Nested QoS: Providing Flexible Performance in Shared IO Environment

Hui Wang
Peter Varman

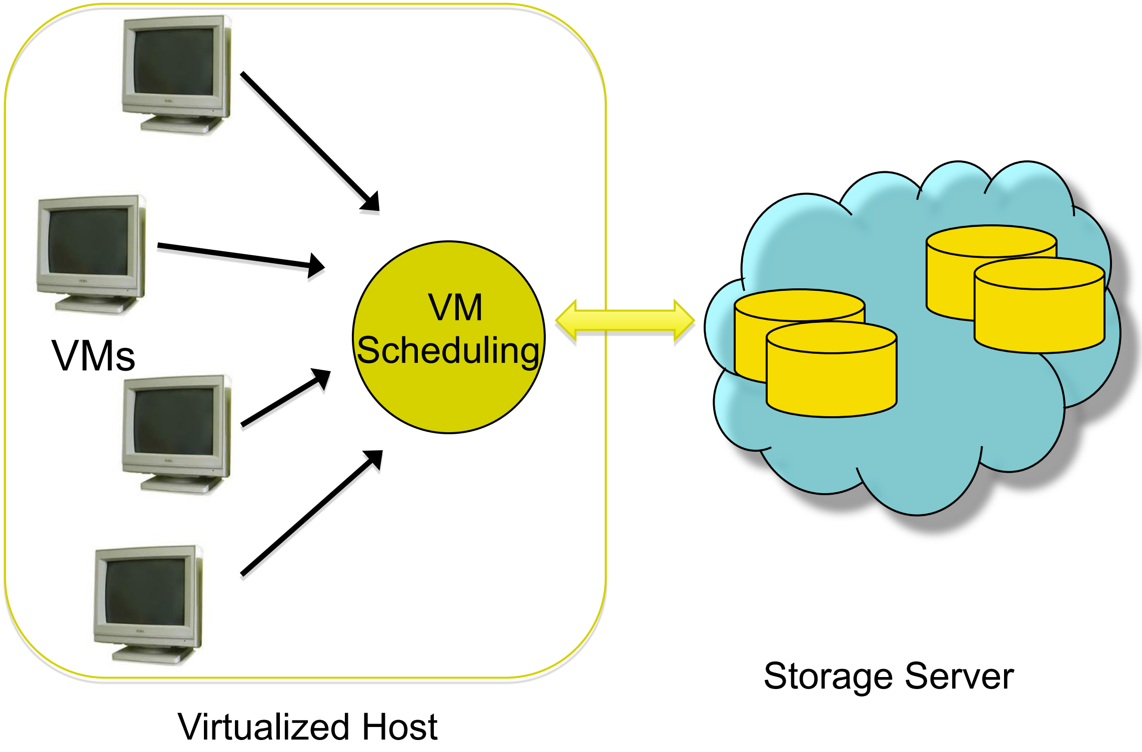
Rice University Houston, TX

Outline

- Introduction
- System model
- Analysis
- Evaluation
- Conclusions and future work

