クラウドの消費電力をおさえる高速 VM スケジューラ

2011SE081 石原宏紀　2011SE289 山田圭悟
指導教員：宮澤元

1　はじめに

非常に多数のサーバを仮想化技術や分散処理技術，大規模コンピュータ運用技術を用いてまとめ，ネットワークを介してさまざまなサービスを提供するクラウドコンピューティング (クラウド) が普及している．クラウドでは常に多数のサーバを動かし続けるので，消費電力削減は大きな課題の一つである．クラウドにおける消費電力を削減するには不要なサーバの電源を落とし，起動しているだけで消費する電力を削減することなどが挙げられる．しかし，サーバの電源を安易に落とすとクラウドそのものの性能を引き出すことができなくなってしまう．そこでリソースを有効活用しつつ，できる限りクラウドの消費電力を削減したい．既に IaaS クラウドにおける仮想マシン (VM) の配置を最適化する VM スケジューラは存在する [1]．しかし扱うサーバ数や VM 数が増大するとそれに伴いスケジューラにおける最適化の計算量も膨大になり，効率的に実現するのは困難である．

本稿では扱う既存 VM と物理サーバを絞り込むことで高速にスケジューリングできるアルゴリズムを提案する．物理サーバが起動しているだけで消費する電力は VM の容量 1 単位あたりに比例して消費する電力に比べ小さく，既存 VM の移動についても同様のことがいえることに着目し，計算対象となる既存 VM と物理サーバの数を大幅に減らすことができる．計算実験により，提案アルゴリズムの有効性を示す．

2　IaaS クラウドにおける VM スケジューリング

従来のクラウド基盤ソフトウェアにおける VM のスケジューリングアルゴリズムとしてグリーディ法（貪欲法）による割り当てとラウンドロビン法による割り当てが広く使われている．グリーディ法は VM を順番に物理サーバにリソースが足りなくなるまで割り振り，足らなくなったら次の物理サーバに同様に適用するといったものである．ラウンドロビン法は各物理サーバに対して順番に VM を割り振る手法である．これらの手法は消費電力について考慮しておらず，消費電力を低減するための様々な研究が行われている．

2.1　クラウドの省電力化手法

クラウドの消費電力削減や限られたリソースの有効活用は大きな研究テーマになっており，様々な工夫がなされている．ここでは，クラウドの消費電力を削減するための研究のいくつかを紹介する．

- VM 間のトラフィック交流を考慮した仮想サーバの効率的な配置方法の提案 [2]
  VM 間のトラフィック交流に着目し，物理サーバのリソースを有効活用する VM 配置方法を提案した．
- 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 [3]
  消費エネルギー予測に基づいた電力を削減する VMM を開発し，評価を行った．
- 大規模 VM 負荷予測・配置制御技術によるシンクライアント・データセンターのグリーン化 [4]
  VM 負荷予測技術を用いることでサーバ全体の状態を把握し，VM の配置を制御する．空きサーバが出来ることにより，サーバ起動による消費電力の削減を実現した．

2.2　OR による VM スケジューリング手法

坂本らは OR による最適化モデルを用いた VM スケジューラにより，物理サーバに対する VM の配置を行うことで IaaS クラウドの消費電力削減を行っている [1]．この研究では OR による最適化モデルの計算に以下の要素を用いる．

- 各サーバが起動しているだけで消費する電力
- 各サーバの VM 一単位あたりに対しての消費する電力
- VM が移動する際に消費する電力

2.2.1　問題点

坂本らの方法では消費電力は最適化できるものの，扱う VM とサーバの数が多くなると最適化計算に非常に時間がかかる．例えば，サーバを起動するかしないかの組み合わせは 2 のサーバ数乗通り，1 台の新規 VM のサーバへの配置の組み合わせはサーバ数通り，サーバ間の VM 移動の組み合わせはサーバ数の既存 VM 数乗通りを考える必要がある．**図 1** に坂本らの研究でのスケジューリングの最適化計算にかかる時間を示す．VM 数が 8 になると 2 秒ほどかかり，VM 増加に伴い処理時間は急激に増加していることがわかる．

