POSTECH

Regulating I/O Performance of Shared Storage with a Control Theoretical Approach

Junkil Ryu (Han Deok Lee)

lancer@postech.ac.kr

NASA/IEEE MSST 2004
12th NASA Goddard/21st IEEE Conference on Mass Storage Systems & Technologies
The Inn and Conference Center
University of Maryland University College
Adelphi MD USA
April 13-16, 2004

IEEE COMPUTER SOCIETY

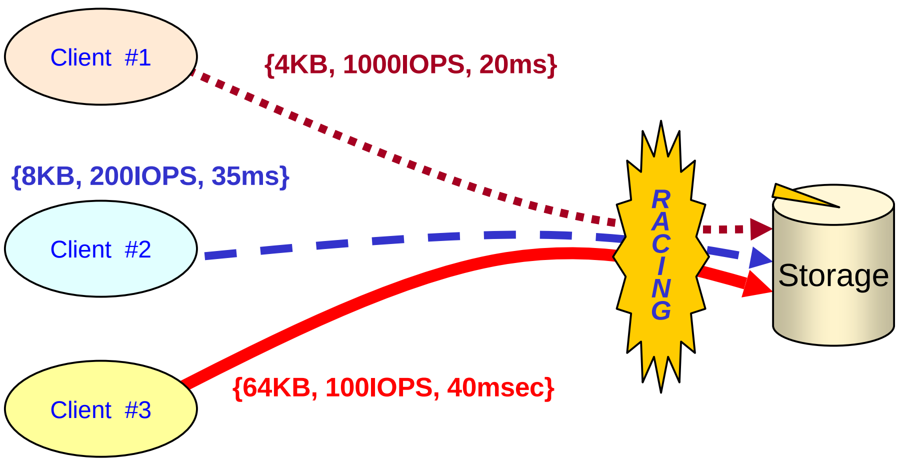
Talk Outline

- Introduction
- Proposed Scheme
- Performance Evaluations
- Conclusion & Future Work

Introduction

❑ Why Regulating I/O Performance?

- Different clients demand different types of storage services
- When multiple clients share storage, a racing problem may occur
- However, storage itself doesn't provide any solution to the problem



Introduction

Problem Description

- Given
  - a set of (storage) clients that share the same storage
  - demanded storage services (QoS) for each client

- Devise a control scheme that
  - assures the demanded storage services (statistically)
  - keeps the storage utilized as high as possible

