An Experimental Study of Latency Long Tail Problem: Impact of Very Short Bottlenecks in Cloud Environments

Calton Pu

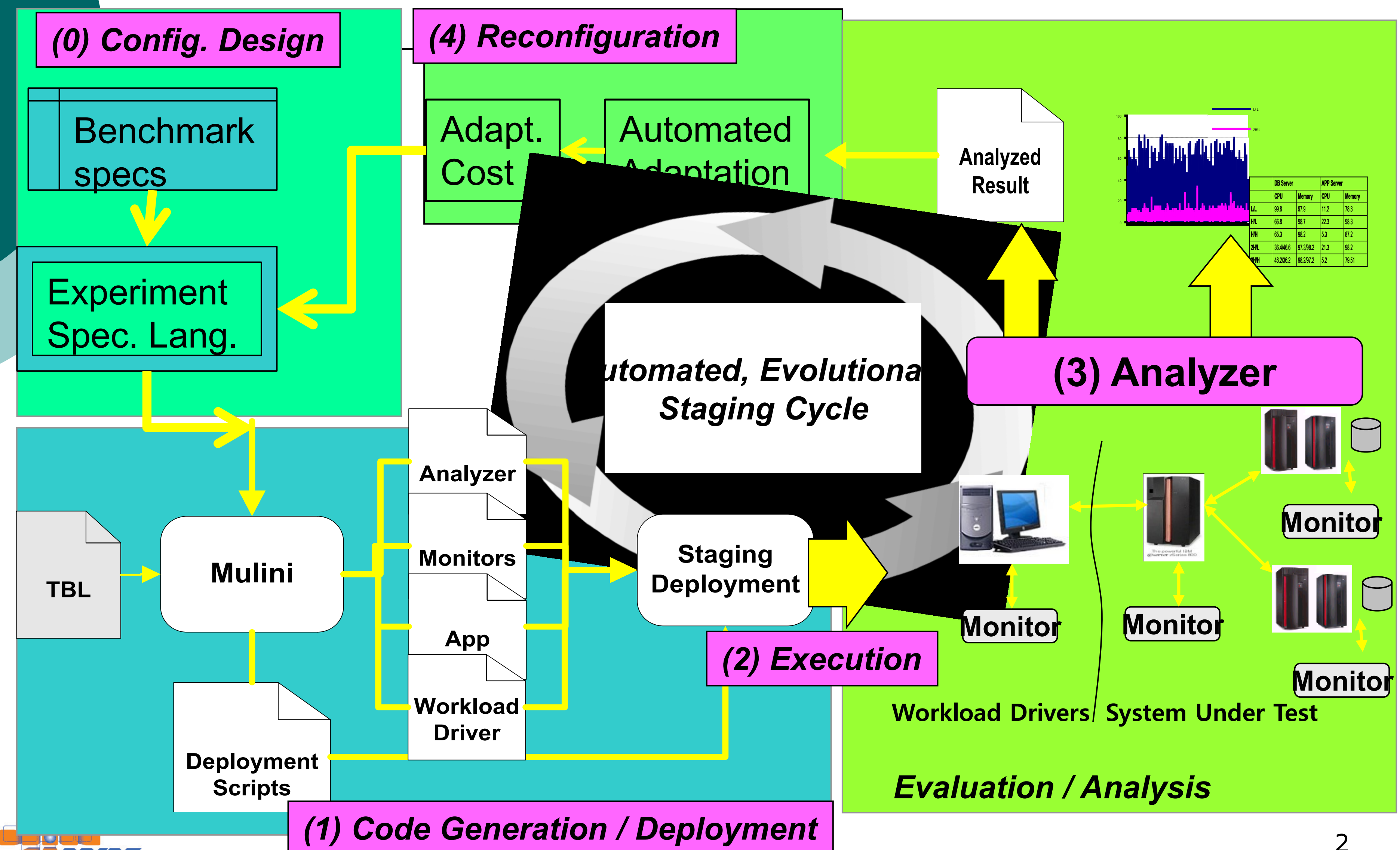
Professor and J.P. Imlay Chair in Software

CERCS, Georgia Institute of Technology

Many PhD, MS, Undergrad students

Collaborators from **HP Labs** (CA), **ATT Labs** (NJ), **IBM Research** (NY), **Intercontinental Exchange** (GA), **Wipro** (India), **Fujitsu Labs** (Japan), **NEC Labs** (CA), **Intel ISTC-CC** (PA), **Univ. Freiburg** (Germany), **Univ. Tokyo** (Japan), and other companies

Elba: Automated Measurements

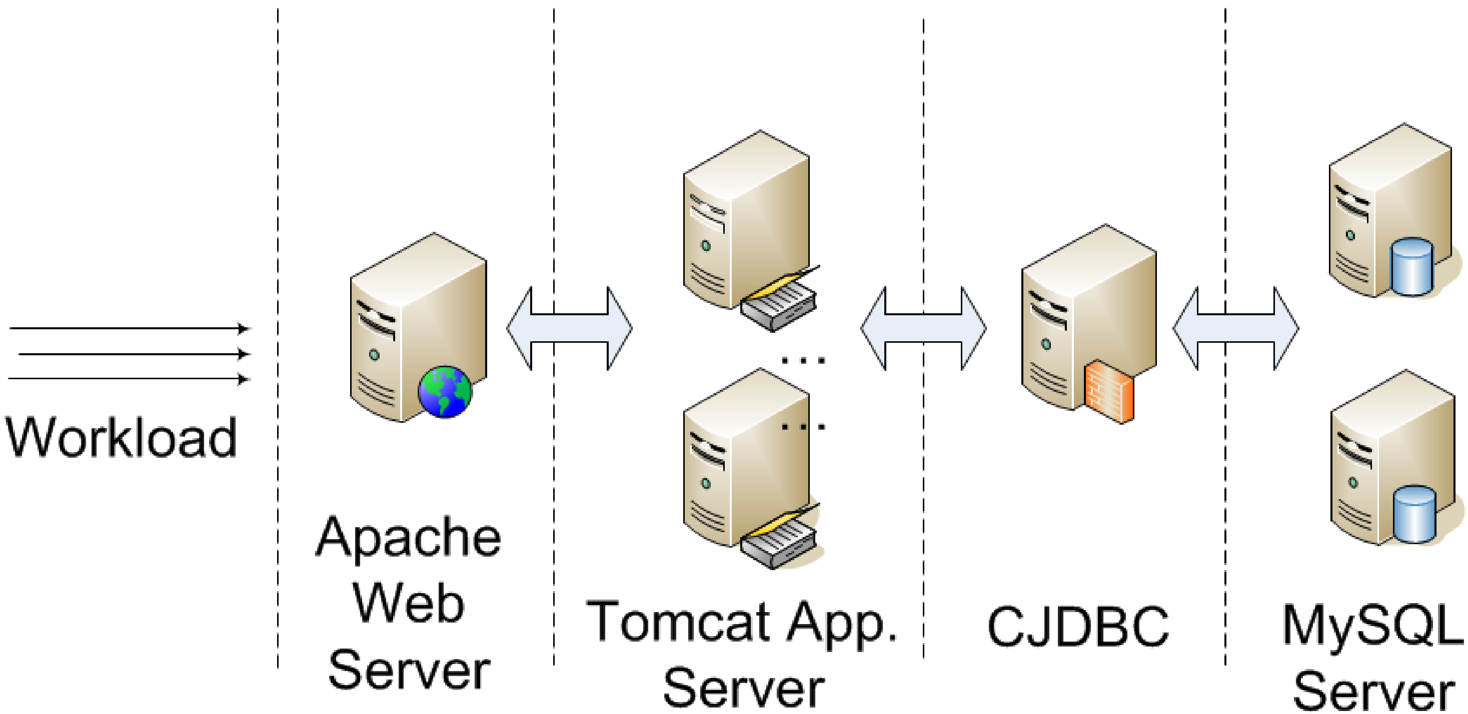


Elba Focus and Publications

- Experimental studies analyzing performance data from production-scale experiments on “real data centers”
  - More than 40 papers (2005 – 2015)
- Since 2013: 12 papers
  - IEEE CLOUD, SCC, ICDCS, IRI, BigData Congress, BigData, ACM TRIOS, Middleware
- 4 papers on *transient bottlenecks,* now renamed Very Short Bottlenecks (VSB)

Web-Facing Multi-Tier Apps

- Example: RUBBoS benchmark based on Slashdot
  - Sample configuration (1/2/1/2)

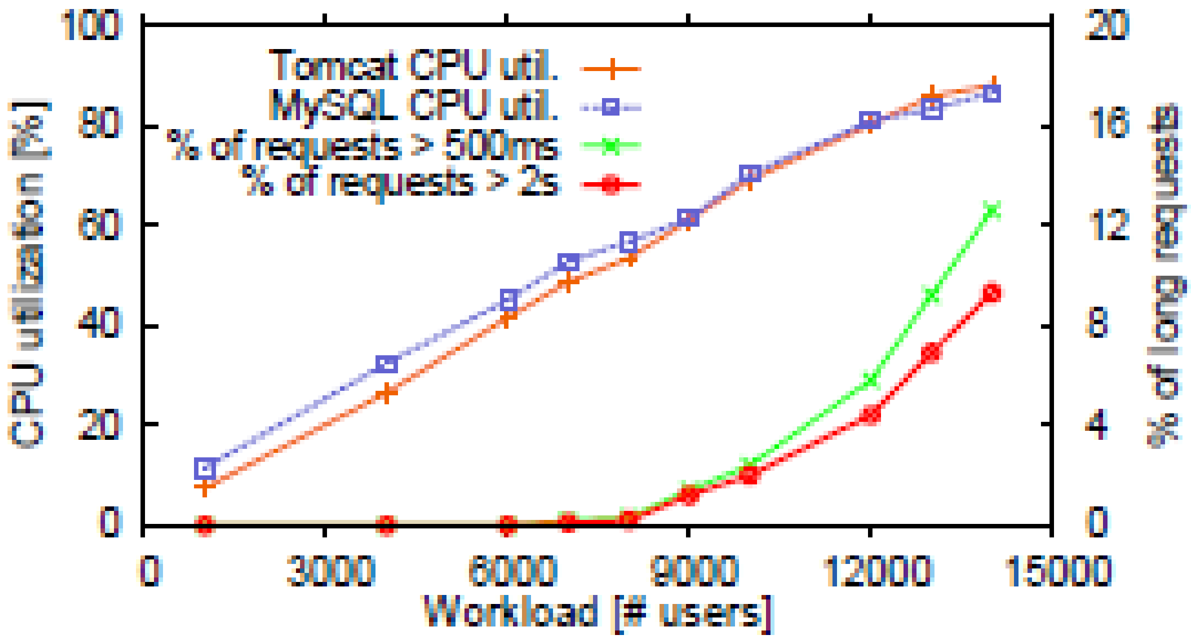


Importance of N-Tier Systems

- A scalable distributed architecture
  - Division of labor for low-latency tasks
  - Web servers for parsing/HTML handling
  - App servers for business logic handling
  - DB servers for consistent data management
- Separation of stateless from stateful
  - DB servers handle the difficult stateful part
  - Web and App servers are “stateless” so more instances can be easily added if needed

Latency Long Tail Problem

- At moderate CPU utilization levels (about 60% at 9000 users), 4% of requests take several seconds, instead of milliseconds

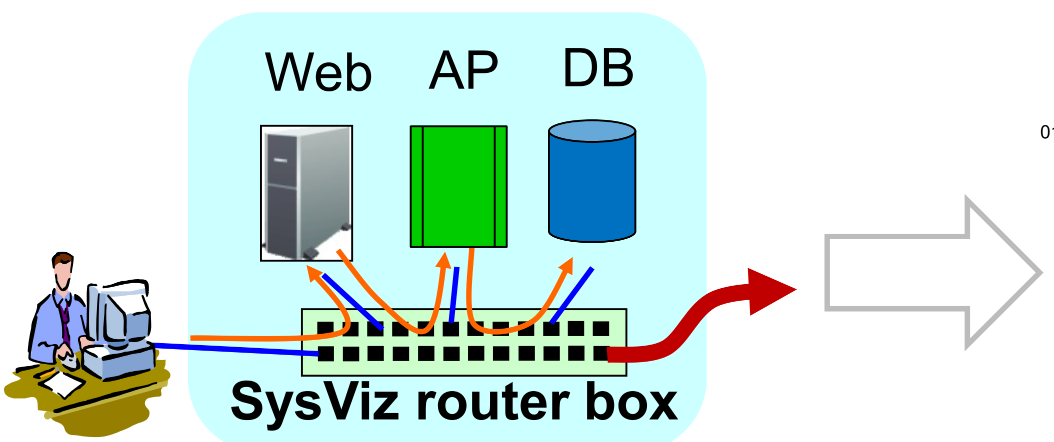


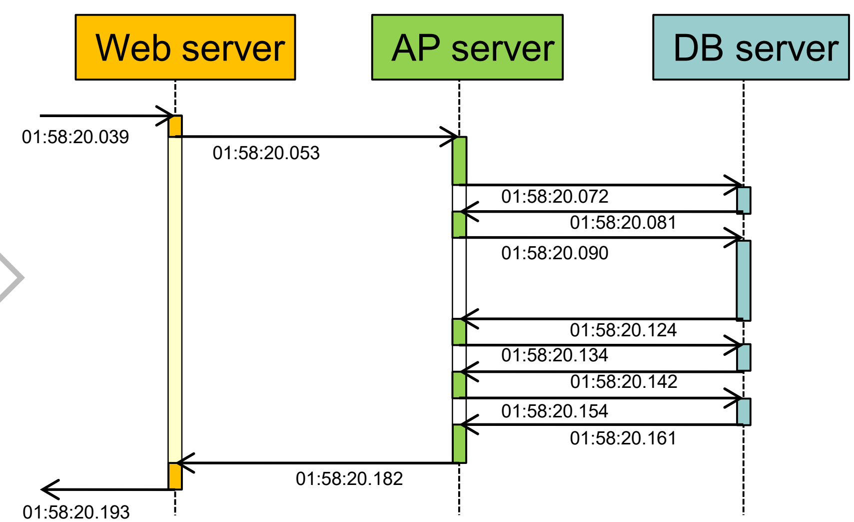
Latency Long Tail: A Serious Research Challenge

- No system resource is near saturation
  - Very Long Response Time (VLRT) requests start to appear at moderate utilization levels (often at 50% or lower)
- VLRT requests themselves are not bugs:
  - They only take milliseconds when run by themselves
  - Each run presents different VLRT requests
- VLRT requests appear and disappear too quickly for most monitoring tools

