YouChoose: A Performance Interface Enabling Convenient and Efficient QoS Support for Consolidated Storage Systems

Xuechen Zhang, Yuehai Xu, and **Song Jiang**

Department of Electrical and Computer Engineering
Wayne State University

Data-intensive Applications using Consolidated of Storage Systems

- Applications become more data intensive
  - Scientific applications may analyze large data sets.
  - Internet search and E-commerce rely on efficient data access.
  - Applications' performance highly depends on I/O service quality.
- Advantages of consolidated storage system
  - High utilization due to resource sharing.
  - Cost-effectiveness of centralized management.
  - Lower operating cost.
- Each user essentially reserves a virtual storage device.
  - Contractual quality of services (QoS) requirements (SLA).
  - How to specify the I/O QoS requirements?

An Example Issue: Amazon Elastic Compute Cloud (Amazon EC2)

Available Instance Ty

Standard Instances

Instances of this family are well suit … t applications.

Small Instance (default)*

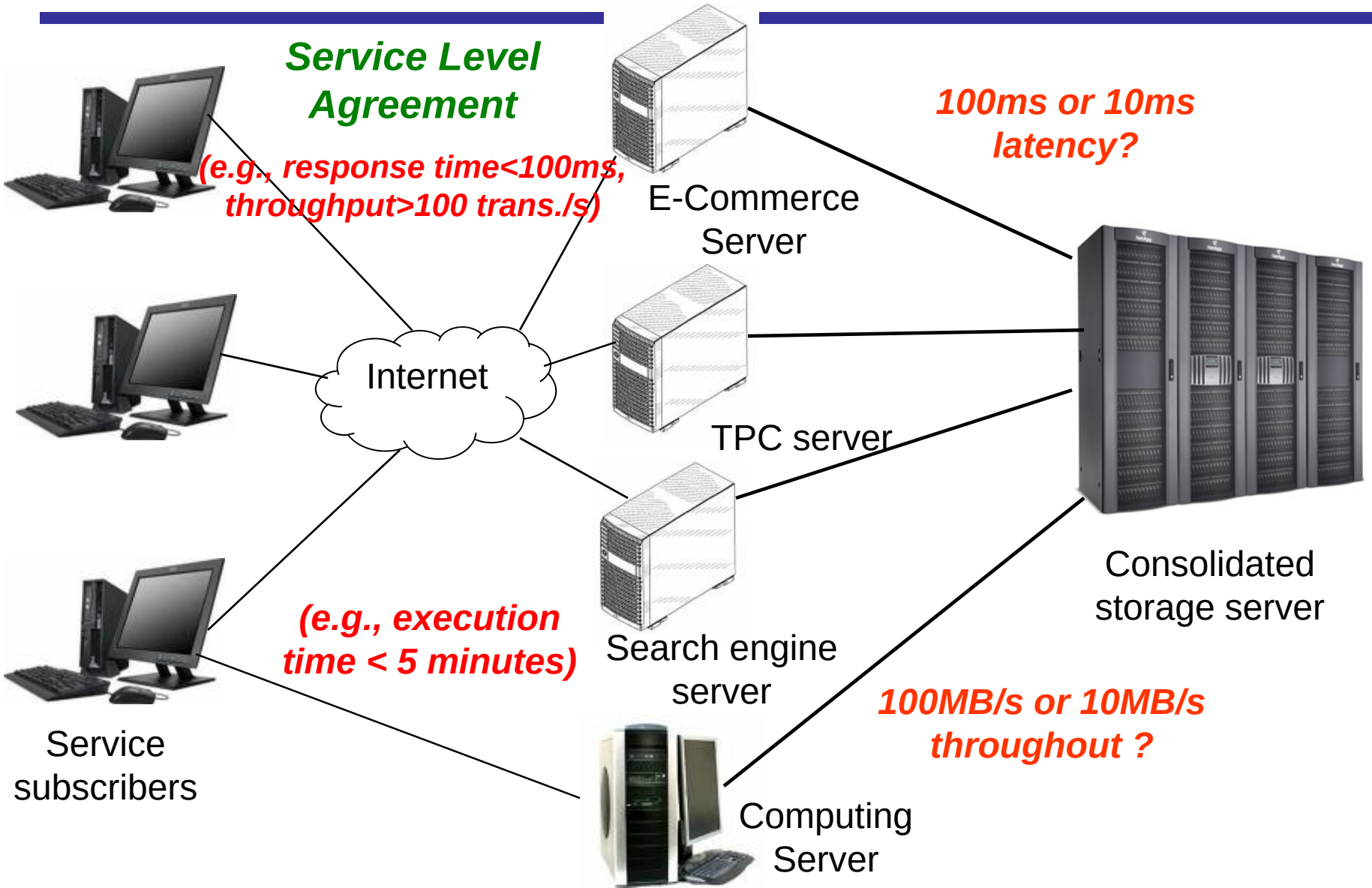
1.7 GB memory
1 EC2 Compute Unit (1 virtual core with
160 GB instance storage (150 GB plus 1(
32-bit platform
I/O Performance: Moderate
Price: $0.10 per instance hour

Large Instance

7.5 GB memory
4 EC2 Compute Units (2 virtual cores wit
850 GB instance storage (2×420 GB plus
64-bit platform
I/O Performance: High
Price: $0.40 per instance hour

Equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

System Structure



Issues with the Use of Fixed I/O Bounds

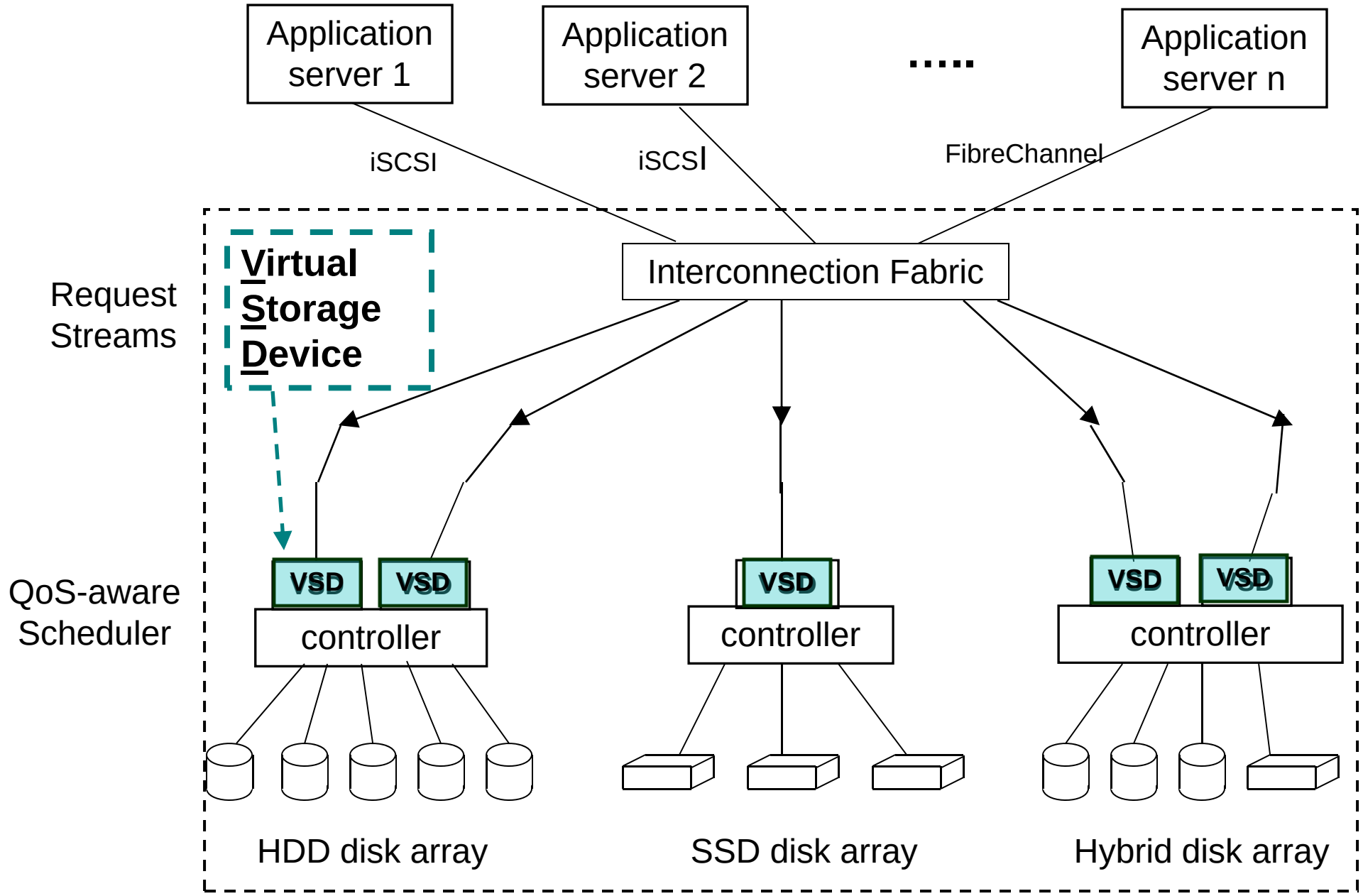
- I/O intensity can change from time to time.
  - Requests in the burst period share the same latency bound with those in the quiet period?
  - If the bound is determined according to requests in the quiet period, how much resources are demanded to meet it during the busy period?
- Request size can be highly variable.
  - One common latency bound for small and large requests?
  - If the throughput is in form of MB/s, any incentive to aggregate small requests into one large one?
- Spatial locality of requests can vary substantially.
  - One common throughput bound for random and sequential requests?
  - Shall the bound be determined according to random requests or sequential ones?

