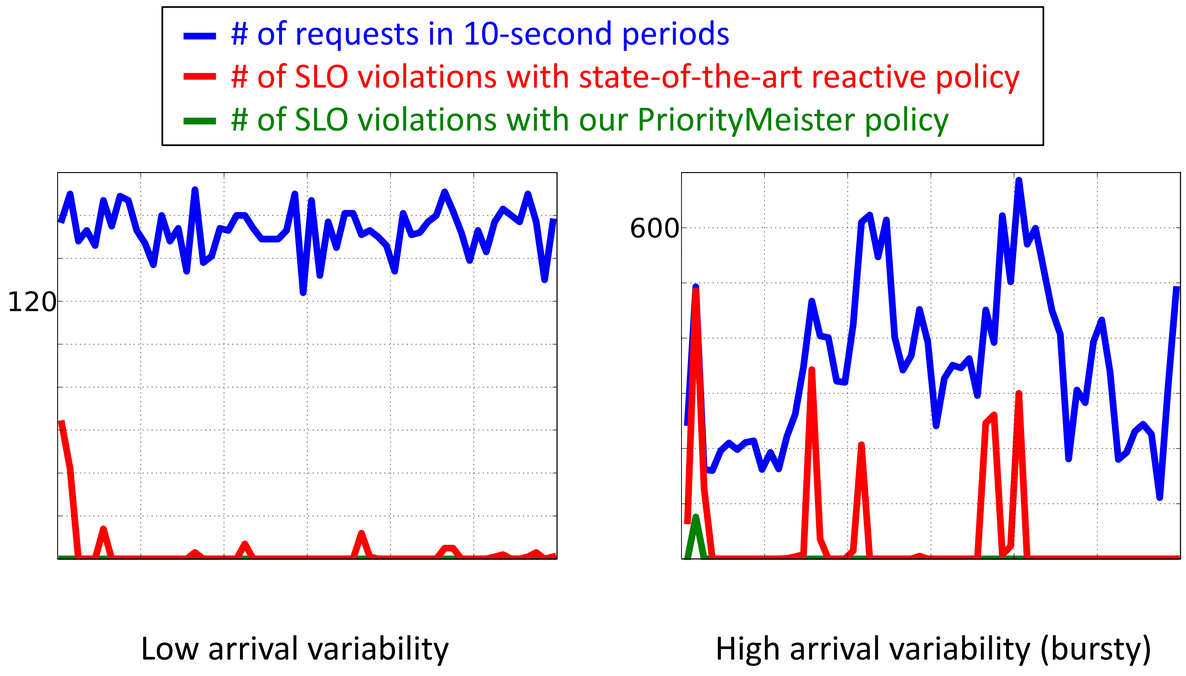
PriorityMeister: Tail Latency QoS for Shared Networked Storage

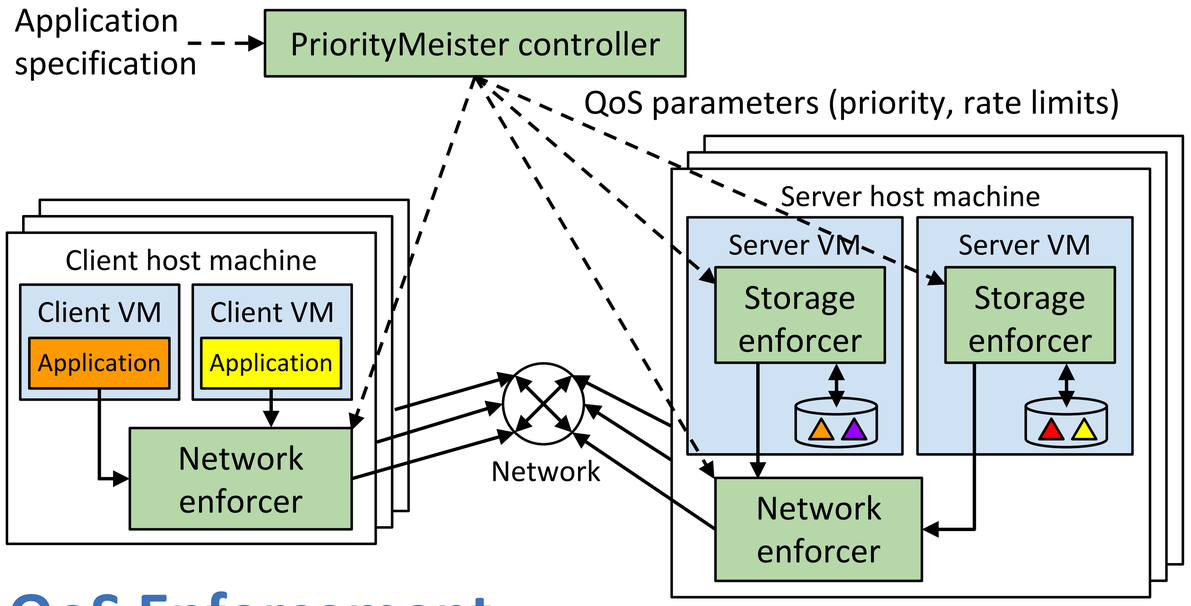
Timothy Zhu* Alexey Tumanov* Michael A. Kozuch† Mor Harchol-Balter* Greg Ganger* CMU* Intel Labs†