Resource consolidation in data centers

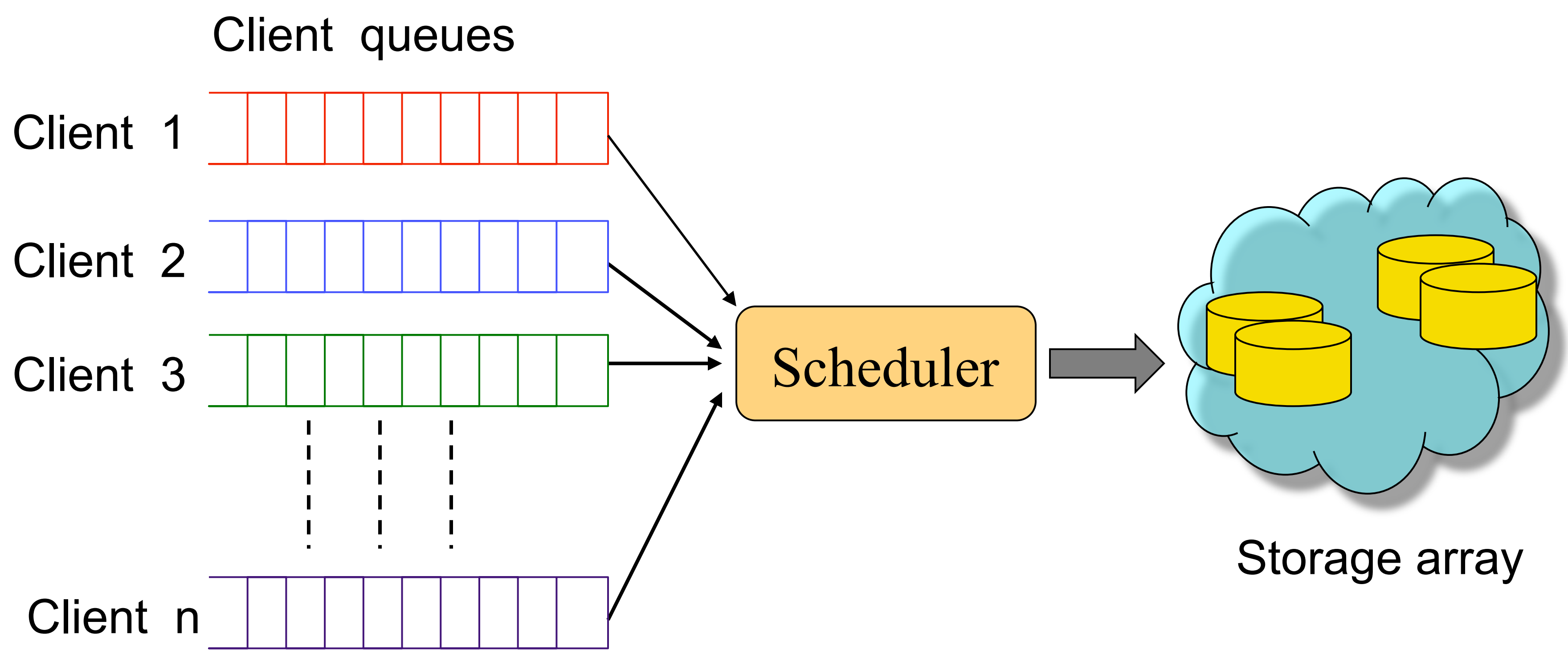
- Centralized storage
  - Economies of scale
  - Easier management
  - High reliability
  - VM-based server consolidation



Issues in resource sharing

- Challenges
  - Performance guarantees
    - QoS models
  - Resource management
  - Capacity provisioning
  - Difficulties:
    - sharing of multiple clients
    - bursty nature of storage workloads

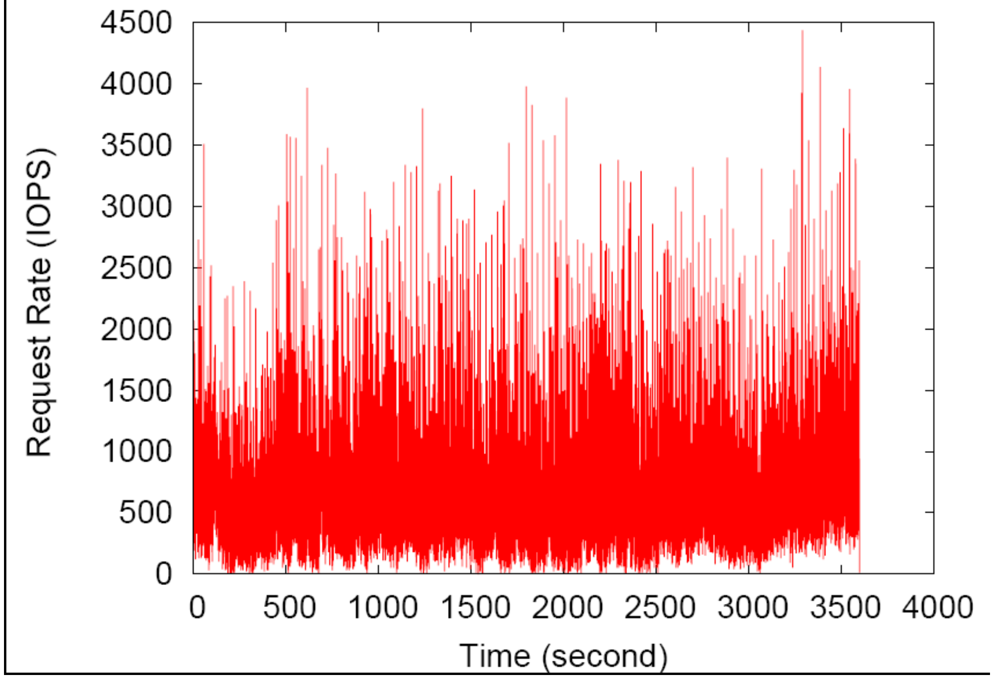
System model for shared I/O



Sharing: The server has to properly allocated resource to concurrent clients to guarantee their performance.