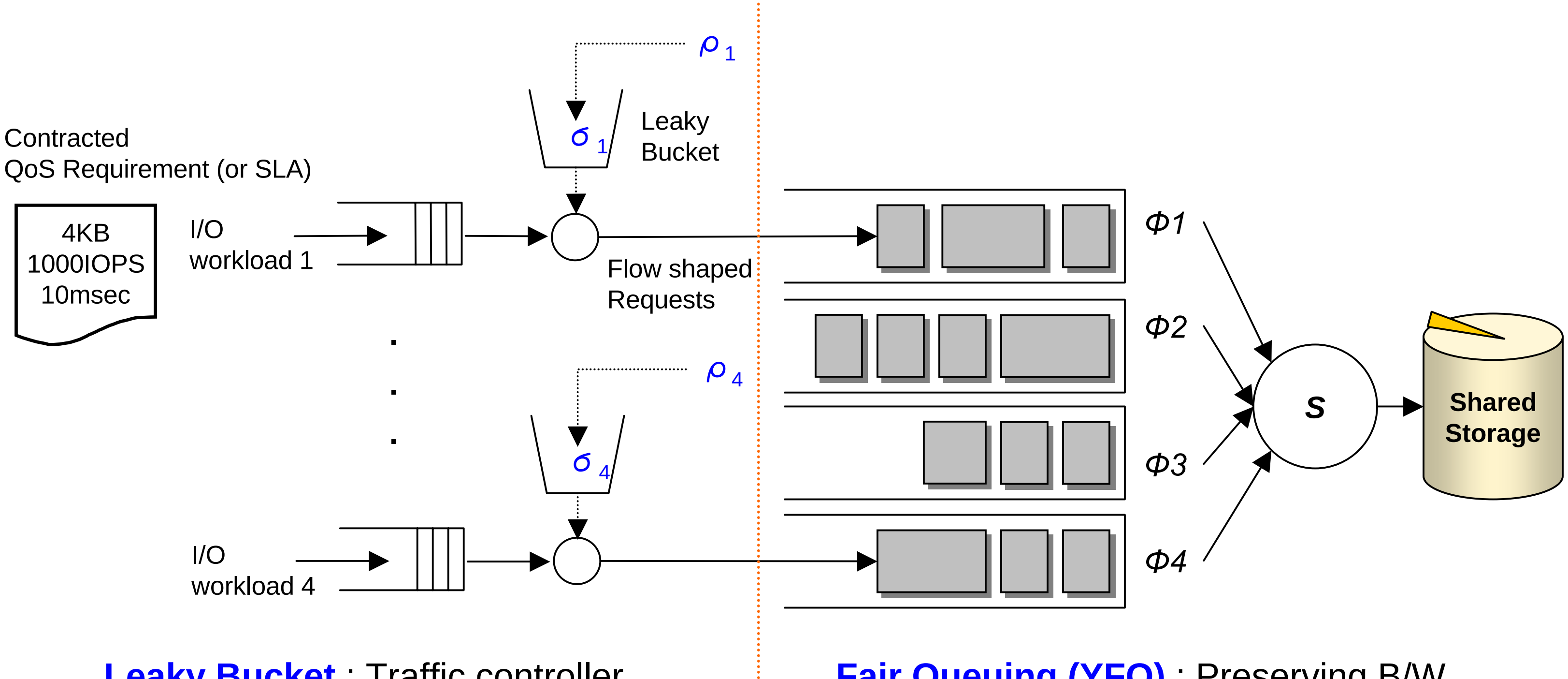
Specification of Storage Service per Client

- Request size
- Target IOPS
- Target response time

Introduction

❑ Previous Solution in Network Domain – FQ w/ "Leaky Bucket"

