Mitigating Large Response Time Fluctuations through Fast Concurrency Adapting in Clouds

Jianshu Liu[*], Shungeng Zhang[*], Qingyang Wang[*], Jinpeng Wei[†]

[*]Louisiana State University, [†]University of North Carolina-Charlotte

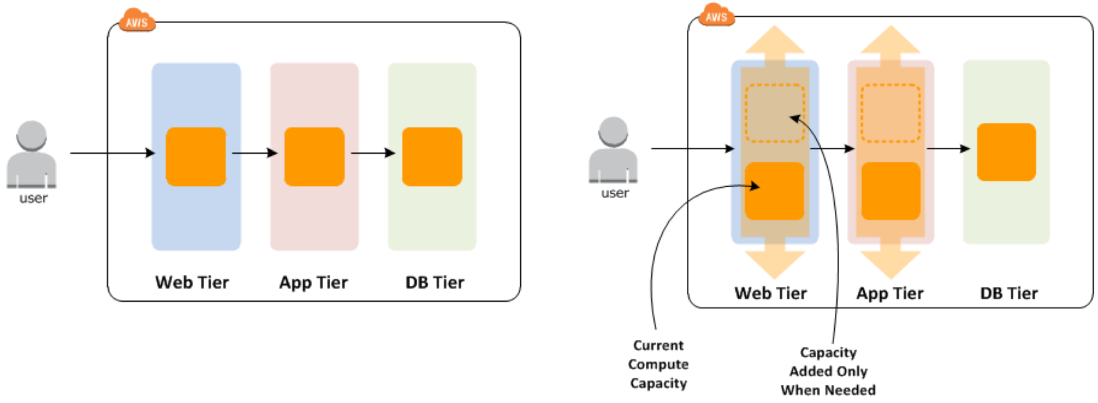
LSU

LOUISIANA STATE UNIVERSITY

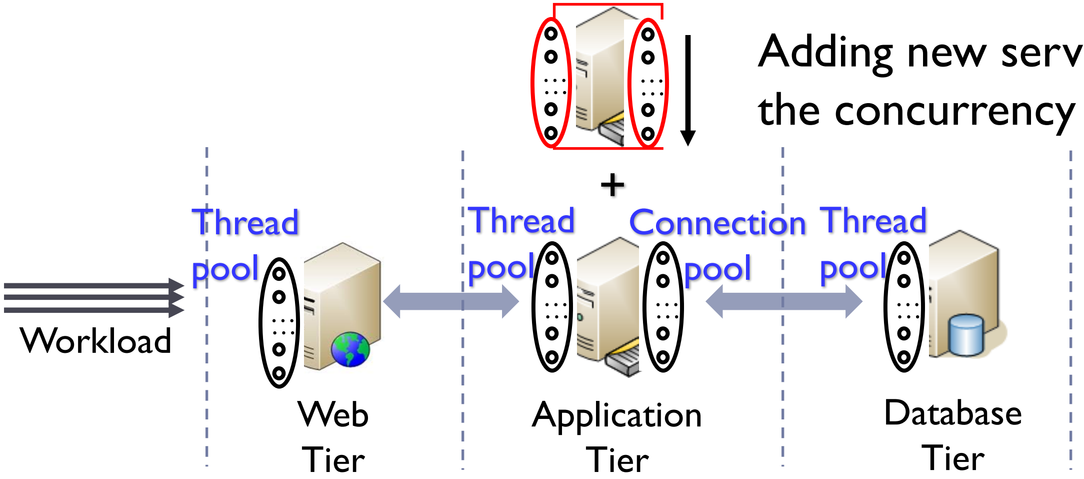
