時系列予測モデルの学習時間及び計算資源削減手法の検討

高橋佑里子† 鈴木 成人†† 田原 司睦†† 小口 正人†

† お茶の水女子大学 〒 112–8610 東京都文京区大塚 2 丁目 1 － 1
†† 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 丁目 1 － 1
E-mail: †{yuriko-t,oguchi}@ogl.is.ocha.ac.jp, ††{shigeto.suzuki,tabaru}@fujitsu.com

あらまし 仮想環境において，計算資源のオーバーコミットに由来する仮想マシン (Virtual Machine: VM) の性能低下を防ぐことを目的として，VM の CPU 使用率を予測し，その結果に基づいて制御を行う技術が知られている．VM とそこで実行されるアプリケーションは時々刻々と変化するため，環境の変化に合わせて予測モデルを継続的に学習し，モデルを更新することで予測精度を担保する．ここで，大規模仮想環境における予測モデルの継続的な学習には多大な学習時間及び計算資源を必要するため，適応が困難であった．そこで本研究では，環境の変化を監視・評価し，精度担保に必要と判断された場合のみ予測モデルの学習・更新を行うことで，学習時間及び計算資源の削減を実現する手法について検討する．

キーワード 時系列データ，機械学習，仮想環境

1 はじめに

近年のクラウドサービスにおいて物理サーバ (Physical Machine: PM) の CPU 使用率は低く，そのパフォーマンスを十分に発揮できない状態が続いている [1]．これを改善すべく，事業者では，サーバを仮想化することで使用率を向上させ，PM 数を削減する取り組みが行われている．この取り組みでは，PM が自身の CPU 資源を超えた CPU を割り当てられるオーバーコミット状態に陥ることで，仮想マシン (Virtual Machine: VM) の性能が低下する可能性がある [2]．これを防ぐことを目的として，VM の CPU 使用率を予測し，その結果に基づいて図 1 のように制御を行う技術が知られている [3]．

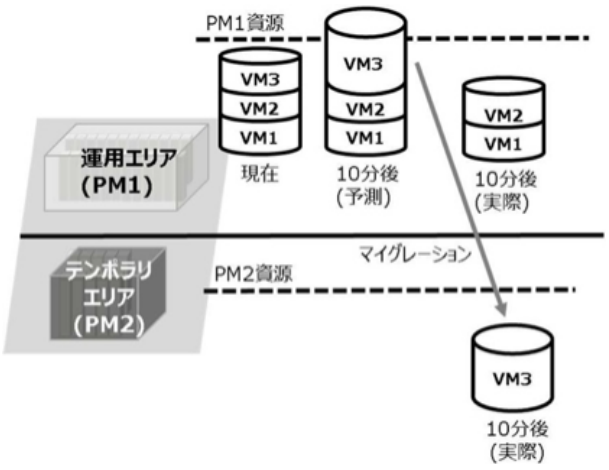

図 1 VM 制御のイメージ

VM とそこで実行されるアプリケーションは図 2 のように時々刻々と変化するため，環境の変化に合わせて予測モデルを継続的に学習し，モデルを更新することで予測精度を担保する．しかし，大規模仮想環境における予測モデルの継続的な学習には多大な学習時間及び計算資源を必要するため，適応が困難であった．

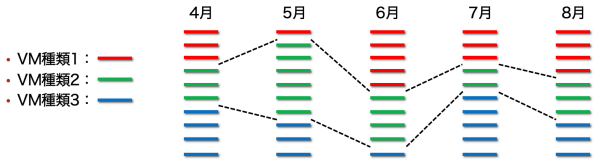

図 2 傾向変化のイメージ

そこで本研究では，環境の変化を監視・評価し，精度担保に必要と判断された場合のみ予測モデルの学習・更新を行うことで，学習時間及び計算資源の削減を実現する手法の検討を行う．過去のデータとそれを基に生成されたモデルがあることを前提に，直近のデータと過去のデータの類似度を算出し，学習を行うか否かを判断することで，ある程度の精度を担保しながら学習時間を大幅に削減できることを確認した．

