TailBench: A Benchmark Suite and Evaluation Methodology for Latency-Critical Applications

Harshad Kasture, Daniel Sanchez

IISWC 2016

tailbench.csail.mit.edu

Massachusetts Institute of Technology

Executive Summary

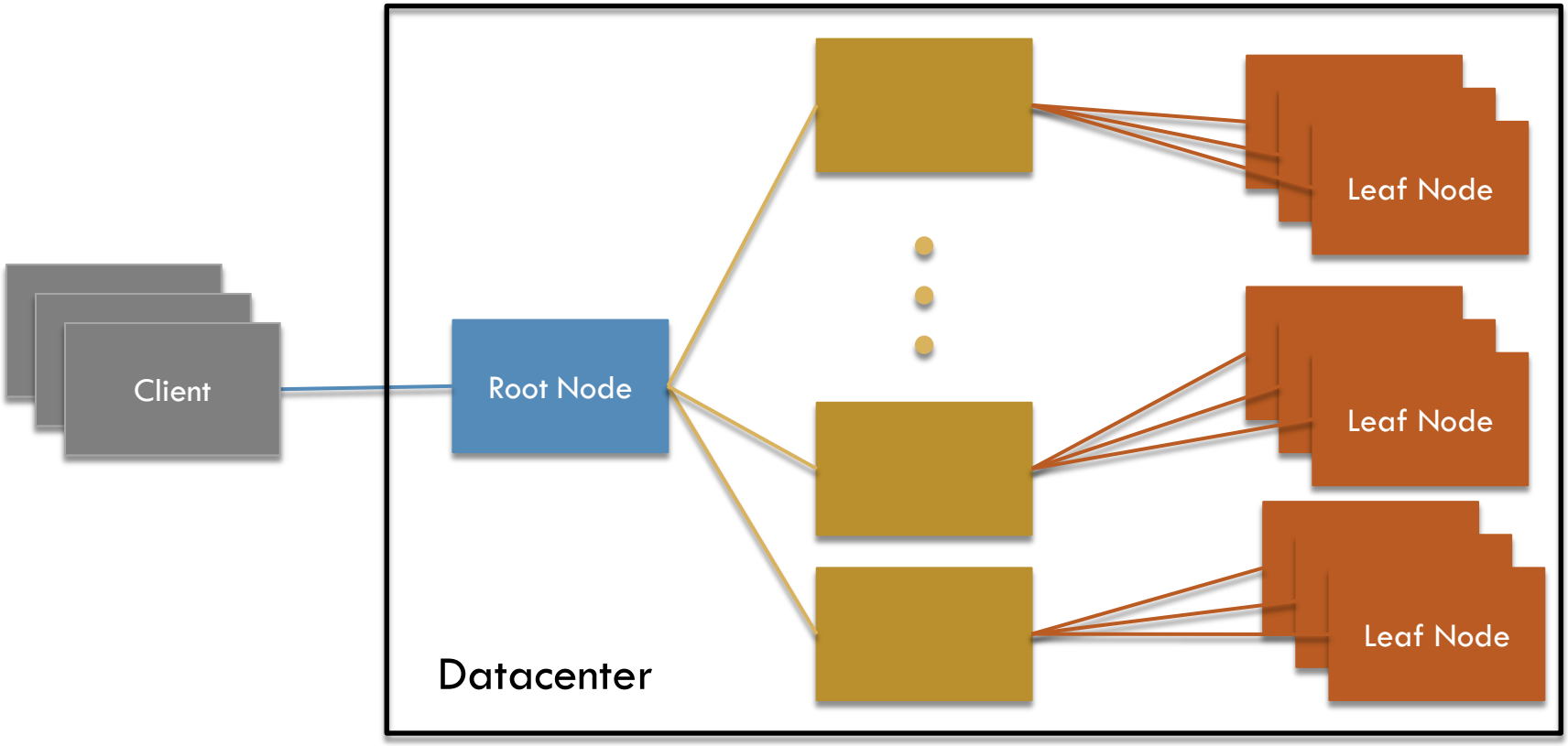
- Latency-critical applications have stringent performance requirements → low datacenter utilization
  - Wastes billions of dollars in energy and equipment annually
- Research in this area hampered by the lack of a comprehensive benchmark suite
  - Few latency-critical applications → limited coverage
  - Complicated setup and configuration
  - Methodological issues

  } Inaccurate latency measurements
- TailBench makes latency-critical applications easy to analyze
  - Varied application domains and latency characteristics
  - Standardized, statistically sound methodology
  - Supports simplified load-testing configurations

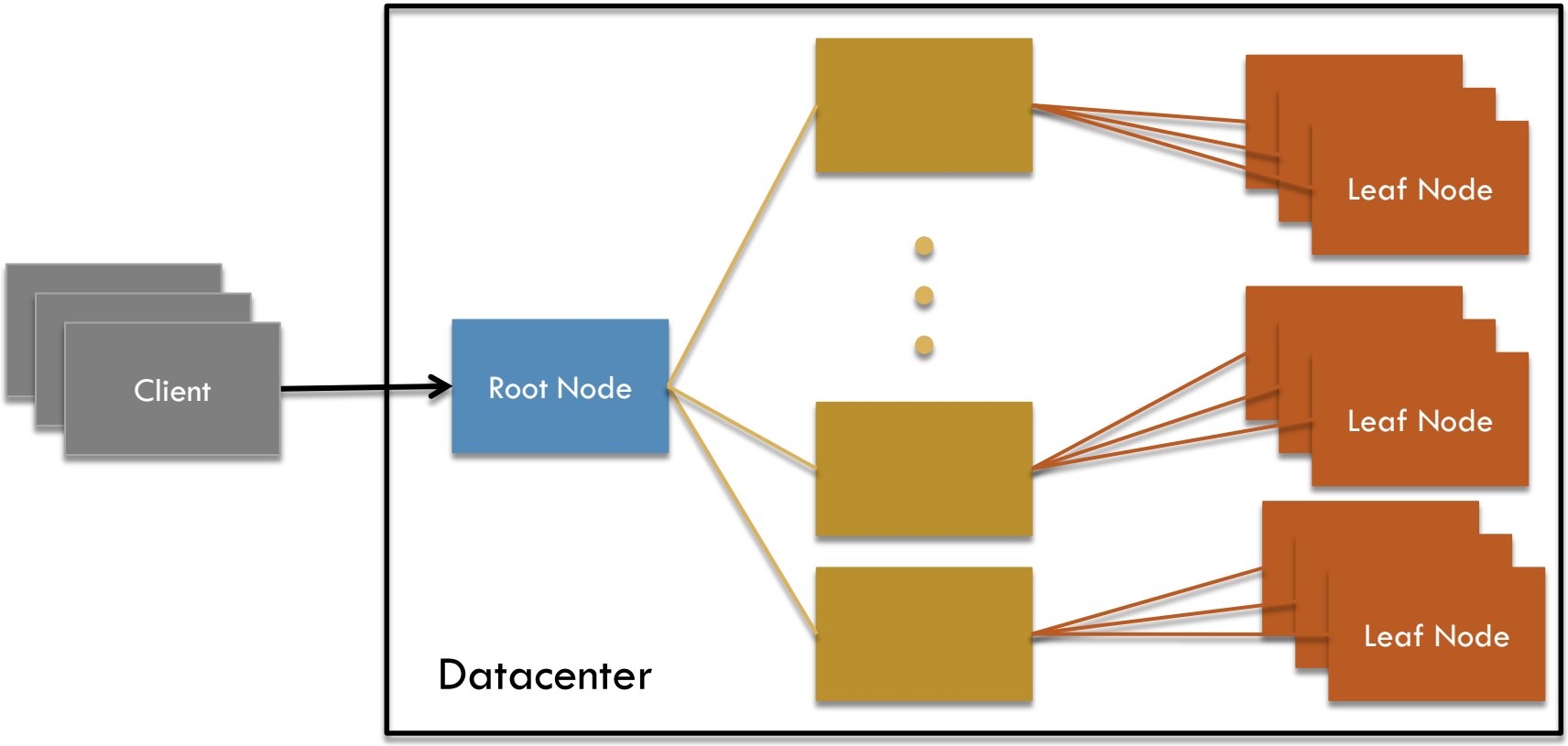
Outline

- Background and Motivation
- TailBench Applications
- TailBench Harness
- Simplified Configurations

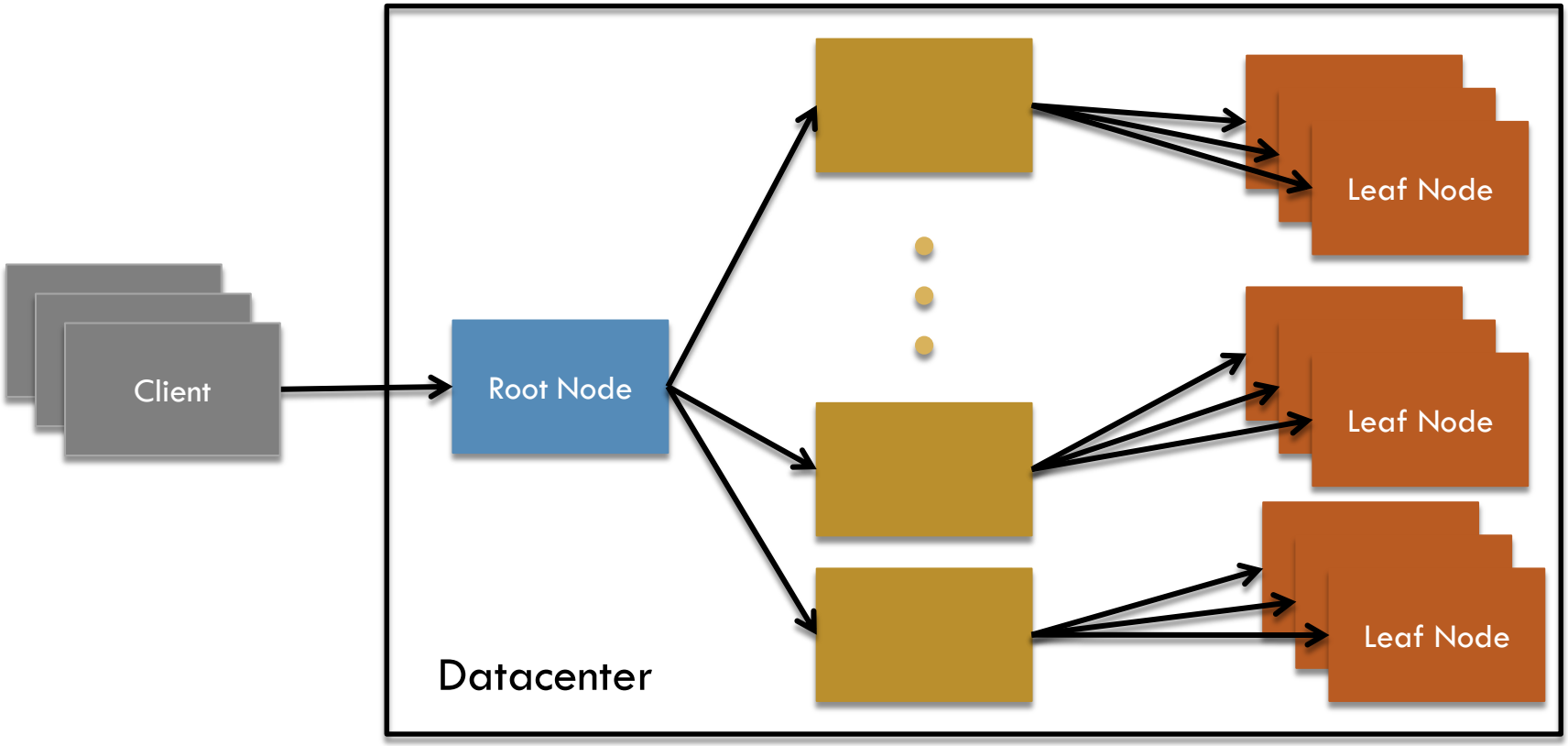
Understanding Latency-Critical Applications