UNC CHARLOTTE

Hardware-only Scaling is Not Enough

- Amazon EC2-AutoScale only scales hardware resources to handle bursty workload

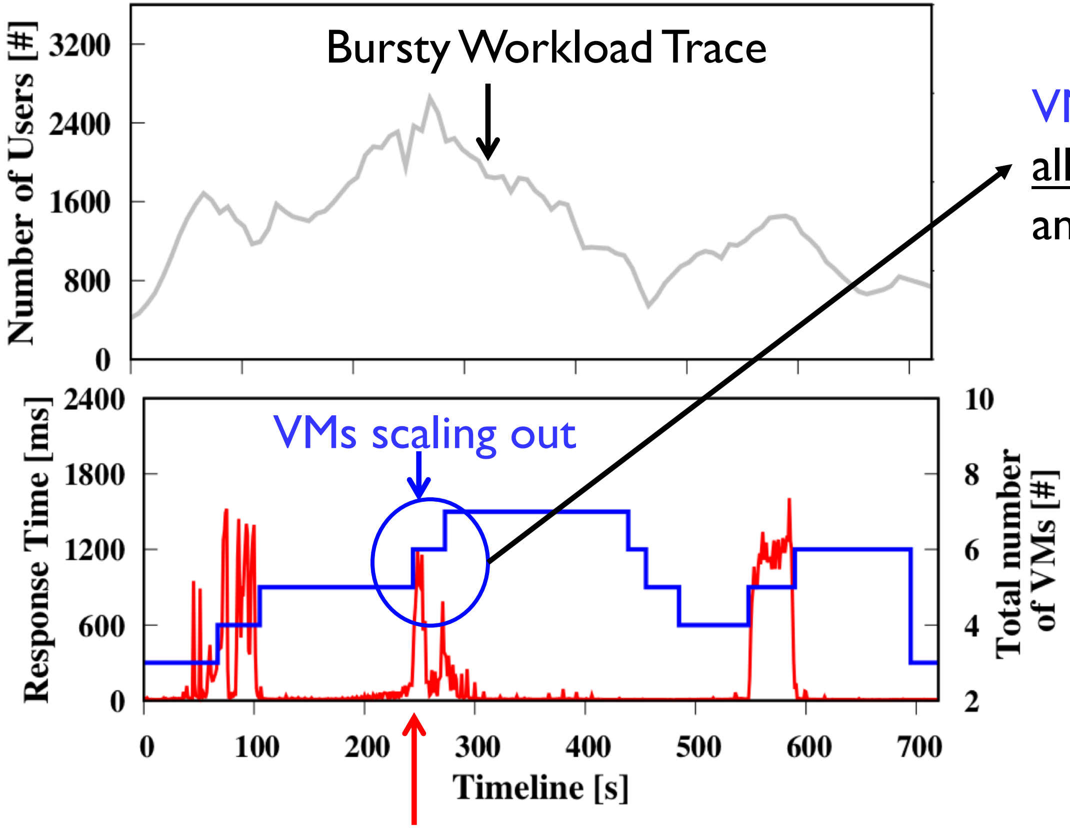


- However, soft resources also need to be scaled for optimal performance



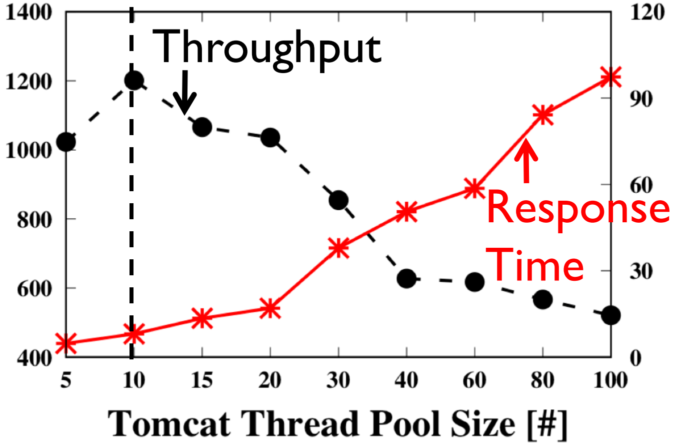
Adding new servers also changes the concurrency of the system

