Toward Resource-Efficient Cloud Systems: Avoiding Over-Provisioning in Demand-Prediction Based Resource Provisioning

Liuhua Chen
ECE, Clemson University
Haiying Shen
CS, University of Virginia

2016 IEEE International Conference on Big Data

Presenter: Yi-Ning Chang

October 19, 2017

Outline

1. Introduction
2. System Design
3. Experiment
4. Conclusion

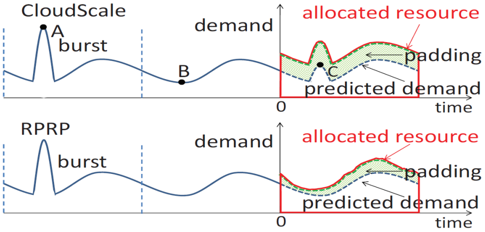
1. Introduction
2. System Design
3. Experiment
4. Conclusion

Introduction

- In cloud systems, cloud providers abstract resources in physical machines into virtual machines and sell them to the tenants.
- To ensure resource provisioning for guaranteeing SLOs[1], clouds can use *demand-prediction based resource provisioning schemes*.
- Achieving the tradeoff between the penalties associated with *SLO violations* and *high resource utilization* requires an accurate demand prediction methodology.

[1] SLO: Service Level Objectives

Previous Work - CloudScale



- CloudScale predicts the demand at a time period based on a historical record.
- Padding: using the high-frequency spectrum or the average of the latest prediction error.
- Online Adaptive: to handle underestimation, raising the resource allocation by $\alpha > 1$ until an error is corrected.

RPRP[1]

- RPRP excludes bursts in demand prediction and specifically handles bursts to avoid resource over-provisioning.
- Algorithm
  - *burst-exclusive prediction algorithm*
  - *load-dependent padding algorithm*
  - *responsive padding algorithm*
- Algorithm 1 and 2 aim to exclude bursts, and algorithm 3 aims to handle bursts.

[1] RPRP: Resource-efficient Predictive Resource Provisioning system

1 Introduction

2 System Design

3 Experiment

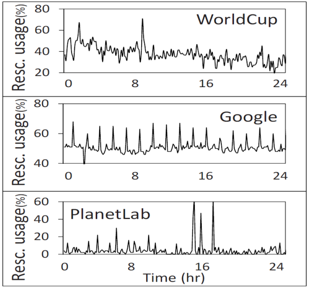
4 Conclusion

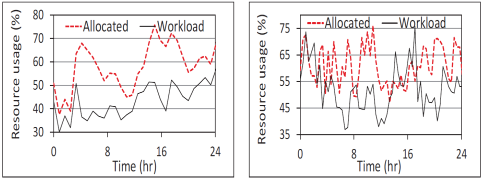
Objective

- Denote a VM's records:
  - *workload demand*: $D = \{d_{t_1}, ..., d_{t_i}, ..., d_{t_N}\}$
  - *allocated resource*: $A = \{a_{t_1}, ..., a_{t_i}, ..., a_{t_N}\}$
  - *utilized resource*: $U = \{u_{t_1}, ..., u_{t_i}, ..., u_{t_N}\}$
  - *resource capacity*: C
- And from the historical records, we have:
  - *predict demand*: $P = \{p_{t_{N+1}}, p_{t_{N+2}}, ..., p_{t_{N+T}}\}$
  - *allocated resource*: $A = \{a_{t_{N+1}}, a_{t_{N+2}}, ..., a_{t_{N+T}}\}$
- Goal: determine allocated resource A such that
  - $d_{t_i} \leq a_{t_i} \leq C$
  - and meanwhile to minimize $a_{t_i} - d_{t_i}, \forall t_i > t_N$

Algo.1: Burst-exclusive Prediction

- Trace analysis and CloudScale prediction + padding.

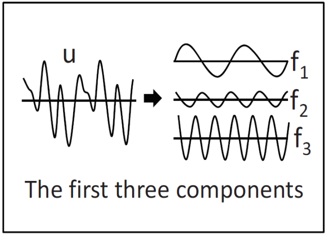




(a) Low burst density. (b) High burst density.

Algo.1: Burst-exclusive Prediction

- RPRP relies on FFT to exclude the burst.
- FFT is applicable for predicting workload demand in repeated periodic patterns P based on the historical utilization series U.



Algo.2: Load-dependent Padding

Algo.3: Responsive Padding

1 Introduction

2 System Design

3 Experiment

4 Conclusion

Experiment

1 Introduction

2 System Design

3 Experiment

4 Conclusion

Conclusion

Future Work