPU が組になってスケジューリングされる確率は小さいといえる．また，式 (1) から，コア数が増えると，$p(c,n)$ が指数関数的に小さくなることがわかる．CPU のコア数はこれからも増え続けていくといわれているため[5]，全てのコアに低い周波数の VCPU がスケジューリングされる確率はさらに小さくなると予想される．



(a) ランキューソート前　(b) ランキューソート後

図 2　ランキューソート

以上のことから，全てのコアに低い周波数の VCPU をスケジューリングするためには，VCPU の周波数を考慮したスケジューリングが必要であるといえる．

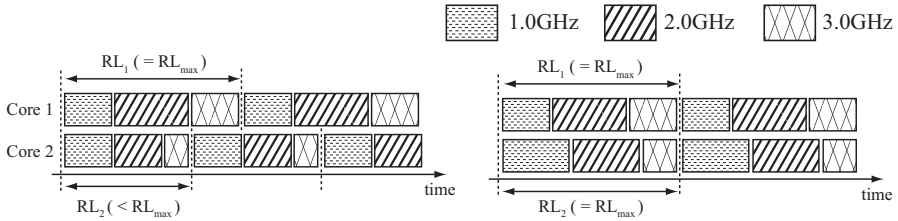
3. 提案: Accele スケジューラ

本研究では，マルチコア CPU の消費電力特性を考慮した VCPU スケジューラである Accele スケジューラを提案する．先の実験で示した通り，マルチコア CPU の消費電力を大きく削減するためには，全てのコアの周波数を下げる必要がある．そこで，全てのコアに低い周波数の VCPU をスケジューリングする確率を高くすることで，DVFS による消費電力削減効果を大きくする．さらに，クラウド環境で一般に行われている VM ごとの優先度割り当てを可能にするため，プロポーショナルシェアスケジューリングを行う．

ここで Accele スケジューラは，VCPU の周波数が適切な値に設定されていることを前提とし，VCPU の周波数は変更しない．仮想化環境では，各ゲスト OS が電力制御を行うことで，VCPU の周波数を決定する[13),17)]．

3.1 ランキューソート

Accele スケジューラは，全てのコアに低い周波数の VCPU をスケジューリングする確率を高くするために，各ランキューが持つ VCPU を周波数の昇順にソートする．これにより，各コアにおいて一定周期で周波数の低い VCPU から順にスケジューリングされるため，全てのコアで同時に低い周波数の VCPU がスケジューリングされる確率が高くなる．ここで，各ランキューが持つ全ての VCPU をスケジューリングする 1 つの周期をラウンドと呼ぶ．ランキューソートの様子を図 2 に示す．ランキューをソートする前は，周波数を考慮せず



(a) ラウンド長統一前　　(b) ラウンド長統一後

図 3　ラウンド長の統一

に VCPU が並んでいるため，全てのコアに低い周波数の VCPU がスケジューリングされる確率が低い（図 2(a)）．一方，図 2(b) のようにランキューソートを行うと，低い周波数の VCPU が同時にスケジューリングされる確率が高くなる．

ランキューソートによって低い周波数の VCPU を同時にスケジューリングするためには，全ランキューのラウンドが同じタイミングで始まる必要がある．各ランキューの VCPU が周波数順にソートされていても，それらを同じタイミングでスケジューリングし始めなければ，低い周波数の VCPU が同時にスケジューリングされないためである．この問題に対処するため，全てのコアで同期を取り，ラウンドの開始時間を揃える．ただし，ラウンドの開始時に毎回コア間で同期を取ると，頻繁にコア間の同期処理が発生し，同期のオーバーヘッドが大きくなってしまう．そのため，コア間で同期を取るのは一定周期 EL (e.g. 1 sec) に一度とする．この周期をエポックと呼ぶ．

