Façade: virtual storage devices with performance guarantees

Christopher R. Lumb, Arif Merchant, Guillermo A. Alvarez
Hewlett-Packard Laboratories

Presentation: Jeff Pound

Outline

- Introduction
- Service Level Objectives (SLO)
- Façade
- Empirical Results
- Discussion

The Problem

- Rapidly-changing workloads compete for access to common storage devices

- Workloads…
  - are independent
  - require a predictable **Quality of Service** (QoS)

Goals

- Performance Guarantees
  - Each workload should get the performance specified by it's *Service Level Objective (SLO)*
  - The performance experienced by a workload should not suffer from variations of other workloads
- Achieve the best utilization of physical resources possible

Common Approaches

- Over-provision resources to ensure QoS can be met for each workload
  - Expensive
  - Poor utilization of resources
- Assign each workload to it's own physical resource
  - No fault-tolerance
  - Still a poor utilization of resources

Façade's Approach

- Allow a virtual I/O layer to *schedule* the I/O requests from each workload
- Throttle the device *queue length* to control latency at the device and maximize throughput of the system
- Specify a *Service Level Objective* for the system to meet

Service Level Objective

- The Service Level Objective (SLO) is defined as:
  - two curves: read and write latency as a function of request rate
  - Window length *w* (time is divided into epochs of length *w*)
  - $((r_1,tr_1,tw_1), (r_2,tr_2,tw_2), \ldots, (r_n,tr_n,tw_n))$
    - r = I/Os / second ($0 < r_1 < r_2 < \ldots < r_n$)
    - tr = target read latency
    - tw = target write latency