3　アプローチ

本節では, 坂本らの方法 [1] を踏まえ, OR による VM スケジューリングの最適化計算の近似手法について述べる．

3.1　前提条件

まず坂本らの研究結果 [1] より，ある物理サーバの容量が 27208 単位に対して物理サーバの容量を 1 単位利用するときに消費する電力は 16W であり，一方でサーバが起動しているだけで消費される電力は 50W と見積もってい

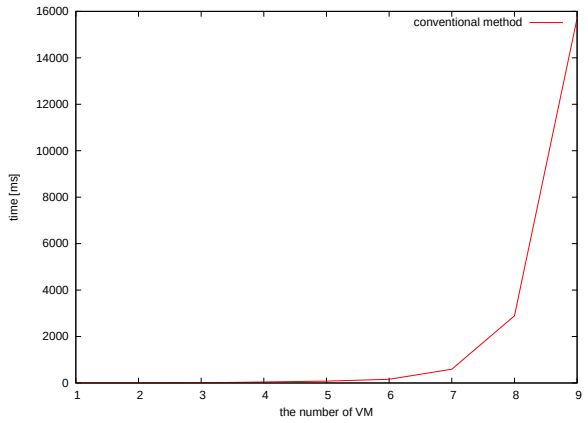

図 1　従来の手法における実行時間

たので，VM の要求に対して処理を行うのに消費する電力の方が大きいといえる．これをふまえ私たちは以下の前提条件を設定した．

- サーバが起動しているだけで消費する電力は VM の要求に対しての処理を行うのに消費する電力に比べ小さい

この前提条件に基づき，VM の単位要求あたりの処理にかかる消費電力を基準にサーバを使用する際の優先順位をつけることで，実際に計算時に扱うサーバ数を制限する．

また，今回は簡単のため以下の前提をおいた．

- サーバそれぞれについて，サーバ容量，サーバの容量を 1 単位利用するのにかかる電力，サーバが起動しているだけで消費する電力が事前に与えられているとする．
- 利用にあたっては，初期状態（既存 VM の数が 0）からこの手法を適用し，新規 VM は一度に一つずつ追加するものとする．その際，n 番目の新規 VM を追加し，VM のスケジューリングが終了するまでの工程を第 n フェーズと呼ぶことにする．
- ここではマイグレーションのコストは考慮せず，できるだけ最適な配置を追求する．

3.2　提案アルゴリズム

新規 VM を追加する際は，新規 VM をどのサーバに配置するかだけではなく，既存 VM のサーバ間の移動も考慮し，できる限り電力が抑えられるような配置を求める．そのため 1 フェーズ毎にそれぞれのサーバについて，既存 VM と新規 VM の中から配置する VM を選択していくことでスケジューリングを行う．

アルゴリズムの大まかな手順を示すと次のようになる．

1. 利用するサーバの優先順位決定
2. VM の配置
3. VM 配置の確定判定

以下に詳細を述べる．

3.2.1　サーバの優先順位決定

消費電力が少なくなる VM の配置では，基本的にサーバの容量を 1 単位利用するのにかかる電力が小さいサーバをできるだけ多く利用するという特徴がある．そこで，その特徴を用いて，**図 2** に示すように，サーバを利用する優先順位を決め，優先度の高いサーバから順にサーバ容量が満杯になるまで VM を配置していくという方法をとる．

