Vol.2013-OS-127 No.8
Vol.2013-EMB-31 No.8
2013/12/3

消費エネルギー予測に基づいたKVM仮想化環境における省電力制御の研究

DOUANGCHAK SITHIXAY[1,a)]　佐藤 未来子[1]　山田 浩史[1]　並木 美太郎[1]

概要：本稿では，仮想化環境における省電力化を目的とし，消費エネルギー予測に基づいた省電力化を行う仮想マシンモニター（VMM）の設計，実装と評価について述べる．従来手法では，OS やアプリケーションを修正する必要があるが，本研究では VMM レイヤにおいて動的電圧・周波数制御（DVFS）時の仮想マシン（VM）ごとの演算性能，消費電力，ネットワーク I/O，ディスク I/O の予測に基づいた省電力制御を行うことで，VM のゲスト OS やアプリケーションに対して透過的に省電力を実現できるのが特徴である．本手法では，VM 実行時のキャッシュミス率やプロセッサ全体のメモリアクセス頻度などのハードウェアパフォーマンスカウンタの情報と仮想 NIC，仮想 I/O の情報を用いて，多変量回帰分析手法により求めた最適な演算性能，消費電力またはスループットをもとに DVFS 制御を実施する．実装には，Linux カーネルを用いた VMM である KVM を用いた．評価より，コア単位で DVFS 制御可能なマルチコア環境において，演算性能とスループットの条件の範囲内で VM が 1 台ある場合，最大でメモリバウンドベンチマークにて 38.3%，ディスクベンチマークにて 35.1%，ネットワークベンチマークにて 46.0%，VM が複数台ある場合，最大で 44.3%のプロセッサの消費エネルギーの削減率を確認した．

1. はじめに

近年の計算機の複雑化と演算性能向上に伴って，システムの消費電力の増加の一途を辿っている．計算機の効果的な利用を通じて環境保護をしつつあり，地球に優しい活動であるグリーンコンピューティングが活発に行われている．グリーンコンピューティングに関して，資源の有効活用や消費エネルギーの削減など様々な手法が提案された．その一つの手法として，仮想化技術が挙げられる．クラウドでは，仮想化技術を利用することにより，運用効率を向上し，必要なハードウェアを減らすことで，コストを削減する．しかし，クラウドの成長に伴い，計算機の台数増加によって，莫大な電力が消費されたため，仮想化環境でも省電力化の要求が高まっている．

計算機システムの中でも特に複雑化したプロセッサはシステム全体の消費電力のうち，大きな割合を占める場合が多いことから，省電力化の研究が盛んに行われている．ハードウェアレベルによる省電力化手法は，プロセス技術の改善やハードウェアが自動的に電源制御を行う．例として，Intel 社の EIST(Enhanced Intel Speedstep Technology)[1]，AMD 社の CoolCore[2] などの，様々な手法が存在する．一方，ソフトウェアレベルによる省電力化手法は，ハードウェアが提供する省電力化機構である動的な電源電圧・周波数制御 DVFS(Dynamic Voltage and Frequency Scaling) がある．これは一般に，プロセッサの動作周波数を下げると電圧も低下させることができ，消費電力を削減するための技術である．当然ながら動作周波数を下げると演算性能またはネットワークのスループットも落ちるため，必要な時にのみ周波数制御を行うことで省電力化が可能である．DVFS を用いたソフトウェアによる省電力化の例としては，Linux システムに搭載された cpufreq[3] モジュールと各種 governor システムを併せたものと Windows の Cool'n'Quiet[4] がある．

従来の計算機単体の省電力化の研究は DVFS 手法を利用して省電力化を行うことが中心であり，OS やアプリケーションを修正して，直接ハードウェアをコントロールした．仮想化環境の場合は，複数の VM が存在し，全ての VM のゲスト OS を修正することが困難である．さらに，一般的に VM やゲスト OS は直接ハードウェアをコントロールすることができない．そのため，従来の省電力手法をそのまま仮想化環境に適用することが難しい．省電力化は仮想化環境という新しい分野に挑戦することになる．

仮想化は，準仮想化と完全仮想化に大別することができる．準仮想化を提供する代表的なものは Xen[5] があり，完

[1] 東京農工大学
Tokyo University of Agriculture and Technology
[a)] joe@namikilab.tuat.ac.jp

全仮想化を提供する代表的なものはKVM[6]がある．本研究は，完全仮想化を提供するKVMを用いて省電力化を目指す．KVMは，Linuxにマージされて Linuxカーネル自体をVMMとする仕組みである．KVMのVMMレイヤで Linuxが提供しているスケジューラ，I/O管理を利用して，VMの演算性能，ディスクI/O，ネットワークI/Oに関する情報を追跡し，VMの特性を予測することが可能である．さらに，Linuxカーネルの進化とともに，新しい機能が提供されている．今後，KVMを含むLinuxをベースとしたクラウドに発展していくことが予想され，KVMの適用はさらに増えていくと考えている．

演算性能を予測し，演算性能をできるだけ落とさずに省電力化を目指す研究としては，M.Weiserらの研究 [9] とW.Wuらの研究 [10] がある．また，実際のLinuxシステムに適用するW.Yuanらの研究 [11] もある．従来の省電力化研究では，可能な限り周波数を落すことを前提としている場合が多い．しかし，DVFSにより周波数を落すと，演算性能またはスループットも低下する．演算性能が低下した場合，実行時間が延びて必ずしも周波数を最低にすれば消費エネルギーを最も削減できるとは限らない．さらに，ディスクI/Oがボトルネックになっている場合，演算性能に影響を与えると考えられる．また，DVFS時にスループットが大幅に低下した場合，スループットあたりの消費エネルギーが得にならない可能性がある．仮想化環境においてDVFS制御を行うC.M.Kamgaら [15] の研究では，XenのVMMレイヤと管理ドメインに省電力制御を実装し，VMのCPU負荷状況を判断して，CPU負荷が少ない時だけCPU周波数を落している．しかし，メモリアクセスが多いプログラムを実行しているVMにおいては，CPU負荷が少ないと判断していしまいCPU周波数を落すこととなるが，演算性能をほとんど落すことはできず，DVFS制御を適切に行えないという問題が残されている．

本研究では，VMの演算性能と消費電力の予測に加えて，VMのネットワークI/Oとディスク I/Oの予測を行うことで，VMのスループットとディスクI/Oの処理待ちに費やした処理時間を考慮して省電力化が可能となる．また，本研究で用いるKVMにより生成されたVMは，ホストOSのLinuxのスケジューラにより管理される．VMのスケジューリングは，ホストOSが提供した機能を利用することができるため，Xenのような独自ポリシーを持つスケジューリングに合わせて省電力制御を設ける必要がない．

