

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

室蘭工業大学
情報メディア教育センター

長谷川剛

大阪大学
サイバーメディアセンター

村田正幸

大阪大学
大学院情報科学研究科

1

2

研究の背景と目的

■背景

- 弾力性 (elasticity)
 - クラウドコンピューティングの重要な概念の一つ
 - 処理負荷に応じて動的に計算資源を拡大縮小すること
 - ピーク値時に必要な計算資源の配備と計算資源の利用率向上を両立
- トラフィック変動特性や構成変更の負荷を考慮した制御が必要

■目的

- 処理負荷の変動に適応して、制御パラメータを自律的に変更していく制御方式の実現

3

目次

- ▶ 研究の背景と目的
- ▶ クラウドシステムの制御モデル
- ▶ 評価
- ▶ まとめと今後の課題

従来研究と課題

■弾力性実現に向けた計算資源の自動スケーリング手法は広く研究されている[1]

- 二つのアプローチ：負荷変動に対してリアクティブ/プロアクティブな手法
- 変動検知から制御完了までにSLA違反が発生するときはプロアクティブな手法

■プロアクティブな手法

- 将来の処理負荷（リクエスト到着レート、CPU利用率）を予測する必要
 - ARIMA(自己回帰和分移動平均)モデルやHolt-Winters手法他の適用
 - 予測精度は負荷特性、トレーニングデータ、学習期間などに依存
 - 学習の困難なフラッシュクラウドへの対応が難しく、SLA違反が発生

[1] Y. Al-Dhuraihi, 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.

4

本研究のアプローチ

■ターゲット：

急激な負荷変動（フラッシュクラウド）に適応性のあるプロアクティブな自動スケーリング

- 負荷変動に対してSLA違反を減らすため制御間隔を自律的に調整
- 処理負荷の増減にあわせて制御間隔を変更
 - 負荷変動が早いときに、制御時間間隔を短く（制御回数を多く）

5

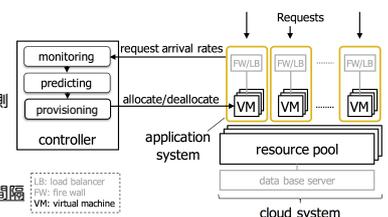
制御モデル - システム構成

■基盤サービス (IaaS) 上のアプリケーションシステムを想定

- 資源プールから仮想サーバを切り出して利用

■コントローラの機能

- (1)リクエスト到着レートを監視
- (2)将来のリクエスト到着レートを予測
- (3)必要な仮想サーバ台数を算出しスケールアウト・イン
- リクエスト到着レートの監視は一定間隔
- 仮想サーバの構成変更は必要時にのみ



6

制御モデル

- 適応型制御間隔の定式化

■一定時間ごとにリクエスト到着レート監視、時刻 t のリクエスト到着レート: λ_t

■可変時間幅の区間 (タイムスロット) ごとに制御

・ i 番目のタイムスロットの幅は k_i 個の監視をまとめた幅とする

■目的: 仮想サーバの構成変更回数を最小化

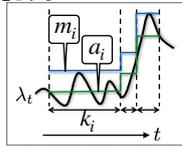
$$\text{minimize } \frac{1}{k_i} \left\{ \begin{array}{l} \text{平均のタイム} \\ \text{スロット幅} \end{array} \right.$$

■制約: 各タイムスロットの余剰資源を一定値以下に保つ

・タイムスロット i でリクエスト到着レートの最大値 m_i を処理できるよう仮想サーバ配備

・リクエスト到着レート最大値 m_i と平均値 a_i の差を仮想サーバ d 台の処理性能以下に

$$\text{subject to } \forall i, \left| \frac{m_i}{\mu} - a_i \right| \leq d\mu \left\{ \begin{array}{l} \text{仮想サーバ}d\text{台} \\ \text{の処理性能} \end{array} \right.$$



制御モデル

- リクエスト到着レートの予測

■リクエスト到着レート最大値予測値 \hat{m}_i と平均値予測値 \hat{a}_i

・一つ前のタイムスロットで予測した値

・予測には、季節変動 (周期性) を考慮した ARIMA (自己回帰和分移動平均) モデルを利用

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

リクエスト到着レートの最大値 m_i または平均値 a_i を定常化させた値
p次自己回帰過程
q次移動平均過程

9

評価用データ

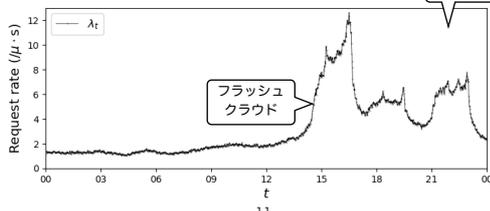
- (1) ワールドカップウェブ

■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

・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 [2]

・1998年5月20日~1998年6月20日

・仮想サーバの処理能力 (μ): 202リクエスト/s



[2] The Internet Traffic Archive, "1998 world cup web site access logs," <http://ita.ee.tsl.gov/html/contrib/WorldCup.html>, accessed Nov. 19, 2014.