さらに，各ランキューのラウンド長が異なると，エポック内でラウンドの開始時間がずれてしまい，周波数の低い VCPU が同時にスケジューリングされなくなってしまう（図 3(a)）．ラウンドの長さは，ランキューが持つ VCPU のタイムスライスの合計となる．そこで，タイムスライスの和が最も長いランキューのラウンド長を基準とし，他のランキュー内の VCPU のタイムスライスを一時的に長くすることで，全ランキューのラウンド長を同じにする（図 3(b)）．ラウンド内における VCPU i のタイムスライスを RS_i，VCPU i が属しているランキュー j のラウンド長を RL_j，最長のラウンド長を RL_{max} とすると，VCPU i のタイムスライスを，$RS'_i = RS_i * \frac{RL_{max}}{RL_j}$ とすることで，全てのランキューのラウンド長が等しくなる．

ただし，Accele スケジューラはプロポーショナルシェアリングを行うため，各 VCPU の実行時間は予め設定された割合（シェア）に従う必要がある．そこで，一時的にタイムスライスを長くした VCPU は，後にタイムスライスを短くすることでシェアを保つ．この方法については，3.3 節で詳述する．

3.2　ランキューの構成

ランキューソートによって低い周波数の VCPU が同時にスケジューリングされるようにするためには，低い周波数の VCPU が各ランキューに分散している必要がある．低い周波数の VCPU を偏って配置してしまうと，低い周波数の VCPU を多く持つランキューと，高い周波数の VCPU を多く持つランキューができてしまい，全てのコアに低い周波数の VCPU をスケジューリングできる確率が低くなってしまう．例えば，1 GHz の VCPU と 3 GHz の VCPU が複数ある場合に，一つのランキューに 1 GHz の VCPU を全て入れてしまうと，全てのコアに 1GHz の VCPU をスケジューリングすることができない．そこで Accele スケジューラは，VCPU の周波数を考慮してランキューを構成し，VCPU を周波数に関して均等に分配する．ただし，Accele スケジューラはプロポーショナルスケジューリングを行うため，ランキューを構成する際には VCPU のシェアも考慮する．

シェアを考慮しつつ，周波数に関して均等に VCPU を配分するため，以下の手順でランキューの構成を行う．まず，全ての VCPU を周波数ごとに分類し，リストを作る．そして，各リストの VCPU を，シェアの大きさの昇順にソートする．そして，周波数が低いリストの先頭から順に，保持している VCPU のシェアの合計が最小のランキューへ格納していく．この方法を，VCPU が無くなるまで繰り返す．この方法により，周波数が低い VCPU から順に格納していくため，VCPU が周波数に関して均等に分散される．また，シェア順にソートし，シェアの和が最小のランキューへ VCPU を格納していくため，各ランキューのシェアの合計の差が小さくなり，シェアに関してもほぼ均等にすることができる．

ここで，上記の方法を用いて構成したランキューが，常に同じ構成であり続けるとは限らない．その原因として，VCPU のスリープや VM 起動による VCPU の新規作成などが挙げられる．また上記の方法では，ランキューの長さが全て同一になるとは限らないため，プロポーショナルシェアリングを満たせない場合がある．ランキューの長さが異なる場合，前節で述べたラウンド長の統一を行うためである．この問題に対処するため，一定周期に一度ランキューの再構成を行う．ここで，ランキューの構成を変更する際は，全てのコアのランキューを操作するため，全てのコアで同期を取る必要がある．そこで，3.1 節で述べたエポックの開始時にランキューの再構成も行うことで，コア間の同期頻度を少なくする．

3.3　タイムスライスの調整

Accele スケジューラはプロポーショナルシェアリングを行うため，各 VCPU のタイムスライスは，それぞれに設定されたシェアによって決まる．ところが，VCPU がスリープした

場合や，3.1 節で述べたラウンド長の調整などにより，タイムスライスがシェアで決められる値と異なってしまう場合がある．そこで，タイムスライスの調整を行うことで，プロポーショナルシェアを実現する．