予測を行う方法として，定性的に対象システムをモデル化する方法と，定量的にモデル化する方法が存在する．しかし，定性的にシステムを分析する方法では，モデル式を作ることが困難で，多くのプラットホームへの適用が難しい．そこで，本研究では，演算性能，消費電力，ネットワークI/O，ディスクI/O予測の方法として，予測のモデル化を容易に，かつ機械的に行うことのできる，定量的なモデル化手法に注目した．また，VMMレイヤで統一的に省電力制御を適用するため，VMのゲストOSごとに省電力制御を搭載する必要がない．

一般的に，多くのシステムにおいて，稼働時間帯の中に繁忙時間帯が存在する．繁忙時間帯では性能目標を高く設定することで，性能を最大限に引き出すことができる．一方，繁忙時間帯以外では，性能目標を低く設定し，省電力化することができる．本研究は，この性能目標を達成する値を閾値とする．閾値は演算性能閾値とスループット閾値の二種類があり，システム管理者により設定される．

本手法では，まず，ハードウェアから得られる各種情報と仮想NIC，仮想I/Oの情報を元に，定量的にモデル化を行う．演算性能モデル，スループットモデルと消費電力モデルに分けて，演算性能モデルとスループットモデルを元にVM単位でDVFS制御を行い，与えられた演算性能閾値とスループット閾値の両方を上回る演算性能とスループットを保ちつつ，消費電力モデルから消費エネルギーが最小となるようなシステムを設計，実装した．

2. 省電力制御の概要

本研究では，コアをn個持つホモジニアスマルチコアプロセッサ上の仮想化環境を想定する．仮想化環境に省電力制御を適用することで，システム管理者によって与えられた各VMの演算性能閾値とスループット閾値の両方を上回る演算性能とスループットを保ちつつ，各VMの消費エネルギーが最小となる最適な周波数・電圧を求め，VM単位およびコア単位でDVFSを行う．

完全仮想化環境を提供するKVMのVMMレイヤにおいて，個々のVMの消費エネルギーを予測する．消費エネルギー予測に使われるVMの演算性能，消費電力，ネットワークI/O, ディスクI/Oの予測は定量的に多変量回帰分析し，モデル化を行う．定量的な分析手法を採用することで，モデルを機械的に最適化できるようになる．予測モデルを図1に示す．この予測モデルは，CPU，ネットワークそしてディスクに関する情報を用いて，VMの挙動を追跡し消費エネルギーを予測する．

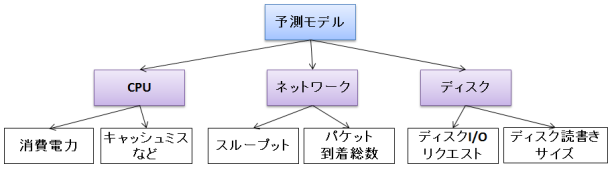

図 1 予測モデル

本研究では，マルチコア上での仮想化環境において複数のVMが存在し，さらに，各VMが複数のVCPUを持つ．VCPUごとに物理CPU(PCPU)がずっと固定されずに，各VCPUは全てのPCPUに対応付けられる．ただし，

$VCPU \leq PCPU$ と仮定する．省電力制御は，予測式をモデル化する学習モードと VM の消費エネルギーが最小化となるような省電力モードからなる．省電力モードで最適な周波数・電圧を求めながら実行していく．そのため，事前に学習モードで初期パラメータを習得する必要がある．

学習モードでは，1 台の VM 上であらかじめ様々なベンチマークを様々な周波数で動作させ，統計情報による VM の演算性能，消費電力，ネットワーク I/O，ディスク I/O の予測を行う．これにはキャッシュミス率やプロセッサ全体のメモリアクセス頻度などのハードウェアカウンタの値と仮想 NIC，仮想 I/O から得られた値を説明変数，全コアにおける最高周波数時との相対比（演算性能比，消費電力比，スループット比）を目的変数として学習を行い，多変量回帰分析を行う．多変量回帰分析による演算性能比，省電力比，スループット比を予測するための回帰方程式の回帰係数を算出し，各種の予測式を最適化する．なお，VM が複数の VCPU を持つと前提しため，ハードウェアカウンタの値の取得は，VM コンテキストスイッチ毎ではなく，VCPU コンテキストスイッチ毎に行う．各 VCPU コンテキストスイッチ毎に取得した値の合計は VM のタイムスライス単位の値として，この値を説明変数とする．また，仮想 NIC と仮想 I/O からの値に対しても VM のタイムスライスごとに取得する．

省電力モードでは，学習モードで求めた回帰方程式の回帰係数を用いて，演算性能比，消費電力比，スループット比の予測式を立てる．このモードで，複数台の VM が動作している状態で，システム管理者が各 VM の演算性能閾値とスループット閾値を設定する．その後，本システムは演算性能閾値とスループット閾値の両方を上回る演算性能とスループットを保ちつつ，各 VM の消費エネルギーが最小となる最適な周波数・電圧を求め，VM 単位およびコア単位で DVFS を行う．最適な周波数・電圧は VMM レイヤで VM のタイムスライス単で求めて，VCPU コンテキストスイッチ毎に DVFS を行う．

VMM レイヤで，VM ごとの演算性能，消費電力，ネットワーク I/O，ディスク I/O の予測を行うことで統一的に電力制御を行い，VM のゲスト OS やアプリケーションに変更を加える必要がなく随時省電力制御が可能となる．また，この予測モデルの最適化により，様々な計算機に対して，必要な計算機演算性能や，その計算機の消費電力特性に応じて，最適な周波数・電圧制御を行うことができる高度な省電力化機構を仮想化環境へ適用できる．

3. 設計

本章では，仮想化環境における省電力制御に用いる予測モデルの最適化とその適用について述べる．先行研究として，金井ら [12] は Linux を対象とし，林ら [13] は L4 マイクロカーネルを対象とし，定量的な予測モデル化を行い，計算機システム上で動作するプロセスを対象にして，省電力化を行った．本研究では，プロセス単位ではなく VM に適用し，VM 単位に異なる演算性能とスループット条件を設定することができて，より細かい粒度で仮想化環境における省電力制御を提案する．

筆者ら [14] は，パフォーマンスカウンタのみで演算性能，消費電力の予測に基づいた仮想化環境における省電力制御を研究してきた．しかし，VM の挙動によりディスク I/O がボトルネックになった場合の予測ミスや DVFS 時にスループットが大幅に低下した問題があった．本研究は，演算性能，消費電力の予測に加えて，ネットワーク I/O とディスク I/O の予測を行うため，ディスク I/O やスループット考慮した省電力制御を可能とする．

3.1 仮想化環境での省電力制御の設計

本研究では，仮想化環境と物理ハードウェアの中間にある VMM レイヤに省電力制御を設ける．まず，予測モデルの予測式を立てるために，仮想化環境に関する情報と物理ハードウェアに関する情報が必要となる．具体的には，仮想化環境では，各 VM の仮想 NIC，仮想 I/O に関する情報，コンテキストスイッチ情報，演算性能が必要であり，物理ハードウェアでは，プロセッサから得られるパフォーマンスカウンタの値と DVFS 情報が必要である．VM 上で，あらかじめ様々な特性を持つベンチマークを動作させ，学習機構で演算性能，消費電力，ネットワーク I/O，ディスク I/O に関する学習をさせて予測式の回帰係数を定め，予測モデルを立てる．その予測モデルから，周波数決定機構で消費エネルギーが最小となるような周波数を決定する．