Large Response Time Spikes when only Scales Hardware Resources



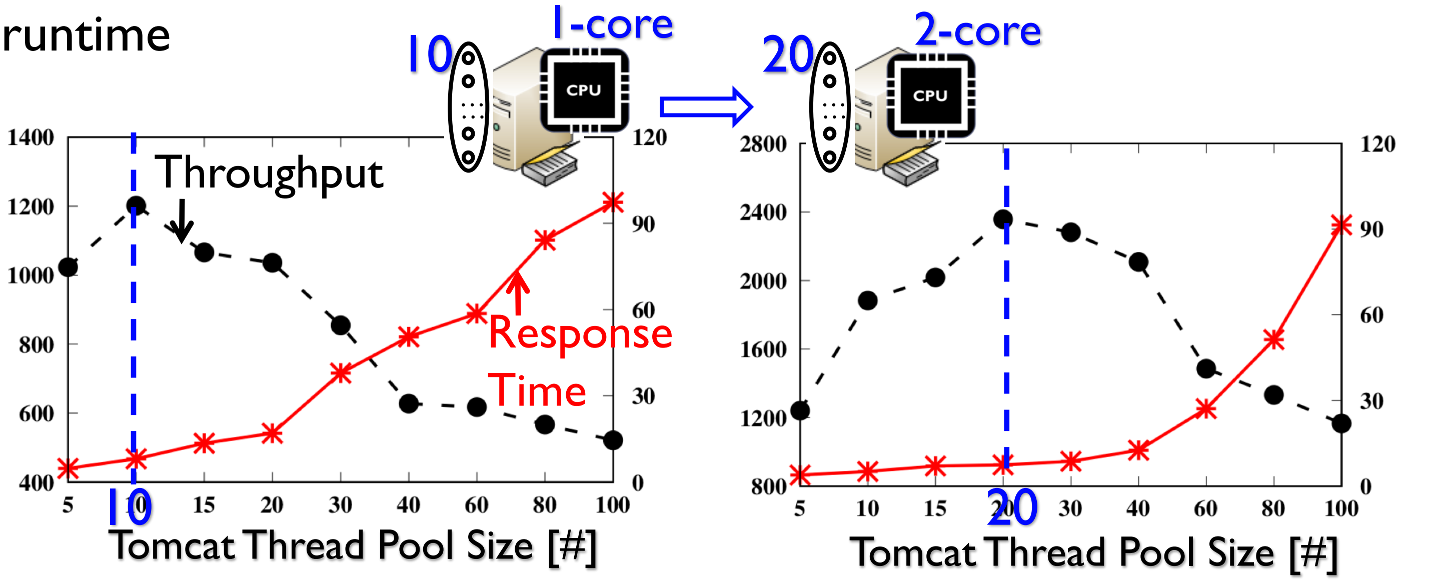
Response Time spikes observed

VMs scaling out changes soft resources allocation → changes system throughput and response time



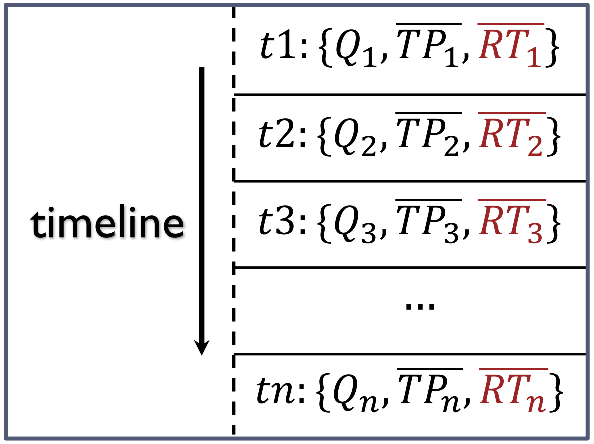
Problem Statement

- ☐ State-of-the-art approach: pre-profiling to determine the optimal soft resource allocation *–[Wang et al. TPDS'19]*
  - Problem: Optimal soft resource allocation changes during runtime



How can we quickly determine the optimal soft resource allocation of each server in system?