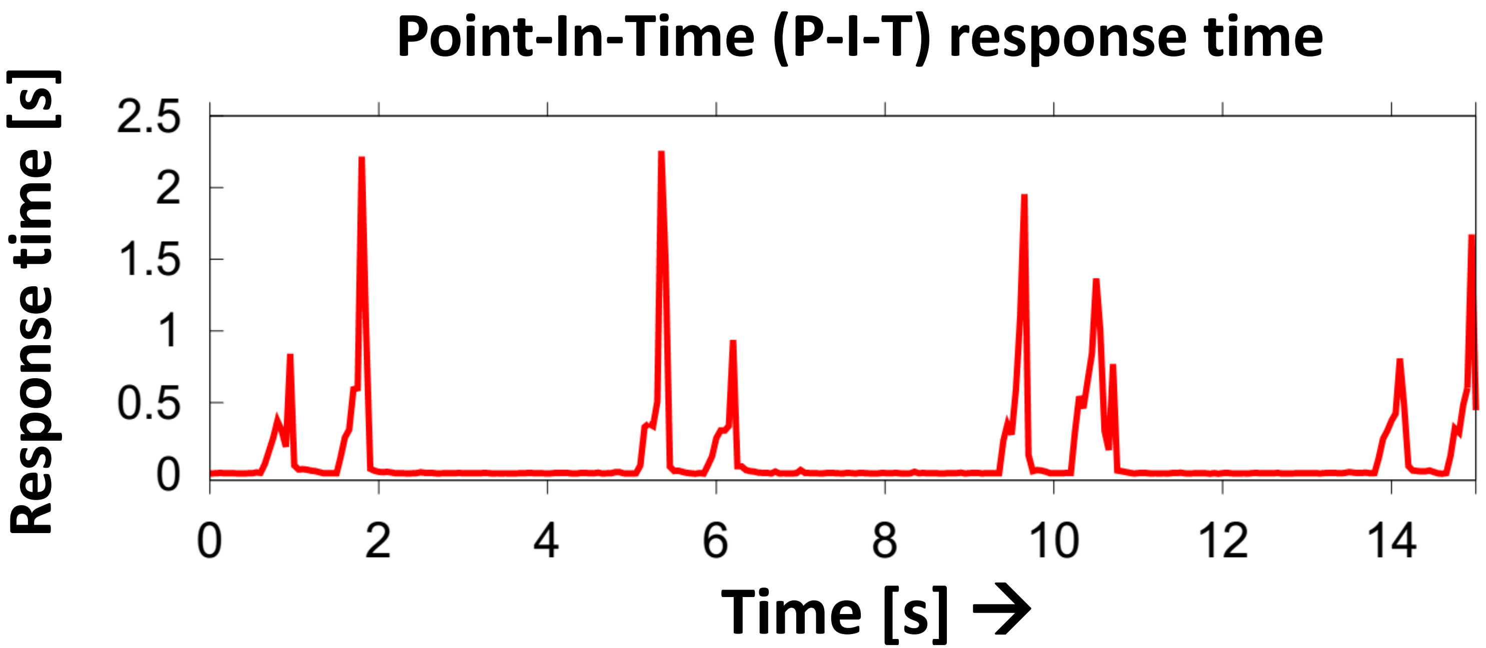
Passive Message Timestamping Infrastructure

- Fine-grain (μs) timestamps on each event
- Exact knowledge of each request processing at each tier boundary





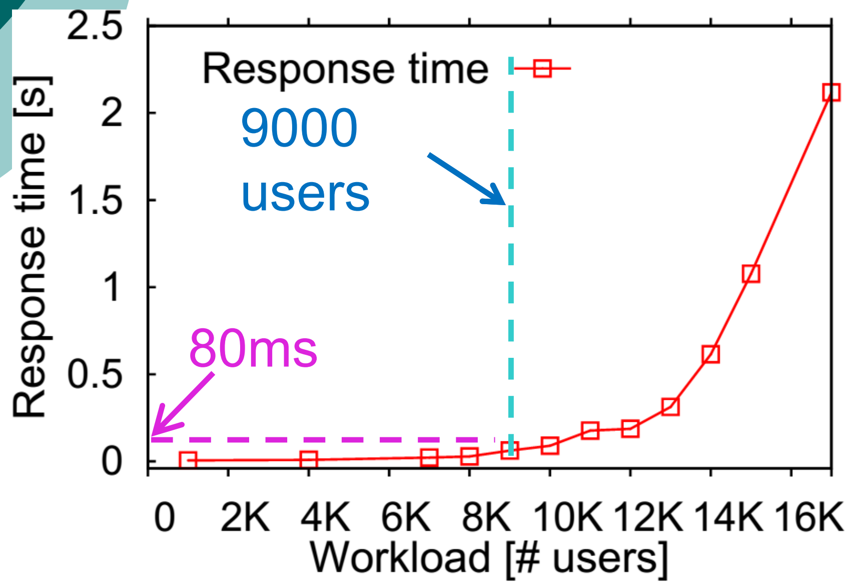
Now You See Me….

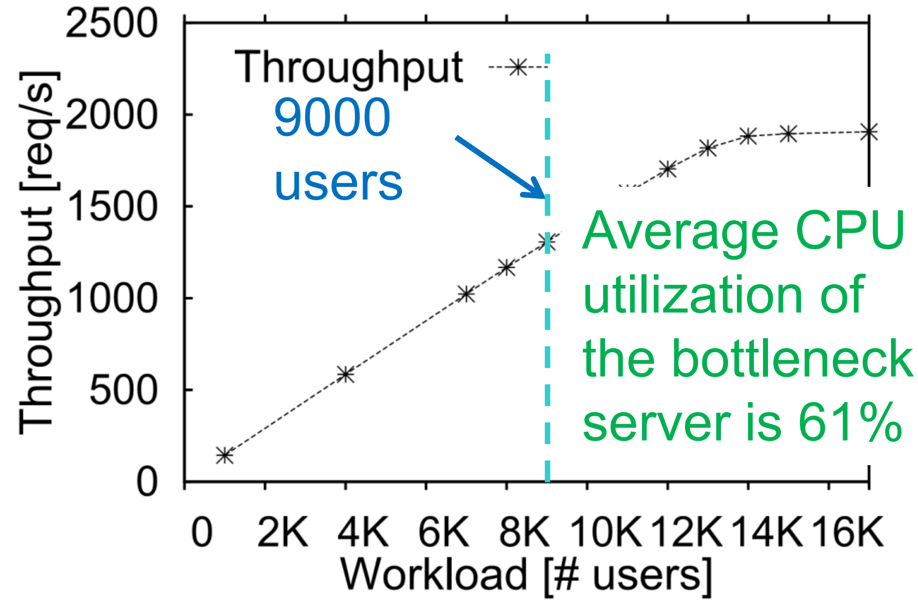


- Microsecond-resolution timestamps on messages
- 50-millisecond resolution on resource utilization sampling

Measured Average System Performance

- Response time & throughput of a 3-minute benchmark on the 4-tier application with increasing workloads.

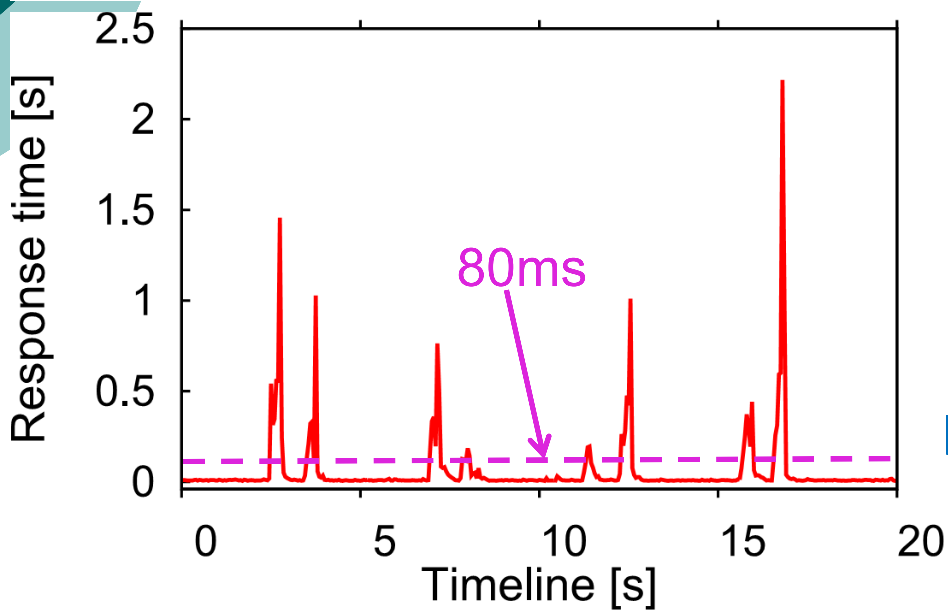




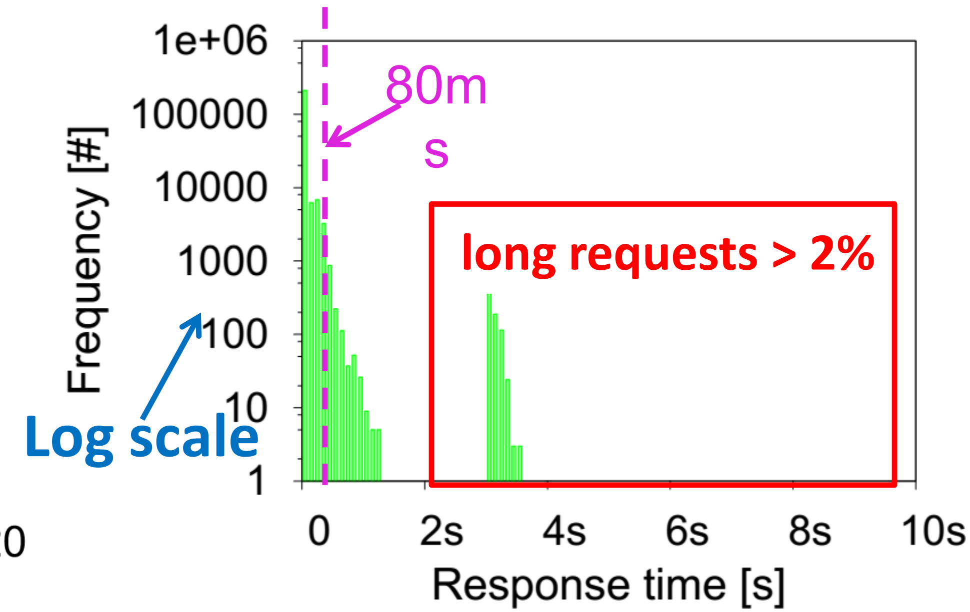
Average system response time is low at workload 9000 users, how about Point-In-Time response time?

VLRT Requests (Case 1: JVM GC)