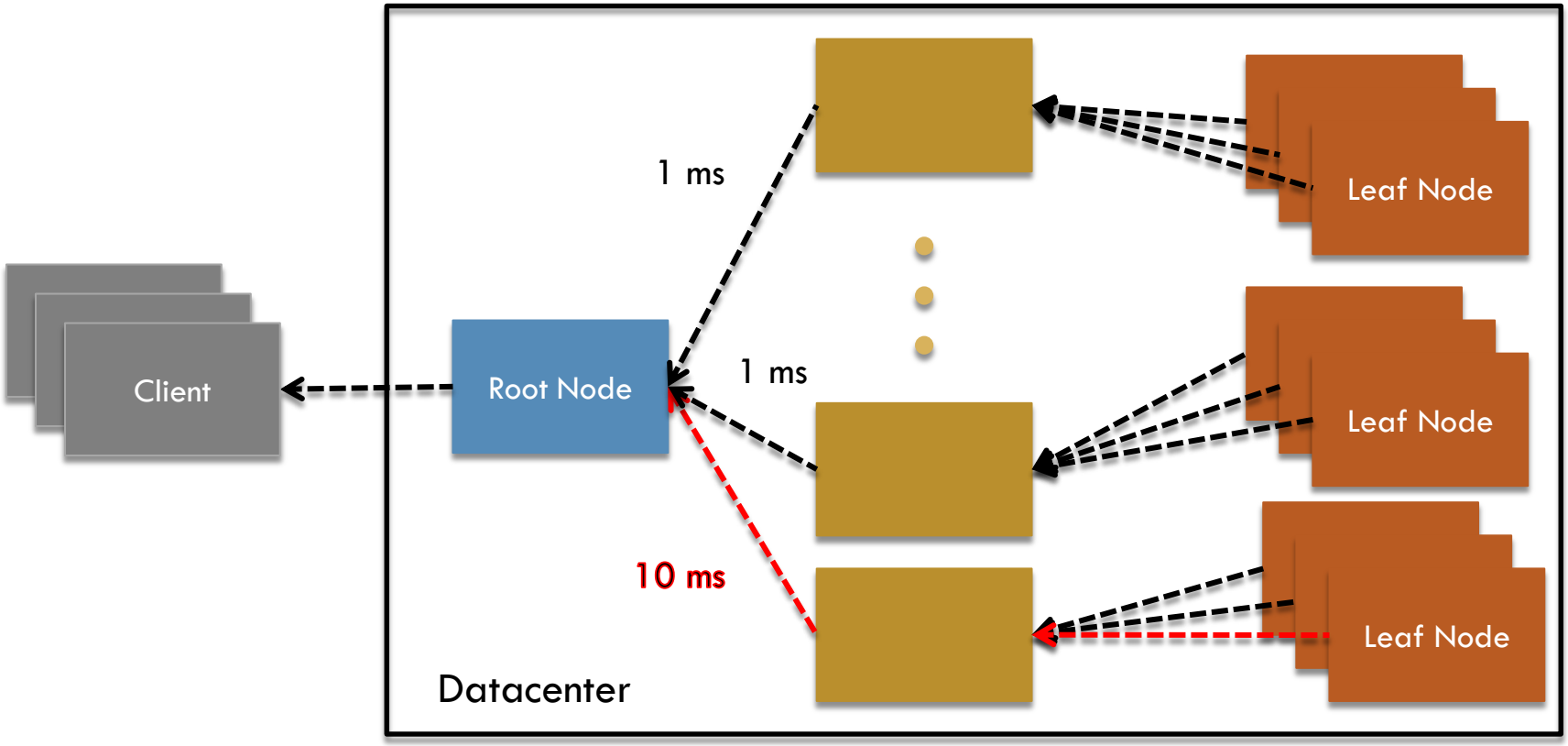
Understanding Latency-Critical Applications



Understanding Latency-Critical Applications

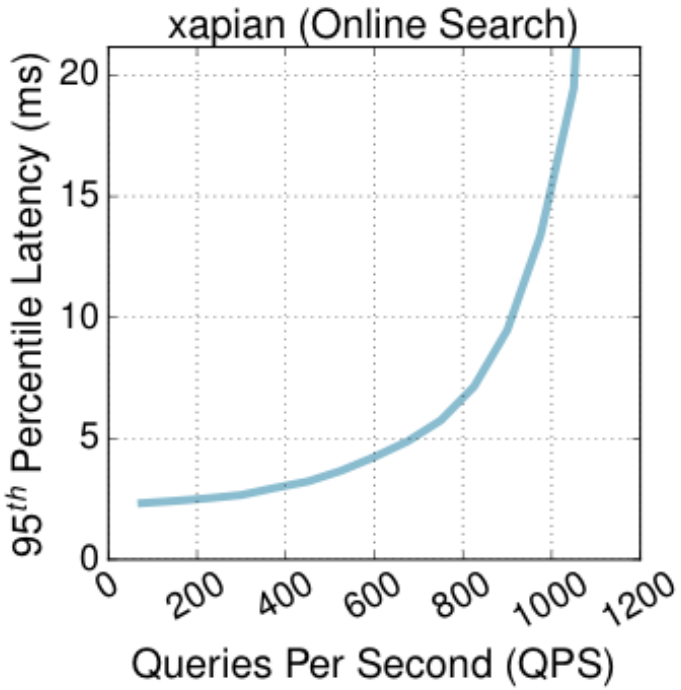


Understanding Latency-Critical Applications



- The few slowest responses determine user-perceived latency
  - Tail latency (e.g., 95th / 99th percentile), not mean latency, determines performance

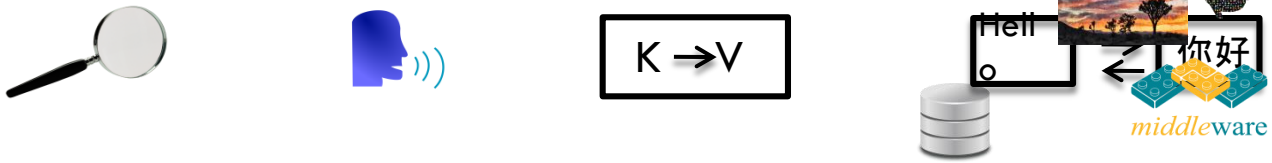
Latency Requirements Cause Low Utilization



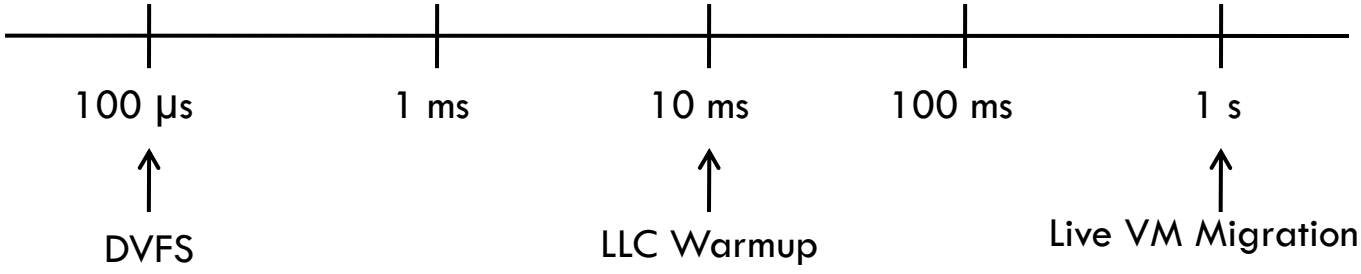
- End-to-end latency increases rapidly with load
  - Must keep utilization low to keep latency within reasonable bounds
- Traditional resource management techniques (e.g., colocation) often cannot be used since they degrade latency
- Low resource utilization wastes billions of dollars in energy and equipment
  - Sparked research in latency-critical systems

Benchmark Suite Design Goals

- Applications from a diverse set of domains



- Applications with diverse tail latency characteristics



- Easy to set up and run
  - Support different measurement scenarios
- Robust latency measurement methodology

Outline

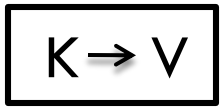
- Background and Motivation
- TailBench Applications
- TailBench Harness
- Simplified Configurations

