

# 指数平滑化法による KVM 仮想化環境における VM 動作予測に基づいた省電力制御

郭林瞻<sup>†1</sup> 佐藤未来子<sup>†1</sup> 並木美太郎<sup>†1</sup>

**概要** : 近年, クラウドを利用するデータセンターにおいて技術革新が進んでおり, 資源利用率の低下はデータセンター問題の一つなので, 仮想化技術を利用することにより, 実行性能を向上すると同時に, 消費電力削減が問題となる. 本研究では, KVM 仮想化環境による処理性能を保ちつつ省電力化可能とする手法を提案する. 事前学習を行わず仮想マシン (VM) 実行中に CPU, メモリ及び I/O の得た情報に基づき, 稼働状況の予測を実施する. 予測精度を向上させるため, Holt-Winters の指数平滑化を導入した. 各 VM の稼働状態により, 統計情報を分析しながら, システム全体の省電力化を実現する. 本稿では, KVM における各 VM の動作予測により, VMM レイヤで動的電圧・周波数制御 DVFS (Dynamic Voltage and Frequency Scaling) 手法を用いて制御する. 提案方式について実装, 評価を行い, 最大で 26.46% のプロセッサの消費エネルギーの削減率を実現した.

**キーワード** : 省電力制御, Holt-Winters 法, 動作予測

## Energy-saving Control based on Dynamic Prediction in KVM Virtualized Environments with Exponential Smoothing

LINZHAN GUO<sup>†1</sup> MIKIKO SATO<sup>†1</sup>  
MITARO NAMIKI<sup>†1</sup>

**Abstract**: In recent years, data center utilizes cloud to advance innovating technology. Resource utilization become one of the most important problems in the data center. Previous researches used virtualization technology to deal with the problem. However, they didn't consider both performance and energy-saving completely. In this paper, we propose a method to reduce electric power consumption with an efficient performance level. To assure a high precise prediction result for VM behavior according to the information of CPU, memory and IO without prior learning, predicted result is smoothed by holt-winters exponential smoothing method. The system analyzes these statistic information according to working status of each VM. And electric power control for the system will be implemented. To provide an energy saving control method for OS and processor, this paper utilizes dynamic voltage and frequency scaling method in VMM layer to control voltage and frequency, according to the behavior prediction for each VM on KVM. The system implemented the proposal and evaluation, the result showed the biggest 26.46% energy-saving.

**Keywords**: Energy-saving Control, Holt-Winters, Behavior Prediction

### 1. はじめに

近年, クラウドを利用するデータセンターにおいて技術革新が進んでいる. クラウドにおいては, サーバの集約やマシン構成の柔軟な変更, ロケーションによる自然災害への対応, 低価格化などのメリットから仮想化技術が多用されている. 仮想化技術を利用すると, 一つの物理ハードウェアを複数の独立した仮想化環境に分割し, 未利用の資源を効率的に活用できる. このような技術を利用して実行性能は向上可能だが, 消費電力の増大が問題となる. データセンターの中, サーバ, ネットワークデバイスからクーリングデバイスまで, データセンター全体の消費電力に影響

を及ぼすため, 全体の消費電力を削減するためには, それぞれの省電力化について研究する必要がある.

計算機システムの中, システム全体の消費電力のうち, プロセッサは多くの割合を占める場合が多いことから, プロセッサに関する省電力化の研究が盛んである. 現在, 仮想システムにおける消費エネルギーを考慮する CPU スケジューリングの研究[1]や, 仮想マシンの利用資源を考慮する動的マイグレーションの研究[2]があり, 省電力技術による研究が行われている. データセンターでは, CPU の消費エネルギーを削減するため適用される. 周波数と電圧に関する DVFS 制御手法がある.

従来の研究では, CPU から得られる情報を用いて DVFS 手法[3]による省電力制御を行うことが多かったが, ディスク, ネットワークなどの情報を考慮しないため, I/O を含

<sup>†1</sup> 東京農工大学  
Tokyo University of Agriculture and Technology

めた省電力制御が不十分である。先行研究[4]では、事前学習を用いて消費エネルギー予測に基づいた省電力制御を行った。しかし、仮想化環境における VM では、利用状況によって CPU などの負荷が変わることが多く、また、事前学習には手間がかかるため、学習データのないプログラムに適用できないという欠点がある。

本稿では、KVM 仮想化環境による計算機の処理性能を保ちつつ更なる省電力化の手法を提案する。事前学習を行わず各 VM の稼働状態を管理し、CPU 演算性能、ディスク I/O、ネットワーク I/O に関する統計情報を収集する。そして、Holt-Winters 法[5]により各 VM の動作特性を予測する。本手法による予測結果により、CPU 演算性能とスループットを保ちつつ消費エネルギーが最小となるような最適な周波数を決定する。最後に、DVFS 制御による各仮想マシンに最適な周波数と電圧を制御し、CPU 演算性能とスループットを保ちつつ消費エネルギー削減できる。

## 2. 関連研究

本章では、計算機環境における各種制御手法により省電力化の関連研究を簡単に説明する。

### 2.1 仮想化技術に基づく省電力制御

仮想化という用語は 1960 年代にすでに幅広く使われていた。計算機のリソースを抽象化することに基づき、異なる物理的位置にあるリソースを統合して資源利用率を改善でき、消費エネルギーを削減できる技術である。現在、クラウドの環境でもよく使われる仮想化技術では、単一計算機に複数仮想マシンを立ち上げ、マイグレーション、負分散などの技術を用い、資源利用率の増大や消費エネルギーの削減などが可能になる。

