パブリッククラウドの品質制御に向けた物理サーバのリソース使用量推定と VM の配置変更先決定技術

辻拓人[†1]　野間唯[†2]　宇治橋善史[†3] 樋口淳一[†4]　横山乾[†5]
† 株式会社富士通研究所 〒 211-8588 神奈川県川崎市中原区上小田中 4-1-1
E-mail: † tsuji.takuto@jp.fujitsu.com

概要：パブリッククラウドは社会に欠かすことのできない技術になっている。また、パブリッククラウドでは、複数の利用者の VM が同じサーバ上で動作し、同じサーバ上の各 VM が CPU などのリソースを分け合うという特性を持っている。このとき、同じサーバ上で動作する各 VM が、同じ時間帯にリソースを同時に利用することでリソース競合が発生し、利用者のサービスの応答時間の悪化が発生するという問題がある。この問題に対して、パブリッククラウドで行うサーバの定期的なメンテナンスやローリングアップデートで生じる VM の配置変更を利用することにより、事前に応答時間の悪化の発生を抑制する方法が知られている。

本研究では、VM のリソース使用量の時間変化を表すモデルに確率分布を利用し、サーバのリソースの使用量を推定することで、VM の配置変更を行う方法を提案する。これにより、リソースの使用量に周期性がない VM でも応答時間の悪化が抑制されるサーバに配置変更することが可能になった。実際に動作していた VM の CPU 使用率のログを用いて、評価実験を行った。提案手法と比較手法の VM の配置変更先に関する比較を行い、様々な配置状況でも応答時間の悪化の発生が抑制されるサーバに VM を配置変更することが可能なことを確認した。また、実験により、提案手法はパラメータの最適化が不要なことを確認した。

キーワード：パブリッククラウド、仮想マシン、配置変更、CPU 使用率、カーネル密度推定、マルコフ連鎖モンテカルロ法

Resource usage estimation of physical server and decision of VM placement for control of public cloud quality

TAKUTO TSUJI[†1]　YUI NOMA[†2] YOSHIFUMI UJIBASHI[†3] JUNICHI HIGUCHI[†4]
KEN YOKOYAMA[†5]
† FUJITSU LABORATORIES LTD. 4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588 Japan
E-mail: † tsuji.takuto@jp.fujitsu.com

Abstract: Public cloud becomes important technology in society. It has a character that Virtual Machines (VMs) of multiple users are operated on a same server and share physical resources. Due to this character, resource congestion occurs and the response time of services of the user become worse because VMs on a same server use resource at the same time. In order to solve this problem, there are methods that suppress the response time worsening in advance by moving VMs to another server. One of the method utilizes the period for periodic maintenances or rolling updates operated by the public cloud provider, or maintenances to respond to user-claims to move VMs.

We proposed a probability-based selection method to which a VM is moved in order to suppress the response time worsening in advance. Our method is applicable to the period for periodic maintenances and rolling updates. We modeled a VM's resource usage with respect to time in a probability distribution and estimated the resource usage of the server by using the models of VMs on the server. As a result, we are able to select a server that has low probability of high load from the servers even when there is no periodicity in resource usage. We evaluated our method by using CPU usages of VMs that are operated in a cloud. We confirmed that our proposed method can select the server that suppresses the response time worsening in many cases compared with the existing method. Furthermore, we confirmed robustness of our proposed method by the experiments.

Keywords: Public cloud、 Virtual machine、 Live migration、 CPU usage、 Kernel density estimation、 Markov chain Monte Carlo

1. 背景

パブリッククラウド(Amazon Web Service, Google Cloud, Microsoft Azure, NIFCLOUD など)は、クラウド事業者がネットワークを通して、利用者が必要とするとき、必要な量を契約してリソース(CPU、メモリ、ネットワーク、ストレージ)を利用者に提供するサービスである。このため、利用者は自分たちで各リソースを物理的に用意する必要がなくなり、また、リソースの見積もりを行うことなく自分のサービスの開発に専念することが可能になる。そのため、パブリッククラウドの利用者は多く、パブリッククラウドは社会において重要な技術となっている。しかし、パブリッククラウドは問題が存在している [1]。その中でクラウド事業者の観点だと、ほとんどリソースを使用しない利用者の Virtual Machine (VM、仮想マシン)しかないサーバは電力を無駄に浪費してしまうといった問題が存在している。一方で、利用者側の観点だと、必要とするリソース以上のリソースをクラウド事業者と契約してコストが増加したり、他の人のリソース使用量の影響を受けて応答時間の悪化が