Our Solution: Real-time Online Scatter-Concurrency-Throughput (SCT) Model



timeline

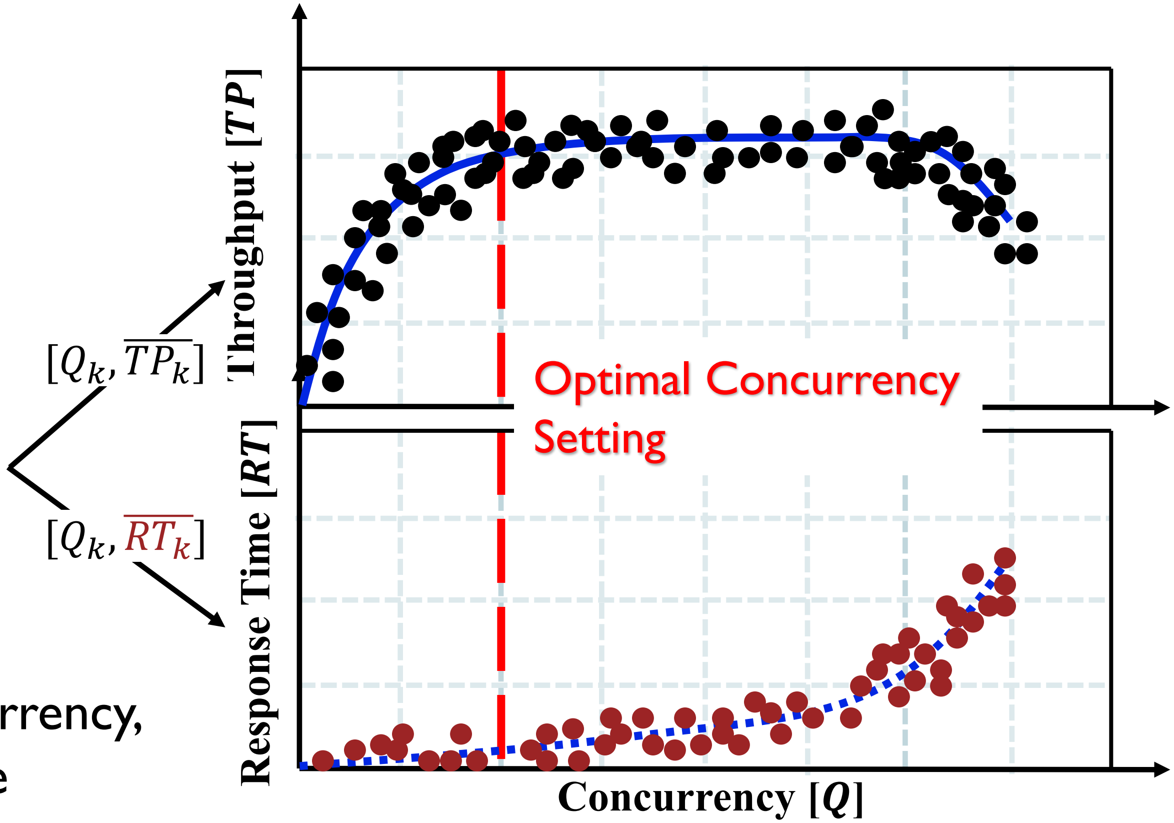
$t1: \{Q_1, \overline{TP_1}, \overline{RT_1}\}$