TailBench Applications

xapian

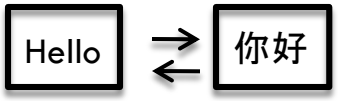
Online Search

masstree



Key-Value Store

moses



Statistical Machine Translation

sphinx

Speech Recognition

img-dnn

Image Recognition

specjbb

Java Middleware

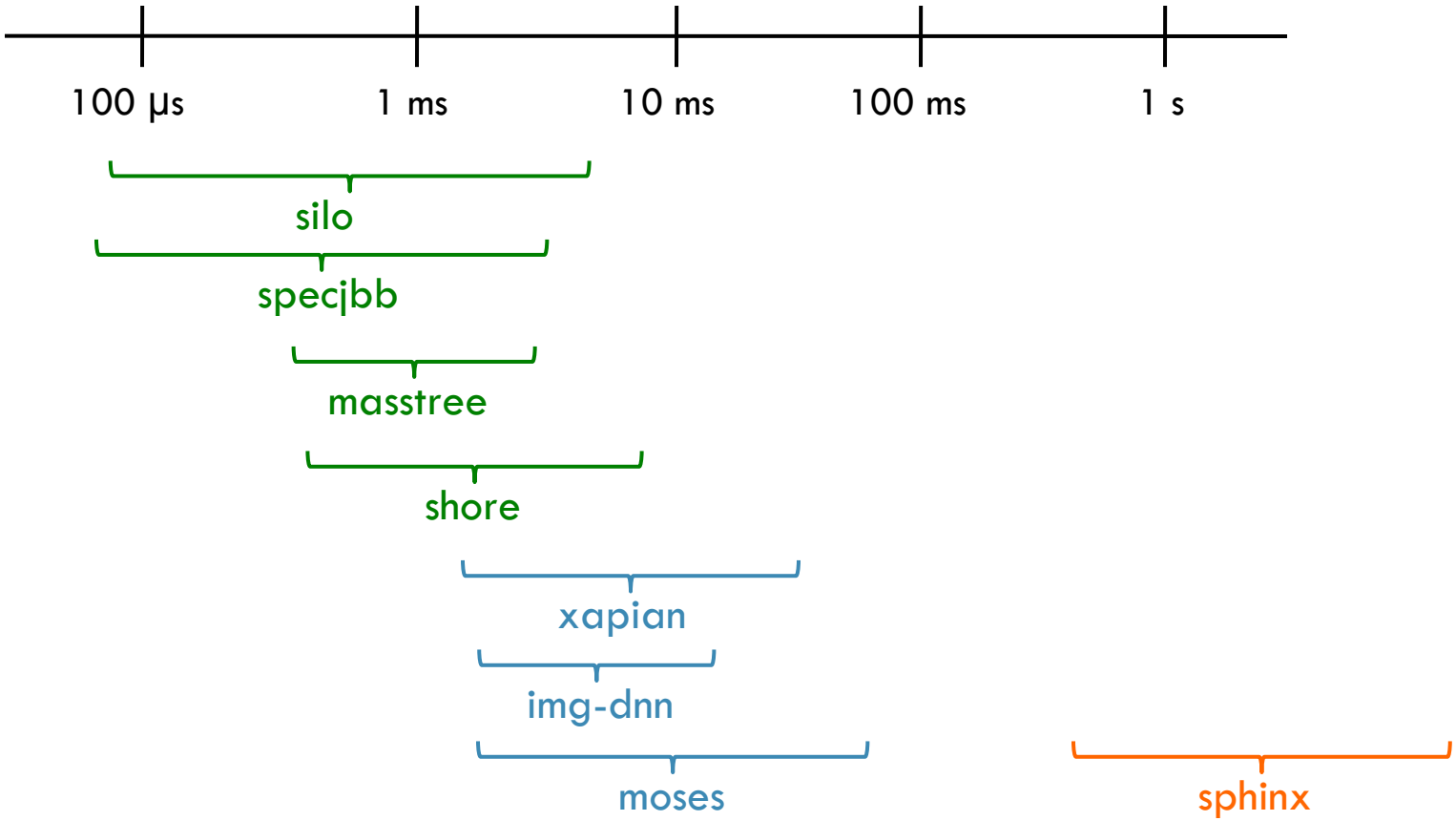
silo

In-memory Database

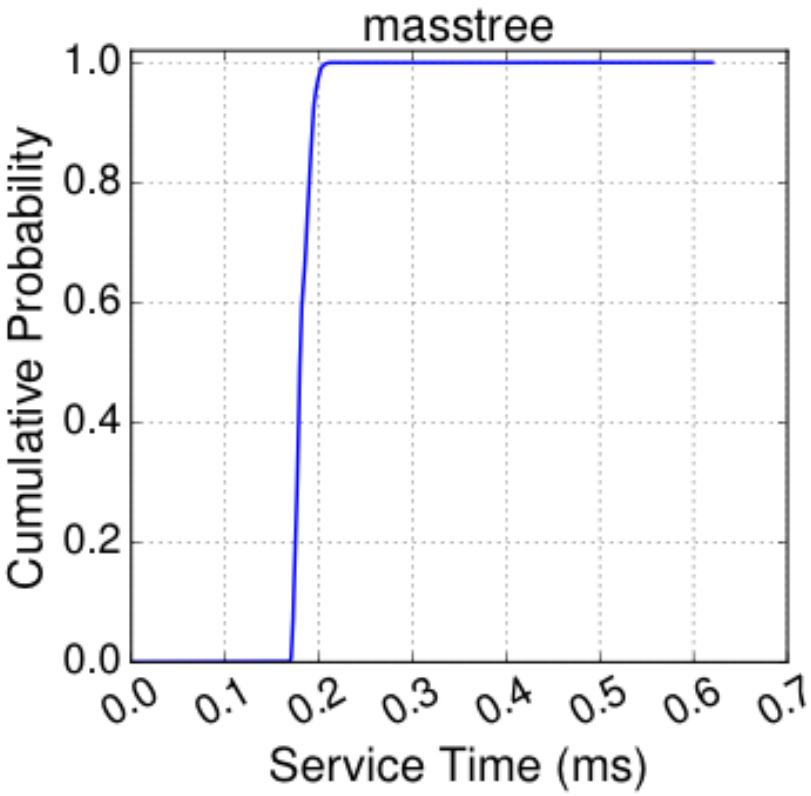
shore

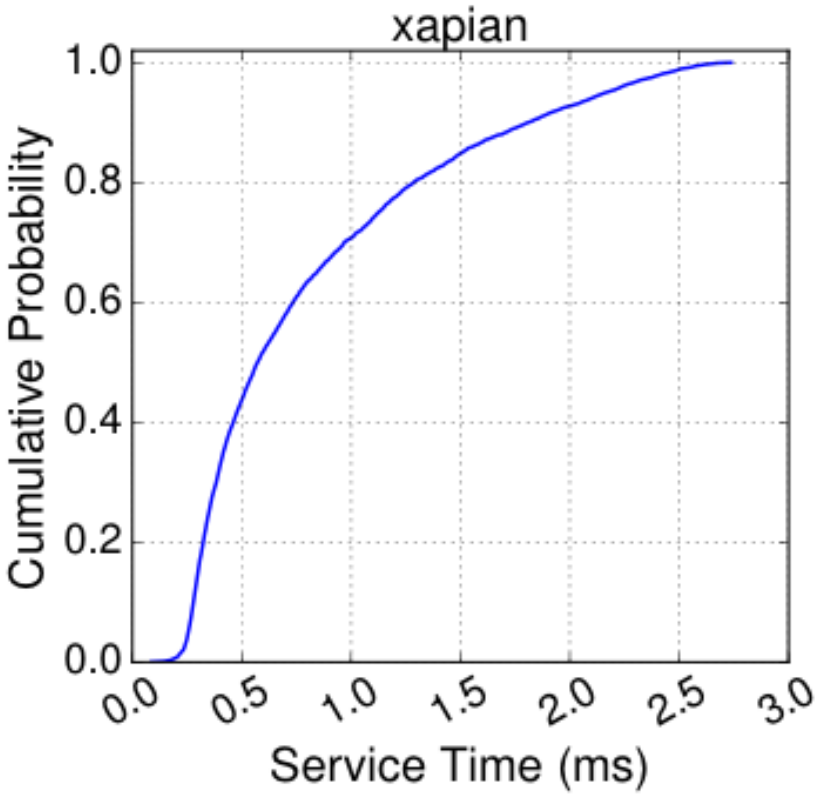
On-disk Database

Wide Range of End-to-End Latencies



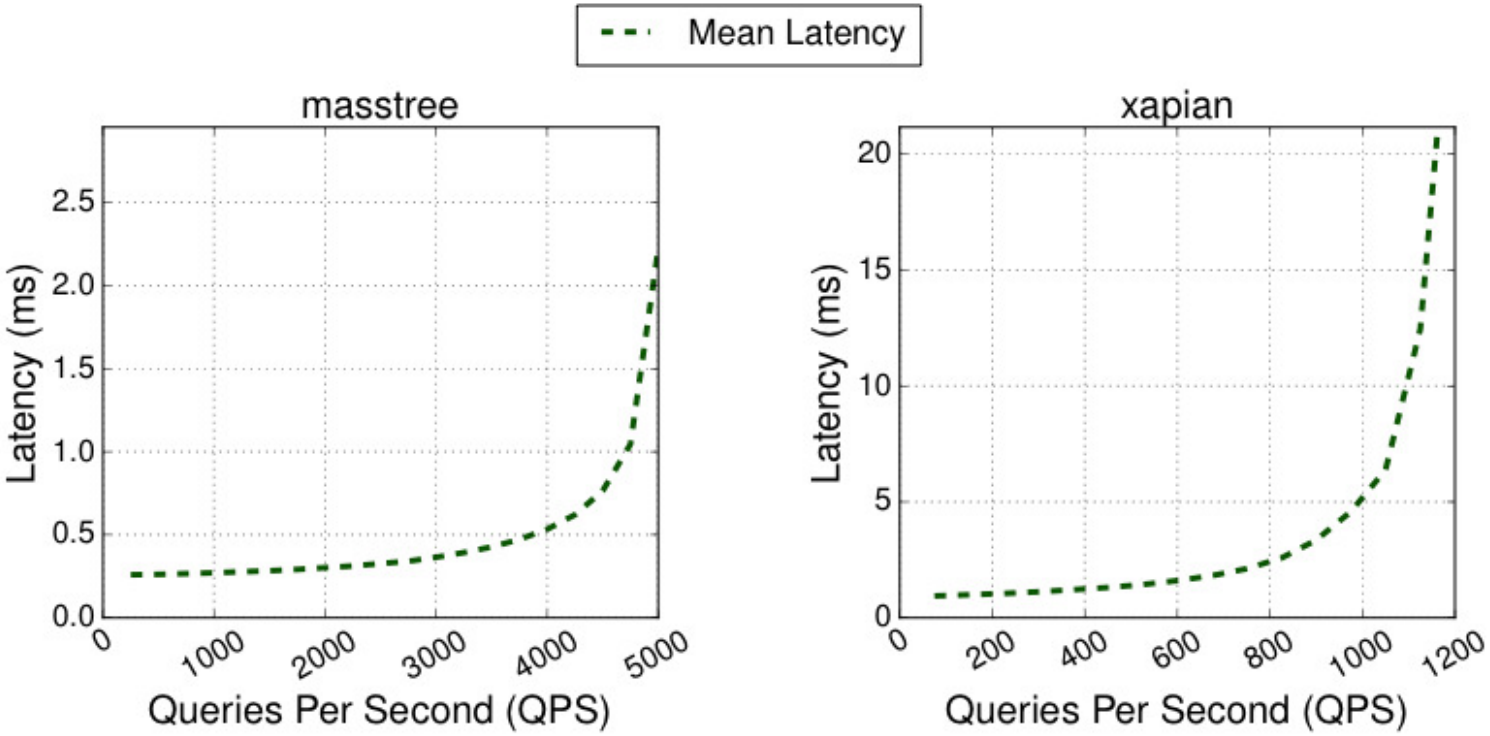
Varied Service Time Characteristics



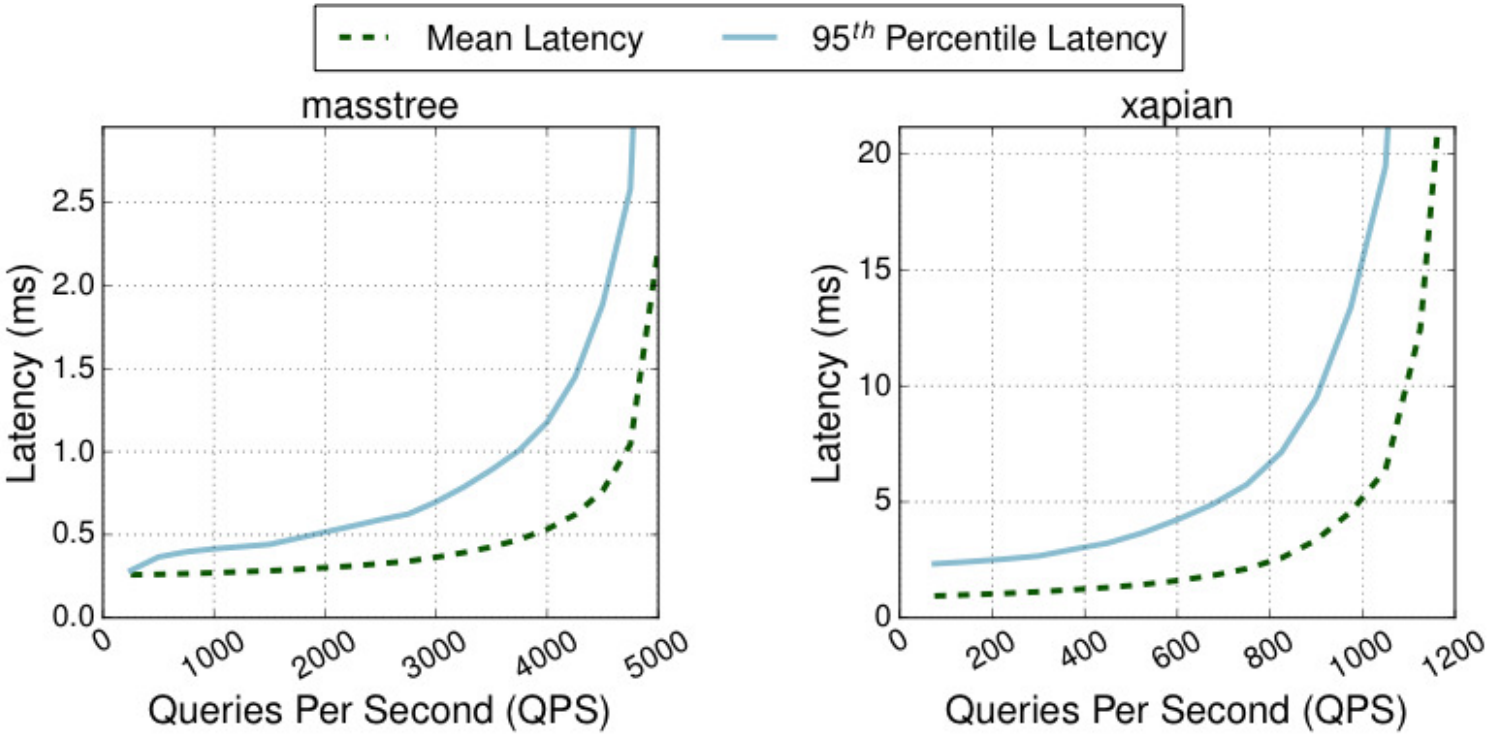


- masstree service times are more tightly distributed
- xapian service times are more loosely distributed