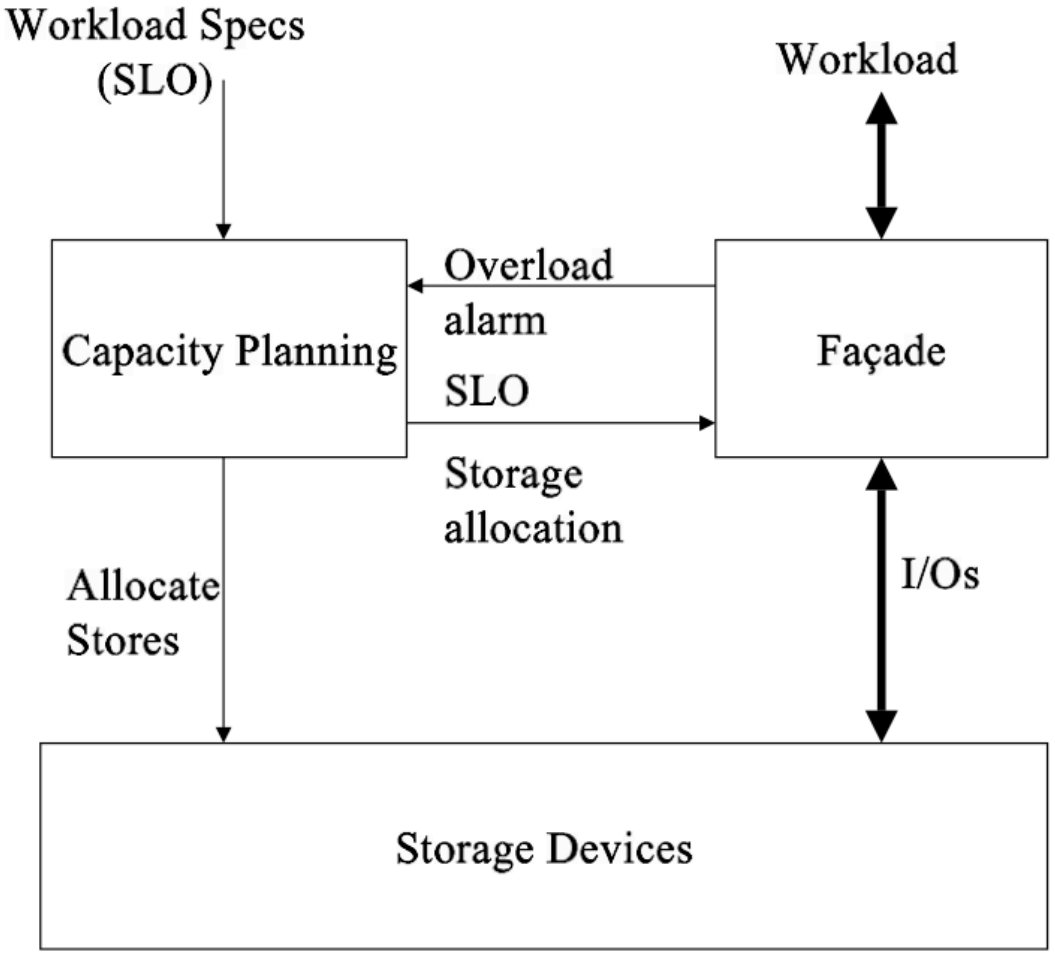
Service Level Objective

- The measured latency is *averaged* over the time window
  - Latency should not exceed the calculated *target latency*

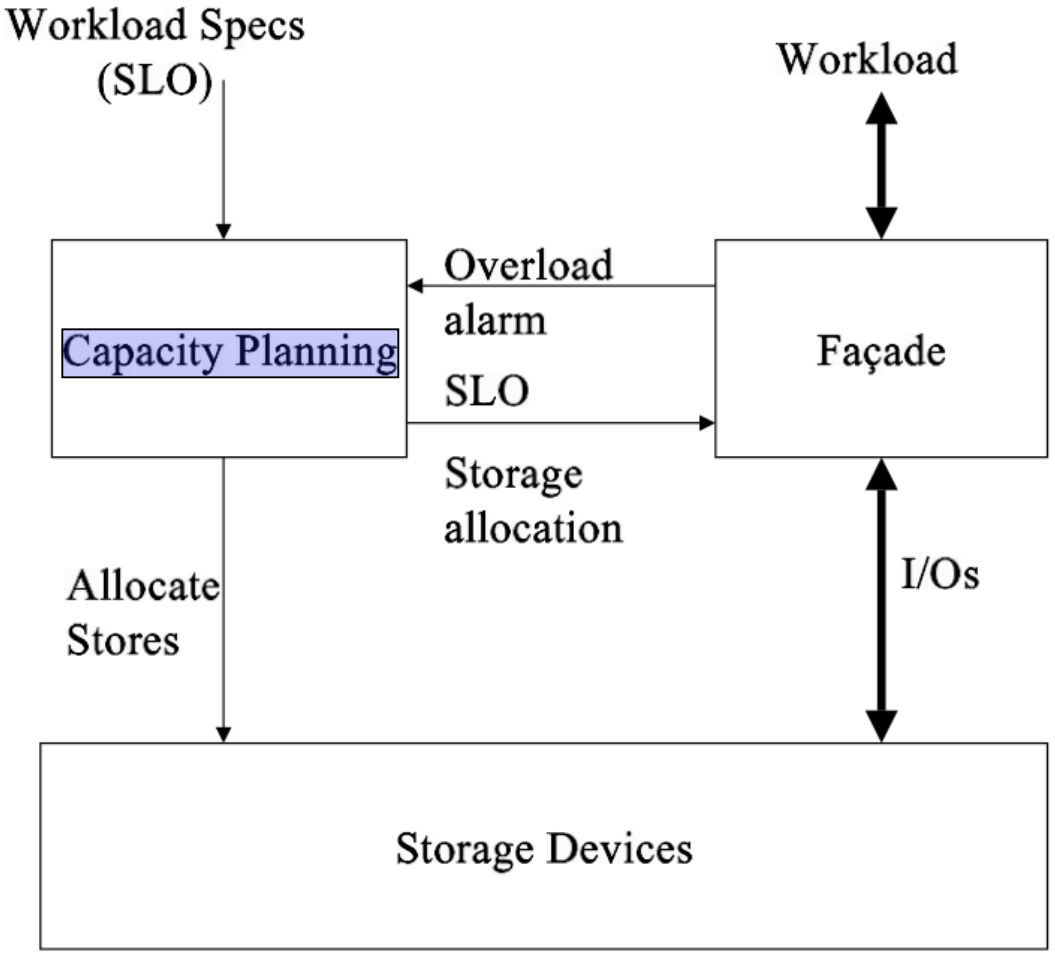
Façade

- Real-time scheduling of I/O requests.
  - Earliest Deadline First (EDF) scheduling
- Feedback-based control of the length of the storage device queue.
  - Increase length => increases overall throughput (better device utilization)
  - Decrease length => reduces the latency at the device