発生したりといった問題が存在している。本論文では、応答時間の悪化の発生に関する問題について着目する。

パブリッククラウドでは様々な利用者の VM が同じサーバ上で動作し、サーバ上の各々の VM はリソースを共有している。空いているサーバのリソースを効率的に利用するため、ベストエフォートでリソースが提供されることがある。このとき、同じサーバ上の各 VM が互いにリソースを取り合うことで、リソース競合と呼ばれる現象が発生する。リソース競合が発生すると、利用者は契約した分のリソースを使用することが出来なくなり、利用者がエンドにユーザに提供するサービスで応答時間の悪化が発生してしまう。このため、クラウド事業者が、VM の配置変更を行ったり、利用者の利用可能なリソースの使用量に制限をかけたりすることで、リソース競合による応答時間の悪化の発生を解決、もしくは抑制をする必要がある。応答時間の悪化の発生を抑制する方法として、パブリッククラウドの定期的なサーバのメンテナンス、ローリングアップデートで生じる VM の配置変更を利用するものが存在している[2]。この方法は VM のリソース使用量の周期性を利用して、長期間の VM のリソース使用量の時間変化を表すモデルを作成する。このようにして作成した VM のリソース使用量のモデルを利用して、事前に応答時間の悪化の発生が抑制されるサーバに VM を配置変更する。

実際に、図 1.1 のように VM 上では、VM のリソース使用量が周期的になるプログラムが行う定期的なバッチ処理が存在している。一方で、図 1.2 のように VM のリソース使用量が周期的にならない人が行う事務処理やデータ分析といったサービスが存在している。このような状況で VM のリソース使用量の周期性を利用したモデルに基づく VM の配置変更を行うと、VM のリソース使用量のモデルと実際のリソース使用量の間で乖離するため、VM の配置変更先のサーバで応答時間の悪化が発生してしまう。

このような問題がどのくらい発生するか把握するため、実際に動作している 2000 台の VM の CPU 使用率を分析した。周期性が存在するかの判断基準として自己相関関数を利用した。1 週間の時間差での自己相関関数の値が 0.7 以上ならば、VM のリソースの使用量に周期性が存在しているものとした。この分析の結果、VM のリソースの使用量に周期性が存在するものが全体の 15%しかなかった。したがって、VM のリソースの使用量の周期性を利用したモデル化[2]は実際のクラウドシステムでは不適切であると判断した。

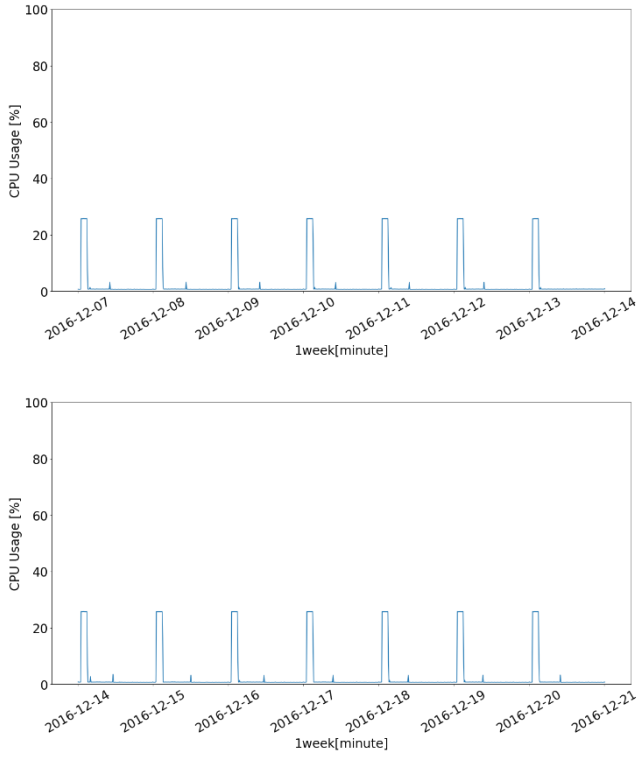

図 1.1 実際に動作する VM の CPU 使用率で周期性があるものの一例。上図は 2016 年 12 月 7 日金曜日から 1 週間の VM の CPU 使用率となり、下図は次の 1 週間の VM の CPU 使用率となっている。次の週の VM の CPU 使用率も類似した変化となり、1 週間の時間差の自己相関関数の値は 0.87 となっている。