End-to-End Latency vs. Load

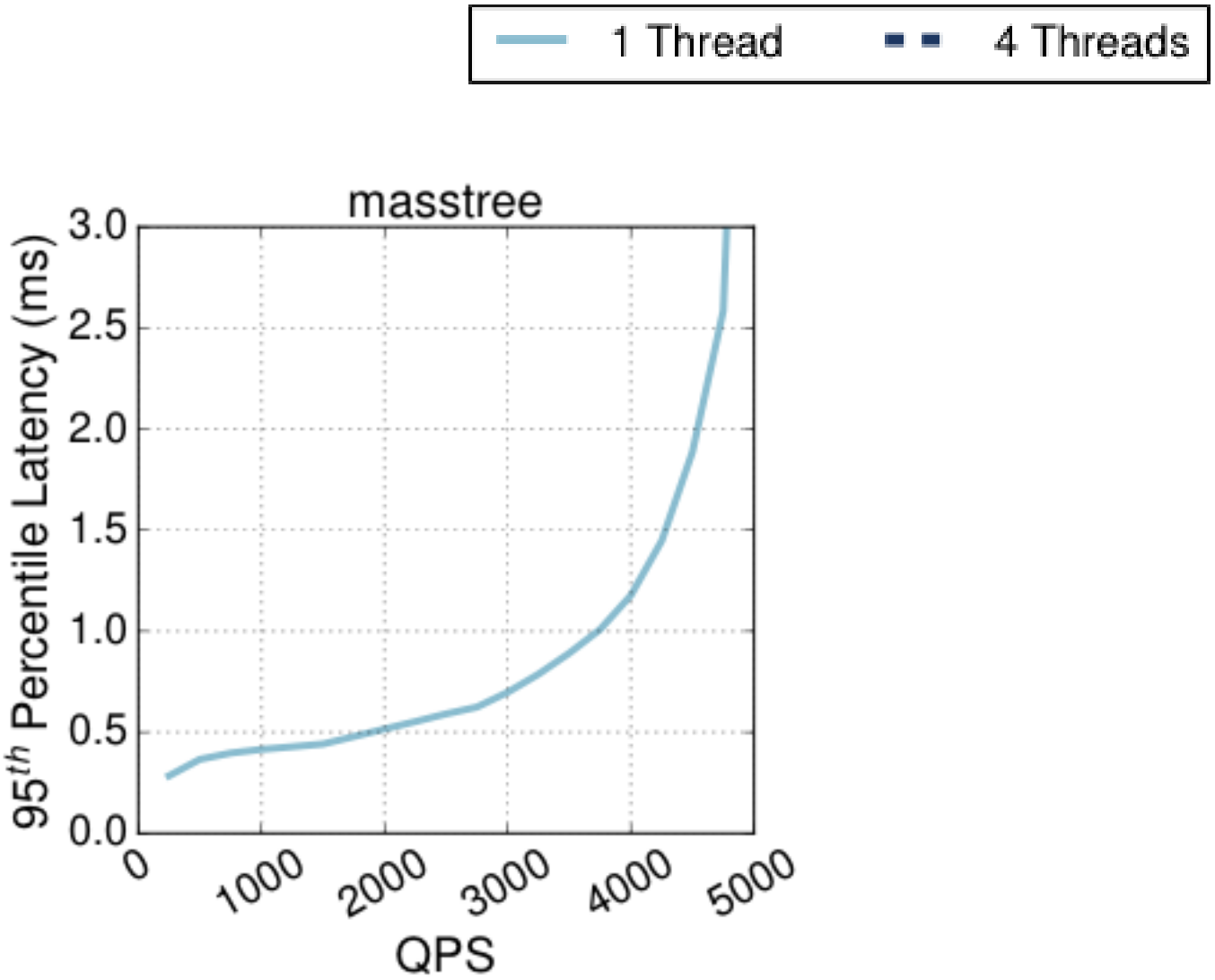


Tail ≠ Mean

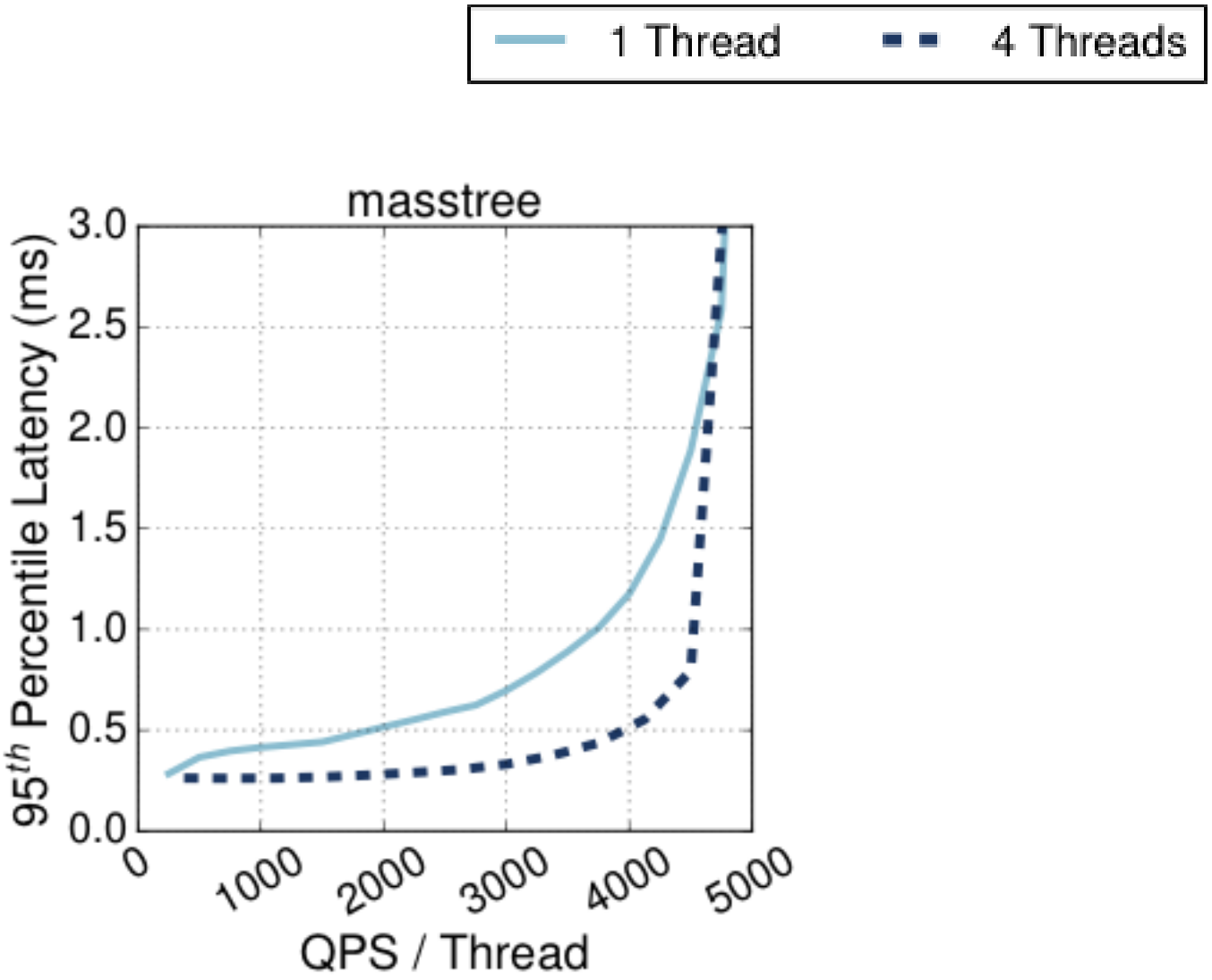


- Tail latency increases more rapidly with load than mean latency
- Relationship between mean and tail latencies is hard to predict

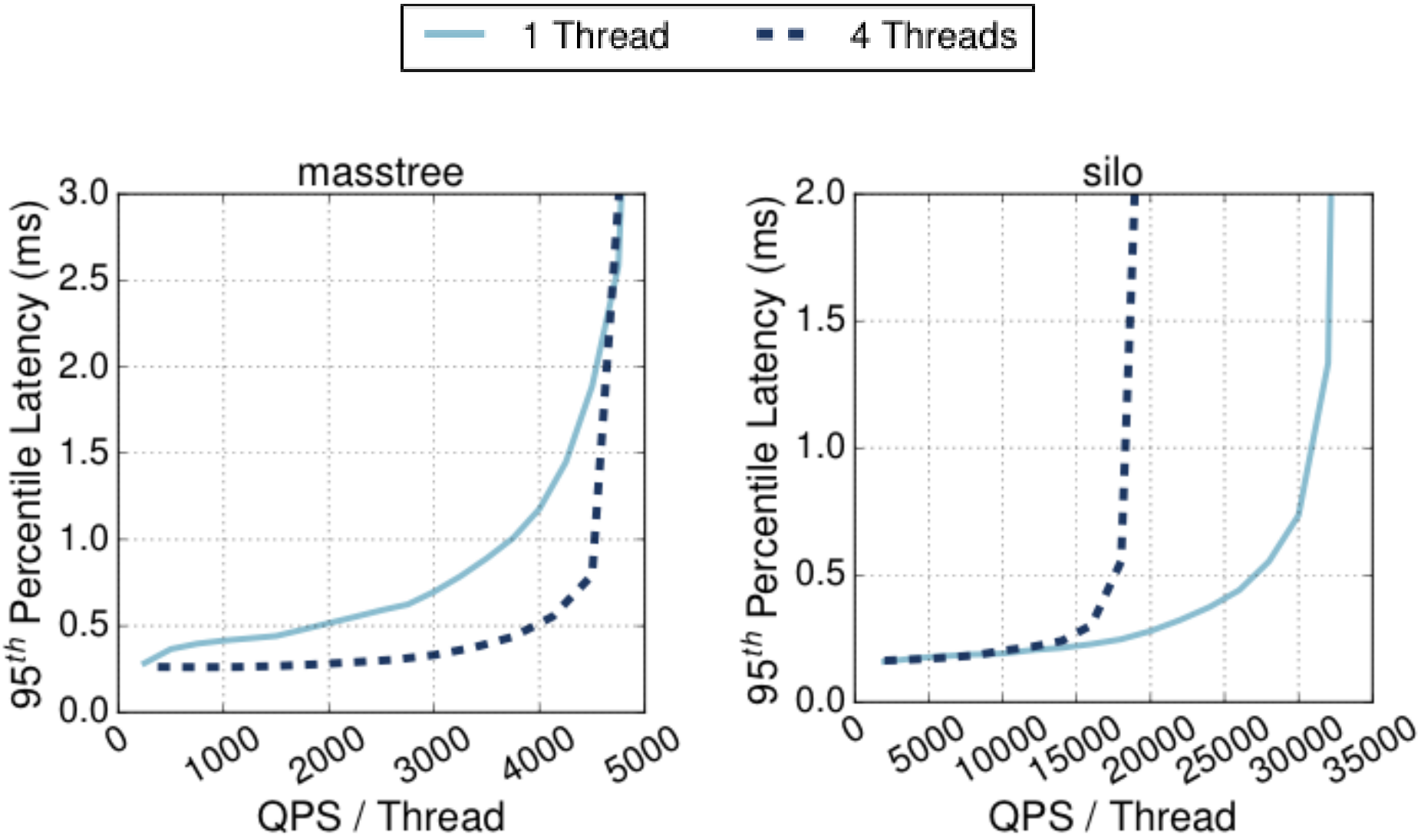
Impact of Parallelism



Parallelism Helps Some Applications



…But Hurts Others



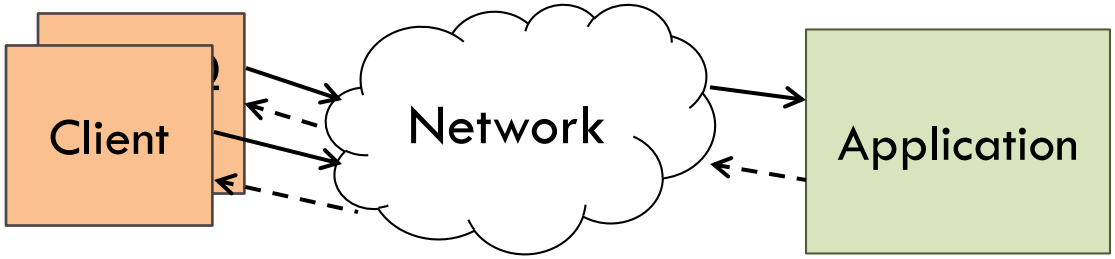
Outline

- Background and Motivation
- TailBench Applications
- TailBench Harness
- Simplified Configurations

TailBench Harness

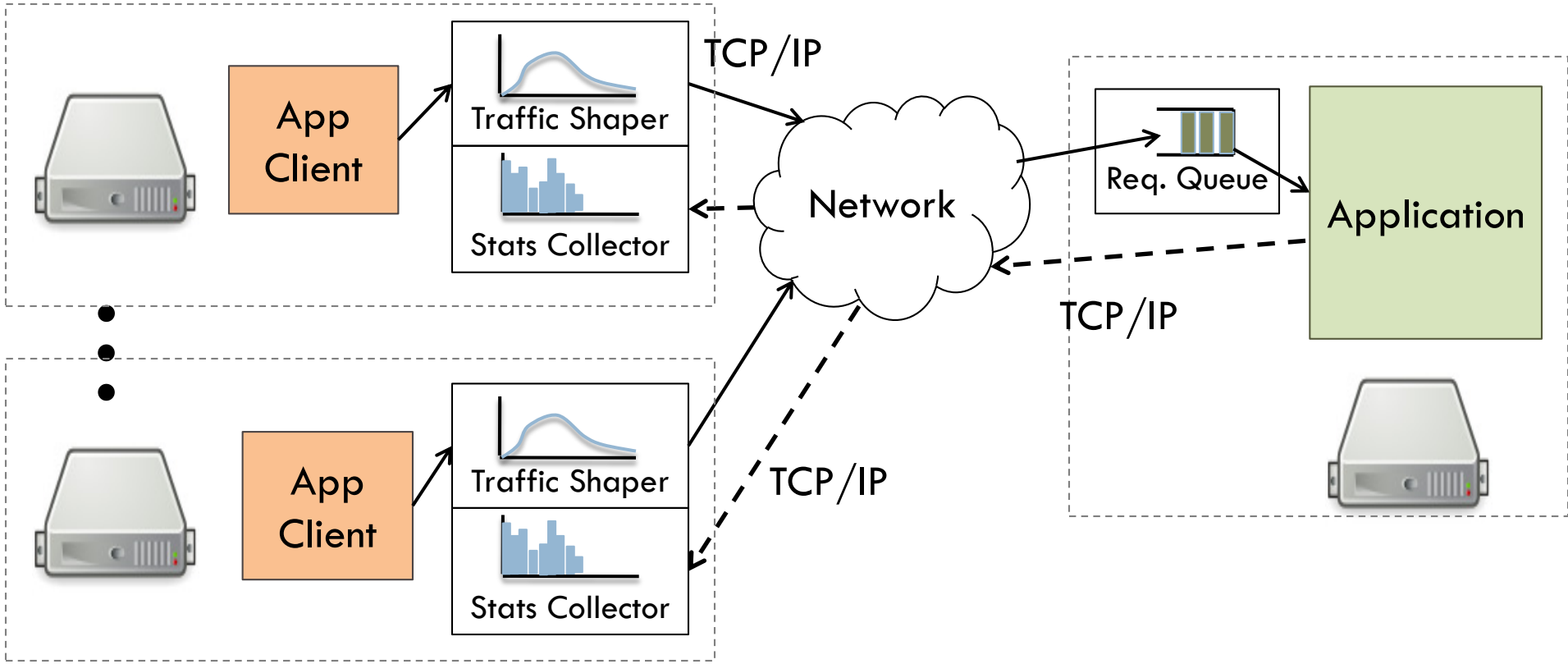
- Measuring tail latency accurately is complicated
  - Load generation, statistics aggregation, warmup periods…
- Harness encapsulates most of the complexity
- Harness makes TailBench easily extensible
  - New benchmarks reuse existing harness functionality
- Simplified harness configurations enable different measurement scenarios
  - Trade off some accuracy for reduced setup complexity