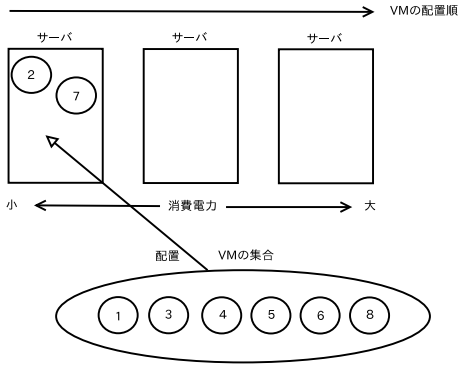

図 2　VM の配置の流れ

利用するサーバの優先順位を求めるにあたっては，次の要素を利用する．

1. サーバの容量を 1 単位利用するのにかかる電力
2. サーバの容量
3. サーバが起動しているだけで消費する電力

優先順位の求め方は，まず，1 の小さいものから並べる．もしその順の中で 1 が同じ値になるものがある場合は，その部分の中で 2 が大きいものからになるよう並べ替える．さらに 1，2 が同じ値になる部分がある場合は，その部分の中で 3 が小さいものからになるよう並び替える．

3.2.2　VM の配置

VM をサーバに配置する際は，前もって決定したサーバの利用優先順位にしたがってその順に配置していく．配置する VM はサーバの容量を超えない，最も大きな VM の組み合わせを配置する．すなわち，そのサーバにおいて最も無駄ができないように VM を配置する．配置されていない VM が無くなるまで，サーバの優先順に沿って VM を配置していく．

新規 VM を追加する際は，**図 3** に示すように再度最初のサーバから順に，新規 VM を含んだ VM の集合の中から最も無駄のない VM の組み合わせを配置していく．

3.2.3　配置の確定

フェーズがある程度進み，VM の数が多くなった場合は，考えられる組み合わせの数が膨大になり，配置する VM を決定するのに膨大な計算時間がかかってしまう．一方，配置した VM がサーバ容量にぴったり収まっていて無駄が無い状態になったときはそのサーバだけについて考えれば，それ以上の効率の良い組み合わせは見つからない．

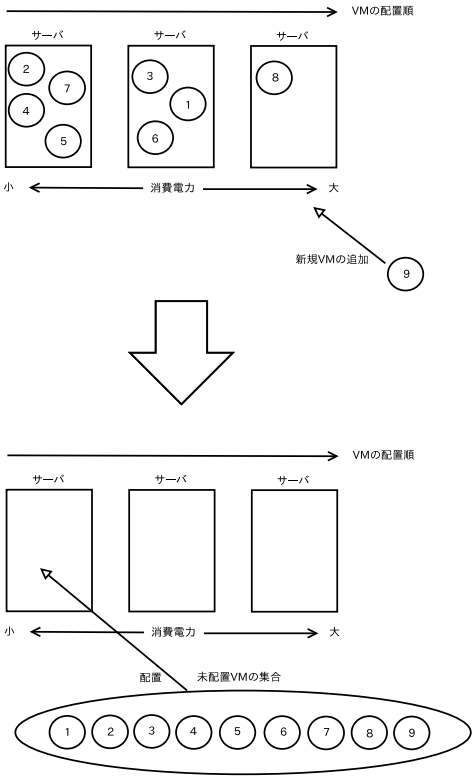

図 3 新規 VM 追加の流れ

そこで，その状態を保存しておき，次のフェーズでも，その組み合わせをそのまま利用する．ここでは VM がサーバ容量にぴったり収まっている場合，配置が確定したと表現し，そのサーバを確定サーバ，そのサーバに配置された VM を確定 VM と呼ぶことにする．次のフェーズでは，VM の組み合わせを計算する際に，確定 VM 以外の組み合わせのみを考えれば良いので，組み合わせを計算する VM の数を減らすことができる．

3.2.4 余白の設定

配置の確定を考えるようにしたが，VM の数が多くなってもぴったりサーバの容量を満たす組み合わせが見つからず，確定しないというパターンも考えられる．そこで**図 4** に示すように，サーバにある程度の余白を設定し，配置された VM の合計が，サーバ容量から余白を差し引いた部分を越えた場合やぴったり収まった場合は満杯とみなし，確定させる．ここでいうサーバにおける余白とは，その部分に VM を配置することができないというものではなく，そのサーバが確定するかどうかを判定することのみに利用する．このような余白を設定することにより使用するサーバが確定する割合が上がり，未確定 VM の数が多くなりすぎるのを防ぐことができる．