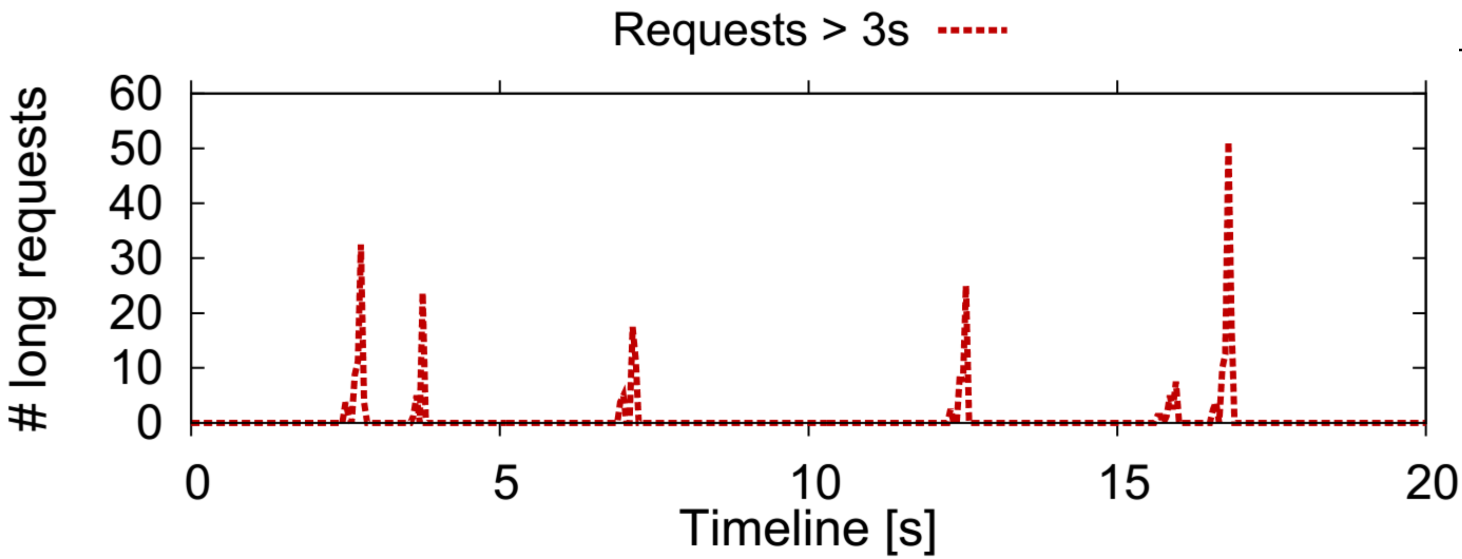
P-I-T Response time at 9000 users

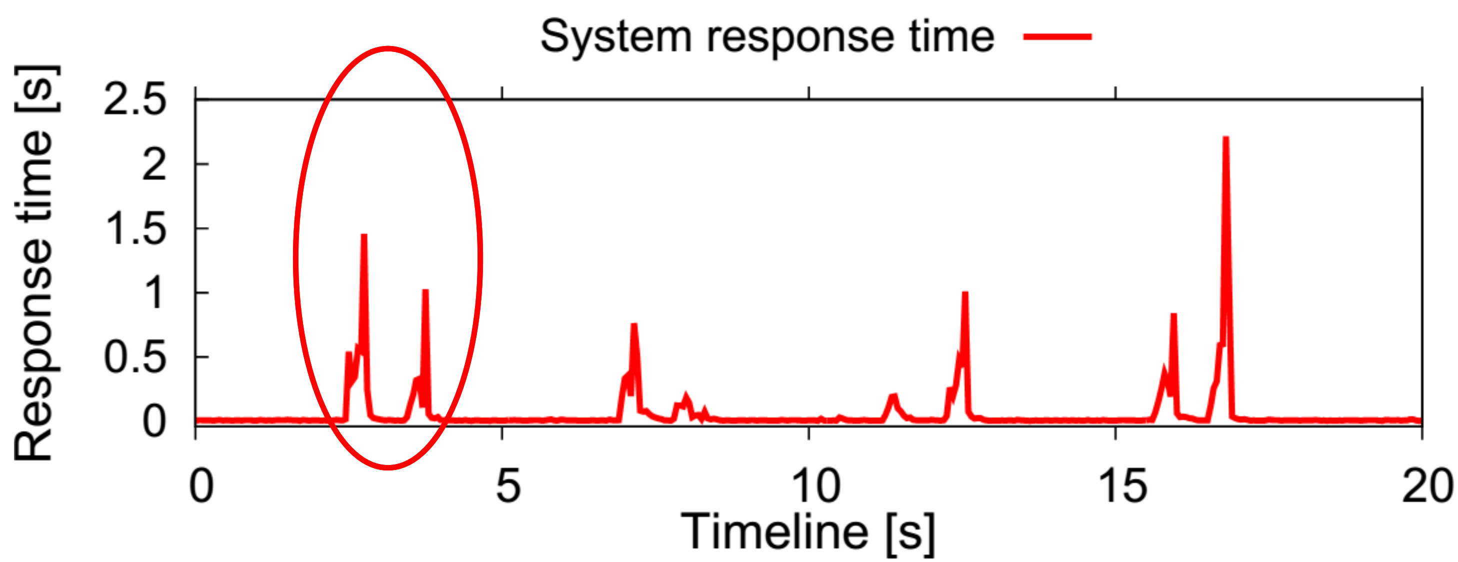


Request response time distribution at 9000 users

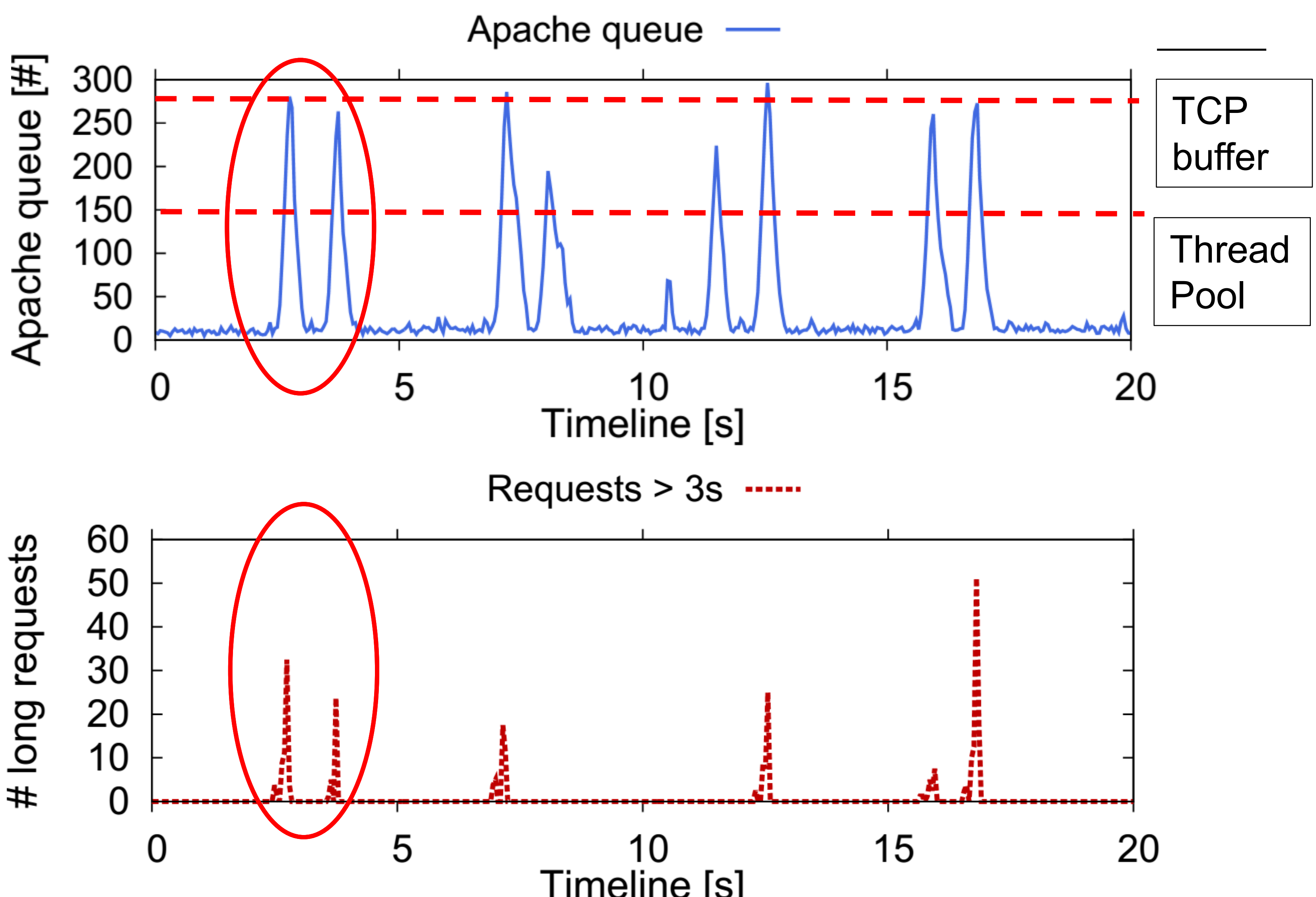


Dropped Packets/Requests ⇒ VLRT Requests

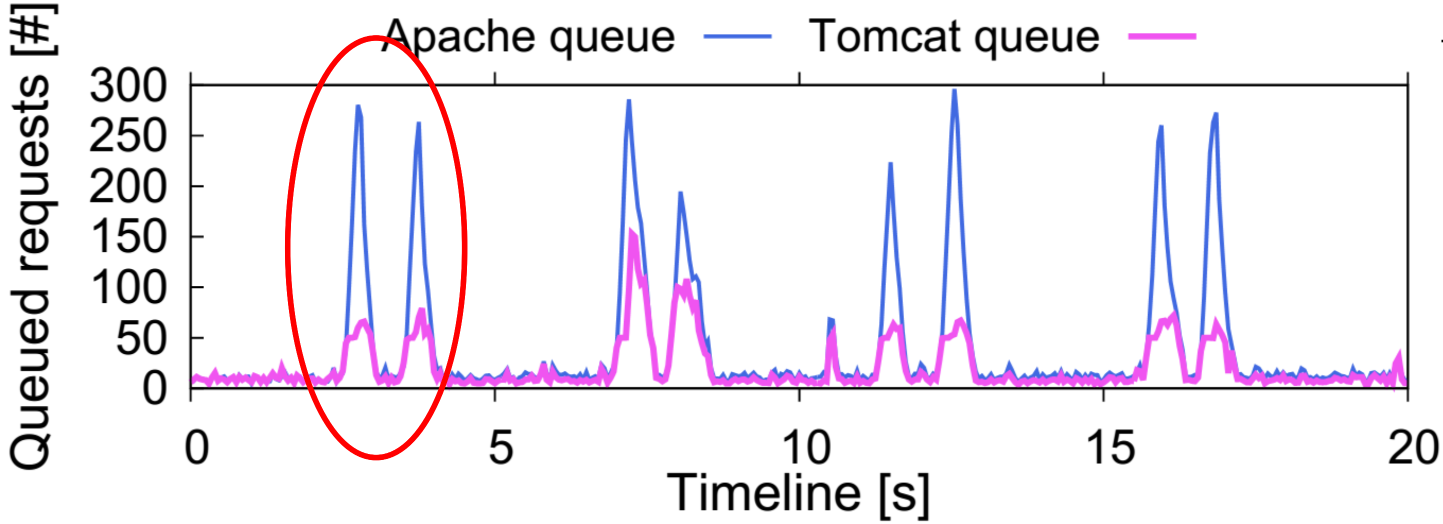


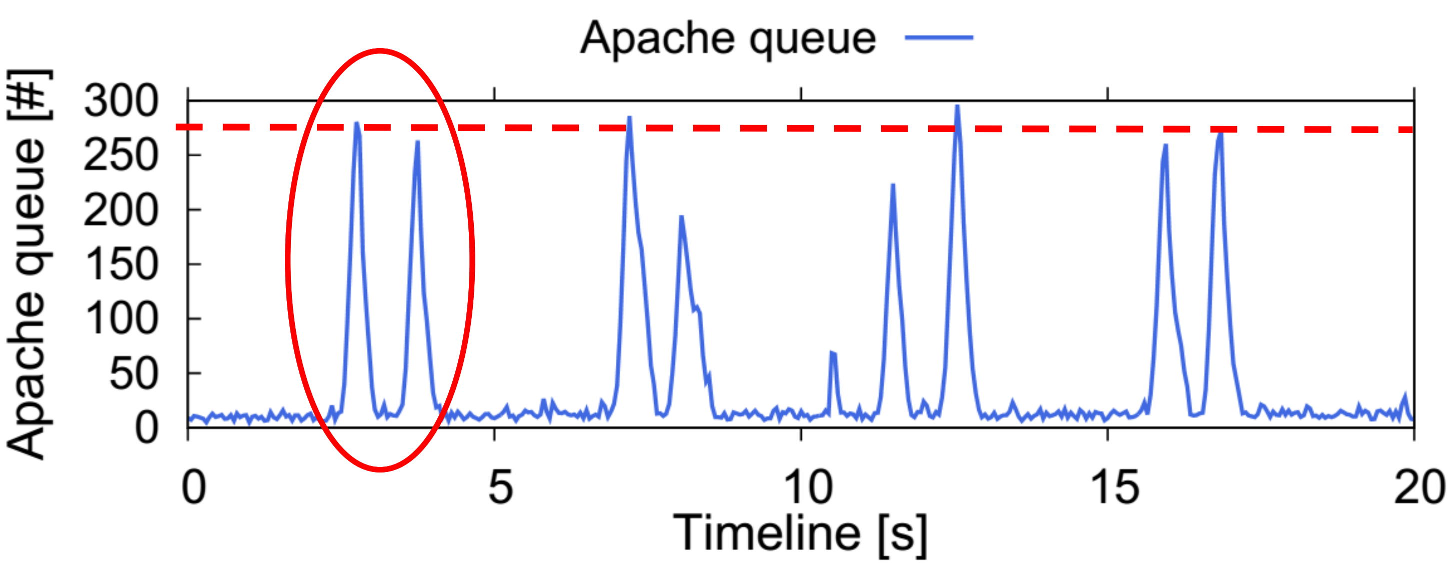


Queue Overflow ⇒ Dropped Packets

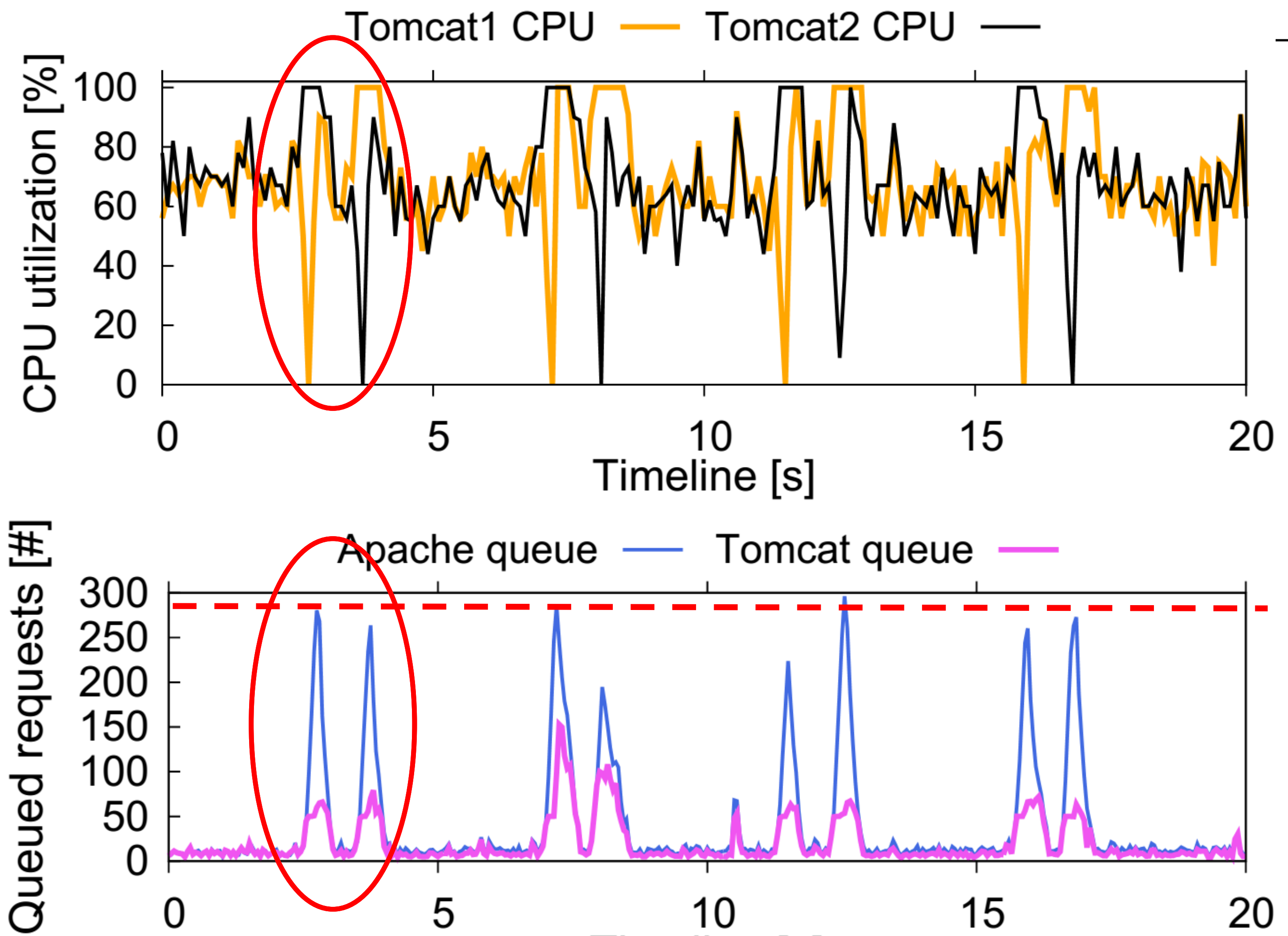


Queue Amplification ⇒ Queue Overflow

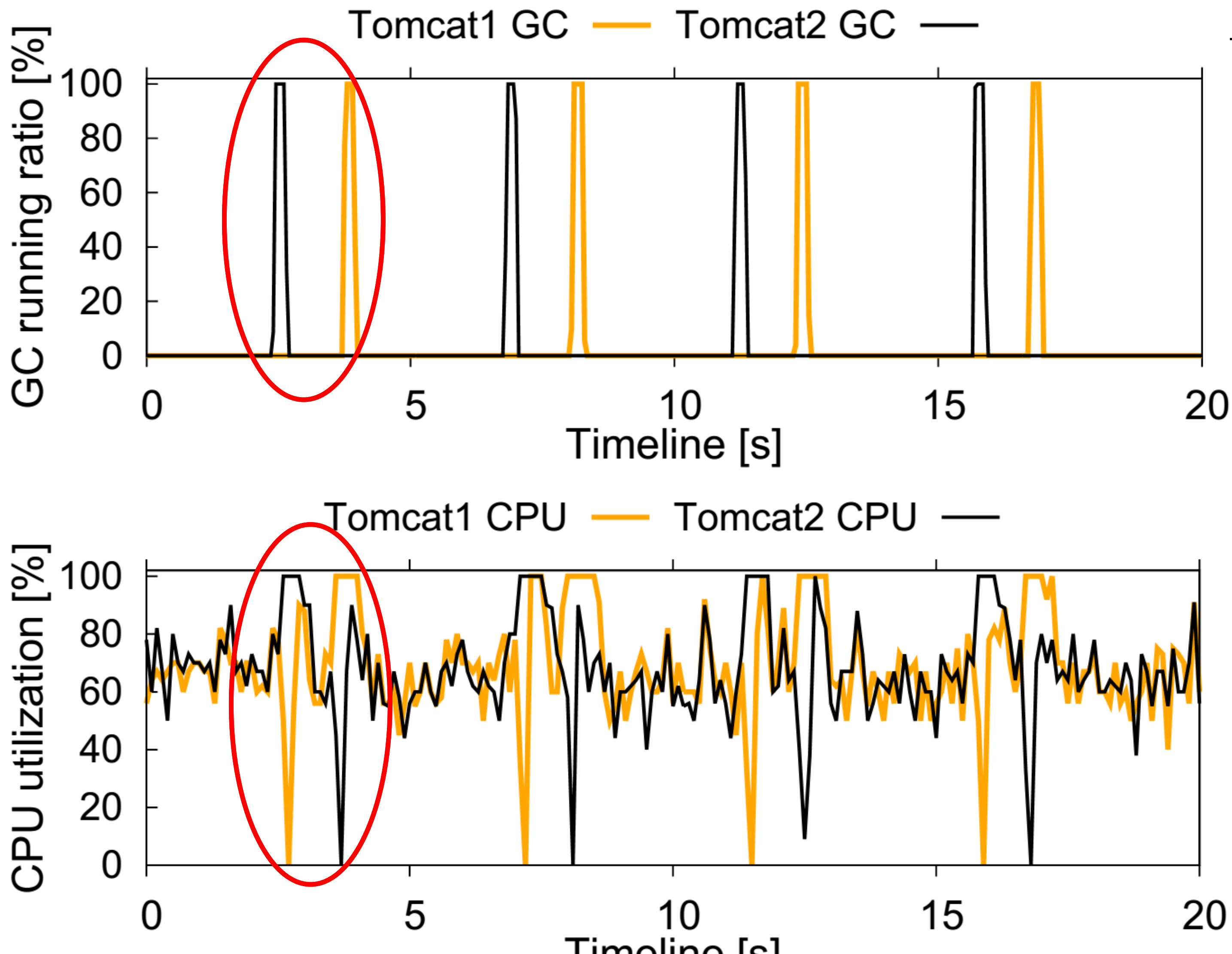




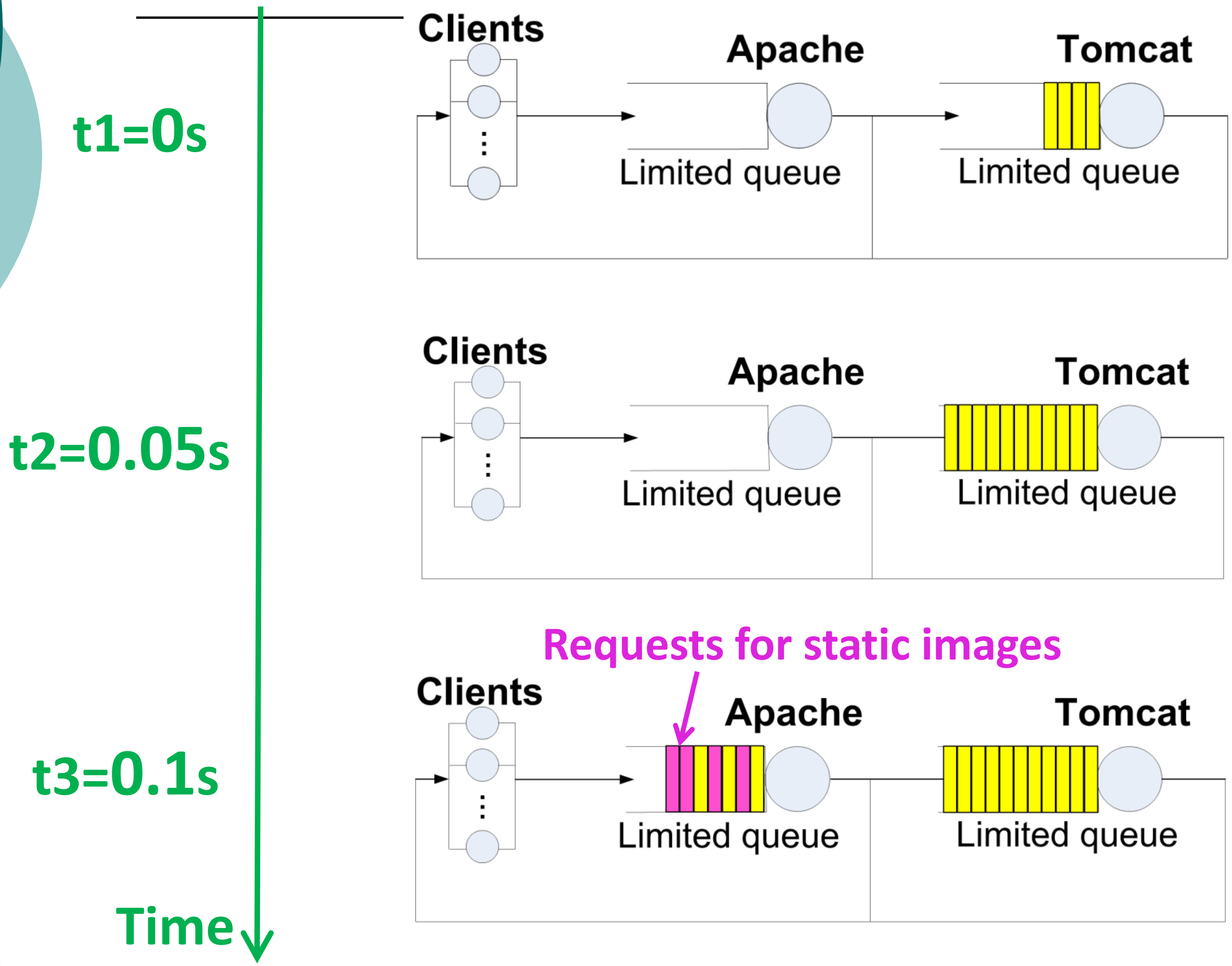
Very Short Bottlenecks ⇒ Queue Amplification



JVM Garbage Collection ⇒ Very Short Bottlenecks



Upstream Queue Amplification

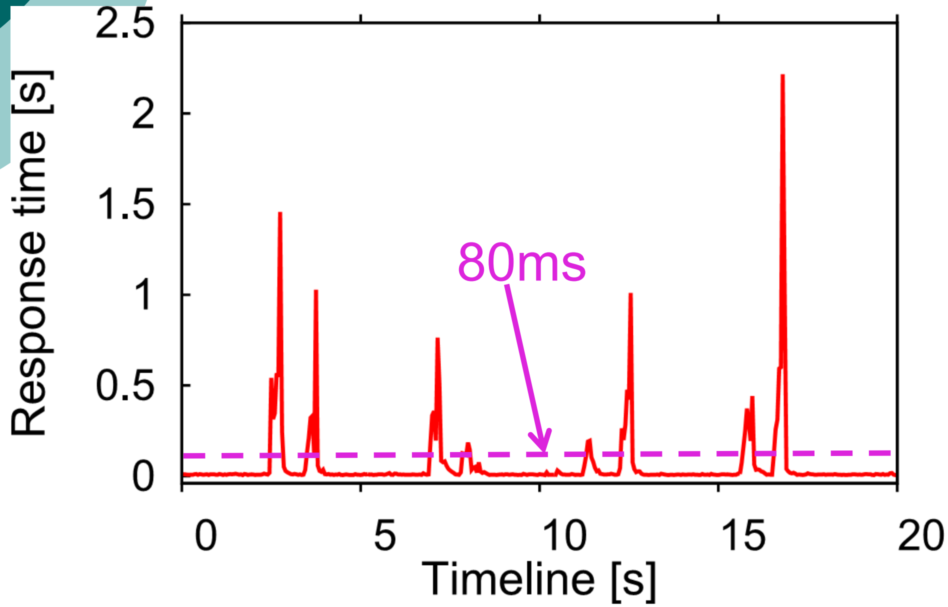


Impact of VSBs

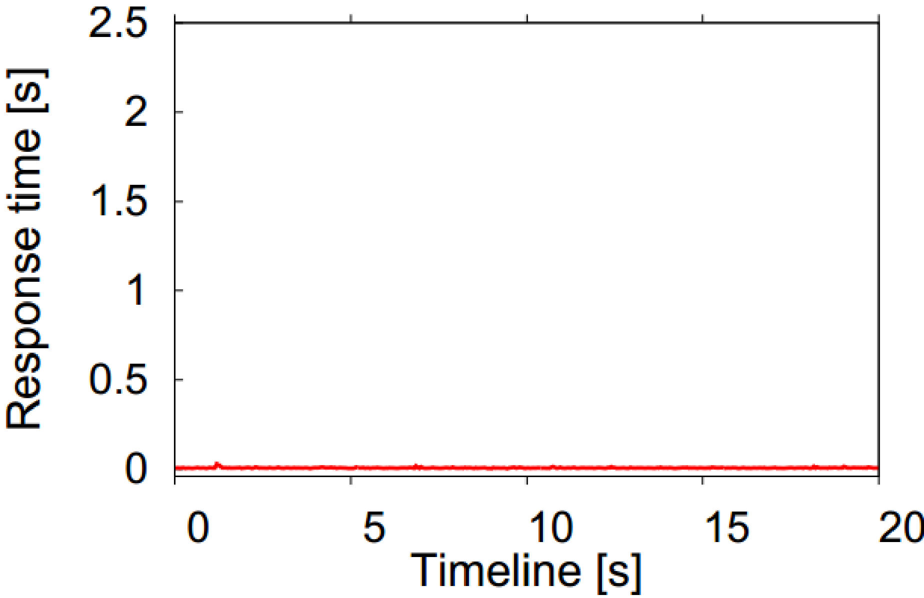
- Very short bottlenecks (<100ms) cause VLRT requests (>3sec)
  - Cannot be avoided if bursty workload
  - Often start at less than 50% average utilization
- Caused by queue amplification
  - Queuing in upstream tiers