Example: Open- vs. Closed-Loop Clients

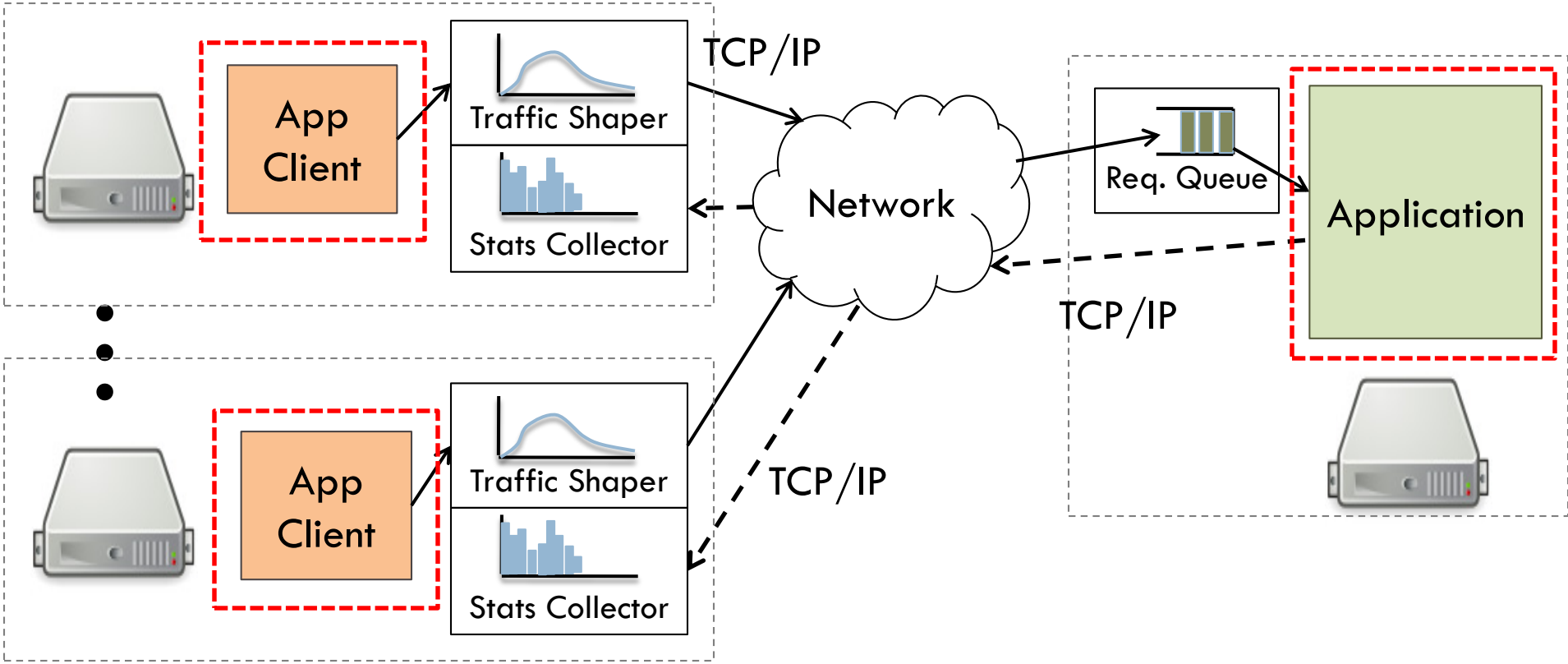


- Many popular load testers use closed-loop clients
  - Clients wait for response before submitting next request
  - Increase in application load throttles client request rate
- Latency-critical applications typically service a large number of independent clients
  - Request rate independent of application load
  - Better modeled by open-loop clients
- Closed-loop clients can underestimate latency by orders of magnitude [Tene LLS 2013, Zhang ISCA 2016]

Networked Harness Configuration

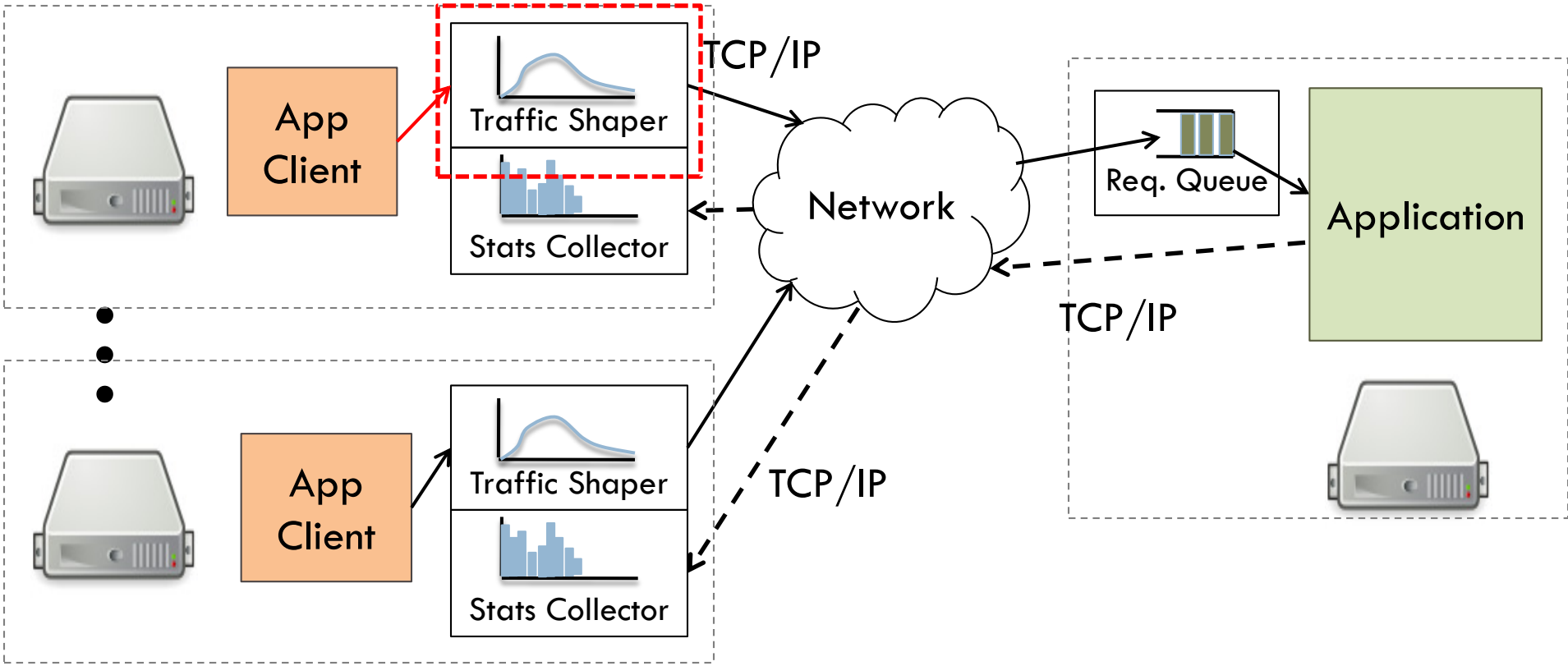


Networked Harness Configuration



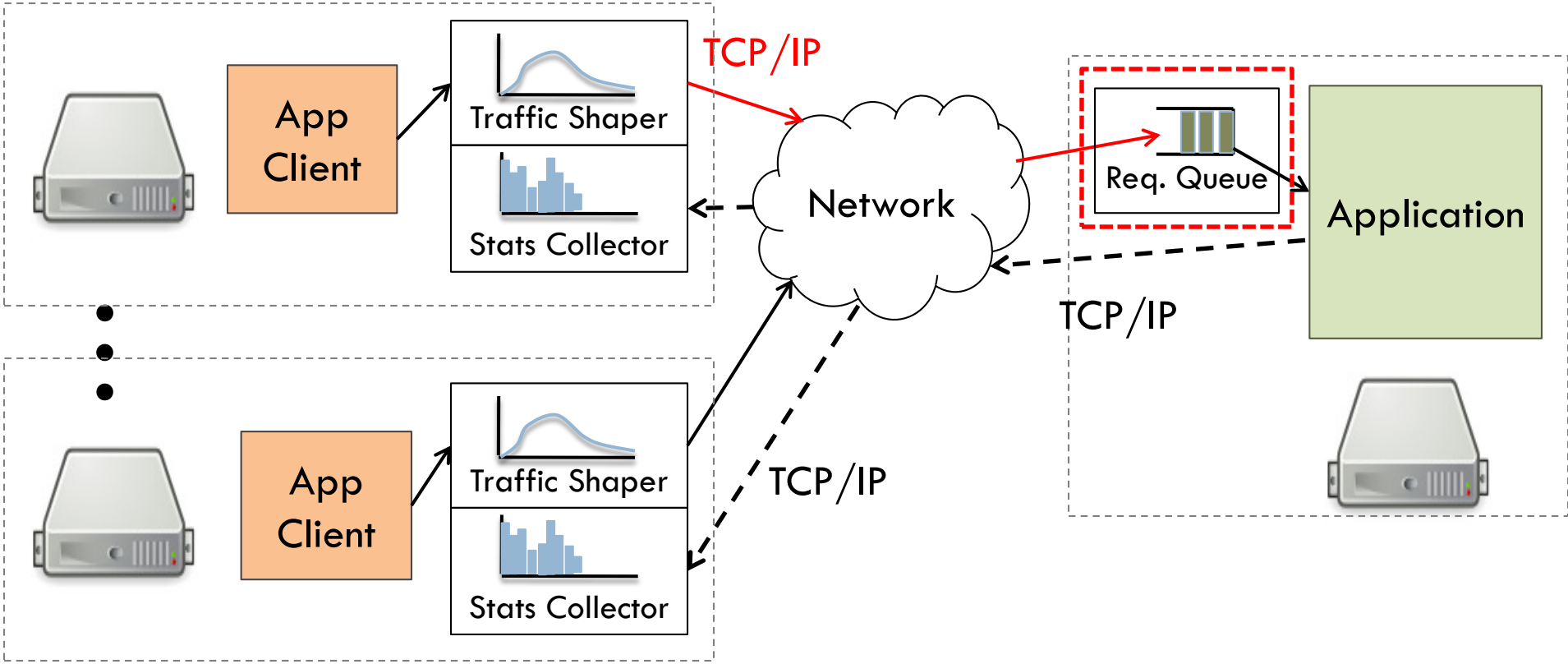
- Application and the clients run on separate machines

