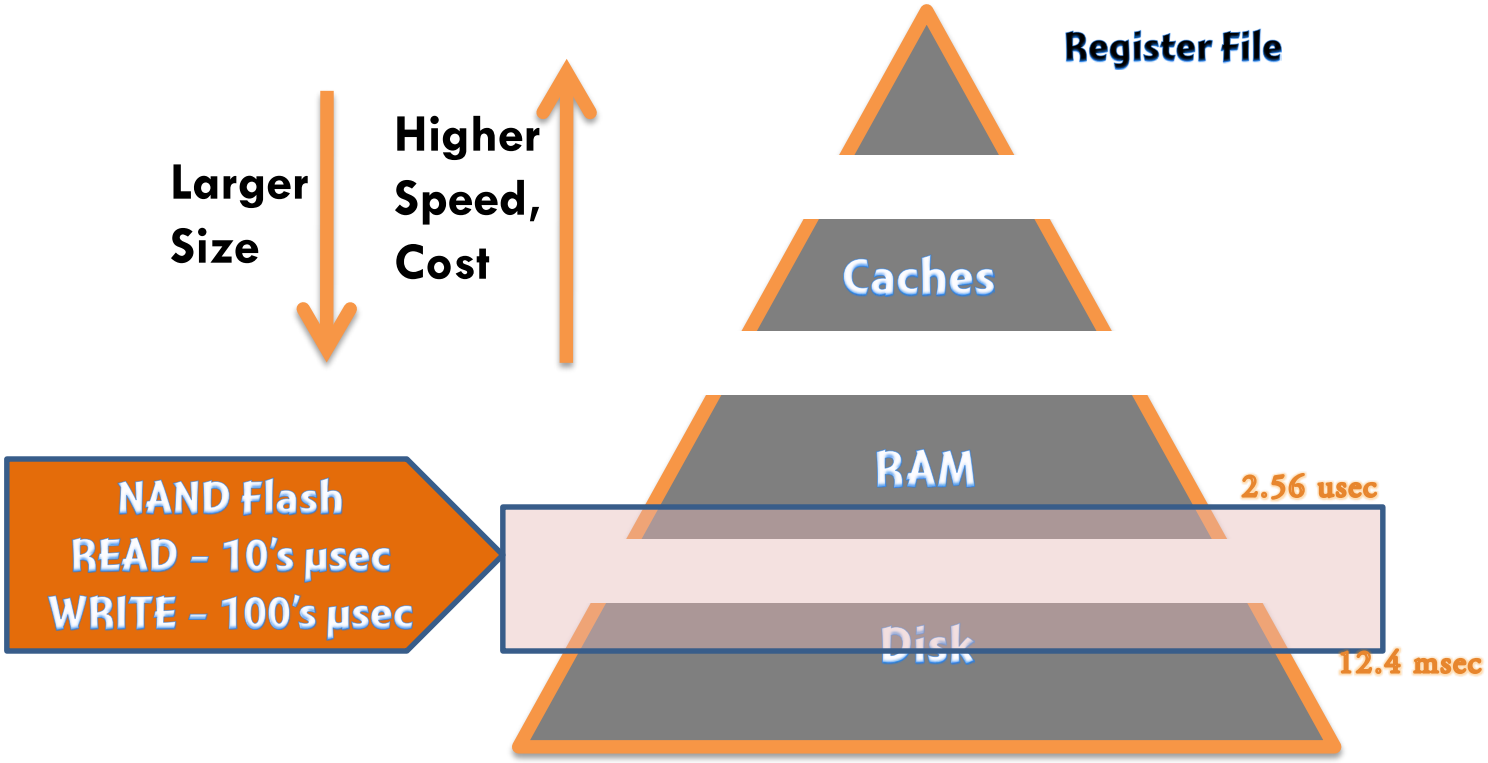
Basics of Storage Systems

Jihong Kim
Dept. of CSE, SNU

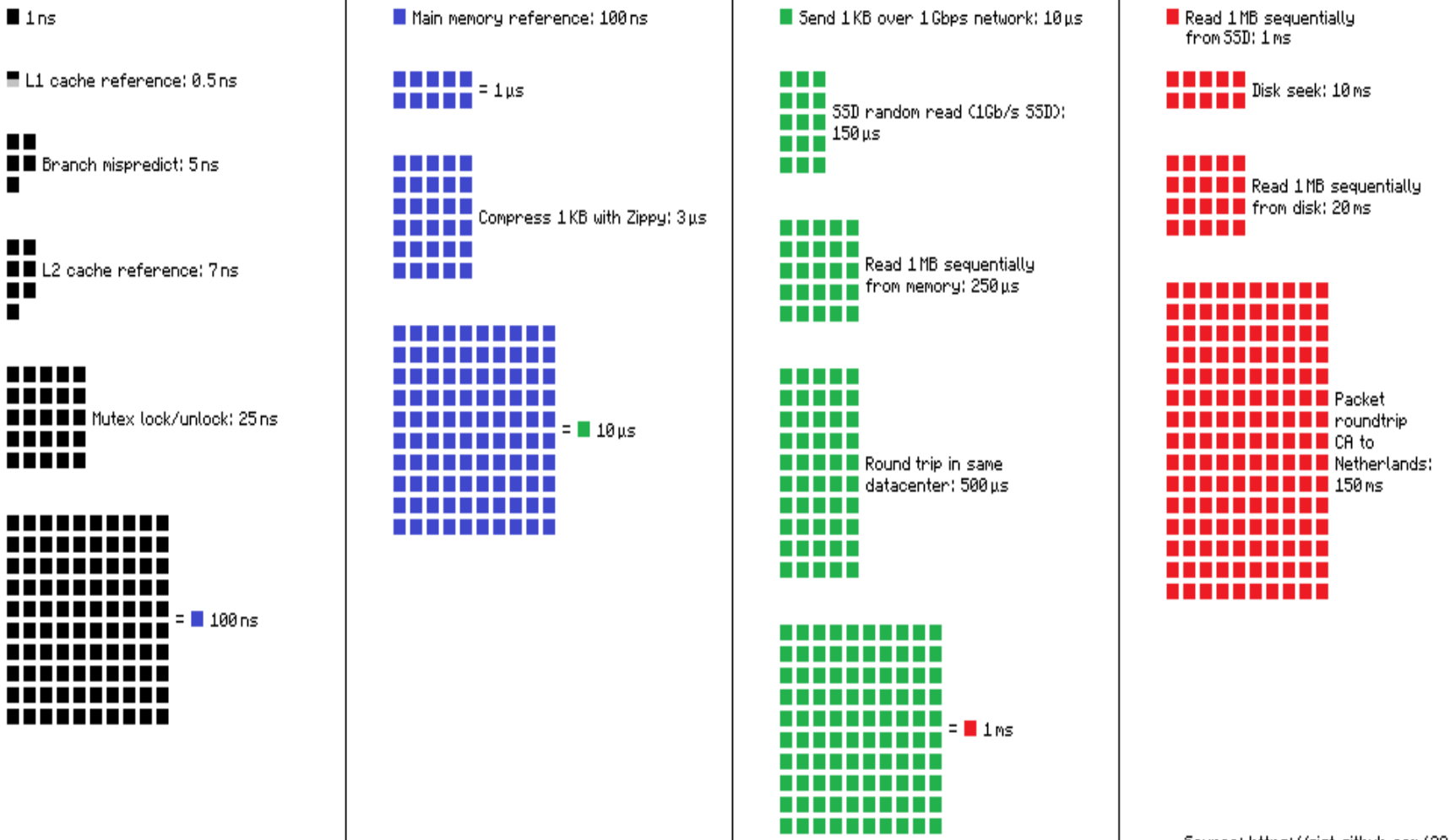
Outline

- Storage Systems
- Performance of I/O Systems
- Performance Metrics
- Storage Performance Benchmarking