$t2: \{Q_2, \overline{TP_2}, \overline{RT_2}\}$

$t3: \{Q_3, \overline{TP_3}, \overline{RT_3}\}$

…

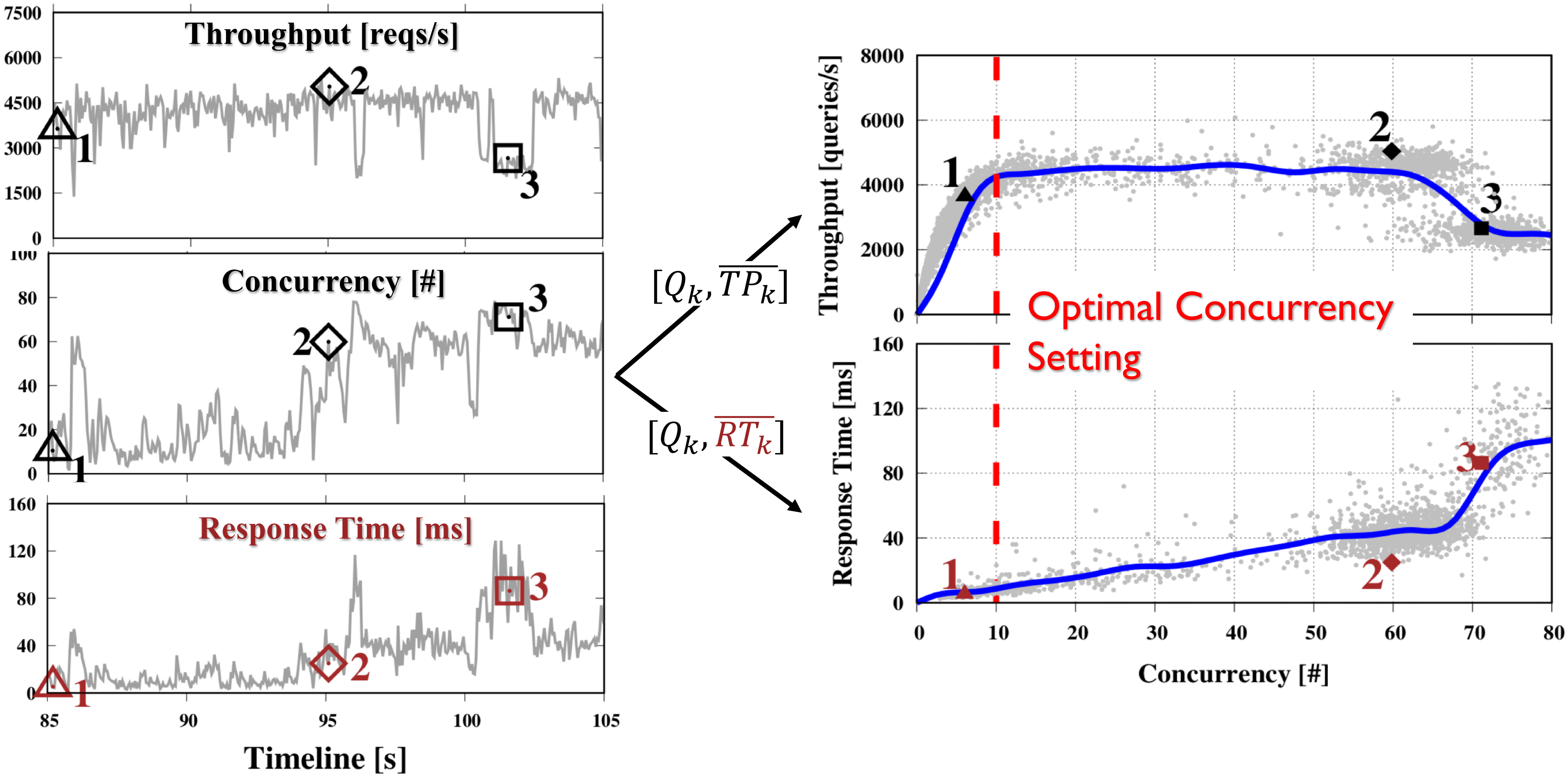
$tn: \{Q_n, \overline{TP_n}, \overline{RT_n}\}$



Optimal Concurrency Setting

Real-time (e.g., every 50ms) measurement of server concurrency, throughput, and response time

Applying SCT Model to MySQL

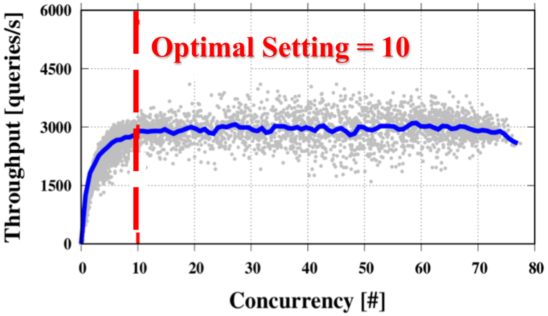


Optimal Concurrency Setting

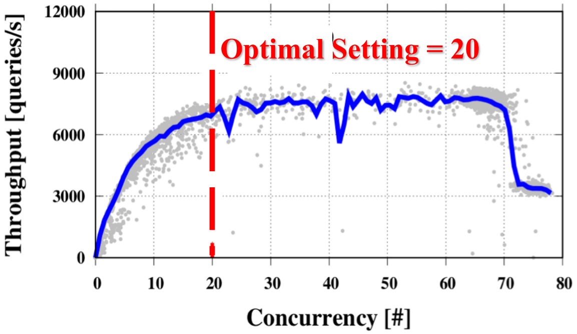
Applying SCT Model when Runtime Environment Changes