Nathuji ら[6]の研究では、消費電力に関する管理技術と仮想化技術を統合し、大規模なデータセンターにより最適化省電力化手法—VirtualPower を提案している。VirtualPower により VM ごとに省電力化を制御でき、異なる仮想化環境のプラットフォームの間や同一プラットフォームである VM の間に、消費電力のコントロールを協調でき、システム全体の省電力化を実現する。

Beloglazov ら[7]の研究では、仮想化環境におけるマイグレーション技術とアイドルノード制御を統合し、物理リソースの利用率、消費電力、演算性能などことを考慮し、VM の動的再分配手法を提案した。マイグレーションの消費電力が低くなる方式に基づき、資源利用率が低い VM をマイグレーションし、アイドル物理マシンをシャットダウンする。マイグレーションに関する方式として、HPG アルゴリズムを利用する。この手法により CPU の利用状況をコントロールできるが、マイグレーションのコストを検討していないことが問題となる。

仮想化技術は資源利用率を増大したり、実行する消費電力を削減したりするが、計算機システム管理が複雑になる。

### 2.2 DVFS 技術に基づく省電力制御

計算機システムの中、プロセッサは多くの割合を占める場合が多いことから、CPU に関する省電力制御の一つ、DVFS 制御を提供されている。DVFS 制御では、ハードウェアレベルである周波数・電圧を制御する手法である。省電力制御では、ハードウェアレイヤだけの制御が不十分であるから、ハードウェアレイヤである制御手法 DVFS に基づくシステムの状態を協調してソフトウェアの制御を行う手法として、Linux システムに搭載された cpufreq モジュール [8] と各種 governor システムを併せたものと Windows の Cool'n'Quiet [9] がある。

仮想化環境における CPU とメモリの負荷状況により DVFS を制御する研究として、Koala [10] の研究が存在する。この研究では、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。

演算性能を予測し、演算性能を出来る限り落とさず省電力化の研究として、W. Yuan ら [11] の研究が存在する。従来の研究では、DVFS 制御手法を利用して周波数を落とすと、演算性能またはスループットも低下する。演算性能が低くなる場合、実行時間を延長し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。この方式では、本研究の目的としているディスク I/O、ネットワーク I/O のスループットの低下を考慮した省電力制御に対応できない。筆者らの先行研究では事前学習を用いて消費エネルギーを予測したが、利用状況によって CPU などの負荷が変わることが多く、事前学習に基づく手法はこの変化に対応できない。

## 3. 課題と目標

省電力化に関する既存研究の問題に基づき、本研究で提案する省電力化の制御手法の位置づけを明確にするため、研究の課題と目標を示す。

### 3.1 仮想化環境による性能を保ちつつ更なる省電力化

本研究では、仮想マシンの特徴を把握するため、CPU 演算性能、ディスク I/O、ネットワーク I/O から統計情報を収集し、各 VM の稼働状況を予測する。各 VM の稼働状態により、消費エネルギーが最小となる最適な周波数と電圧を決定し、演算性能をできるだけ落さずに省電力化を目指す。

### 3.2 実行時情報だけによる省電力制御

先行研究は学習が必要であり、学習データから消費エネルギーを予測した。本研究では、仮想化環境に向けて VM 稼働変動への即応性を向上するため、事前学習を行わず VCPU コンテキストスイッチごとに VM の特性の統計情報

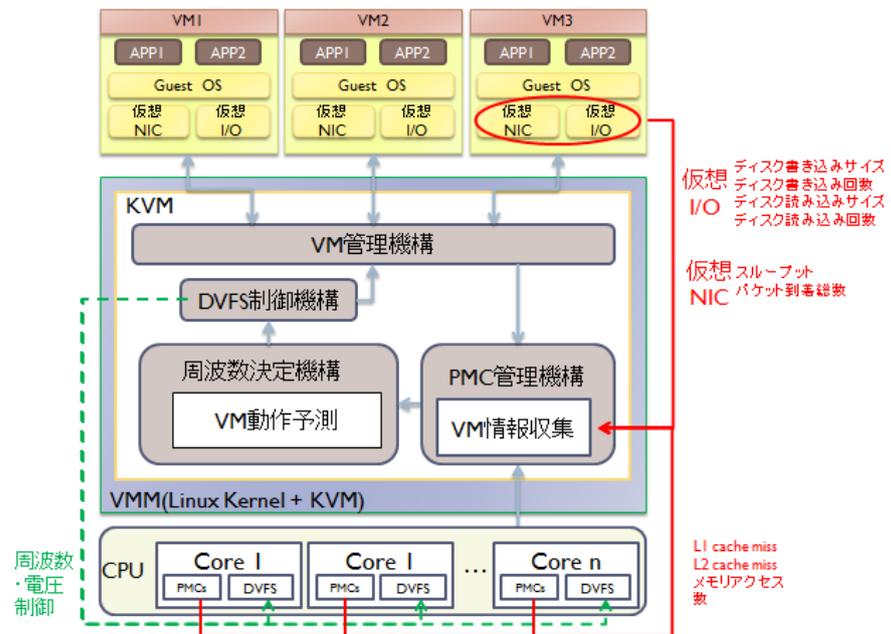


図 1 システムの構成

Figure 1 The configuration of system.