Problem/Motivation

- **Goal: Meet per-application tail latency Service Level Objectives (SLOs)**
  - **in shared networked storage infrastructures**
  - **with bursty applications**
- **Challenges:**
  - **End-to-end latency is affected by all stages (storage & network)**
  - **Bursts affect tail latencies of workloads sharing infrastructure**

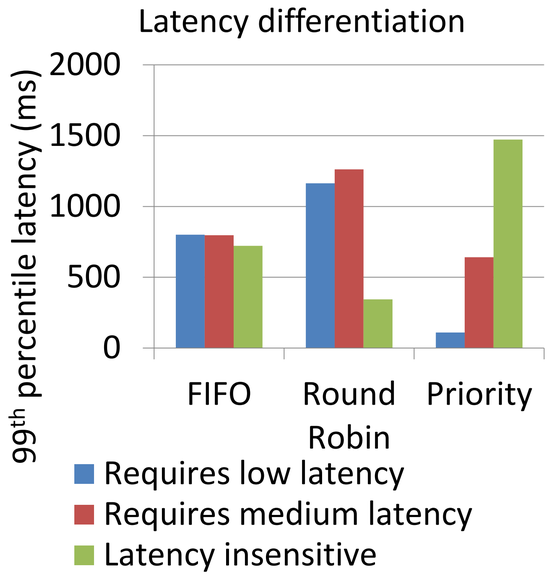


System Architecture

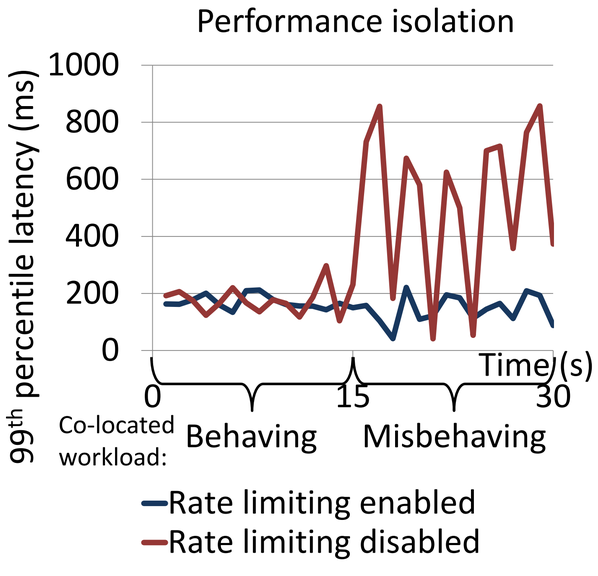


QoS Enforcement

- **Storage & network QoS is enforced at end-hosts only using local state**
  - **Priority - provides latency differentiation**
  - **Rate limits - provides performance isolation and prevents starvation**