Memory Hierarchy

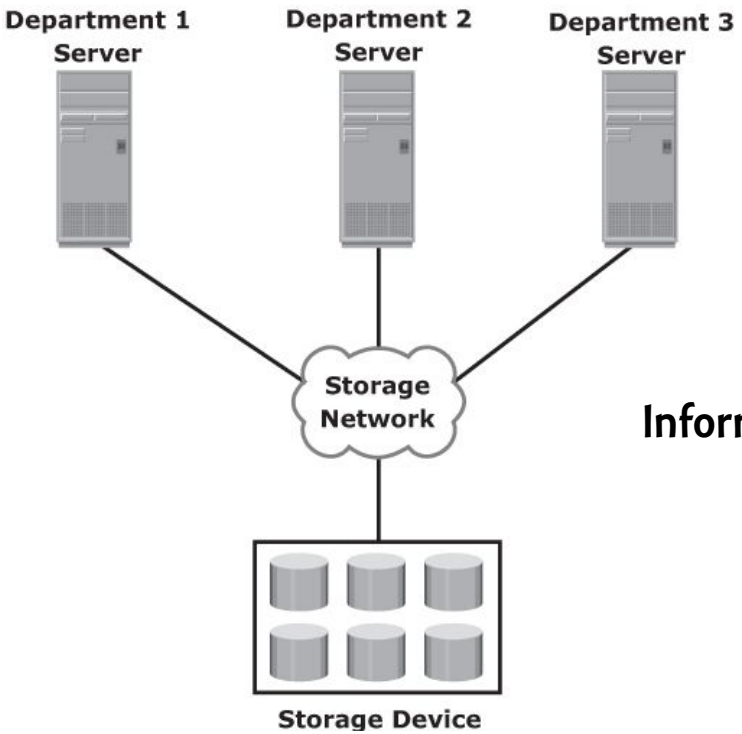


Latency Numbers Every Programmer Should Know



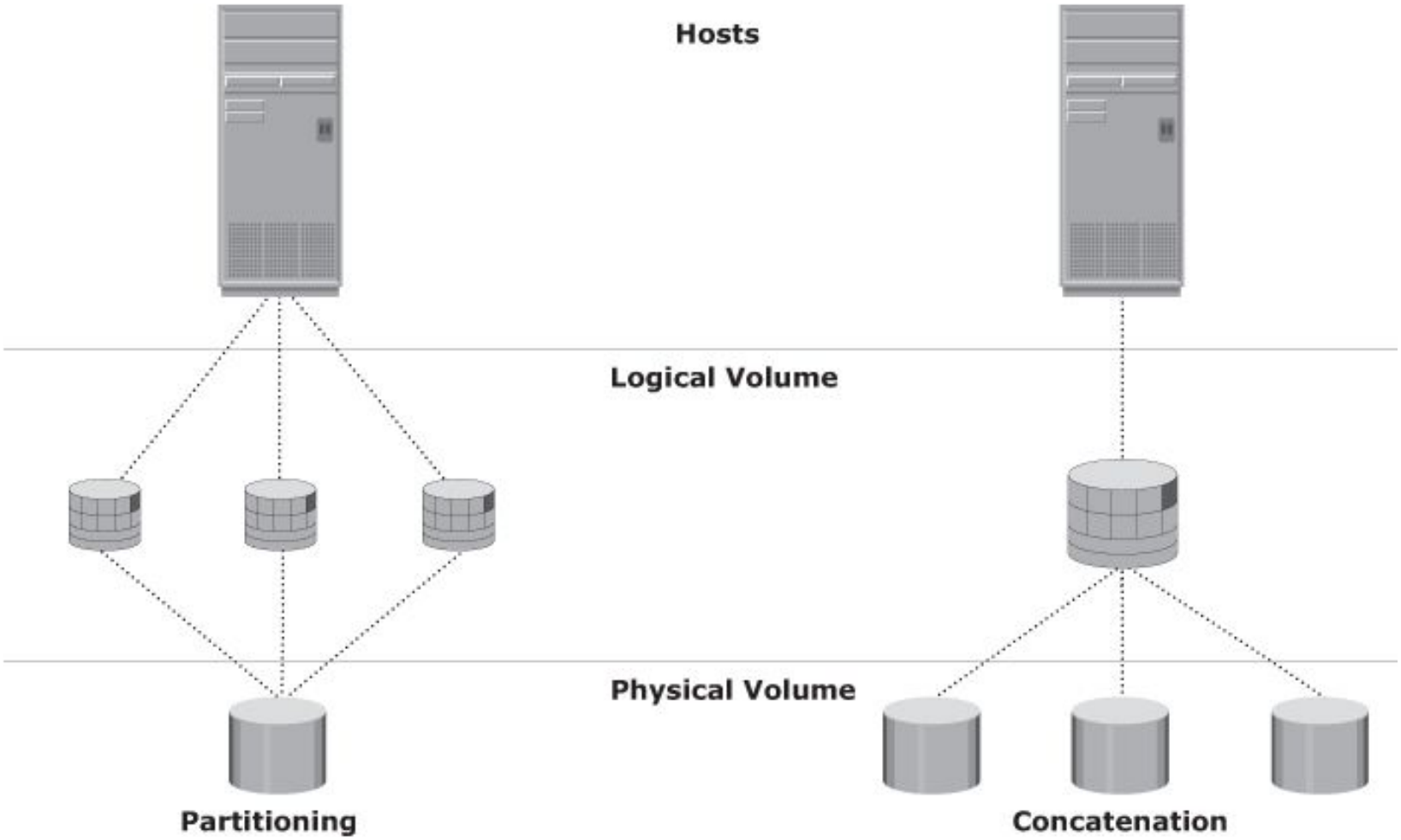
Storage Architecture

Server-Centric Storage Architecture