を収集し、各 VM の稼働状況を予測する。

省電力化の研究手法として、回帰分析をよく応用した。しかし、回帰分析では研究変量間の線形影響程度を分析し、分析結果はデータ量によって正確さが異なる。さらに、回帰分析では変量間の相互作用効果と非線形の影響を考慮しない。本研究では、Holt-Winters 法に基づいた高い精度で予測結果を得られるために、予測結果を平滑化し、CPU 演算性能とスループットを保ちつつ消費エネルギー最小となる各 VM の動作特性を予測する。OS とプロセスに適用な省電力を目標とする。

仮想化環境では一台のマシン上に複数の VM が存在し、各 VM で動作するアプリケーションが必ず同一特性を持つとは限らない。マルチプロセッサの上で、様々なアプリケーションが異なる特性を持ち、実行動作の予測もそれぞれ異なり、従来の計算機単体の省電力化の手法をそのまま利用することが困難である。以上の課題を踏まえて、KVM 仮想化環境で VMM レイヤにおける省電力機構を提案する。システムに関する具体的な方式は次章に説明する。

## 4. システム設計

本章では、KVM 仮想化環境における省電力化制御のシステム設計について述べる。本研究では、マルチコアプロセッサ上で稼働する仮想化環境を対象とした省電力制御手法を提案する。本手法を、VM ごとに適用し、VM 単位に異なる性能条件を設定できるようにすることで、VM の状況に応じた省電力制御を提案する。

### 4.1 全体構造

本節では、KVM を用いて仮想化環境を作成する。KVM は仮想マシンの管理に Linux の機能やノウハウを活用でき、

メモリ、ドライバ、スケジューラという機能は基本的に Linux カーネル機能を利用している。プロセッサの仮想化支援機構の利用を前提にすることで、KVM は他の仮想化技術よりもシンプルに実装できている。仮想化支援機構を利用することで、CPU の仮想化ができるが、支援機能が及ばない部分については QEMU などのエミュレーションのソフトウェアで実装している。KVM 利用時の QEMU はホスト OS 上ではユーザープロセスとしており、KVM を利用して Linux に仮想マシンの管理を行わせている。

仮想化環境における省電力制御の全体構成を図 1 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。以下では各機構の概要について説明する。

#### (1) VM 管理機構

VM 管理機構による各 VM の稼働状態を管理する。VM の情報とは、VM を識別するユーザープロセス ID、VM が持っている統計情報であるパフォーマンスカウンタの値と仮想 NIC・仮想 I/O に関する値を示す。VM 情報管理機構は、VM の挙動に変化がある場合、VM の情報を追加、削除、更新を行う。

#### (2) PMC 管理機構

パフォーマンス管理機構では、VM ごとのパフォーマンスカウンタ情報を管理する。周波数決定機構から命令を受け、パフォーマンスカウンタのカウント開始、カウント停止、カウント値の取得またはカウンタのクリアの処理を行う。

#### (3) 周波数決定機構

周波数決定機構を通じて Holt-Winters 法による各 VM の動作特性を予測する。この予測結果により複数台の VM が

動作している状態で、システム管理者が与えた演算性能とスループット制約の中で消費エネルギーが最小となるような最適な周波数を決定する。

#### (4) DVFS 制御機構

DVFS 制御機構では、周波数決定機構から命令を受け、実際にプロセッサの周波数と電圧を変更する。DVFS 機構において各仮想マシンに最適な周波数と電圧を制御し、CPU 演算性能とスループットを保ちつつ消費エネルギーを削減できる。

#### 4.2 統計情報の取得

VMの動作の予測には各VMの統計情報を利用するが、利用状況によってVMごとに動作予測が異なり、VMごとに稼働の動作特性や最適な周波数も異なる。そのため、VM単位にPMC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を保存する。

統計情報について、CPU、ネットワーク、ディスクに関する情報を取得する。つまり、タイムスライスごとにVM上で実行されたアプリケーションの実行時情報をVMM レイヤで取得する。CPUについては、キャッシュミス回数などをパフォーマンスカウンタを用いて取得し、ネットワークからは、スループットとパケット到着総数、また、ディスクからは、ディスクの読書き回数と読書きサイズを取得する。

##### (1) CPU に関する統計情報の取得

本研究では複数台VMを稼働する状況で行い、VMごとに複数VCPUが存在することが可能であり、さらに各VCPUはホストマシンの物理CPUに対応する状態が動的に変化する。そのため、本研究でVMごとにVMのタイムスライス単位として、VMごとのコンテキストスイッチではなく、VCPUコンテキストスイッチごとにCPUの統計状況を取得する。各VMのVCPU対応状況はVM管理機構により保存し、現時点のVMごとにすべてのVCPUに関するPCPUであるL1のキャッシュメモリアクセス、L2のキャッシュミス回数、L3のキャッシュミス回数を取得し、実行命令数は、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。