- □ MySQL optimal setting doubles after MySQL CPU core scales up from 1 to 2

The 1-core MySQL server case

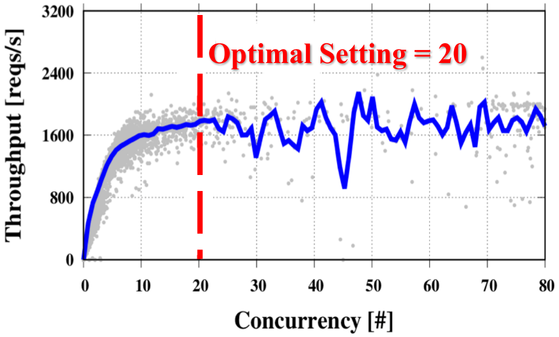


The 2-core MySQL server case

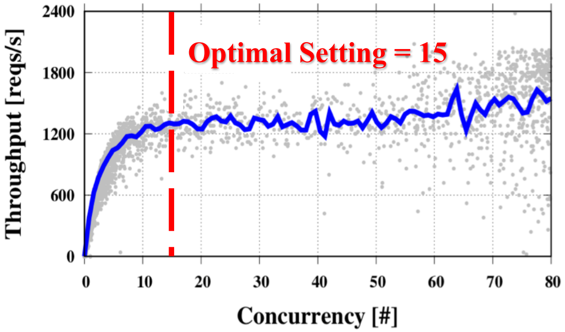


- □ Tomcat optimal setting decreases from 20 to 15 after RUBBoS dataset size doubles

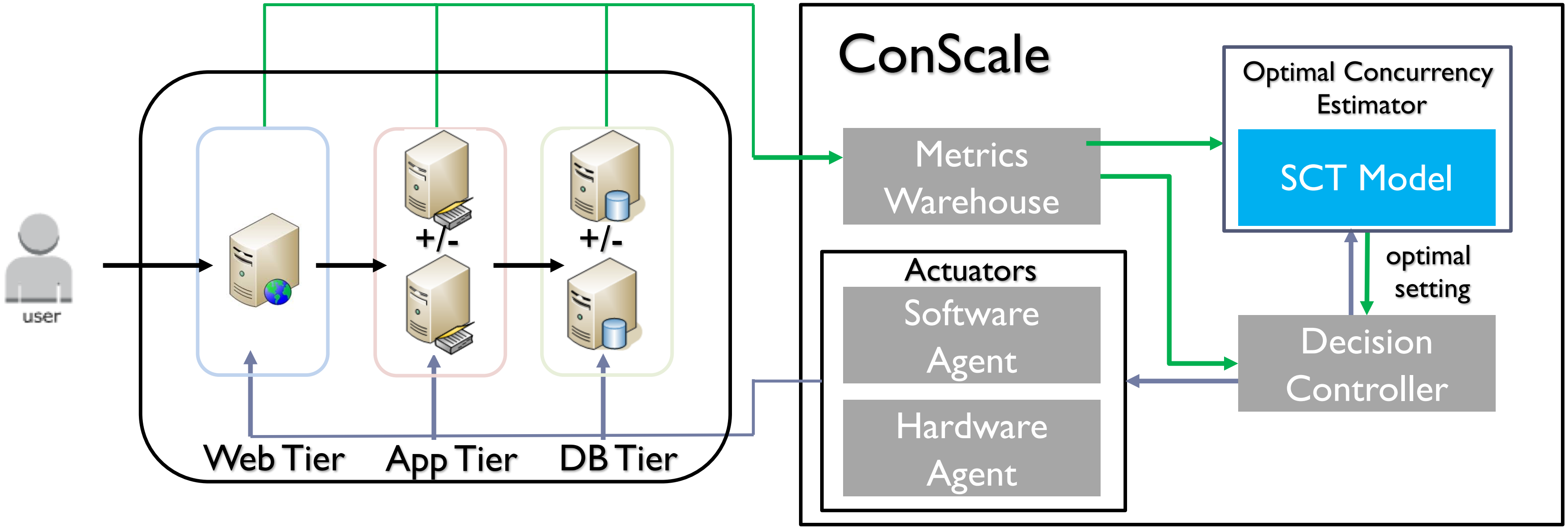
The Tomcat server case with original RUBBoS dataset