  - Dropped packets when queues are full
- JVM GC was “fixed” in JVM 1.6

Java GC VSB Resolved

JDK1.5 case in Tomcat



JDK1.6 case in Tomcat

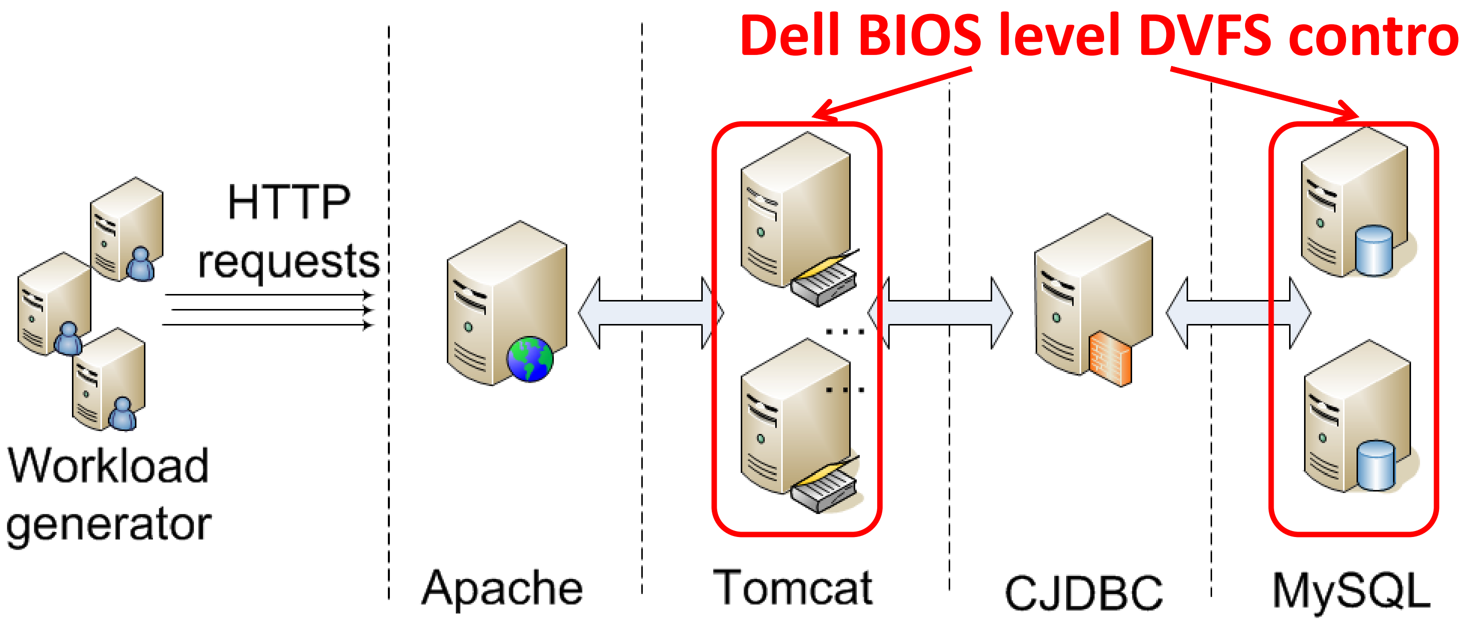


P-I-T Response time of system at 9,000 users

Case 2: DVFS

- Dynamic Voltage and Frequency Scaling (DVFS) adjust CPU voltage/frequency on-demand
  - Designed to save power when workload fluctuates
  - Should be good for bursty workloads
- Problem: anti-synchrony between CPU requirement and DVFS adjustment can cause very short bottlenecks
  - This happens when workload burst length nears DVFS control period (e.g., 500ms)

DVFS Experimental Setup

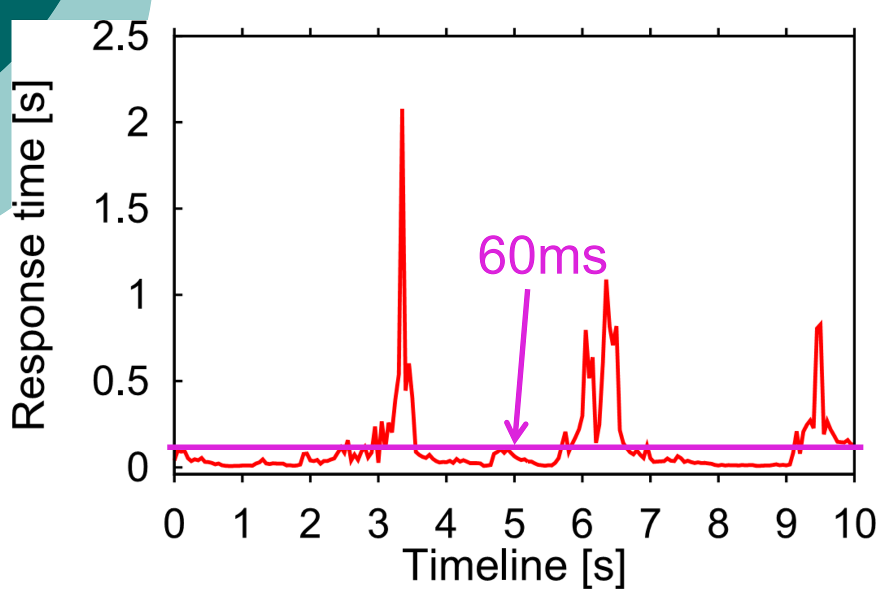


- ❑ RUBBoS benchmark: a bulletin board system like Slashdot (www.slashdot.org)
- ❑ Workload (number of emulated users)
  Browse-only workload (CPU intensive)
  Naturally bursty

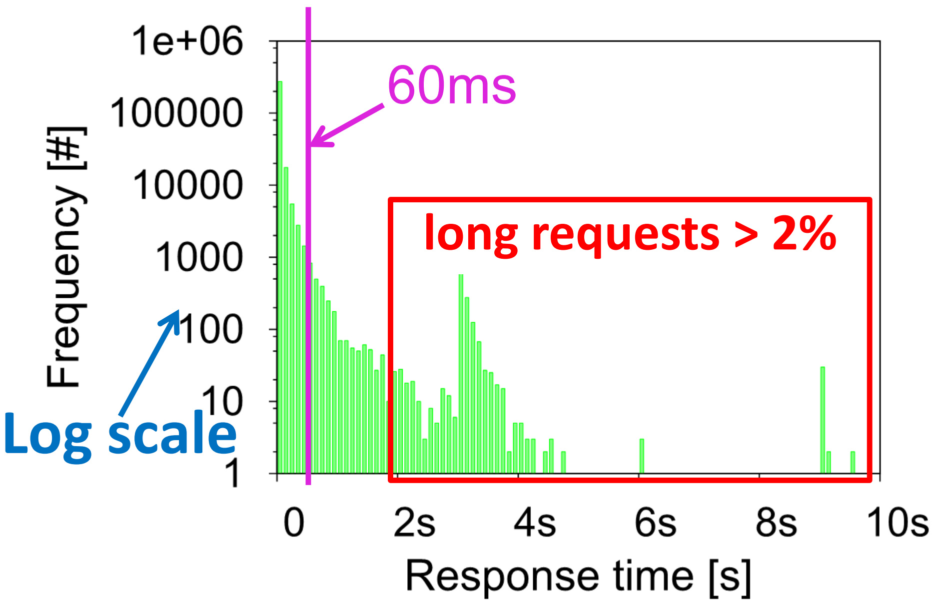
- ❑ Intel Xeon E5607
  2 quad-core 2.26 GHz
  16 GB memory
- ❑ **Support P0~P8**
  P0: (2.26GHz/1.35v)
  P8: (1.12 GHz/0.75v)