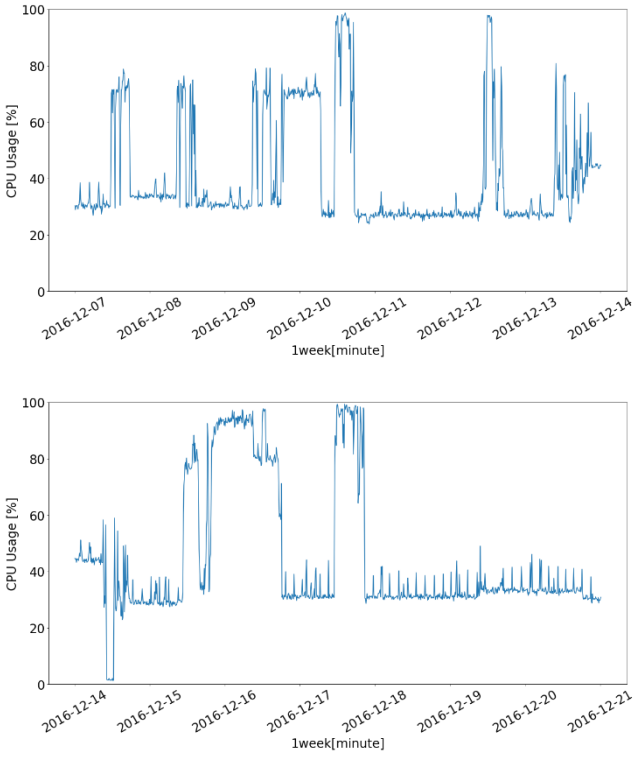

図 1.2 実際に動作する VM の CPU 使用率で周期性がないものの一例。上図は 2016 年 12 月 7 日金曜日から 1 週間の VM の CPU 使用率となり、下図は次の 1 週間の VM の CPU 使用率となっている。次の週では VM の CPU 使用率が大幅に変化している時間帯があり、1 週間の時間差の自己相関関数の値は 0.22 となっている。

我々はこの問題を解決するために、各曜日、時間帯ごとの VM のリソース使用量の確率分布を考えることにより、VM のリソース使用量の時間変化を表すモデルを作成する方法を提案した。そして、作成したモデルからサーバのリソースの使用量を推定する方法と VM の配置変更先の決定方法までを含めた提案を行った。実際に動作する VM の CPU 使用率を利用してシミュレーションを行うことにより、様々な配置状況での評価を行うことにより、比較手法に対する有効性と提案手法の配置変更先を決定するときのパラメータに頑健性があり最適化が不要なことを確認した。

2. 関連研究

VM のリソース競合に関する研究をいくつか挙げる。短期間の VM のリソース使用量の監視を行い、動的に VM の配置変更を行うことでサーバのリソース競合の発生を解決する研究が存在している[3]。[3]は VM のリソース使用量を短い周期(e.g. 5 分)で監視して、過剰にリソースが使用されていたら、Live Migration (ライブマイグレーション、再起動を必要としない VM の配置変更[4])により、動的に VM の配置変更を行う。[2]とは異なり、長期間のリソース使用量を考慮しないので、VM の配置変更後にすぐに VM の配置変更が発生する可能性がある。さらには予め応答時間の悪化が発生しづらいサーバに配置変更する用途には向いてない。

他にも短期間の VM のリソース使用量を利用して VM の配置変更やリソースの使用量に制限をかけることによりリソース競合を解決する手法が存在している。短期間の VM のリソース使用量を時系列モデルと分布(短期間の時間窓内のリソース使用量のヒストグラム)の両方を利用して分析を行うことにより動的な VM の配置変更を行うもの[5]が存在している。他にも PID 制御によるリソースの動的な制御[6]、指数移動平均

による動的なロードバランシング[7]などが存在している。

長期間のリソース使用量を利用した VM の配置変更技術に関連する研究で、背景で述べた[2]以外だと、VM のリソース使用量の相互相関関数を利用し、リソースの使用量の変化が互いに関係する VM をまとめることでリソース使用量をモデル化する方法[9]が存在している。この手法の問題点として、[2]と同様にリソース使用量に周期性が存在しないとき、自己相関関数が低くなるのと同様に相互相関関数も低くなってしまうので適切に VM のリソース使用量をモデル化できないという問題がある。

3. 提案手法

3.1 アプローチ

周期性が低い、オンデマンドなサービスを提供する VM のリソース使用量に対してもうまく表現できるモデルを作成するために、各曜日の時間帯ごとのリソース使用量の確率分布を考える。このように考えた理由は、確率分布を利用することにより、時系列モデルで推定ができない周期性がなく前週との変化が大きい場所でも、確率的に把握可能と考えたためである。また、各曜日の時間帯に考えた理由は、各曜日時間帯の単位での周期性は考慮したいと考えたためである。これにより、周期性が低い VM のリソース使用量が存在していても、適切に VM のリソース使用量をモデル化することが可能になると考えた。

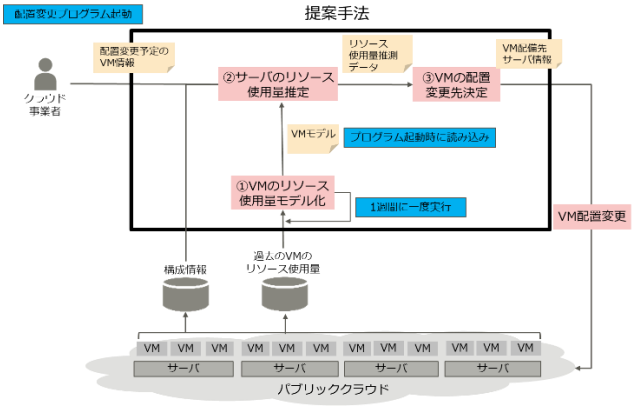

図 2 提案手法の全体像。クラウド事業者が配置変更する VM を特定できる情報(e.g. VM の名称、ID)を渡すことで、対象の VM を応答時間の悪化が発生しづらいサーバに配置変更する。