The Tomcat server case with enlarged RUBBoS dataset



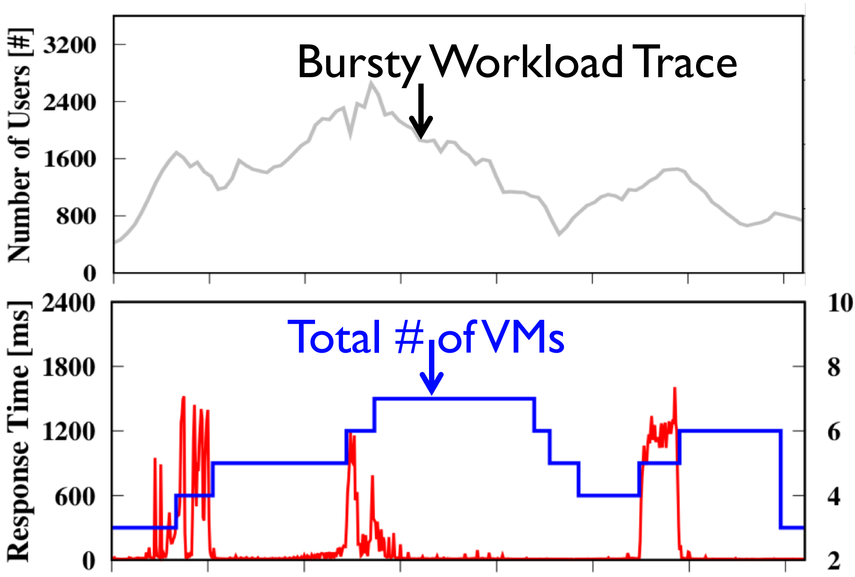
Integrate SCT Model to System Scaling Design (ConScale)



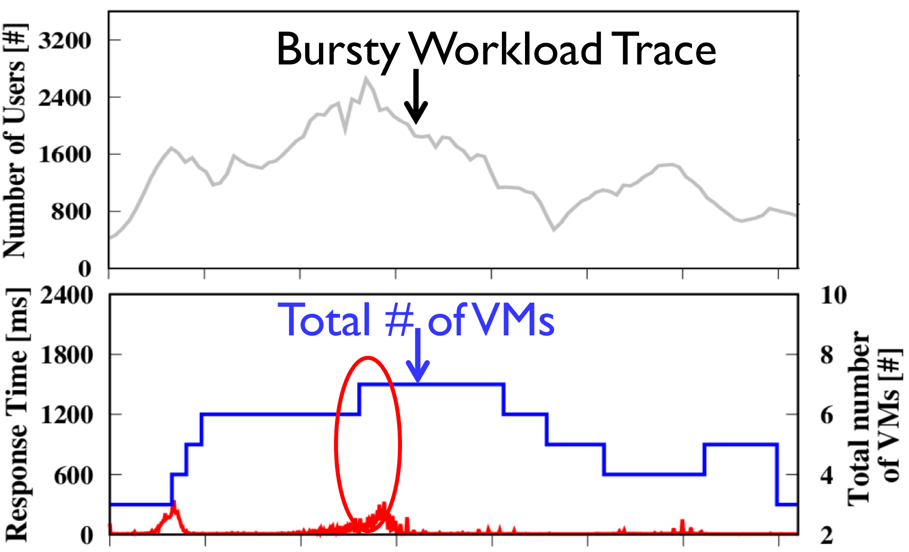
ConScale guarantees optimal soft resource resetting after hardware resources scaling

ConScale Mitigates the Large Response Time Fluctuations

EC2-AutoScale



ConScale



soft resource readapting after hardware scaling

Conclusion

Achieving good performance by scaling n-tier applications in Cloud requires the quick optimal soft resource reallocation of each server in the system

Contributions:

- Developed the online SCT model to quickly determine the optimal soft resource allocation of each server in an n-tier application
- Studied several factors that affect the optimal concurrency setting of servers
- Implemented the ConScale framework to realize fast and intelligent soft resources adaption in system scaling design

Author's Contact Information
Name: Jianshu Liu
E-mail Address: jliu96@lsu.edu

Paper Access here
Video would be available at here