VLRT Requests

P-I-T Response time at 12000 users

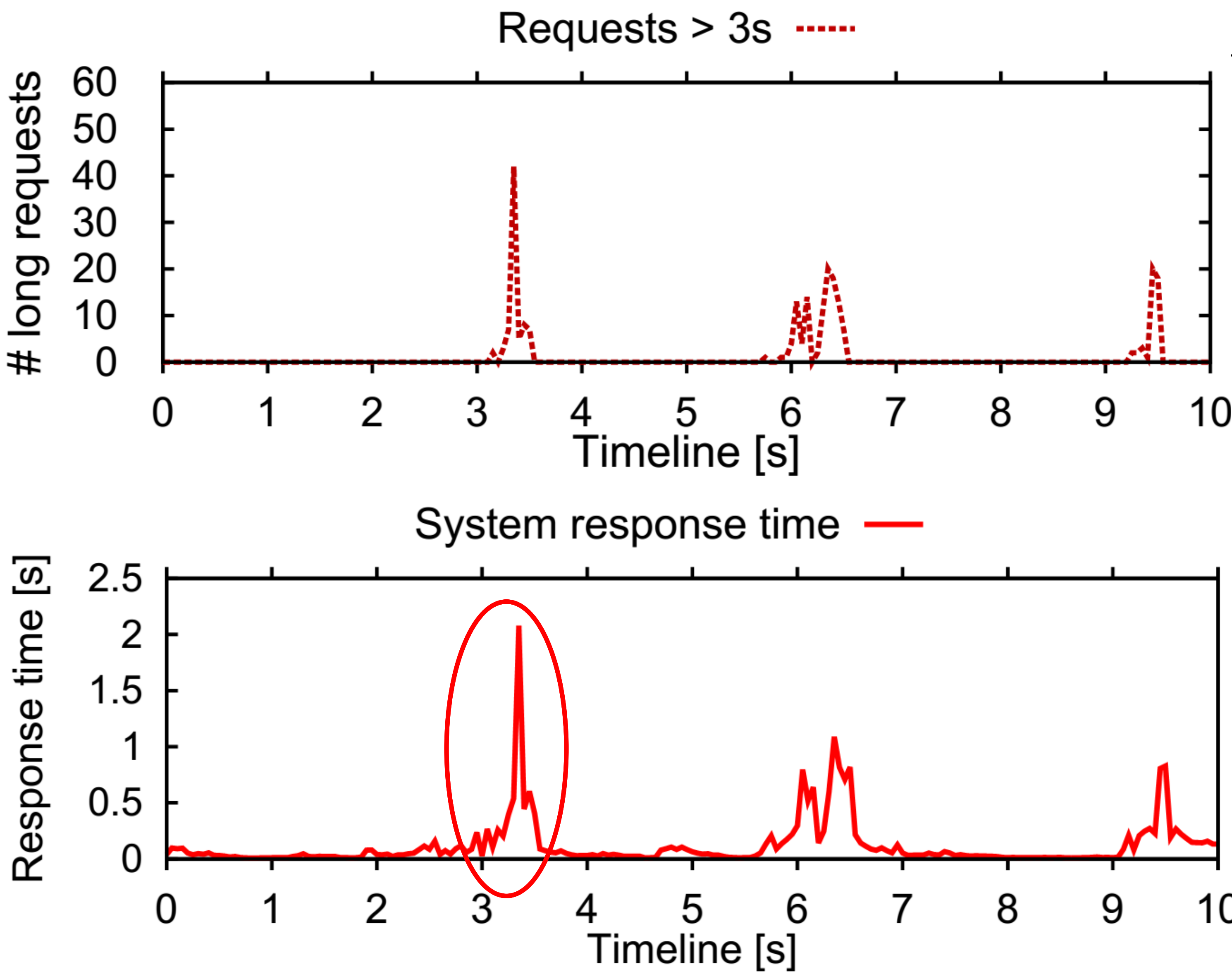


Request response time distribution at 12000 users

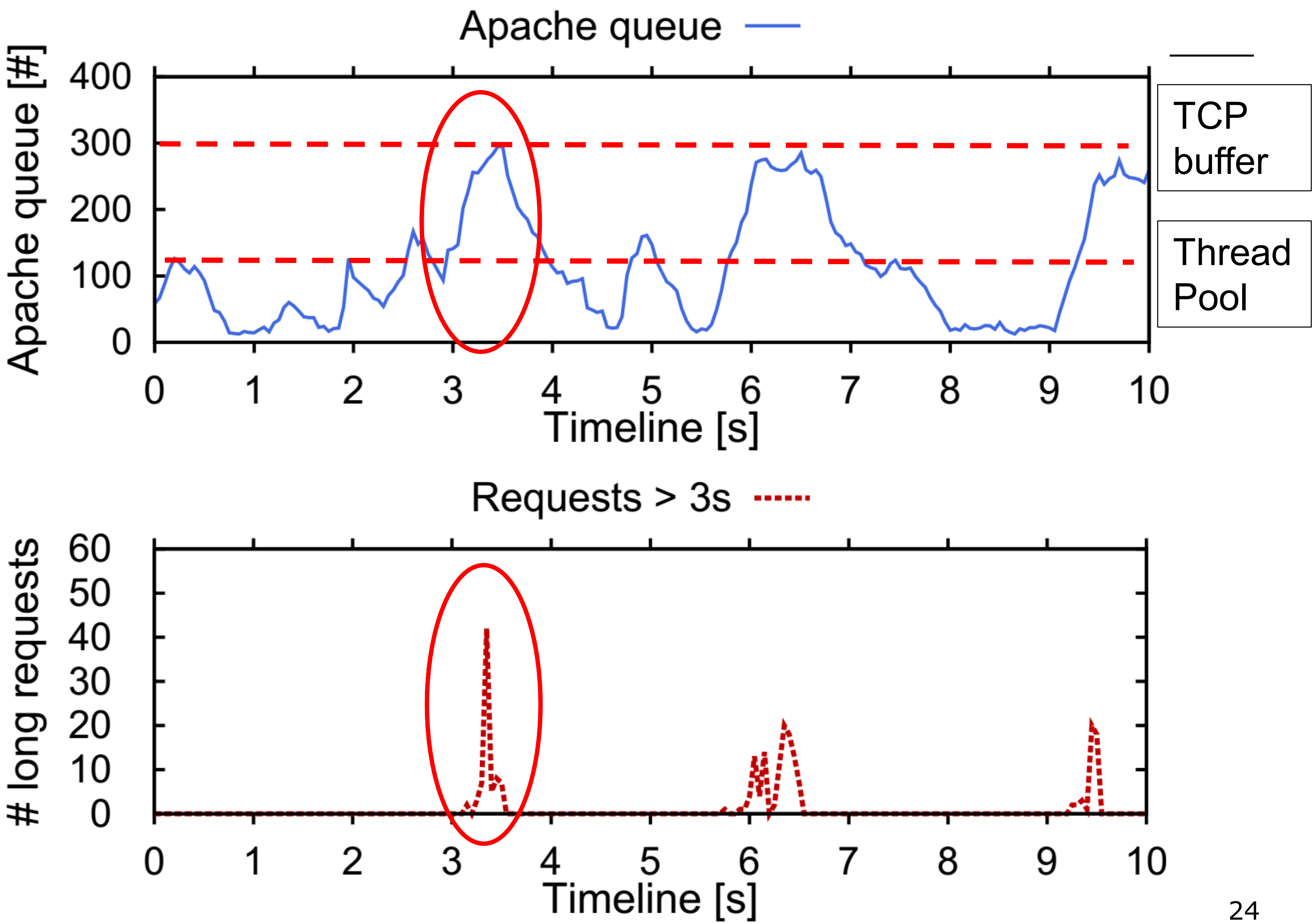


Average system response time is 60ms, and Average CPU utilization of the bottleneck server is 78.7%.