2 関連研究

クラウドサービスにおける VM の CPU 使用率の予測や制御に関する研究は，以前から多く行われている．[4] では，ラックユニットの消費電力を分析し，API 連携を用いて VM の管理ソフトウェアと連携したシステムの提案を行っている．仮想データセンタを想定して計算したところ，実利用エリアの稼働率は常に 90%以内に収まることや，フットプリントを 40 %削減できること，コンピュータルームエアコンとサーバを合わせた消費電力も 6.8%削減できることが示されている．また，[5] では，エネルギー消費量に基づいて，リアルタイムに VM を統合するフレームワークの提案を行っている．提案されたフレームワークを使用することで，フレームワークを使用していないデータセンタと比較して最大 80%の改善が示され，PM の高使用率と省エネルギーの実現に成功している．

3 提案手法

提案手法のフローチャートを図 3 に示す．

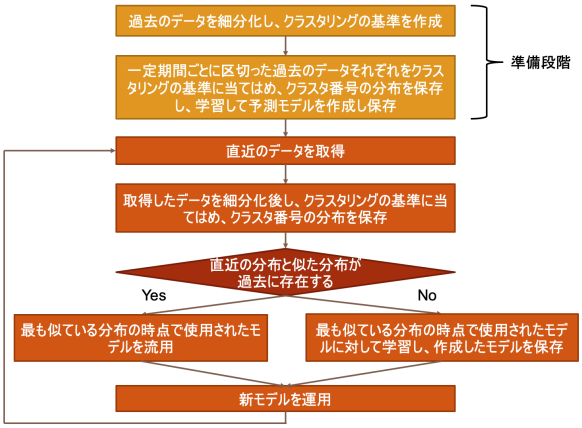

図 3　提案手法のフローチャート

準備として，過去のデータを細分化し，クラスタリングの基準の作成を行う．次に，事前に数種類の傾向それぞれに対してクラスタリングの基準を当てはめ，クラスタ番号の分布を保存し，学習を行い予測モデルを作成し保存する．運用段階では，最初に直近のデータを取得し，細分化を行った後，準備段階で作成した基準に当てはめ，得られたクラスタ番号の分布を，過去の複数の分布と比較する．もし，過去に似た分布があった場合，学習を行う必要がないため，その時点で使用されたモデルを流用する．一方，過去に似た分布がなかった場合，最も似た分布の時点で使用されていたモデルに対して学習を行い，作成したモデルを保存して使用する．この一連の流れを一定間隔で繰り返すというものである．

次に，フローチャート内で使用している言葉が本研究内で示している内容を，使用している技術と合わせて説明する．

3.1 細　分　化

本研究では，図 4 のように，データを 1 点ずつずらしながら学習元データ数と正解データ数の合計値ごとに区切る処理を細分化と呼ぶ．また，本研究では，学習元データ数を 200，正解データ数を 1 と設定しているため，201 点ごとに区切る処理を行っていることになる．

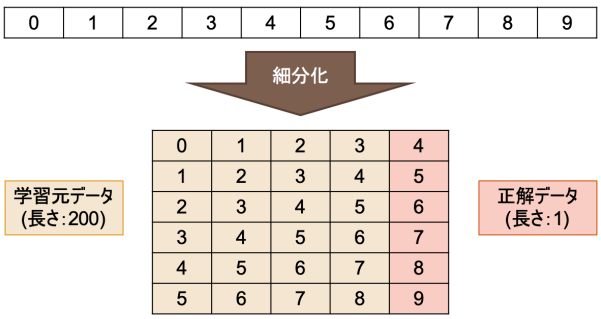

図 4　細分化のイメージ

このような処理を行う理由は，時系列データの機械学習で使用するデータの特徴にある．時系列データの学習を行う場合は通常，図 5 のように，学習元データ数と正解データ数を設定し，データを開始点を 1 点ずつずらしながら，長さ学習元データ数の学習元データとそれに対応する直後の長さ正解データ数の正解データのペアを作成し，学習させる．つまり，学習には長い波形をそのまま使うというわけではなく，細かい波形をいくつも使っているということになる．そこで本研究では，この特徴を踏まえ，長い時系列データを多数の細かい時系列データに変形することで，データの特徴を適切に判断できるのではないかと考えた．