ただ，余白の部分を大きくとってしまうと，サーバが利用していない無駄な部分が多く発生してしまう．これでは最適配置の精度が下がってしまう．逆に余白を小さくしすぎると，ほとんど確定が起こらず，未確定 VM の数が多くなりすぎてしまう可能性がある．そこで余白の大きさを未

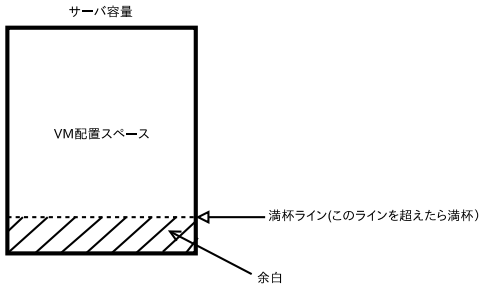

図 4 サーバにおける余白

確定 VM の数に応じて動的に変化させるようにする．未確定 VM の数が少ない場合は余白を 0 にして最適配置の精度を重視し，ある程度未確定 VM の数が増えてきた場合は膨大な計算時間がかかってしまうのを防ぐために高速化を重視し，余白を大きくする．

3.3 近似点の考察

最適配置を計算するにあたり，近似化を行った．近似化することによって最適配置の精度が下がる可能性がある点として，

1. サーバ容量よりも消費電力を優先している．
2. 余白が存在するまま確定することがある．
3. 配置する VM の組み合わせは，現在配置しようとしているサーバについてのみ考慮する．
4. 確定したサーバでの VM の組み合わせは変更されない．

などが挙げられる．しかし上記の項目を近似することによって最適配置の精度が下がるということが頻繁に起こるとは考えにくいので，近似を行っても大きな誤差にはならないと考えられる．

4 実験

本節では，提案手法が OR を用いた従来の手法と比べ高速な処理を実現しつつ，精度が落ちていないことを検証する．ここでは，提案手法において設定する余白は未確定 VM 数が 30 以下の場合は 0 を設定し，30 を超える場合においては，VM 数と 30 との差分の 2 乗を余白として設定している．

実験に使用したコンピュータの性能を**表 1** に示す．

表 1 実験に使用したコンピュータの性能

CPU	Intel Core i7-3770
コア数	4 コア (8 スレッド)
クロック周波数	3.40GHz
メモリ	8GB
HDD	1TB
OS	Ubuntu Server 12.10

4.1 計算時間の比較

提案手法において，フェーズごとにどのくらいの時間がかかったかを計測した．その結果を**図 5** に示す．

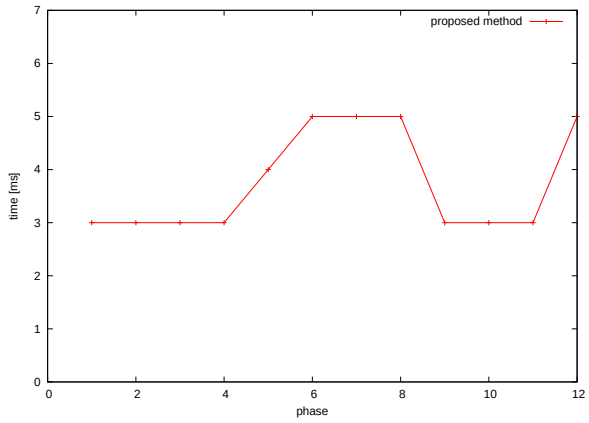

図 5 提案手法におけるフェーズごとの実行時間

図 1 の従来の手法と比較すると圧倒的に処理時間が短縮されていることが分かる．また，配置の確定を行うことにより，VM 数が増えても処理時間の増加が抑えられていることが分かる．ここでは，第 8 フェーズで確定が起こっており，第 9 フェーズから処理時間が短くなっている．