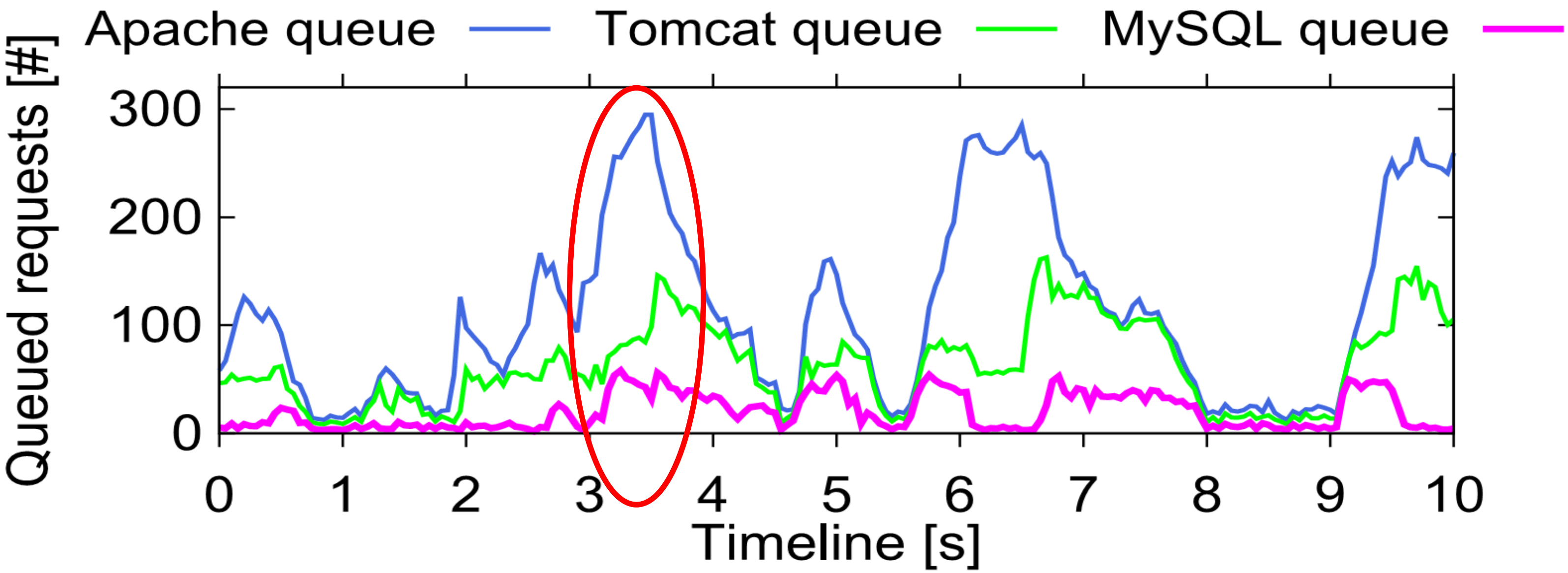
Dropped Packets ⇒ VLRT Requests

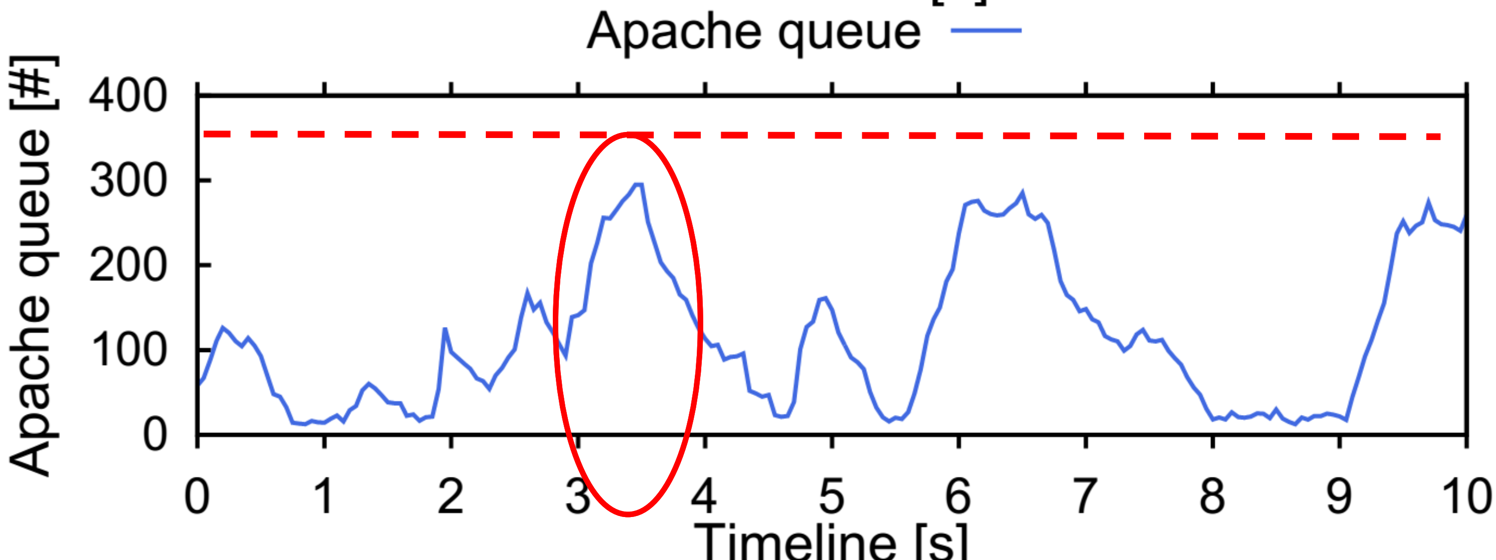


Queue Overflow ⇒ Dropped Packets

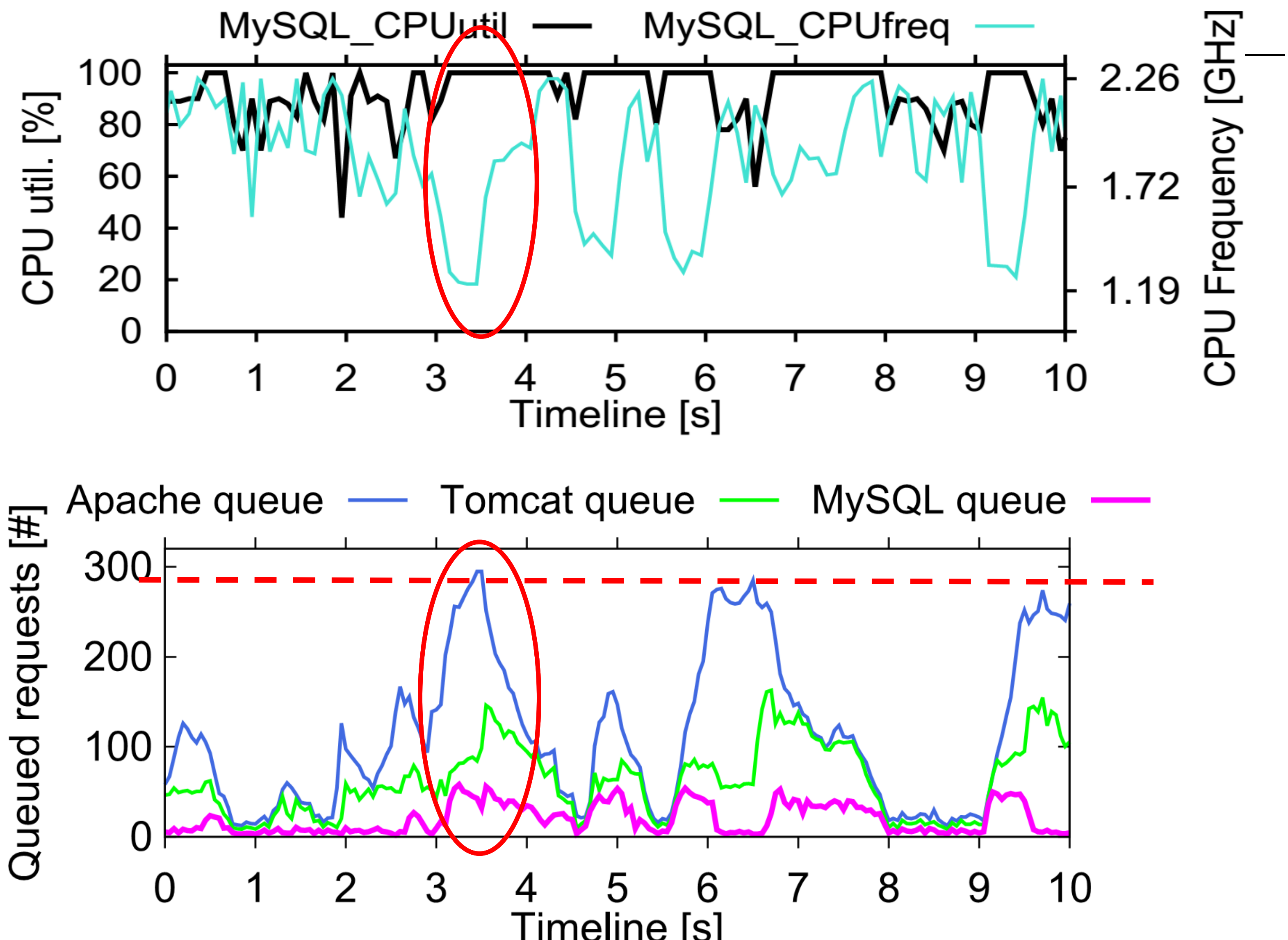


Queue Amplification ⇒ Queue Overflow



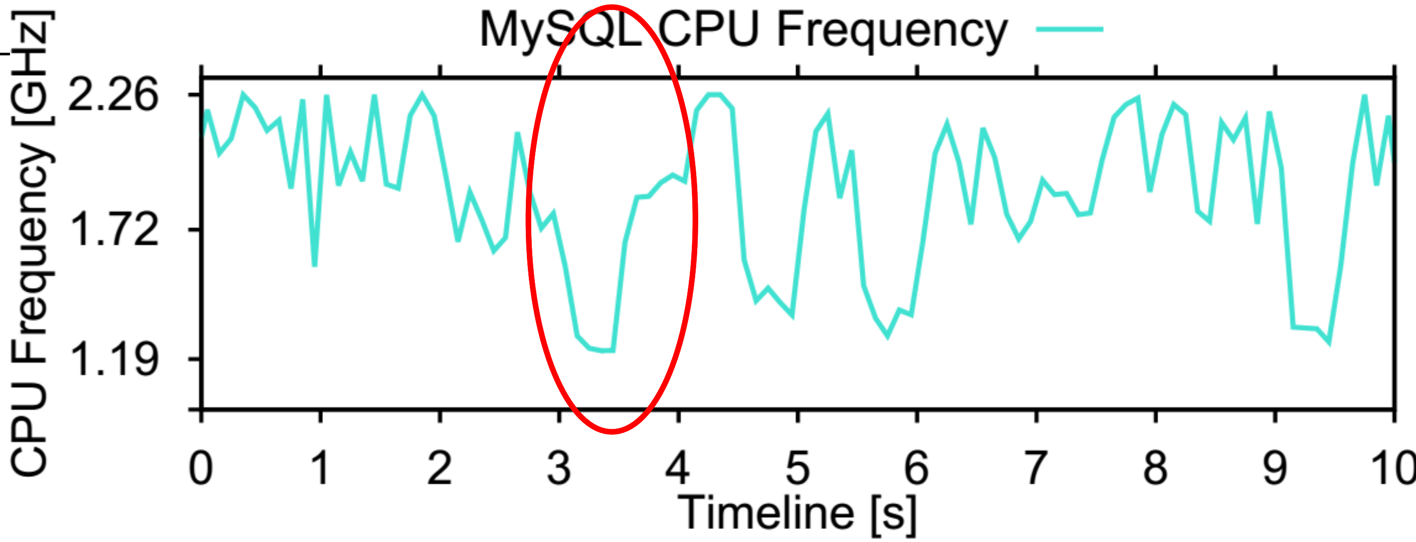


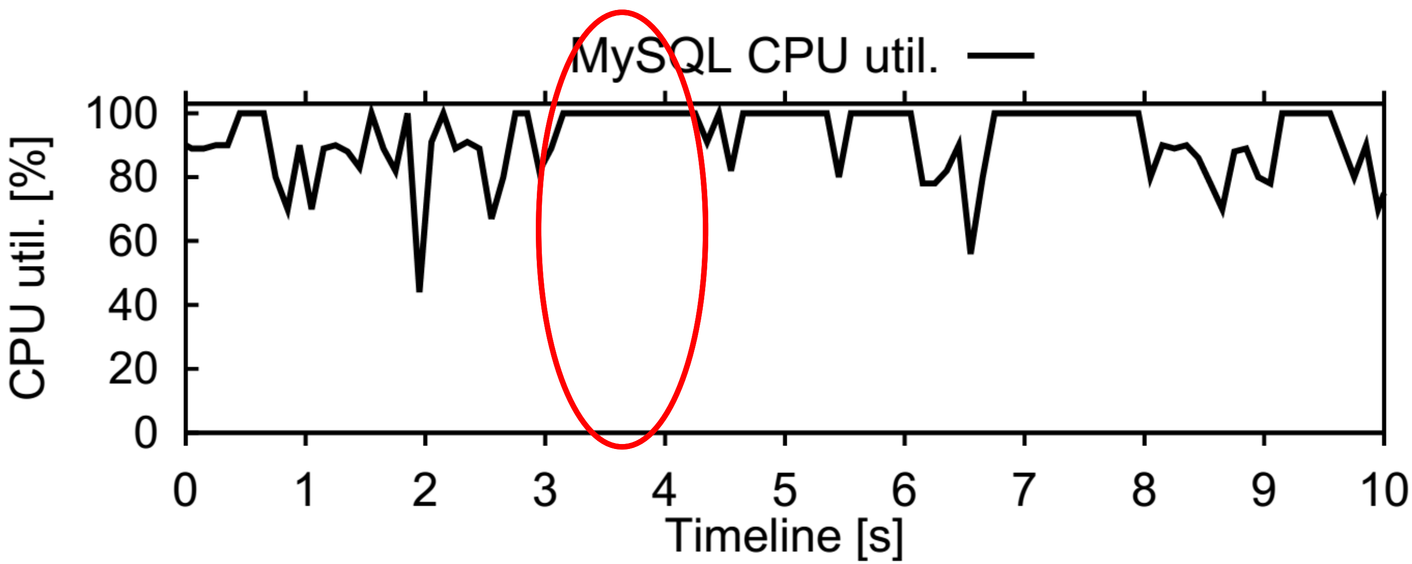
Very Short Bottlenecks ⇒ Queue Amplification



Anti-Synchrony in DVFS ⇒ Very Bottlenecks

MySQL at 12000 users



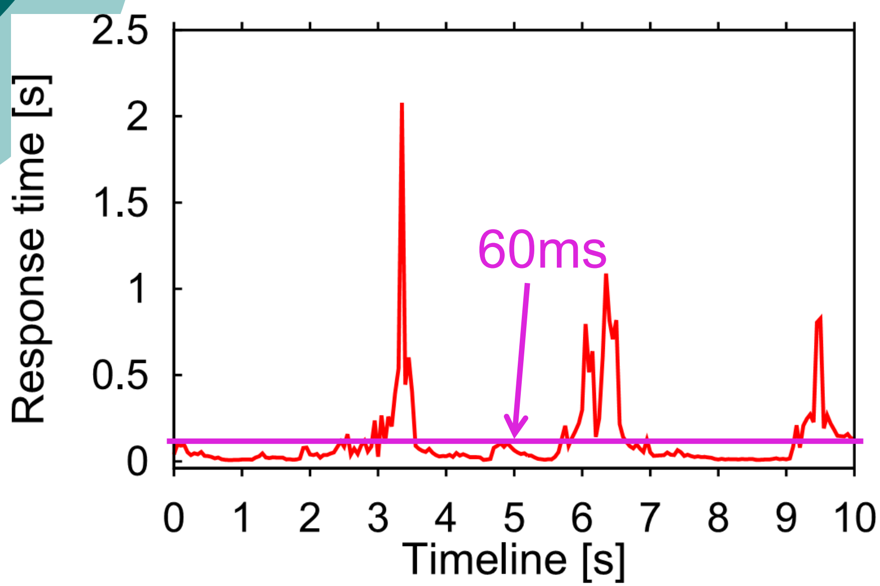


Note on DVFS Experiments

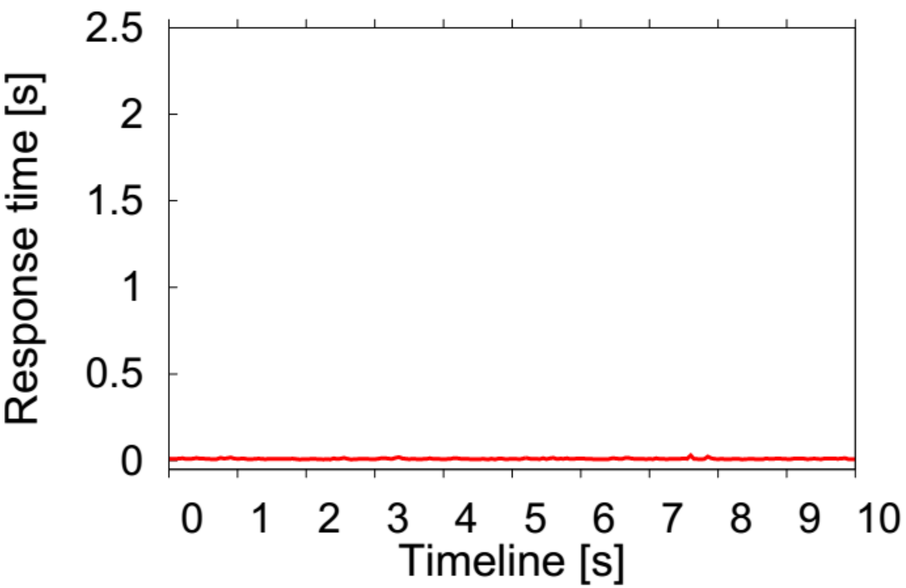
- JVM garbage collection episodes are deterministic (in time)
  - Very short bottlenecks and VLRT requests are deterministically reproducible
- DVFS control periods are less deterministic (in time)
  - Very short bottlenecks and VLRT requests are reliably reproducible whenever DVFS anti-synchrony happens

DVFS Anti-Synchrony Can Be Resolved (by Turning It Off)

DVFS-On



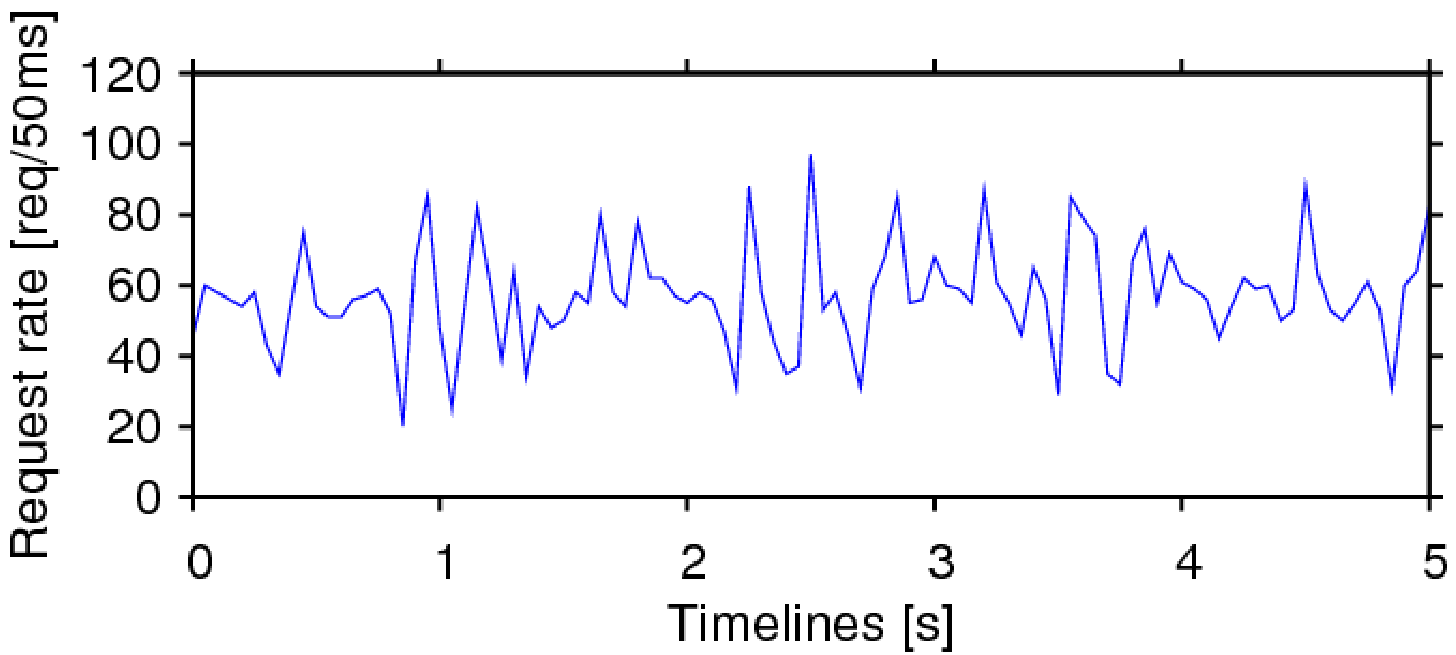
DVFS-Off



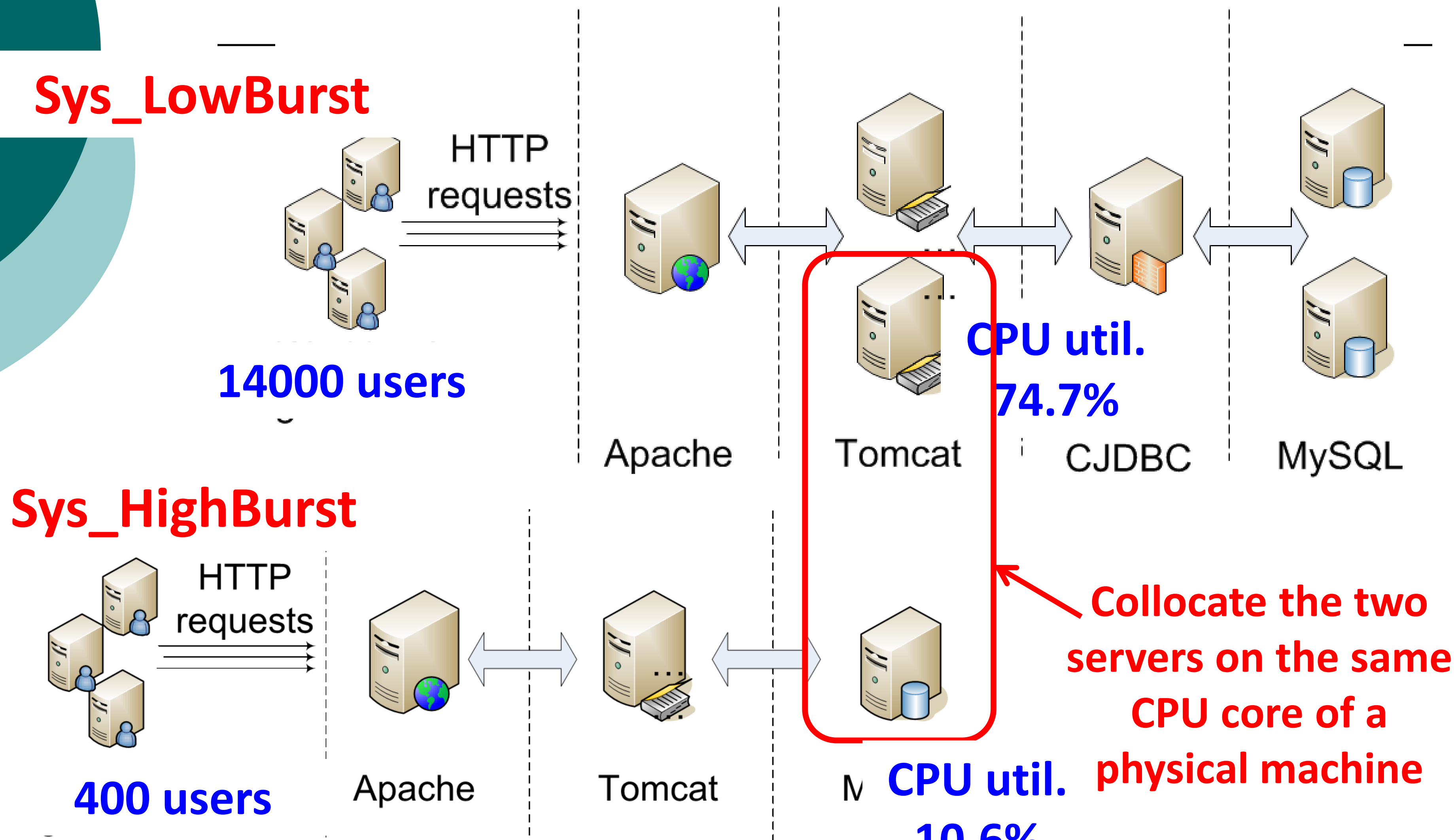
P-I-T Response time of system at 12,000 users

