### Problem/Motivation

- **Goal:** Meet per-application tail latency SLOs in shared networked storage infrastructures with bursty applications.

- **Challenges:**
  - End-to-end latency is affected by storage & network.
  - Bursts affect tail latencies of workloads sharing system.

### System architecture

- **Application requirements**
  - PriorityMeister controller
  - QoS parameters (priority, rate limits)

- **Server host machine**
  - Server VM
  - Storage enforcer
  - Network enforcer

- **Client host machine**
  - Client VM
  - Application

### QoS enforcement

- **Priority** – reduces latency for workloads that care most
- **Rate limits** – protects workloads from misbehaving workloads

### Results

- **PriorityMeister (PM) outperforms Cake, bySLO, EDF, ps**
- **Only PM meets SLOs for all workloads at tail**

### PriorityMeister controller design

- Analyzes workloads to identify bottlenecks in storage & network.
- Configures priorities and rate limits to meet SLOs.

### Graphs

- **90th percentile latency**
- **99th percentile latency**
- **99.9th percentile latency**
- **99.99th percentile latency**