##### (2) ネットワークとディスクに関する統計情報の取得

本研究では、LinuxカーネルをホストVMとするKVM仮想化環境を用いて省電力化制御を行う。ネットワークとディスクに関するI/Oの統計情報を取得するため、I/Oの機能をエミュレーションするQEMUを利用する。QEMUからネットワークのパケット到着総数、ディスクの読込み回数、ディスクの読込みサイズ、ディスクの書込み回数、ディスクの書込みサイズを取得する。

#### 4.3 演算性能とスループットの見積り

DVFS 制御手法を利用して周波数を低くすると、演算性能またはスループットも低下する。演算性能が低くなり、実行時間が延びるため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。本研究では、統計情報によりVMの動作予測の上、各VMの演算性能とスループットを考慮し、最適な省電力制御を実現する。

#### (1) 演算性能

VMの演算性能の予測を実現するにあたり、アプリケーションの実行時のVMの演算性能を何らかの形で数値化する必要がある。一般的に、計算機における演算性能指標として、秒あたりの実行命令数IPS(Instruction Per Second)が用いられる。本研究の提案手法でも、アプリケーション実行時にVMの演算性能指標としてIPSを用いる。プロセッサのあるコアの周波数 $f$ において、アプリケーションの実行時間 $T_{app}$ 内に合計 $N_{inst}$ の命令が実行されたとする。このときのVMのIPS値を $IPS(f)$ とすると、 $I(f)$ は、

$$IPS(f) = \frac{N_{inst}}{T_{app}} \quad (1)$$

となる。実際にあるアプリケーションを実行したときのIPS値は、実行命令列に依存する。そのため、同じアプリケーションが実行される場合、各周波数 $f$ におけるIPS値は常に一定となる。

#### (2) スループット

VMがサーバとして使われている場合、外部のネットワークと通信することがある。外部のネットワークにあるクライアントの要求によって、必要とされる帯域幅が異なる。時間あたりの帯域幅をネットワーク通信のスループット値とする。実際に、物理ネットワークカードから各VMのスループット値を測定することは困難であるため、本研究では、VMに搭載する仮想ネットワークカード(仮想NIC)の値を利用する。プロセッサのあるコアの周波数 $f$ において、VMの実行時間 $T_{vm}$ 内に測定した仮想NICの帯域幅が $B$ とする。このときのVMのネットワークの平均スループット値を $TH(f)$ とすると、 $TH(f)$ は

$$TH(f) = \frac{B}{T_{vm}} \quad (2)$$

となる。

#### 4.4 消費電力の取得

本研究において、VMの消費電力の値は、VMがプロセッサのコア上で動作するときに、外部の測定器を使って実測する。ただし、マルチコアのプロセッサにおいては、共有キャッシュなどの各コアで共有する資源が存在することにより、各コアの消費電力の内訳を測定することは困難である。そこで本研究では、プロセッサ全体の消費電力をコア数で除算した値を各コアの消費電力と見なす。プロセッサのあるコアの周波数 $f$ において、アプリケーションの実行時間 $T_{app}$ 内に電力量 $P$ が消費されたとする。このときのVMの平均消費電力値を $P(f)$ とすると、 $P(f)$ は、

$$P(f) = \frac{P}{T_{app}} \quad (3)$$

となる

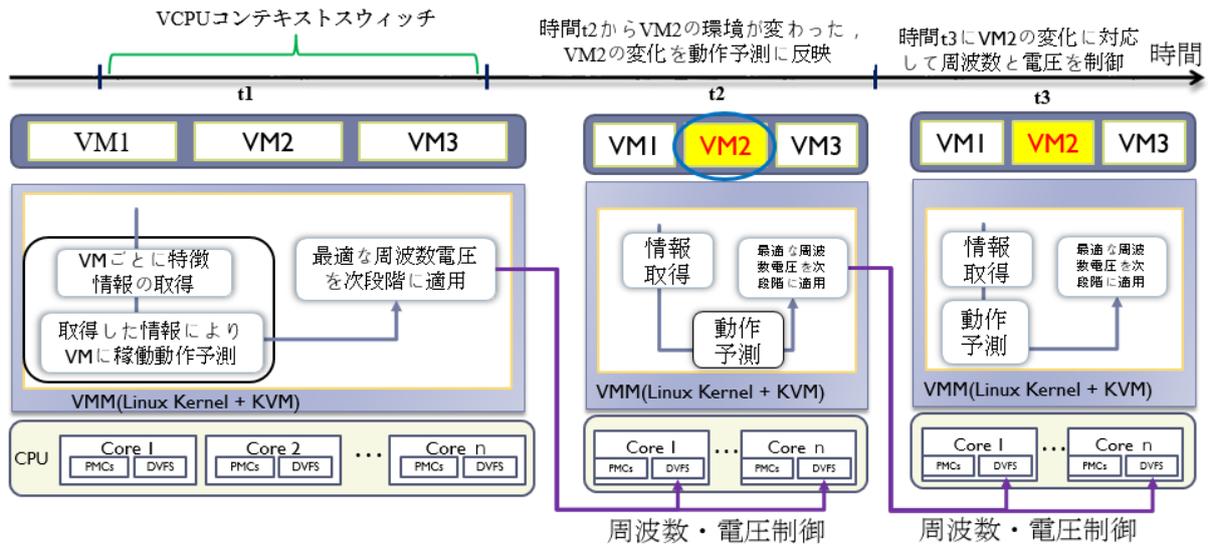


図 2 予測手順

Figure 2 Predicted operation.