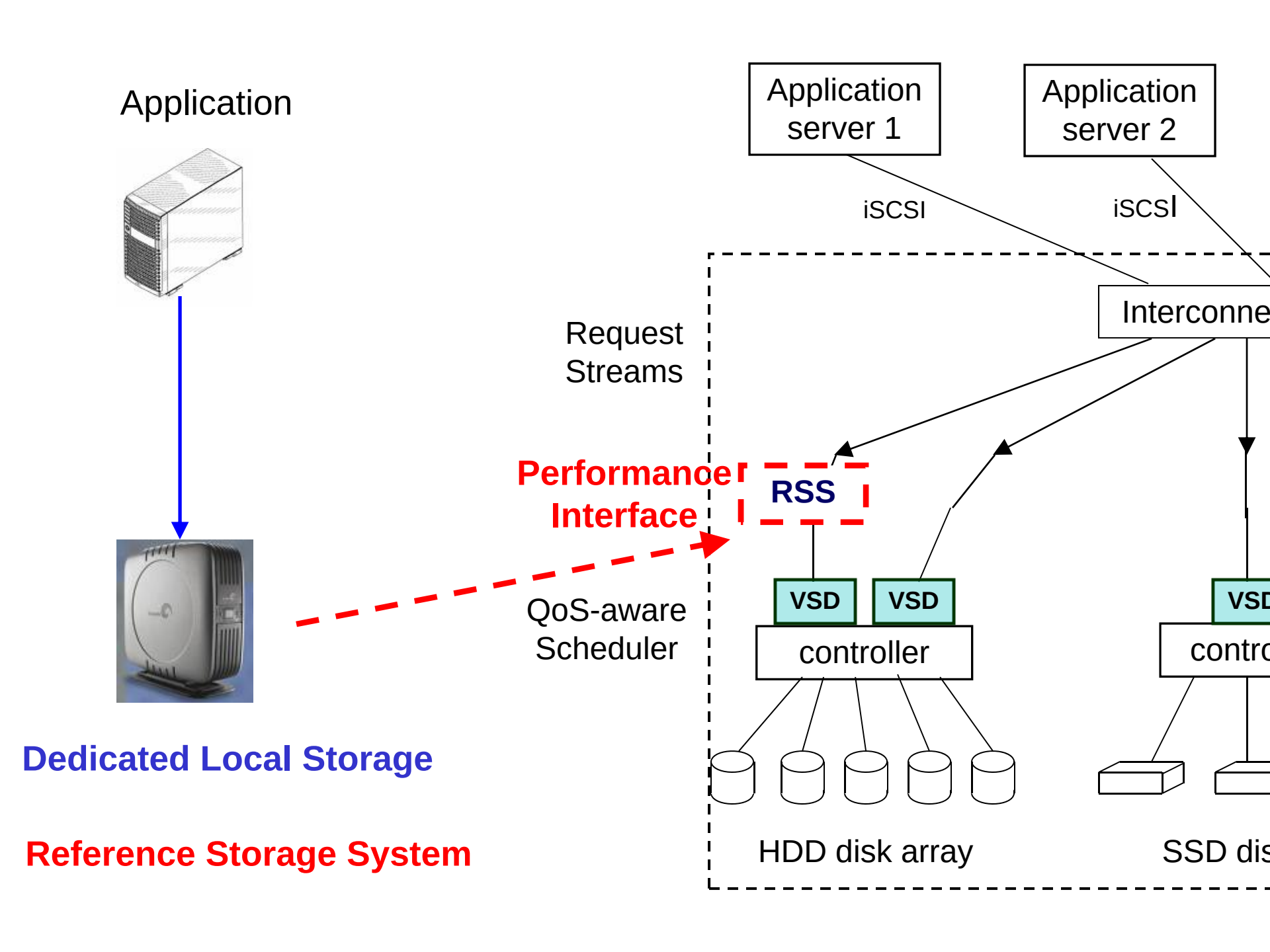
Implications of Fixed I/O Bounds

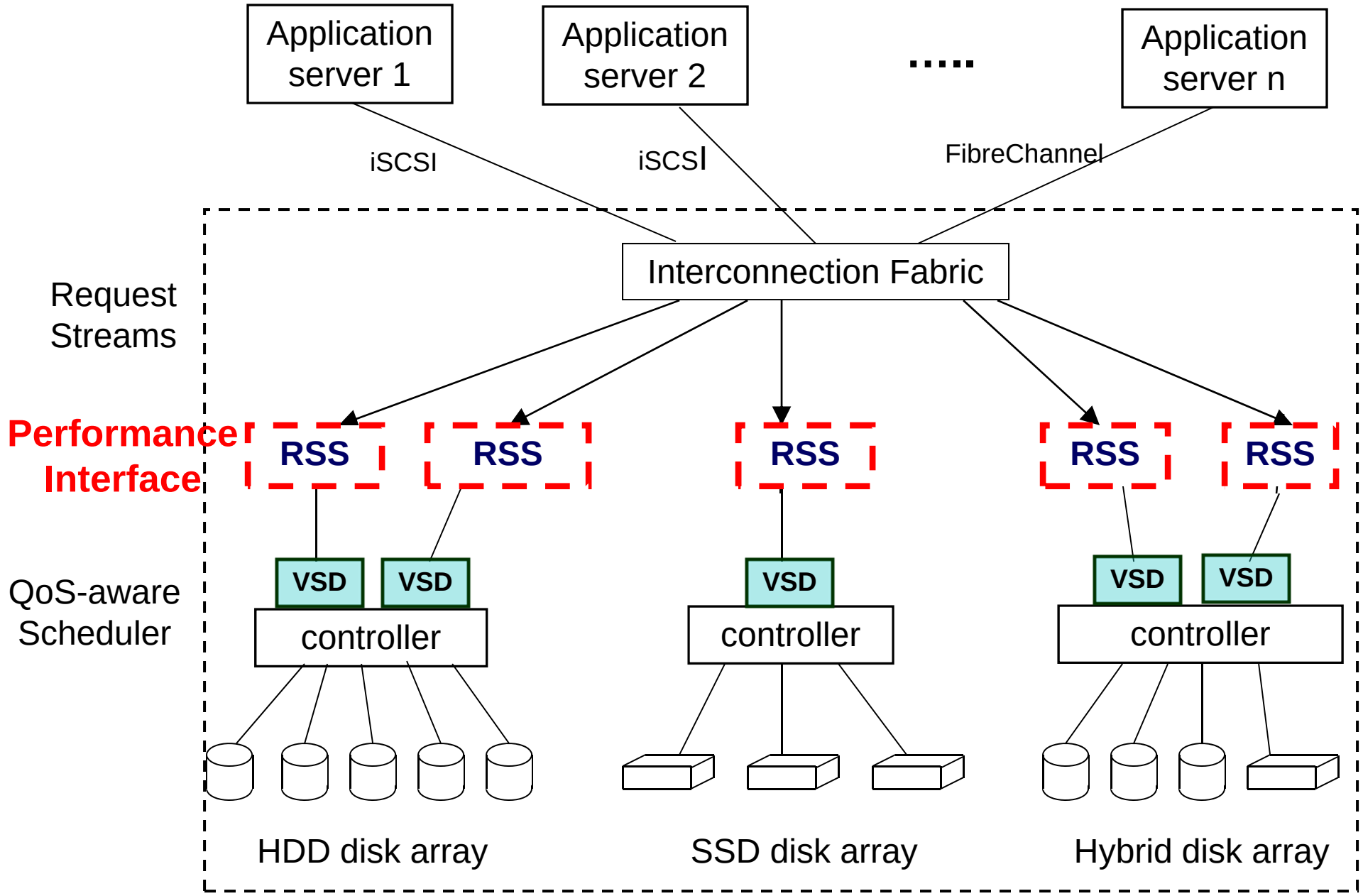
- They may not reflect applications' real QoS needs.
- They may discourage programmers' efforts on the optimization of I/O requests.
- They can pose highly variable resource demands on the storage system.

Our Solution: Use Reference Storage System as Performance Interface

- Assume that a user can receive satisfactory application performance with use of a dedicated storage system.
  - He wants to keep the performance after outsourcing I/O service to a shared storage system.
- The dedicated storage system is used as its performance interface.
  - The interface is called Reference Storage System (RSS)
  - By implementing the interface, the user will receive performance at least as good as that received on the RSS.
- The RSS interface is **not** subject to variation of I/O behaviors.
  - The interface is tangible to end users and is more relevant to application performance.
  - The interface can easily bound the resource demand on the shared storage system.


