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

Liuhua Chen
Clemson University
liuhuac@g.clemson.edu

Haiying Shen
University of Virginia
hs6ms@virginia.edu

Background and Motivation

To ensure **resource provisioning** for guaranteeing service level objectives (SLOs), clouds can use demand-prediction based resource provisioning schemes that allocate physical resources to VMs according to the dynamically estimated VM demands. Inaccurate demand estimation could lead to **over-provisioning** (hence resource under-utilization) or **under-provisioning** (hence SLO violations). Providing more resources achieves low SLO violations while leading to low resource utilization, and vice versa. Achieving the trade-off between the penalties associated with **SLO violations** and **high resource utilization** (hence revenue maximization) requires an accurate demand prediction methodology.

Current solutions

Deals with demand mispredictions by adding a padding to a predicted demand.

PSRPS: uses the average of the latest ten prediction errors as the padding.

CloudScale: applies reverse FFT over the high frequency components of the original resource usage time series to synthesize the burst pattern. CloudScale then uses either the maximum or the 80th of the burst values as the padding based on the extracted burst pattern.

Drawbacks: do not exclude bursts.

Our approach

RPRP: excludes bursts in demand prediction and specifically handles bursts to avoid resource over-provisioning.

1. burst-exclusive prediction
2. load-dependent padding
3. burst-resilient shared padding
4. responsive padding

Design Details

1. Burst-Exclusive Prediction:
predicts demands based on history records

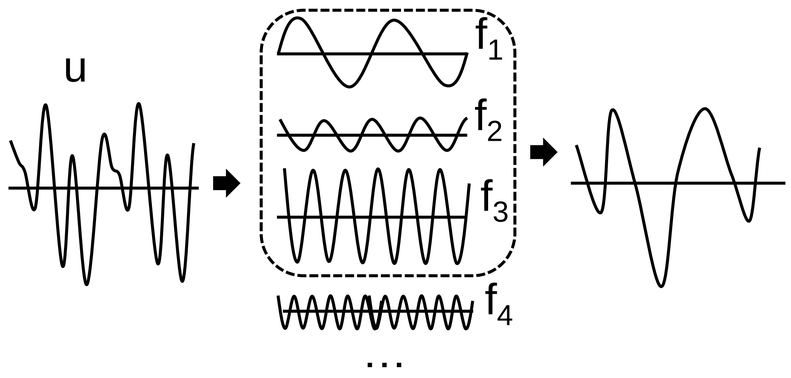

Figure 1 FFT-based burst-exclusive prediction.

2. Load-dependent Padding:
given the probability distribution of the predicted demand levels $\hat{p}_j$, the probability distribution of the actual demands for each $\hat{p}_j$ and allowed violation rate ε from SLO, how can we determine the padding value $\delta(\hat{p}_j)$ for each $\hat{p}_j$ to achieve $\overline{P_r} \geq 1-\varepsilon$ (probability that the allocated resource is sufficient) and meanwhile minimize the expected total allocated resource.

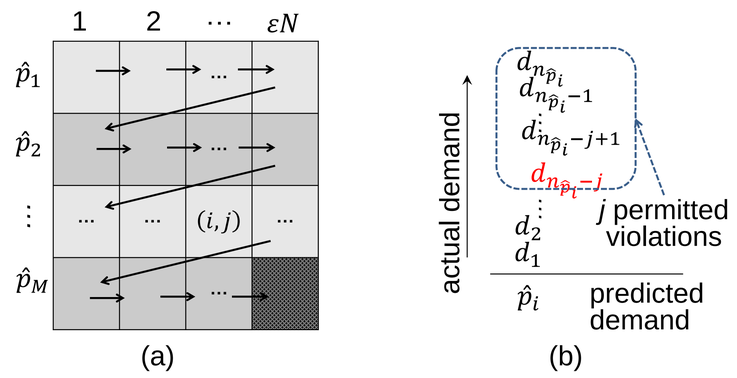

Figure 2 Dynamic programming algorithm. (a) an $M \times \varepsilon N$dynamic programming matrix. (b) procedure to determine the allocated resource if we place j permitted violations on predicted demand level $\hat{p}_j$.