以上のシステム構造により、時間単位で各VMから取得する情報を管理し、本研究で提案する予測手法を用い、予測モデルを構築する。そして演算性能やスループットを考慮しながら最適な制御を行う。次に予測手法について述べる。

### 5. VM 動作予測に基づいた省電力制御

省電力化の研究手法として、さまざまな手法が存在し、その中でも回帰分析がよく用いられてきた。しかし、回帰分析では研究変量間の線形影響程度を分析し、分析結果はデータ量によって正確さが異なる。さらに、回帰分析では変量間の相互作用効果と非線形の影響を考慮しない。

本研究では、VM ごとに性能を保ちつつ消費エネルギーの動的予測における省電力化を行う。短時間情報による高い精度で予測結果を得られるために、Holt-Winters 法を利用し、Holt-Winter の指数平滑化によりデータを平滑化しながら予測する。

準仮想化環境を提供する KVM の VMM レイヤにおいて、各種の周波数で動作させ、Holt-Winters 法を使って個々の VM の動作を予測する。予測した動作による最適な省電力制御を行う。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電圧・周波数を見つけ、VM 単位およびコア単位で DVFS を行う。

#### 5.1 Holt-Winters 予測手法

Holt-Winters 法の指数平滑法を、式(4)により示す。取得した過去のデータのうち、前回実測値  $r$  と前回の予測値  $p$  を用いて移動平均値を計算する方法である。

$$\text{予測値} = \alpha r + (1 - \alpha) p \quad (4)$$

$\alpha$  が平滑影響因数である ( $0 < \alpha < 1$ )。前回の実測値が予測値からどれほど外れたかを算出し、それに一定の係数  $\alpha$  を掛けて得た修正値を、前回の予測値に加減して今回の予測値を導き出している。このとき、実測値は修正値の算出

に使われるのみで、今回の予測値のベースになるのは前回の予測値である点がポイントである。これにより特異な実測値があった場合に影響が出過ぎることを避けている。前回の予測値と実測値があれば、今回の予測値が算出できる簡便さも特徴である。ただし、前回の予測値は前々回の予測値から算出されるため、連続する過去予測データの影響もわずかだが残る。

本研究では Holt-Winters 法を用い、KVM 仮想化環境における各 VM の統計情報により予測を行う。次に、指数平滑化手法である Holt-Winters を利用して各 VM の動作予測について述べる。予測手順は図 2 で示し、VM ごとに VM 特徴に関する統計情報を取得し、統計情報により VM の稼働状況を予測する。その予測結果に基づき、最適な周波数・電圧を制御する。仮想化環境における CPU などの負荷が変わることが多くため（例えば時間  $t_2$  に VM2 の環境が変わった場合）、前の制御方式を変化した VM に適用できない。再び各 VM から統計情報を取得し、VM2 の変化に対応した新たな制御を行う。

#### (1) 統計情報の取得

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 は全ての物理 CPU (PCPU) に対応付けられることを想定する。4.2 節で述べたように、統計情報は VCPU コンテキストスイッチ毎にパフォーマンスカウンタから取得する。統計情報とは、具体的には第 4 章で述べた CPU や IO などから取得する命令あたりの L2 キャッシュミス回数、命令あたりの L3 キャッシュミス回数、命令あたりの L1 キャッシュメモリアクセス回数、ネットワークのパケット到着総数、ディスクの読み込み回数、ディスクの読み込みサイズ、ディスクの書き込み回数、ディスクの書き込みサイズである。

そこで、統計情報の実測値  $X_t$  (命令あたりの L2 キャッシュミス回数  $x_{1,t}$ 、命令あたりの L3 キャッシュミス回数  $x_{2,t}$ 、

命令あたりのL1キャッシュメモリアクセス回数 $x_{3,t}$ , ネットワークの packets 到着総数 $x_{4,t}$ , ディスクの読み込み回数 $x_{5,t}$ , ディスクの読み込みサイズ $x_{6,t}$ , ディスクの書き込み回数 $x_{7,t}$ , ディスクの書き込みサイズ $x_{8,t}$ , 統計情報の予測値 $P_t$  (命令あたりのL2キャッシュミス回数 $p_{1,t}$ , 命令あたりのL3キャッシュミス回数 $p_{2,t}$ , 命令あたりのL1キャッシュメモリアクセス回数 $p_{3,t}$ , ネットワークの packets 到着総数 $p_{4,t}$ , ディスクの読み込み回数 $p_{5,t}$ , ディスクの読み込みサイズ $p_{6,t}$ , ディスクの書き込み回数 $p_{7,t}$ , ディスクの書き込みサイズ $p_{8,t}$ ) を定義する.