Application server 1
Application server 2
.....
Application server n
iSCSI
iSCSI
FibreChannel
Request Streams
Virtual Storage Device
Interconnection Fabric
QoS-aware Scheduler
VSD
VSD
controller
VSD
controller
VSD
VSD
controller
HDD disk array
SSD disk array
Hybrid disk array


Application
Application server 1
Application server 2
iSCSI
iSCSI
Interconne
Request Streams
Performance Interface
RSS
VSD
VSD
VSD
controller
contro
QoS-aware Scheduler
Dedicated Local Storage
Reference Storage System
HDD disk array
SSD dis

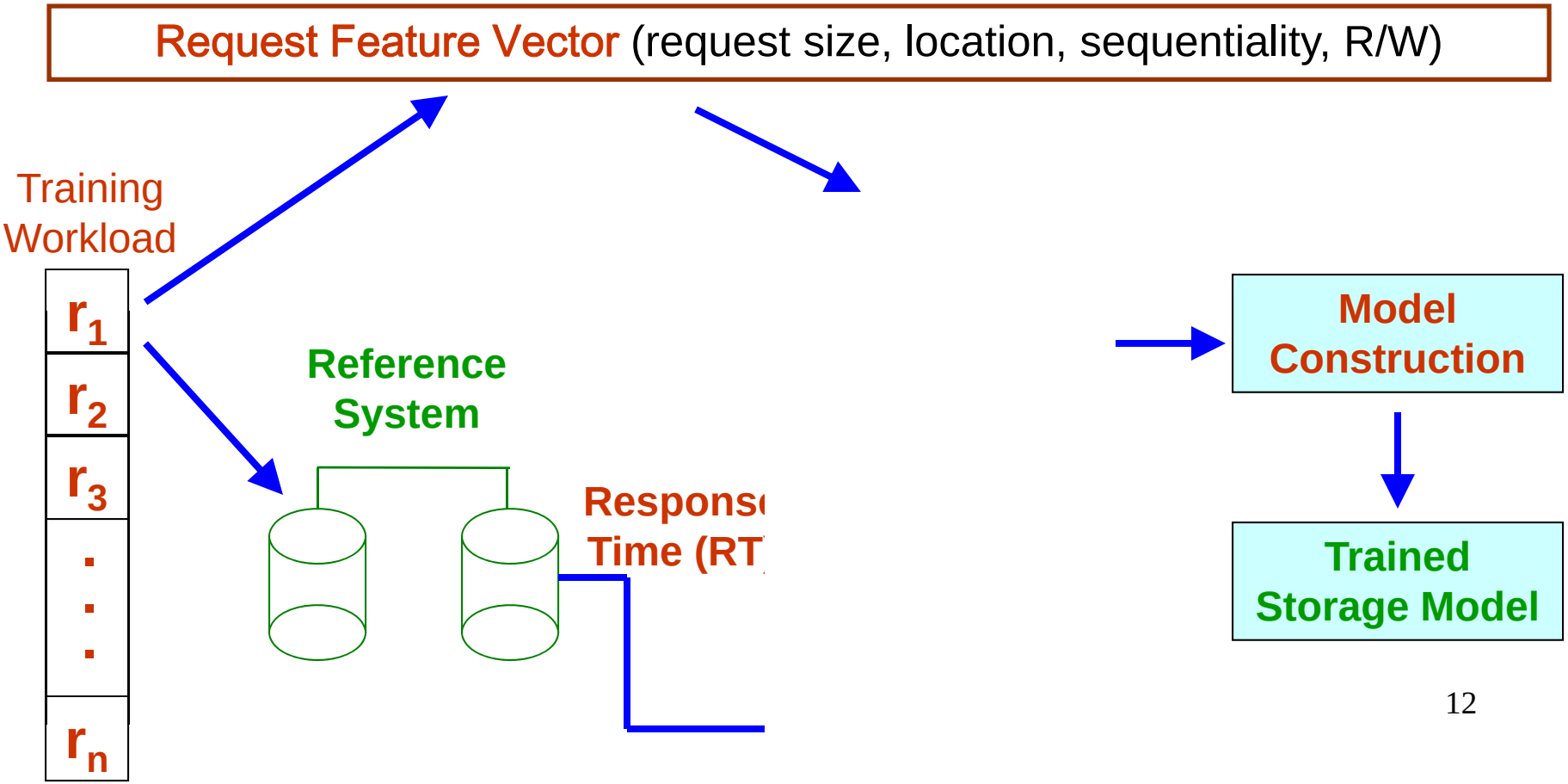

Application server 1
Application server 2
.....
Application server n
iSCSI
iSCSI
FibreChannel
Interconnection Fabric
Request Streams
Performance Interface
RSS
RSS
RSS
RSS
RSS
QoS-aware Scheduler
VSD
VSD
VSD
VSD
VSD
controller
controller
controller
HDD disk array
SSD disk array
Hybrid disk array

YouChoose: Implementation and challenges

- Interpret RSS for the I/O scheduler to implement the interface
  - Predict what the latency of an arriving request is if it was received by RSS.
  - It's a challenge with different access patterns and system configurations.
- Efficiently implement the RSS interface.
  - Meet simultaneously RSS requirements for different VSDs
  - Able to exploit request locality for system efficiency.
- Migrate virtual storage devices (VSDs) for high device utilization.
  - Different disk arrays exhibit various efficiency in hosting VSDs.
  - Automatically place and migrate VSDs to host arrays for high efficiency.

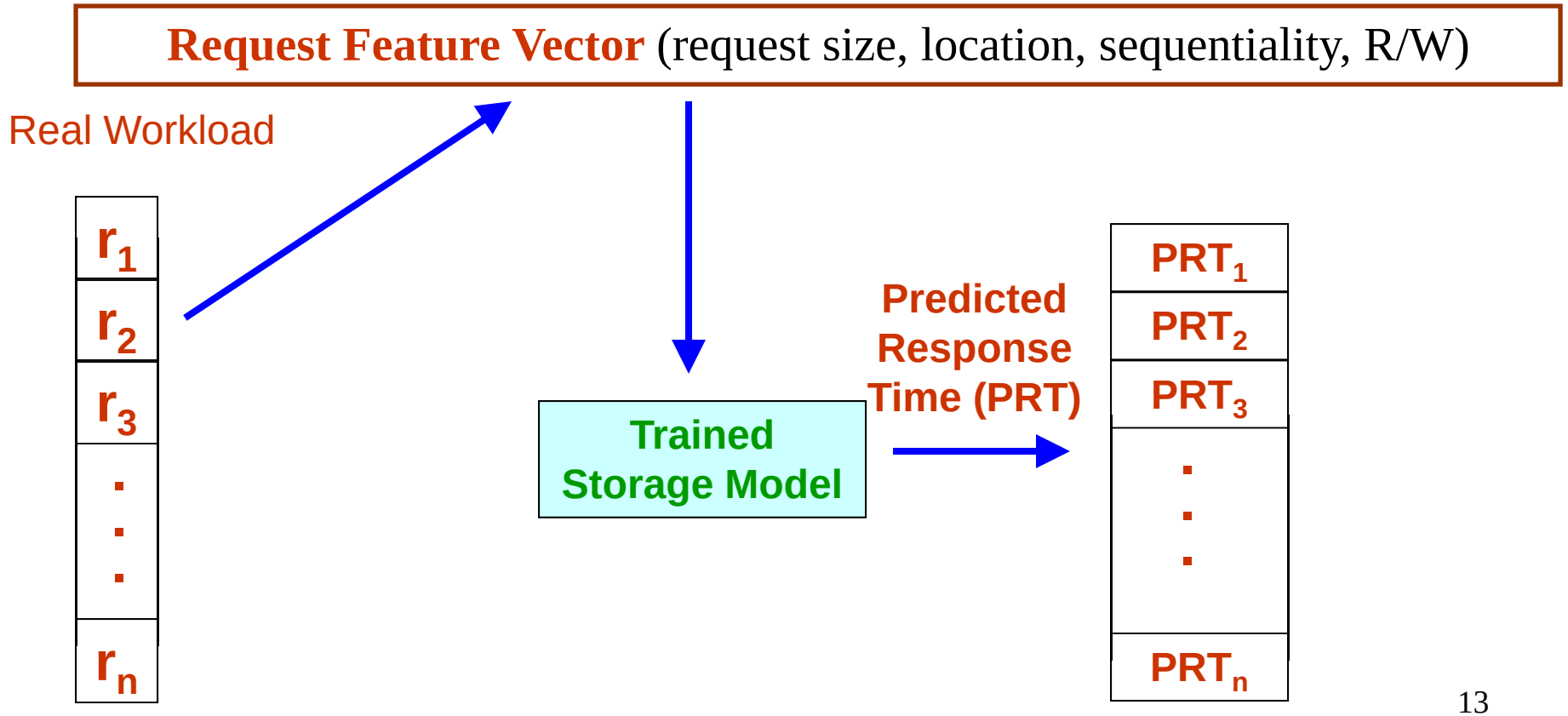
Prediction with the CART Tool

- The CART (Classification And Regression Trees) Tool
  - Known for its efficiency and accuracy.
- Model Training



Prediction with the CART Tool (cont'd)

