一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
NS2018-66 (2018-07)

クラウドシステムにおける適応型制御間隔の検討とフラッシュクラウドでの評価

小川祐紀雄† 長谷川 剛†† 村田 正幸†††

† 室蘭工業大学 情報メディア教育センター 〒 050–8585 北海道室蘭市水元町 27-1
†† 大阪大学 サイバーメディアセンター 〒 560–0043 大阪府豊中市待兼山町 1–32
††† 大阪大学 大学院情報科学研究科 〒 565–0871 大阪府吹田市山田丘 1–5
E-mail: †y-ogawa@mmm.muroran-it.ac.jp, ††hasegawa@cmc.osaka-u.ac.jp, †††murata@ist.osaka-u.ac.jp

あらまし クラウドシステムにおいて計算資源の利用効率を向上するためには、リクエスト到着レートの変動に応じて動的に計算資源をスケーリングすることが必要である。計算資源の利用効率とスケーリングの制御間隔はトレードオフの関係にあるが、最適な制御間隔はリクエスト到着レートの変動特性に依存する。フラッシュクラウドのようなリクエスト到着レートの急激な変動に対応するために、本報告では、システムの制御間隔を可変にすることで必要時に動的に制御間隔を小さくしつつ、全体としてシステム構成を過剰に変更することなく余剰計算資源を一定に保つことを実現する。まず、リクエスト到着レートの変動量と予測誤差が制御間隔の幅に収まるように制御間隔の定式化を行い、ついで、実システムのトレースデータを用いて評価を行う。これにより、フラッシュクラウドのような急激なリクエスト到着レートの変動に対して提案方式が有効であることを示す。
キーワード クラウドシステム、自動スケーリング、制御間隔、フラッシュクラウド

A Study on Adaptive Control Interval in Cloud Systems and an Evaluation of Flash Crowd Events

Yukio OGAWA†, Go HASEGAWA††, and Masayuki MURATA†††

† Center for Multimedia Aided Education, Muroran Institute of Technology
†† Cybermedia Center, Osaka University
††† Graduate School of Information Science and Technology, Osaka University
E-mail: †y-ogawa@mmm.muroran-it.ac.jp, ††hasegawa@cmc.osaka-u.ac.jp, †††murata@ist.osaka-u.ac.jp

Abstract Adaptive resource scaling is necessary for improving utilization in cloud systems. There is a trade-off relationship between utilization of computing resources and the control interval of them, and the optimal control interval depends on the changing behavior of arrival requests. In this report, we make the length of control interval changeable and adaptive so as not to increase the total number of reconfiguration while keeping a constant amount of the excess resources, even when a flash crowd event occurs in cloud systems. We first formulate dynamic control intervals to which changes in request arrival rate and their prediction errors are fitted. We then conducted numerical simulations with real-world arrival traces including flash crowd events. The results indicate that our proposed approach is effective for accommodating a sudden spike in request arrivals.
Key words cloud system, auto-scaling, control interval, flash crowd

1. はじめに

クラウドコンピューティングを特徴づける重要な概念の一つに弾力性 (elasticity、処理要求量に応じて動的に計算資源を拡大縮小すること) がある [1]。計算資源を「弾力的に」配備することで、ピーク値に対して適切な計算資源を配備しつつ、計算資源の全体的な利用率も高めることができる。ただし、この計算資源のスケーリングのためにはトラヒックの変動特性および構成変更の負荷を考慮し適切な時間間隔で制御を行うことが必要になる。本報告では、リクエスト到着レートの変動に適応す

るよう構成変更の時間間隔を動的に変更することで、計算資源の最適利用を実現することを目的とする。

クラウドコンピューティングにおける弾力性の実現を目的とした計算資源の自動スケーリング手法は広く研究されてきた [2]。自動スケーリング手法は、処理負荷の変化を検知した後に制御を行うリアクティブな手法と、処理負荷の変化を予測して事前に制御するプロアクティブな手法に分けられれ [3]、変化を検知してから制御するまでの間に発生する計算資源の過不足を低減するためには、プロアクティブな制御が必要になる。プロアクティブに制御を行うためには、将来の処理負荷が既知である必要があるが、対話型処理を扱うアプリケーションシステムでは事前の把握は難しい。そこで、処理負荷、例えばリクエスト到着レートや CPU 利用率を、ARIMA(自己回帰和分移動平均) モデルや Holt-Winters 手法により予測することが幅広く行なわれている [2]。さらに、予測精度の向上を目指して、パターンマッチングとマルコフ連鎖の組み合わせによる手法 [4]、フラクタルモデルによる手法 [5]、サポートベクターマシンとニューラルネットワークなどの機械学習による手法 [6]、脳の仕組みに着想を得た機械学習による手法 [7] などが提案されている。しかしながら、予測精度は処理負荷の特性、トレーニングデータの選び方、また学習期間などに依存し、予測誤差をなくすことは難しい。そこで本報告では、処理負荷の予測において誤差が生じることを前提とし、予測誤差を制御に逐次反映していく方法を検討する。