- Static I/O traffic policing
- Likely to under-utilize the storage resources

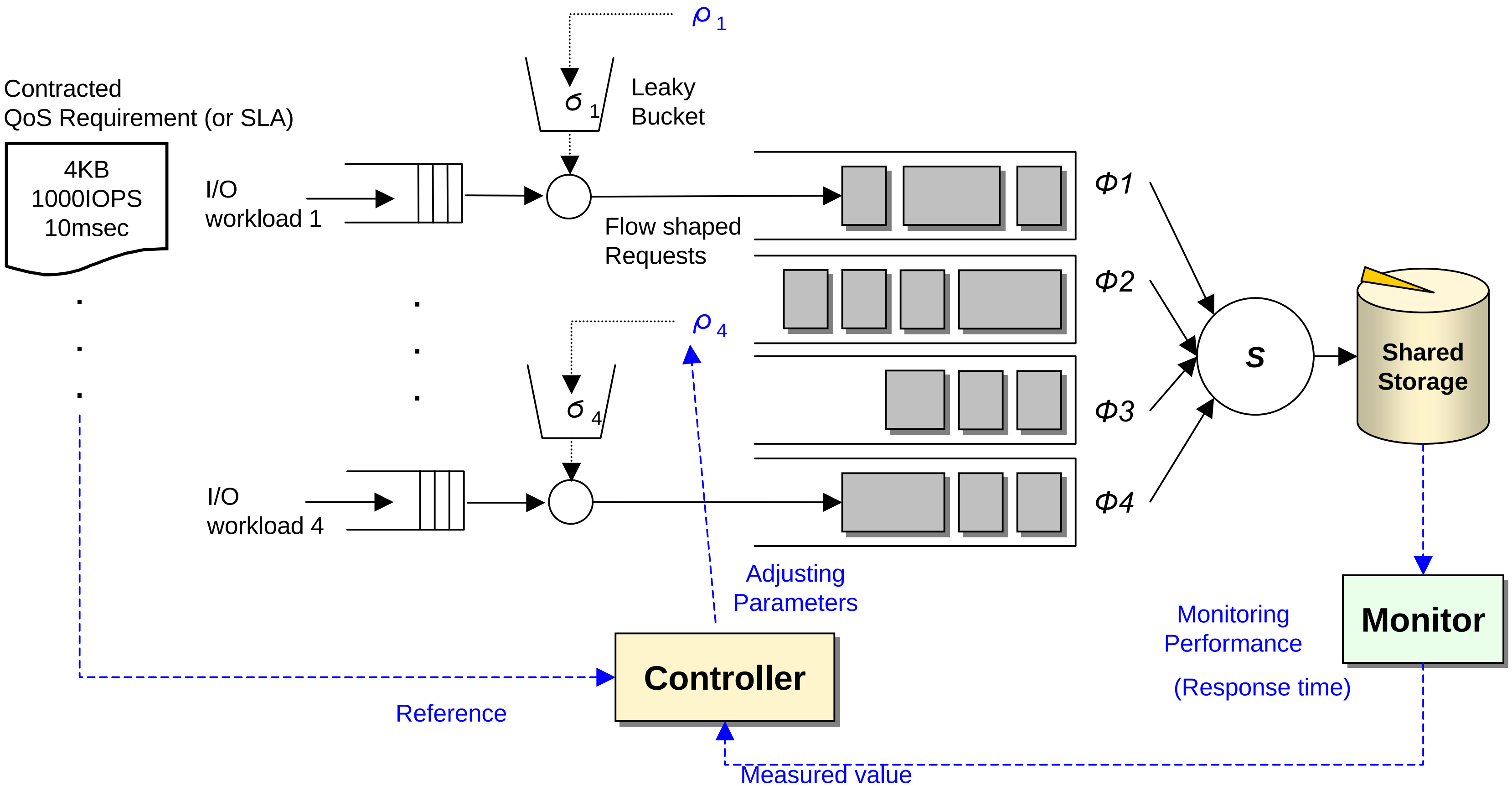


Leaky Bucket : Traffic controller

Fair Queuing (YFQ) : Preserving B/W

Proposed Scheme

- ❑ Our Solution – FQ w/ "Feedback-controlled Leaky Bucket"
  - Adjusting each $\rho_i(k)$ according to current RT
  - Maximizing the utilization of storage resources (w/ better perf.)

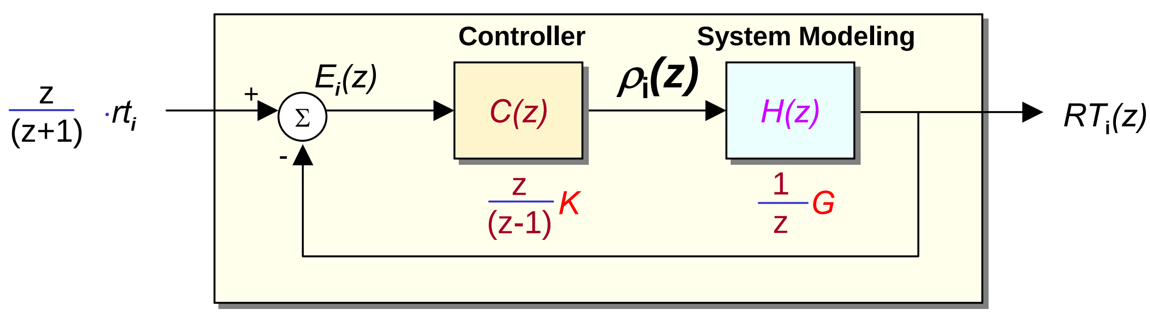


Proposed Scheme

❑ Controller Design

- Estimating error: $E_i(k) = \mathrm{rt}_i - RT_i(k)$
- Computing LB param. of $\rho_i(k)$: $\rho_i(k) = \rho_i(k-1) + K \cdot E_i(k)$ $\dashrightarrow$ $\rho_i(z) = \boxed{\frac{z}{(z-1)} K} \cdot E_i(z)$
- $RT_i(k+1) - RT_i(k) = G(\rho_i(k) - \rho_i(k-1))$ $\dashrightarrow$ $RT_i(z) = \boxed{\frac{1}{z} G} \cdot \rho_i(z)$
- Computing transfer function $\dashrightarrow$ $H_c(z) = \frac{C(z)H(z)}{1+C(z)H(z)} = \frac{KG}{z-(1-KG)}$