まず，各 VCPU の基本となるタイムスライスを定める．1 つのラウンドで VCPU i に与えるタイムスライスを RS_i とし，この値を各 VCPU のシェアに従って決める．基準となる時間を BS，VCPUi のシェアを S_i，全 VCPU のシェアの平均を $\bar{S}$ とし，タイムスライス RS_i を $RS_i = BS * \frac{S_i}{\bar{S}}$ と定める．これにより，各 VCPU のシェアの比によって，各 VCPU のタイムスライスが求まる．このタイムスライス RS_i を，VCPUi の基本タイムスライスとする．

そして，エポック α で VCPU i に与えるタイムスライス $ES_{\alpha,i}$ を，エポックが含むラウンド数 n と基本タイムスライスから，$ES_{\alpha,i} = n * RS_i$ とする．

プロポーショナルシェアが保てていない場合，あるエポック α における VCPU i のタイムスライス $ES_{\alpha,i}$ と，VCPU i がエポック α で実際に動作した時間 $EU_{\alpha,i}$ が異なる．そこで，エポック開始時にタイムスライスのずれを考慮し，各 VCPU のシェアを調整してからランキューを再構成することで，プロポーショナルシェアを保つ．

エポック $\alpha+1$ における VCPU i のシェアの値 $S_{\alpha+1,1}$ を，$S_{\alpha+1,i} = S_{\alpha,i} * \frac{ES_i}{EU_i}$ とする．この式から，VCPU i が ES_i より長い時間動作していた場合，次のエポックにおけるシェアが小さくなるため，与えられるタイムスライスが短くなる．逆に，ES_i より短い時間しか動作しなかった場合，次のエポックにおけるシェアが大きくなるため，与えられるタイムスライスが長くなる．また，この式で決められたシェアを使って，3.2 節で述べたランキューの再構成を行うため，各 VCPU のシェアのずれを調整するようにランキューが構成される．

4. 実　　装

Accele スケジューラを，オープンソースの VMM である Xen 3.4.1 に実装した．Xen は，Credit スケジューラや SEDF スケジューラなど，複数のスケジューリングポリシーを持っており，起動時にこれらを選択することができる．そこで，Xen のスケジューリングポリシーの 1 つとして，Accele スケジューラを実装した．スケジューラのソースコードは，C 言語で 1274 行である．

本実装では，各 VCPU のタイムスライスの基準である BS を 30 msec とした．これは，Credit スケジューラにおけるタイムスライスの長さの最小単位であり，タイムスライスの長さとして用いるのに妥当な数字といえる．エポックの長さ EL は，1 sec に設定した．

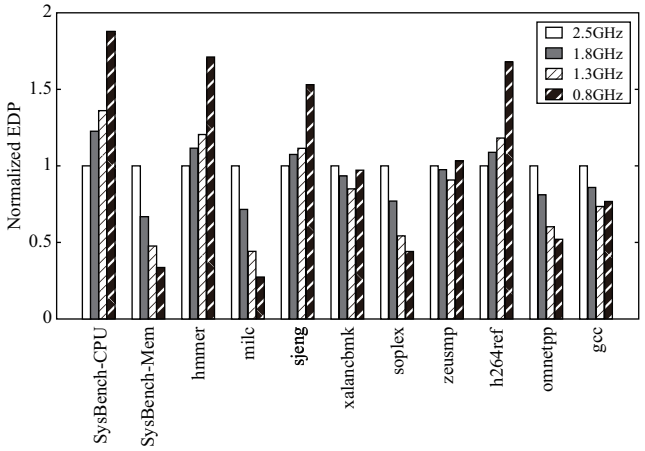


図 4　各ワークロードの CPU 周波数に対する EDP の値

5. 実　　験

Accele スケジューラが CPU の消費エネルギーを小さくできることを示すために実験を行った．Accele スケジューラと Xen のデフォルトスケジューラである Credit スケジューラで同様の実験を行い，結果を比較した．Credit スケジューラは，マルチコア CPU 環境において VM 間でプロポーショナルシェアを行う VCPU スケジューラである．