クラウドシステムにおける計算資源の構成変更間隔は、当初は課金周期に合わせて時間単位であったと考えられるが、オンデマンドでの仮想サーバのプロビジョニングは数分程度で完了すること [8]、また、商用クラウドが 1 秒単位で課金されるようになったこと [9] を考慮すると、今後、数 10 分単位から数分単位での変更間隔にまで短縮されていく可能性がある。このような短周期の制御は、計算資源の利用率を向上させる一方で、構成変更のオーバーヘッドを発生させる。そこで、計算資源の利用率と構成変更オーバーヘッドのバランスを取るために、計算資源の制御にモデル予測制御 [10] を適用することが提案されている。この提案による自動スケーリングでは、制御ステップごとに計算資源やサービスレベルに関するコストと構成変更にかかるコストをトータルで最小化するように構成変更の実施を決定する。例えば、サービスレベルの違反や構成変更のオーバヘッドを抑えながら、処理リクエストの変動に応じてサーバを動的に調整する手法 [11, 12]、複数サービスを稼働させる資源プールのサイズを調整する手法 [13]、物理サーバおよび仮想サーバのスケーリングを調整する手法 [14, 15] が提案されている。これらの手法により、決められた間隔で制御を行いながら計算資源の過剰な構成変更を防ぐことが示されている。しかしながら、制御間隔が適切かどうかという議論は行われていない。

例えば、フラッシュクラウド [16] と呼ばれる短時間のうちにアクセス数が急増する現象において、計算資源拡張に必要な時間間隔よりアクセス変動の時間間隔が短ければ、計算資源の拡張がアクセスの増加に追いつかない。このような場合は、計算資源の拡張に加えて計算資源の制御時間を短くすることが必要になる。想定されるアクセス変動の最大値に合わせて計算資源の制御時間間隔を常に最短に設定しておくことは可能であるが、特定のイベントの有無により時間帯ごとのアクセス数の差が非常に大きいシステムでは、固定的な制御時間の設定は効率的ではない恐れがある。

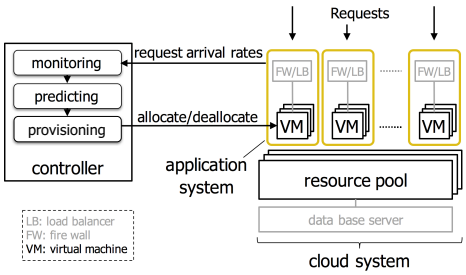


図 1　クラウドシステムの構成概要

そこで、本報告では、クラウドコンピューティングにおけるさらなる弾力的な資源制御のために、アプリケーションシステムへのリクエスト到着レートの増減に合わせて、計算資源を増減させると同時に制御の緩急を調整することを検討する。具体的には、まず、対話型処理を扱うクラウド上のアプリケーションシステムを対象とした自動スケーリングにおいて、制御間隔ごとにリクエスト到着レートの予測を行い、仮想サーバの台数を増減させる。このとき、リクエスト到着レートの変動量と予測誤差が制御間隔の幅に収まるように制御間隔の定式化を行う。ついで、フラッシュクラウドの例を含む実システムのトレースデータを用いて評価を行い、従来の固定的な時間間隔での制御に比較して、全体での構成変更回数を削減しつつ余剰計算資源量を保つことができることを示す。

以下、2. 章において制御間隔に着目したクラウドシステム最適化に関する定式化を提案し、3. 章で評価を行う。最後に、4. 章で本報告のまとめと今後の課題を述べる.

2.　クラウドシステムの制御モデル

2.1　システム構成

本報告が対象とするシステム構成を図 1 に示す。基盤サービス (Infrastructure as a Service) プロバイダでは、資源プールから切り出した仮想サーバ、ファイアウォール、ロードバランサなどからアプリケーションシステムを構成する。アプリケーションシステムの管理者は、コントローラを用いてリクエスト到着レートを監視して将来のリクエスト到着レートを予測し、さらに、予測値に基づき必要な仮想サーバ台数を算出してスケールアウト・インの制御を行う。コントローラは、リクエスト到着レートの監視は一定間隔ごとに行い、仮想サーバの構成変更は必要時にのみ行う。なお、本報告では、仮想サーバ以外の機器は台数を変更せずに利用する前提とする。