Providing QoS for Bursty Workloads

- Requests have response time QoS
- Storage workloads are bursty
  - Large capacity needed to meet response time during bursts
  - Low average server utilization
- Providing QoS for bursty workloads which have response time QoS requirement



Eg. Open Mail trace, with 100ms window size

- Average rate:~700 IOPS
- Peak rate: 4500 IOPS

Related Work

- Proportional Resource Sharing
  - Algorithms:
    - Fair Queuing, WFQ, WF2Q, Start Time Fair Queuing , Self-Clocking
  - Allocate active clients bandwidth (IOPS) in proportion to their weight w_i
  - Limitations:
    - Response time is not independently controlled
      - Low throughput transactions requiring short response time
      - High throughput file transfer insensitive to response time
    - No provisioning for bursts

Related work (cont'd)

- Providing response time guarantees
  - Algorithms:
    - SCED, *p*Clock
  - Client traffic must be within a specified traffic envelope then client requests are guaranteed a maximum response time of δi
  - Limitations:
    - No isolation of non-compliant part of workload
      - Loss of QoS guarantee over extended (unbounded) portions
    - Only a single response time guarantee is supported
      - Lack of flexibility & high capacity requirement

Performance QoS

- QoS often specified as a percentage of workload meeting the response time bound
- Absolute percentage guarantees are hard to support
  - Can provide response time guarantees if entire workload is bounded by a traffic envelope
    - Requires high capacity
  - Guarantee any fixed percentage (say 90%) of the workload
    - Unrestricted traffic above the traffic envelope can decrease the guaranteed percentage arbitrarily

Nested QoS

- We propose:
  - Multiple traffic envelops (classes) to describe one bursty workload
  - Performance guarantees based on portion of traffic that satisfies traffic envelope (not percentage)
  - Different performance guarantees for different classes

Outline

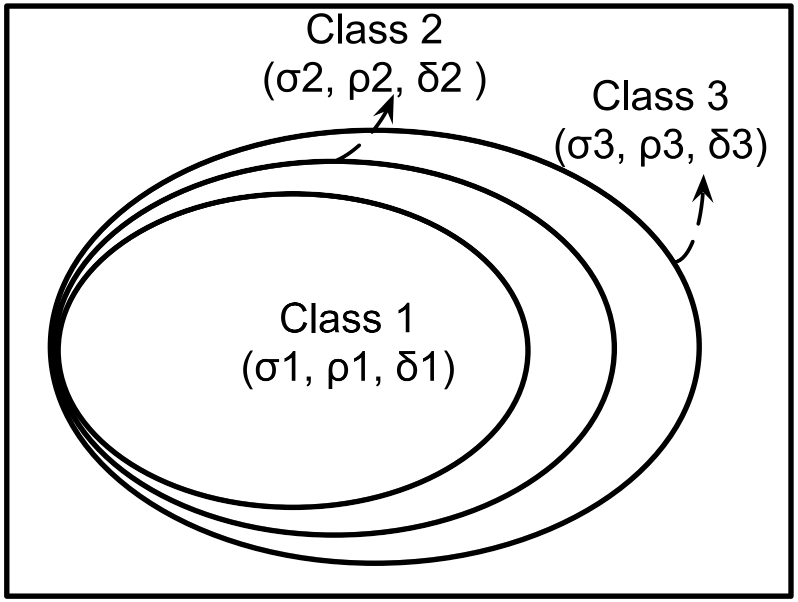
- Introduction
- System Model
- Analysis
- Evaluation
- Conclusions and future work

Traffic envelopes

- Abstract model

- Each class i has
  - Traffic envelope (Token bucket) (σi, ρi)
  - Response time δi

- Eg: 3-class Nested QoS model
  - (30, 120 IOPS, 500ms)
  - (20, 110 IOPS, 50ms)
  - (10, 100 IOPS, 5ms)