Networked Harness Configuration



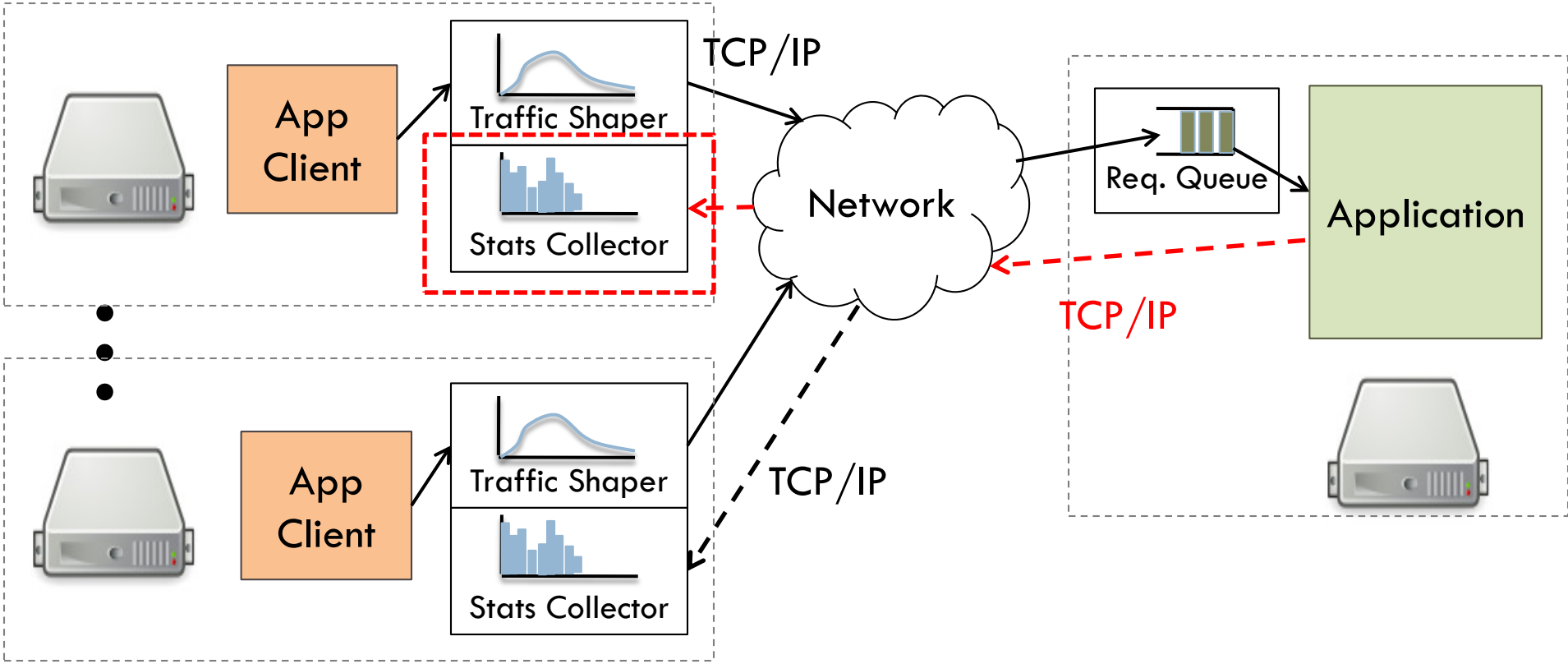
- Application and the clients run on separate machines
- Traffic Shaper inserts inter-request delays to model load

Networked Harness Configuration



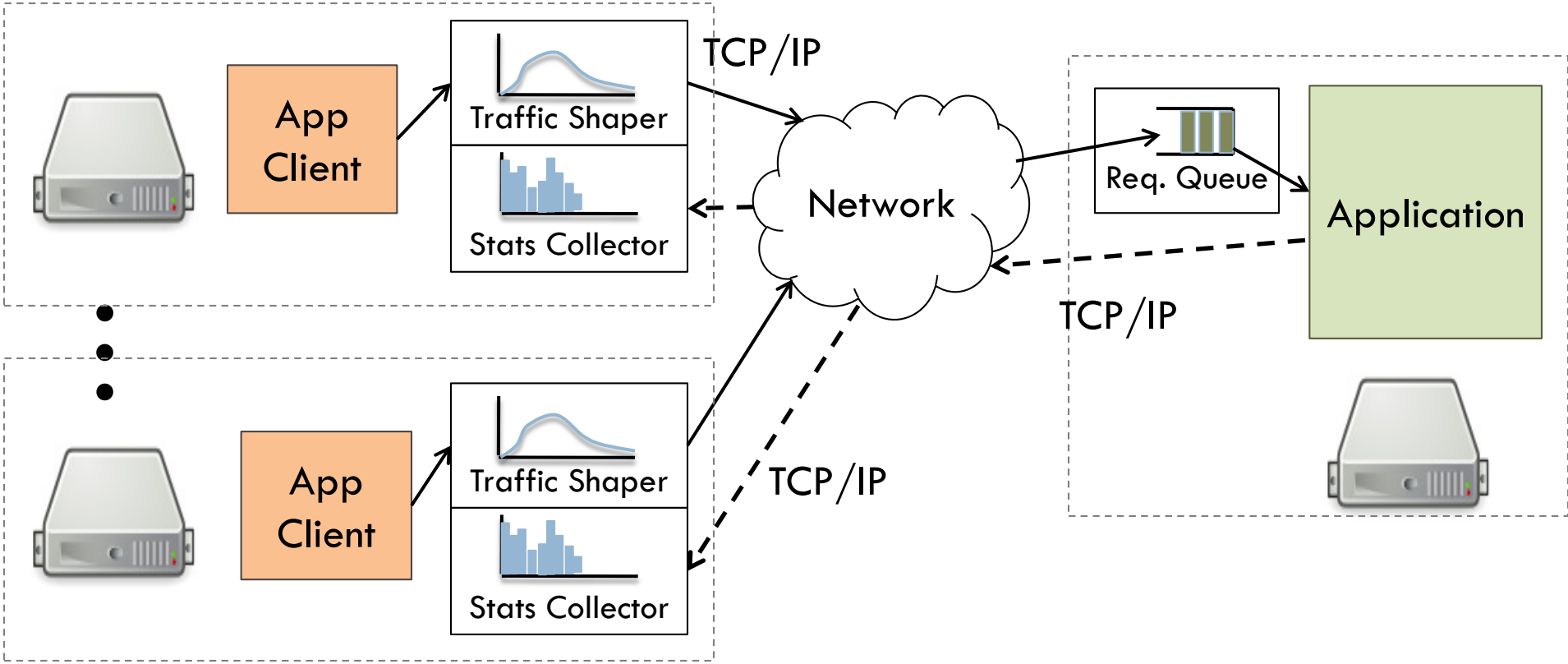
- Application and the clients run on separate machines
- Traffic Shaper inserts inter-request delays to model load
- Request Queue enqueues incoming requests and measures service times and queuing delays

Networked Harness Configuration



- Application and the clients run on separate machines
- Traffic Shaper inserts inter-request delays to model load
- Request Queue enqueues incoming requests and measures service times and queuing delays
- Statistics Collector aggregates latency data

Networked Harness Configuration



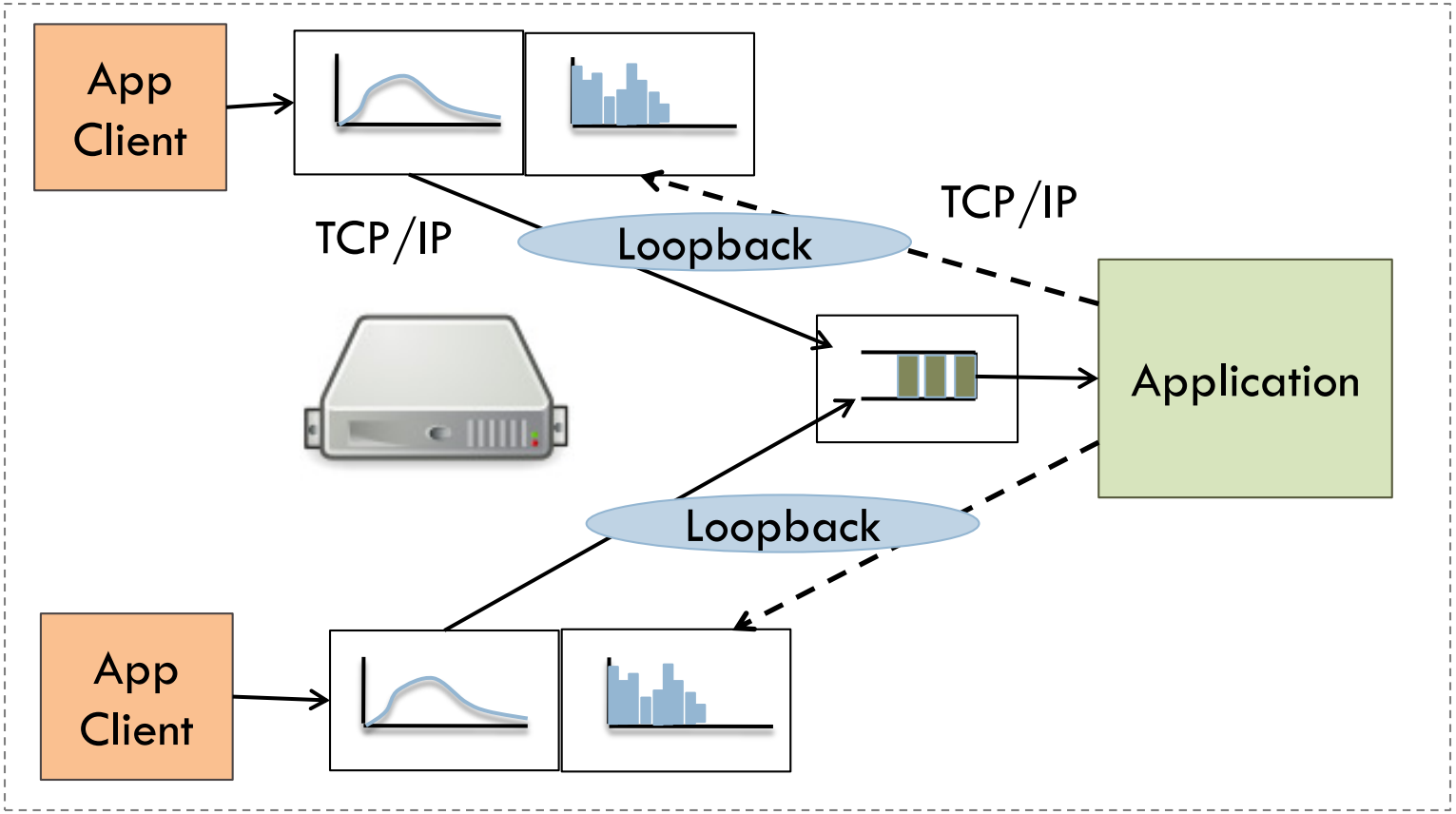
✓ Faithfully captures all sources of overhead

X Difficult to configure and deploy

Outline

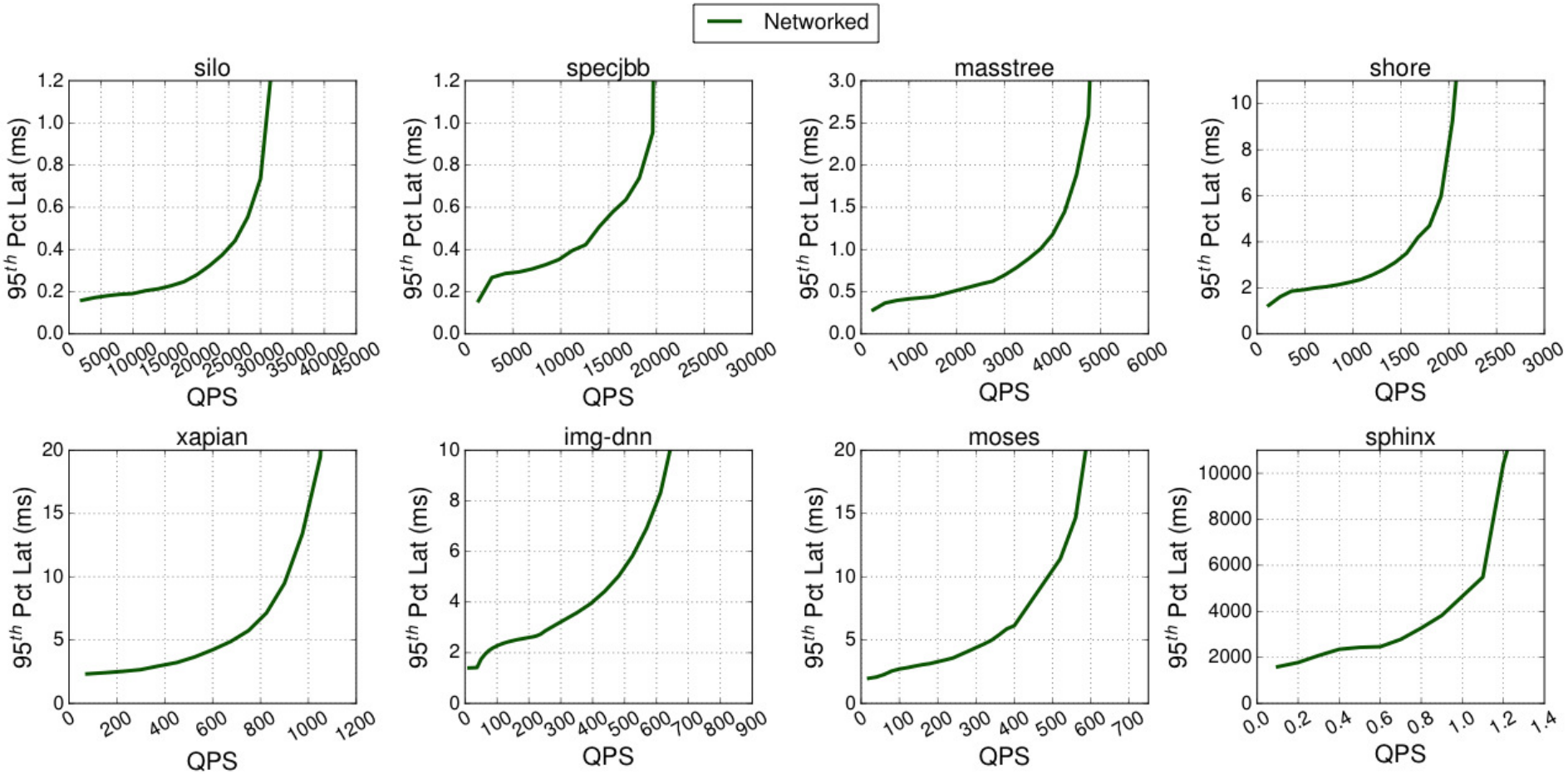
- Background and Motivation
- TailBench Applications
- TailBench Harness
- Simplified Configurations