- Use the Model

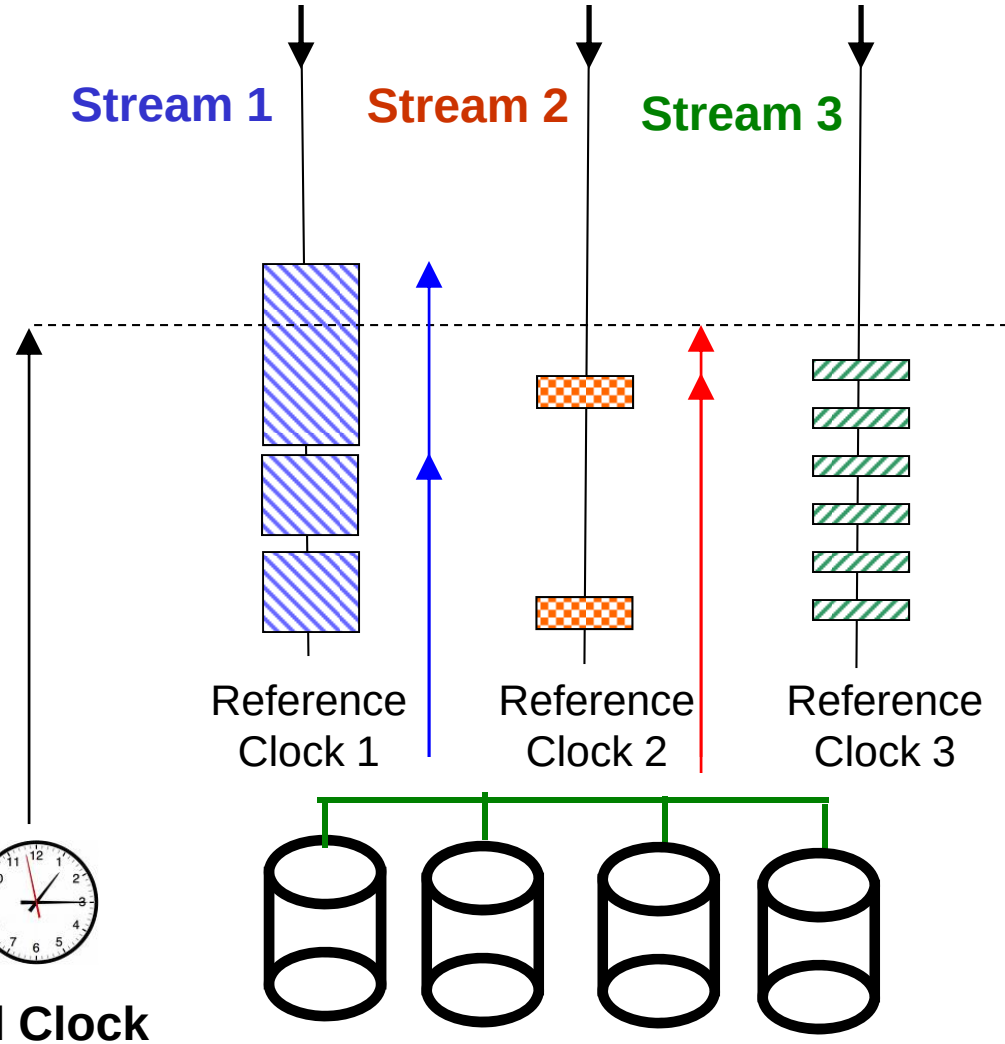


YouChoose Request Scheduling

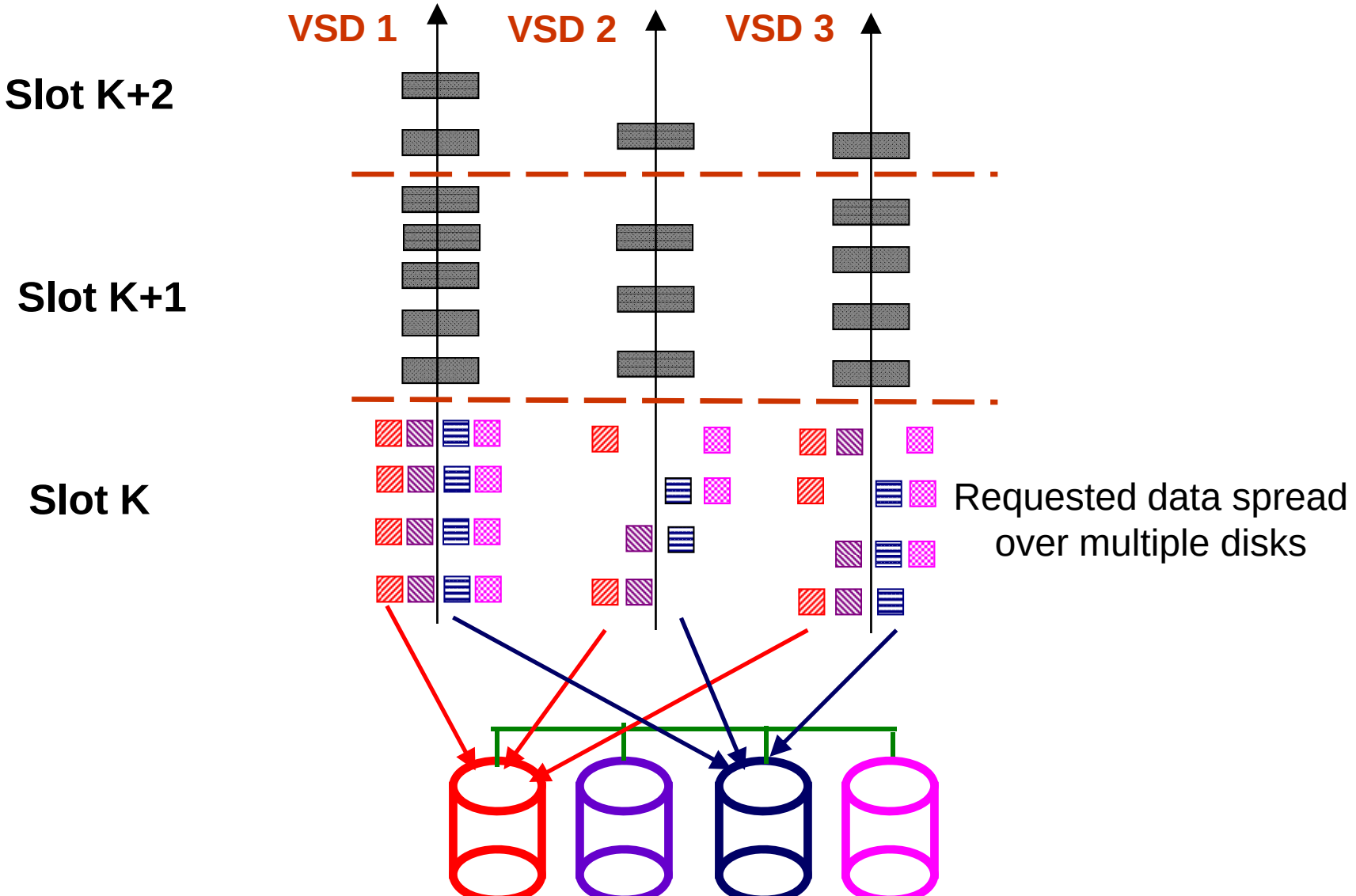
- We can predict a request's service time on RSS (ref_time)
- N+1 clocks:
  - One wall clock (wall_clock)
  - N reference clocks (ref_clock).
- When the stream is considered for scheduling:
  - If its request is dispatched, then

    $ref\_clock \mathrel{+}= ref\_time_{req}$
  - No pending requests, then

    ref_clock = wall_clock.



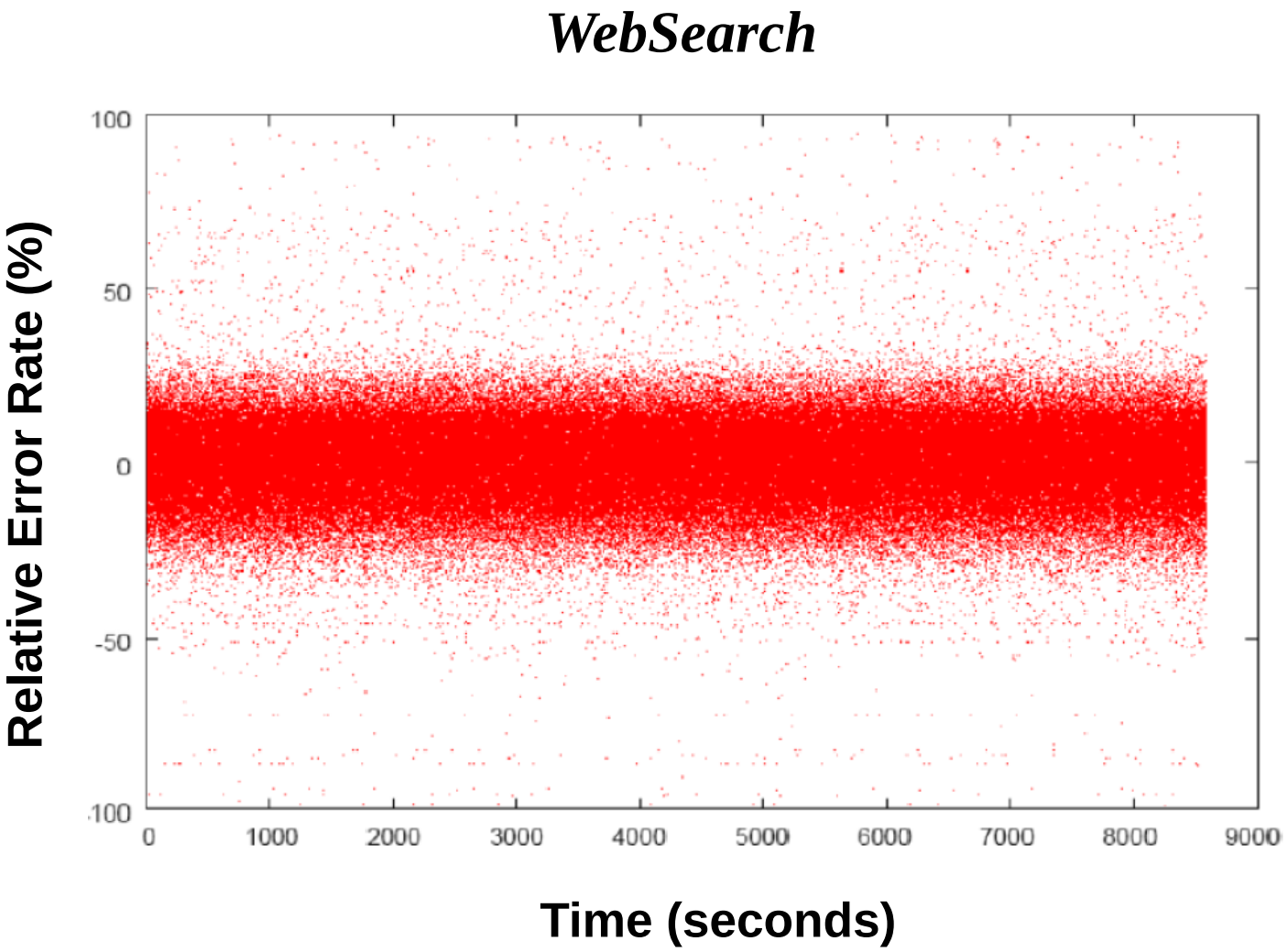
Serving Requests in Batches for Efficiency