省電力制御の全体構成を図 2 に示す．本研究では，完全仮想化環境を提供する KVM の VMM レイヤに本手法を適用する．本システムは，リストを用いて，VM ごとに備える省電力制御に関する情報を管理する．KVM により，VM が起動される場合，リストに VM の情報を追加して，VM がシャットダウンされた場合，リストからその VM の情報を削除する．仮想化環境における省電力制御には，VM の演算性能，消費電力，ネットワーク I/O，ディスク I/O の予測式をモデル化する学習モードと消費エネルギーが最小化となるような省電力モードからなる．以下，この二つのモードについて述べる．

(1) 学習モード

学習モードでは，省電力モードで用いる予測式の学習を行う．VM 単位で多変量回帰分析の回帰係数を算出するために必要な統計を取り，予測式をモデル化する．学習モードは，

- VM 管理機構
- パフォーマンスカウンタ管理機構
- 学習機構
- DVFS 制御機構

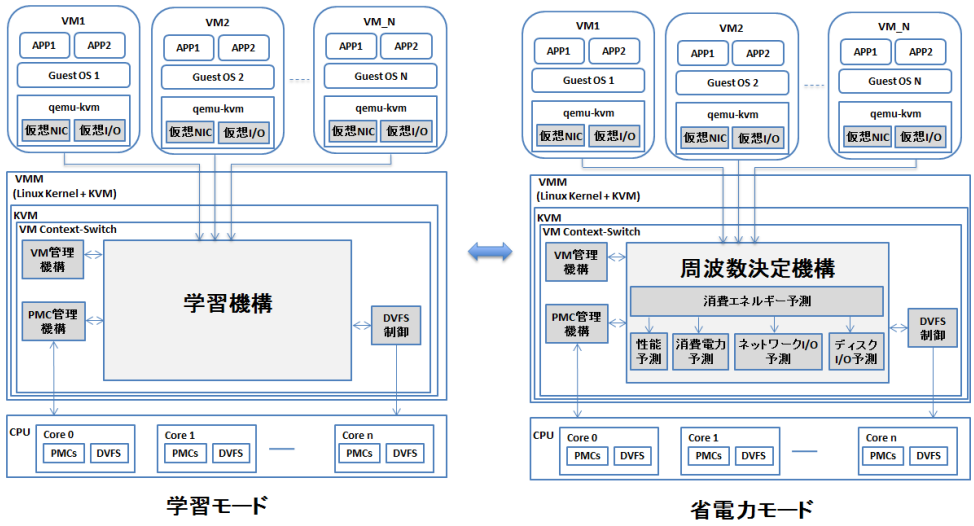

図 2 省電力制御の全体構成

からなる．VM 管理機構は，稼働中の VM の情報を管理する．VM はホスト OS から見れば，単一のユーザプロセスである．VM 管理機構は，このプロセス ID を利用し VM に関する統計情報を管理を行う．パフォーマンスカウンタ管理機構は，VM 単位で多変量回帰分析に用いるパフォーマンスカウンタと仮想 NIC, 仮想 I/O の集計を行う．そのために，パフォーマンスカウンタ管理機構は，VM 管理機構から得られた VM の情報と VMM から VM のディスパッチタイミングで VM ごとのパフォーマンスカウンタを読み取る．学習機構は，DVFS 制御機構と連携して，動作周波数ごとの学習を行う．また，電力測定器と連携し，測定された計算機のプロセッサ全体の消費エネルギーと学習機構から得られた統計情報を多変量回帰分析を行い，最適な回帰係数を求め，予測式を立てる．

(2) 省電力モード

省電力モードでは，学習モードでモデル化した演算性能，消費電力，ネットワーク I/O，ディスク I/O の予測式を用いて，システム管理者が与えた演算性能，スループット制約の中で実際に消費エネルギーが最小となるように省電力化を行う．省電力モードは，

- VM 管理機構
- パフォーマンスカウンタ管理機構
- 周波数決定機構
- DVFS 制御機構

からなる．周波数決定機構は，コンテキストスイッチ毎にパフォーマンスカウンタ管理機構から得られた統計情報，仮想 NIC，仮想 I/O から得られた情報と各種の予測式から VM の演算性能，消費電力，スループットの予測を行う．システム管理者により設定された演算性能，スループット制約の範囲で，消費エネルギー予測を行い，消費エネルギーが最小なるような周波数を算出する．そして，VM の VCPU が動作するコアの周波数を変更する．

3.2 統計情報による VM の演算性能，消費電力，ネットワーク I/O，ディスク I/O の予測

各 VM の演算性能，消費電力，ネットワーク I/O，ディスク I/O の予測を行うために，図 1 で示した CPU，ネットワーク，ディスクに関する統計情報を用いる．VM の演算性能予測は，演算性能に影響を与える全コア共有 L3 キャッシュミス回数を用いて，VM の消費エネルギー予測は，消費電力に影響を与えるメモリアクセス，各コア L2 キャッシュミス回数，全コア共有 L3 キャッシュミス回数を用いる．また，VM のネットワーク I/O とディスク I/O の予測は，仮想 NIC と仮想 I/O からスループット，パケット到着総数，ディスク I/O リクエスト，ディスクの読書きサイズを用いる．

VM の消費電力は，VM の特性により，つまりゲスト OS 上で動作するアプリケーションにより異なる．今回の実装対象に選択した Phenom プロセッサをはじめ，多くのシステムではプロセッサとメモリのクロックが非同期で動作することが多い．このような環境では，DVFS によるプロセッサの周波数・電圧変更での前後で，メインメモリへアクセス頻度の異なる VM では消費電力比が異なると予想される．例として，メモリへアクセス頻度の高い VM は，元々プロセッサ内の ALU/FPU など各種演算機能，キャッシュアクセスに要する動作時間が少なく，プロセッサの DVFS 制御を行い，動作周波数・電圧を下げても，その分の動作時間の消費電力しか減らない．一方，メモリへアクセス頻度の低い VM は，消費電力が大幅に削減できると予想される．

ターゲット周波数・電圧毎に VM 単位で各種の予測を行い，システム管理者により与えられた個々の VM の演算性能閾値とスループット閾値を下回らず消費エネルギーが最小となる周波数・電圧で動作させることで，省電力化を行う．本方式は，事前に学習したプロファイリングデータに

基づいて分析するため，キャッシュの構成などを意識することなく，様々なプラットフォームで容易に予測式を立てることができる．

本省電力制御は，3.1 節で示した様に，予測式をモデル化する学習モードと VM の消費エネルギーが最小化となるような省電力モードからなる．省電力モードで最適な周波数・電圧を求めながら実行していく．そのため，事前に学習モードで初期パラメータを習得しておく．以下，各モードにおける予測式のモデル化を述べる．