Loopback Harness Configuration

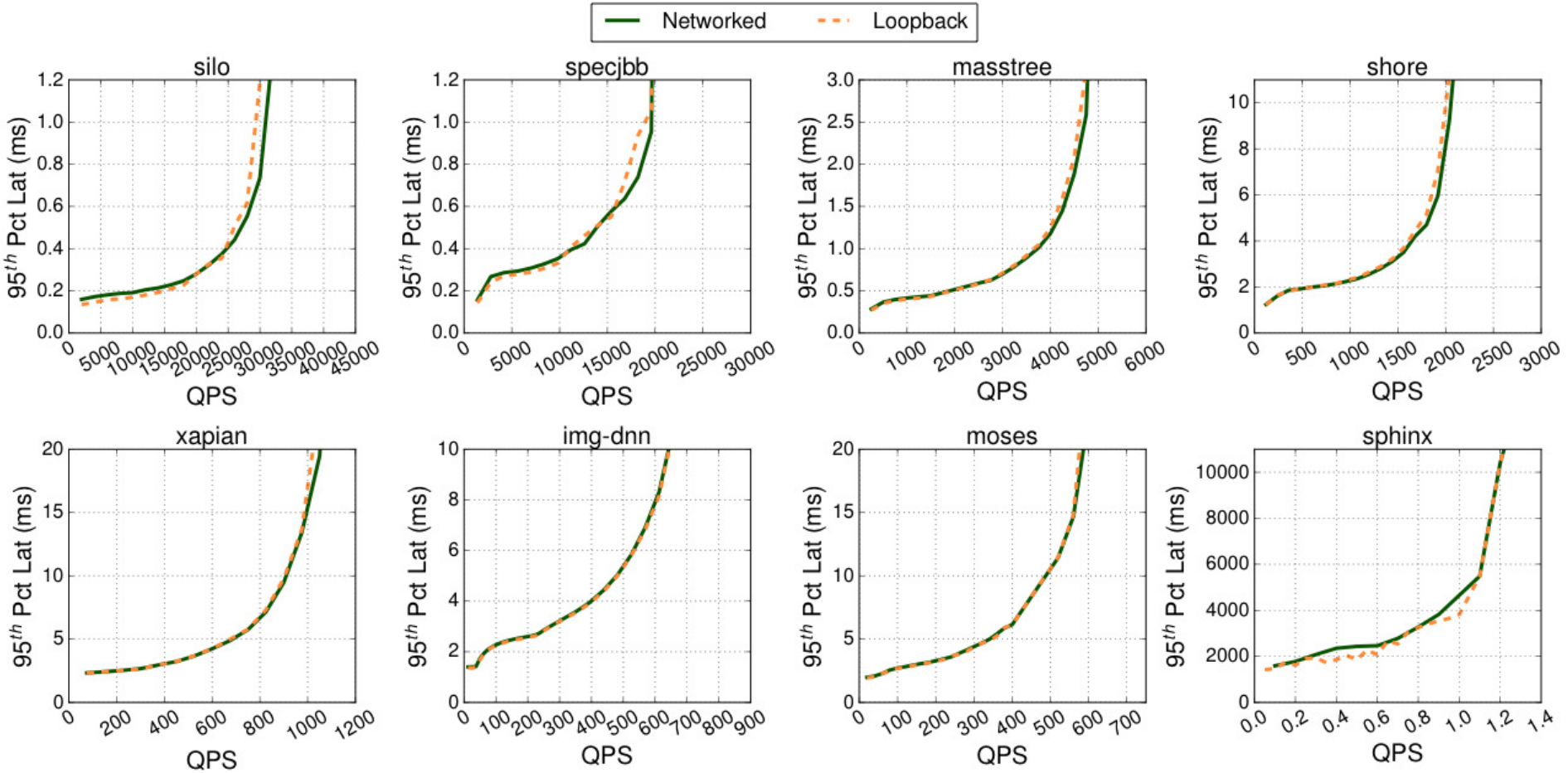


- Application and clients reside on the same machine
- ✓ Reduced setup complexity
- ✓ Highly accurate in many cases

Load-Latency for Networked Configuration

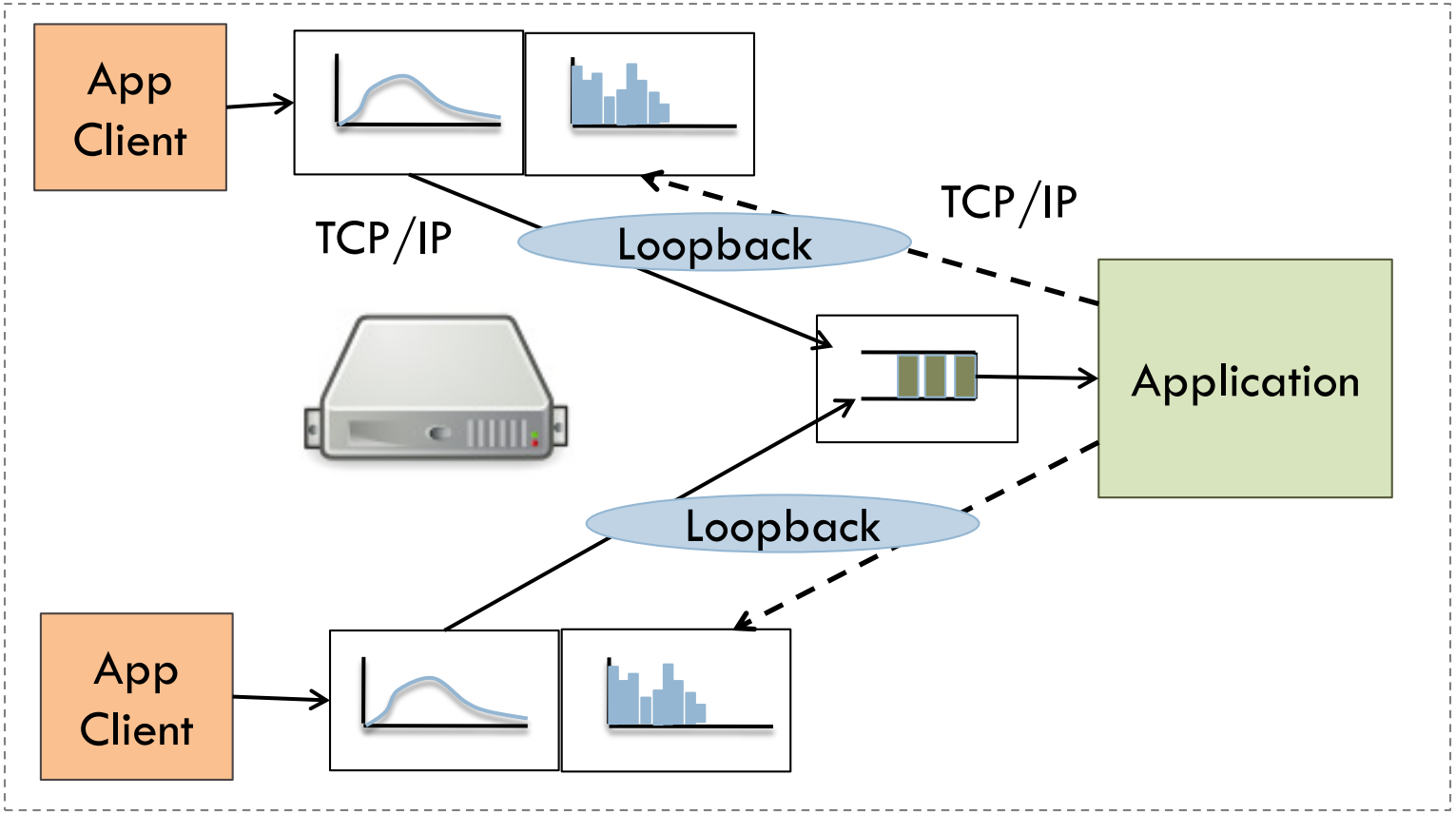


Loopback Configuration Highly Accurate



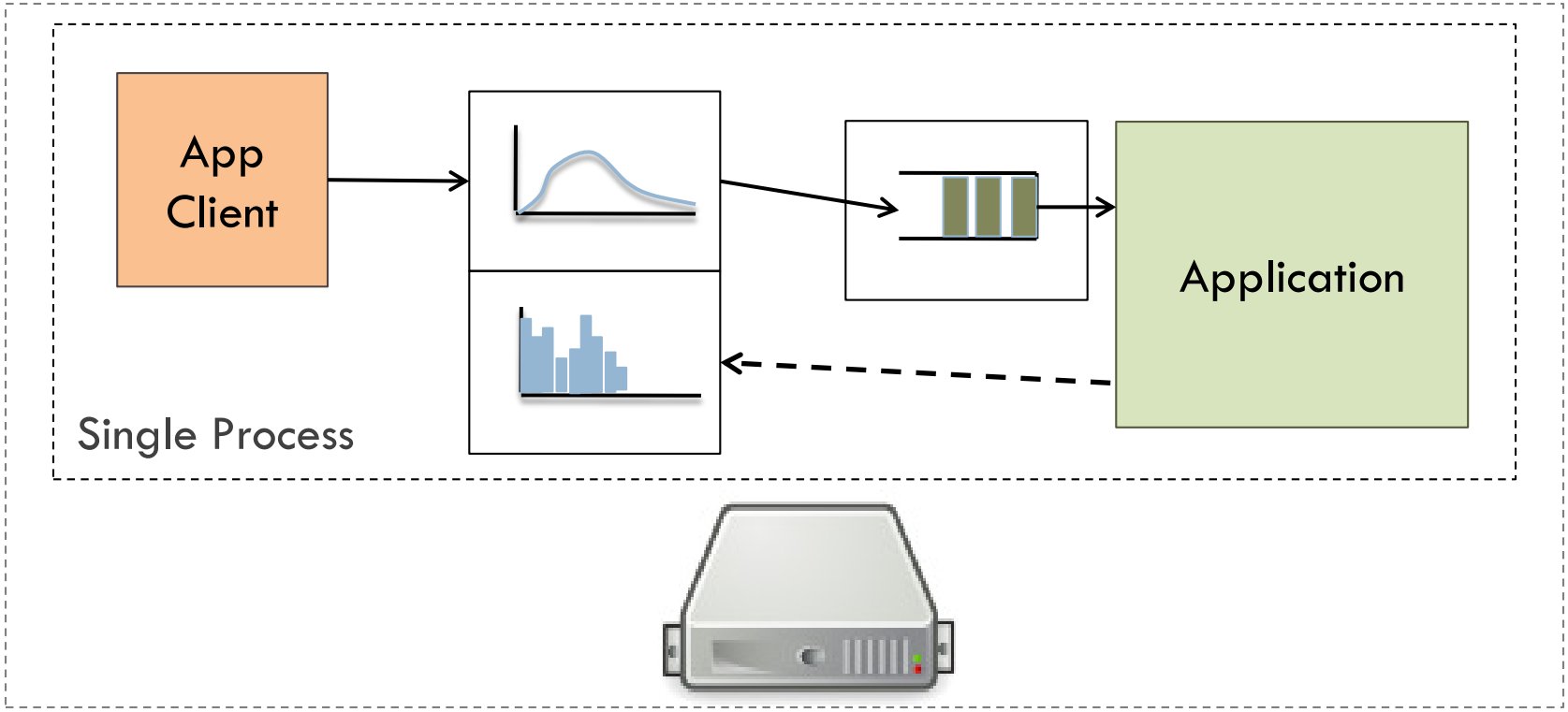
- Loopback and Networked configurations have near-identical performance
  - Networking delays minimal in our setup