3. Burst-Exclusive Prediction:
reserves common resources shared by co-located VMs for handling bursts.

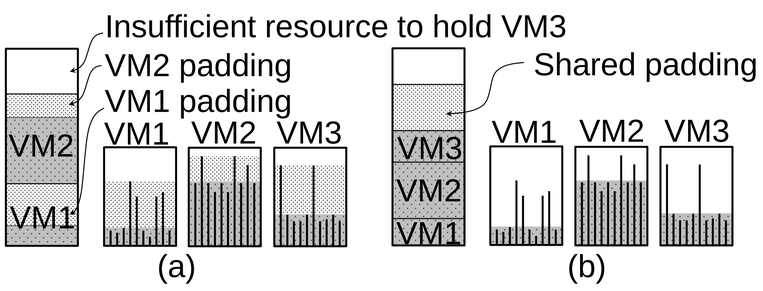

Figure 3 Burst-resilient shared padding.

4. Responsive Padding:
keeps the resource utilization efficiency while satisfying SLO dynamically.

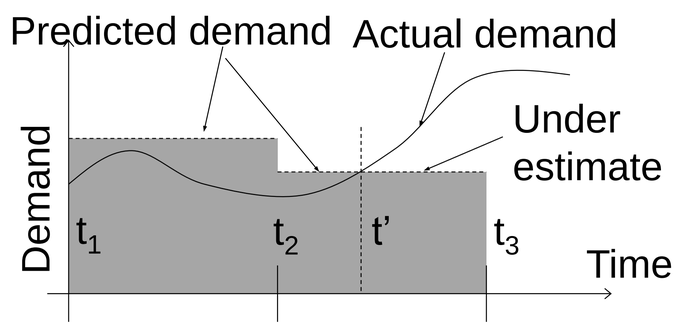

Figure 4 Underestimate correction. Demand prediction and resource allocation are performed at time t_1, t_2 and t_3. Responsive padding is performed at time t' where the allocated resource becomes insufficient for the demand before next prediction and allocation.

Experimental Results

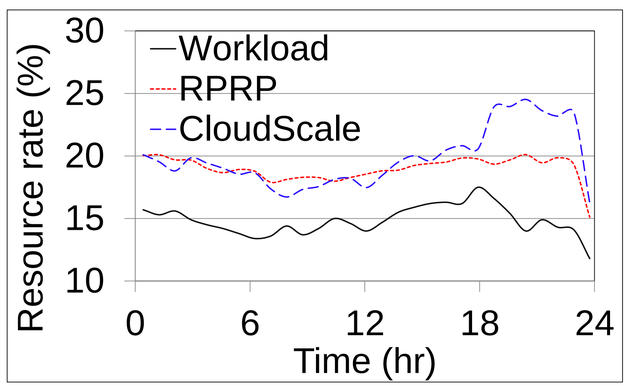

Figure 5 Performance of the prediction algorithms.

Result: Higher prediction accuracy.

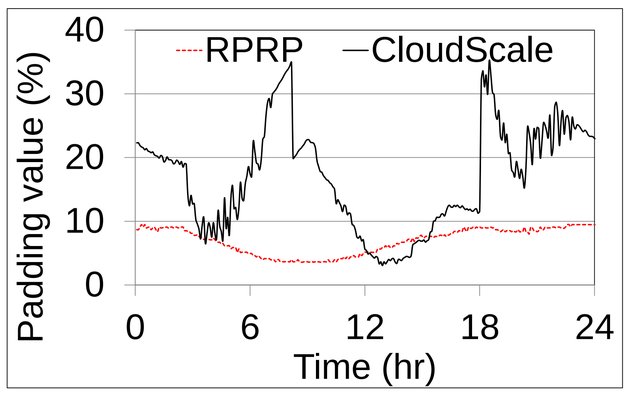

Figure 6 Performance of the padding algorithms.

Result: Lower padding while satisfying SLO.

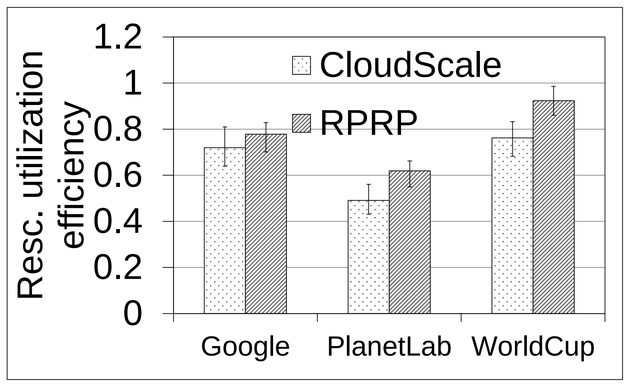

Figure 7 Performance of the padding algorithms.

Result: Higher utilization efficiency.

Acknowledgments

U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, IBM Faculty Award 5501145 and Microsoft Research Faculty Fellowship 8300751.

Future Work

In the future, we will extend RPRP to deal with resource provisioning for multiple co-located VMs with various priorities.