(1) 学習モード

学習モードでは，1 台の VM 上であらかじめ様々なベンチマークを様々な周波数 f_c で動作させ，学習を行い，演算性能，消費電力，スループットの予測式をモデル化する．全コアを最高周波数時の実行時間，消費電力，スループット（つまり演算性能比 Y_{fc}^{perf}，消費電力比 Y_{fc}^{pow}，スループット比 Y_{fc}^{thrp}）を目的変数とする．説明変数は，性能演算比を求める時に，命令あたりの L2 キャッシュミス回数 $X_{fc,l2}^{perf}$，命令あたりの L3 キャッシュミス回数 $X_{fc,l3}^{perf}$，命令あたりのメモリアクセス回数 $X_{fc,mem}^{perf}$，ネットワークのパケット到着総数 $X_{fc,packet}^{perf}$，ディスクの読込み回数 $X_{fc,read}^{perf}$，ディスクの読込みサイズ $X_{fc,read\_size}^{perf}$，ディスクの書込み回数 $X_{fc,write}^{perf}$，ディスクの書込みサイズ $X_{fc,write\_size}^{perf}$ とおき，$a_{fc,0}^{perf}$〜$a_{fc,8}^{perf}$ を回帰係数とおく．消費電力比とスループット比も同様に考え，演算性能比，消費電力比，スループット比の予測式は，

$$\begin{aligned} Y_{fc}^{perf} &= a_{fc,0}^{perf} + a_{fc,1}^{perf} X_{fc,l2} + a_{fc,2}^{perf} X_{fc,l3} \\ &+ a_{fc,3}^{perf} X_{fc,mem} + a_{fc,4}^{perf} X_{fc,packet} \\ &+ a_{fc,5}^{perf} X_{fc,read} + a_{fc,6}^{perf} X_{fc,read\_size} \\ &+ a_{fc,7}^{perf} X_{fc,write} + a_{fc,8}^{perf} X_{fc,write\_size} \end{aligned} \quad (1)$$

$$\begin{aligned} Y_{fc}^{pow} &= a_{fc,0}^{pow} + a_{fc,1}^{pow} X_{fc,l2} + a_{fc,2}^{pow} X_{fc,l3} \\ &+ a_{fc,3}^{pow} X_{fc,mem} + a_{fc,4}^{pow} X_{fc,packet} \\ &+ a_{fc,5}^{pow} X_{fc,read} + a_{fc,6}^{pow} X_{fc,read\_size} \\ &+ a_{fc,7}^{pow} X_{fc,write} + a_{fc,8}^{pow} X_{fc,write\_size} \end{aligned} \quad (2)$$

$$\begin{aligned} Y_{fc}^{thrp} &= a_{fc,0}^{thrp} + a_{fc,1}^{thrp} X_{fc,l2} + a_{fc,2}^{thrp} X_{fc,l3} \\ &+ a_{fc,3}^{thrp} X_{fc,mem} + a_{fc,4}^{thrp} X_{fc,packet} \\ &+ a_{fc,5}^{thrp} X_{fc,read} + a_{fc,6}^{thrp} X_{fc,read\_size} \\ &+ a_{fc,7}^{thrp} X_{fc,write} + a_{fc,8}^{thrp} X_{fc,write\_size} \end{aligned} \quad (3)$$

と表すことができる．学習モードで，目的変数 Y と説明変数 X の値を測定し，多変量回帰分析により，回帰係数の a を求める．Y と X は VM 単位で測定するため，パフォーマンスカウンタからの統計情報とディスク I/O とネットワーク I/O の統計情報を VM のタイムスライス単位で測定する．学習モード開始時に，Y と X を測定し始めて，VM の統計情報として格納する．学習モード終了時に，多変量回帰分析により，回帰係数の a を求める．

(2) 省電力モード

省電力モードでは，学習モードで求めた回帰方程式の回帰係数を用いて，演算性能比，消費電力比，スループット比の予測式を立てる．このモードで，複数台の VM がある状態において任意 VM_i 上でベンチマークが動作される時に，周波数 f_c の任意 VM_i の演算性能比 $Y_{vm_i,fc}^{perf}$，消費電力比 $Y_{vm_i,fc}^{pow}$，スループット比 $Y_{vm_i,fc}^{thrp}$ の予測式は，

$$\begin{aligned} Y_{vm_i,fc}^{perf} &= a_{fc,0}^{perf} + a_{fc,1}^{perf} X_{vm_i,fc,l2} + a_{fc,2}^{perf} X_{vm_i,fc,l3} \\ &+ a_{fc,3}^{perf} X_{vm_i,fc,mem} + a_{fc,4}^{perf} X_{vm_i,fc,packet} \\ &+ a_{fc,5}^{perf} X_{vm_i,fc,read} + a_{fc,6}^{perf} X_{vm_i,fc,read\_size} \\ &+ a_{fc,7}^{perf} X_{vm_i,fc,write} + a_{fc,8}^{perf} X_{vm_i,fc,write\_size} \end{aligned} \quad (4)$$

$$\begin{aligned} Y_{vm_i,fc}^{pow} &= a_{fc,0}^{pow} + a_{fc,1}^{pow} X_{vm_i,fc,l2} + a_{fc,2}^{pow} X_{vm_i,fc,l3} \\ &+ a_{fc,3}^{pow} X_{vm_i,fc,mem} + a_{fc,4}^{pow} X_{vm_i,fc,packet} \\ &+ a_{fc,5}^{pow} X_{vm_i,fc,read} + a_{fc,6}^{pow} X_{vm_i,fc,read\_size} \\ &+ a_{fc,7}^{pow} X_{vm_i,fc,write} + a_{fc,8}^{pow} X_{vm_i,fc,write\_size} \end{aligned} \quad (5)$$

$$\begin{aligned} Y_{vm_i,fc}^{thrp} &= a_{fc,0}^{thrp} + a_{fc,1}^{thrp} X_{vm_i,fc,l2} + a_{fc,2}^{thrp} X_{vm_i,fc,l3} \\ &+ a_{fc,3}^{thrp} X_{vm_i,fc,mem} + a_{fc,4}^{thrp} X_{vm_i,fc,packet} \\ &+ a_{fc,5}^{thrp} X_{vm_i,fc,read} + a_{fc,6}^{thrp} X_{vm_i,fc,read\_size} \\ &+ a_{fc,7}^{thrp} X_{vm_i,fc,write} + a_{fc,8}^{thrp} X_{vm_i,fc,write\_size} \end{aligned} \quad (6)$$

と表すことができる．省電力モードは，学習モードで求めた回帰係数の a と実際に測定した説明変数 X の値を用いて，式 (4)〜式 (6) により，任意 VM_i の演算性能比 $Y_{vm_i,fc}^{perf}$，消費電力比 $Y_{vm_i,fc}^{pow}$，スループット比 $Y_{vm_i,fc}^{thrp}$ を求めることができる．X の値は学習モードと同様に，パフォーマンスカウンタからの統計情報とディスク I/O とネットワーク I/O の統計情報を VM のタイムスライス単位で測定する．