2.2　適応型制御間隔の定式化

コントローラが一定間隔で観測するアプリケーションシステムへのリクエスト到着レートについて、時刻 t において観測されたリクエスト到着レートを λ_t とする。一方、仮想サーバのスケールアウト・イン制御は、ある可変の時間幅で区切られた時

間 (タイムスロットと呼ぶ) ごとに行う。i 番目のタイムスロットでは k_i 個 ($k_i \in \mathbb{Z}^+$) のリクエスト到着レートを観測した後に構成変更を行う。また、この k_i 個のリクエスト到着レートの集合を S_i と表記する。なお、本報告では、この観測個数 k_i を i 番目のタイムスロットの幅と呼ぶ。

本報告ではクラウドシステムの制御の目的を、制御期間全体を通じてアプリケーションシステムの余剰資源を一定以下に保ちながら、構成変更回数を最小化することと定める。この制御期間全体での k_i の平均値を $\bar{k_i}$ と表記する。また、i 番目のタイムスロットにおいて k_i 個のリクエスト到着レートの最大値と平均値を、それぞれ m_i、a_i と表記する。仮想サーバ 1 台の処理能力は、どの仮想サーバでも同一であるとし、その値を μ とする。つねにリクエスト到着レート以上の処理能力を提供するように仮想サーバを配備すると定めると、i 番目のタイムスロットにおいて必要な仮想サーバ台数は $\lceil {}^{m_i}/_{\mu} \rceil$ となる。以上の表記を用いて、最適化問題を以下に定式化する。

$$\text{minimize} \quad \frac{1}{\bar{k_i}} \tag{1}$$

$$\text{subject to} \quad \forall i, \ \left\lceil \frac{m_i}{\mu} \right\rceil \mu - a_i \leqq d\mu \tag{2}$$

$$m_i = \max_{\lambda_t \in S_i} \lambda_t \tag{3}$$

$$a_i = \frac{1}{k_i} \sum_{\lambda_t \in S_i} \lambda_t \tag{4}$$

目的関数 (1) は、単位時間あたりの平均構成変更回数 (平均タイムスロット数) を最小化することを示す。制約 (2) は、各タイムスロットにおける平均の余剰処理能力が、仮想サーバ d 台 (d は正の定数) の処理能力以下であることを示す。制約 (3) および制約 (4) は、m_i と a_i が、それぞれ、i 番目のタイムスロットで観測したリクエスト到着レートの最大値と平均値であることを示す。

仮想サーバの制御は、リクエスト到着レートの予測値に基づいて行う。つまり、i 番目のタイムスロットにおけるリクエスト到着レートの最大値 m_i と平均値 a_i は、一つ前のタイムスロットにおいて予測した値である。それぞれの予測値を $\hat{m}_i$、$\hat{a}_i$ とし、それぞれの予測誤差は、$N(0, \sigma_{m_i})$、$N(0, \sigma^2_{a_i})$ の確率分布にしたがうと仮定すると、リクエスト到着レートの最大値 m_i と平均値 a_i は次の確率分布で表すことができる。

$$m_i \sim N(\hat{m}_i, \sigma^2_{m_i}) \tag{5}$$

$$a_i \sim N(\hat{a}_i, \sigma^2_{a_i}) \tag{6}$$

制約 (2) において、$m_i \gg \mu$ と仮定し、$\lceil {}^{m_i}/_{\mu} \rceil \mu \sim m_i$ と近似する。i 番目のタイムスロットの平均の余剰処理能力の近似値を r_i とおき、その確率分布を、式 (5) と式 (6) が互いに独立な分布であると仮定して、次の式で表す。

$$r_i = m_i - a_i \sim N(\hat{m}_i - \hat{a}_i, \sigma^2_{m_i} + \sigma^2_{a_i}) \tag{7}$$

式 (7) において、$\hat{m}_i - \hat{a}_i$ は i 番目のタイムスロットにおけるリクエスト到着レートの変動に起因し、$\sigma^2_{m_i} + \sigma^2_{a_i}$ は予測誤差に起因する。リクエスト到着レートが一定でない限りは、タイムスロット幅 k_i が小さくなるほど、リクエスト到着レートの変動を包含する処理能力の余剰は小さくなる。一方、リクエスト到着レートの予測誤差を包含する処理能力の余剰は、予測精度に依存する。予測精度とタイムスロット幅 k_i の関係は、それぞれのアプリケーションシステムのリクエスト到着レートの変動特性による。

式 (7) を利用して、次式で表すように、目的関数 (1) を、それぞれのタイムスロットにおいて平均余剰処理能力 r_i が $d\mu$ 以下となる確率を一定値 (C とする) 以上にする k_i のうち、その最大値を見つけることに帰着させる。

$$\forall i, \ k_i = \arg\max_{k_i} \{r_i | P(r_i \leqq d\mu) \geqq C\} \tag{8}$$