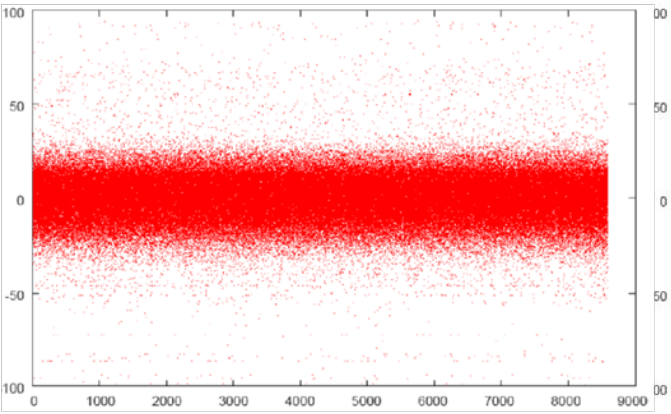
Performance Evaluation

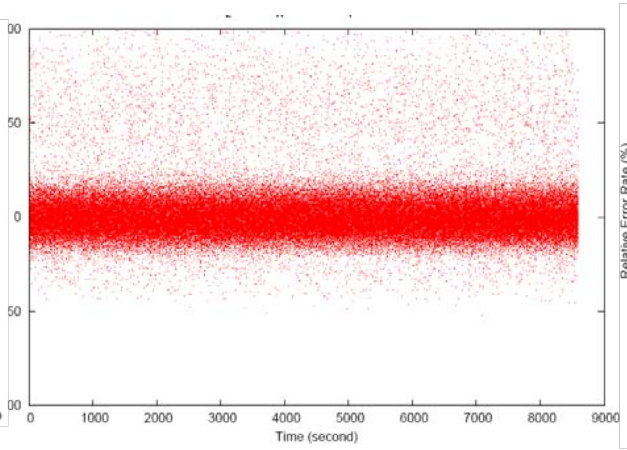
- Disk arrays simulated by DiskSim
  - Fast disks: QUANTUM TORNADO (10025RPMs, 1.245ms)
  - Slow disks: SEAGATE ST32171W (7200RPMs, 1.943ms)
- Synthetic traces
  - Request size: 4KB
  - Spatial locality x% ∈ [0%-100%]: the probability of two consecutive requests for contiguous data.
- Real-world I/O traces
  - Financial : traces from OLTP applications at two large financial institutions.
  - WebSearch: traces from a popular search engine.
  - OpenMail: collected on a production e-mail system running the HP OpenMail
  - VideoStreaming: collected when playing a movie (sequential access)

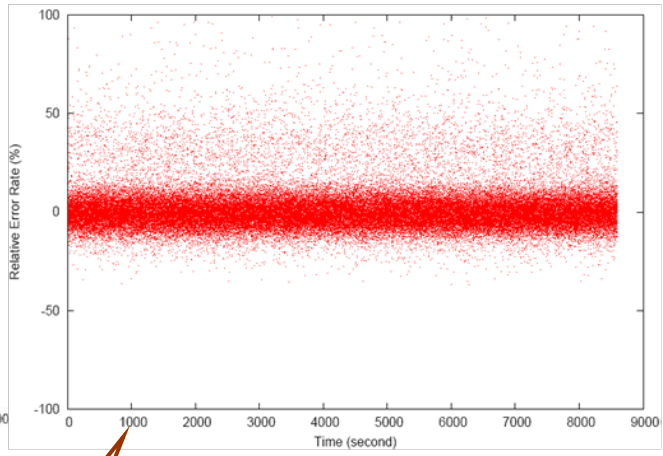
Accuracy of the RSS Interface interpreted by CART