$|1-KG|<1$ $(0<K<2/G)$ for system stabilization



z-Transformed Feedback System

Performance Evaluations

❑ Simulation Environments

- Simulator specification
  - Disksim 2.0 w/ proposed scheme
  - two(2) clients
  - synthetic I/O workloads
  - shared storage spec.
    - IBM_DNES-309170W
    - 7200RPM
- Operational parameters
  - clients' resource weight = 2:1
  - clients' σ (bucket size) = 2:1
  - monitoring period: every 1sec
- Requested perf. requirement

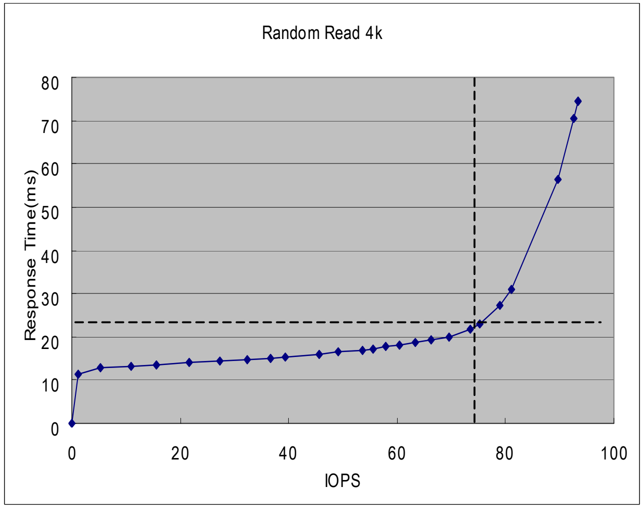
Parameter	Client 1	Client 2
size	4KB	4KB
iops	40	20
rt(msec)	35	38
access pattern	random	random
resource weight	2	1

- Sketch of our evaluations
  - perform simple admission control
  - determine K&G for controller
  - analyze system behavior w/ different pole locations
  - analyze system behavior w/ three types of competing workloads (step/pulse/active)

Performance Evaluations

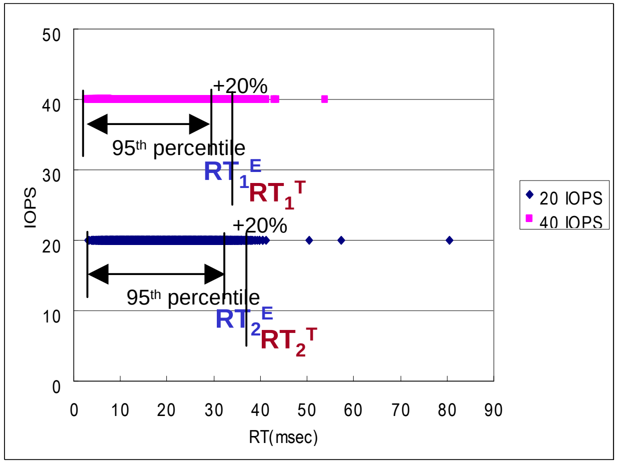
❑ Admission Control

- Underlying storage performance
  - serves "75" 4KB-sized read I/O request per second



