PriorityMeister: Tail Latency QoS for Shared Networked Storage

SOCC 2014

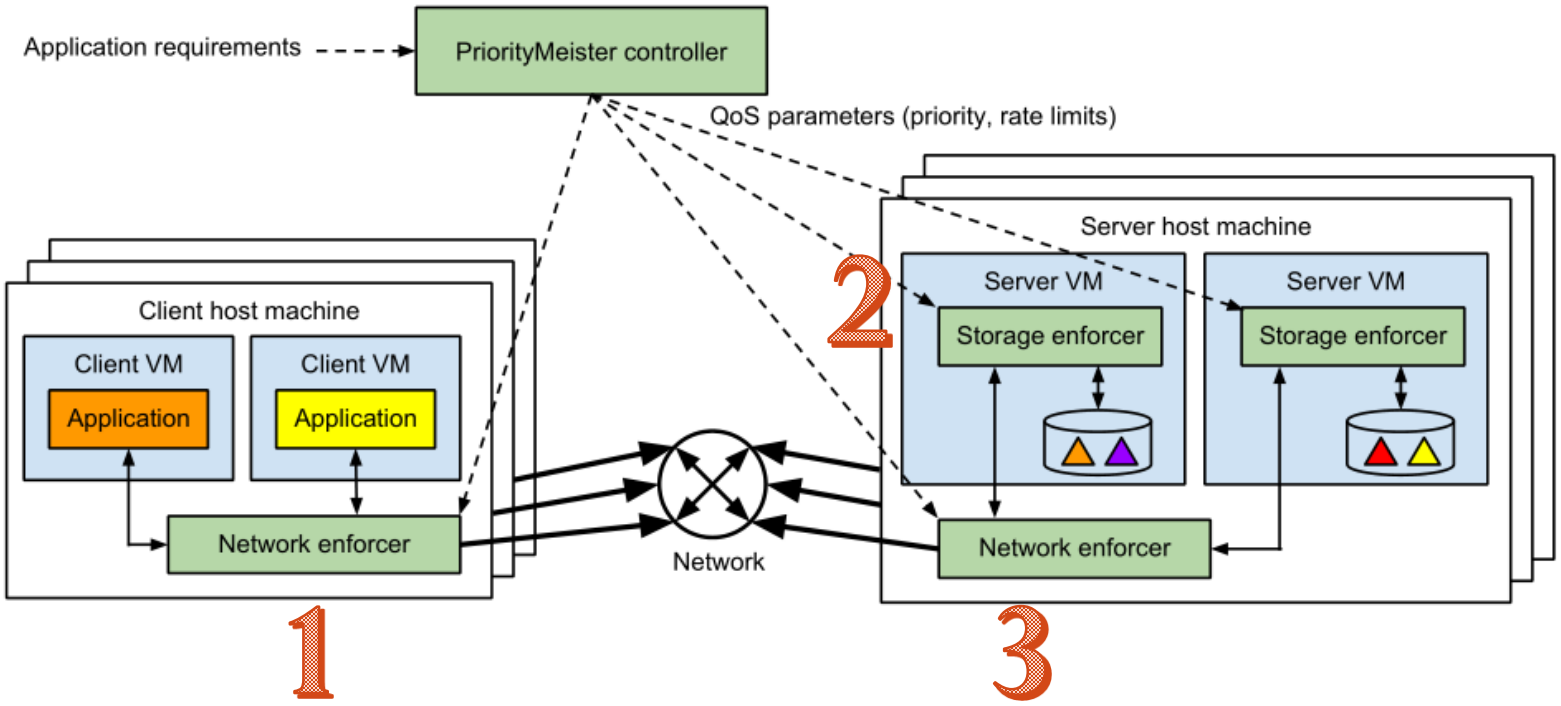
Timothy Zhu Alexey Tumanov Michael A. Kozuch