Estimation Accuracy for Time Windows







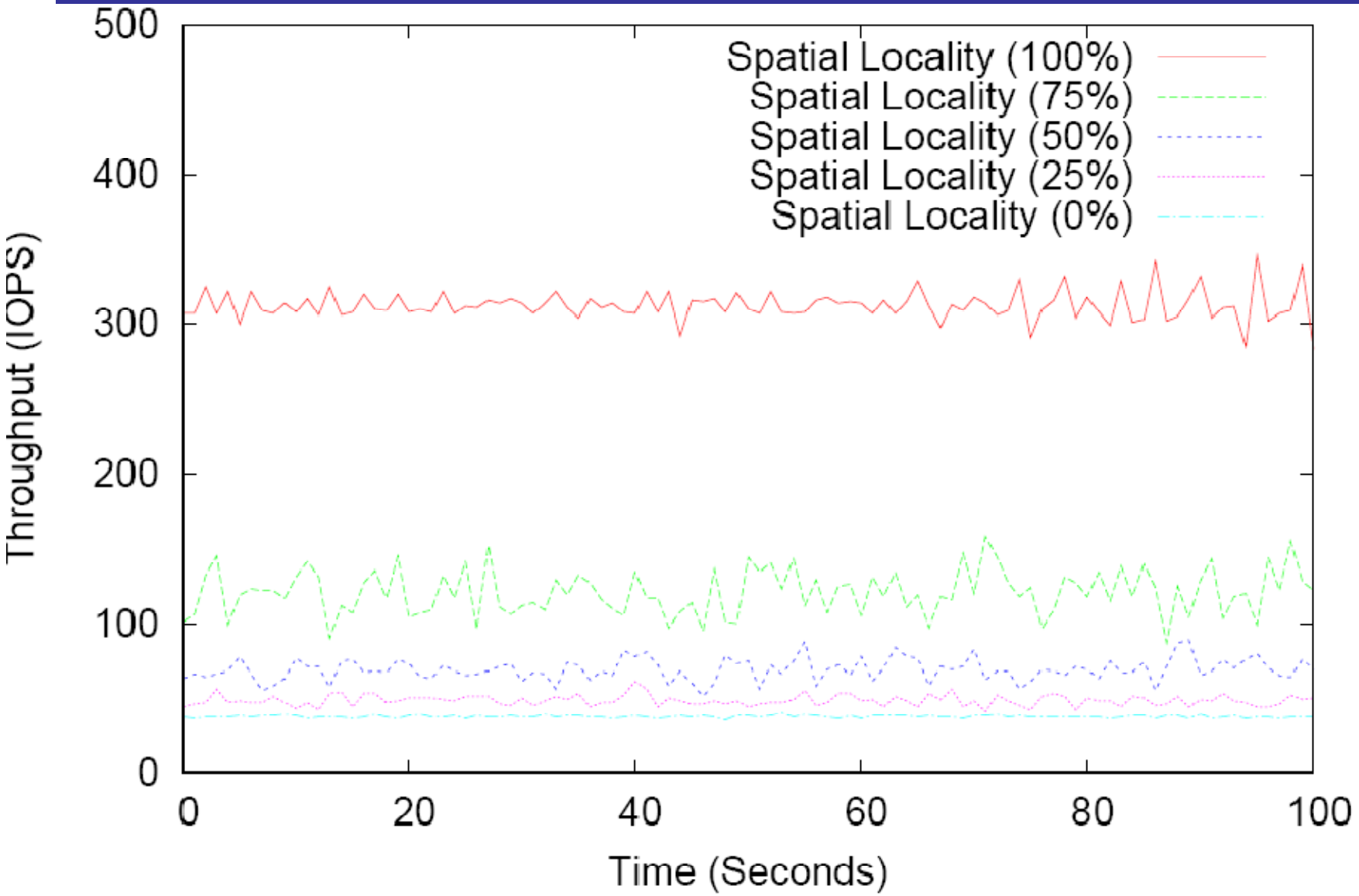
for Individual Requests

for 0.04s Time Window

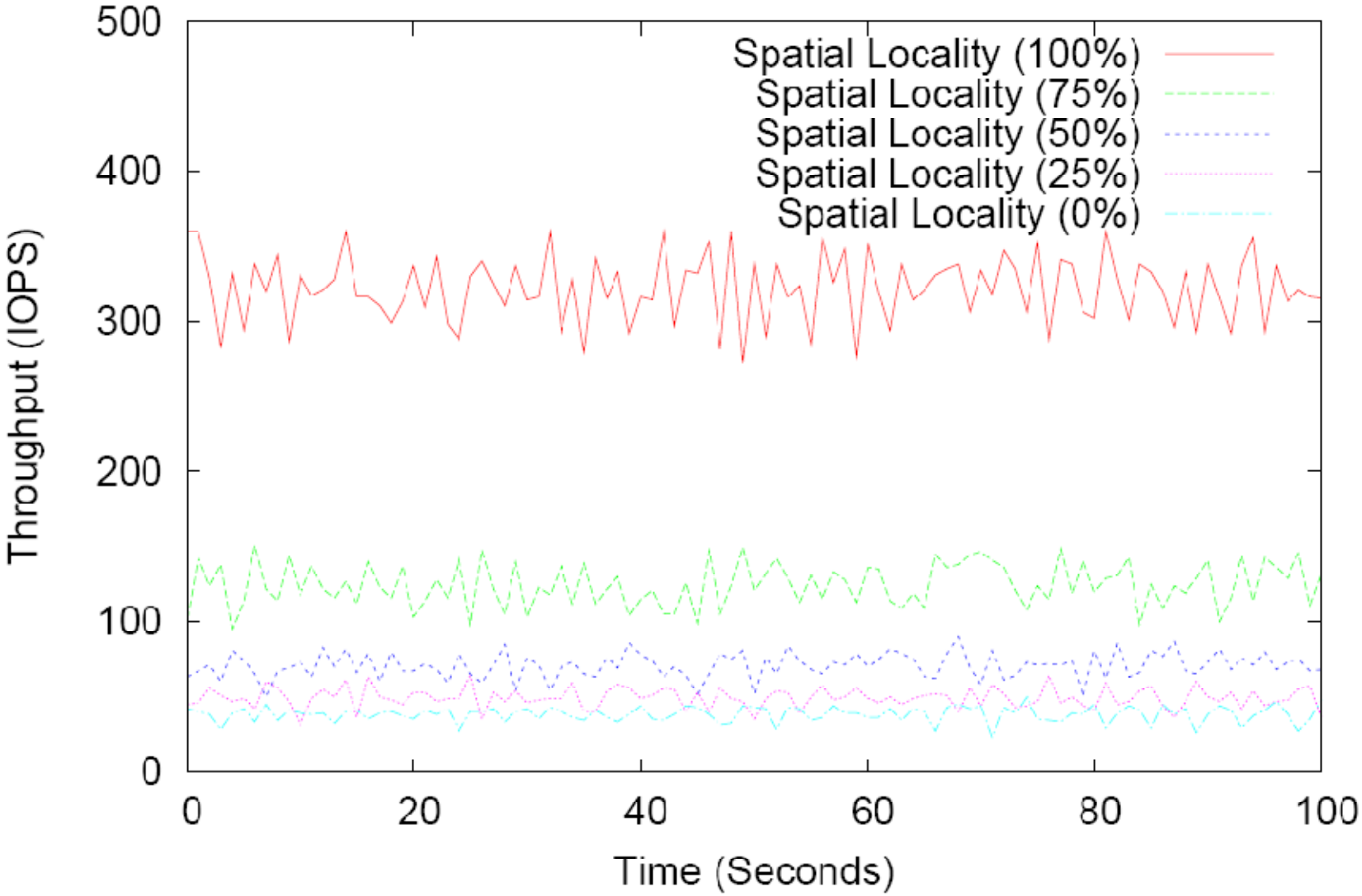
for 0.08s Time Window

More than 85% of relative errors are smaller than 15%

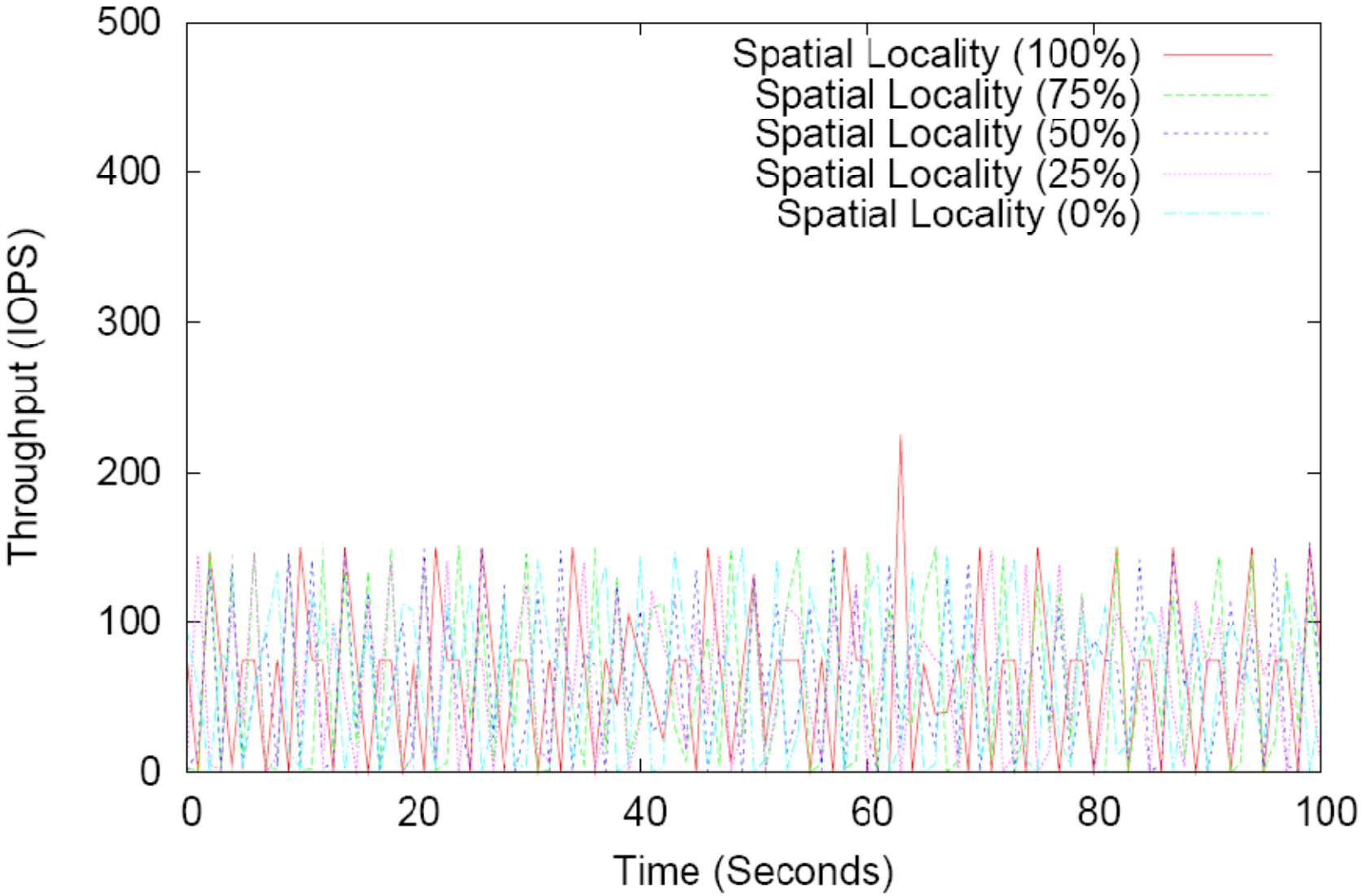
Impact of Spatial locality (on Dedicated RSS)