$P(r_i \leqq d\mu)$ は、平均余剰処理能力 r_i が d 台の仮想サーバの処理性能 $d\mu$ 以下となる確率を表す。これは、予測誤差の影響が小さければ、タイムスロット幅 k_i が小さくなるほど大きくなる。

2.3 リクエスト到着レートの予測

i 番目のタイムスロットにおけるリクエスト到着レートの最大値と平均値は、一つ前のタイムスロット において予測した値である。こられらの予測値 $\hat{m}_i$、$\hat{a}_i$ の算出のために、本報告では、季節変動 (周期性) を考慮した ARIMA モデル [17] を適用する。以下では、x_j は、m_i または a_i のいずれかを表す。なお、x_j は、タイムスロット幅を一定値 (m_i または a_i のタイムスロット幅) に固定したときの時系列データであり、混乱を避けるために i を j と置き直している。ラグ演算子 $L(Lx_j = x_{j-1})$ を導入し、x_j に対し d 階差分演算と D 階季節差分演算を行って定常化させた時系列 y_j を定める。

$$y_j = (1-L)^d (1-L^s)^D x_j \tag{9}$$

s は周期変動の期間である。系列 y_j は、過去のデータと誤差の多項式として次のように表される

$$y_j = \sum_{v=1}^{p} \phi_v L^v y_j + (1 + \sum_{w=1}^{q} \theta_w L^w)\epsilon_j \tag{10}$$

上式において、ϕ_v、θ_w は係数である。ϵ_i は、$\epsilon_j \sim N(0, \sigma^2)$ に従う誤差項である。この分散 σ^2 は、式 (5) の分散 $\sigma^2_{m_i}$ または式 (6) の分散 $\sigma^2_{a_i}$ に対応する。式 (10) の次数や係数は、過去の観測値に対し最尤推定および赤池情報量基準（AIC）を適用することにより求める。σ は、モデルを当てはめた推定値と観測値の残差より求める。1 期先の予測値は、式 (10) において j を $j+1$ と置き換え、誤差項予測値 $\hat{\epsilon}_{j+1} = 0$ とし、観測値および誤差を代入して計算する。

リクエスト到着レートは、一般にレートが大きくなるほどばらつきが大きくなることが多い。そこで、ばらつきを抑えリクエスト到着レートの分布を正規分布に近づけるために次の Box-Cox 変換 [17] を行った上で、式 (9) と式 (10) の各パラメータを定める。

$$B(x_j) = \begin{cases} b^{-1}(x_j^b - 1) & (b \neq 0) \\ \log x_j & (b = 0) \end{cases} \tag{11}$$

この変換は線形変換ではないため、式 (8) において、

$$P(r_i \leqq d\mu) = P(a_i + d\mu - m_i \geqq 0) \qquad (12)$$

と置き直し、さらに次式のように変換してタイムスロット幅 k_i の最大値を求める。

$$P(B(a_i + d\mu) - B(m_i) \geqq 0) \qquad (13)$$

式 (13) は、リクエスト到着レート平均値 a_i の時系列の代わりに、$a_i + d\mu$ の時系列の予測を行うことを示している。

3. 評　　価

3.1 評 価 手 法

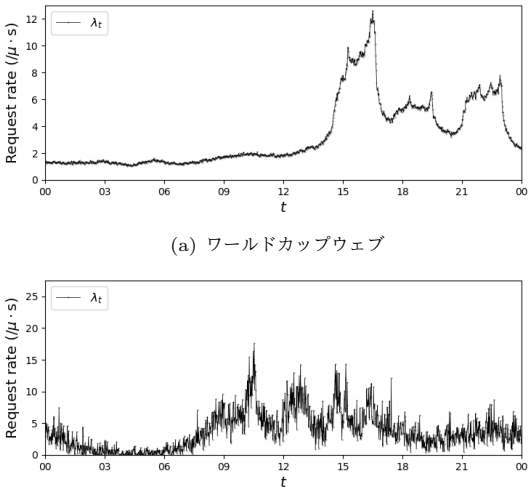
式 (1)〜 (4) の最適化問題を解くためには、タイムスロットごとに式 (8) を満たすタイムスロット幅 k_i を導出する必要がある。タイムスロット幅 k_i を減少させた際に $P(r_i \leqq d\mu) - C$ が単調増加する場合には、二分法により $P(r_i \leqq d\mu) - C = 0$ となる k_i を導出した。また、タイムスロット幅 k_i の減少に対して $P(r_i \leqq d\mu) - C$ が探索区間内で上に凸、かつ、最大値が 0 以下である場合には、探索区間を複数に分割し、代表点の値が最大となる区間について代表点の近傍の k_i を求めた。