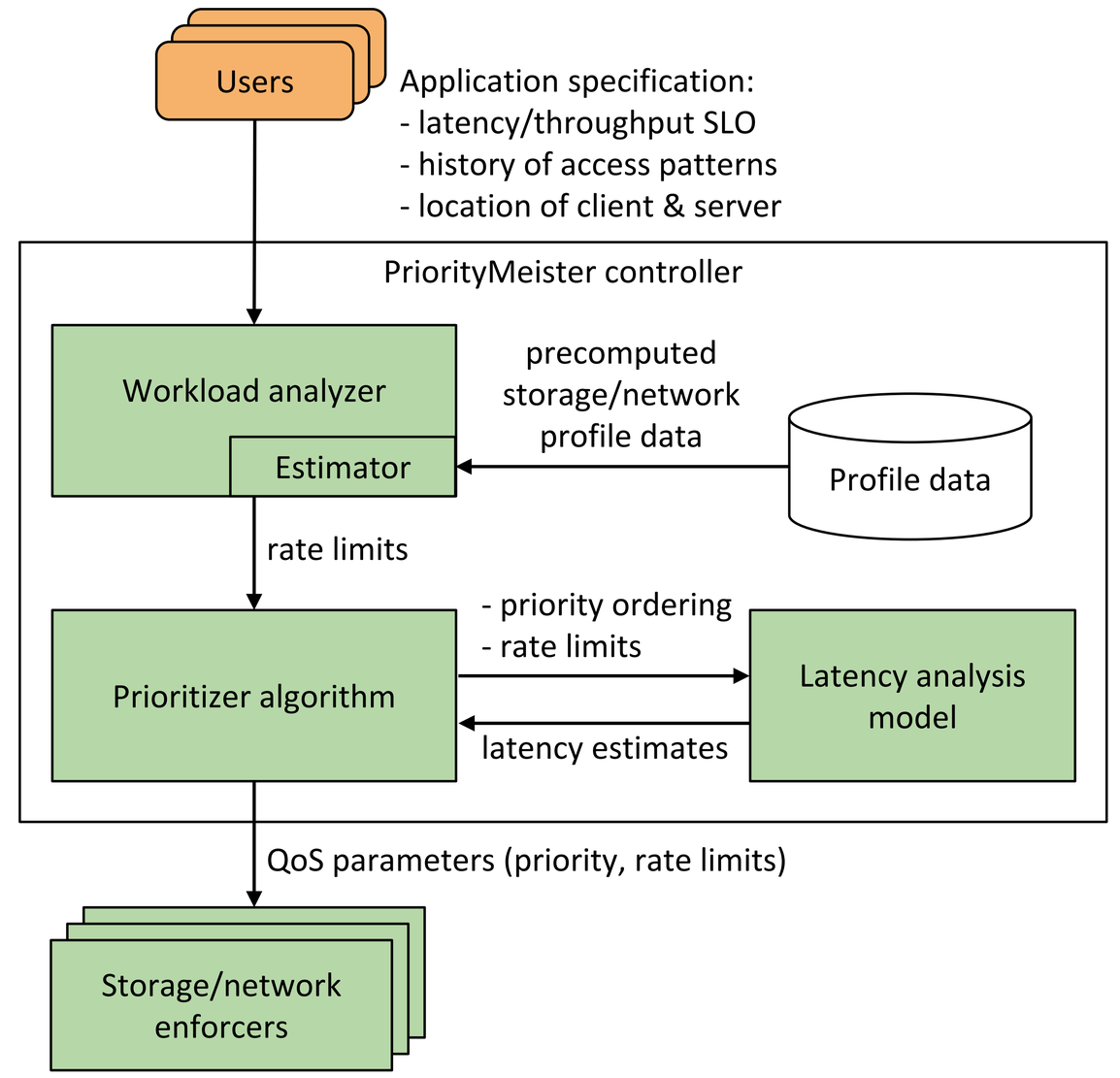
Priority reduces latency for workloads that care most