- Deliverable response times

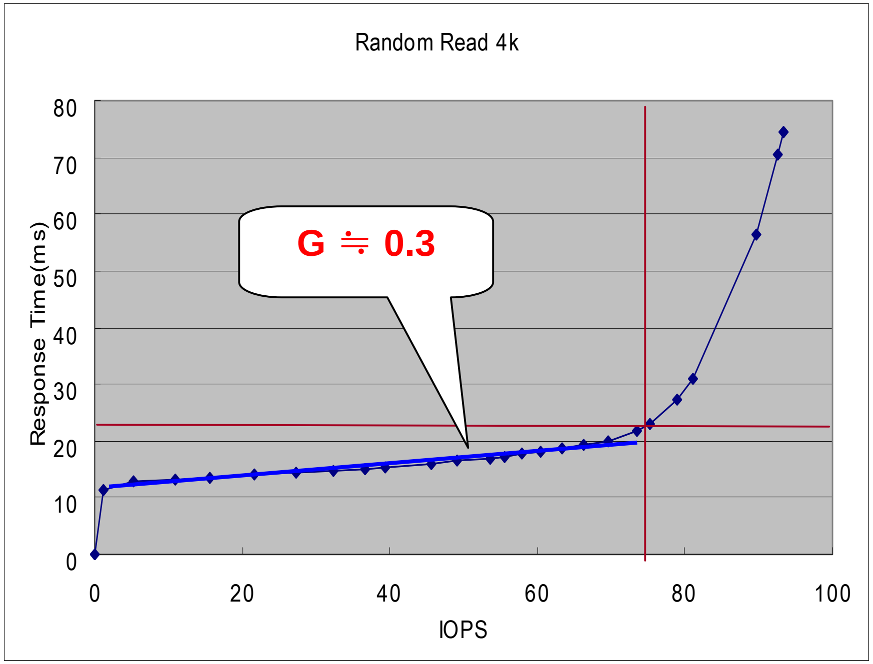
	IOPS^T	RT^E (*msec*)	RT^T (*msec*)	rt (*msec*)
Client 1	40	29.08	**34.1**	**35**
Client 2	20	31.38	**37.65**	**38**



Performance Evaluations

❑ Determination of K, G parameters for Controller

- Obtaining G value
  - from IOPS vs. RT relationship
  - find the slope (sensitivity) in a reasonable area (lower-left box)