3.2 評価用データセットと予測手法

評価では、1998 年ワールドカップウェブサイトへの HTTP リクエストログ [18](1998 年 5 月 20 日〜1998 年 6 月 20 日) と、3 万人規模の大学における大学ウェブサイトへの HTTP リクエストログ (2014 年 4 月 16 日〜2014 年 5 月 17 日) からなる、二つ実アプリケーションシステムのそれぞれ約 1 ヶ月間のトレースデータを用いた。以降では、前者のシステムおよびデータをワールドカップウェブと呼び、後者のシステムおよびデータをキャンパスウェブをよ呼ぶ。ワールドカップウェブは、インターネット上の不特定多数の利用者向けのシステムの例であり、フラッシュクラウドのトレースを含んでいる。一方、キャンパスウェブは、ある特定の組織に属するシステムの例である。なお、各アプリケーションシステムを構成する仮想サーバの処理能力 (μ) には、リクエスト到着レートの中央値を用いた。ワールドカップウェブでは $\mu = 202$ リクエスト/s であり、キャンパスウェブでは $\mu = 1.13$ リクエスト/s である。

各アプリケーションシステムの観測時間の間隔を 1 分とし、リクエスト到着レートの 1 分ごとの平均値をリクエスト到着レート観測値 (λ_t) とした。それぞれのデータついて、リクエスト到着レート観測値の 1 日分の例を図 2 に示す。ワールドカップウェブについては、15 時付近にフラッシュクラウドが見られる。ワールドカップウェブは、リクエスト到着レートが大きく、データの分単位の変動より時間単位の変動の方が顕著に現れている。一方、キャンパスウェブでは、10 時 30 分、12 時 30 分、15 時付近のピークに見られるような時間単位の変動と、分単位の変動が同時に確認できる。

リクエスト到着レートの予測においては、2. 3 節の方式にしたがい、k_i 個の観測値を集約したタイムスロットごとに、前 3 週間分のデータを用いて、Box-Cox 変換を行ったのち、式 (9) の D を 1 週間、d を 1 タイムスロットとおき (p, q) 値を求めた [17]。タイムスロット幅は、従来の制御時間である 1 時間 ($k_i = 60$) から、仮想サーバの最小起動時間に近い 2 分 ($k_i = 2$) まで変化させた。なお、以降においては、タイムスロット幅 (k_i) の減少と構成変更回数 ($1/k_i$) の増加を同義で用いている。



(a) ワールドカップウェブ

(b) キャンパスウェブ

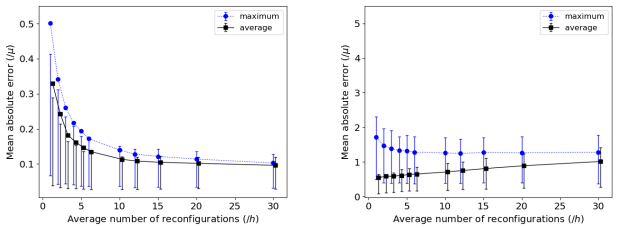
図 2　評価用トレースデータの例

3.3 固定長タイムスロットでの予測誤差

アプリケーションシステムごとの予測誤差特性を把握するために、まず、1 時間あたりの構成変更回数 ($1/\bar{k_i}$) を 1 回から 30 回まで増加させたときの予測値の絶対誤差を確認した（図 3)。ここでは、仮想サーバの処理能力 (μ) を単位として、各タイムスロットでリクエスト到着レート最大値を予測したときの誤差 $(m_i - \hat{m}_i)/\mu$ と、リクエスト到着レート平均値を予測したときの誤差 $(a_i - \hat{a}_i)/\mu$ について、全タイムスロットにおける平均値を示している。なお、エラーバーは、分布の第 1 四分位数から第 3 四分位数までを示す。