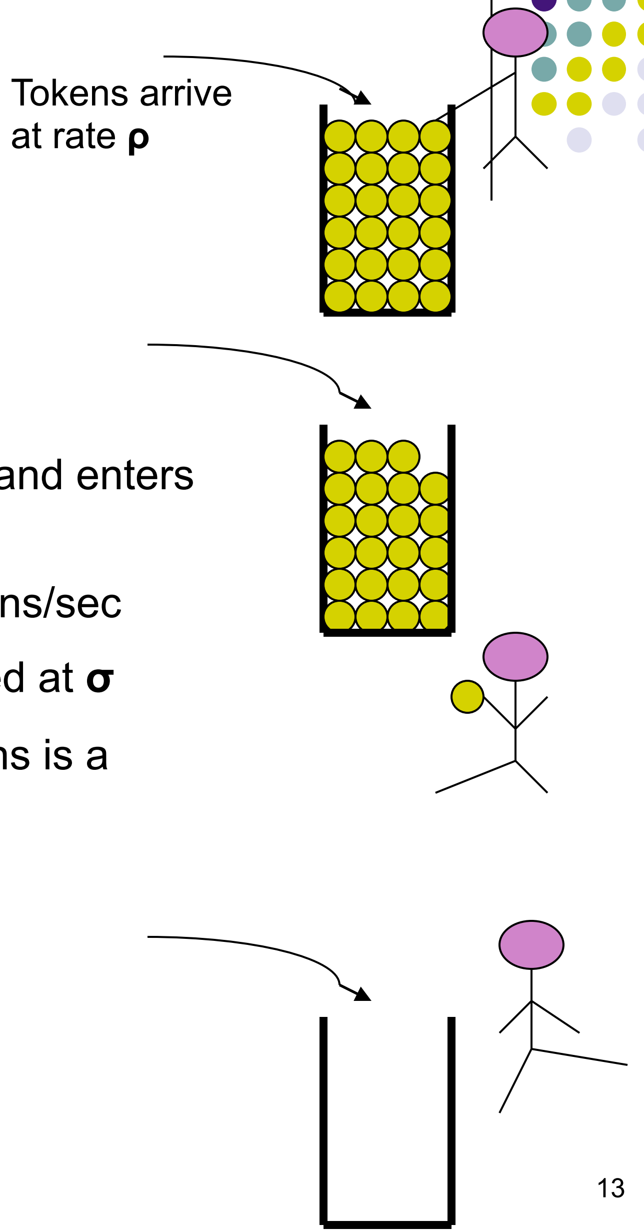


Token Bucket Regulation

- Traffic Envelope

  Arrival Curve Limit

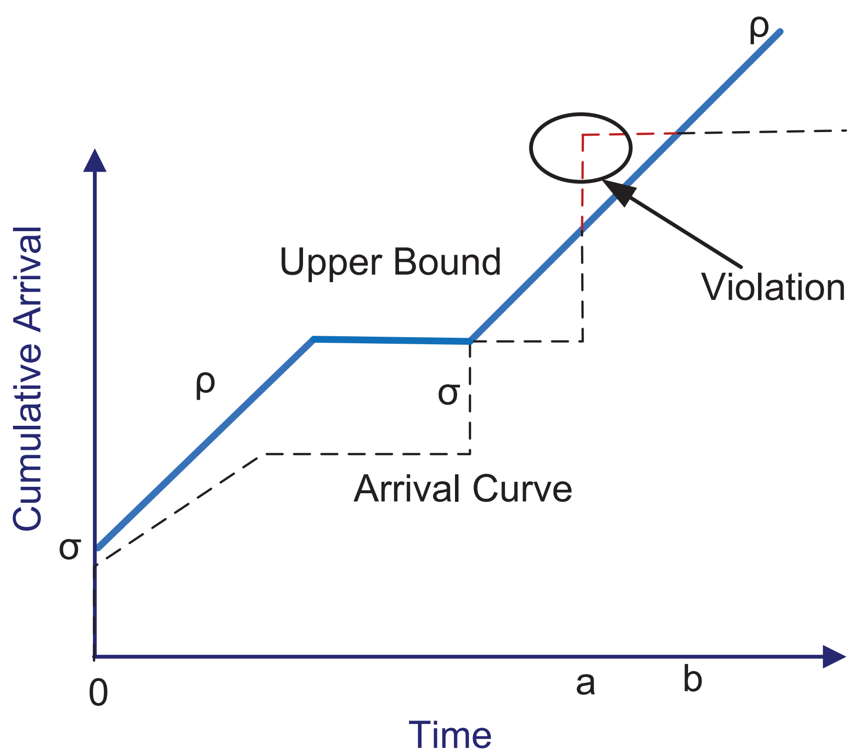
  - (**σ**, **ρ**) Token Bucket Model



- Bucket of capacity is **σ** tokens;
- Arriving request takes a token from the bucket and enters system
- Tokens replenished at a constant rate of **ρ** tokens/sec
- Maximum number of tokens in bucket is capped at **σ**
- A request that arrives when there are no tokens is a violation of traffic envelope (constraints)

