Online Optimization in Cloud Resource Provisioning: Predictions, Regrets, and Algorithms

Joshua Comden
Stony Brook University
Stony Brook, NY, USA
joshua.comden@stonybrook.edu

Sijie Yao
Stony Brook University
Stony Brook, NY, USA
sijie.yao@stonybrook.edu

Niangjun Chen
Institute of High Performance
Computing
Singapore
chenjn@ihpc.a-star.edu.sg

Haipeng Xing
Stony Brook University
Stony Brook, NY, USA
haipeng.xing@stonybrook.edu

Zhenhua Liu*
Stony Brook University
Stony Brook, NY, USA
zhenhua.liu@stonybrook.edu

CCS CONCEPTS

• **Social and professional topics** → **Computing equipment management**; • **Theory of computation** → **Online learning algorithms**; **Regret bounds**; • **Computer systems organization** → **Cloud computing**.

KEYWORDS

online optimization; meta-algorithms; resource allocation

ACM Reference Format:

Joshua Comden, Sijie Yao, Niangjun Chen, Haipeng Xing, and Zhenhua Liu. 2019. Online Optimization in Cloud Resource Provisioning: Predictions, Regrets, and Algorithms. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '19 Abstracts)*, June 24–28, 2019, Phoenix, AZ, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3309697.3331516>

1 INTRODUCTION

Several different control methods are used in practice or have been proposed to cost-effectively provision IT resources. Due to the dependency of many control methods on having accurate predictions of the future to make good provisioning decisions, there has been a great deal of literature on prediction workload demand. However, even with all of this literature on workload predictions and their utilization in control algorithms, the understanding of prediction error and how to handle it remains an important open issue and research challenge [1].

In this paper [4] we aim to mend this gap by making the following contributions: (i) Prediction error is modeled to aid in proving worst-case dynamic regret bounds for control algorithms. (ii) Upper bounds on dynamic regret are proven for a variety of algorithms in terms of the prediction error model. In order to choose which algorithm to run without prediction error knowledge, a simple online

meta-algorithm is designed. (iii) A detailed analysis of prediction accuracy is done for cloud computing by fitting real-world CPU utilization traces of Azure virtual machines to popular prediction models. (iv) Using real-world trace based simulations of CPU allocation for virtual machines, the proposed meta-algorithm is shown to outperform a popular algorithm selection policy and perform very closely to that of the best algorithm chosen in hindsight.

2 PROBLEM FORMULATION

In practice, the cloud service provider tackles the lack of future information by making predictions on future demands. Denote $\hat{y}_{t|\tau}$ as the prediction of resource demands y_t using information up to $\tau - 1$ for $\tau \leq t$. Assuming a prediction window of w steps ahead, we can model the decisions and information flow of the cloud service provider in the following manner. At each timeslot $t \in \{1, \dots, T\}$: (i) The cloud service provider predicts the future demands $\hat{y}_{t|t}, \dots, \hat{y}_{t+w-1|t}$ and makes an allocation decision $x_t \in \mathcal{X}$. (ii) Subsequently, the true demand y_t is revealed and the cloud service provider incurs the operational cost $f(x_t, y_t)$ and the switching cost $\alpha \|x_t - x_{t-1}\|$.

To understand how to choose online algorithms under realistic predictions, we introduce a simple prediction error model that connects prediction errors to the potential performance loss. This is done by restricting the adversary's power with a budget $B_{k,T}$ in deciding how poor the predictions can be in terms of possible performance loss.

DEFINITION 1. *The sequence of true demands and their k -step ahead predictions $(y_k, \hat{y}_{k|1}), \dots, (y_T, \hat{y}_{T|T-k+1})$ satisfy a k -step prediction budget $B_{k,T}$ if*

$$\sum_{t=k}^T \sup_{x \in \mathcal{X}} |f(x, y_t) - f(x, \hat{y}_{t|t-k+1})| \leq B_{k,T}. \quad (1)$$

To compare an online algorithm's performance to that of the offline optimal solution, we employ the worst-case performance metric *dynamic regret*:

DEFINITION 2. *We define the L -constrained dynamic regret of online algorithm $\mathcal{A}$ with prediction error budgets $B_{1,T}, \dots, B_{w,T}$ as:*

$$\mathcal{R}_T^{\mathcal{A}}(B_{1:w,T}, L) := \sup_{y \in \mathcal{Y}_{1:w,T}} \{ \text{cost}(\mathcal{A}, y_{1:T}) - \text{cost}(OPT_L, y_{1:T}) \} \quad (2)$$

where OPT_L is the optimal solution with a constrained path-length $\sum_{t=1}^T \|x_t^* - x_{t-1}^*\| \leq L$.

*Corresponding Author

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMETRICS '19 Abstracts, June 24–28, 2019, Phoenix, AZ, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6678-6/19/06.