3.2 全体の流れ

提案手法の全体像は図 2 のようになっている。図 2 の① VM 負荷モデル化は、クラウド事業者がこのプログラムを利用するときに、すぐにモデルを利用できるよう、1 週間に一度 VM のモデルの作成を行う。また、パブリッククラウドの基盤から取得できる過去 3 週間の VM のリソース使用量を利用して、セクション 3.3 で述べる方法により VM の 1 週間のリソース使用量の変化を表す VM のモデルを作成する。提案手法のプログラムは、クラウド事業者が利用者からの要望を受ける、もしくは、クラウド事業者がパブリッククラウドの基盤のメンテナンスを行うときに実行する。クラウド事業者は、提案手法のプログラムの実行時に、配置変更予定の VM を指定する。また、パブリッククラウドの基盤は、VM のリソース使用量(10 分周期で保存される平均 CPU 使用率、VM に割り当てられた論理 CPU すべてが 100%になるとき、この値は 100%になる)を監視しており、構成情報(各サーバの名称、各サーバ上で起動している VM の名称、サーバの物理 CPU 数、VM の論理 CPU 数)と、

過去 3 週間分の各 VM の 10 分周期毎の VM のリソース使用量が取得可能であるとことが前提である。

図 2 の②サーバのリソース使用量推定のステップで、セクション 3.4 で述べる方法により作成済みの VM のモデルを利用して、仮に配置変更予定の VM を配置候補のサーバに配置変更したとき、そのサーバのリソース使用量の次の 1 週間の変化を推定する。こうして作成した配置候補のサーバの使用量から、最終的にどの配置変更候補のサーバを選べばいいのかを図 2 の③VM の配置変更先決定のステップで行う。どのサーバに VM を配置変更するかは、推定されたサーバのリソース使用量の確率分布の高負荷な領域をセクション 3.4 で述べる方法により比較することで決定する。このようにして配置変更予定の VM の配置変更先のサーバを決定して、VM の配置変更を行う。

3.3 VM のリソース使用量モデル化

3 週間分のリソース使用量を利用して、曜日 1 時間ごとの確率分布を推定する。ここで、D_{tk}を 10 分毎の VM の CPU 使用率とする。kはどの VM のリソース使用量なのかを表すインデックスとなっている。そして、(1)のように 3 週間分のデータを曜日と 1 時間ごとに分割する。

$$C_{ijk} = \{D_{tk} |\ \forall w \in week, \forall m \in minute\} \quad (1)$$
$$t = w * 7 * 24 * 6 + i * 24 * 6 + j * 6 + m$$

$w \in week = \{0,1,2\}$は何週間目のデータなのかを整数であらわしたインデックスになり、$i \in day = \{0,1,2,3,4,5,6\}$は何曜日なのかを示すインデックスで、$j \in hour = \{0,1,\ldots,23\}$はどの時刻なのかを示すインデックスになっている。$m \in minute = \{0,1,2,3,4,5\}$はそれぞれ 0 分、10 分,…,50 分を表すインデックスになっている。また$|C_{ijk}| = 18$となる。C_{ijk}はある 1 時間の時間帯のリソース使用量を過去 3 週間分まとめたものになっている。

ここで、利用する確率分布について説明を行う。曜日、時間帯によって、VM の CPU 使用率の分布は大きく異なる。そのような状況でも各分布に合致した確率分布を推定したい。また、各モデルに利用できるデータの数が少ないので、矩形分布だと多くの領域で確率が 0 となってしまうので、そのような領域の補間を行いたい。これらの理由によりカーネル密度推定を利用する。

分割した$|day| * |hour| = 168$ 個のC_{ijk}ごとにカーネル密度推定により(2)のような確率分布を VM、曜日、時間帯ごとに作成を行う。

$$f_{ijk}(x) = \frac{1}{N} \sum_{x' \in C_{ijk}} \frac{1}{\sqrt{2\pi h^2}} \exp\left(-\frac{(x - x')^2}{2h^2}\right) \quad (2)$$

kはどの VM の確率分布なのかを示すインデックスになっている。式(2)の確率分布に従う確率変数をVM_{ijk}とする。パラメータhはデータに対する確率分布の近似度合で、hが大きいほど確率分布はなだらかになる。パラメータの決定にはグリッドサーチを利用した。また、確率分布がオーバーフィッテングすることを避けるために、交差検証を利用した。

3.4 サーバのリソース使用量推定

作成した曜日、時間帯ごとの VM のモデルからサーバの CPU 使用率の確率分布の推定を行う。サーバの CPU 使用率は、配置変更予定の VM のモデルとサーバ上で起動している VM のモデルの同じ曜日、時間帯の確率変数を、各々の VM が利用する CPU 数(vm_cpu_k)で重みづけして和を取り、サーバの CPU 数($server\_cpu_l$)で割ったものをサーバのリソース使用量の確率変数$server_{ijl}$とする。

$$server_{ijl} = \frac{1}{server\_cpu_l} \left(\sum_{k \in S_l} vm\_cpu_k * VM_{ijk} \right) \quad (3)$$