Rate limiting protects workloads from co-located misbehaving workloads

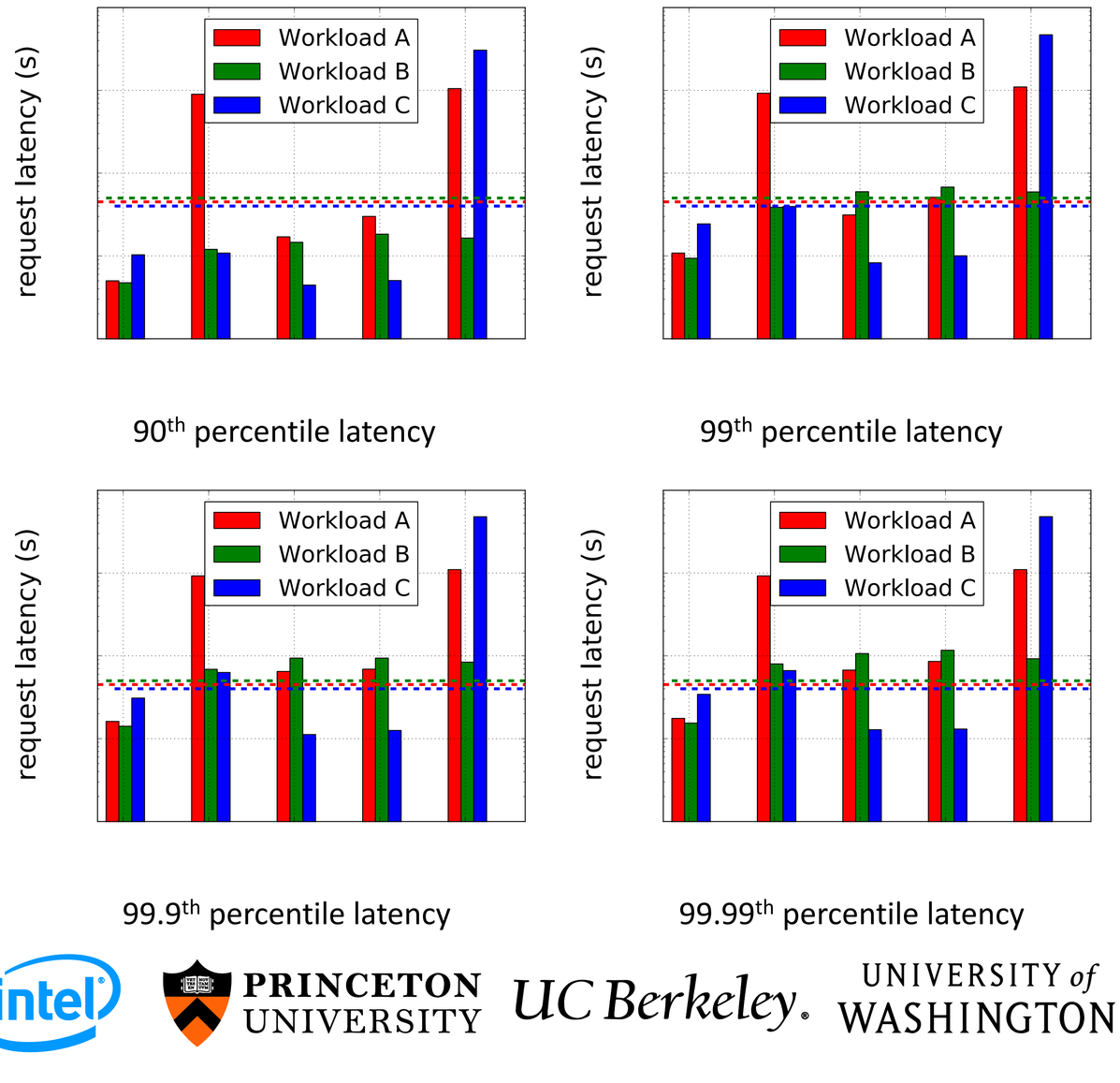
PriorityMeister Controller Design

- **Profiles workloads to proactively identify bottlenecks in storage & network**
- **Configures priorities and rate limits to meet SLOs under a worst-case model**
- **Application specification:**
  - **Latency SLO: Maximum acceptable latency of a request**
  - **Throughput SLO: Total time to complete a set of requests**
  - **History of access patterns: Trace of recent requests**
  - **Location of client & server: Pair of IP addresses for client and server VMs**



Results

- **Tail latencies for 3 co-located workloads across multiple policies**
- **Only PriorityMeister (PM) satisfies SLOs (dashed lines) across all percentiles**



Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University Georgia Tech intel PRINCETON UNIVERSITY UC Berkeley UNIVERSITY of WASHINGTON