5.1 実験環境

複数の VCPU 上でワークロードを動作させ，CPU の消費エネルギーとワークロードの実行時間を測定した．また，エネルギー効率を表す値である Energy Delay Product (EDP) を算出し，エネルギー効率の比較も行った．EDP はスループット当たりの消費エネルギーを表す指標であり，EDP が小さいほどエネルギー効率が高い．

ベンチマークとして，Sysbench と SPEC CPU2006 に含まれるワークロードを用いた．SysBench は，CPU バウンド，メモリバウンド，I/O バウンドなど，様々なワークロードを含んでいる．今回は，CPU バウンドなワークロード（SysBench-CPU）とメモリバウンドなワークロード（SysBench-Mem）を用いて実験を行った．SPEC CPU2006 からは，CPU バウンドなワークロードである h264ref, hmmer, sjeng と，メモリバウンドなワークロードである gcc, milc, omnetpp, soplex を用いた．

表 1　実験で用いたワークロードの組み合わせ）

シナリオ	ワークロード		
	VM1	VM2	VM3
1	SysBench-CPU (2.5GHz)	Sys-Mem (1.3GHz)	-
2	hmmer (2.5GHz)	milc (1.3GHz)	-
3	sjeng (2.5GHz)	xalancbmk (0.8GHz)	-
4	zeusmp (1.3GHz)	soplex (0.8GHz)	-
5	h264ref (2.5GHz)	gcc(1.3GHz)	omnetpp (0.8GHz)

ここで，Accele スケジューラは，各 VCPU の周波数が適切な値に設定されていることを前提としている．そこで予備実験を行い，ワークロードごとに最適な周波数を決定した．各ワークロードを，実験で用いる CPU で設定可能な全ての周波数で動作させ，EDP が最小となる周波数を，そのワークロードにおける最適な周波数とする．実験に用いた CPU は，2.1 節の予備実験と同じく AMD Opteron Quad-Core 2384 である．予備実験の結果は図 4 に示す通りとなった．

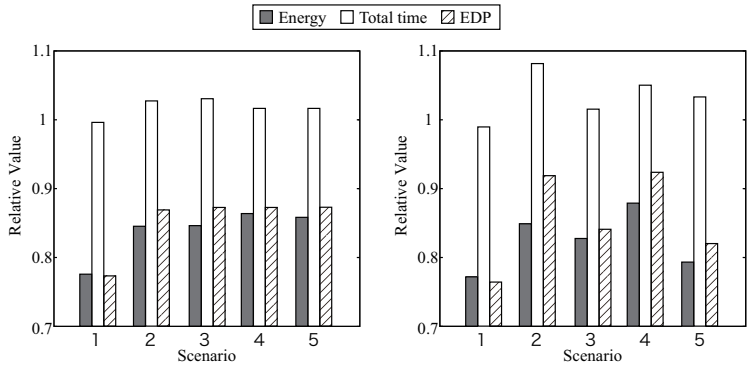
予備実験の結果から各ワークロードを動かす周波数を決定し，表 1 に示すシナリオで実験を行った．シナリオ 1 は SysBench を用いた．シナリオ 2 から 5 では SPEC CPU2006 のワークロードを用い，最適な周波数が 2.5GHz のもの，1.3GHz のもの，0.8GHz のものを様々に組み合わせて実験を行った．

マシン環境は次の通りである．CPU は AMD Opteron Quad-Core 2384，メモリは 16GB DDR2 SDRAM を使用した．各 VM に割り与えるメモリ容量は，VCPU 数が 2 の場合は 2GB，VCPU 数が 4 の場合は 3GB とした．物理 CPU のコア数と各 VM の VCPU 数は等しくし，コア数が 2 の場合，4 の場合の 2 通りで実験を行った．CPU の消費電力の測定には，シネジェテック社の電力測定器を用い，測定周期は 1 kHz とした．