3.3 周波数決定方式

周波数決定に用いる閾値はシステム管理者により設定する．VM の閾値は，演算性能閾値とスループット閾値の二種類があり，VM ごとに指定することができる．VM の閾値を設定することにより，各 VM の演算性能とスループット閾値の両方を上回る演算性能を保ちつつ，消費エネルギーを最小化することができる．最適な周波数・電圧は VMM レイヤで VM のタイムスライス単で求めて，VCPU コンテキストスイッチ毎に DVFS を行う．

VM ごとに設定された閾値を用いて周波数決定機構は，以下のアルゴリズムを用いて，最適な周波数を決定する．

VM の演算性能閾値とは，最高周波数で動作した場合に，

その VM 上である処理を行うのに要する時間に対する時間の比として定義する．例として，任意 VM_i 上である処理が，最高周波数かつ，他のコアで何も実行されていない際に 1 分で終了する処理について，他のコアの状況にかかわらず，2 分で終了すればいい場合には，任意 VM_i の演算性能閾値として 50%を指定する．与えられた任意 VM_i の演算性能閾値 $Thred_{perf,i}$ に対して，$Thred_{perf,i}$ のみを上回るコア c の周波数の集合 A は，

$$A = \{x \mid (Thred_{perf,vm_i} \leq Y^{perf}_{vm_i,x})\} \quad (7)$$

と表すことができる．また，スループット閾値 $Thred_{thrp,i}$ に対して，$Thred_{thrp,i}$ のみを上回るコア c の周波数の集合 B は，

$$B = \{x \mid (Thred_{thrp,vm_i} \leq Y^{thrp}_{vm_i,x})\} \quad (8)$$

と表すことができる．演算性能閾値 $Thred_{perf,i}$ とスループット閾値 $Thred_{thrp,i}$ の両方を上回る周波数集合は $A \cap B$ となる．その後，周波数集合 $A \cap B$ の中から，最も消費電力比が小さい周波数を選択する．

$$f_{opt} = min\{x \mid (Y^{power}_{vm_i,x}, x \in A \cap B)\} \quad (9)$$

f_{opt} は任意 VM_i の最適な周波数となり，任意 VM_i の全ての VCPU が割り当てる物理コアの周波数および電圧を変更する．従って，任意 VM_i の演算性能とスループット閾値の両方を上回る演算性能を保ちつつ，消費エネルギーを最小化することができる．

3.4 周波数決定機構

本システムは周波数決定機構を通して，システム管理者により設定された各 VM の演算性能閾値とスループット閾値の両方を上回る性能とスループットを保ちつつ，消費エネルギーが最小化となるように，VM のタイムスライス単位・コア単位で周波数選択を行う．本節では，この設計について述べる．

周波数決定機構では，コンテキストスイッチ毎に 3.2 節で述べた方法により，VM の演算性能，消費電力，ネットワーク I/O，ディスク I/O の予測を行う．その後，3.3 節で述べた方法で予想される消費エネルギーが最小となる周波数を決定する．演算性能，消費電力の予測にはパフォーマンスカウンタの統計情報を用いるが，VM ごとに統計データの傾向は異なり，演算性能や消費電力の特性や最適な周波数は異なる．そのため，本機構では VM 単位パフォーマンスカウンタ管理機構を利用して，VM ごとに VM のタイムスライス単位で統計情報の保存する．また，ネットワーク I/O，ディスク I/O の統計情報に対しても VM のタイムスライス単位で取得して保存する．

本研究では，仮想化環境において複数の VM が存在し，さらに，各 VM が複数の VCPU を持ち，各 VCPU は全ての PCPU に対応付けられることを想定する．そのため，周波数決定機構は VM コンテキストスイッチ毎ではなく，VCPU コンテキストスイッチ毎にパフォーマンスカウンタの取得を行う．そして，VM のタイムスライス単位で周波数を決定する．VCPU コンテキストスイッチ時，つまり，KVM が VCPU を PCPU 上にスケジューリングされた時に，パフォーマンスカウンタの取得を開始する．VCPU が PCPU から解放された時，パフォーマンスカウンタの取得を終了し，統計情報として，VM 管理機構にあるリストに保存する．取得する情報は，各 VCPU の統計情報となり，各 VCPU の統計情報の合計は VM のタイムスライス単位の統計情報となる (図 3)．取得された VM の統計情報を説明変数として，式 (4) と式 (6) に代入して，目的変数である演算性能比とスループット比の算出する．次に，式 (7) と式 (8) により，演算性能閾値，スループット閾値を上回る周波数集合を求める．その後，式 (5) と式 (9) により，最も消費電力比が小さい周波数を選択し，VM の最適な周波数となる．最後に，VM の全ての VCPU が割り当てる物理コアの周波数および電圧を変更する．

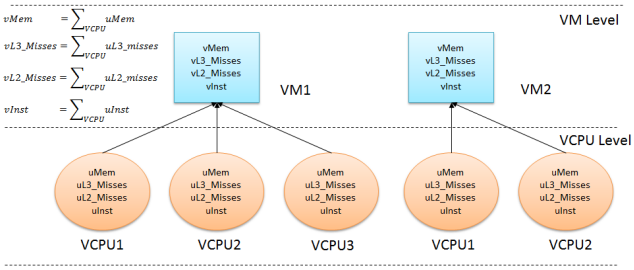

図 3 VCPU スケジューリング時のパフォーマンスカウンタ取得

4. 実装

3 章で述べた設計に基づき，カーネルモジュールである kvm-kmod-3.4 と I/O などをエミューレートする qemu-kvm-1.0 に機能追加を行った．KVM は，Linux カーネル自体を VMM とする仕組みであり，Intel VT-x[7] や AMD-V[8] といった CPU の仮想化支援機能を活用し，完全仮想化による仮想化環境を提供する．KVM は Linux カーネルに統合されることから，注目を集める．ホスト側の Linux のスケジューラ，メモリ管理などをそのまま利用するため，KVM 自体のスケールは小さい．また，KVM は独自のポリシーを持たないため，省電力化の手法だけに集中することができる．本研究では，ターゲットとなるアーキテクチャは x86 とし，同じ x86 の中でも特に電源電圧・周波数制御は CPU やチップセット依存となるため，AMD の Phenom9500 をターゲットとした．

既存の KVM カーネルモジュールと qemu-kvm に対するコード変更箇所は以下のとおりである．

- kvm のメインクラスの関数

kvm_init()：KVM を読み込む時
kvm_exit()：KVM をアンロードした時
kvm_create_vm()：VM を起動した時
kvm_destroy_vm()：VM をシャットダウンした時
kvm_sched_in()：コンテキストスイッチ開始
kvm_sched_out()：コンテキストスイッチ終了
kvm_dev_ioctl()：キャラクタデバイス経由のデータやり取り

- qemu-kvm のクラスの関数
  qemu_deliver_packet(): パケット到着とスループットの取得
  bdrv_aio_readv(): ディスク読込み情報の取得
  bdrv_aio_writev(): ディスク書込み情報の取得