- Service Level Agreement (SLA):
  - Client traffic limited by the Traffic Envelope
  - Response time is guaranteed on requests

Bounding the arrival curve with traffic envelope (token bucket)



Token-bucket regulator:

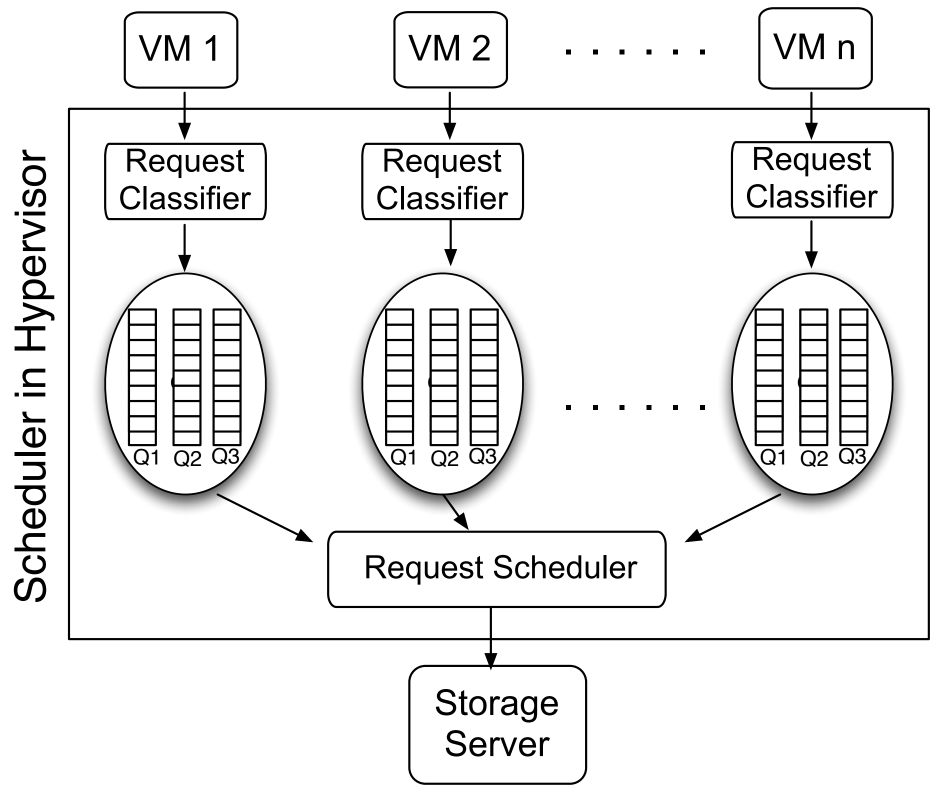
ρ: token-generation rate
σ: maximum tokens / instantaneous burst size

Maximum # requests arriving in any time interval t: ≤ **σ + ρ*t**

If the arrival curve lies below the Upper Bound then all requests will meet their deadlines