ここでS_lはインデックスが$l \in server_set = \{0,1,2,4\}$のサーバ上で起動している VM のインデックスと、配置変更予定の VM のインデックスを要素として持つ集合になっている。$server_{ijl}$は、インデックスlのサーバのある曜日、時刻のリソース使用量の確率変数となっている。式(3)は確率変数の和による確率変数となっており、そのような確率変数が従う確率分布は、和を取った各確率変数の畳み込み積分の形式で解析的に計算することができる。しかし、サーバのリソース使用量の確率分布の計算量がサーバ上の VM の台数に対して、指数的に増大してしまう。そこで、現実的な時間で計算を終了させるために、Markov chain Monte Carlo methods (MCMC、マルコフ連鎖モンテカルロ法)を利用する。S_lに含まれるインデックスの VM に対する式(2)の確率分布からサンプリングをして、式(3)を計算することによって、インデックスl のサーバのリソース使用量の確率分布の近似を行う。

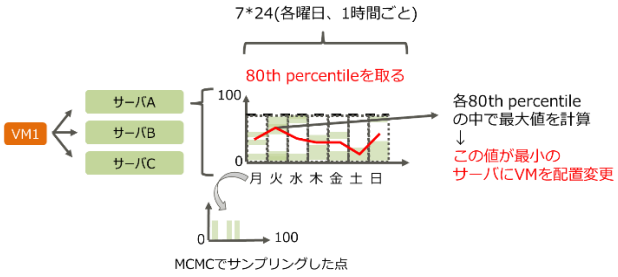

図 3　推定したサーバのリソース使用量のから配置変更先を決定する方法を示す図。パーセンタイルと最大値を利用して決定。

3.5 VM の配置変更先の決定

配置変更先の候補となる全サーバの各曜日、1 時間ごとのリソース使用量の確率分布からサンプリングが完了したら、どのサーバに VM を配置変更すべきか決定するために評価を行う。評価方法の概要は図 3 となっている。式(3)により計算されたある曜日、時間のサーバl の N 個のサンプリング点をソートしたものが式(4)となる。サーバのリソース使用量の外れ値の影響を受けないようにするため、パーセンタイル値を利用することで、高負荷の度合いを評価する。具体的には式(5)により各曜日、1 時間ごとのサーバの確率分布からサンプリングした点の集合に対してパーセンタイル値を取ることによって評価する。その後、式(6)により各曜日、1 時間ごとのパーセンタイル値 7*24 個の点の中で最大値を取る。この値を最終的なサーバの評価結果とし、式(7)によりこの値が最小になるサーバdestを求めて、配置変更予定の VM の配置変更先とする。

$$\{s_{ijln} | s_{ijln+1} \geq s_{ijln}, 0 \leq n < N\} \quad (4)$$

$$\text{percentile}_{ijl}(p) = s_{ijlm}, m = \left\lceil \frac{p}{100} * N \right\rceil \quad (5)$$

$$high_l = \max_{i \in day, j \in hour} \text{percentile}_{ijl}(p) \quad (6)$$

$$dest = \underset{l \in server\_set}{\text{argmin}} \; high_l \quad (7)$$

4. 評価

4.1 評価の概要

比較手法に対する有効性と提案手法の配置変更先を決定するときのパラメータに安定的であり、最適化が不要なことを確認するための評価実験を行う。

実際に動作する VM の CPU 使用率を用いたシミュレーションにより、複数の VM が稼働するサーバを複数台作成した。あるサーバでメンテナンスが行われ、メンテナンスが行われるサーバ上の VM が 1 台ずつ配置変更されるという状況を想定した。提案手法と比較手法で VM の配置変更先のサーバを決定し、配置後にそのサーバで発生する応答時間が悪化する期間を比較する。

4.2 評価に利用したデータ

評価用のデータとして、4 週間分の実際にパブリッククラウドで動作していた VM の CPU 使用率(10 分毎の平均)のログを利用した。全部で 2000 台分のデータが存在している。4 週間分のデータのうち、前半 3 週間分はモデル作成用のデータ、後半 1 週間分のデータは評価用のデータとして利用した。特に高負荷が発生している VM(モデル作成用の期間 3 週間で平均 CPU 使用率が高い順に 75 個の VM)のデータを評価のために利用する。

4.3 比較手法のモデルと配置変更先の決定方法

文献[2]の手法にならい、比較手法は平均によるモデルを利用する。概要は図 4 で示したものとなっており、同一時間軸上の VM のリソース使用量の平均を取ることによってモデルの作成を行う。VM のモデルは式(8)で表せる。VM の配置変更先は式(9)により作成したモデルから計算する。これにより、サーバリソース使用量の平均が最も低いサーバに dest に VM を配置変更する。

$$g_k(i,j,m) = \frac{1}{|week|} \sum_{w \in week} D_{tk} \quad (8)$$

$$t = w * 7 * 24 * 6 + i * 24 * 6 + j * 6 + m$$

$$dest = \underset{l \in server\_set}{\text{argmin}} \frac{1}{C_l} \sum_{k \in S_l, i \in day, j \in hour, m \in minute} g_k(i,j,m) \quad (9)$$