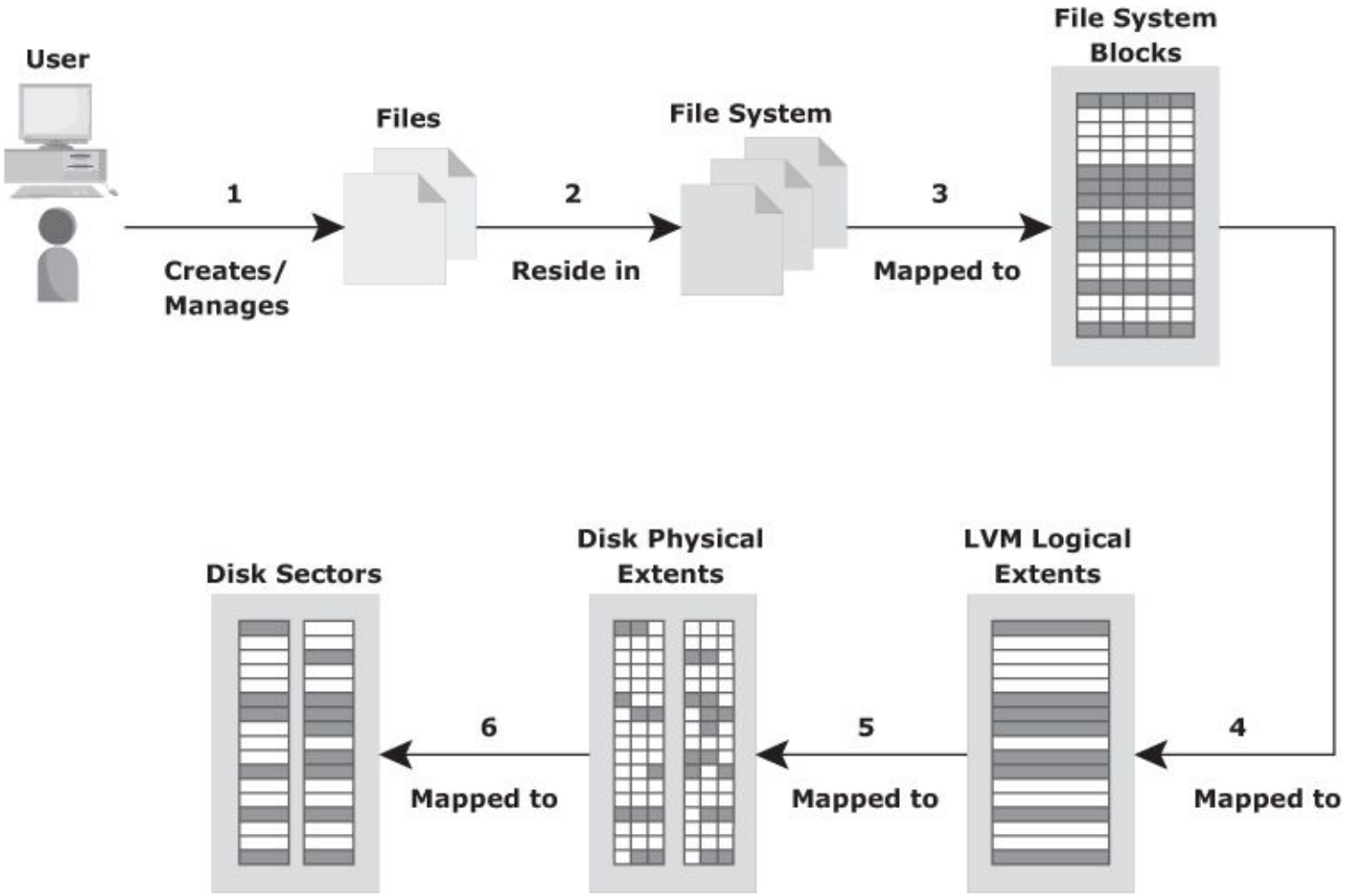


Information-Centric Storage Architecture

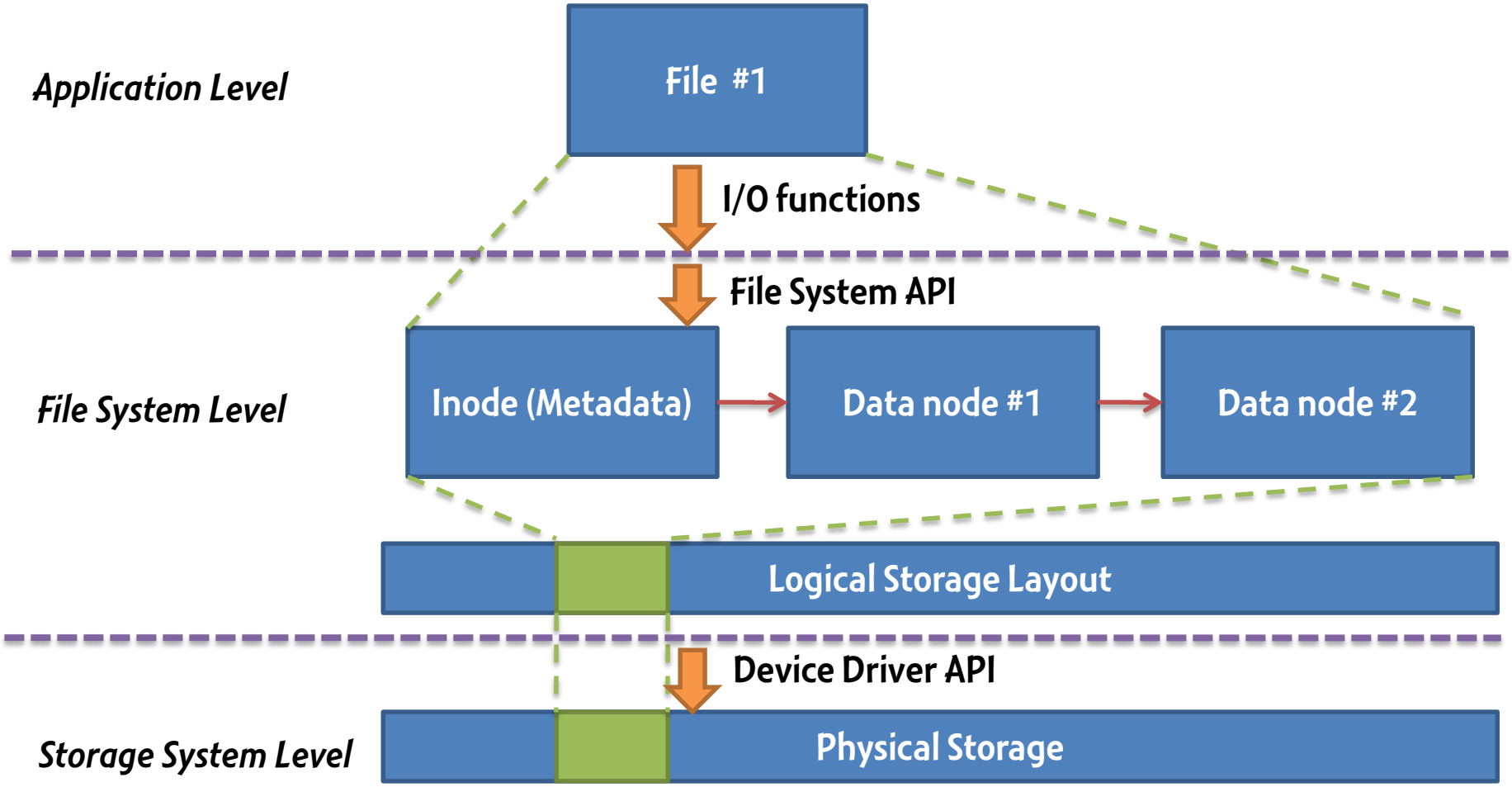
Logical Volume Manager



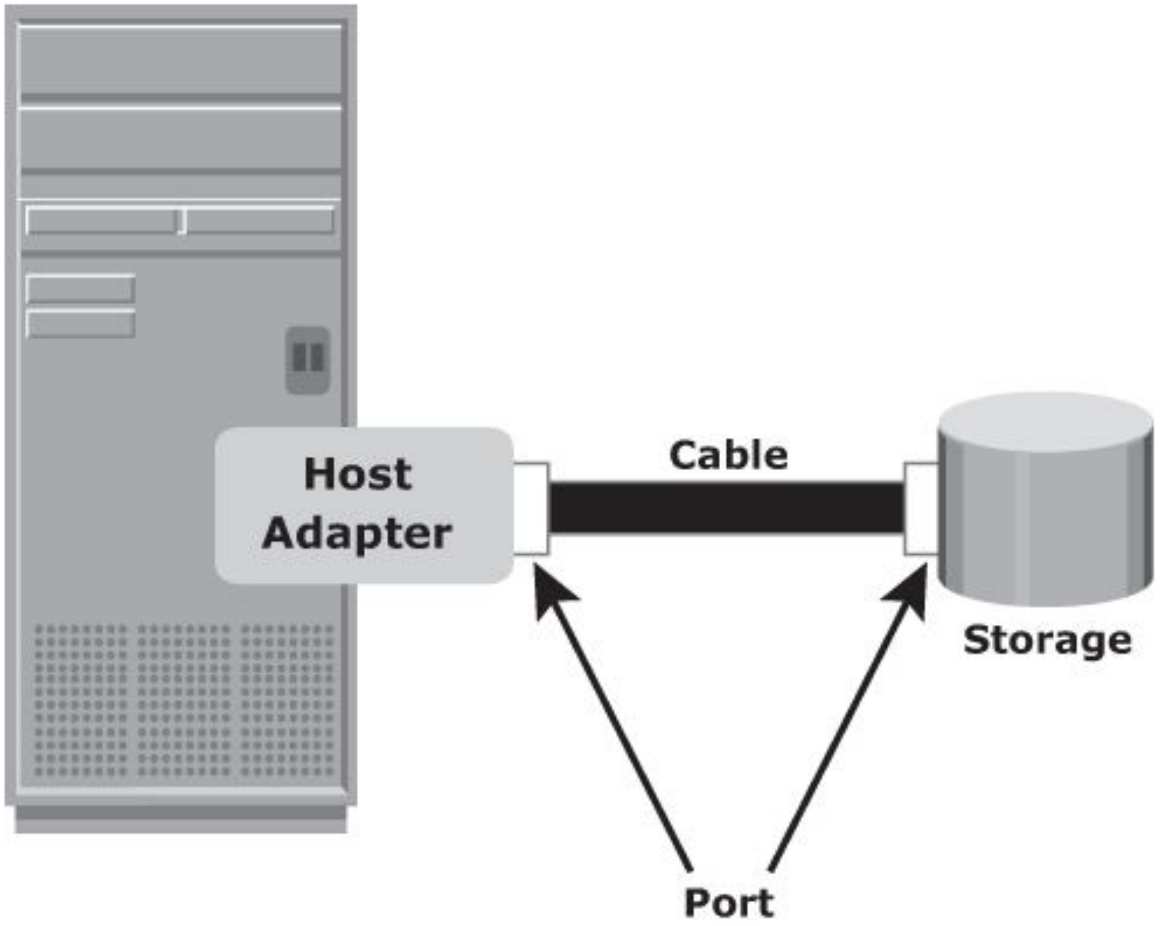
Files to Storage



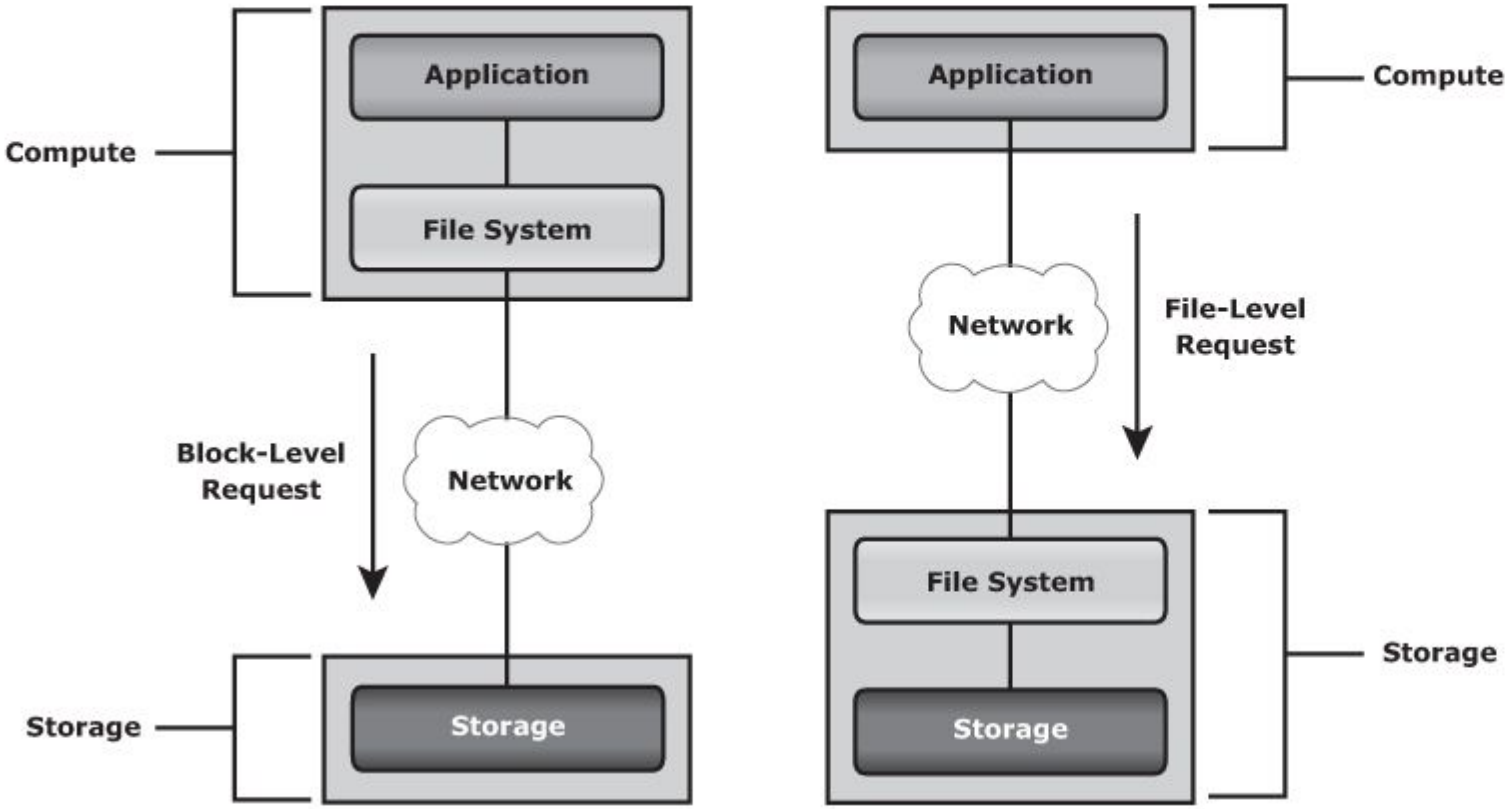