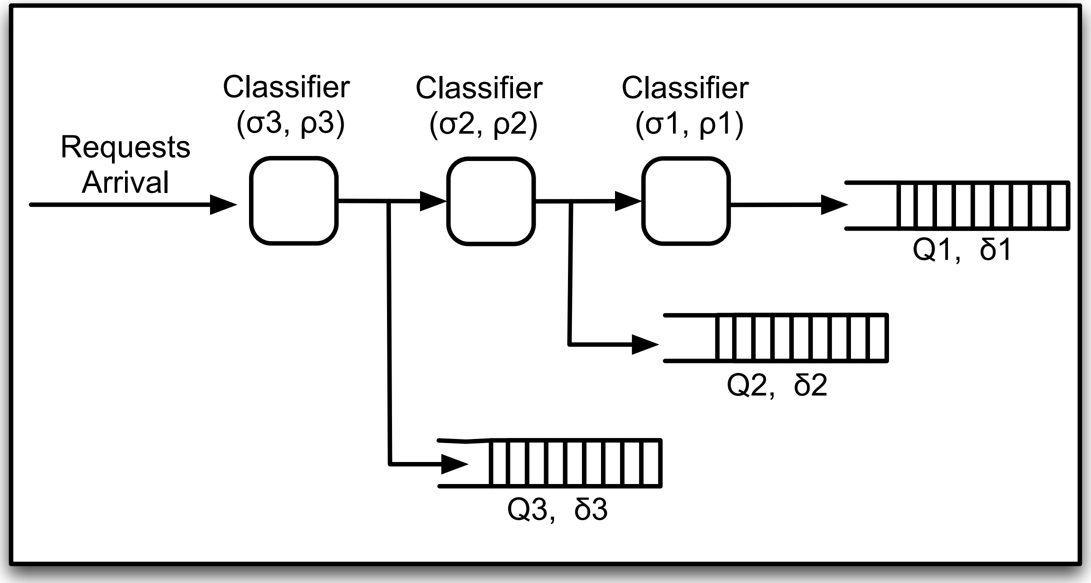
Architecture in VM environment

- Request Classification
  - Multiple token buckets
- Request Scheduling
  - Two levels: EDF within VM queues and FQ across VMs
  - Alternative: 1-level EDF
    - Pros: Capacity & Simplicity
    - Cons: Low robustness to capacity variation



Request Classification

- Queues
- Token Buckets



Outline

- Introduction
- System Model
- Analysis
- Evaluation
- Conclusions and future work

Analysis

Lemma 1 The capacity required for all requests to meet their deadlines in the Nested QoS model, when all ρ_i are equal to ρ, is given by: $max_{1 \leq j \leq n}\{\sigma_j/\delta_j + \rho(1 - \delta_1/\delta_j), \rho\}$.

Lemma 2: Let $\alpha = \delta_{i+1}/\delta_i$, $\beta = \sigma_{i+1}/\sigma_i$ and $\lambda = \beta/\alpha$ be constants. The server capacity required to meet SLOs is no more than: $max_{1 \leq j \leq n}\ \{\rho,\ \lambda^j(\sigma_1/\delta_1) + \rho(1 - 1/\lambda^j)\}$. For $\lambda < 1$, the server capacity is bounded by $\sigma_1/\delta_1 + \rho$, which is less than twice the capacity required for servicing C_1.

- Proof see paper.

Outline

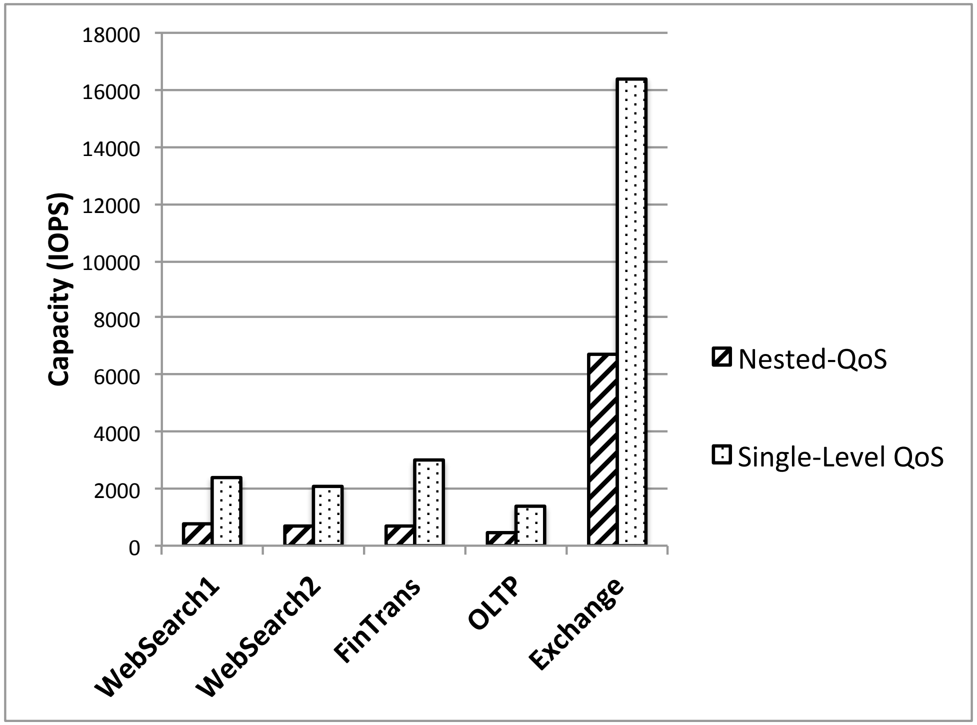
- Introduction
- System Model
- Analysis
- Evaluation
- Conclusions and future work

Evaluation

- Determine the parameters empirically
  - *Number of classes* & *traffic envelope*
  - Tradeoff between capacity required (cost) and performance.
- Workloads
  - Block-level workloads from trace repository

