

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

2011SE051 堀 将大 2011SE224 坂本 光司

指導教員：宮澤 元

1 はじめに

現在、コンピュータネットワーク経由で様々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）というコンピュータの利用方法が普及している。クラウドはクラウド基盤ソフトウェアを用いて多数の物理サーバを集約することで実現されている。クラウドのサービスモデルは大きく3種類に分類され、アプリケーションソフトウェアを提供する SaaS (Software as a Service)、データベース等のプラットフォームを提供する PaaS (Platform as a Service)、仮想マシン (VM: Virtual Machine) やネットワークを提供する IaaS (Infrastructure as a Service) がある。

どのサービスモデルも大量の物理サーバを集約して構築されており、常に稼動していることが求められるので、クラウドの消費電力が大きな問題となっている。例えば、2013年の国内データセンターにおける年間消費電力量は122億5千万キロワット時となっている [1]。また、データセンターで消費される電力のうち IT 機器が30%を占めている [2]。そのため、クラウドの省電力化や限られたリソースの有効活用は大きな研究テーマとなっており、様々な工夫がなされている。

クラウドの省電力化を実現する1つの手段として、VMが動作していない物理サーバの電源を落とすことがある。しかし、単に物理サーバの電源を落として消費電力を可能な限り抑えようとすれば、リソースを有効活用することはできず、クラウド全体の性能低下に繋がってしまう。また、リソースを有効活用して VM の性能を引き出そうとすれば、クラウドの消費電力は大きくなってしまふ。

我々は3種類のサービスモデルの中で最もハードウェアに近く、計算資源そのものを提供する IaaS に注目し、IaaS クラウドにおける VM の配置を最適化する VM スケジューラの開発を行っている。VM スケジューラの開発にあたって、我々は消費電力に影響を与えるリソースを実験によって調べる。この実験結果をもとにオペレーションズ・リサーチ (OR: Operations Research) の最適化手法を用いて、VM のスケジューリングを行う。この VM スケジューラを利用することで、クラウドにおける VM の物理サーバに対する配置が最適化され、消費電力の削減とリソースの有効活用を両立させることができる。

2 クラウド基盤ソフトウェア

本節ではクラウド基盤ソフトウェアについて、その構成及び省電力化の観点から見た VM スケジューリングについて述べる。

2.1 クラウド基盤ソフトウェアの一般的な構成

クラウド基盤ソフトウェアはクラウド処理の基盤を提供するソフトウェアである。サーバやネットワーク機器、ストレージ等を仮想化してクライアントに提供し、クラウドに対する様々な要求に柔軟に対応する。実際に利用されているクラウド基盤ソフトウェアとして、商用では Amazon EC2/S3 [3, 4] 等があり、オープンソースソフトウェアでは CloudStack [5], OpenStack [6] 等がある。

一般的なクラウド基盤ソフトウェアは物理サーバ上で動作し、VM を提供するソフトウェア、VM に対してストレージを提供するソフトウェア、クラウド全体の管理を行うソフトウェア等から構成される。

2.2 CloudStack

具体的なクラウド基盤ソフトウェアの構成として、図1に CloudStack の構成を示す。CloudStack は主に管理サーバ、プライマリストレージ、セカンダリストレージ、ホストから構成される。

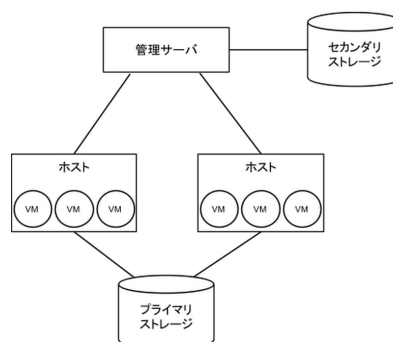


図1 CloudStack の構成

管理サーバ

クラウドリソースの管理を行い、利用者に対して Web 管理画面や API を提供する。この管理サーバが VM のスケジューリングを行っている。また、VM にストレージと IP アドレスを割り当てる。