Carnegie Mellon University , Intel Labs

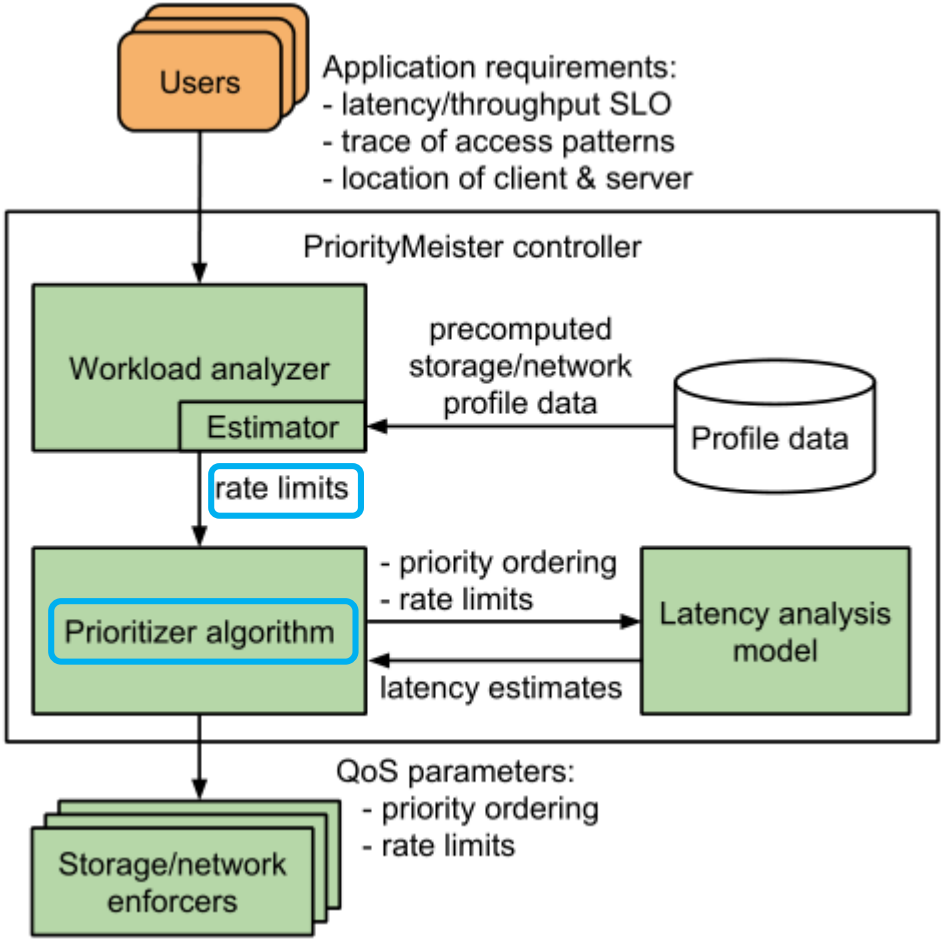
Introduction

- End-to-end tail latency QoS in a shared networked storage system is an important problem
- Normally, one might look at tail latency at the 90th or 95th percentiles
- Increasingly, researchers and companies like Amazon and Google are starting to care about long tails at the 99th and 99.9th percentiles

Architecture



Controller

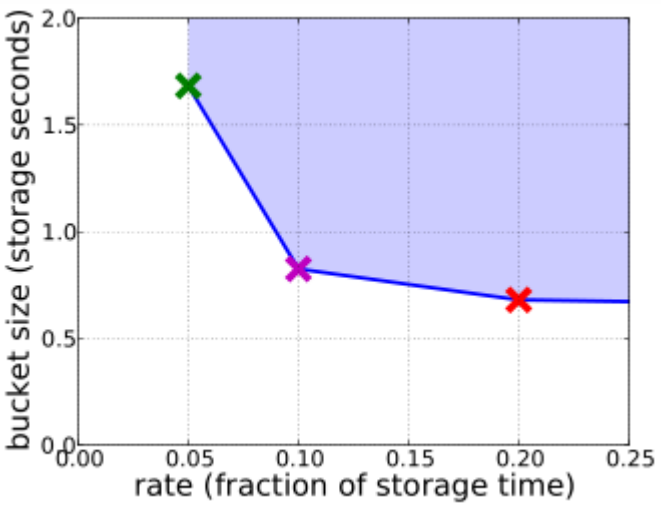


- 输入
  - Goal
  - Access pattern
  - Location of client & server
- 输出
  - Priority
  - Rate limits