Façade in a Storage Management System

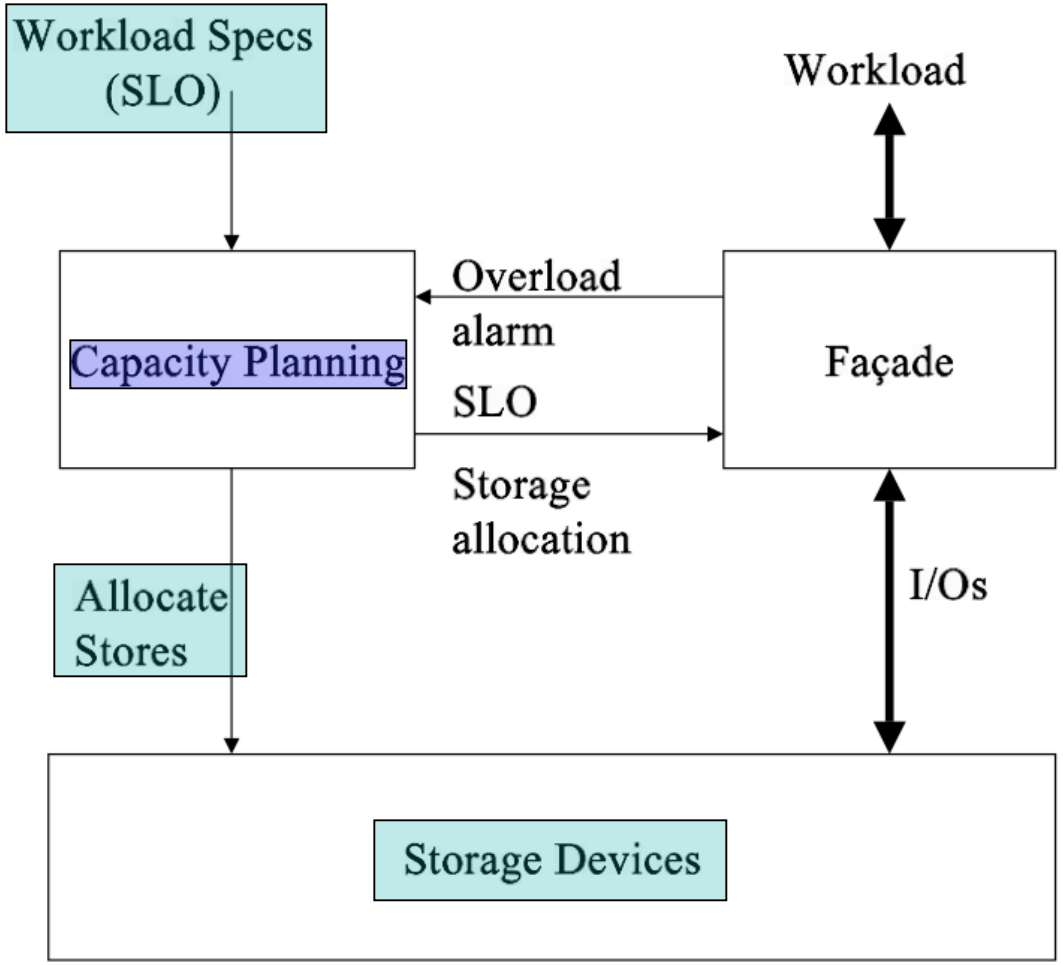


Façade in a Storage Management System



-Capacity Planner is assumed to exist.