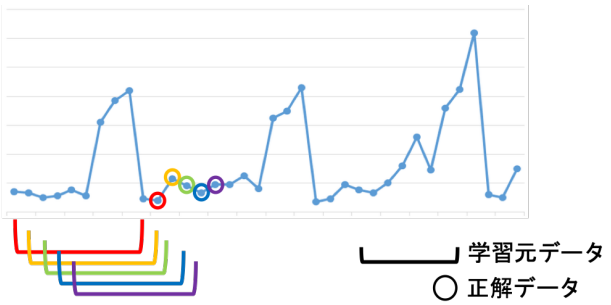

図 5　時系列データ学習の特徴

3.2 クラスタリングの基準

クラスタリングには，scikit-learn [6] の中にある sklearn.cluster.Kmeans クラスを使用した．このクラスの fit メソッドを使うことで，与えられたデータとクラスタ数を元にクラスタリングの基準を作成することができ，predict メソッドを使用することでその基準に基づいてクラスタ番号を割り振ることができる．本研究では，準備段階において，過去のデータを細分化したものを元にクラスタ数を 30 としてクラスタリングの基準を作成し，運用段階においてもその基準を使い続けることとする．

3.3 分布の類似度

分布の類似度の基準は，細分化後のデータをクラスタリングの基準に当てはめた際のクラスタ番号分布のコサイン類似度とした．コサイン類似度とは，ベクトルの内積を用いて類似度を計算する方法である．数値はベクトル同士の成す角度の近さを表現しており，-1 から 1 までの値をとるが，数値が大きくなるほど類似度が高いということになる．以下のような式で計算される．

$$cos(\vec{q},\vec{d}) = \frac{\vec{q}\cdot\vec{d}}{|\vec{q}|\cdot|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\cdot\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

例として，図 6 の 2 つのクラスタ番号分布のコサイン類似度は約 0.96 である．目視では，全体的に似てはいるが若干異なった分布だという印象を受ける．次元数が 30 と大きいため，コサイン類似度の数値は全体的に大きくなりやすい．

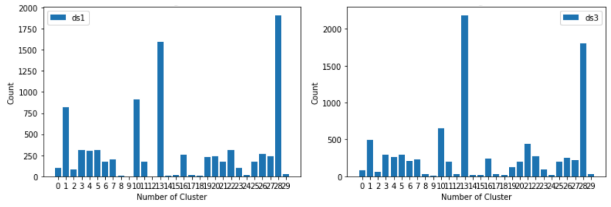

図 6　分 布 の 例

3.4　学　　習

本研究では，計算量を軽微にする目的で GRU(Gated Recurrent Unit) を使用し，GRU を 2 層繋げた深層学習ネットワークを構築した．GRU は，再帰型構造を持つニューラルネットワークである RNN(Recurrent Neural Network) の一種で，RNN を長期依存を扱うことができるよう改良した LSTM(Long Short-Term Memory) の忘却ゲートと入力ゲートを更新ゲートとして一つに統合することで，計算量が比較的少なくなるよう改良したものである．GRU の構造を図 7 に示す．

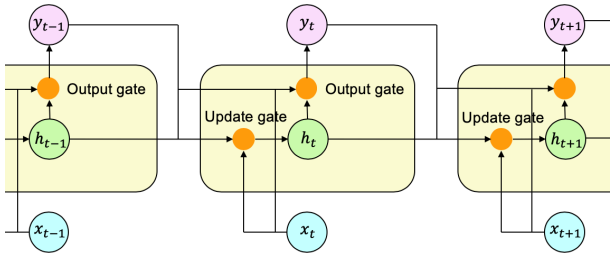

図 7　GRU の構造

ライブラリは，TFLearn [7] を使用した．TFLearn は，Tensorflow [8] 向けの高レベル API で，これを使用することで簡潔なコードで深層学習モデルを記述することでき，実験を迅速に行うことが可能になる．

また，モデルの評価指標として RMSE(Root Mean Squared Error) を使用した．RMSE は，二乗平均平方根誤差と呼ばれる回帰モデルの誤差を評価する指標の 1 つである．この値が小さく 0 に近いほど精度が良いということになる．長さ n の時系列データの正解値を y_t，予測値を $\tilde{y}_t$ とすると，以下のような式で計算される．

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \tilde{y}_t)^2}$$

4　実　　験

4.1　概　　要

本研究では，Microsoft 社が提供している Azure の VM トレースデータセット [9] [10] の一部の，”avg cpu”項目を使用した．このデータセットは，1 点が 5 分間隔となっている．まず，異なる傾向を持つデータセットを作成する目的で，データセットを正規化した後 360 点 (30 時間分に相当) に細分化し，段階的に k-means 法クラスタリングを行うことで 100 種類の波形を抽出した．ここでクラスタリングを用いたのは，抽出する 100 種類をなるべくユニークなものにするためである．抽出した 100 種類の波形の例を図 8 に示す．これら 100 種類の波形を異なる割合で含む傾向を乱数を用いて 30 種類作成し，準備として 10 種類の傾向で学習モデルを作成し，残り 20 種類の傾向を運用段階とした．大規模な環境を想定しているため，VM 数は 10000 とした．