$$C_l = \text{server\_cpu}_l * |day| * |hour| * |minute|$$

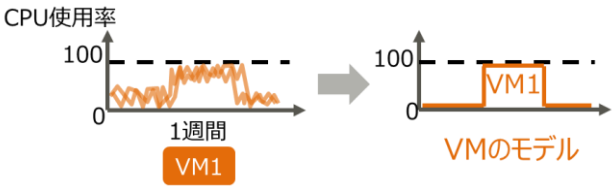

図 4　比較手法のモデルの作成法を表す図。比較手法のモデルは、同一時間軸上の平均を取ることによって作成。

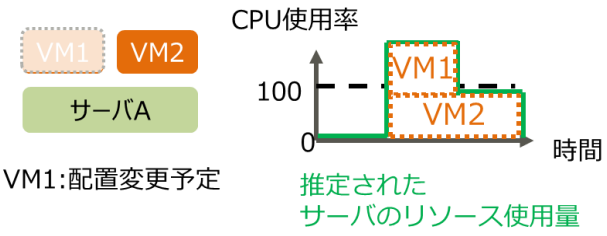

図 5 比較手法のサーバのリソース使用量の推定法を表す図。比較手法のサーバのリソース使用量は各 VM のモデルを CPU で重みづけして足し合わせることで推定。

4.4 シミュレーション

次に、シミュレーションの内容についての説明を行う。提案手法で説明したサーバのインデックスをl = {0,1,2,3}と設定する。4 台のサーバの物理 CPU の数はserver_cpu$_l$ = 8として共通の値を設定した。応答時間の悪化が発生するかどうかを評価するため、オーバーコミットの設定を想定

した。オーバーコミットの設定を行うことで、サーバの物理 CPU 数以上の CPU 数を VM に割り当てることが可能になる。例えばオーバーコミット率が 2 倍だと、サーバ上の VM が利用可能なサーバの物理 CPU の数が 2 倍になる。また、オーバーコミット率は全部で 2,3,4 の 3 種類の設定で行う。

様々な VM の配置の設定での比較を行うために、配置変更予定の VM として、6 種類の異なる VM を利用した。それぞれの配置変更予定 VM に対して、高負荷な 75 個の VM から、ランダムにサーバ上で起動する VM を選択することで、異なる 20 種類のサーバの状態を用意する。このとき、サーバ上の VM と配置変更予定の VM が利用する CPU 数の和が、オーバーコミットを考慮入れた、サーバの物理 CPU の数と等しくなるように選択する。

これにより、オーバーコミットの種類(3)、配置変更予定 VM の種類(6)、サーバ上に配置されている VM の種類(20)の組み合わせで全 360 通りの設定で、提案手法と比較手法の 2 つの方法での VM の配置変更先の決定、配置変更後の比較を行う。

VM の配置変更先のサーバを決定したら、選択したサーバの情報と、配置変更予定の VM の情報を利用して、式(10)のようにして、VM を配置した後の 1 週間のサーバのリソース使用量を計算する。ここの$t = 0$はモデルに利用した次の時間のインデックスを表す。

$$\mathrm{server\_cpu\_usage_{tl}} = \frac{1}{\mathrm{server\_cpu_l}}\left(\sum_{k\in S_l} \mathrm{vm\_cpu_k} \ * \mathrm{D_{tk}}\right) (10)$$

$$t = \mathrm{w} * 7 * 24 * 6 + \mathrm{i} * 24 * 6 + \mathrm{j} * 6 + \mathrm{m}$$

オーバーコミット率を 1 以上にしているため式(10)は 100%以上の値も取る。現実のサーバの CPU 使用率はオーバーコミットを利用していても 100%が上限なのでリソース使用率が 100%で飽和し、確実に各 VM の処理で応答時間の悪化が発生する。式(10)により計算した次の 1 週間の CPU 使用率が 100%を越えていると、実際に応答時間の悪化が発生すると考える。これを示した図が図 6 となっている。4 台のサーバすべてに対して上記が計算できるので、応答時間の悪化の発生期間が一番短いサーバを計算できる。このサーバを最適な配置変更先とする。そして、提案手法と比較手法が選択したサーバの次の 1 週間の CPU 使用率が 100%以上となる期間を比較する。この期間が提案手法の方が短くなれば、提案手法によって、応答時間の悪化が発生しづらいサーバに配置変更することができたとする。

実際の物理サーバでは、CPU 使用率が 100%以下であっても応答時間の悪化が発生する可能性がある。簡単な例だと、サーバの物理 CPU が 2 個存在し、利用 CPU 数が 1 の VM が 2 台サーバ上で動作している時を考える。このとき、物理サーバの CPU 利用率が 50%になったとする。物理サーバの CPU 使用率は各コアの使用率の平均を取ったものであるので、1 つの VM の CPU 使用率が 100%で、もう 1 つの VM の CPU 使用率が 0%、もしくは 2 つの VM とも CPU の使用率が 50%であるかもしれない。1 つ目の状態だと CPU 使用率が 100%になっている VM で応答時間の悪化が発生してしまうが、2 目の状態だとどちらの VM も応答時間の悪化が発生することはない。このようなとき、どれくらいの確率で応答時間の悪化が発生するかを考慮するのは、その時の各 VM の処理プロセスやハイパーバイザのスケジューリングを考慮する必要がありそれらすべてを考慮するのは難しい。そのため、今回の評価では 100%以下で発生する可能性のある応答時間の悪化については考慮しない。