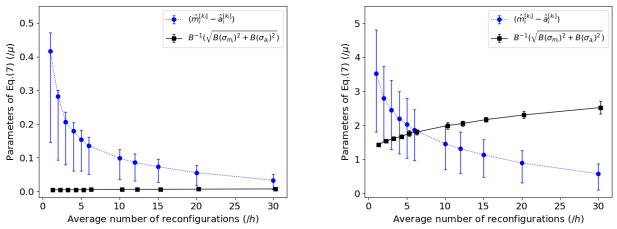
ワールドカップウェブでは、構成変更回数を増加させ、より小さなタイムスロット幅で制御を行う方が、リクエスト到着レート最大値、リクエスト到着レート平均値ともに予測誤差は小さくなる。これは、図 2(a) の 15 時におけるリクエスト到着レートの急増に起因している。このような場合、予測値は実測値より小さな値になるが、小さなタイムスロット幅で制御を行うことによって、予測値の実測値からの差分を小さく保つことができる。一方、キャンパスウェブでは、構成変更回数を増加させたとき、ワールドカップウェウのような予測誤差の減少はなく、リクエスト到着レート平均値の予測誤差が大きくなる。これは、タイムスロット幅を小さくすると図 2(b) に示される数分単位の変動の影響が顕著になり、これが予測誤差を増大させることにつながるからである。

次に、リクエスト到着レート最大値とリクエスト到着レート平均値の予測誤差が、式 (8) の正規分布のパラメータに与える影響について評価を行った (図 4)。これは、最適なタイムスロット幅は式 (8) を用いて算出するが、この式 (8) は式 (7) の確率分布に依存するからである。なお、$\sigma^2_{m_i} + \sigma^2_{a_i}$ については Box-Cox 変換先で求めた値を逆変換している。図より、リクエ



(a) ワールドカップウェブ　(b) キャンパスウェブ

図 3　リクエスト到着レート最大値と平均値の予測における誤差



(a) ワールドカップウェブ　(b) キャンパスウェブ

図 4　式 (7) のパラメータ評価

スト到着レート最大値の予測値とリクエスト到着レート平均値の予測値の差分 $\hat{m}_i - \hat{a}_i$ は、タイムスロット幅を小さくするにつれて減少することがわかる。これは、リクエスト到着レートの変動は、より小さな幅のタイムスロットで捉えた方が小さくなるからであり、この傾向はワールドカップウェブでもキャンパスウェブでも変わらない。一方、予測値算出の際の ARIMA モデル当てはめの残差である $B^{-1}(\sqrt{B(\sigma_{m_i})^2 + B(\sigma_{a_i})^2})$ については、ワールドカップウェブでは相対的に小さな値であるのに対し、キャンパスウェブでは大きな値となっている。これは、ワールドカップウェブの予測誤差 (図 3(a)) は時間単位程度の変動に起因しているのに対し、キャンパスウェブの予測誤差 (図 3(b)) は分単位の変動に起因しており、後者はタイムスロット幅が小さくなるにつれてより顕著になるからである。

3.4　可変長タイムスロットでの自動スケーリング

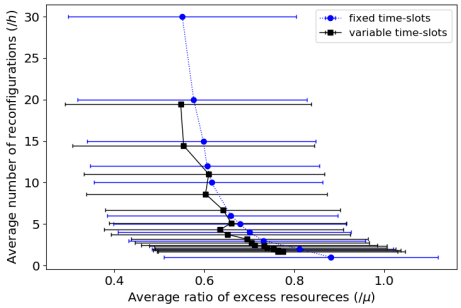
式 (8) において $C = 0.9$ に固定し、$d = 1$ から $d = 5$ まで 0.25 間隔で変化さたときの仮想サーバの平均余剰処理能力 ($\lceil \hat{m}_i/\mu \rceil \mu - a_i$ の平均値) と、1 時間あたりの仮想サーバの構成変更回数 ($1/\bar{k_i}$) を、図 5 に示す。なお、各測定ごとの分布を表すため平均値に第 1 四分位数から第 3 四分位数をエラーバーとして加えている。また、図 5 では、比較のためにタイムスロット $k_i = 60$ から $k_i = 2$ までの固定タイムスロット幅での評価結果を追記している。この固定タイムスロット幅での評価では、タイムスロットごとにリクエスト到着レート最大値の予測を行い $\lceil \hat{m}_i/\mu \rceil$ 台の仮想サーバを割り当てたときの結果である。

