Linear Quadratic Regulator for Resource-Efficient Cloud Services

Youngsuk Park[1], Kanak Mahadik[2], Ryan A. Rossi[2], Gang Wu[2], Handong Zhao[2]
[1]Stanford University [2]Adobe Research

1 PROBLEM AND MOTIVATION

The run-time performance of modern applications deployed within containers in the cloud critically depends on the amount of provisioned resources. Provisioning fewer resources can result in performance degradation and costly SLA violations; while allocating more resources leads to wasted money and poor resource utilization. Moreover, these applications undergo striking variations in load patterns. To automatically adapt resources in response to changes in load, an autoscaler applies predefined heuristics to add or remove resources allocated for an application based on usage thresholds. However, it is extremely challenging to configure thresholds and scaling parameters in an application- agnostic manner without deep workload analysis. Poor resource efficiency results in high operating costs and energy expenditures for all cloud-based enterprises.

To improve resource efficiency over safe and hand-tuned autoscaling schemes reinforcement learning (RL) based approaches [3, 4] learn an optimal scaling actions through experience (trial-and-error) for every application state, based on the input workload, or other variables. After the learning agent executes an action, it receives a response from the environment, based on the usefulness of the action. The agent is inclined to execute actions that provide higher rewards, thus reinforcing better actions. However, these proposed approaches suffer from the *curse of dimensionality* [2]. The state and action space to be discretized grows exponentially with the number of state variables, leading to scalability problems, manifested in unacceptable execution times in updation and selection of the next action to be executed in online setting. Moreover, these methods require huge amount of of samples to learn and thus often lack stability and interpretability of policy.

2 OUR APPROACH

To deal with this problem, we provide a formulation of model-based RL for Linear Quadratic Regulator (LQR) in the context of cloud resource allocation. The class of LQR [1] is known for having a linear optimal policy and a quadratic optimal value function both of which are simple enough to interpret and deploy. Moreover, these can be efficiently computed at scale in continuous state and action space, unlike previous discrete RL strategies such as Q-learning.

Let $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, $w_t \in \mathbb{R}^n$ be a state, an input (or action), and some random noise at time t. At every t, the system suffers a stage-cost $g : \mathbb{R}^n \times \mathbb{R}^m$ and transits based on a transition dynamic $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ when input u_t is taken in state x_t.

Infinite horizon LQR: Among the class of LQR, we adopt trajectory tracking LQR. For an augmented state $x_t = [\bar{x}_t^T, 1]^T$, our goal is to find the optimal stationary policy $\pi : \mathbb{R}^n \to \mathbb{R}^m$ that solves

$$\min \sum_{t=0}^{\infty} \gamma^t \mathbb{E}[g(x_t, u_t)] \quad \text{subject to } x_{t+1} = f(x_t, u_t, w_t), u_t = \pi(x_t).$$

SoCC '19, November 20–23, 2019, Santa Cruz, CA, USA
2019. ACM ISBN 978-1-4503-6973-2/19/11.
https://doi.org/10.1145/3357223.3366028

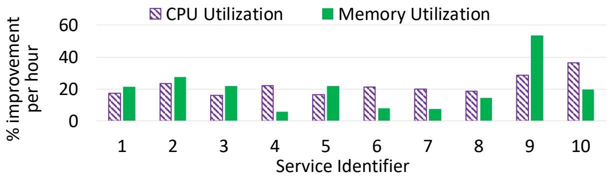

Figure 1: % improvement using LQR over autoscaler in CPU and memory utilization per hour for 10 Adobe services. Metrics measured at 5-minute intervals for a period of 30 days (15 Aug - 13 Sept 2019)

with some discounted factor $\gamma \in [0, 1]$. Here, the stage cost is quadratic and dynamic is linear, i.e.,

$$g(x_t, u_t) = x_t^T Q x_t + u_t^T R u_t, \quad f(x_t, u_t, w_t) = Ax_t + Bu_t + w_t$$

where $Q, A \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times m}$, and $w_t \sim \mathcal{N}(\mu, W)$.

We cast the problem of adaptive resource allocation in the above setting, where each (unaugmented) state $\bar{x}_t$ represents the % utilization of a resource (ratio of the used resource to the allocated resource) at each time t. Q captures the risk of state and R reflects the expense of action. The action of adding or removing resources is obtained by applying the affine policy on state $\bar{x}_t$ (or linear on augmented state x_t) at each time step. We derive the proper dynamics, i.e., A, B. With this dynamics, we select Q, R, μ, W via model-based approach with domain knowledge on wastage cost and risk measure. After estimating the system, we compute the optimal policy using Riccati algorithm.

3 RESULTS

Figure 1 shows that across all services our approach achieved higher CPU (average 21.2%, range 16.1-36.4%) and memory (average 19.9%, range 5.7-53.3%) utilization than the autoscaler, since it dynamically adapts to the resource usage and computes the action based on usage data at that time step. On the other hand, the autoscaler is restricted to fixed thresholds and scaling settings, and cannot make any data-based decisions. To summarize, in this paper, we innovatively apply LQR to manage resources for enterprise services and show promising results on real data-sets.

REFERENCES

[1] Rudolf Emil Kalman. When is a linear control system optimal? *Journal of Basic Engineering*, 86(1):51–60, 1964.
[2] Tania Lorido-Botran, Jose Miguel-Alonso, and Jose A Lozano. A review of auto-scaling techniques for elastic applications in cloud environments. *Journal of grid computing*, 12(4):559–592, 2014.
[3] Fabiana Rossi, Matteo Nardelli, and Valeria Cardellini. Horizontal and vertical scaling of container-based applications using reinforcement learning. In *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, pages 329–338. IEEE, 2019.
[4] Gerald Tesauro, Nicholas K Jong, Rajarshi Das, and Mohamed N Bennani. A hybrid reinforcement learning approach to autonomic resource allocation. In *2006 IEEE International Conference on Autonomic Computing*, pages 65–73. IEEE, 2006.