- Obtaining K value

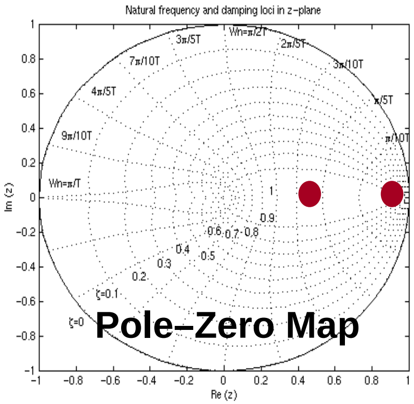
$$H_c(z) = \frac{KG}{z-(1-KG)}$$

0<K< 2/G ⇒ 0<K<6.67

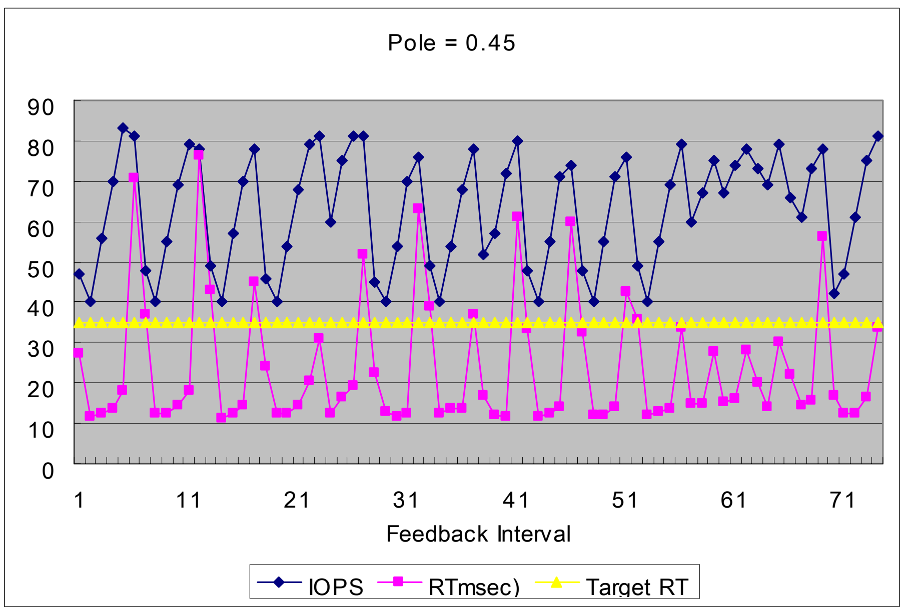
G ≒ 0.3
K = (1-pole) / G

Performance Evaluations

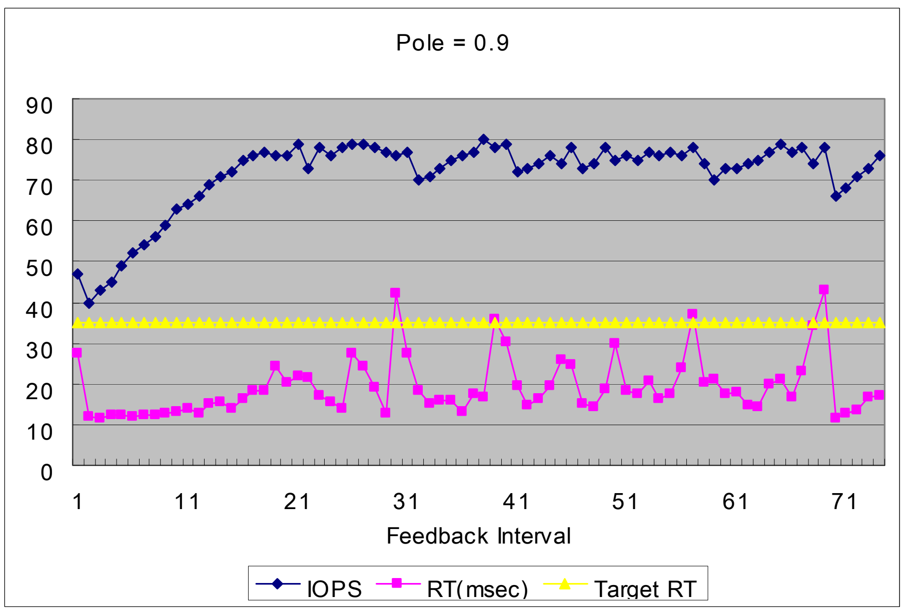
- System Behavior w/ Different Pole Locations
  - Left half of the unit circle (pole-zero map)
    - fast response; overshooting
  - Right half of the unit circle
    - stable; slow response