上記において，既存の KVM カーネルモジュールと qemu-kvm に 77 行を変更し，新規部分は合計 1396 行となった．新規追加部分について以下に述べる．

DVFS 制御機構では，CPU のモデル固有レジスタ (MSR, Model Specific Register) の読み書きを行った．実際には，RDMSR と WRMSR という命令で MSR にアクセスし，CPU の周波数・電圧を変更した．パフォーマンスカウンタ管理機構では，DVFS 制御機構と同様に，MSR にアクセスしパフォーマンスカウンタの値を取得する．VM 管理機構では，KVM のメインクラスと連携し，Linux カーネルに実装されるリンクリストを用いて，VM の情報を管理する．学習機構では，演算性能の学習時に，VM に関する演算性能を計算する必要がある．VM の演算性能指標の単位として IPS(Instruction Per Second) を導入した．VM の演算性能は，VM タイムスライス単位で IPS を計測した．

今回は Phenom9500 を実装のターゲットにしているが，近年多くの CPU においてパフォーマンスカウンタを設計し，本手法を利用することで他のアーキテクチャでも適用できる．

5. 評価

本手法について，SPEC2006，ネットワーク，ディスクのベンチマークを用いて，消費エネルギーの評価を行った．本章ではこれらの詳細について述べる．評価には，Phenom9500 を搭載した PC を評価マシンとして利用した．評価マシンの環境を表 1 に示す．

5.1 VM1 台の評価

仮想化環境で VM を 1 台起動し，その上で様々な特性を持つベンチマークを動作し，評価を行った．CPU またはメモリバウンドであるベンチマークとして SPEC2006 から 444.namd，450.soplex，456.hmmer，462.libquantum，470.lbm，ディスクベンチマークとして bonniee++，ネットワークベンチマークとして httperf を評価ベンチマークとした．閾値（=演算性能閾値=スループット閾値）は 70%と

表 1 評価マシン環境

CPU	AMD Phenom 9500(4 Cores)
メモリ	4GB
DVFS	Step0(2.2Ghz,1.200V), Step1(2.0Ghz,1.150V) Step2(1.8Ghz,1.125V), Step3(1.6Ghz,1.100V) Step4(1.4Ghz,1.050V)
仮想化	kvm-kmod-3.4 , qemu-kvm-1.0
各 VM の VCPU	2
各 VM のメモリ	512 MB

40%に設定した．

全ての評価に対して，VM の閾値（演算性能閾値とスループット閾値）を上回る性能で終了させることができた．VM の閾値が 40%の時に最大でメモリベンチマークが 38.3%，ディスクベンチマークが 35.1%，ネットワークベンチマークが 46.0%のプロセッサの消費エネルギーの削減を確認した．

SPEC2006 とディスクベンチマークの評価結果を図 4 に示し，ネットワークベンチマークの評価結果を図 5 に示す．

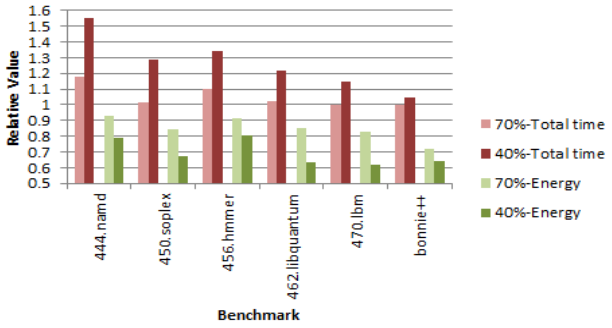

図 4 SPEC2006 とディスクベンチマークの評価

図 4 において，横軸は SPEC2006 とディスクベンチマーク，縦軸は各ベンチマークにおける処理時間および消費エネルギーの値を，全コアで最高周波数に設定した場合を 1 とした時の相対値を表している．ただし，VM 上でこられのベンチマークが動作された時に，外部と通信を行われていないため，スループットに関する評価を省略する．SPEC2006 の中で消費エネルギーの削減は最大でメモリバウンド 470.lbm が 38.3%となった．VM 上でメモリバウンドベンチマークが実行される場合，処理中でのキャッシュミスが頻繁に起きたとわかった．CPU のキャッシュに必要なデータが存在しないため，メインメモリからデータを取得してく必要がある．その時，本システムは，CPU の周波数を下げても演算性能がほとんど低下しないと判断し，低い閾値で消費エネルギーの削減率が高かった．また，メモリバウンドベンチマークの処理時間を見ると，増加率は低かった．一方，CPU バウンド 444.namd は消費エネルギーの削減が低かった．VM 上で CPU バウンドベンチマークが実行される場合，キャッシュヒット率が高いとわかった．

ほとんどの処理時間でキャッシュアクセスを行い，周波数を下げることができないため，消費エネルギーの削減率が低かった．また，CPU バウンドベンチマークの処理時間を見ると，増加率は高かった．本システムは，できる限り消費エネルギーを削減するため，演算性能制約の上限まで処理時間を延ばそうとすることがわかった．

筆者ら [14] の先行研究の評価結果で，パフォーマンスカウンタのみで演算性能予測が外れた 456.hmmer に対しても，消費エネルギーが削減された．456.hmmer は処理中でディスク I/O が頻繁に発生したため，本方式でディスク I/O の予測を行うことでディスクに関する情報を追跡でき，消費エネルギーが削減されたと考える．さらに，ディスクベンチマークの bonnie++に対しても，ディスク I/O 予測の効果により消費エネルギーの削減が 35.1%となった．

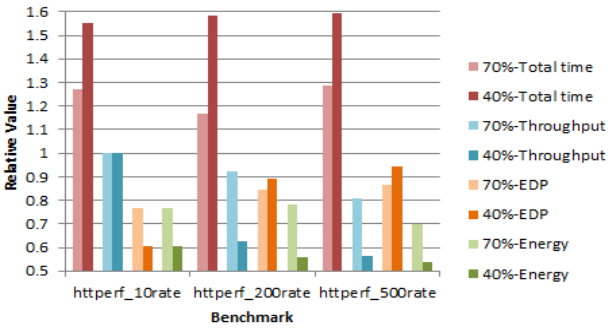

図 5 ネットワークベンチマークの評価

ネットワークベンチマークとして httperf を用いて評価を行った．評価方法として，1 台の VM1 を Web サーバとして起動しており，別のクライアントから VM1 にネットワークベンチマークを用いて負荷をかけた．httperf を使うことにより，VM1 にアクセスして index.html をリクエストした．また，httperf は 1 秒あたりのリクエスト数（以降，rate と呼ぶ）を変更するとこにより，VM1 に負荷を加減することができた．図 5 において，横軸はベンチマーク，縦軸は rate を変えて実行した httperf における処理時間，スループット，EDP 値および消費エネルギーを，全コアで最高周波数に設定した場合を 1 とした時の相対値を表している．