Case 3: VM Consolidation

- Consolidating VMs is one way to increase hardware utilization
  - Workloads for web applications are **naturally bursty** *[Mi,Middleware'08]*
  - Sharing is better than isolation *[Kanemasa, SCC'13]*



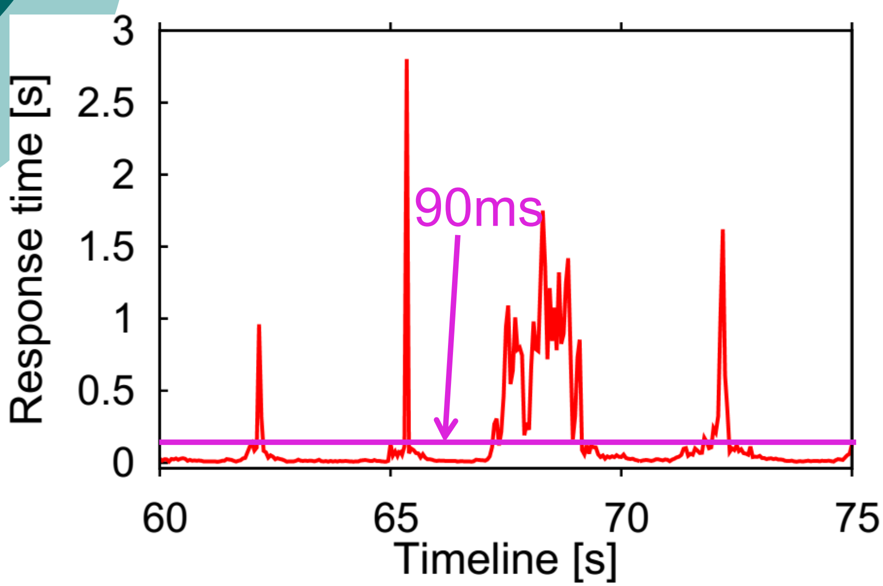
Consolidation Setup



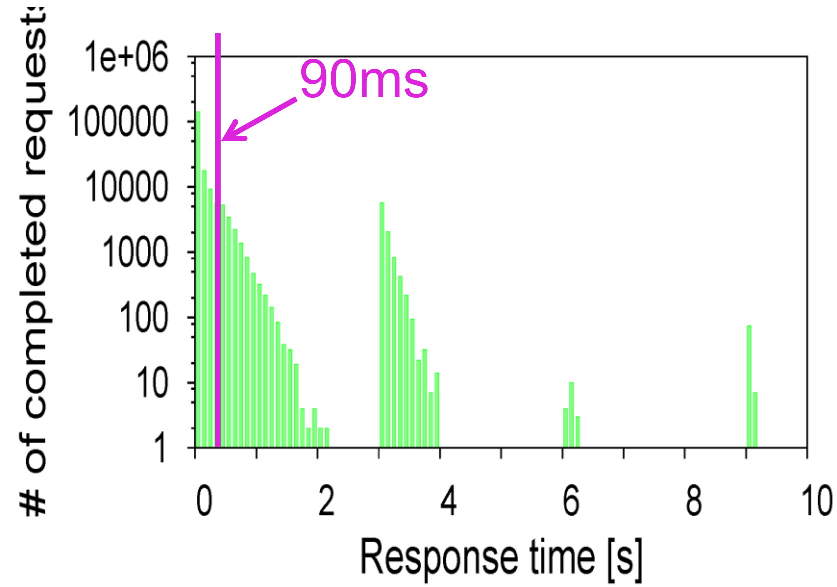
JVM 1.6, DVFS off

VLRT Requests (Measured in Sys-LowBurst)

P-I-T Response time of Sys_LowBurst



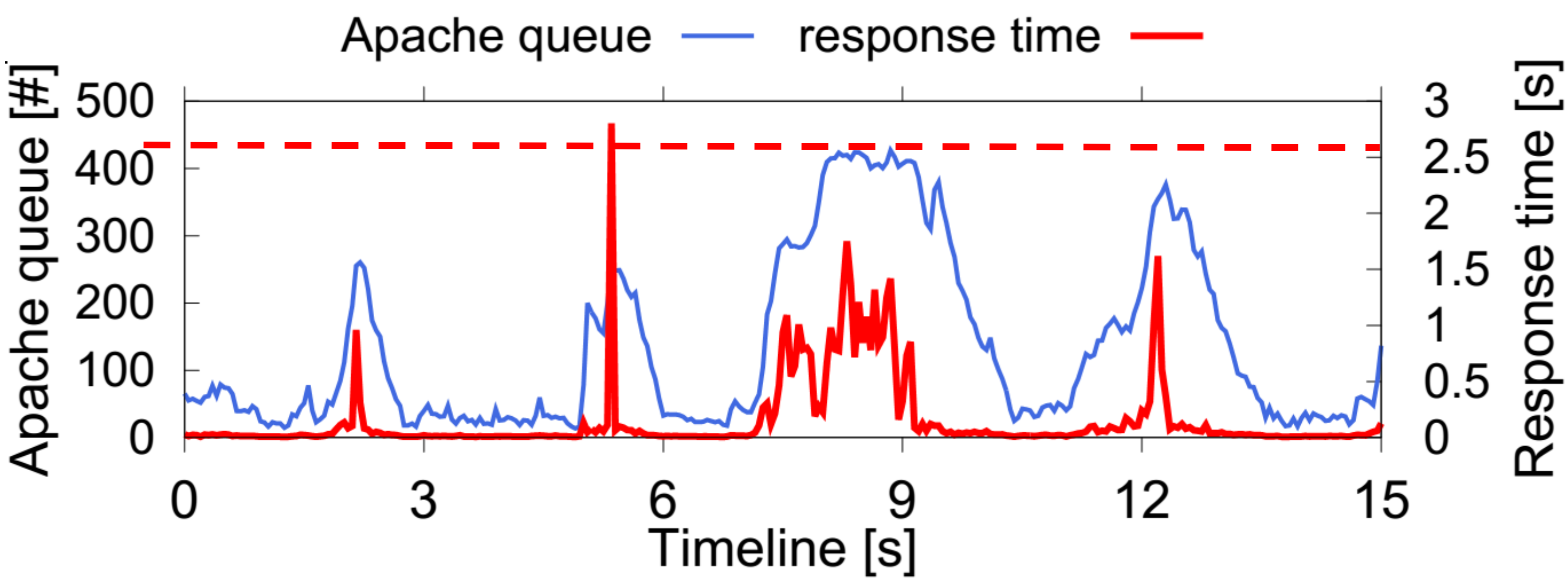
Request response time distribution of Sys_LowBurst



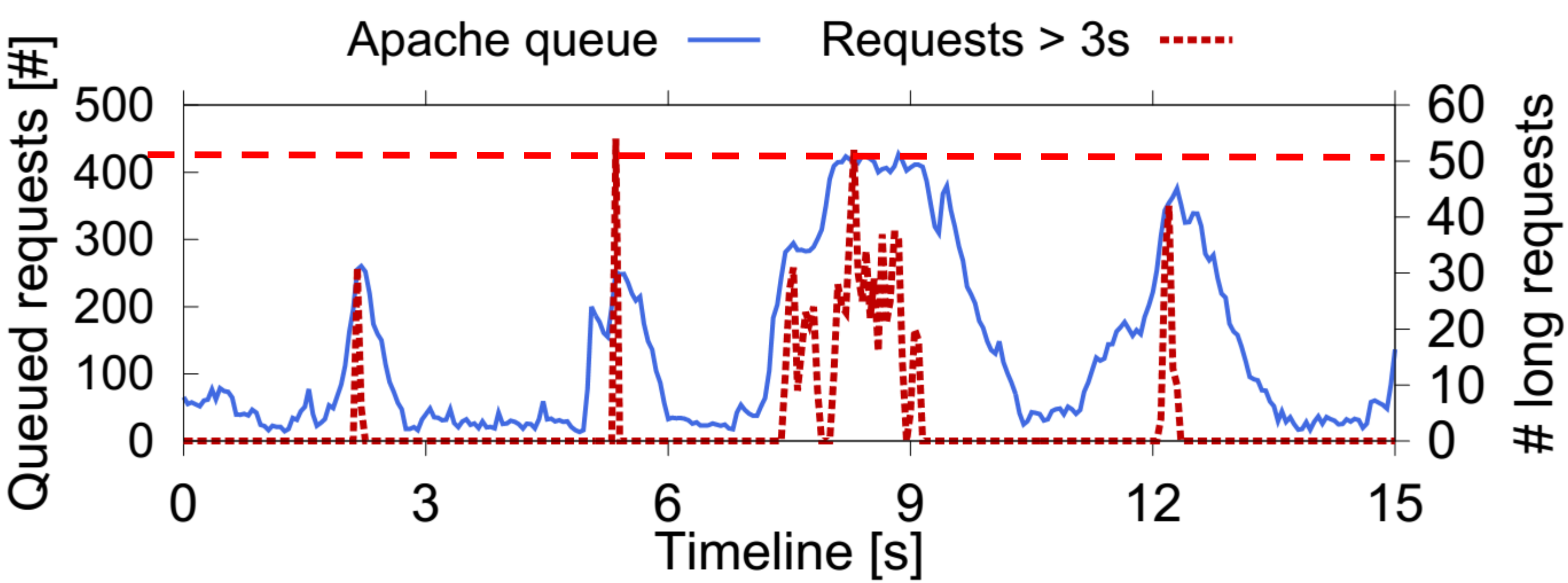
Average system response time is 90ms, and Average CPU utilization of the bottleneck server is 74.7%.