<https://doi.org/10.1145/3309697.3331516>

Table 1: Upper Bounds on Dynamic Regret by Algorithm

Alg.	Dynamic Regret
OGD	$\mathcal{R}_T^{\text{OGD}} \leq 2\sqrt{D\bar{\kappa}(2D\bar{\kappa} + L)G\left(\frac{G}{2} + \frac{\alpha}{\bar{\kappa}}\right)T}$
OBD	$\mathcal{R}_T^{\text{OBD}} \leq 2B_T + \alpha\sqrt{\frac{2GLT}{m}}$
CHC	$\left(\frac{1}{w+1}T - 1\right)\alpha D \leq \mathcal{R}_T^{\text{CHC}} \leq \frac{2}{v}\left(D\alpha T + \sum_{k=1}^v B_{k,T}\right)$

Algorithm 1 Meta-algorithm

```

1: Initialize selected algorithm,  $\mathcal{A} \in \Gamma$ .
2: for  $t = 1, \dots, T$  do
3:   if  $t \bmod \pi = 0$  then
4:     Select  $\mathcal{A} := \arg \min_{i \in \Gamma} \left\{ \text{cost}(i, y_{1:t-1}) + \alpha \|x_{t-1}^{\mathcal{A}} - x_{t-1}^{(i)}\| \right\}$ 
5:   end if
6:   Receive  $x_t^{(i)}$  from every algorithm  $i \in \Gamma$ .
7:   Implement  $x_t^{\mathcal{A}}$ .
8: end for

```

3 ONLINE ALGORITHMS

There exist many online algorithms which can be categorized by how much prediction they utilize. We prove dynamic regret bounds for many of them in the presence of prediction errors using the prediction error budget model. Specifically, Table 1 summarizes our theoretical results on the dynamic regret bounds of Online Gradient Descent (OGD) [6], Online Balanced Descent (OBD) [3], and Committed Horizon Control (CHC) [2]. Let D be the diameter of the action space $\mathcal{X}$ and let $f(x, y)$ be G -Lipschitz continuous for any given demand y with respect to its provisioning decision x .

With all of the available algorithms described above, the decision becomes which one(s) should be chosen at runtime and at what parameter settings. In this direction, we design a simple meta-algorithm that selects which algorithm from a finite set Γ to implement and periodically updates its selection according to past performance. The technical details are given in Algorithm 1. It essentially learns which algorithm was the best in hindsight and switches to that while taking into account the cost of switching from its current algorithm trajectory to that of another.

4 VIRTUAL MACHINE PREDICTION

We predict the CPU utilization of 1,003 long running virtual machines (VMs) from the Microsoft Azure Public Dataset [5] using the following prediction models: Naive prediction, Random Forest (RF) regression, Seasonal Auto-Regressive Moving Average (SARMA), and Seasonal Exponential Smoothing (SES). The Cumulative Absolute Prediction Error (CAPE) versus time for four VMs is shown in Figure 1 which gives evidence that the prediction error budget $B_{k,T}$ increases linearly in time.

5 PERFORMANCE EVALUATION

We tested our Meta-Algorithm (Algorithm 1) to dynamically select prediction models with online algorithms using 55 VM traces from the Azure Public Dataset [5]. It was evaluated against the popular Weighted Majority Algorithm and selecting the best (prediction model, online control algorithm) pair in hindsight called Offline Select. Figure 2 gives the normalized cost averaged over the 55

Figure 1: Time v.s CAPE(s) for 4 VMs **Figure 2: Average cost incurred by the meta-algorithm (Dyn Select) vs. switching cost coefficient α .**

VMs versus the switching cost coefficient α . Our meta-algorithm performs much closer to that of selecting the best pair in hindsight than the Weighted Majority Algorithm, especially under high switching cost environments.

6 ACKNOWLEDGMENTS

This research was partially funded by NSF grants CNS-1617698, CNS-1717588, CNS-1730128, DGE-1633299, and DMS-1612501.

REFERENCES

- [1] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah, and Philippe Merle. 2018. Elasticity in cloud computing: state of the art and research challenges. *IEEE Transactions on Services Computing* 11, 2 (2018), 430–447.
- [2] Nianguan Chen, Joshua Comden, Zhenhua Liu, Anshul Gandhi, and Adam Wierman. 2016. Using predictions in online optimization: Looking forward with an eye on the past. In *Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science*. ACM, 193–206.
- [3] Nianguan Chen, Gautam Goel, and Adam Wierman. 2018. Smoothed Online Convex Optimization in High Dimensions via Online Balanced Descent. In *Proceedings of the 31st Conference On Learning Theory (Proceedings of Machine Learning Research)*, Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet (Eds.), Vol. 75. PMLR, 1574–1594. <http://proceedings.mlr.press/v75/chen18b.html>
- [4] Joshua Comden, Sijie Yao, Nianguan Chen, Haipeng Xing, and Zhenhua Liu. 2019. Online Optimization in Cloud Resource Provisioning: Predictions, Regrets, and Algorithms. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 1 (2019), 16.
- [5] Eli Cortez, Anand Bonde, Alexandre Muzio, Mark Russinovich, Marcus Fontoura, and Ricardo Bianchini. 2017. Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 153–167.
- [6] Martin Zinkevich. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 928–936.