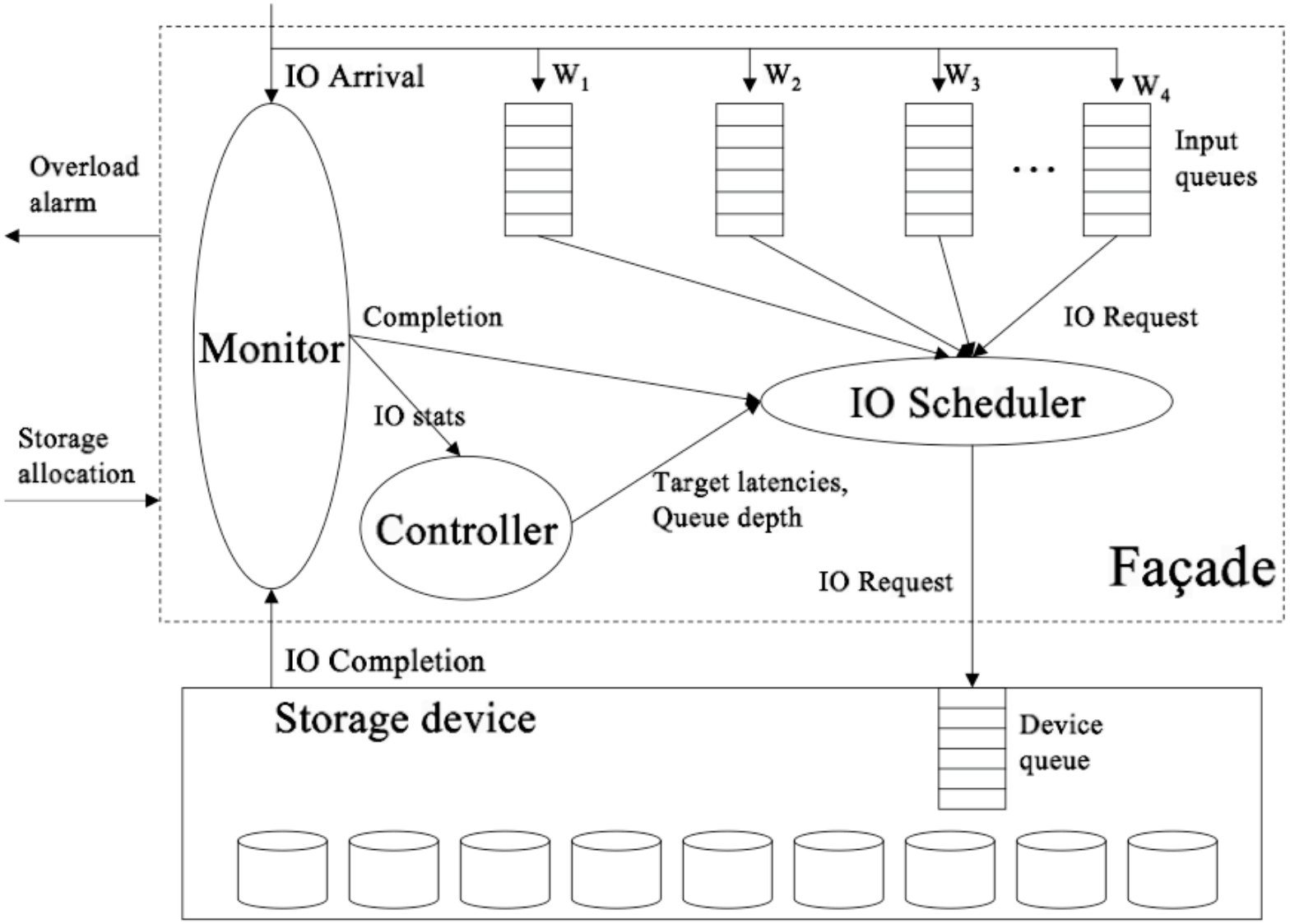
Façade in a Storage Management System



-Capacity Planner is assumed to exist.