K=1.83



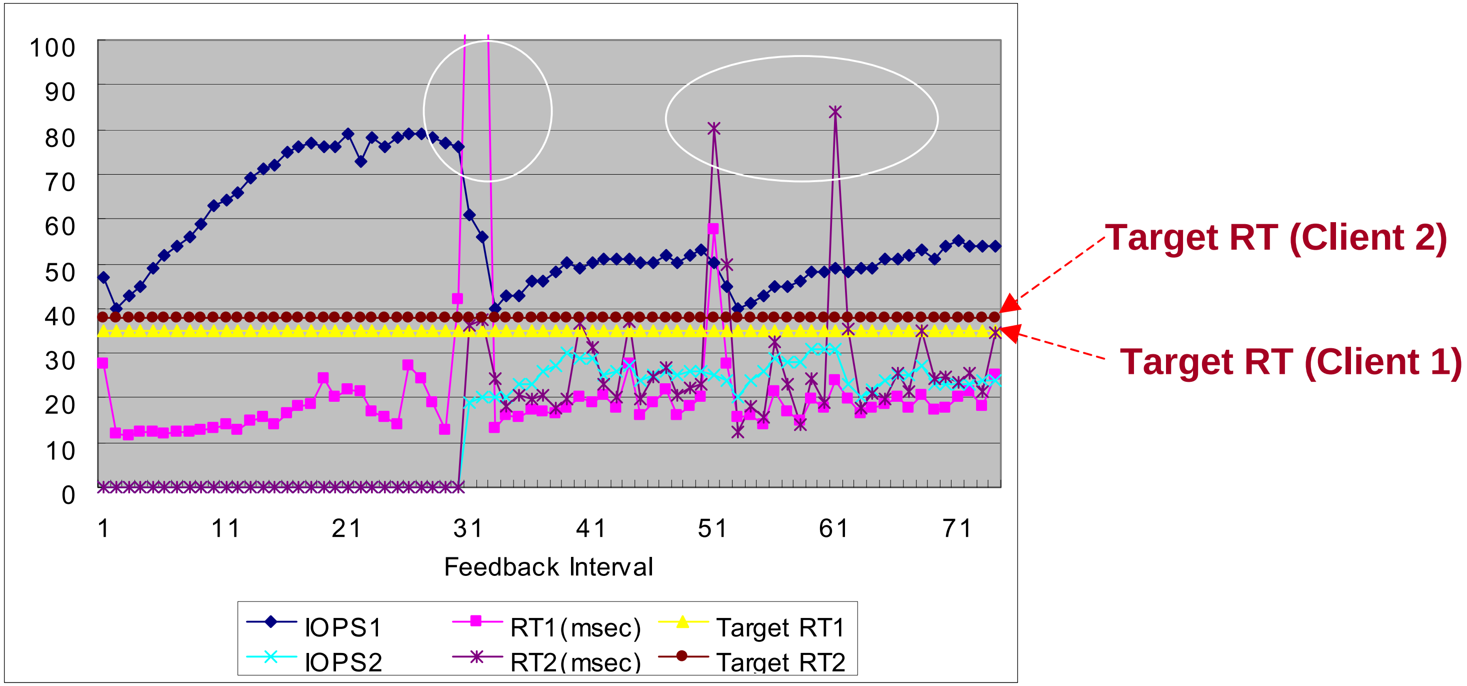
K=0.33



Performance Evaluations

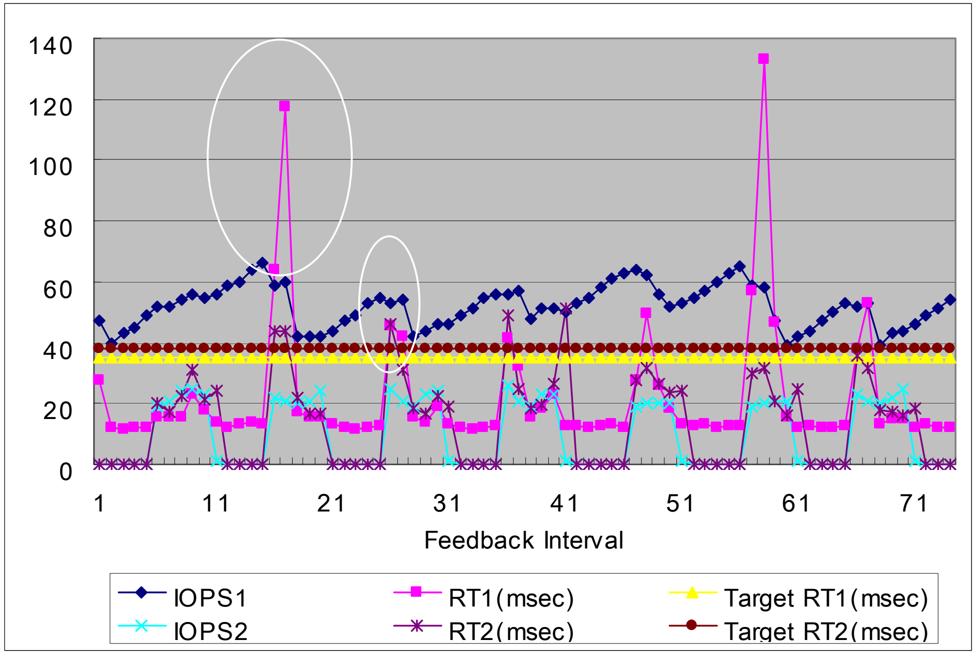
❑ System Behavior w/ Step Workload

- Client 2: I/O workload is issued @ 30 sec
- Client 1: high RT is observed @ 30sec due to the large # of backlogged I/O requests with the use of full B/W
- Target RT violation < 3%