Rate limits

- Rate (r)
- Token bucket size (b)
- Rate limit (r, b)

b

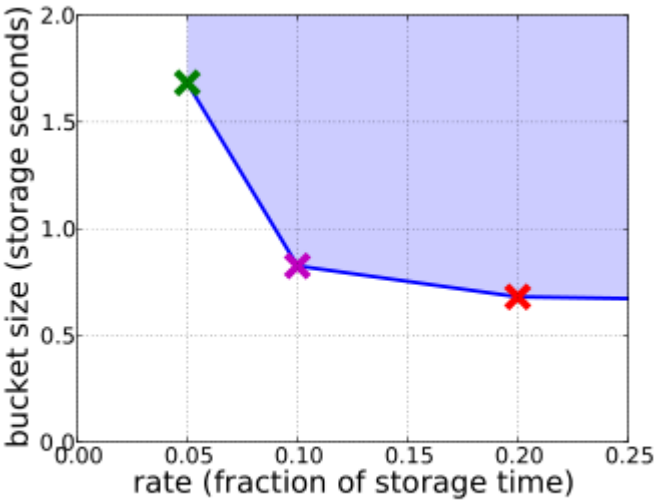


r

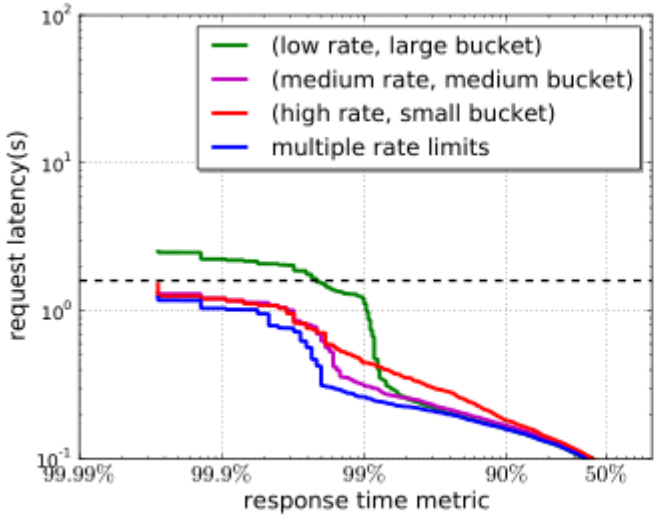
W run in an unfettered manner

Rate limits

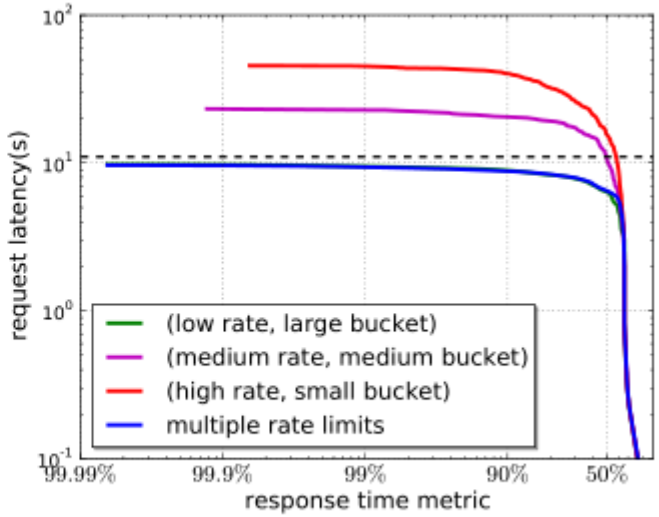
- How (r, b) limit the effect



(a) rate limit pairs of high priority workload



(b) latency of medium priority workload



(c) latency of low priority workload

Implementation

- **(r, b)**

Select a rate r and replay the workload's trace using an unbounded token bucket with the selected rate.