5.2 実 験 結 果

実験結果を図 5 に示す．X 軸はシナリオ番号，Y 軸は各シナリオに関して，消費エネルギー，処理時間，EDP の値を表している．また各値は，Credit スケジューラの場合を 1 とした相対値である．グラフより，コア数 2, 4 双方において，全てのシナリオで Accele スケジューラの消費エネルギーが，Credit スケジューラの場合より小さくなっていることがわかる．消費エネルギーの減少は最大で 22.8%，最小で 12.1% となった．このことから，Accele スケジューラが消費エネルギーを大きく削減できていることがわかる．

ところが，消費エネルギーが減少している一方で，ワークロードの実行時間が増加している．処理時間の増加は，最大 8.3% となった．これは，メモリバスへのアクセスが競合する



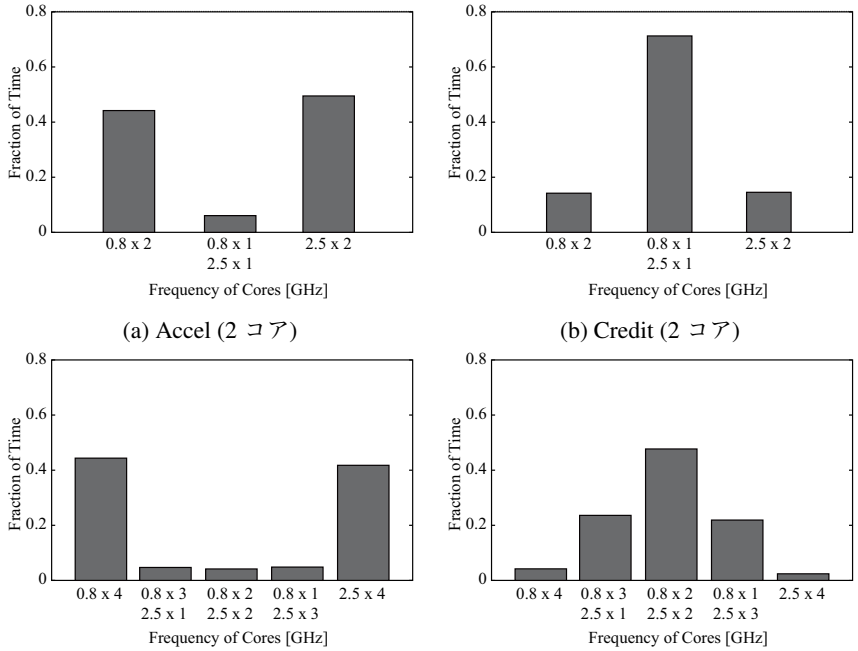
(a) コア数 2　　(b) コア数 4

図 5　各シナリオにおける消費エネルギー，処理時間，EDP の比較

ことによるパフォーマンス低下が原因と考えられる．Accele スケジューラは，コアの周波数を揃えるように VCPU をスケジューリングするため，メモリバウンドなワークロードが複数のコアで同時にスケジューリングされる確率が高い．これによって，メモリバスに対するアクセスが競合し，ワークロードのスループットが低下してしまう．実際，既存研究において，マルチコア CPU ではメモリバスの競合によってシステムのパフォーマンスが下がると報告されている[12]．

ここで，消費エネルギーと処理時間の双方を考慮して得られる値である EDP を比較すると，全てのシナリオにおいて，Accele スケジューラの方が小さくなっている．したがって，処理時間の増加より消費エネルギーの減少の効果の方が大きく，結果として Accele スケジューラを使うと CPU のエネルギー効率が上がることがわかった．このことから Accele スケジューラは，消費エネルギーを減少させ，さらにエネルギー効率も高められるといえる．

さらに，コア数が 2 つの場合と 4 つの場合を比較すると，次のことがわかる．まず，消費エネルギーに関しては，コア数 4 の場合の方が，Credit スケジューラと比較して小さい値となっている．これは，2.2 節で述べた通り，コア数が多くなると，偶然に全てのコアの周波数が揃う確率が小さくなるためと考えられる．逆に，処理時間を比較すると，コア数 4 の場合の方が，Credit スケジューラに対する処理時間が大きくなっている．これは，コア数が多い方が，同時にスケジューリングされるメモリバウンドなワークロードの数が多くなるため，メモリアクセスの競合が大きくなるためと考えられる．マルチコア CPU のコア数はさ