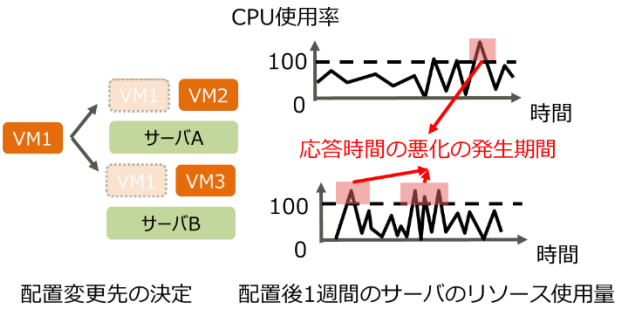

図 6　応答時間の悪化が発生した期間を示す図。VM の配置変更後 1 週間でサーバの CPU 使用率が 100%を越えたら応答時間の悪化が発生していると判定。

4.5 評価結果

CPU intel core i5 7500 3.4GHz 4 core で実行し、実行時間はサンプル数を N=2000 としたときに 710 秒となった。設定した 360 個の異なる状況で、それぞれ提案手法と比較手法の 2 つで VM の配置変更先の決定し、配置変更先のサーバでの応答時間の悪化が発生する期間を比較したものが図 7 のグラフとなる。1 つの点は 1 つの VM の配置状況における VM の配置変更先の応答時間の悪化の発生期間を表し、全部で 360 点存在している。VM の横軸は、比較手法により VM の配置変更先を決定した時に、配置変更先のサーバで、応答時間の悪化が発生した期間になっている。縦軸は、提案手法により VM の配置変更先を決定した時に、配置変更先のサーバで応答時間の悪化が発生した期間になっている。そのため、対角線上に存在する点は 2 つの選択方法で選んだサーバの応答時間の悪化の発生期間が同じことを意味している。これは同じサーバを選ぶ、もしくはたまたま VM の配置変更先の応答時間の悪化の発生期間が同じ異なるサーバを選んだ場合に相当する。このような点は図 7 の 360 個の中で 145 個存在している。グラフの対角線に対して上にくる点は、つまり図 7 の水色の領域は、提案手法に

より応答時間の悪化の発生期間が短いサーバを選択できたことを意味している。このような点は図 7 の 360 個の中で、155 個存在している。グラフの対角線に対して下にくる点は提案手法により応答時間の悪化の発生期間が短いサーバを選択できなかったことを意味している。このような点は図 7 の 360 個点の中で 60 個存在している。これより、図 7 でパーセンタイル$p = 80$を利用して評価した場合、対角線の上にくる点の数が、対角線の下にくる点の数よりも多いので、提案手法の方が多くの VM の配置状況で、応答時間の悪化が発生しづらいサーバに VM を配置変更可能であると考える。

また、提案手法のほうが適切なサーバを選択できている 155 個の中でも特に、縦軸上に存在する点が 52 個存在している。この点は、比較手法により VM の配置変更先を決定すると、応答時間の悪化が発生するが、提案手法により VM の配置変更先を決定すると、応答時間の悪化の発生期間が 0、つまり応答時間の悪化が発生しないサーバを選択できたということを意味している。

次に提案手法のモデルの評価方法を変更した場合の比較を行い、パーセンタイルpの値を$p = 99$としても図 8 のようになる。これより、提案手法はパラメータに対して安定的であることを確認できる。

ここで、最適な配置変更先との比較を行う。図 9,10 は最適な配置変更先に比較となっている。図 9 から、提案手法による配置変更で抑制した応答時間の悪化期間が最適な配置変更先を基準にしたとき、比較手法に対して改善していることを確認できる。

図 10 から、提案手法が、360 パターンの中で 6 割が最適な配置変更先に配置できていることを示している。これにより、提案手法による配置変更が比較手法より適した配置変更先を選択できていることを確認できる。

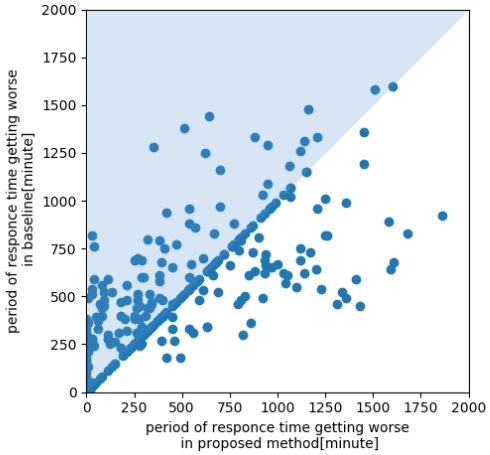

図 7　提案手法(24*7 個の確率分布の 80 パーセンタイル値を取り、その中の最大値を評価値と利用)と比較手法で VM の配置変更先として、決定したサーバで応答時間が発生した期間を比較するグラフ。