図 5 のグラフの右下の評価点が $d = 5$ に相当し、左上の評価点が $d = 1$ に相当する。d を小さくするほど余剰処理能力の許容値が小さくなるため、より小幅なタイムスロットで制御を行い、結果として構成変更回数が増加する。この曲線は、構成変更回数と余剰処理能力のパレートフロントに相当する。ワールドカップウェブ (図 5(a))、キャンパスウェブ (図 5(b)) ともに、可変長タイムスロットは、固定長タイムスロットの曲線を縦軸方向に 2/3 から 1/2 程度に縮めたような曲線になり、可変長タイムスロットを用いるとより少ない構成変更回数で制御できることを示している。このことを示す具体的な例を、ワールドカップウェブにおける結果を用いて示す。図 6 は、図 2(a) と同じ評価区間であり、リクエスト到着レート λ_t に加えて、配備した仮想サーバの処理能力 ($\lceil \hat{m}_i/\mu \rceil \mu$ と、処理能力の余剰 ($\lceil \hat{m}_i/\mu \rceil \mu - a_i$) のタイムスロットごとの変化を示している。タイムスロット幅を 1 時間に固定して制御を行うと (図 6(a))、15 時付近のフラッシュクラウドによる急増の際に、仮想サーバの過少割り当てとなり処理能力が不足する。一方、タイムスロット幅を 3 分に固定して制御を行うと (図 6(b))、15 時付近のフラッシュクラウドによる急増に追随し、仮想サーバの過少割り当てを抑えることができるが、0 時から 12 時までの区間では、リクエスト到着レートの変動の少ないにもかかわらず過剰に制御を繰り返している。これらに対し、タイムスロット幅を可変にして制御を行うと (図 6(c)、$d = 3$ とした例)、リクエスト到着レートの変動の少ない 0 時から 12 時までの区間では構成変更回数を抑えつつ、15 時付近以降の急増および急減時には数分間隔で制御を行っており、これにより、仮想サーバの過少割り当て期間を抑え、全体として構成変更回数の増大を防いでいることが示されている。

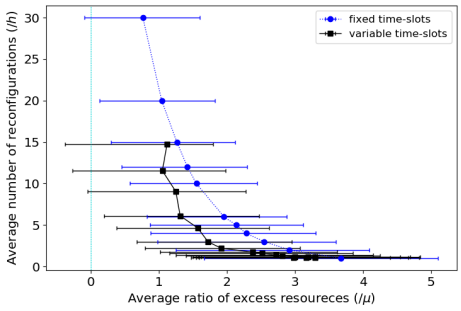
ただし、キャンパスウェブ (図 5(b)) では、固定長タイムスロットに比較して余剰処理能力のばらつきが大きく、d を小さくし余剰処理能力を抑えると処理能力不足になる割合が増加する。これは、タイムスロット幅を小さくすると予測誤差 (ARIMA モデル当てはめ残差である図 4(b) における $B^{-1}(\sqrt{B(\sigma_{m_i})^2 + B(\sigma_{a_i})^2})$) が大きくなることが影響している。その具体例を図 7 に示す。リクエスト到着レートの大きい 10 時および 12 時 30 分から 17 時付近では、タイムスロット幅が小さくなるが、その結果、予測誤差の影響がより大きくなり、仮想サーバ過少割り当てが何度も発生している。キャンパスウェブの場合は、予測誤差の影響を抑制するためには、タイムスロット幅を小さくし過ぎないことが必要になる。

4.　おわりに

本報告では、リクエスト到着レートの変動に適応して仮想サーバの台数と同時に制御間隔も変更することで、全体での構成変更回数を抑制しつつ余剰処理能力を一定に保つ手法の提案を行った。さらに、実システムのトレースデータを用いた評価では、従来の固定長タイムスロットでの制御に比較して、タイムスロット幅を可変とすることで、従来の 2/3 から 1/2 程度の構成変更回数で、従来同等の余剰処理能力が保たれることを示した。一方、リクエスト到着レートの分単位の変動の大きなアプリケーションシステムでは、タイムスロット幅を小さくすると予測誤差が大きくなり、その結果仮想サーバの過少割り当てが発生することを示した。今後の課題として、予測誤差の影響を抑えるタイムスロット幅の決定方法を示すことがあげられる。さらに、提案手法とモデル予測制御を組み合わせて制御回数をさらに抑制することも検討する。

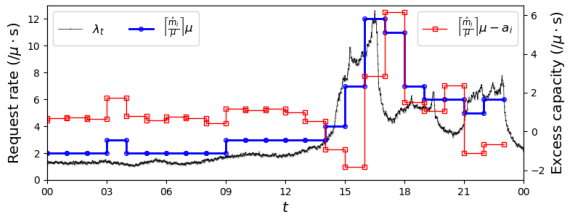


(a) ワールドカップウェブ

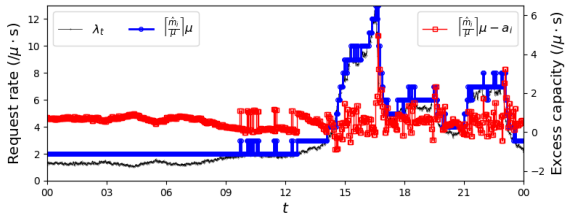


(b) キャンパスウェブ

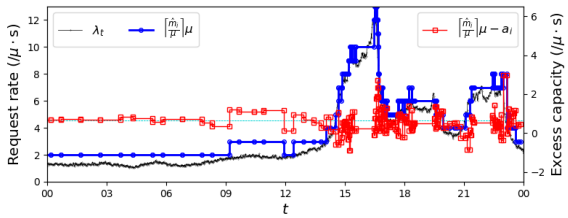
図 5　構成変更回数と余剰処理能力



(a) タイムスロット幅を固定したとき ($k_i = 60$)