(a) Accel (2 コア)　(b) Credit (2 コア)

(c) Accel (4 コア)　(d) Credit (4 コア)

図 6　シナリオ 1 における各コア周波数の組み合わせの動作時間

らに増加していくと言われているため，今後は，メモリバスの競合とマルチコア CPU の電力消費特性の双方を考慮にするスケジューラが必要であると考えられる．

ここで，Accele スケジューラが同じ周波数のコアを同時にスケジューリングしていることを示すため，シナリオ 1 の実験における VCPU スケジューリングのログを記録した．結果を図 6 に示す．X 軸は各コアの VCPU の周波数の組み合わせ，Y 軸は各組み合わせがワークロードの実行時間全体に占める割合を表す．結果のグラフから，Accele スケジューラでは，コア数 2，4 の場合共に同じ周波数の VCPU がスケジューリングされている時間が長いことがわかる．コア数 2 の場合，全ての VCPU が 2.5GHz で揃っている場合が実行時間全体の 44.4%, 0.8GHz で揃っている時間が 49.6% となっており，動作時間の 94.0% という高い割合で 2 つのコアの周波数が揃っている．一方，Credit スケジューラの場合，2 つのコアの周波数が揃っているのは全体の 27.3% という低い割合となっている．コア数 4 の場合も同様に，Accele スケジューラの場合は高い割合で全てのコアの周波数が揃っている．これにより，Accele スケジューラでは，周波数の低いものから組み合わせて実行することができていることがわかる．ここで，Credit スケジューラの場合，コア数が 2 の場合より 4 の場合の方が，全てのコアの周波数が揃う確率が小さくなっている．これは，2.2 節で述べた通り，VCPU の周波数を考慮せずにスケジューリングすると，コア数が増えるに従って，全てのコアの周波数が揃う確率が小さくなることを示している．

その他のシナリオにおいてもスケジューリングのログを調べた結果，全てのシナリオにおいて，Accele スケジューラは高い確率で全てのコアの速度を揃えてスケジューリングを行っていることが確かめられた．

6. 関 連 研 究

データセンタにおける消費電力を削減するために，VM 環境における電力制御手法が研究されている．Nathuji らは，VM 環境において，VM と VMM が協調して電力制御を行うためのフレームワークである VirtualPower を提案している[13]．各ゲスト OS が VCPU に対して発行する DVFS 関連の命令を VMM がフックすることで，VM が行う電力制御を考慮しながら VMM による電力制御を行う．また，Stoess ら[17] や Kansal [9] らは，VM 環境において，システム全体の消費エネルギーを，各 VM にアカウンティングする手法を提案している．各 VM に対して正確に消費エネルギーをアカウンティングすることで，各 VM が定めたエネルギー内で動作するように動作制限を行う．pMapper [19] は，VM の移送を利用し，エネルギー効率のよいマシンに VM を集約することで消費電力を削減する．Accele スケジューラは，このような VM 環境における電力制御機構と補完的に動作することで，さらに消費エネルギーを削減する．

Weiser らが実システムのトレースを用いて DVFS の有用性を示して以来[21]，DVFS を用いた CPU の電力削減手法が数多く研究されてきた．Hsu[7] らは，プログラムをコンパイルする際にコードを解析し，適切な箇所に DVFS を利用するコードを自動挿入するコンパイラを提案している．Koala [16] は，DVFS がパフォーマンスと消費エネルギーに与える変化を正確に予測するために，CPU だけでなく，メモリとメモリバスを考慮に入れたモデルを作成し，電力制御を行う．Chameleon [11] は，アプリケーション主導で電力制御を行うシステムである．アプリケーションは，電力制御に必要なメトリクスを OS から取得し，OS が提供するインターフェースを介して電力制御を行う．これらの手法は，仮想化環境において，