- **Determine the number of tokens**

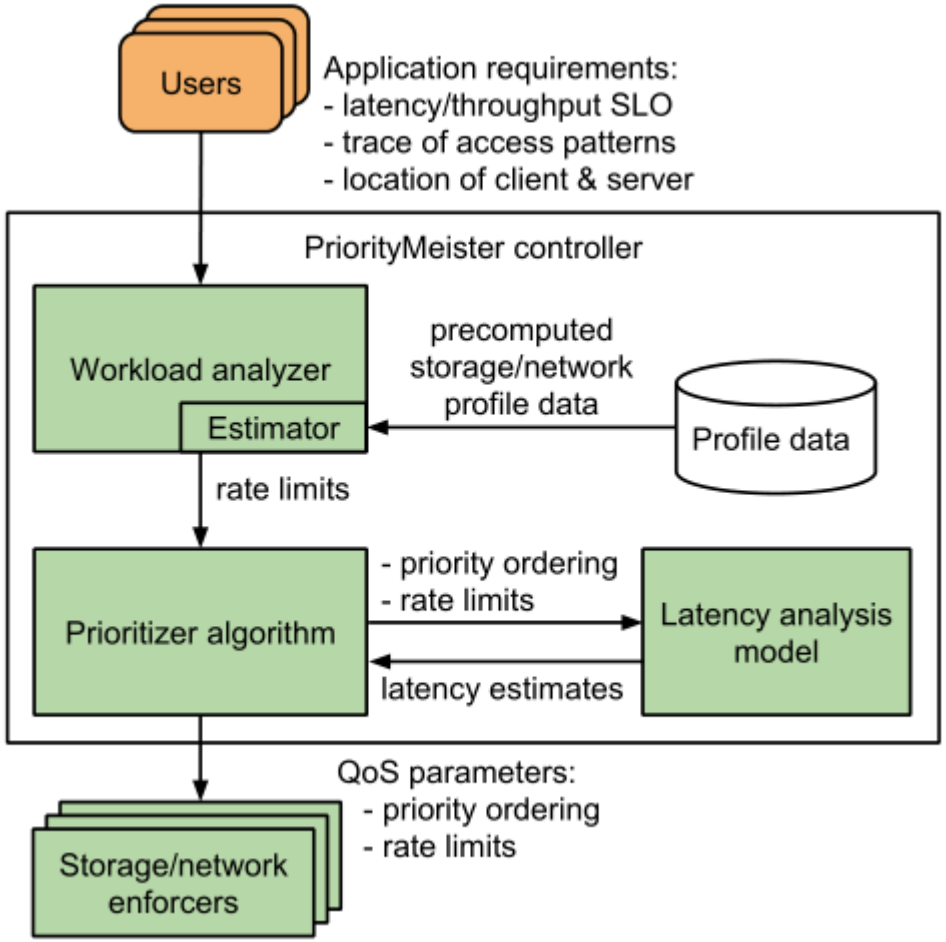
Storage: $\frac{request\ size}{bandwidth}$ + base time

Network: number of bytes

- **Prioritizer algorithm**

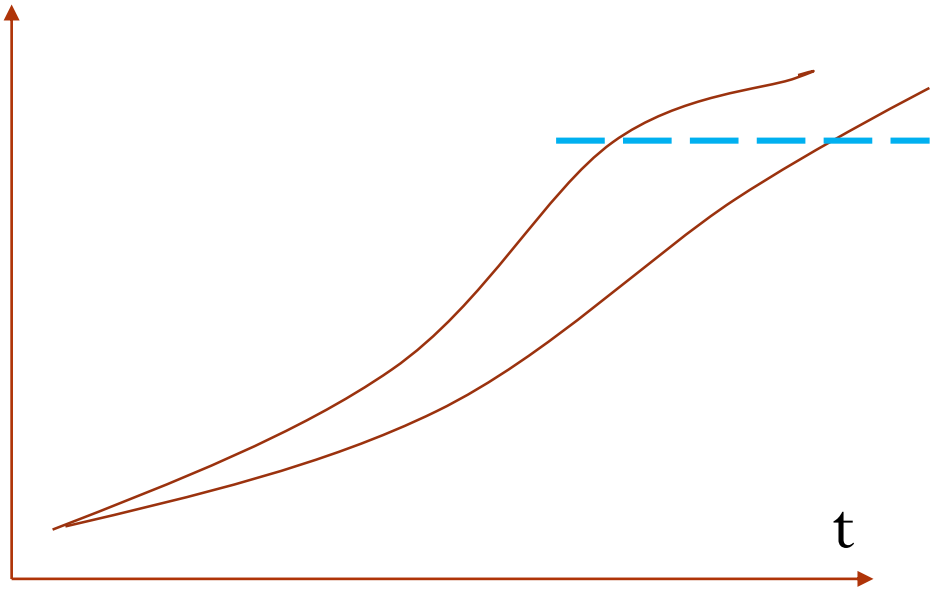
Assign the lowest priority to each workload