Data in Storage Systems



Interface Protocols



Host Access to Data



Metrics for Storage Systems

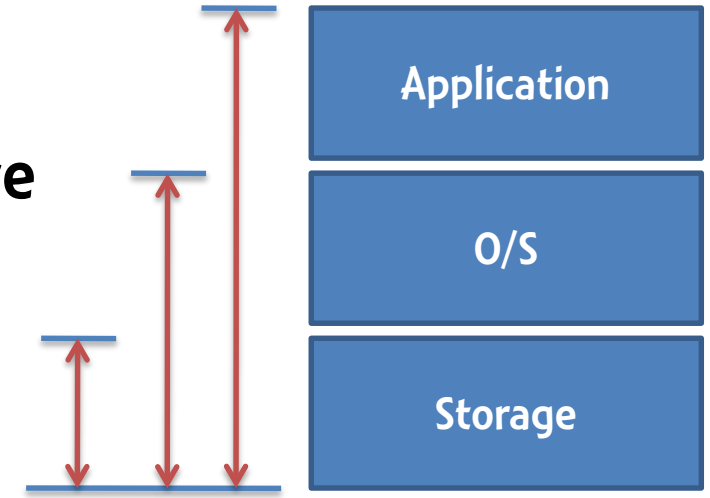
- Throughput
- Response Time
- Capacity
- Reliability
- Cost
- …