ゲスト OS が電力制御に用いる手法であり，Accele スケジューラはこれらが決定した VCPU 速度を基に VCPU スケジューリングを行う．

これまで VCPU スケジューリングの研究は，主にシステムのパフォーマンスの向上を目的として行われてきた．Time balloning[18] は，マルチプロセッサ上で動作する VM 環境において，ゲスト OS のプロセススケジューラが正しくロードバランスを行えるようにする．Govindan ら[4] は，VM 間の通信遅延を小さくするための VCPU スケジューリング手法を提案している．Lee ら[10] は，VM 環境において，ソフトリアルタイムアプリケーションのレイテンシを小さくする手法を提案している．しかし今まで，消費電力削減のための VCPU スケジューリング手法は提案されていない．

Merkel らは，マルチコア CPU におけるメモリバスの競合を避けるようにアプリケーションをスケジューリングすることで，CPU のエネルギー効率が高くなることを示している[12]．マルチコア CPU において，メモリバウンドなワークロードを同時にスケジューリングすると，メモリバスの競合によるパフォーマンスの低下が起き，DVFS を利用してもエネルギー効率が悪くなることがある．しかし，本研究の実験結果では，メモリバウンドなワークロードを同時に動かすことによるパフォーマンス低下より，DVFS による消費電力削減の効果の方が大きく，エネルギー効率が高くなった．これは，DVFS による消費電力の削減が CPU によって異なるためと考えられる．Merkel らが実験に用いた CPU は最低周波数が 1.6GHz であり，本研究で用いた CPU の最低周波数 0.8 GHz よりも大きいため，DVFS による消費電力の削減が小さい．本研究と Merkel らの研究は補完関係にあり，今後，メモリバスの競合とマルチコア CPU の電力消費特性の双方を考慮するスケジューラが必要であると考える．

7. まとめと今後の課題

本論文では，VM 環境において，マルチコア CPU の電力消費特性を考慮した VCPU スケジューラである Accele スケジューラを提案した．マルチコア CPU には，全てのコアの周波数を下げないと，DVFS による消費電力の削減が小さいという特性がある．Accele スケジューラはこの特性を考慮し，全てのコアの周波数が下がる確率が高くなるように VCPU をスケジューリングする．具体的には，各ランキューにおいて，一定周期で VCPU を周波数順にソートしてからスケジューリングを行う．SysBench と SPEC CPU2006 を用いた実験の結果から，Accele スケジューラが大きく消費エネルギーを削減できることが示された．ただし同時に，メモリアクセスの競合によってワークロードのパフォーマンスが低下することがわかった．