-Façade assumes the physical devices are sufficient to handle the workloads

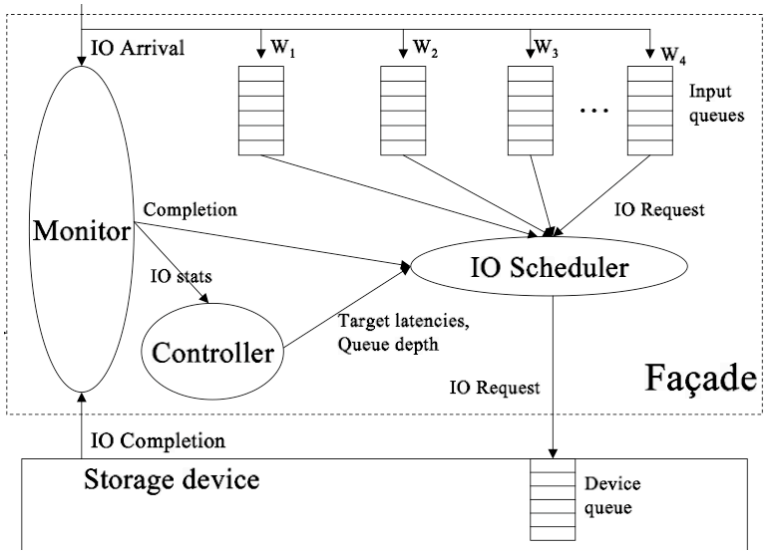
Façade Architecture



Monitor

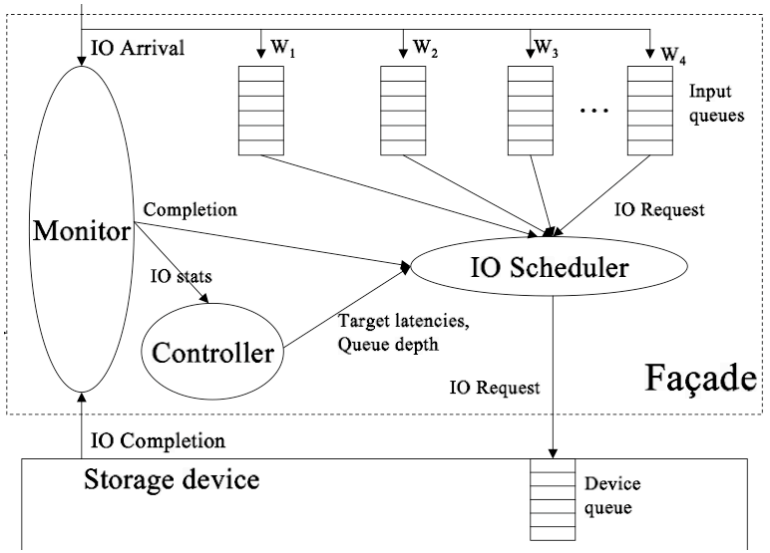
- Monitors
  - I/O arrivals
  - I/O completions
- Computes (for active workloads)
  - average latency
  - request rates

- Sends I/O stats to the *Controller*
- Notifies the *Scheduler* of completions



Scheduler

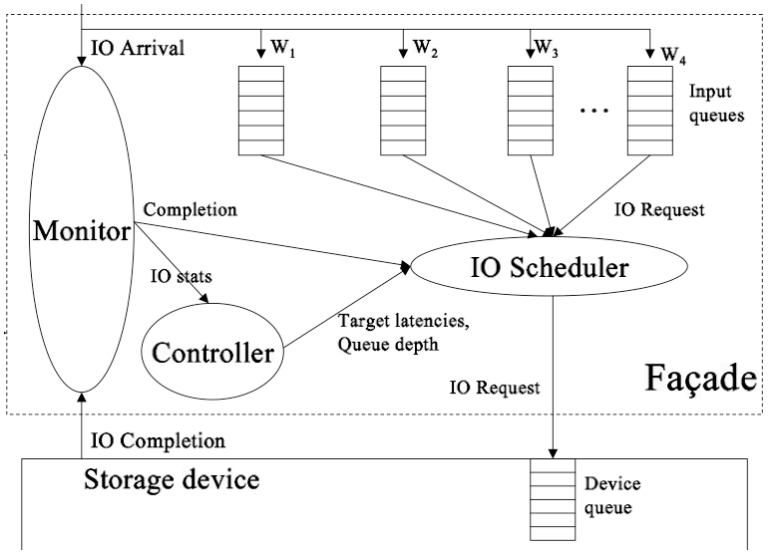
- Schedules I/O requests from workloads
  - EDF scheduling: deadline for a workload is the deadline of it's oldest pending request
- Maintains
  - Target latencies
  - Target queue length