制御モデル

- 適応型制御間隔の定式化 (続き)

■リクエスト到着レート最大値 m_i と平均値 a_i は、予測に基づき算出

・予測値の確率分布 $m_i \sim N(\hat{m}_i, \sigma_{m_i}^2)$, $a_i \sim N(\hat{a}_i, \sigma_{a_i}^2)$

・リクエスト到着レートの最大値と平均値の差分の分布を近似

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

リクエスト到着レートの変動量に依存
予測誤差の大きさに依存

■目的関数を近似してタイムスロット i の幅 k_i を算出

・タイムスロット i の平均余剰処理能力 r_i が $d\mu$ 以下となる確率を一定値 C 以上にする k_i の最大値を見つける

$$\forall i, k_i = \arg \max_k \{ r_i | P(r_i \leq d\mu) \geq C \}$$

8

評価条件・手法

■評価条件

・アプリケーションシステムの観測時間の間隔: 1分

・リクエスト到着レートの1分ごとの平均値を、観測値 λ_i に利用

・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利用

■評価手法

・前3週間の観測値からリクエスト到着レート最大値 \hat{m}_i 平均値 \hat{a}_i を予測

・二分法を用いてタイムスロット i の幅 k_i を算出

・タイムスロット幅 k_i は、最大60分から最小2分まで減少させる

(1時間あたりの構成変更回数 $\frac{1}{k_i}$ を1回から30回まで増加させる)

10

評価用データ

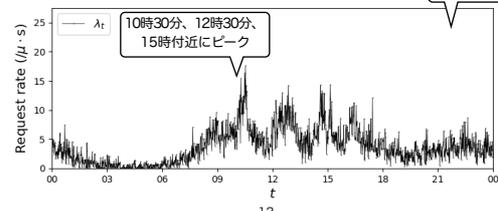
- (2) キャンパスウェブ

■特定の組織に属するシステムの例

・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

・仮想サーバの処理能力 (μ): 1.13リクエスト/s



12

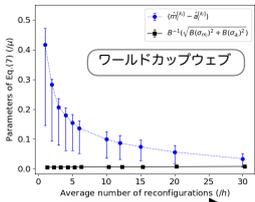
予備評価の結果

- 固定幅タイムスロットでの予測差分分布

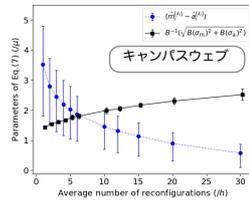
- リクエスト到着レート最大値 m_i と平均値 a_i の予測の差分分布の特性を調査

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

- ・ 最大値と平均値の差 ($\hat{m}_i - \hat{a}_i$) は、タイムスロット幅減少に伴い減少
- ・ 予測の偏差量 ($B^{-1}(\sqrt{B(\sigma_m)^2} + B(\sigma_a)^2)$) はデータの特性に依存



構成変更回数：1回/時→30回/時



13

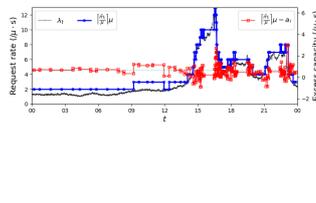
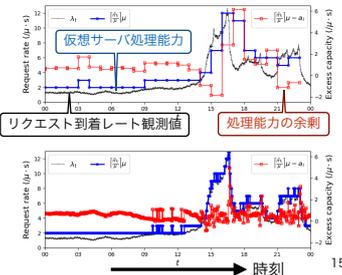
評価結果の具体例 1

- 可変幅タイムスロットによる構成変更回数削減

- ワールドカップの例：フラッシュクラウドへのタイムスロット幅の適応

タイムスロット幅を固定 (上：1時間、下：3分)

タイムスロット幅を可変に ($d = 3$)



15

まとめと今後の課題

■ まとめ

- ・ 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、自律的に制御間隔も変更し、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を提案
- ・ 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
- ・ タイムスロット幅を小さくすると予測誤差が大きくなる特性をもつアプリケーションシステムでは、仮想サーバの過少割り当てが発生

■ 今後の課題

- ・ 予測誤差の影響を抑えるタイムスロット幅の決定方法

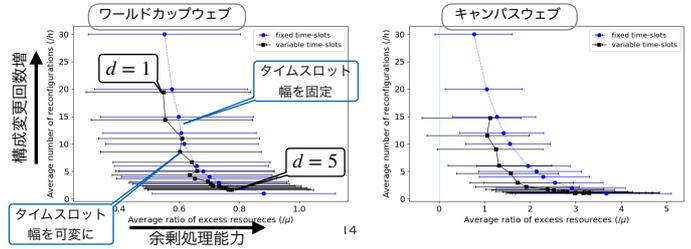
17

評価結果

- 可変幅タイムスロットでの自動スケーリング

- 余剰処理性能しきい値を仮想サーバ5台分 ($d = 5$) から1台分 ($d = 1$) まで変化させ、余剰処理能力 $[\frac{\hat{m}}{P}] \mu - a_i$ と構成変更回数 $\frac{1}{k_i}$ を算出

1. 可変幅タイムスロットは、固定幅タイムスロットより構成変更回数が2/3~1/2に減少
2. 可変長タイムスロットは、予測偏差により余剰処理能力のばらつき大(キャンパスウェブ)



14

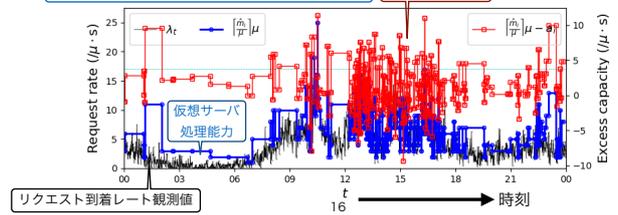
評価結果の具体例 2

- 可変幅タイムスロットによる余剰処理能力のばらつき

- キャンパスウェブの例

- ・ タイムスロット幅が小さくなると予測誤差が増加する
- 仮想サーバの過少割り当て (マイナスの余剰) が発生

タイムスロット幅を可変に ($d = 1.25$)



Q & A

ご静聴ありがとうございました

18