## (2) 指数平滑法に基づいた VM の動作予測

KVM 仮想化環境における VM の動作に影響される各種統計情報を, 式(4)に代入し, VM の動作を予測する. 統計情報  $i$  の予測式の説明変数は, その統計情報の実測値 $x_{i,t}$ と前回の予測値 $p_{i,t}$ とおき,  $\alpha_{i,t}$ を平滑影響因数とおく. 統計情報  $i$  の予測結果を $p_{i,t+1}$ とする. 予測式は式(5)となる.

$$p_{i,t+1} = \alpha_{i,t}x_{i,t} + (1 - \alpha_{i,t})p_{i,t} \quad (5)$$

実測値が予測値からどれほど外れたかを算出し, それに平滑影響因数 $\alpha$ を掛けて得た修正値を, 前回の予測値に加減して今回の予測値を導き出している. 一方, VM が起動した直後は前回の予測値がないため, 最初期の平滑影響因数 $\alpha_{i,0}$ を1とおく. 式(6)により平滑影響因数 $\alpha$ を求める.

$$\alpha_{i,t} = \frac{x_{i,t} - p_{i,t-1}}{x_{i,t-1} - p_{i,t-1}} \quad (6)$$

## (3) フィルター

CPUの周波数を下げると演算性能も下がり, 実行時間が延びるため, 必ずしも周波数を最低にすれば消費エネルギーを最小にできるとは限らない. そこで本研究では, VM の演算性能とスループットを保つため, 周波数と電圧の限界条件を設定する. 周波数と電圧の決定に用いる閾値はシステム管理者の運用によって設定する. 二種類の閾値である演算性能閾値とスループット閾値は, VMごとに指定する. 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最適な周波数と電圧を算出する. VMMレイヤでVM単位に求めた周波数と電圧をVCPUに割当ててPCPUへ設定することで省電力制御を行う. 演算性能, スループットと消費電力の取得は4.3節と4.4節に述べた.

本研究では, 各VMに対する過去のスループットと演算性能の平均値を, スループット閾値 $Ave_{thrp,t}^{v,f}$ と演算性能閾値 $Ave_{perf,t}^{v,f}$ として設定する.  $Ave_{thrp,t}^{v,f}$ は式(1)とVMの実行時間により得られる.  $Ave_{perf,t}^{v,f}$ は式(2)とVMの実行時間により得られる

統計情報集合の実績値 $X_t$ , 統計情報集合の予測値 $P_t$ をそれぞれの関係式の説明変数とおき, 演算性能に関する $A_{perf,t}$ と $b_{perf,t}$ , スループットに関する $A_{thrp,t}$ と $b_{thrp,t}$ , 消費電力に関する $A_{pow,t}$ と $b_{pow,t}$ をそれぞれの関係式の係数と

おく. 統計情報集合の実測値に基づいた演算性能 $y_{perf,t}^{v,f}$ , スループット $y_{thrp,t}^{v,f}$ , 消費電力 $y_{pow,t}^{v,f}$ , 及び統計情報集合の予測値に基づいた演算性能 $q_{perf,t}^{v,f}$ , スループット $q_{thrp,t}^{v,f}$ , 消費電力 $q_{pow,t}^{v,f}$ による関係式を式(7)–式(12)のように示す.

$$y_{perf,t}^{v,f} = A_{perf,t}X_t + b_{perf,t} \quad (7)$$

$$y_{thrp,t}^{v,f} = A_{thrp,t}X_t + b_{thrp,t} \quad (8)$$

$$y_{pow,t}^{v,f} = A_{pow,t}X_t + b_{pow,t} \quad (9)$$

$$q_{perf,t}^{v,f} = A_{perf,t}P_t + b_{perf,t} \quad (10)$$

$$q_{thrp,t}^{v,f} = A_{thrp,t}P_t + b_{thrp,t} \quad (11)$$

$$q_{pow,t}^{v,f} = A_{pow,t}P_t + b_{pow,t} \quad (12)$$

式(7)–式(12)により, 演算性能に関する $A_{perf,t}$ と $b_{perf,t}$ , スループットに関する $A_{thrp,t}$ と $b_{thrp,t}$ , 消費電力に関する $A_{pow,t}$ と $b_{pow,t}$ を求める. 式(7)–式(12)で求めた $\alpha$ と式(5)で求めた統計情報の予測値を用いて, 式(13)–式(15)により, 各VMの演算性能, スループット, 消費電力を予測することができる.

$$q_{perf,t+1}^{v,f} = A_{perf,t}P_{t+1} + b_{perf,t} \quad (13)$$

$$q_{thrp,t+1}^{v,f} = A_{thrp,t}P_{t+1} + b_{thrp,t} \quad (14)$$

$$q_{pow,t+1}^{v,f} = A_{pow,t}P_{t+1} + b_{pow,t} \quad (15)$$

式(16)により, VM動作予測の結果による各種周波数・電圧でスループット及び演算性能が閾値を超える周波数・電圧だけ選択する. 一方, VMの稼働状況による各種周波数・電圧でスループット及び演算性能が閾値より全部低い場合について,  $Ave_{perf,t}^{v,f}$ と $Ave_{thrp,t}^{v,f}$ を0とおく.

$$F = \{f | (q_{perf,t+1}^{v,f} > Ave_{perf,t}^{v,f}) \&\&$$

$$(q_{thrp,t+1}^{v,f} > Ave_{thrp,t}^{v,f})\} \quad (16)$$

## (4) 最適な制御の分析

以上の予測手法に基づき, 式(17)により, 消費エネルギーが最小となる電圧・周波数を見つけ, VM 単位およびコア単位でDVFSを行う.

$$f_{t+1} = \{f | \min(q_{pow,t+1}^{v,f}) | f \in F\} \quad (17)$$

## 6. 実装と評価

本章では, Holt-Winters 指数平滑法によるシステムの実装と評価について述べる. システムの評価として, 仮想化環境における SPEC CPU 2006 ベンチマーク, ネットワークペ