Loopback Harness Configuration



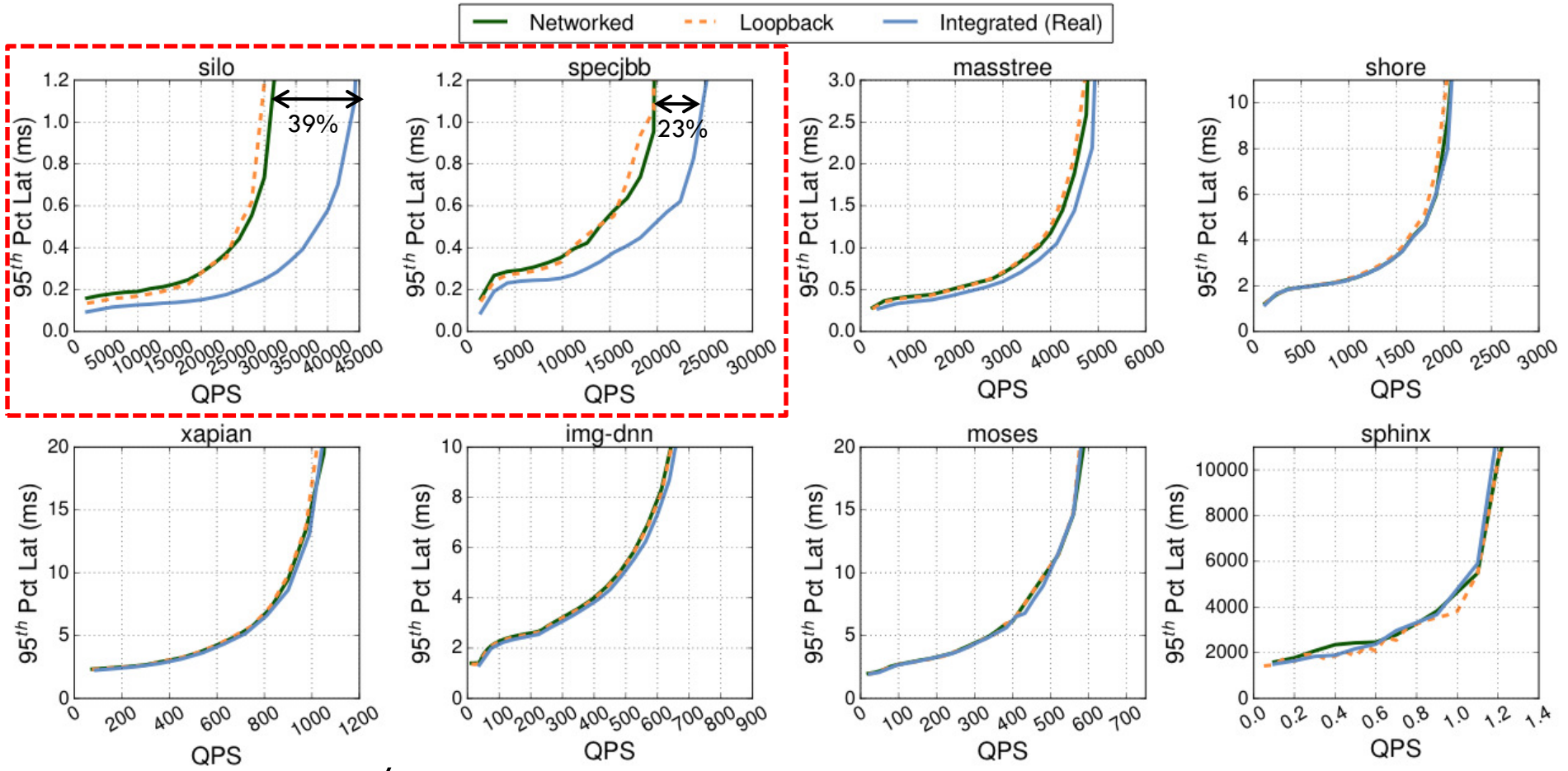
- Application and clients reside on the same machine
- ✓ Reduced setup complexity
- ✓ Highly accurate in many cases
- x Still difficult to simulate

Integrated Harness Configuration



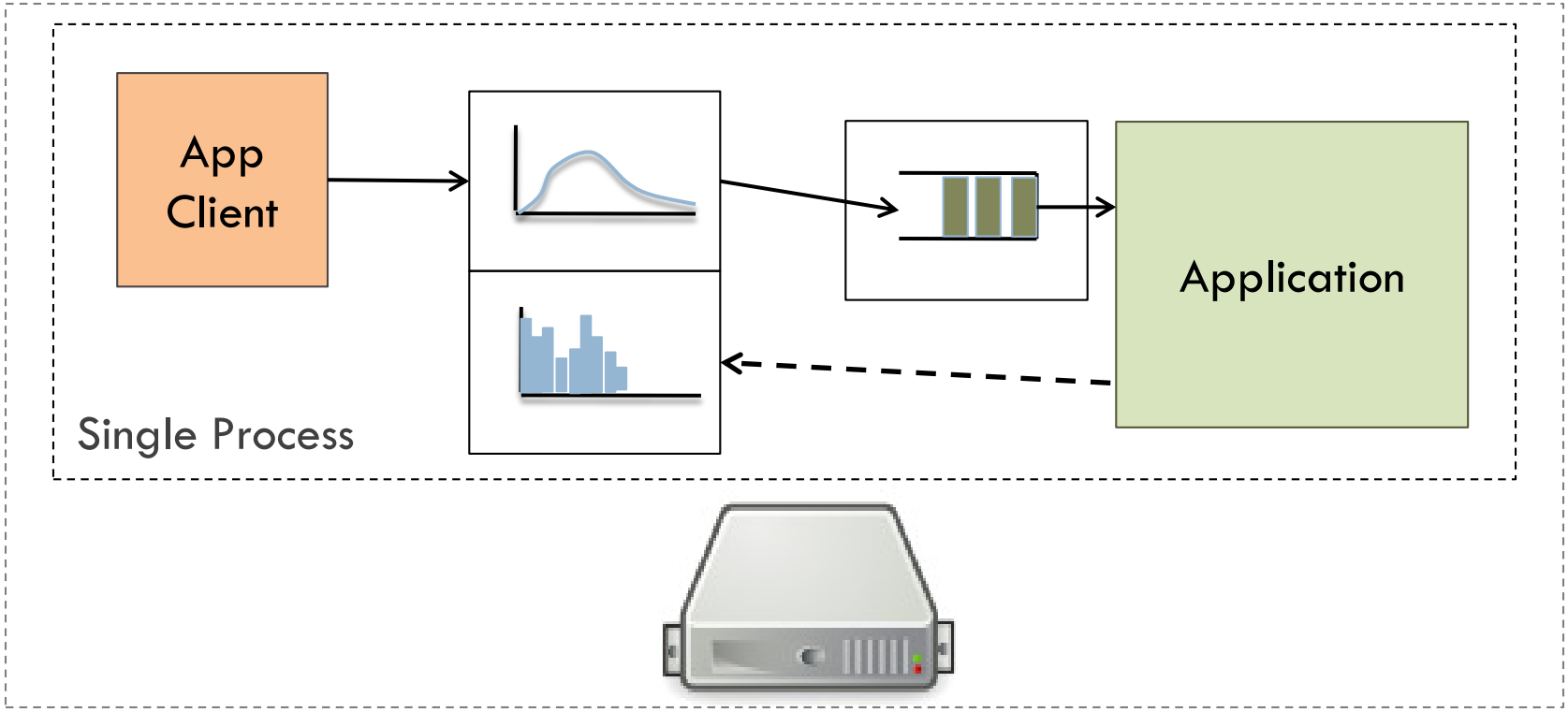
- Application and client integrated into a single process
- ✓ Easy to setup
- X Some loss of accuracy

Integrated Configuration Validation



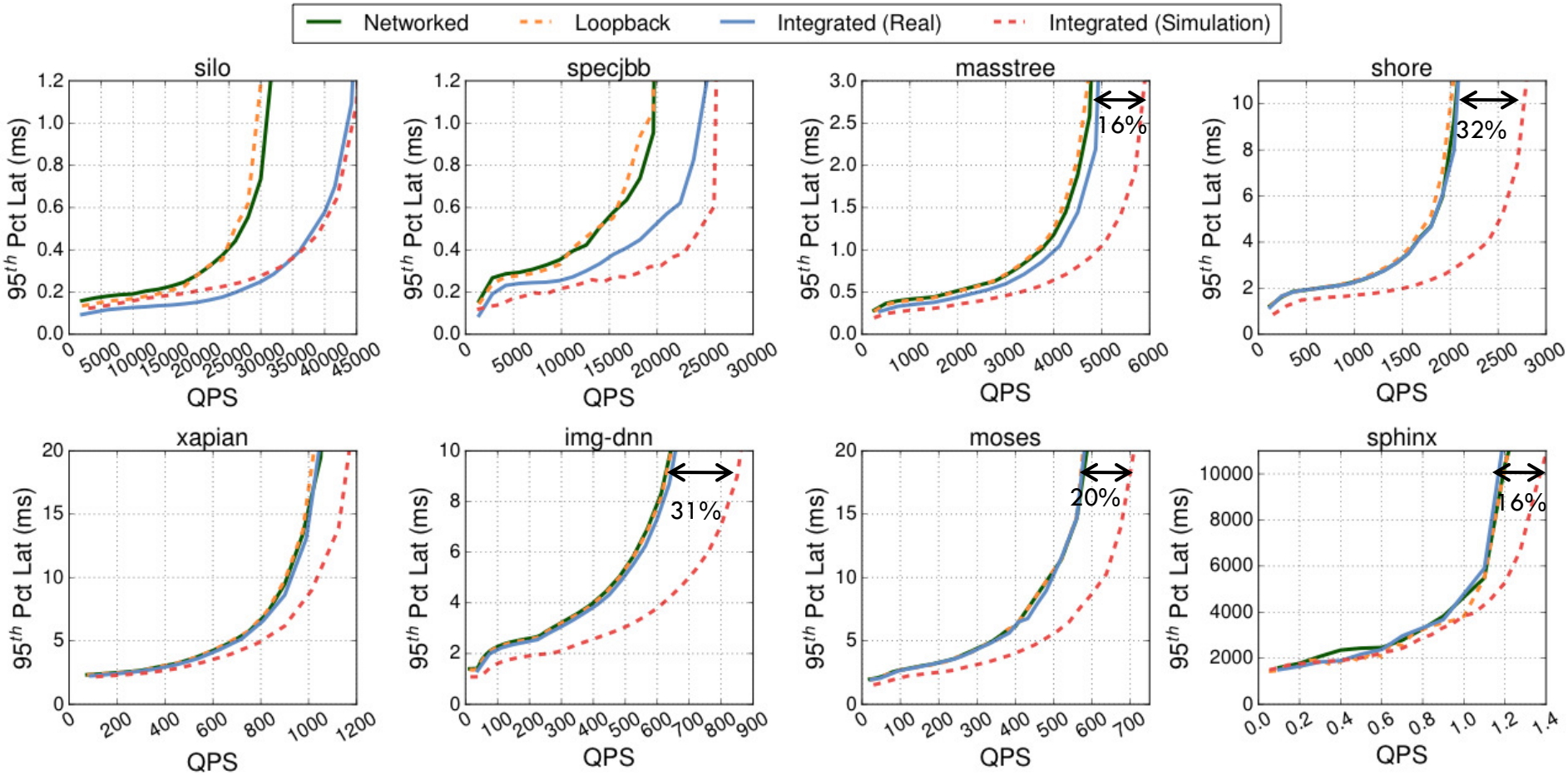
- Networked/Loopback configurations saturate earlier for applications with short requests (silo, specjbb)
  - TCP/IP processing overhead a significant fraction of request

Integrated Harness Configuration



- Application and client integrated into a single process
- ✓ Easy to setup
- X Some loss of accuracy
- ✓ Enables user-level simulations

Simulation vs. Real System



- Performance difference between real and simulated systems well within usual simulation error bounds
  - Average absolute error in saturation QPS: 14%
  - zsim IPC error for SPEC CPU2006 applications: 8.5 – 21%

Conclusions

- TailBench includes a diverse set of latency-critical applications with varied latency characteristics
- TailBench harness implements a statistically sound experimental methodology to achieve accurate results
- Various harness configurations allow trading off configuration complexity for some accuracy
  - Our results show that the integrated configuration is highly accurate for six of our eight benchmarks

Thanks For Your Attention!

Questions?

tailbench.csail.mit.edu

Massachusetts Institute of Technology