プライマリストレージ

物理サーバ上で動作する全ての VM のディスクボリュームを格納する。

セカンダリストレージ

VM を作成するための OS イメージやスナップショットを格納する。

ホスト

単一のコンピューティングノードであり、実際のクラ

ウドサービスはゲスト VM の形式でホストから提供される。各ホストには VM を管理するためのハイパーバイザがインストールされている。

2.3 VM スケジューリング

一般に CloudStack における管理サーバに相当するものが、各物理サーバの性能や負荷状況に応じてクラウドにおける VM のスケジューリングを行っている。

2.3.1 既存の VM スケジューリングアルゴリズム

従来のクラウド基盤ソフトウェアでは VM のスケジューリングアルゴリズムとして、グリーディ法とラウンドロビン法の 2 方式が広く使われている。グリーディ法は 1 台の物理サーバに対して、そのリソースが枯渇するまで VM を割り当て、リソースが枯渇したら次の物理サーバを使用する。ラウンドロビン法は複数の物理サーバに対して順番に VM を割り当て、最後の物理サーバに VM を割り当てると、もう一度最初の物理サーバを使用する。

2.3.2 問題点

グリーディ法による割り当てとラウンドロビン法による割り当てを比較した場合、グリーディ法は使用する物理サーバの数が少なく済み、クラウド全体の消費電力を抑えることができる。しかし、グリーディ法による割り当ては VM が指定されたとおりの性能を發揮できるかどうかについて考慮しているとはいえない。

また、CloudStack では利用可能な CPU リソースをサービスオフリングと呼ばれる VM 作成時に指定された VM の静的な要求リソースをもとに決定する問題がある。利用可能な CPU リソースは物理サーバの CPU コア数と周波数を掛けた値から、各 VM のサービスオフリングの CPU コア数と周波数を掛けた値を引いて計算されるので、アイドル状態の VM が存在し、実際には物理サーバのリソースに余裕がある状態でも、新たに VM を割り当てることができない。

2.4 関連研究

VM スケジューリングによってクラウドの省電力化を実現する様々な研究が行われている。

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 [7]

クラウド利用者によって、動作している VM 数の増減が繰り返され、各物理サーバが提供できる CPU 資源やメモリ資源に断片化が起こる。これを解決するために、VM 間のトラフィック交流を考慮した VM スケジューリングの研究が行われている。

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 [8]

KVM 仮想化環境において、各 VM の消費エネルギー

予測に基づいた省電力制御の研究が行われている。各 VM の演算性能から VM が消費するエネルギーを予測し、それをもとに周波数と電圧を変化させて省電力化を実現している。

3 OR を用いた VM スケジューラ的设计

2.4 節に示すようにクラウドの省電力化について多くの研究が行われているが、OR を用いた VM のスケジューリングはこれまで行われていない。そこで、我々は消費電力に影響を与えるリソースを調べ、そのリソースをもとに OR による最適化モデル [9] を使用して、物理サーバに対する VM の配置を行い、クラウドの省電力化を実現できないかと考えた。

OR による最適化モデルを用いて VM スケジューリングを行うには、最適化モデルにデータを設定する必要がある。図 2 に VM スケジューラに最適化モデルを適用した時の構成を示す。最適化モデルは管理サーバから必要なデータを取得し、最適化計算の結果を管理サーバに返す。管理サーバはその結果をもとに VM スケジューリングを行う。

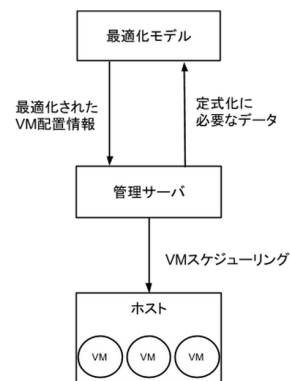


図 2 最適化モデルを適用した VM スケジューリング

3.1 定式化に必要なデータ

OR による VM スケジューリングの最適化モデルを使用するために必要なデータを以下に示す。

- 物理サーバの容量
- 新規 VM の要求
- 既存 VM の要求
- 既存 VM がどの物理サーバを利用しているか
- 既存 VM が物理サーバ間を移動する際にかかるコスト
- 物理サーバが起動しているだけで消費する電力
- 物理サーバの容量を 1 単位利用するのにかかる電力

3.2 設計方針

我々が作成する VM スケジューラ的设计方針として、今回は新規 VM をどの物理サーバに配置するかということに重点を置く。物理サーバ間で既存 VM をマイグレーションする機能は実装しない。サービスオフリングによって

設定されたリソースの値を最低限保証すべき VM の性能とする。利用者からの VM 作成要求が少なく、物理サーバのリソースに余裕がある場合、VM はサービスオフラインで指定したリソースを越えて動作するが、そのような VM については無視する。そのかわり、作成要求が多い場合はその VM が存在する物理サーバに集中的に配置を行う。