Review of controller



Latency analysis model

- Arrival curve and service curve
- An arrival curve $\alpha(t)$ is a function that defines the maximum number of bytes that will arrive in any period of time t
- A service curve $\beta(t)$ is a function that defines the minimum number of bytes that will be serviced in any period of time t



Evaluation

- traces

Workload label	Workload source	Estimated storage load	Estimated network load	Interarrival Variability, C_A^2
Workload A	DisplayAds production trace	5%	5%	1.3
Workload B	MSN storage production trace	5%	5%	14
Workload C	LiveMaps production trace	55%	5%	2.2
Workload D	Exchange production trace (behaved)	10%	5%	23
Workload E	Exchange production trace (misbehaved)	> 100%	15%	145
Workload F	Synthetic low burst trace	25%	5%	1
Workload G	Synthetic high burst trace	25%	5%	20
Workload H	Synthetic very high burst trace	25%	5%	40
Workload I	Synthetic medium network load trace 1	35%	20%	1
Workload J	Synthetic medium network load trace 2	45%	25%	1
Workload K	Synthetic ramdisk trace	N/A	35%	3.6
Workload L	Synthetic large file copy	N/A	N/A	N/A

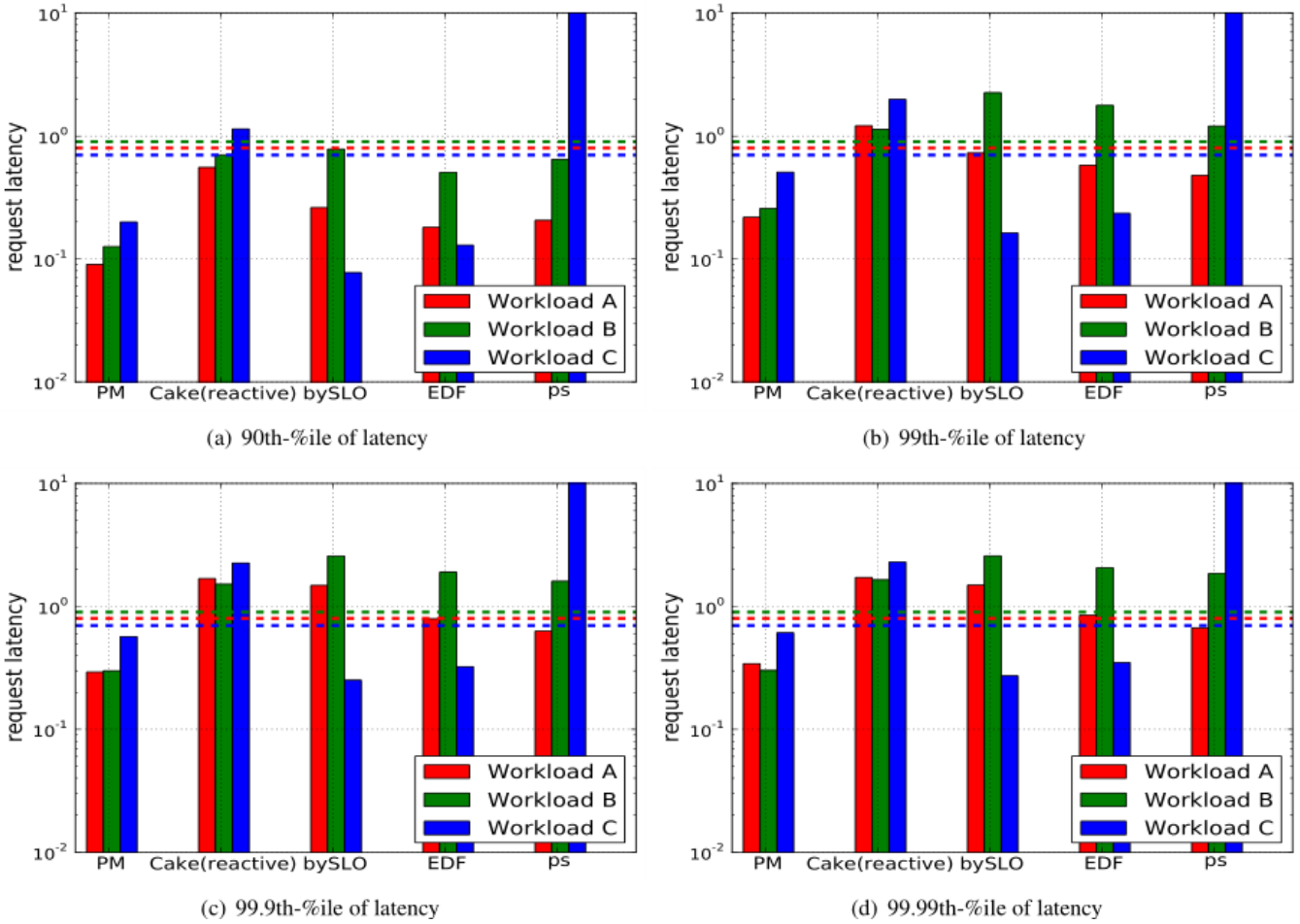
Table 2. Workload traces used in our evaluation.

Policies

- Proportional fair-share (ps)
- Cake
- Earliest Deadline First(EDF)
- prioritization in order by SLO (bySLO)
- Priority Meister (PM)