Throughput

- IO rate
  - IOPS (accesses/second)
  - Used for applications where the size of each request is small
  - e.g. transaction processing
- Data rate
  - MB/Sec or bytes/Sec
  - Used for applications where the size of each request is large
  - e.g. scientific applications

Response Time

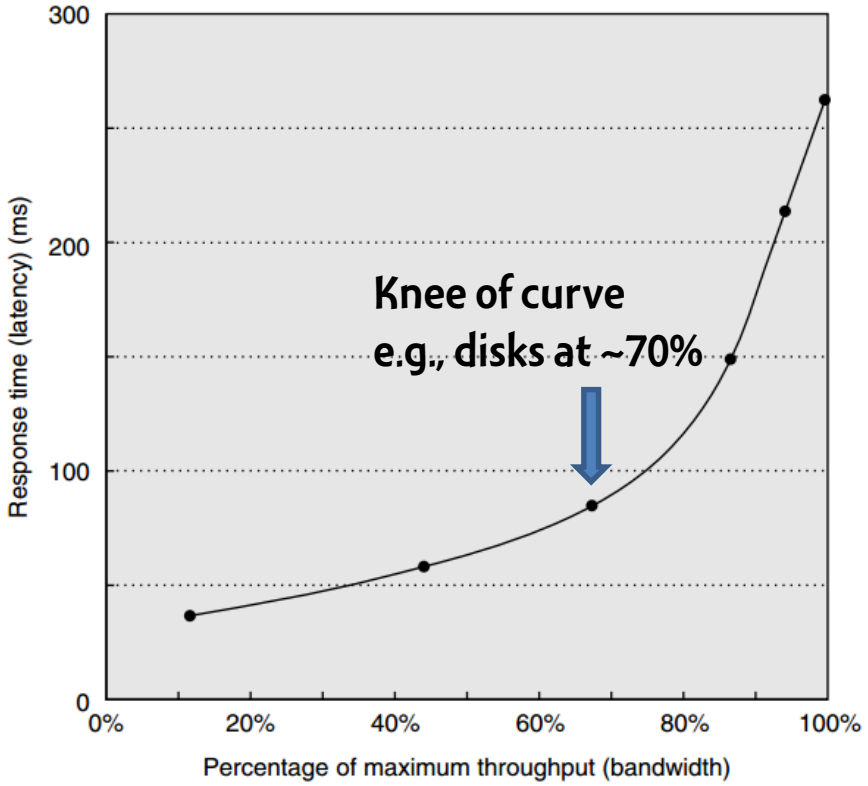
- How long a storage system takes to access data
- Depending on what you view as the storage system
  - User's perspective
  - OS's perspective
  - Disk controller's perspective

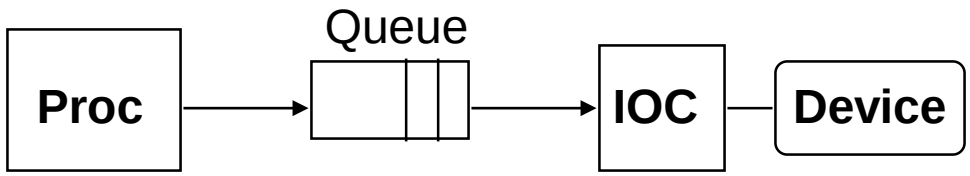


Disk I/O Performance

Metrics:
- **Response Time**
- **Throughput**

Q: how to balance response time & throughput? (e.g., adding more servers)





Response time = Queue + Device Service time

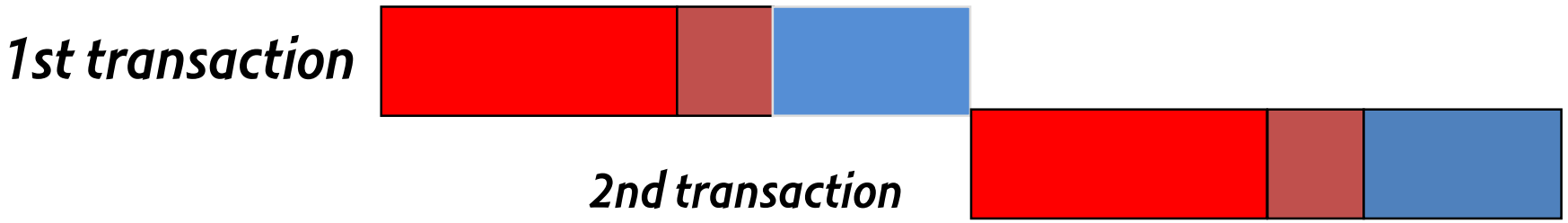
[Hennessy&Patterson]

Response Time vs. Productivity

- **Interactive environments:**

  Each interaction or *transaction* has 3 parts:

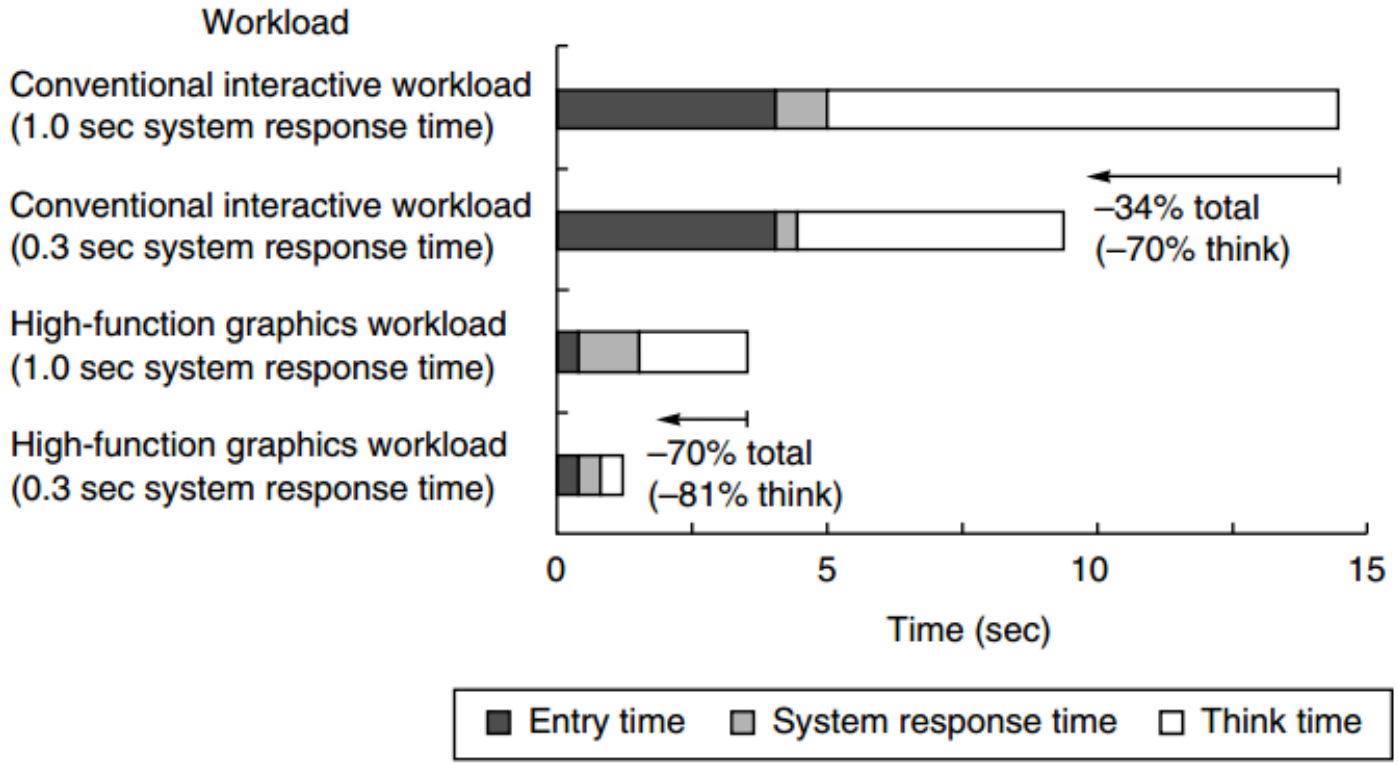
  - *Entry Time*: time for user to enter command
  - *System Response Time*: time between user entry & system replies
  - *Think Time*: Time from response until user begins next command



- **What happens to transaction time as system response time reduced from 1.0 sec to 0.3 sec?**
  - With Keyboard: 4.0 sec entry, 9.4 sec think time
  - With Graphics: 0.25 sec entry, 1.6 sec think time

[Hennessy&Patterson]

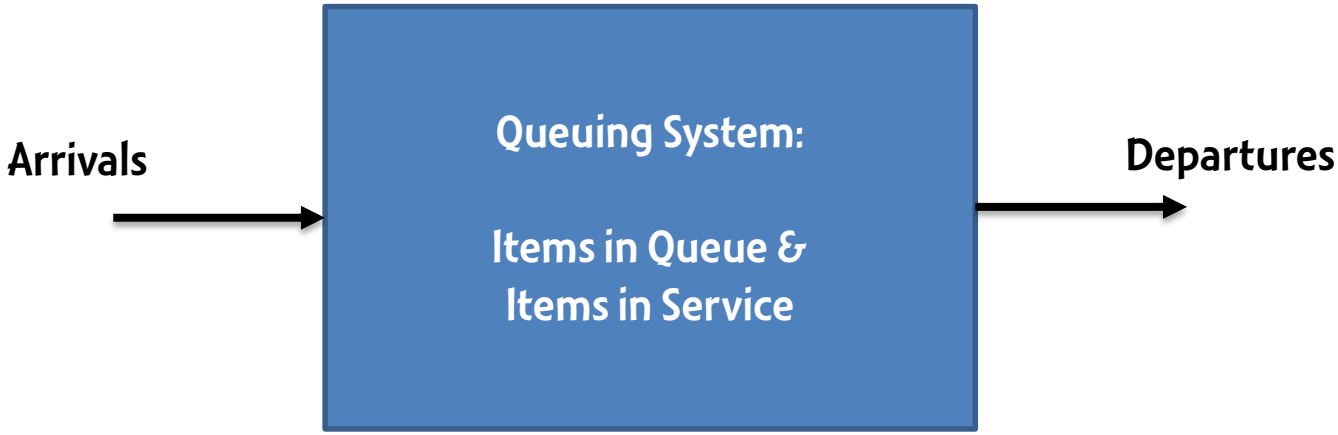
Response Time vs. Productivity



- 0.7sec off response saves 4.9 sec (34%) and 2.0 sec (70%) total time per transaction => greater productivity
  - People need less time to think when given a faster response
- Another study: everyone gets more done with faster response, but novice with fast response = expert with slow

[Hennessy&Patterson]

Little's Law: L = λ W



L = average number of items in the queuing system

W = average waiting time in the system for an item

λ = average number of items arriving per unit time

If you know two, one can be easily computed.

Example 1: Little's Law

- 매년 신입생이 2000명인 학교에서
  - λ = 2000/년
- 평균적으로 학교를 다니는 학생이 3000명이면
  - L = 3000
- 학생 1인당 학교에 다니는 기간은?
  - W = 3000/2000 = 1.5년

Example: Little's Law

For a single disk,

every second, 50 I/O requests are coming in &

avg disk service time = 10 ms,

what is the disk utilization?

답: 50 x 0.01

= 0.50 I/O request/disk

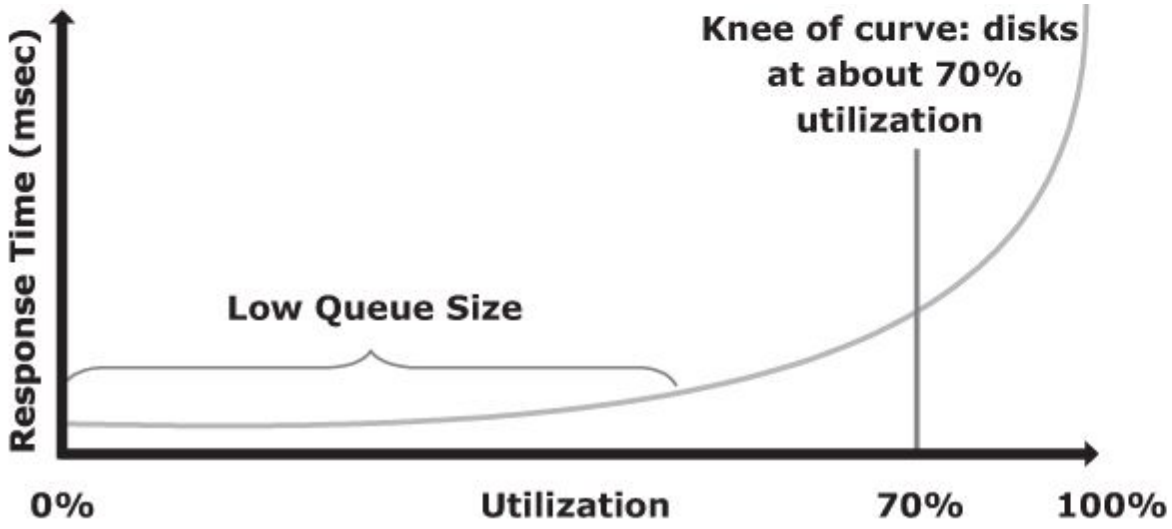
Queuing Theory: M/M/1

M: Expon. Random request arrival

M: Expon. Random service time

1: Single server

$$Time_{system} = Time_{server} \times 1/(1-Server_{utilization})$$



M/M/1 Example

- $Time_{system} = Time_{server} \times 1/(1-Server_{utilization})$
- 40 I/Os per second
- $Time_{server}$ = 20 ms?
  - $Server_{utilization} = 40 \times 0.02 = 0.8$
  - $Time_{system} = 20 \times 1/0.2 = 100$ ms
- What happens if $Time_{server}$ = 10 ms?
  - $Server_{utilization} = 40 \times 0.01 = 0.4$
  - $Time_{system} = 10 \times 1/0.6 = 16.7$ ms
- $Time_{server}$: 20 ms (80%) → 10 ms (40%)
  $Time_{system}$ = 100 ms → 16.7 ms

Useful Reference

- M. Hill, "Three Other Models of Computer System Performance," 2019. Available at https://arxiv.org/pdf/1901.02926.pdf.