EDP(Energy Dealy Product) 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．VM の閾値=40%の httperf_500rate で，最大 46.0%の消費エネルギーの削減を確認した．消費エネルギーの削減率は閾値が低いほど効果が高いとわかった．

VM にかかる負荷が小さい (httperf_10rate) 場合，CPU の周波数を下げてもスループットがほとんど低下しないため，消費エネルギーと EDP は同じぐらいの値に得になった．VM にかかる負荷が大きい (httperf_200rate, httperf_500rate) 場合，CPU の周波数を下げると，スループットが低下した．ただし，本システムはネットワーク I/O の予測を行うことで，スループットを考慮しながら省電力化をするため，スループットの低下を閾値の条件の範囲内で抑えて，EDP 値が常に得になるように最適な周波数・電圧を選択することで，より最適な省電力制御を実施できる方式となっている．

5.2 VM 複数台の評価

VM は三台があり，三台に異なるベンチマークを表 2 の組合せで実行する．評価結果を図 6 と図 7 に示す．

表 2 ベンチマークの組合せ

scenario	ベンチマーク		
	VM1	VM2	VM3
1	444.namd	444.namd	-
2	444.namd	470.lbm	-
3	470.lbm	470.lbm	-
4	444.namd	httperf_500rate	bonnie++
5	httperf_200rate	bonnie++	470.lbm

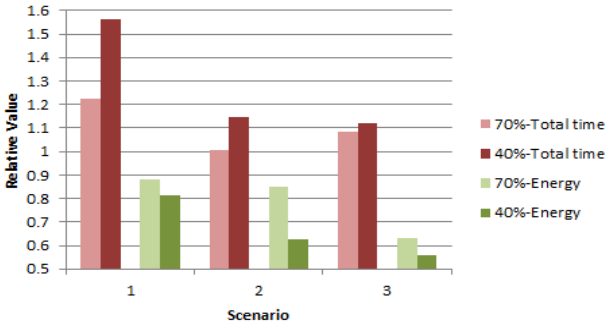

図 6 VM 複数台の評価 1

図 6 は SPEC2006 のベンチマークのみの組合せ (scenario1,2,3) の評価結果である．横軸は scenario1,2,3，縦軸は scenario1,2,3 における処理時間および消費エネルギーの値を，全コアで最高周波数に設定した場合を 1 とした時の相対値を表している．ただし，VM 上でこられのベンチマークが動作された時に，外部と通信を行われていないため，スループットに関する評価を省略する．VM の閾値=40%で，メモリバウンドとメモリバウンドの組合せである scenario3 が最大で 44.3%であり，CPU バウンドと CPU バウンドの組合せである scenario1 が最小で 18.7%の消費エネルギーを削減した．VM 複数台の評価に対しても，本手法が有効であると確認した．

図 7 は SPEC2006，ネットワークベンチマークとディスクベンチマークの組合せ (scenario4,5) の評価結果である．横軸は scenario4,5，縦軸は scenario4,5 における処理時間，スループット，EDP 値および消費エネルギーを，全コアで

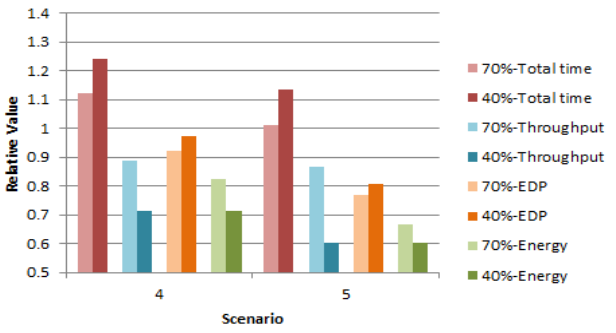

図 7 VM 複数台の評価 2

最高周波数に設定した場合を 1 とした時の相対値を表している．VM の閾値=40%で，scenario5 が最大で 39.7%の消費エネルギーを削減した．本手法は，VM のタイムスライス単で周波数を決定するため，VM が複数台あっても VM1 台の時と同様に各 VM の消費エネルギー予測ができた．

6. 関連研究

仮想化環境における省電力化は，VM の CPU 負荷状況から DVFS 制御を行う研究として，C.M.Kamga ら [15] の研究が存在する．Xen の credit スケジューラに VM の CPU 負荷状況の監視機構と CPU 時間の制御機構を追加している．VM の CPU 負荷状況を判断して，CPU 負荷が少ない時だけ CPU 周波数を落し，既存なプラットフォームの変更が少ないという利点がある．しかし，メモリアクセスが多いプログラムを実行している VM においては，CPU 負荷が少ないと判断していしまい CPU 周波数を落すこととなるが，演算性能をほとんど落すことはできず，DVFS 制御を適切に行えないという問題が残されている．S.Kundu ら [16] は CPU やメモリの使用状況などに基づく VM 上で動作するアプリケーションの演算性能のモデル化を提案している．アプリケーションが動作する間に，物理 CPU やメモリから得た情報を用いて，VM の演算性能を予測している．また，Koala[17] は，CPU だけでなく，メモリとメモリパスを考慮し，演算性能と消費エネルギーを予測し，DVFS による電力制御を提案している．これらに対して，本研究は CPU から得られた情報情報に加えて，仮想 NIC，仮想 I/O から得られた統計情報を用いて VM の消費エネルギー予測するため，ディスク I/O がボトルネックになった時の予測ミスやスループットの低下を考慮して電力制御を行う．VM の特性を判断するために，パフォーマンスカウンタを用いた他の研究も存在する．R.Bertran ら [18] は，パフォーマンスカウンタを用いて回帰分析で VM の消費電力を予測している．ただし，本手法と異なって自動的にコンテキストスイッチ単位ではなく，秒単位でパフォーマンスカウンタを取得するため，学習時に手間がかかっている．Chameleon[19] は，主導となったアプリケーションで電力制御を手法を提案している．アプリケーションは OS が提案するインターフェースを介して電力制御を行う．Stoess ら [20] は，各 VM に消費エネルギーをアカウンティングする手法を提案している．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．この手法は VMM と VM のゲスト OS に修正を加えて行われている．これに対して，本手法は個々のゲスト OS に修正を加えることなく，VMM で電力制御を行っている．G.Dhiman ら [21] は，混合正規分析モデルで VM の消費電力を予測している．しかし，この研究は予測の過程に終わり，実際のシステムに適用して省電力化を行わなかった．また，クライアントーサーバ型の仮想化環境における省電力制御の研究が存在する [22]．サーバが中心となり，クライアントの負荷状況により VM レベルのオン・オフを行っている．また，R.Nathuji ら [23] は，VM と VMM が協調して省電力化を行う手法を提案している．しかし，この手法はディスク I/O やネットワーク I/O を考慮して省電力化を行われていない．

仮想化環境において，VCPU スケジューリングによる省電力化の研究も存在する．吉田ら [24] は，マルチコアの消費電力特性を考慮した VCPU スケジューラを提案している．その結果，全てのコアに低い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効果が高くなると示した．ゲスト OS の負荷と消費電力特性をフィードバックとしては，VCPU のスケジューリングを行う省電力化システムを C.Wen ら [25] が提案している．これらの手法は，本手法と連携して，さらに VM の消費エネルギーを削減できると考える．