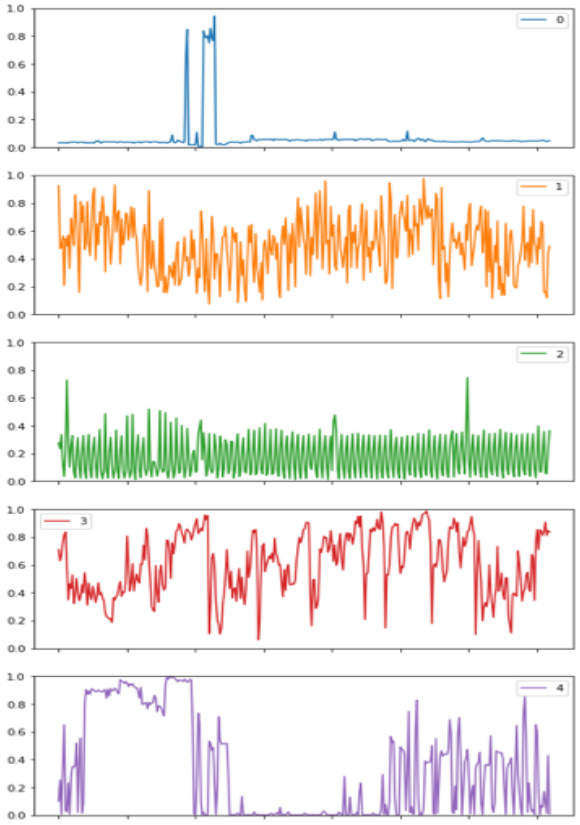

図 8　抽出した波形の例

提案手法における「過去の分布と似た分布が過去に存在する」の基準は，過去の分布とのコサイン類似度の最高値がある値以上と定めることとした．本実験では，基準値が 0.95 の場合と 0.97 の場合の 2 種類で実験を行った．提案手法での学習回数は，基準値を 0.95 とした場合は 5 回，基準値を 0.97 とした場合は 13 回となり，毎回学習を行う場合の 20 回と比較すると，いずれも学習時間が減少した．

そして，提案手法以外に，継続学習を行わずに 1 つのモデルを使用し続けるパターン 10 種類・各区間で毎回学習を行ったパターン・1 つ前の値を繰り返しただけのデータ (以下，repeat 値と呼ぶ) との比較も行った．

4.2　結　　果

上記のような条件で実験を行ったところ，結果は図 9 のようになった．縦軸は 10000 個のデータの RMSE の平均値，横軸は区間番号である．なお，それぞれの区間は独立しているため，このグラフにおいて隣接する区間の時系列における関係性はない．