Nested QoS for a single workload

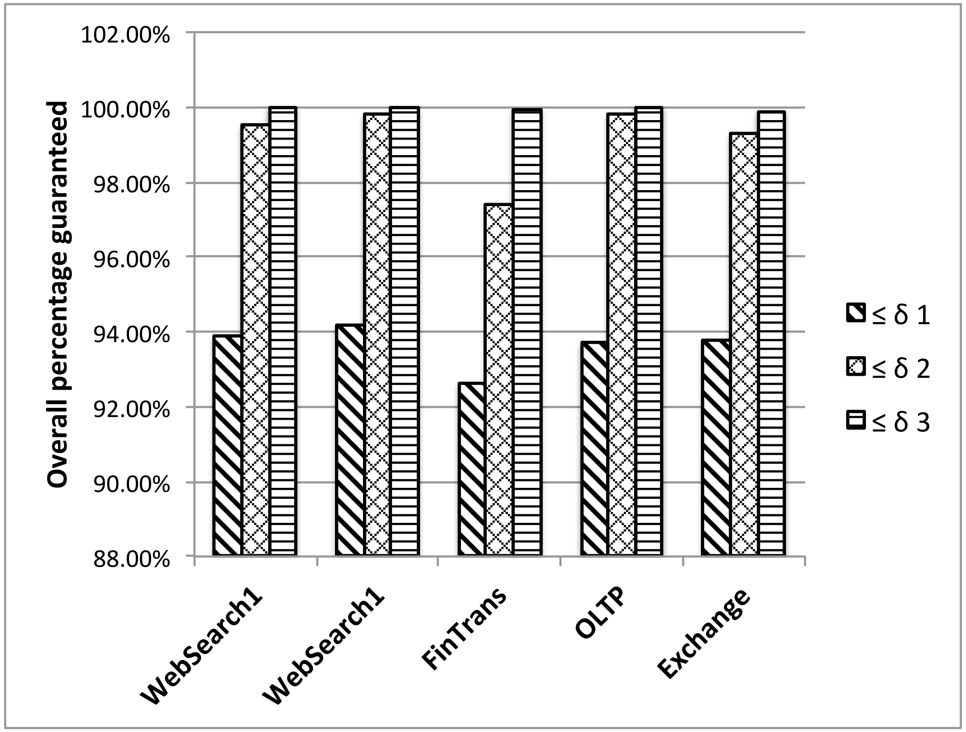
- Workloads
  - WebSearch1: (3, 650IOPS, 5ms)
  - WebSearch2: (3, 650IOPS, 5ms)
  - FinTrans: (4, 400 IOPS, 5ms)
  - OLTP: (3, 650IOPS, 5ms)
  - Exchange: (33, 6600IOPS, 5ms)
- Goal
  - 90% requests in class 1 (5ms)
  - 95% requests in class 2 (50ms)
  - 100% requests in class 3 (500ms)
- Singe level QoS
  - 100% requests in 5 ms



Capacity Requirement

Nested Nested QoS for a single workload

- Goal
  - 90% requests in class 1 (5ms)
  - 95% requests in class 2 (50ms)
  - 100% requests in class 3 (500ms)
- Singe level QoS
  - 100% requests in 5 ms