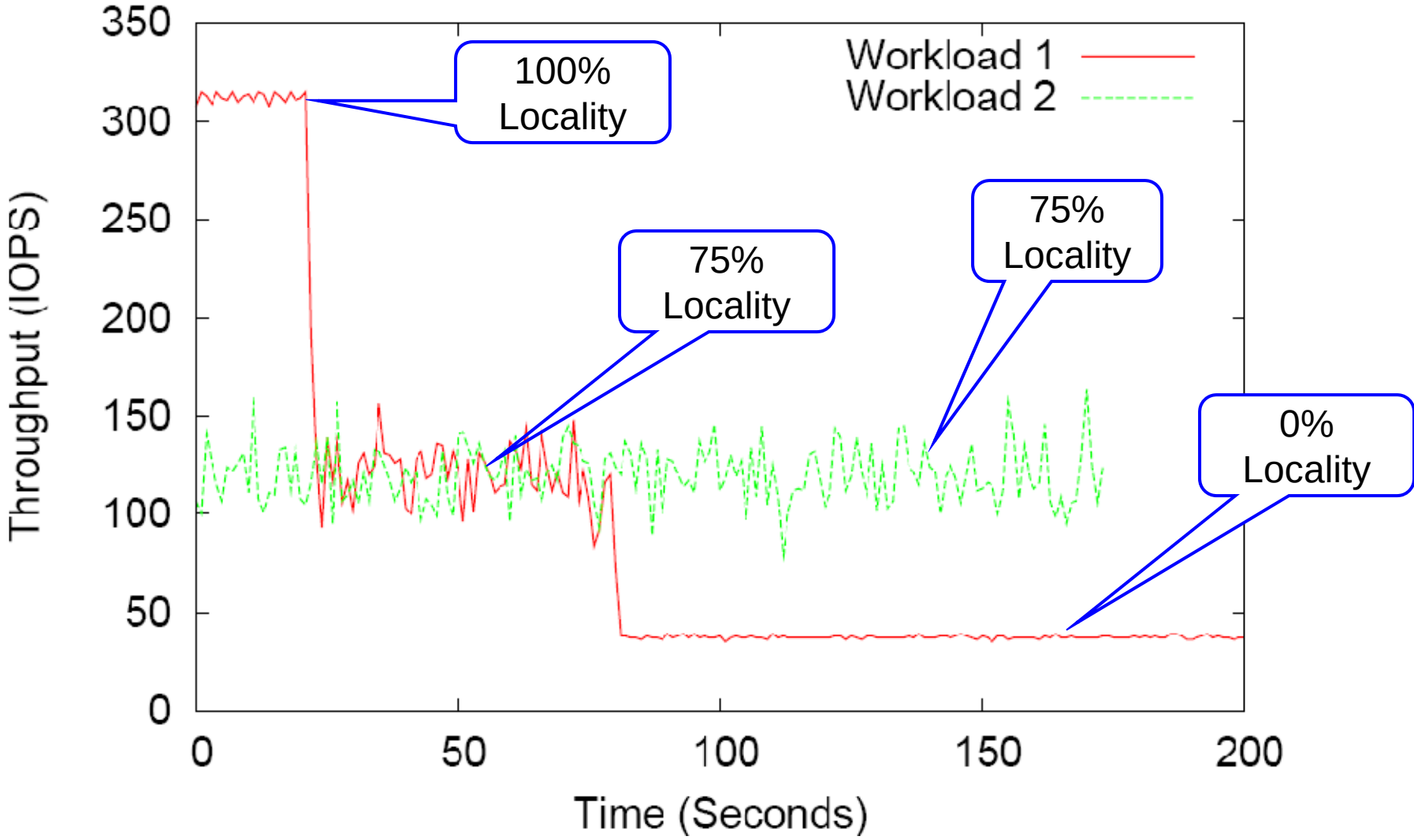
Impact of Spatial locality (on Shared Storage w/ YouChoose)



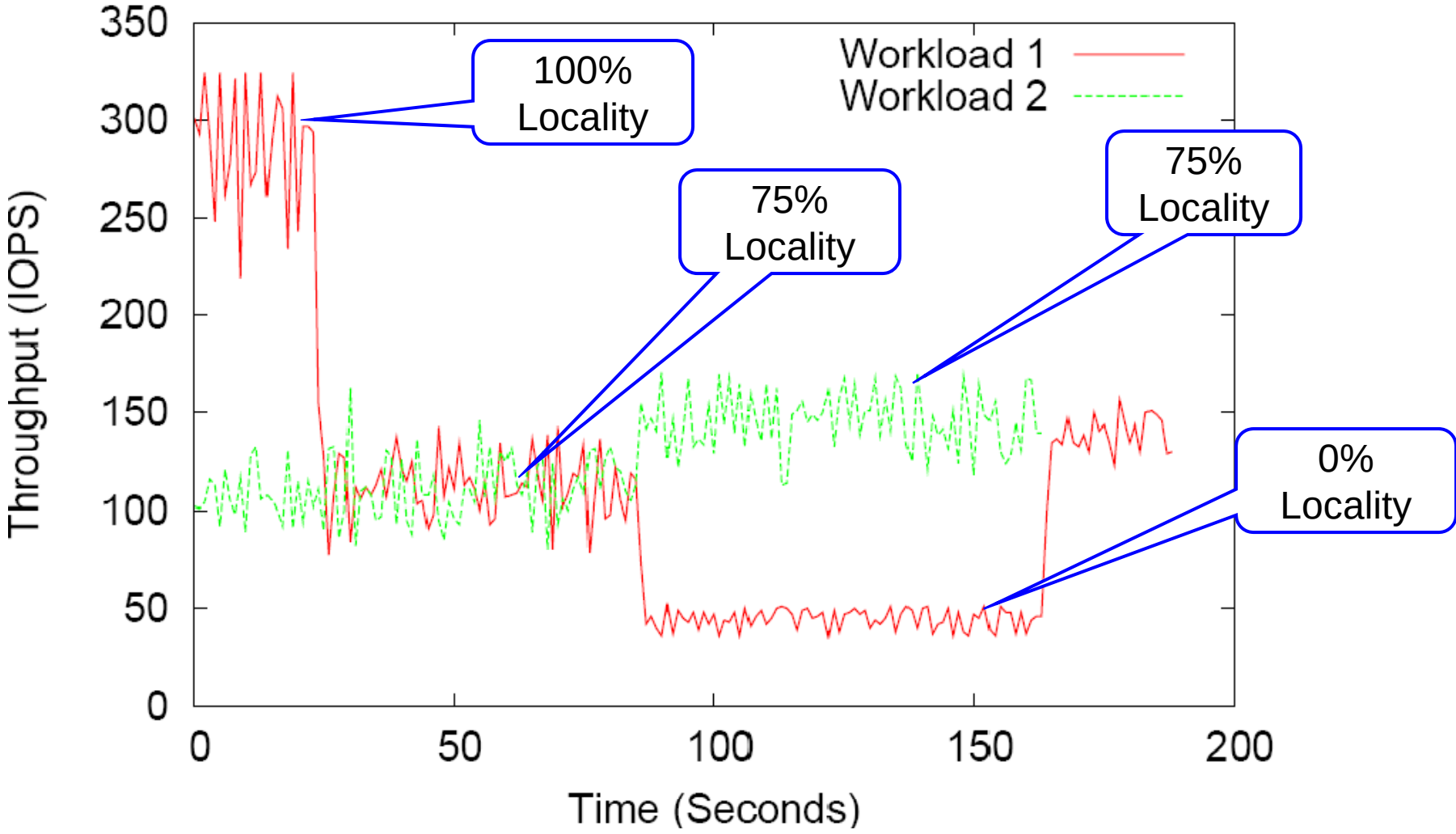
Impact of Spatial locality (on Shared Storage using the *100 IOPS* Bound)



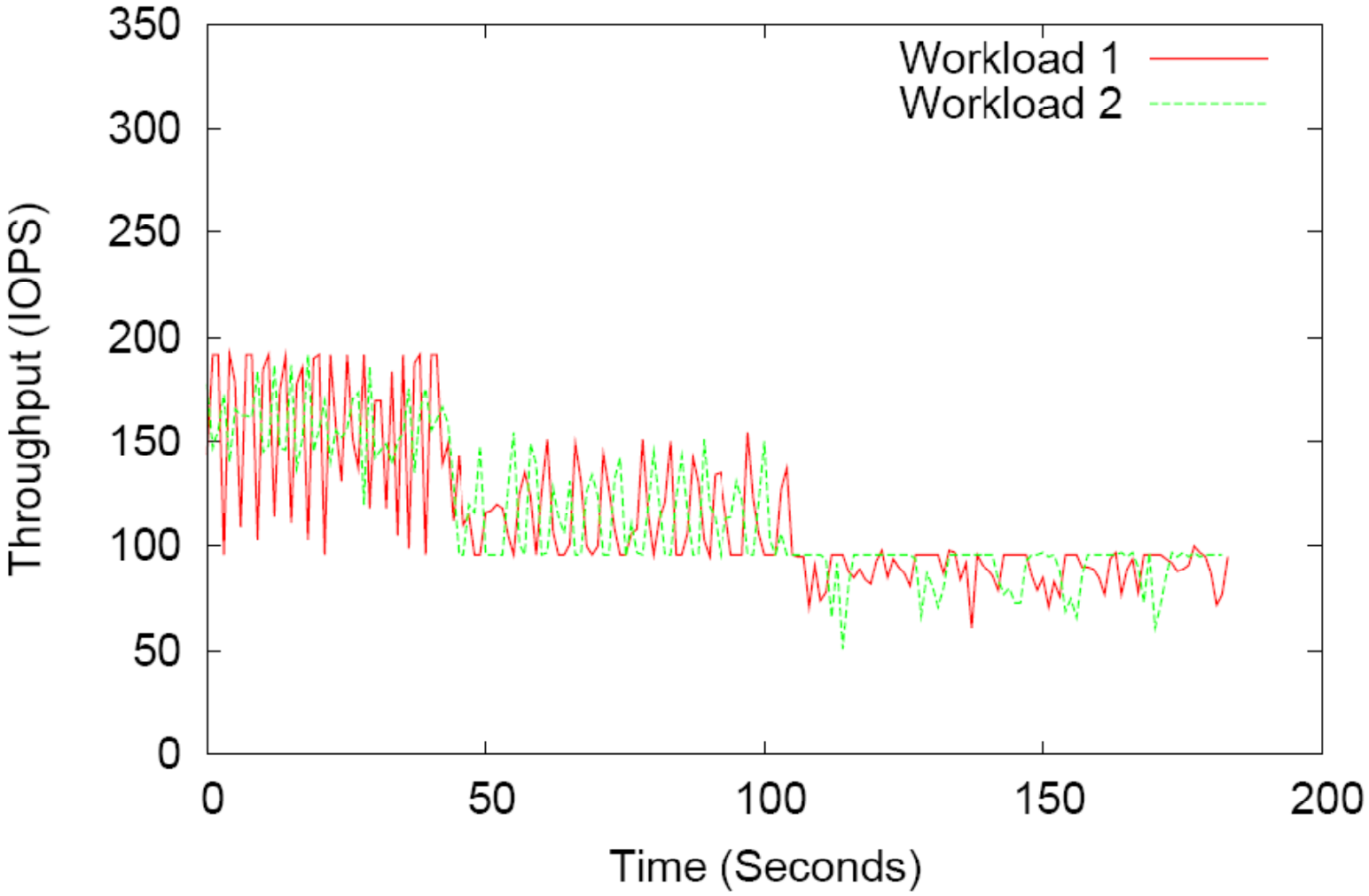
Performance Isolation (on Dedicated RSS)