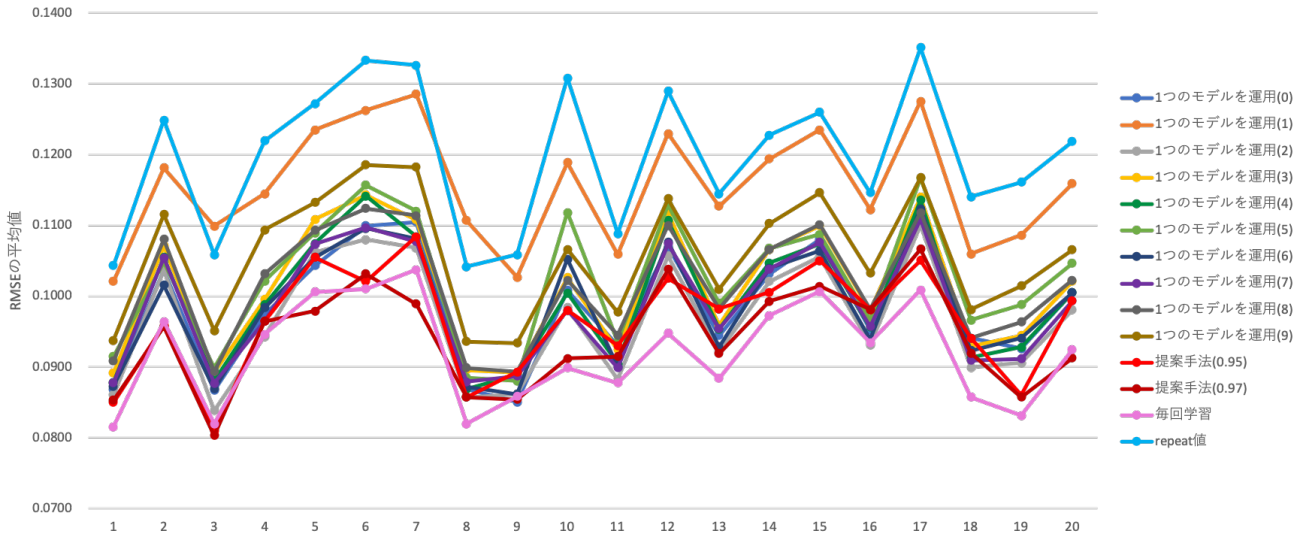

図 9　実 験 結 果

4.3　考　　察

まず，後追いをしているだけの repeat 値 (水色) の結果がほとんどの区間で最も悪いことから，学習を行ったことで予測モデルがうまく機能しているということが読み取れる．また，毎回学習を行った場合 (ピンク色) が最も精度が良いことも読み取れる．

提案手法 2 種類 (赤色) の結果は，repeat 値や 1 つのモデルを使用し続けるパターン 10 種類と比較して，おおよその区間で精度が上回っている．提案手法の 2 種類の結果を比較すると，基準値が 0.97 の結果が基準値 0.95 の結果を上回っていることが多い．中には，1 つのモデルを使用し続けるパターンに劣っている区間もあるが，細分化したデータのクラスタ番号の分布の類似度の情報から，精度が良くなるであろうモデルを適切に選択できていると考えられる．学習時間と予測精度はトレードオフの関係にあることを踏まえると，提案手法の基準値を 0.95 とした場合が最も効率的なモデル運用ができていると考えられる．

5　まとめと今後の予定

蓄積された過去のデータと予測モデルを活用し，精度担保に必要と判断された場合のみ予測モデルの学習・更新を行うことで，学習時間及び計算資源の削減を実現する手法の検討を行った．その結果，時系列データを細分化したものを同一のクラスタリング基準に当てはめ，得られたクラスタ番号分布の類似度を元に学習を行うかを判断することで，学習時間を大幅に削減しながら予測モデルの精度を維持することが可能であることが確認できた．

今後は，時系列予測モデルの効率的な運用に向けて，より良いフローチャートを作成したり，ニューラルネットワークのアーキテクチャを最適化する手法の検討に取り組みたいと考えている．

謝　　辞

本研究の一部はお茶の水女子大学と富士通研究所との共同研究契約に基づくものであり，JST CREST JPMJCR1503 の支援を受けたものである．

文　　献

[1] Josh Whitney and Pierre Delforge. Data center efficiency assessment. *Issue paper on NRDC (The Natural Resource Defense Council)*, 2014.

[2] Rahul Ghosh and Vijay K Naik. Biting off safely more than you can chew: Predictive analytics for resource over-commit in iaas cloud. In *2012 IEEE Fifth International Conference on Cloud Computing*, pages 25–32. IEEE, 2012.

[3] 児玉宏喜, 鈴木成人, 福田裕幸, 吉田英司, et al. マイグレーションを利用したデータセンタの高効率運用手法の提案とオーバコミット時における vm の性能評価. 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 *(OS)*, 2018(13):1–7, 2018.

[4] Hiroyoshi Kodama, Hiroshi Endo, Shigeto Suzuki, and Hiroyuki Fukuda. High efficiency cloud data center management system using live migration. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, pages 733–738. IEEE, 2017.

[5] Salam Ismaeel and Ali Miri. Real-time energy-conserving vm-provisioning framework for cloud-data centers. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0765–0771. IEEE, 2019.

[6] scikit-learn — machine learning in python. `https://scikit-learn.org/`.

[7] Tflearn — tensorflow deep learning library. `http://tflearn.org/`.

[8] Tensorflow. `https://www.tensorflow.org/`.

[9] Mohammad Shahrad, Rodrigo Fonseca, Íñigo Goiri, Gohar Chaudhry, Paul Batum, Jason Cooke, Eduardo Laureano, Colby Tresness, Mark Russinovich, and Ricardo Bianchini. Serverless in the wild: Characterizing and optimizing the serverless workload at a large cloud provider. *arXiv preprint arXiv:2003.03423*, 2020.

[10] Azure/azurepublicdataset: Microsoft azure traces. `https://github.com/Azure/AzurePublicDataset`.