PM tail latency performance

- A, B, C and L co-located



(a) 90th-%ile of latency

(b) 99th-%ile of latency

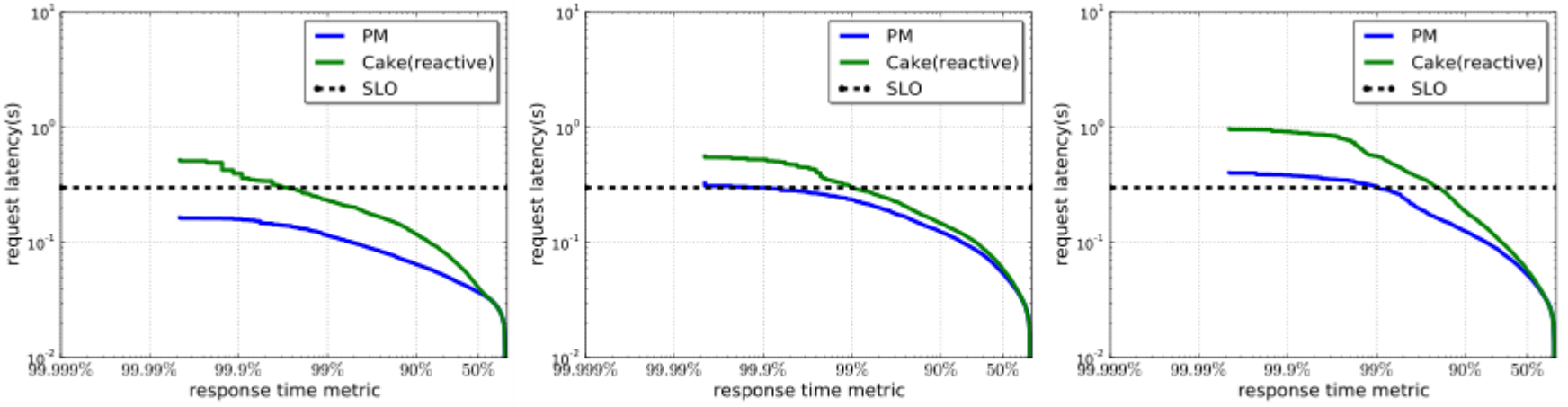
(c) 99.9th-%ile of latency

(d) 99.99th-%ile of latency

Coping with burstiness

- micro-benchmark（different burstiness）F, G, H

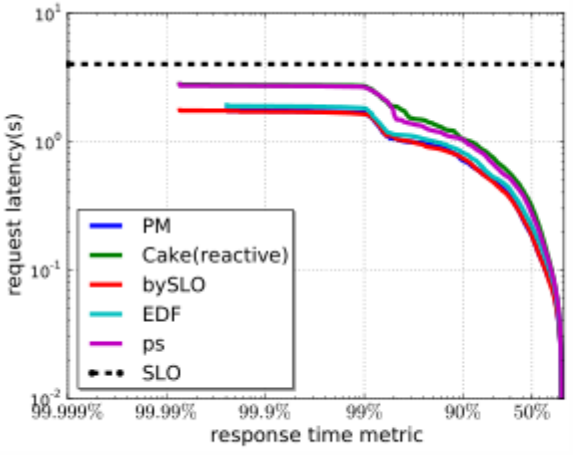
Workload F	Synthetic low burst trace	25%	5%	1