4 予備実験

3.1 節で述べた定式化に必要なデータのうち、物理サーバと VM が消費する電力量を測定し、消費電力に影響を与える VM のリソースを確認する実験を行った。

4.1 実験環境

実験に使用した物理サーバの性能を表 1 に示す。クラウド基盤ソフトウェアは CloudStack 4.4 を使用し、ハイパーバイザは KVM (Kernel-based Virtual Machine) [10] を使用した。VM のゲスト OS には CentOS 5.5 を導入し、VM に負荷をかけるために stress コマンドを用いた。電力計はサンワサプライ社の TAP-TST7 を使用した。

表 1 コンピュータの仕様

物理サーバ	
CPU	Intel®Core™i7-4770
コア数	4 コア (8 スレッド)
クロック周波数	3.40GHz
メモリ	16GB
HDD	1TB
OS	Ubuntu Server 14.04

4.2 実験内容

CloudStack を使用して、電力計を取り付けた物理サーバ 1 台に CPU コア数とメモリ容量を様々に変えて VM を作成し、VM 上で stress コマンドを実行させた時の VM の CPU 使用時間、システム全体の CPU 使用時間とそれに伴う消費電力を測定する。そして、測定した VM とシステム全体の CPU 使用時間から VM の CPU 使用率を計算する。stress コマンドで実行する処理は平方根の計算である。以下に実験の流れを示す。

1. ホスト OS にて VM とシステム全体の CPU 使用時間を出力する
2. ホスト OS から ssh を使い、VM で stress コマンドを実行する
3. 60 秒後にホスト OS にて VM とシステム全体の CPU 使用時間を出力する

この実験を各条件毎に 10 回ずつ繰り返し、CPU 使用率と消費電力の平均値を求めた。

4.3 CPU コア数とメモリ容量を変化させた時の結果

CPU コア数とメモリ容量を変化させた時の CPU 使用率と消費電力を測定した。その結果を表 2 に示す。この

実験から CPU コア数を 2 から 4 に増加させた時、VM の CPU 使用率は約 2 倍になり、消費電力は約 16W 増えることがわかった。また、メモリ容量を 2GB から 4GB に増加させた時、VM の CPU 使用率と消費電力はほとんど同じであることもわかった。なお、アイドル状態での物理サーバ自体の消費電力は約 50W であった。

表 2 VM1 台での CPU 使用率と消費電力

CPU コア数	メモリ	CPU 使用率	消費電力
2	2GB	26.52%	94.5W
2	4GB	26.64%	96.0W
4	2GB	51.47%	112.1W

4.4 CPU コア数を変化させた時の結果

4.3 節の実験結果を踏まえて、CPU コア数だけを 1 から 8 まで変化させて同様の実験を行った。その結果を表 3 に示す。この実験から CPU コア数を増やすことで、CPU 使用率は CPU コア数にほぼ比例して増加することがわかった。それに対して、消費電力は必ずしも CPU コア数に比例して増加しないことがわかった。

表 3 VM1 台での CPU コア数に応じた CPU 使用率と消費電力

CPU コア数	CPU 使用率	消費電力
1	14.04%	80.8W
2	26.52%	94.5W
3	38.88%	107.4W
4	51.46%	112.1W
5	63.73%	125.5W
6	76.15%	127.0W
7	88.60%	128.7W
8	99.77%	129.8W

4.5 複数の VM を動作させた時の結果

1 台の物理サーバ上に複数の VM を動作させて実験を行った。その結果を表 4 に示す。この実験から動作している VM の CPU コア数の合計値が同じ時、動作している VM の数に関わらず、CPU 使用率と消費電力はほとんど同じであることがわかった。また、表には記載していないが CPU の周波数を変化させた時の VM の CPU 使用率と消費電力は、変化させない時と比べてほとんど同じだった。

4.6 考察

それぞれの実験結果から、平方根の演算処理において消費電力に最も影響を与えるリソースは VM の CPU コア数であることがわかった。そして、動作している VM の数に関わらず、VM が使用している CPU コア数の合計値を考慮して VM スケジューラを開発すれば良いこともわかった。