Scheduler

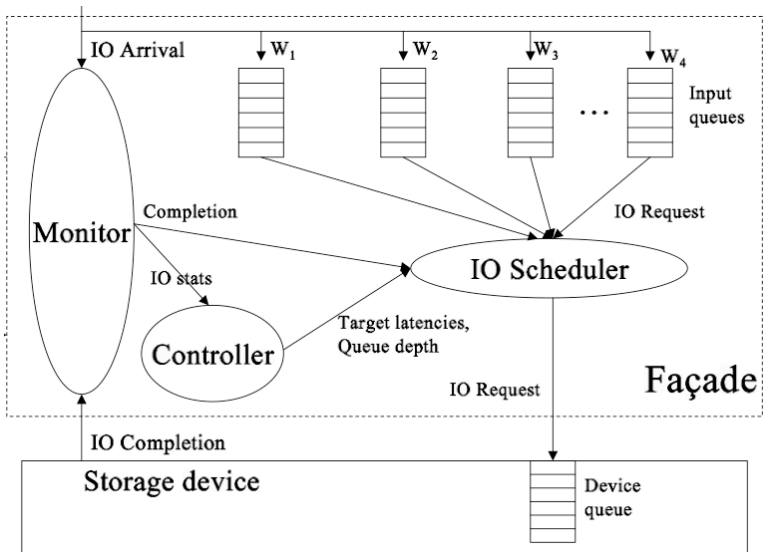
Admits I/O requests to the device queue

1) If the *queue depth* is less than the *target queue length*
2) If the deadline for any workload has past (independent of queue depth)



Controller

- Periodically calculates
  - *target workload latencies*
    - Based on SLO and current request rates
  - *target queue length*
    - Based on latencies



Controller: target latencies

- Given an SLO
  - $((r_1, tr_1, tw_1), (r_2, tr_2, tw_2), \ldots, (r_n, tr_n, tw_n))$
- Let $r_0 = 0$, $r_{n+1} = tr_{n+1} = tw_{n+1} = \infty$
- Let f_r be the fraction of reads

Controller: target latencies

- Given an SLO
  - $((r_1,tr_1,tw_1), (r_2,tr_2,tw_2), \ldots, (r_n,tr_n,tw_n))$
- Let $r_0 = 0$, $r_{n+1} = tr_{n+1} = tw_{n+1} = \infty$
- Let f_r be the fraction of reads

$$latencyTarget(W_k) = tr_i f_r + tw_i(1 - f_r)$$
$$\text{if } r_{i-1} \leq readRate(W_k) + writeRate(W_k) < r_i$$

Controller

- *target latencies* have been calculated
- Actual latencies have been measured

- Now the *target queue length* can be adjusted to control latency at the device, while maximizing overall throughput

Controller: queue length

$$E = \min_k \frac{latencyTarget(W_k)}{L(W_k)}$$

If E ≥ 1, we are doing good

If E < 1, our latency is bigger than our target

$$Q_{new} = \begin{cases} E \cdot Q_{old} & \text{if } E < 1, \\ (1+\varepsilon)Q_{old} & \text{else if } Q_{max} = Q_{old}, \\ Q_{old} & \text{otherwise.} \end{cases}$$

System Summary

- Keep track of the latencies experienced by all workloads
- Calculate the current *target latency* for each workload based on it's current request rate
- Adjust *target queue length to:*
  - Reduce latency if targets are not being met
  - Increase throughput of the system otherwise

Experimental Evaluation

- SLO Compliance
- Performance isolation
- Maximum SLO (meeting the most stringent workload a logical unit can support)
- Multiplexing
- Resource utilization
- Façade overhead
- Performance during failure

Discussion

- Does increasing the target queue length really increase throughput?
- What about workloads that push more I/Os than their service level allows?
  - Service will be cut-off until enough time passes for the I/O rate to drop, even if the physical device can support the load.
- How do we choose the length of the time window (*w*).