Queue Overflow/Dropped Packets ⇒ VLRT Requests

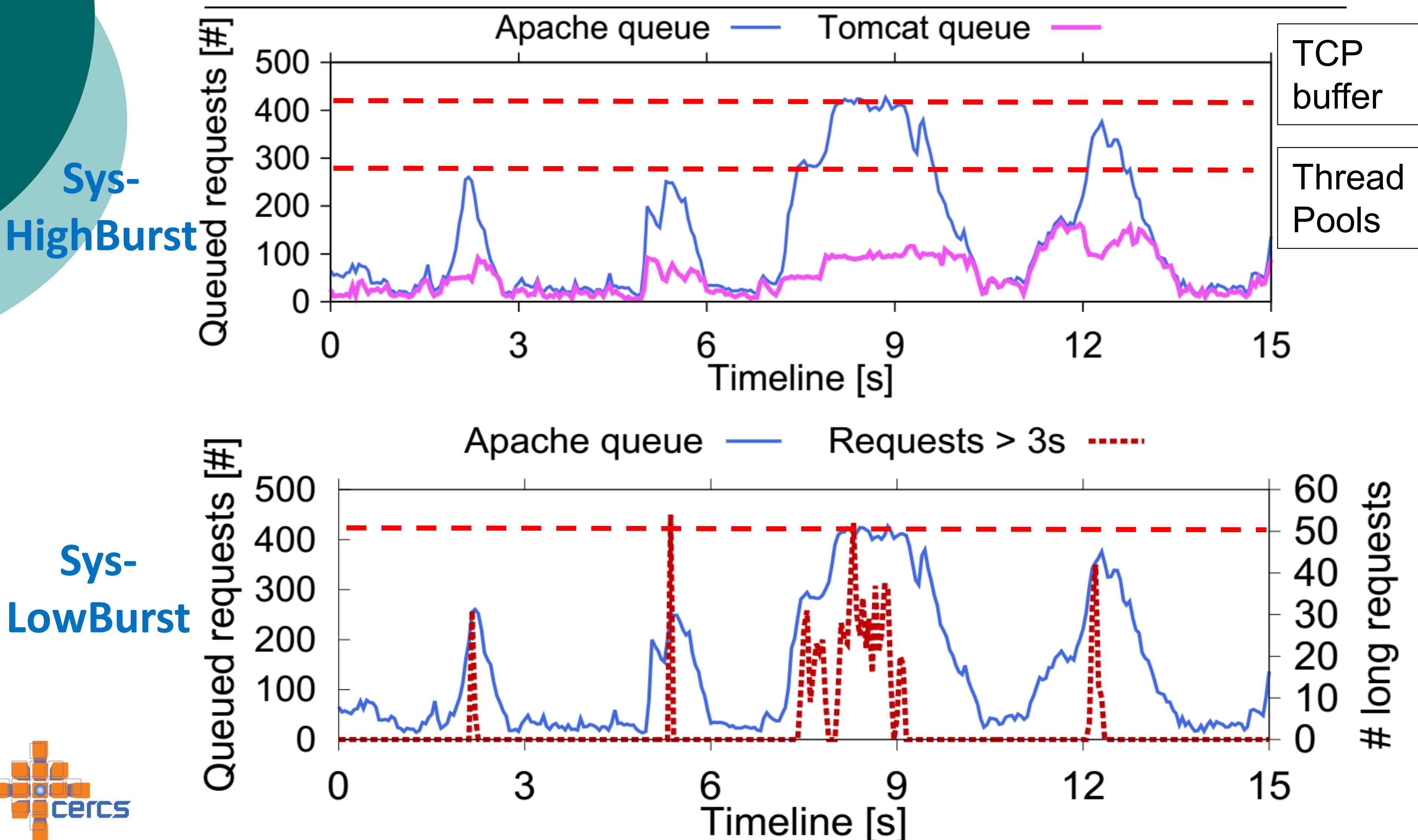
Sys-LowBurst



Sys-LowBurst

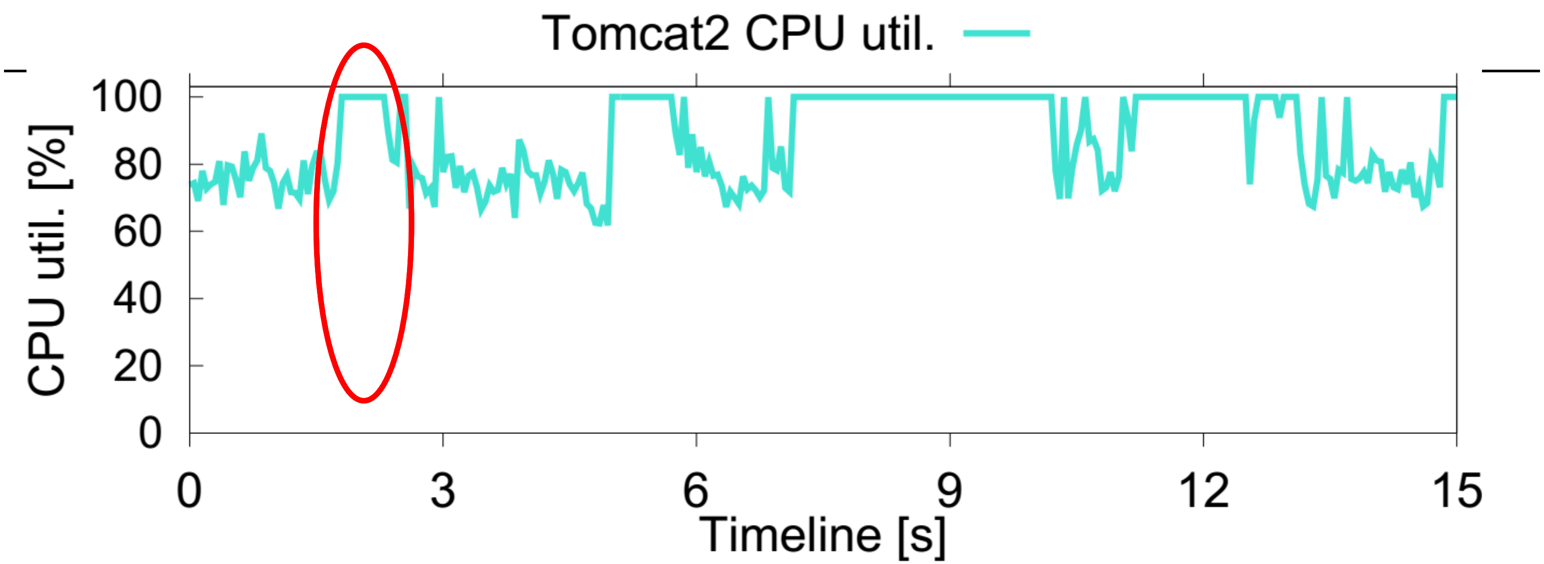


Queue Amplification ⇒ Queue Overflow

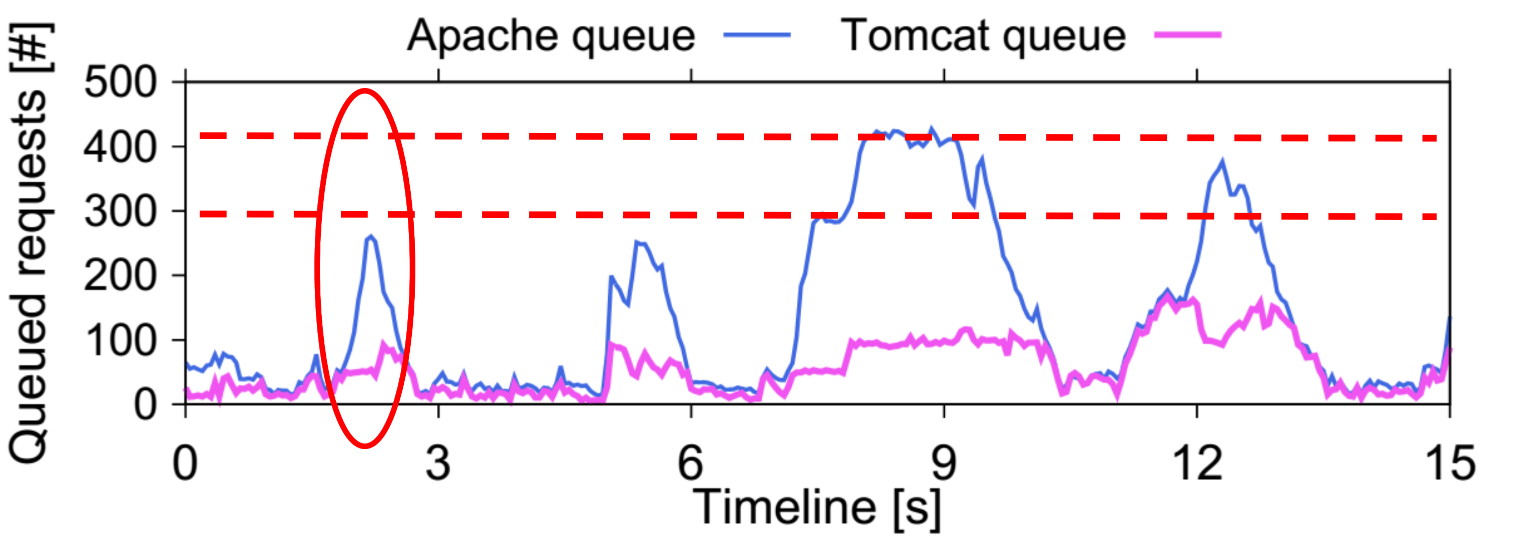


Very Short Bottlenecks ⇒ Queue Amplification

Sys-LowBurst

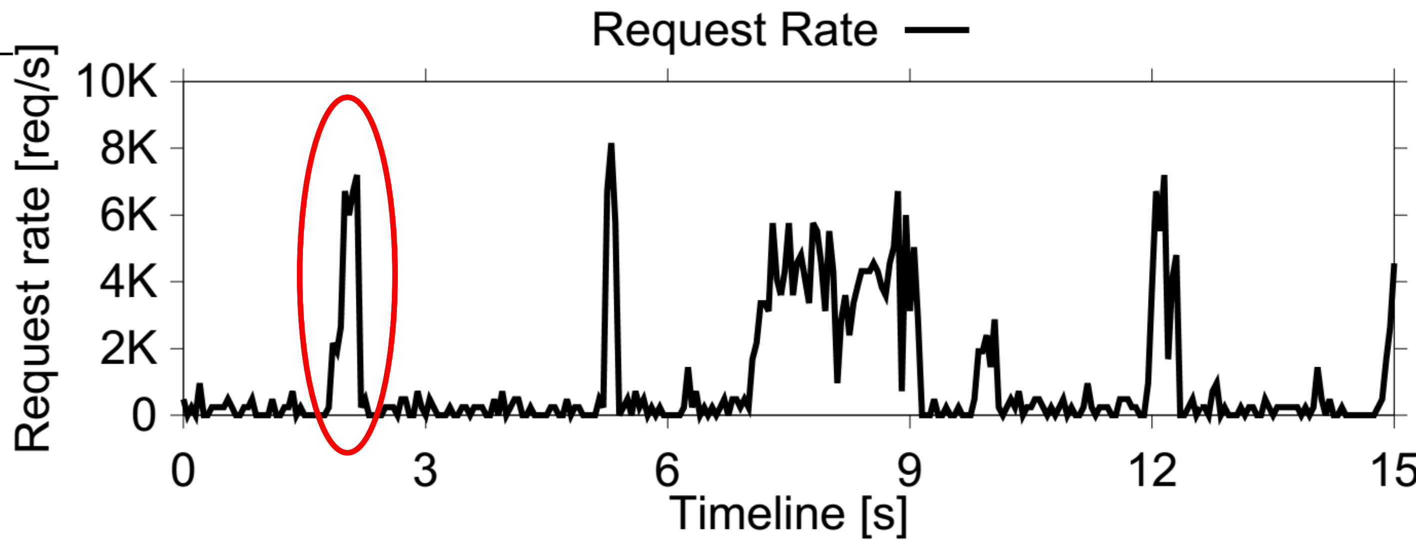


Sys-HighBurst

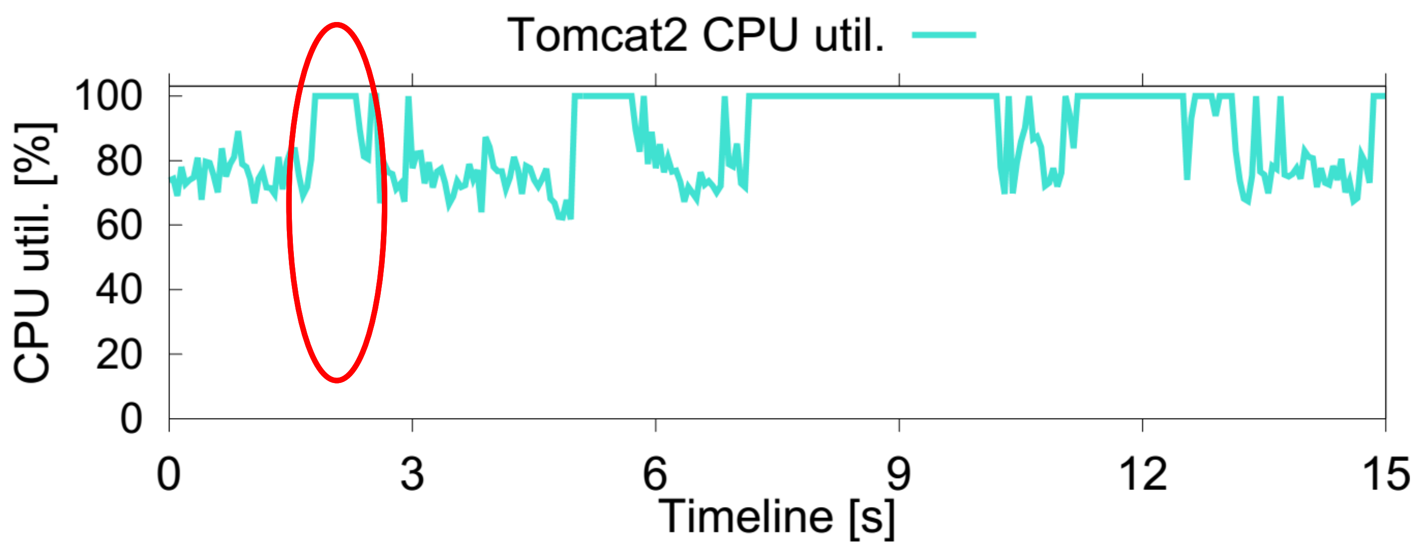


Overlap in VM Workload Bursts ⇒ Very Short Bottlenecks

Sys-HighBurst



Sys-LowBurst

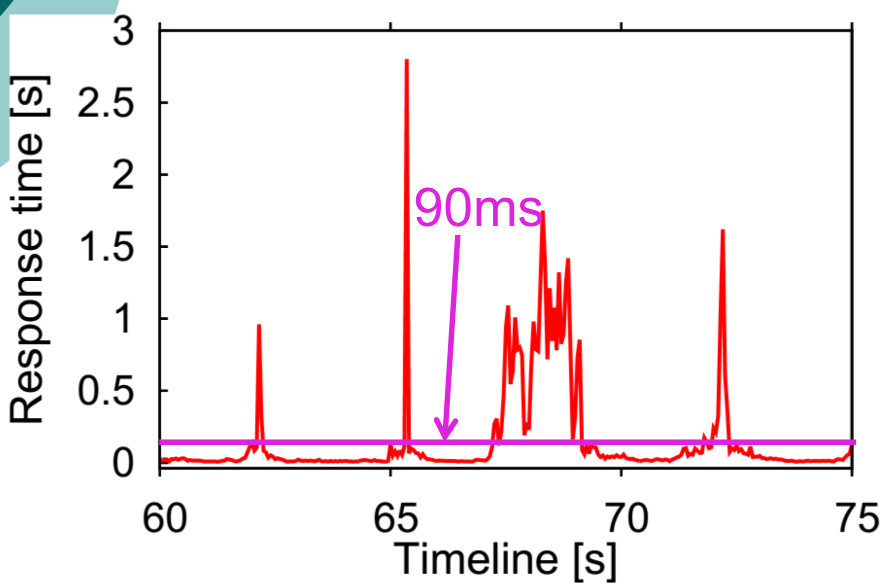


Note on VM Consolidation

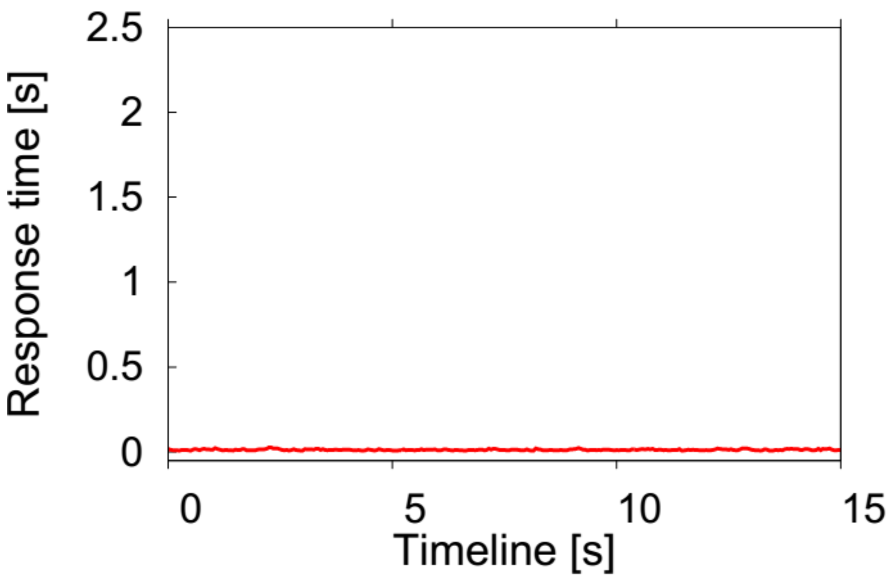
- Each experiment creates different bursts
  - Workload burst overlaps are not exactly the same (may differ in time and duration)
- Nevertheless, measurable overlaps are reliably reproducible (statistics)
  - Very short bottlenecks are reliably associated with the overlaps (whenever they happen)
  - VLRT requests are reliably reproducible, associated with the very short bottlenecks

Stable Response Time without Collocation

Sys-original **with** collocation



Sys-original **without** collocation



P-I-T Response time of system at 14,000 users

Very Short Bottleneck Summary

- Very short bottlenecks happen in different system layers
  - System software: Java garbage collection
  - Processor architecture: DVFS
  - Application Virtual Machine consolidation
- Though **short-lived**, very short bottlenecks have **big impact** on n-tier application performance
  - VLRT requests
  - Queue amplification from n-tier system component dependencies

Discussion on Solutions

- Three kinds of solutions for Latency Long Tail Problem

1. Bug-fix, specific solutions for each case
   - There are many cases/sources of VSBs
2. General solutions for VSBs
   - Current research
3. Last-resort solution
   - Current state-of-art

Last-Resort Solution

- Zero knowledge on causes; just maintain very low utilization on all resources (CPU)
  - Currently the most popular solution
  - Gartner reports on average data center server utilization: 18%; other reports as low as 6%
  - Google reports 30% (including batch jobs)
- Problematic in the long term
  - Obviously not ideal for high ROI (or low cost)
  - A “safe” utilization cap depends on many factors, including burstiness of workload

Research Challenges in Cloud Resource Management

- A challenging problem: latency long tail
  - Very long response time (VLRT) requests
  - Difficult to reproduce, almost invisible
  - We found 3, but ***there are many more***
- ROI can be improved (a lot) for clouds
- Costs can be improved (a lot) for truly large scale deployments (e.g., NFV)