表 4 複数 VM 動作時の CPU 使用率と消費電力

VM 数	CPU コア数	CPU 使用率	消費電力
2	1	27.53%	97.7W
3	1	40.40%	107.4W
4	1	53.38%	113.8W
2	2	52.48%	111.7W
3	2	77.89%	127.8W
2	3	77.36%	127.1W
2	4	100%	130.4W

5 VM スケジューラの実装

第 3 章で述べた VM スケジューラを CloudStack 4.4 を用いて実装した。

5.1 実装の概要

CloudStack では VM の配置先となる物理サーバは Allocator と呼ばれる Java プログラムによって決められている。我々は CloudStack が標準で使用する Allocator にコードを追加する形で VM スケジューラを実装した。主な処理の流れは以下のとおりである。

1. 最適化モデルの計算で必要となる各データを取得してファイルへ出力する
2. 外部の Java プログラムを実行する
 - 出力されたデータファイルをもとに CPLEX[11] を利用して OR による最適化計算を行う
 - 計算結果に基づき VM の配置先となる物理サーバ名を取得してファイルへ出力する
3. 物理サーバ名が書き込まれたファイルを読み込み、その物理サーバの情報を呼び出し元へ返す

5.2 最適化計算機能の実装

Allocator に最適化モデルの計算に必要な各データを取得してファイルへ出力する機能を追加した。このファイルをもとに、CPLEX を使用して最適化モデルの計算を行うが、CPLEX は外部の Java プログラムを実行して使用する。そのため、Allocator に外部のプログラムを実行する機能を追加した。最適化計算により VM を配置する物理サーバが決定し、その物理サーバ名が書き込まれたファイルが出力される。このファイルを読み込み、配置先となる物理サーバの情報を呼び出し元へ返す機能を追加した。これにより、指定した物理サーバが VM の配置先となる。

6 まとめ

我々は IaaS クラウドにおいて、クラウドの消費電力を削減するために OR による最適化計算を用いて、物理サーバに対する VM の配置を最適化する VM スケジューリングについて研究を行った。具体的には、クラウド基盤ソフトウェアである CloudStack を使用して予備実験に

より消費電力に影響を与えるリソースを確認した。その後、CloudStack のソースコードに追加する形で VM スケジューラを実装した。

今回は VM スケジューラの有用性を確認する実験を行うまでには至らなかった。また、今回の実装では静的なりソースの値をもとに新規 VM をどの物理サーバに配置するかということに重点を置いたので、動的なりソースの値をもとにした既存 VM のマイグレーション機能は実装していない。しかし、現実のクラウドでは多数の物理サーバで VM が動作するので、動的なりソースの値をもとにした既存 VM のマイグレーション機能は必要であると考えている。実装した VM スケジューラの有用性の確認と動的なりソースの値をもとにした既存 VM のマイグレーション機能の実装が今後の課題である。

参考文献

- [1] IDC Japan 株式会社: “国内データセンターの電力消費予測を発表”, <http://www.idcjapan.co.jp/Press/Current/20141203Apr.html>, 2014.12.3.
- [2] 環境省: “平成 20 年度環境技術実証事業ヒートアイランド対策技術分野 IT 機器等の消費電力・排熱量”, http://www.env.go.jp/air/tech/model/heat_aeh-wg2-20_02/ref01.pdf, 2014.12.3.
- [3] Amazon Web Services: “Amazon EC2”, <http://aws.amazon.com/jp/ec2/>, 2015.01.07.
- [4] Amazon Web Services: “Amazon S3”, <http://aws.amazon.com/jp/s3/>, 2015.01.07.
- [5] The Apache Software Foundation: “Apache CloudStack: Open Source Cloud Computing”, <http://cloudstack.apache.org/>, 2015.01.07.
- [6] Rackspace Cloud Computing: “Open Source Cloud Computing Software”, <http://www.openstack.org/>, 2015.01.07.
- [7] 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”, FIT2011 第 10 回情報科学技術フォーラム 10(4), 203-204, 2011.09.07.
- [8] 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 2013-EMB-31 (8), pp.1-10, 2013.
- [9] 宮澤 元: “IaaS クラウドの省電力スケジューリング論文投稿の準備メモ”, 2014.12.
- [10] Red Hat: “KVM”, http://www.linux-kvm.org/page/Main_Page, 2014.10.07.
- [11] IBM: “IBM ILOG CPLEX”, <http://www-03.ibm.com/software/products/ja/ibmilogcple>, 2015.1.10.