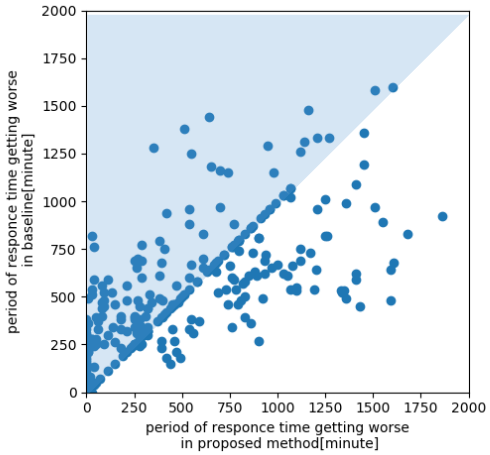

図 8　提案手法のモデルの確率分布のパーセンタイル値の取り方を変更した時の比較のグラフ(99 パーセンタイル値を利用)。

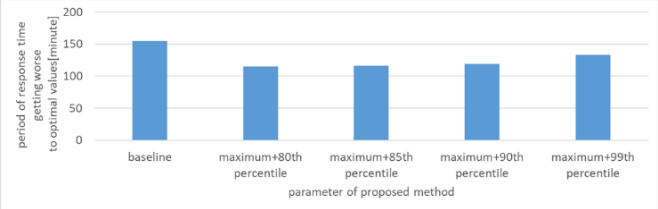

図 9 最適な配置で発生する応答時間の悪化の発生期間に対する差の 360 パターンで平均したグラフ。低いほど応答時間の悪化の発生期間が抑制されることを示す。

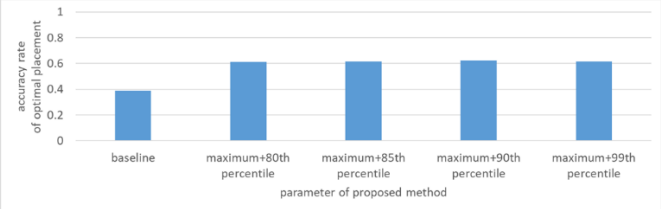

図 10　360 個のパターンの中で、実際に最適な配置できた割合を示した図。高い程、実際に応答時間の悪化が一番抑制されるサーバに VM を配置できたことを示す。

5. まとめ

本研究では、各曜日、時間帯ごとの VM のリソース使用量の確率分布を考えることにより、VM のリソース使用量の時間変化を表すモデルを作成する方法を提案した。また、作成したモデルからサーバのリソースの使用量を推定する方法と VM の配置変更先の決定方法までを含めた提案を行った。

実際に動作していた VM の CPU 使用率のログを利用してシミュレーションを行うことにより様々な配置状況での評価を行うことにより、比較手法に対する有効性と提案手法の配置変更先を決定するときのパラメータに安定的であり、パラメータ最適化が不要なことを確認した。

今後の課題として、今回は CPU 使用率のみに対して評価を行ったので、ネットワークトラフィックやディスク I/O などに、同時に適用可能にする必要がある。

参考文献

[1] Q. Zhang, L. Cheng, amd R . Boutaba, “Cloud computing: state-of-the-art and research challenges,” Journal of Internet Services and Applications, Vol. 1, no. 1, pp. 7–18, May. 2010.

[2] R. N. Calheiros, E. Masoumi, R. Ranjan and R. Buyya, “Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications’ QoS,” IEEE Transactions on Cloud Computing, Vol. 3, no. 4, pp. 449-458, Oct. 2015.

[3] A. Gulati, A. Holler, M. Ji, G. Shanmuganathan, C. Waldspurger, X. Zhu, “VMware Distributed Resource Management: Design, Implementation and Lessons Learned,” VMware Academic Program, Mar. 2012.

[4] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, “Live migration of virtual machines,” in Proc. the 2nd conference on Symposium on Networked Systems Design & Implementation, Vol. 2, pp. 273-286, May. 2005.

[5] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, “Sandpiper: Black-box and gray-box resource management for virtual machines,” Computer Networks, Vol. 53, no. 17, pp. 2923-2938, Dec. 2009.

[6] T. Tachibana, K. Kogiso and K. Sugimoto, “Dynamic Management of Computing and Network Resources with PID Control in Optical Grid Networks,” 2008 IEEE International Conference on Communications, pp. 396-400, May. 2008.

[7] X. Ren, R. Lin and H. Zou, “A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast,” 2011 IEEE International Conference on Cloud Computing and Intelligence Systems, pp. 220-224. Sep, 2011.

[8] A. Khan, X. Yan, S. Tao and N. Anerousis, “Workload characterization and prediction in the cloud: A multiple time series approach,” 2012 IEEE Network Operations and Management Symposium, pp. 1287-1294, Apr. 2012.

[9] W. Iqbal, M.N. Dailey, D. Carrera, P. Janecek, “Adaptive resource provisioning for read intensive multi-tier applications in the cloud,” Future Generation Computer Systems, Vol. 27, no. 6, pp. 871-879, Jun. 2011.