ンチマークである `httperf`、ディスクベンチマークである `bonnie++` を用いて省電力化の評価を行った。

### 6.1 実装

前章までに述べた設計に基づき、本手法の有効性を評価するため、AMD FX-8370 プロセッサを搭載した PC において実装を行った。FX-8370 プロセッサは、動作クロックが通常時 4GHz、TDP が 125W、命令 256KB・データ 128KB の各コア占有 L1 キャッシュ、8MB の L2 キャッシュ、8MB の L3 キャッシュを持つ。本研究に用いた環境で DVFS 省電力機構の周波数は 5 段階で 1400MHz、2100MHz、2800MHz、3400MHz、4000MHz であり、周波数は各コアを変更し、電圧は全コアの電圧を変更した。

仮想化環境における提案手法により、カーネルモジュールである `kvm-kmod` と I/O などをエミュレーションする `QEMU-kvm` に機能追加を行った。KVM は、Intel VT-x[12] や AMD-V[13] といった CPU の仮想化支援機能を利用することを前提に設計され、比較的シンプルな構造で仮想化機能を実現した。KVM は、仮想マシンの管理に Linux の機能やノウハウを活用でき、メモリ、ドライバ、スケジューラという機能は基本的に Linux カーネル機能を利用しているため、KVM 自体のスケールは小さいである。本研究では、x86 アーキテクチャを目標とし、x86 の中で特に電源電圧・周波数制御は CPU やチップセット依存となるため、AMD の FX-8370 をターゲットとした。

本研究では、省電力制御をおこなうため、FX-8370 プロセッサにおける `CoolCore`[14] テクノロジーに基づいた DVFS 省電力機構を利用する。

### 6.2 評価

本手法について評価を行い、AMD FX-8370 プロセッサを搭載した PC を評価マシンとして次の三種類ベンチマークの実験を行った。

- ・SPEC CPU 2006 ベンチマーク
- ・ネットワークベンチマーク : `httperf`
- ・ディスクベンチマーク : `bonnie++`

システムの電力削減効果の評価するために、提案手法を適用したシステムと適用しないシステムにおける各ベンチマークの消費エネルギーを計測した。評価環境を表 1 のように示す。

表 1 評価環境

Table 1 Evaluation environment.

CPU	AMD FX-8370 (8 cores / 4.0GHz)
メモリ	16GB
仮想化	KVM, QEMU
各 VM の VCPU	2
各 VM のメモリ	512MB

本手法による仮想化環境における複数状況を評価するため、SPEC CPU 2006 ベンチマーク、ネットワークベンチマークである `httperf`、ディスクベンチマークである `bonnie++` を用いて省電力化の評価を行い、各ベンチマークを以下に紹介する。

SPEC CPU 2006 はシステム性能評価協会 (SPEC: Standard Performance Evaluation Corporation) により、CPU、メモリなどの処理性能を評価するように開発された。本研究では、CPU バウンドなベンチマークの `444.namd`、メモリバウンドなベンチマークの `462.libquantum` を用いて評価を行った。

ネットワークベンチマークに関する評価は HTTP 負荷実験ツールである `httperf` を用いて実験を行った。評価方法として、ほかの計算機から一台 VM を起動し `httperf` を用いて負荷をかけた。`httperf` は 1 秒あたりのリクエスト数を変更することにより、VM に負荷を加減することができた。本研究では、一秒あたりのリクエスト数を 300 と設定し評価を行った。

ディスク I/O がボトルネックになったときに、本手法が有効であるかどうかを確認するために、`bonnie++` を選んで評価をした。

以上のベンチマークにより、仮想化環境で VM 一台と複数台 VM の評価を行った。システムの電力削減効果の評価するため、提案手法を適用したシステムと適用しないシステムにおける各ベンチマークの消費エネルギーを計測し、適用しないシステムにおける各ベンチマークの消費エネルギーを 1 とおき、それに対する提案手法を適用したシステムにおける各ベンチマークの消費エネルギーを図 3 と図 4 に示す。複数台 VM のベンチマークを表 2 に示す。

表 2 複数台 VM のベンチマーク  
Table 2 Benchmarks of multi-VMs.

Scenario	VM1	VM2	VM3
1	444.namd	462.libquantum	-
2	444.namd	bonnie++	httperf
3	462.libquantum	bonnie++	httperf

図 3 において、横軸はベンチマークであり、縦軸が提案手法を適用したシステムにおける各ベンチマークの消費エネルギーの値を、適用しないシステムにおける各ベンチマークの消費エネルギーを 1 とした時の相対値を表している。一台 VM に対して CPU バウンドのベンチマークが 15.40%、メモリバウンドのベンチマークが 20.38%、ディスクベンチマークが 24.60%、ネットワークベンチマークが 17.76% のプロセッサの消費エネルギーの削減を確認した。

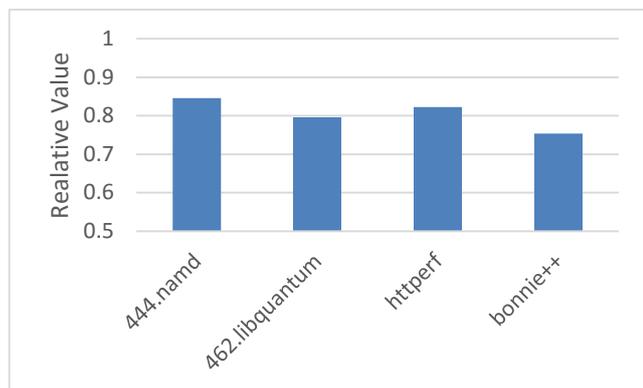


図 3 VM 一台の評価

Figure 3 Evaluation of one VM.