今後の課題として，マルチコア CPU の電力消費特性とメモリアクセスの競合の双方を考慮するスケジューリング手法を考えることが挙げられる．

参 考 文 献

1) Advanced Micro Devices, inc. *AMD PowerNow!^{TM} Technology*, 2000. http://www.amd.com/epd/processors/6.32bitproc/8.amdk6fami/x24404/24404a.pdf.
2) Michael Armbrust, Armando Fox, Rean Griffith, AnthonyD. Joseph, RandyH. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.
3) P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A Ho, R. Neugebauer, I. Pratt, and A.Warfield. Xen and the Art of Virtualization. In *Proc. of the 19th ACM SIGOPS symposium on Operating Systems Principles (SOSP'03)*, pages 299–310, 2003.
4) Sriram Govindan, Arjun R. Nath, Amitayu Das, Bhuvan Urgaonkar, and Anand Sivasubramaniam. Xen and Co.: communication-aware CPU scheduling consolidated xen-based hosting platforms. In *Proc. of the 3rd International ACM SIGPLAN/SIGOPS Conference on Virtual Execution Environments (VEE '07)*, pages 126–136, June 2007.
5) M.D. Hill and M.R. Marty. Amdahl's Law in the Multicore Era. *IEEE Computer*, 41(7):33–38, 2008.
6) HP. *Control power and cooling for data center efficiency*, 2006.
7) Chung-Hsing Hsu and Ulrich Kremer. The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction. In *Proc. of the ACM SIGPLAN 2003 conference on Programming language design and implementation (PLDI'03)*, pages 38–48, 2003.
8) INTEL CORPORATION, http://www.intel.com/design/processor/manuals/253668.pdf. *Intel®64 and IA-32 Architectures Software Developer's Manual*, November 2007.
9) Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, and Arka Bhattacharya. Virtual Machine Power Metering and Provisioning. In *Proc. of the 1st ACM symposium on Cloud Computing (SOCC'10)*, 2010.
10) Min Lee, A.S. Krishnakumar, P. Krishnan, Navjot Singh, and Shalini Yajnik. Supporting soft real-time tasks in the xen hypervisor. In *Proc. of the 6th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments (VEE'10)*, pages 97–108, 2010.
11) Xiaotao Liu, Prashant Shenoy, and Mark Corner. Chameleon: application level power management with performance isolation. In *Proc. of the 13th annual ACM international conference on Multimedia (MULTIMEDIA '05)*, pages 839–848, 2005.
12) Andreas Merkel, Jan Stoess, and Frank Bellosa. Resource-consious Scheduling for Energy Efficiency on Multicore Processors. In *Proc. of the 5th ACM European conference on Com-*

puter systems (EuroSys'10), pages 289–302, 2010.

13) Ripal Nathuji and Karsten Schwan. VirtualPower: coordinated power management in virtualized enterprise systems. In *Proc. of of 21st ACM SIGOPS symposium on Operationg systems principles (SOSP'07)*, pages 265–278, 2007.
14) Ripal Nathuji and Karsten Schwan. Vpm tokens: virtual machine-aware power budgeting in datacenters. In *Proc. of the 17th international symposium on High performance distributed computing (HPDC'08)*, pages 119–128, 2008.
15) Y.Panov and M.Jovanovic. Design Considerations for 12-V/1.5- V, 50-A Voltage Regulator Modules. *IEEE Transactions on Power Electronics*, 16(6), 2001.
16) DavidC. Snowdon, Etienne LeSueur, StefanM. Petters, and Gernot Heiser. Koala: a platform for OS-level power management. In *Proc. of the 4th ACM European conference on Computer systems (EuroSys'09)*, pages 289–302, 2009.
17) Jan Stoess, Christian Lang, and Frank Bellosa. Energy management for hypervisor-based virtual machines. In *Proc. of the 2007 USENIX Annual Technical Conference (ATC'07)*, pages 1–14, 2007.
18) Volkmar Uhling, JoshuaLe Vasseur, Espen Skoglund, and Uwe Dannowski. Towards Scalable Multiprocessor Virtual Machines. In *Proc. of the 3rd Virtual Machine Research And Technology Symposium (VM '04)*, pages 43–56, 2004.
19) Akshat Verma, Puneet Ahuja, and Anindya Neogi. pMapper: power and migration cost aware application placement in virtualized systems. In *Proc. of the 9th ACM/IFIP/USENIX International Conference on Middleware (Middleware'08)*, pages 243–264, 2008.
20) Akshat Verma, Gargi Dasgupta, TapanKumar Nayak, Pradipta De, and Ravi Kothari. Server Workload Analysis for Power Minimization using Consolidation. In *Proc. of 2009 USENIX Annual Technical Conference (ATC'09)*, pages 355–386, 2009.
21) Mark Weiser, Brent Welch, Alan Demers, and Scott Shenker. Scheduling for reduced CPU energy. In *Proc. of the 1st USENIX conference on Operating Systems Design and Implementation (OSDI'94)*, pages 12–23, 1994.
22) W.Wu, N.Lee, and G.Schuellein. Multi-phase buck converter design with two-phase coupled inductors. In *Proc. of IEEE Applied Power Electronics Conference and Exposition*, 2006.