4.2 最適配置精度の比較

従来の手法での配置計算を完全に最適なものとし，提案手法がどれだけの精度を保っているかを比較する．それぞれの手法で同じ要求の値を追加していったところ，第 11 フェーズまでは全く同じ配置になった．これによって提案手法でもかなりの精度で最適配置が行うことができていることが分かる．

4.3 余白による処理時間の検証

上記の実験では，未確定 VM の数が増加しすぎる前に確定が起こっていたので，余白が 0 のままであり，余白を考慮した成果が現れていない．そこで追加する VM の要求を複数組み合わせてもサーバにぴったり収まることのない中途半端な値に設定して，余白を動的に変化させたことよる成果がどの程度得られるかの実験を行った．その結果を**図 6** に示す．

第 39 フェーズで確定が起こり，第 40 フェーズからは高速に実行することができていることが分かる．余白を動的に変化させない場合は，第 40 フェーズにおいても第 39 フェーズにおける実行時間より多くの時間がかかり，さらにフェーズ数が増えていくごとに膨大な実行時間がかかると考えられるので，余白を未確定 VM の数によって変化させることで処理時間を大幅に短縮することができた．

5 議論

本稿で述べた VM スケジューリングアルゴリズムは，現状では VM が増加する場合しか考慮していない．現実のクラウドでは，処理を終了した VM が削除されたり，VM の要求性能が変化するなどの状況に対応する必要がある．このような VM の動的な変化に対応するための本アルゴリズムの改良として，VM の削除や要求の変化が起こった際に確定条件を満たすことができていない場合や，サーバの容量を超えてしまった場合は確定を取り消し，もう一度スケジューリングを考慮するという方法が考えられる．

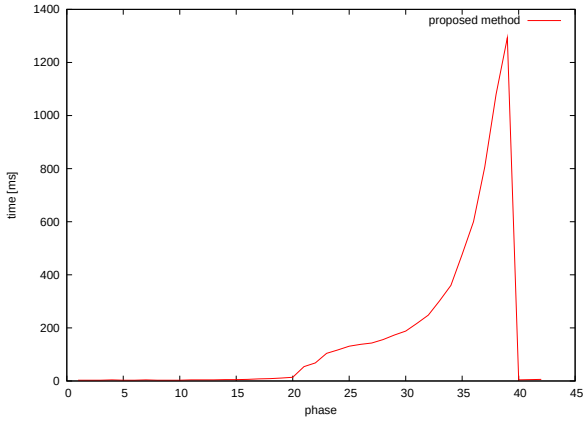

図 6 余白を変動させた実行時間

6 まとめ

我々は IaaS クラウドにおける消費電力削減のために OR による最適化計算を用いて，物理サーバに対する VM の配置の最適化を近似的に行った．物理サーバの使用に優先順位をつけ，ある程度サーバのリソースを使いきることができれば今後考えなくても良いという確定条件を設定することで大規模システムにも対応できる．しかし，提案アルゴリズムではクラウドにおける既存 VM の要求やクラウドを維持しているサーバ状況の動的な変化には対応できていない．今後はこのような動的な変化に対応する実用的な VM スケジューラの開発を目指す．

参考文献

[1] 堀 将大，坂本 光司: “IaaS クラウドにおける消費電力を最適化する VM スケジューラ”，南山大学情報理工学部 2014 年度卒業論文，2015.

[2] 朝倉 浩志，倉上 弘，山田 博司: “VM 間のトラフィック交流を考慮した仮想サーバの効率的な配置方法の提案”，第 10 回情報科学技術フォーラム L-014，2011.

[3] Douangchak Sithixay，佐藤 未来子，山田 浩史，並木 美太郎: “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”，情報処理学会研究報告 Vol.2013-OS-127，2013.

[4] 中村 暢達，喜田 弘司，竹村 俊徳，藤山 健一郎: “大規模 VM 負荷予測・配置制御技術によるシンクライアント・データセンターのグリーン化”，NEC 技報 Vol.62 No.3. pp.101-104.2009.