図 4 において、表 2 に基づいた複数台 VM に各種ベンチマークの組み合わせの評価結果である。横軸は scenario1, 2, 3 であり、縦軸が scenario1, 2, 3 における提案手法を適用したシステムの消費エネルギーの値を、適用しないシステムの消費エネルギーを 1 とした時の相対値を表している。複数台 VM に対して scenario 1 が 20.74%, scenario 2 が 13.21%, scenario 3 が 26.46% のプロセッサの消費エネルギーの削減を確認した。

## 7. 終わりに

本論文では、KVM 仮想化環境における省電力化を目的とし、指数平滑化法による仮想マシンの動作予測に基づいた省電力化を行なう VMM の設計、実装と評価について述べた。提案した省電力化手法は、VMM が各 VM の稼働状態を管理し、CPU 演算性能、ディスク I/O、ネットワーク I/O に関する統計情報を収集する。そして、Holt-Winters 法により各 VM の動作特性を予測する。その予測結果から、CPU 演算性能とスループットを保ちつつ消費エネルギーが最小となるような最適な周波数を決定する。実験結果により、本手法について消費エネルギーが最大 26.46% の削減を確認することができた。

今後は、指数平滑化法による VM 動作の予測結果の誤差について検討する。なお、VM ごとに各デバイスの利用状況に基づいたデバイスに関する周波数・電圧制御を考慮する。

## 参考文献

- 1) 吉田哲也, 山田浩史, 佐々木広, 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, IPSI Trans. on Advanced Computing Systems.
- 2) 広瀬崇宏, 中田秀基, 小川宏高, 伊藤智, 関口智嗣, 仮想マシン技術とサーバー時停止技術を利用した省エネデータセンターシステムの開発, 先進的計算基盤システムシンポジウム SACSIS2010, 情報処理学会, ポスター, May 2010.
- 3) Etienne Le Sueur, Gernot Heiser, Dynamic voltage and frequency scaling: the laws of diminishing returns, Proceedings of the 2010 international conference on Power aware computing and systems, p.1-8, October 03, 2010, Vancouver, BC, Canada.
- 4) DOUANGCHAK SITHIXAY, 佐藤未来子, 並木美太郎: KVM

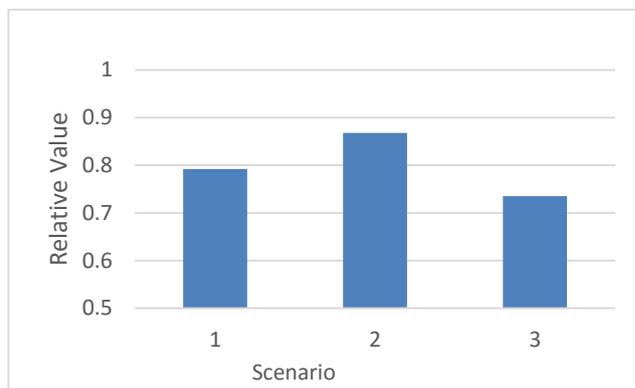


図 4 複数台 VM の評価

Figure 4 Evaluation of Multi-VMs.

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.

5) P. Goodwin "The holt-winters approach to exponential smoothing: 50 years old and going strong", Foresight, pp.3034, 2010.

6) Nathuji, R., K. Schwan, VirtualPower: coordinated power management in virtualized enterprise systems, in Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles. 2007, ACM: Stevenson, Washington, USA. p. 265-278.

7) Beloglazov, A., R. Buyya. Energy Efficient Allocation of Virtual Machines in Cloud Data Centers. in 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), 2010: p. 577-578.

8) D. Brodowski: Linux kernel CPUfreq subsystem, <http://www.kernel.org/pub/linux/utils/kernel/cpufreq/cpufreq.html>

9) Advanced Micro Devices: AMD Cool'n'Quiet, [http://www.amd.com/jp-ja/Processors/ProductInformation/0,,30\\_118\\_9485\\_9487%5E10272,00.html](http://www.amd.com/jp-ja/Processors/ProductInformation/0,,30_118_9485_9487%5E10272,00.html)

10) David C. Snowdon, Etienne LeSueur, Stefan M. 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.

11) W. Yuan, K. Nahrstedt: Energy-efficient soft real-time CPU scheduling for mobile multi-media systems, In Proc. of the 19th ACM Symposium on Operation Systems Principles, pp. 149-163, 2003.

12) Intel, I.: Intel 64 and IA-32 Architectures Software Developer's Manuals. (2007)

13) AMD. AMD-V Nested Paging. White Paper, AMD: <http://developer.amd.com/assets/NPT-WP-1/%201-final-TM.pdf>, July 2008.

14) AMD. AMD Cool'n'Quiet Technology Overview. Retrieved 2009-04-23.