システム全体の消費電力のうち，CPU は大きな割合を埋める場合が多いことから，CPU に関する省電力化の研究が多かった．しかし，ストレージデバイスやネットワークカードの消費電力に関する研究も存在する．L.Ye ら [26] は，VMM と VM のディスク I/O 情報を追跡することにより，ディスク・スピンを減らしてディスクのスリープ時間を延ばすことでディスクの消費電力を削減する手法を提案している．この手法は本研究が提案している CPU の省電力化と補完関係にあり，今後，CPU とディスクの消費電力特性の双方を考慮する手法が必要であると考える．

7. おわりに

本稿では，仮想化環境における省電力化を目的とし，消費エネルギー予測に基づいた省電力化を行う VMM の設計，実装と評価について述べた．本手法では，VM 実行時のキャッシュミス率やプロセッサ全体のメモリアクセス頻度などのハードウェアカウンタの情報と仮想 NIC，仮想 I/O の情報を用いて，多変量回帰分析により求めた最適な演算性能，消費電力またはスループットをもとに DVFS 制御を実施した．実装には，Linux カーネルを用いた VMM である KVM を用いた．評価より，コア単位で DVFS 制御可能なマルチコア環境において，演算性能とスループット

の条件の範囲内で VM が 1 台ある場合，最大でメモリバウンドベンチマークにて 38.3%，ディスクベンチマークにて 35.1%，ネットワークベンチマークにて 46.0%，VM が複数台ある場合，最大で 44.3%のプロセッサの消費エネルギーの削減率を確認した．今後の課題として，VM 上で複数のベンチマークが動作時のリソース競合による消費エネルギーの予測精度の分析が挙げられる．

謝辞 本研究は，科研費基盤研究 (B)「ユーザコンテキストに応じた電力管理による省電力コンピューティング環境の研究」によるものである．

参考文献

[1] Intel: Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor, http://www.intel.com/design/intarch/papers/30117401.pdf
[2] AMD: BIOS and Kernel Developer's Guide For AMD Family 10h Processors, http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/31116.pdf
[3] D. Brodowski: Linux kernel CPUfreq subsystem, http://www.kernel.org/pub/linux/utils/kernel/cpufreq/cpufreq.html
[4] Advanced Micro Devices: AMD Cool' n' Quiet, http://www.amd.com/jp-ja/Processors/ ProductInformation/0,,30_118_9485_9487%5E10272,00.html
[5] Xen project: http://www.xenproject.org/
[6] KVM for Server Virtualization: An Open Source Solution Comes of Age, http://www-03.ibm.com/systems/resources/systems_virtualization_idc__kvmforservervirtualization.pdf
[7] Intel VT-x: Intel Virtualization Technology and Intel Active Management Technology in Retail Infrastructure, http://www.intel.com/design/intarch/papers/316087.pdf
[8] AMD-V: http://sites.amd.com/uk/business/it-solutions/virtualization/Pages/amd-v.aspx
[9] M. Weiser, B. Welch, A. Demers, S. Shenker: Scheduling for reduced CPU energy, In Proc. of Sysmposium on Operating Systems Design and Implementation, Article No.2, 1994.
[10] W.Wu, M.Martonosi, D.W.Clark, V.J.Reddi, D.Connors, Y.Wu, J.Lee, D.Brooks: A Dynamic Compilation Framework for Controlling Microprocessor Energy and Performance, In Proc. of the 38th annual IEEE/ ACM International Symposium on Microarchitecture, pp. 271–282, 2001.
[11] W.Yuan, K.Nahrsted: Energy-efficient soft real-time CPU scheduling for mobile multimedia systems, In Proc. of the 19th ACM Symposium on Operation Systems Principles, pp. 149–163, 2003.
[12] 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：消費エネルギー予測によるマルチコア環境向け省電力化 Linux スケジューラ，情報処理学会 第 20 回コンピュータシステム・シンポジウム，Vol.2008，No.12，pp. 77–86，2008.
[13] 林 和宏，金井 遵，丸山 勝巳，並木 美太郎：L4 マイクロカーネルにおける省電力スケジューラの開発，情報処理学会研究報告，「システムソフトウェアとオペレーティング・システム」，第 108 回研究報告，Vol.2008-OS-108，pp. 147–154，2008.
[14] DOUANGCHAK SITHIXAY, 佐藤 未来子, 並木 美太郎：KVM を用いた仮想化環境における省電力制御の研究，SWoPP 北九州 2013，情報処理学会「システムソフトウェアとオペレーティング・システム研究会」 第 126 回研究発表会，Vol.2013-OS-126，No.8，pp. 1-9，July，2013.
[15] C.M.Kamga, G,S.Tran, L.Broto:Power-aware scheduler for virtualized systems. In Proc. of Green Computing Middleware on Proceedings of the 2nd International Workshop, Article No.5, 2011.
[16] S.Kundu, R.Rangaswami, K.Dutta, M.Zhao:Application Performance Modeling in a Virtualized Environment, In Proc. of High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium, pp. 1–10, 2010.
[17] 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), pp. 289–302, 2009.
[18] R.Bertran, Y.Bercerra, D.Carrera, V.Beltran, M.Gonzilez, X.Martorell, N.Navarro, J.Torres, E.Ayguade:Energy accounting for shared virtualized environments under DVFS using PMC-based power models, In Proc. of Future Generation Computer Systems, pp. 457–468, 2012.
[19] 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), pp. 839–848, 2005.
[20] 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), pp. 1–14, 2007.
[21] G.Dhiman, K.Mihic, T.Rosing: A system for online power prediction in virtualized environments using Gaussian mixture models, In Proc. of the 47th Design Automation Conference, pp. 807–812, 2010.
[22] G.Dhiman, G.Marchetti, T.Rosing: vGreen: A System for Energy-Efficient Management of Virtual Machines, In Proc. of the ACM Transactions on Design Automation of Electronic System, Vol.16 Issue 1, Article No.6, 2010.
[23] R.Nathuji, K.Schwan: VirtualPower: Coordinated Power Management in Virtualized Enterprise Systems, In Proc. of twenty-first ACM SIGOPS symposium on Operating systems principles, pp. 265–278, 2007.
[24] 吉田 哲也，山田 浩史， 佐々木，河野 健二，中村 宏：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，情報処理学会論文誌・コンピューティングシステム（ACS），Vol.4，No2，pp. 25–39，2011.
[25] C.Wen, J.He, J.Zhang, X.Long:PCFS: A Power Credit based Fair Scheduler under DVFS for Multi-core Virtualization Platform, In Proc. of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, pp. 163–170, 2010.
[26] L.Ye, G.Lu, S.Kumar, C.Gniady, John H.Hartman: Energy-Efficient Storage in Virtual Machine Environments, In Proc, of the 6th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pp. 75–84, 2010.