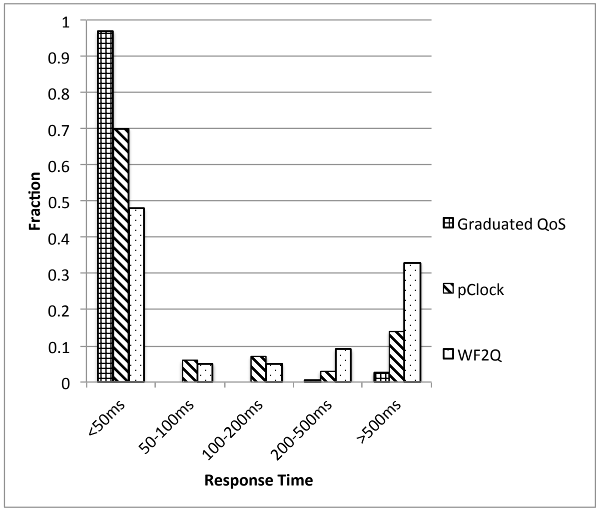


Performance for Nested QoS

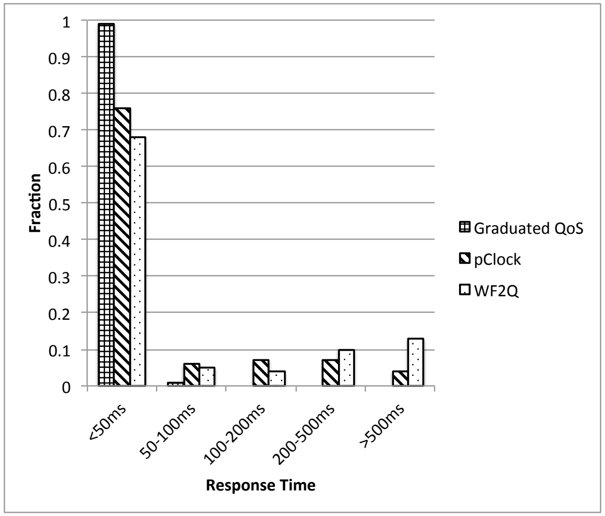
Nested QoS for Concurrent Workloads

- Two workloads
  - W1: Web Search; ~350 IOPS
  - W2: Financial Transaction; ~170 IOPS
  - Total capacity 528 IOPS

- Response times:
  - 50ms for class 1; 500ms for class 2 and 5000ms for class 3



WebSearch performance

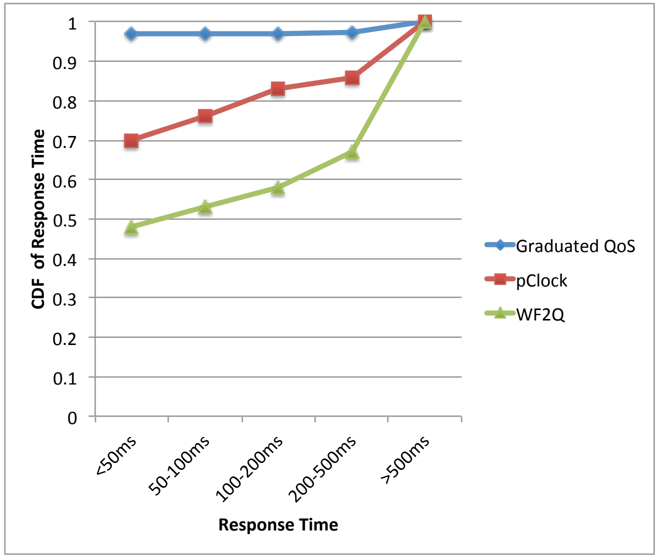


FinTrans performance

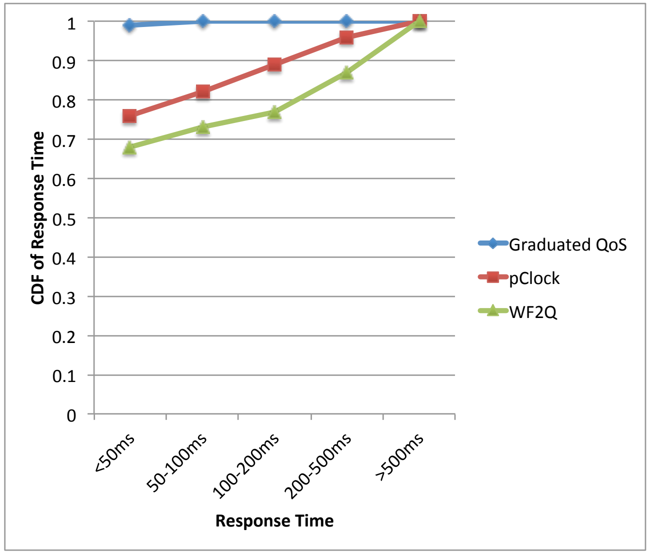
Nested QoS for Concurrent Workloads

- Two workloads
  - W1: Web Search; ~350 IOPS
  - W2: Financial Transaction; ~170 IOPS
  - Total capacity 528 IOPS

- Response times:
  - 50ms for class 1; 500ms for class 2 and 5000ms for class 3



WebSearch: CDF of Response time



FinTrans: CDF of Response time

Outline

- Introduction
- System Model
- Analysis
- Evaluation
- Conclusions and future work

Conclusions and future work

- Conclusions
  - Large reduction in server capacity without significant performance loss
  - Analytical estimation of the server capacity
  - Providing flexible SLOs to clients with different performance/cost tradeoffs
  - Providing a conceptual structure of SLOs in workload decomposition

- Future work
  - Workload characteristics for nested model parameters