Measurement Perspective

	Application	O/S	Storage
Characteristic	- Provides Overall performance - Sum(I/O,O/S, Application)	- Depends on file system behavior - Sum(I/O,O/S)	- Depends on storage device performance and firmware
Pros	- Provide total response time - Realistic usage scenario	- Able to evaluate file system performance - Provide realistic I/O system performance	- Able to evaluate performance of raw storage device
Cons	- Impossible to analyze - Less consistent results	- Less consistent results - Hard to replay	- Unrealistic - Consistent result - Easy to replay

Storage Performance Benchmarking

- Published Benchmark Programs
- Application simulation
  - Test a program with conditions similar to a target application
- Real Application testing

Application I/O Characteristics

- Random vs. Sequential I/O
  - E.g., Random: OLTP Sequential: data backup
- Reads vs. Writes
  - I/O rate = read rate + write rate
  - E.g., OLTP: 67% reads vs. 33% writes
- I/O Request Size

- Question:
  - For cross-layer optimizations, what hints/information should we extract from applications?

Published Benchmarks (1)

- Storage Performance Council
  - www.StoragePerformance.Org
  - SPC-1 test simulates an on-line transaction processing (OLTP) environment.
  - SPC-2 test to simulate large block sequential processing.
- Spec-SFS
  - www.Spec.Org/sfs97r1
  - A good test for measuring performance of file servers and network attached storage.
- TPC-C
  - www.tpc.org
  - TPC-C for testing OLTP, TPC-H for decision support and TPC-W for web e-commerce.

Published Benchmarks (2)

- IOMeter
  - An open source tools used to emulate the disk or network I/O load of any program or benchmark
  - Examines and records the performance of I/O operations and their impact on the system
  - Iometer is an I/O subsystem measurement and typographical tool.
  - Equips user with
    - Workload generator (to stress the system)
    - Measurement tool (examines and records the IO performance)

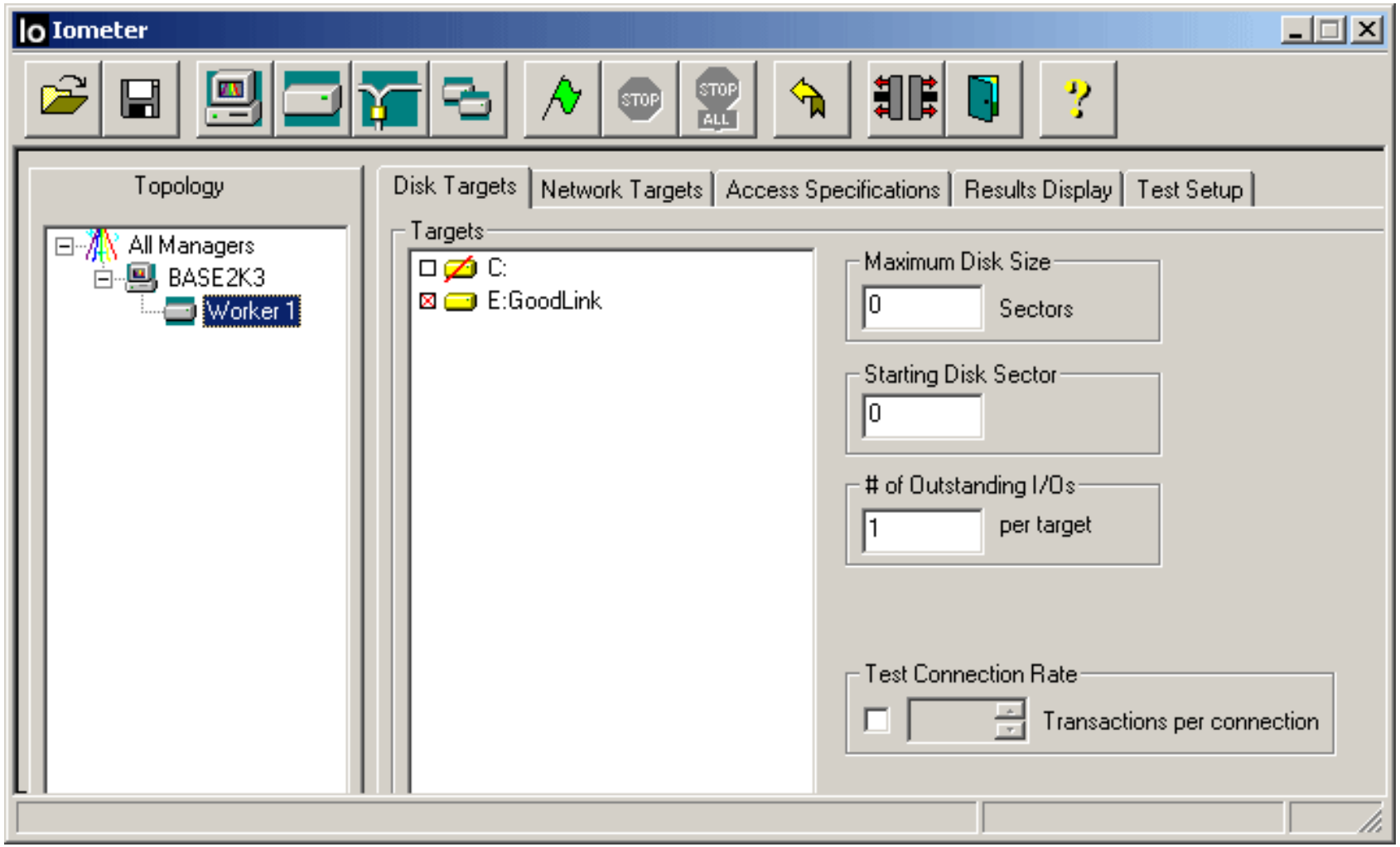
IOMeter Configuration

- The workload type
  - Read, Write, Random, Sequential, …
- Set operating parameters
  - Test time, request interval, …
- Collects the resulting data
- Summarizes the results in output files