Workload G	Synthetic high burst trace	25%	5%	20
Workload H	Synthetic very high burst trace	25%	5%	40



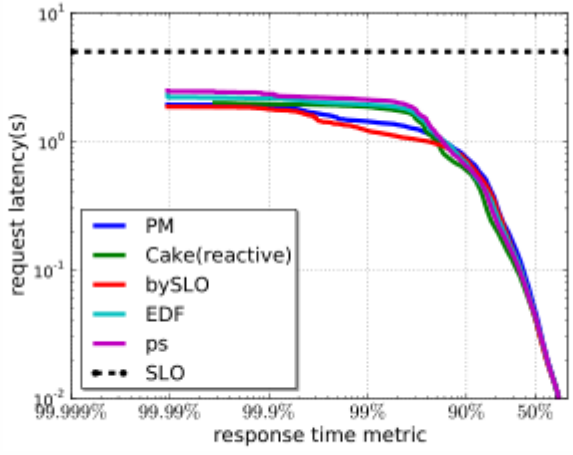
(a) low burstiness, $C_A^2 = 1$, Workload F (b) high burstiness, $C_A^2 = 20$, Workload G (c) very high burstiness, $C_A^2 = 40$, Workload H

Misbehaving workloads

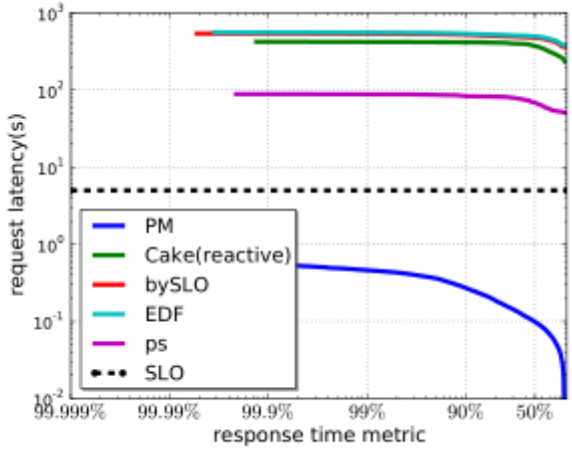
- Well-behaved(D) to misbehaved(E)