Performance Evaluations

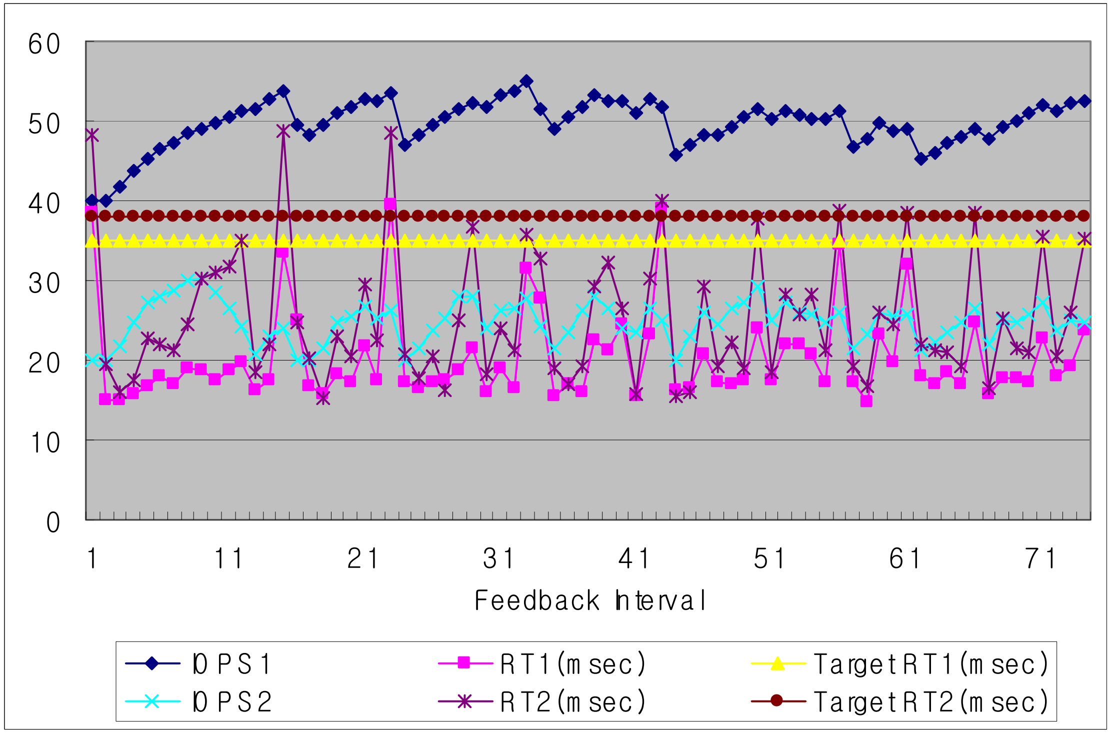
- System Behavior w/ Pulse Workload
  - Client 2: I/O workload is on for 5 sec & off for 5 sec
  - Client 1: spike is observed in RT periodically; disappeared quickly after 2~3 sec
  - Target RT violation < 19(xx)% with higher I/O t-put



Performance Evaluations

❑ System Behavior w/ Two Active Workloads

- Client 1/2: both issue I/O workloads concurrently
- Target RT violation < 3% with higher I/O t-put



Conclusion & Future Work

Conclusion

- We proposed a new I/O performance regulation scheme that
  - comprises LB-based traffic control & fair-queuing algorithm
  - adjusts an LB param(ρ) based on "feedback-controlled" loop by monitoring the current RT

- Simulation results proved that
  - the proposed scheme could efficiently utilize storage resource
  - while assuring the demanded storage services for each clients (esp. target RT)

Future Work

- Testing the proposed scheme with real I/O workloads
- Evaluating different types of feedback controllers (PD, PID)
- Support for assuring more complex storage services (QoS); for example, multiple pairs of target IOPS & RT

Backup Slides

❑ Previous Solutions

- YFQ [Bruno'99]
  - \+ packet-based fair queuing (SFQ+WFQ)
  - \+ t-put guarantee
- Cello framework [Shenoy'98/'02]
  - \+ two-level scheduling, t-put guarantee
  - – *time-interal : adhocacy in the order of visiting class-specific queues*
  - – *accumulated errors of an amount of received service (t-put)*
  - – *hard to integrate this with other resources (CPU, network)*
- Facade [Lumb'03] : EDF with I/O deadline
- SLEDs [Chamb'03] : traffic control w/ leaky bucket

virtual finish tag = virtual start tag + block length/ w_i