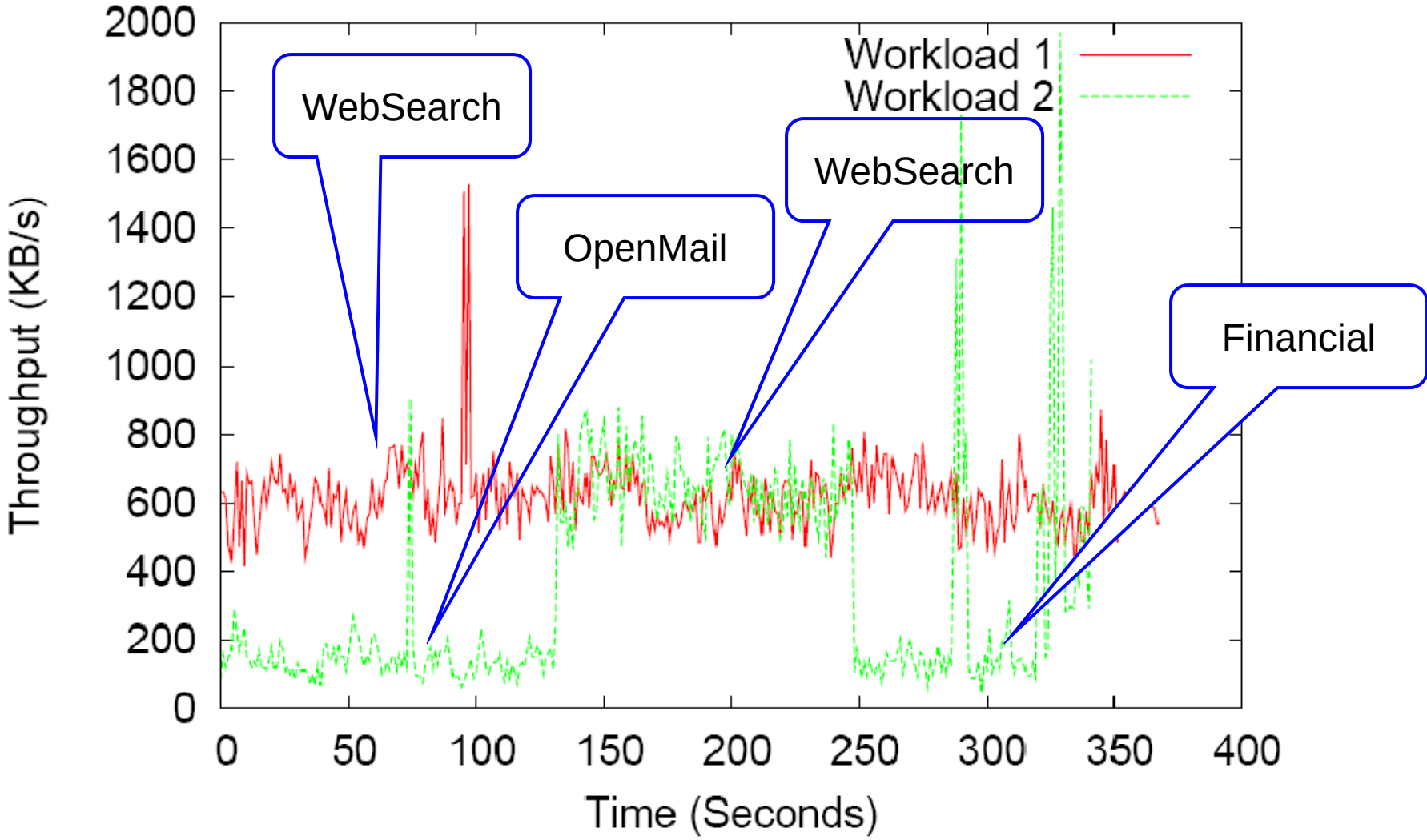
Performance Isolation
(on Shared Storage w/ YouChoose)



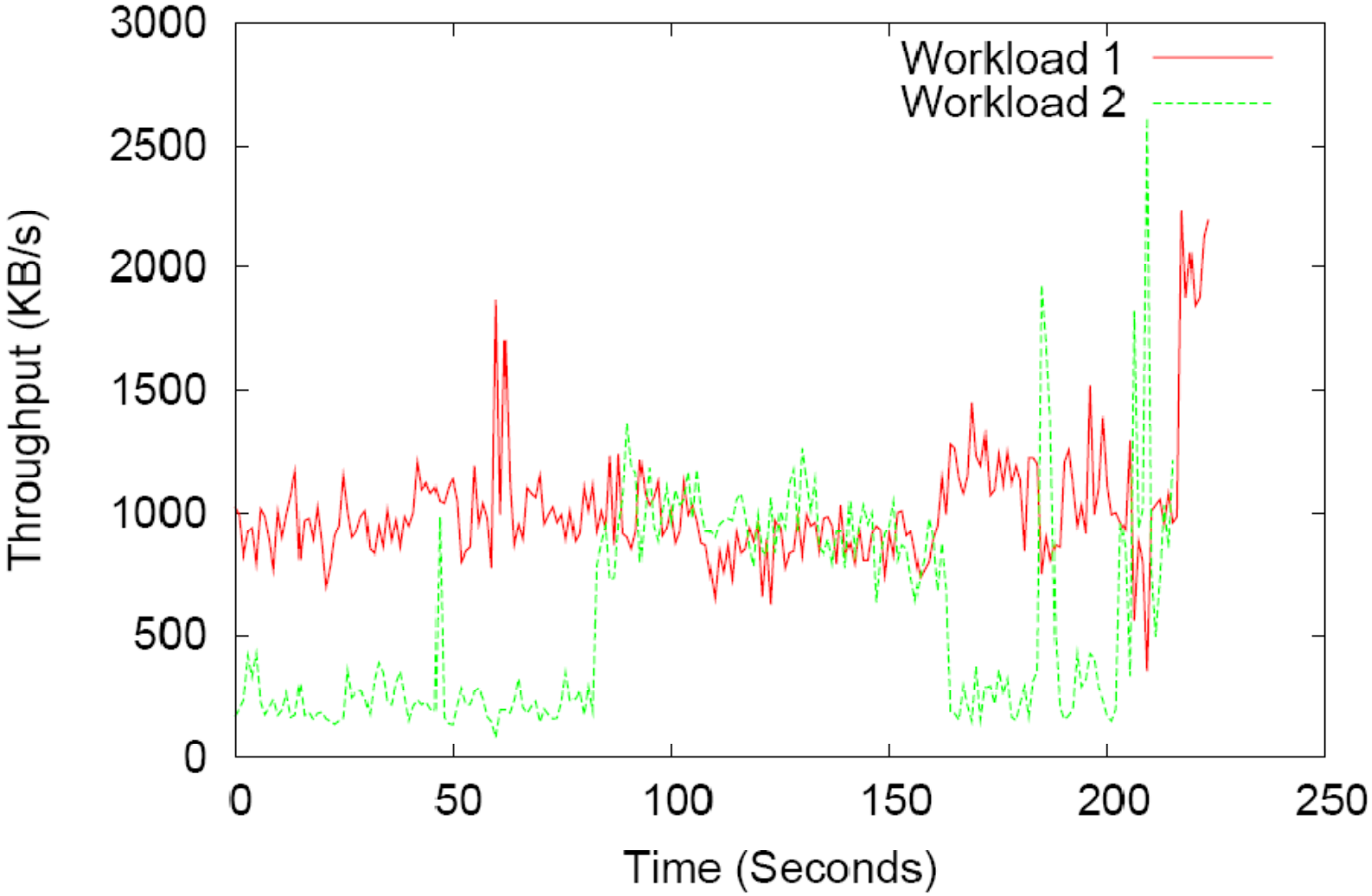
Performance Isolation
(on Shared Storage using the *100 IOPS* Bound)



Performance Isolation
(Real-world workloads on dedicated RSS)



Performance Isolation (Real-world workloads)



Conclusions

- Introduce reference storage system as the performance interface.
  - Dynamic access behavior is well accommodated in the interface.
  - Resource demand is well capped.
- Use the machine learning technique to implement the RSS interface.
- Achieve system efficiency with batched request scheduling.