(a) Workload D, behaved



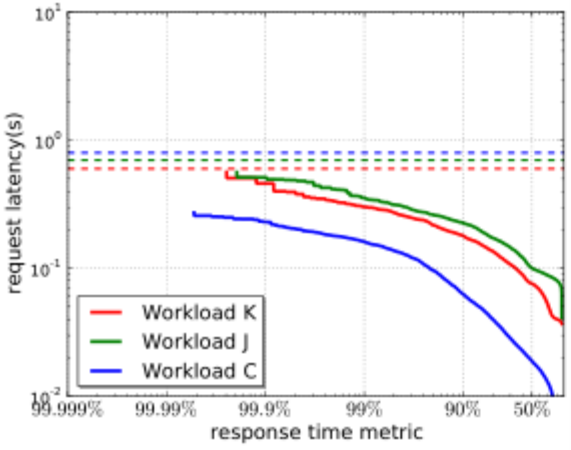
(b) Workload C, behaved



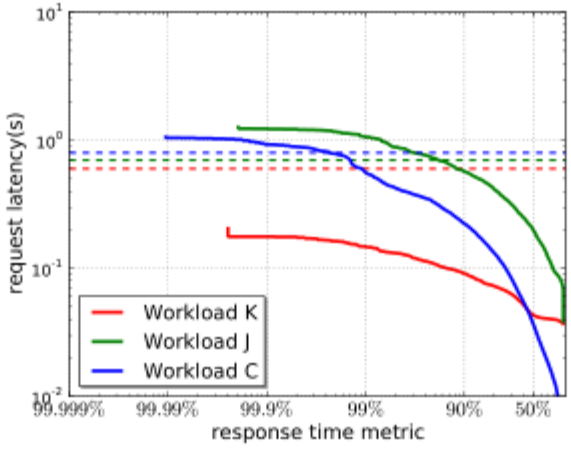
(c) Workload C when Workload D misbehaves

Multi-resource performance

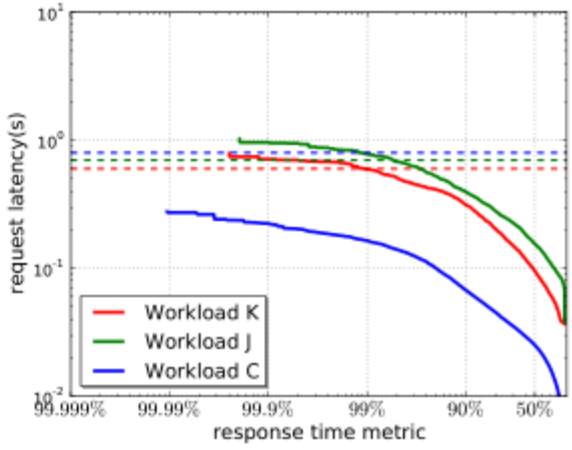
- 4 workloads on servers with a ramdisk and multiple disks



(a) PM: all workloads' SLOs satisfied



(b) bySLO: green SLO violated @ 95th-%ile



(c) no QoS: green SLO violated @ 98th-%ile

Related Work

Scheduler	Latency SLO	Multi-resource
Argon [20]	No	No
SFQ(D) [11]	No	No
AQuA [22]	No	No
mClock [10]	No	No
PARDA [9]	No	Yes
PISCES [18]	No	Yes
Maestro [16]	Average latency	No
Triage [12]	Average latency	No
Façade [15]	Average latency	No
pClock [8]	Average latency	No
Avatar [25]	95th percentile	No
Cake [21]	99th percentile	Yes
PriorityMeister	$>$ 99th percentile	Yes

Related Work

	场景	方法
PM	共享网络存储, nfs	Priorities + rate limites
Cake	分布式共享存储 Hbase+HDFS	2级调度， priority+split+throughput contract + feedback
mClock	Hypervisor IO调度, Vmware ESX	Proportional Shares+reservations+limits
Pisces	共享键值存储, membase	Partition Placement+Weight Allocation+Relica Selection+Weighted Fair Queueing
FAST	共享块存储	不同类型的IO选择不同的副本节点
IOFlow	IO路径 18+ different layers	不同的stage实施策略

Conclusion

- 构建了一个QoS系统，包含存储和网络
- 在每个stage自动分配优先级以及速率限制
- 良好的鲁棒性：存储性能错估、不同程度的负载突发性以及workload misbehavior

VM share ssd block storage