(b) タイムスロット幅を固定したとき ($k_i = 3$)



(c) タイムスロット幅を可変にしたとき ($d = 3$)

図 6　ワールドカップウェブにおける制御例

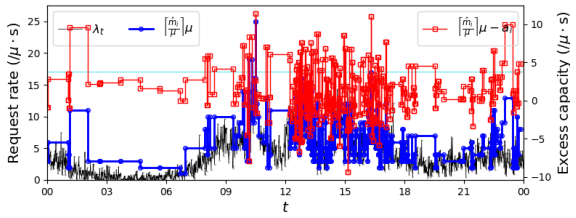


図 7　キャンパスウェブにおける可変長タイムスロットでの制御例

文　　献

[1] N.R. Herbst, S. Kounev, and R. Reussner, “Elasticity in cloud computing: What it is, and what it is not,” Proc of ICAC ’13, pp.23–27, June 2013.

[2] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, “Elasticity in cloud computing: State of the art and research challenges,” IEEE Transactions on Services Computing, vol.PP, no.99, pp.1–1, 2017.

[3] G. Galante and L.C.E.d. Bona, “A survey on cloud computing elasticity,” Proc of UCC ’12, pp.263–270, IEEE Computer Society, Nov. 2012.

[4] Z. Gong, X. Gu, and J. Wilkes, “Press: Predictive elastic resource scaling for cloud systems,” Proc of CNSM 2010, pp.9–16, Oct. 2010.

[5] M. Ghorbani, Y. Wang, Y. Xue, M. Pedram, and P. Bogdan, “Prediction and control of bursty cloud workloads: A fractal framework,” Proc of CODES+ISSS ’14, pp.1–9, Oct. 2014.

[6] A.Y. Nikravesh, S.A. Ajila, and C.-H. Lung, “Towards an autonomic auto-scaling prediction system for cloud resource provisioning,” Proc of SEAMS ’15, pp.35–45, SEAMS ’15, IEEE Press, Piscataway, NJ, USA, May 2015.

[7] Y. Cui, C. Surpur, S. Ahmad, and J. Hawkins, “Continuous online sequence learning with an unsupervised neural network model,” CoRR, vol.abs/1512.05463, pp.1–16, 2015. http://arxiv.org/abs/1512.05463

[8] M. Mao and M. Humphrey, “A performance study on the vm startup time in the cloud,” Proc of IEEE CLOUD ’12, pp.423–430, June 2012.

[9] Amazon Web Services, Inc., “Amazon EC2 Pricing,” `http://aws.amazon.com/ec2/pricing/`. accessed Mar. 27, 2018.

[10] J.M. Maciejowski, Predictive control: with constraints, Pearson education, 2002.

[11] N. Roy, A. Dubey, and A. Gokhale, “Efficient autoscaling in the cloud using predictive models for workload forecasting,” Proc. of IEEE CLOUD, pp.500–507, July 2011.

[12] T. Lu, M. Chen, and L.L.H. Andrew, “Simple and effective dynamic provisioning for power-proportional data centers,” IEEE Transactions on Parallel and Distributed Systems, vol.24, no.6, pp.1161–1171, June 2013.

[13] H. Ghanbari, M. Litoiu, P. Pawluk, and C. Barna, “Replica placement in cloud through simple stochastic model predictive control,” Proc of IEEE CLOUD ’14, pp.80–87, June 2014.

[14] D. Kusic, N. Kandasamy, and G. Jiang, “Combined power and performance management of virtualized computing environments serving session-based workloads,” IEEE Transactions on Network and Service Management, vol.8, no.3, pp.245–258, Sept. 2011.

[15] M. Gaggero and L. Caviglione, “Predictive control for energy-aware consolidation in cloud datacenters,” IEEE Transactions on Control Systems Technology, vol.24, no.2, pp.461–474, March 2016.

[16] J. Jung, B. Krishnamurthy, and M. Rabinovich, “Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites,” Proc of WWW ’02, pp.293–304, 2002.

[17] R.J. Hyndman and G. Athanasopoulos, “Forecasting: principles and practice,” `https://www.otexts.org/book/fpp`. accessed May 10, 2015.

[18] The Internet Traffic Archive, “1998 world cup web site access logs,” `http://ita.ee.lbl.gov/html/contrib/WorldCup.html`. accessed Nov. 19, 2014.