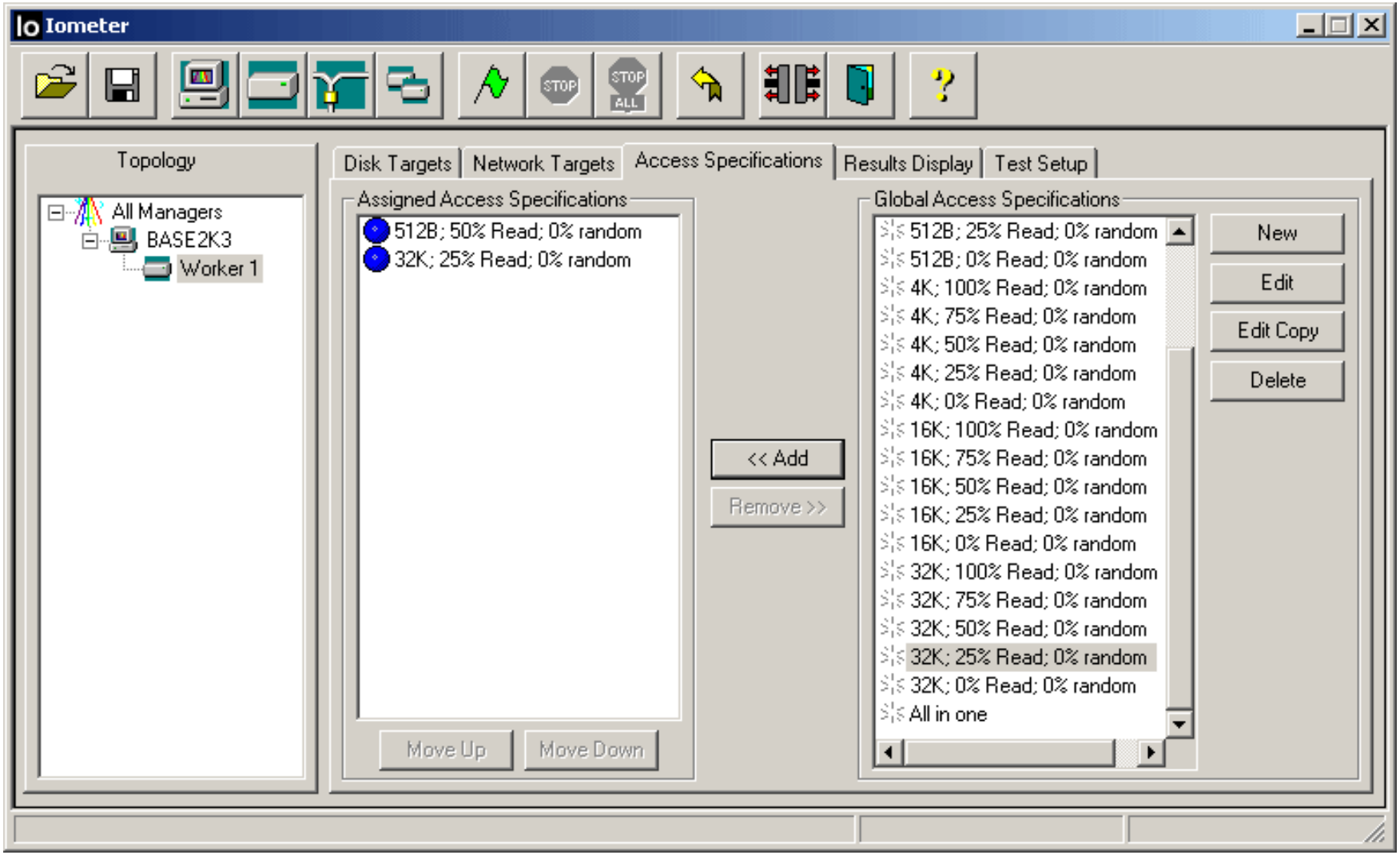
Measurement using IOMeter

- System-level I/O performance
- CPU utilizations
- Performance of disk and network controllers
- Bandwidth and latency capabilities of buses
- Network throughput to attached drives
- Error in reading and writing

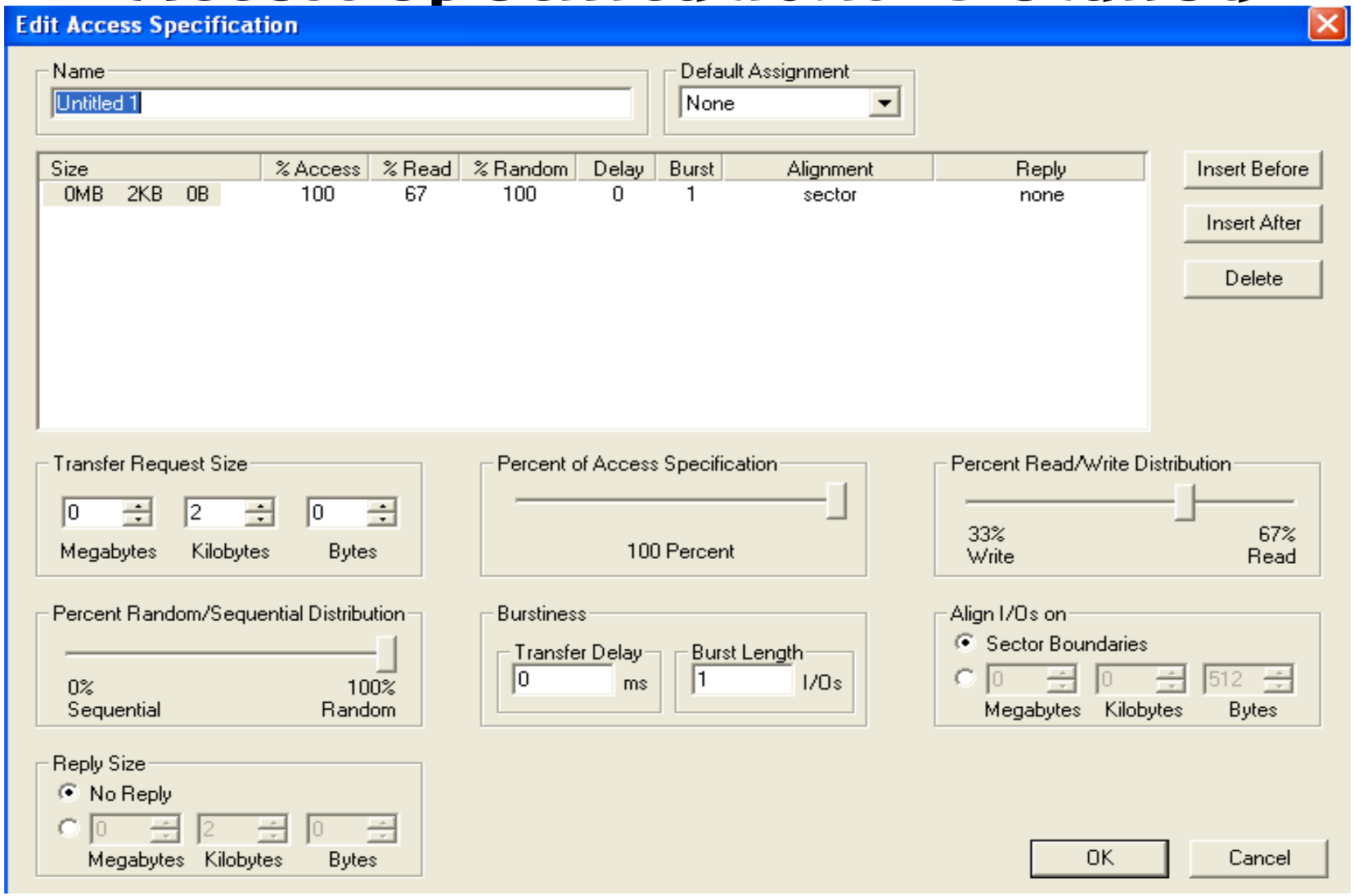
Topology and Disk targets



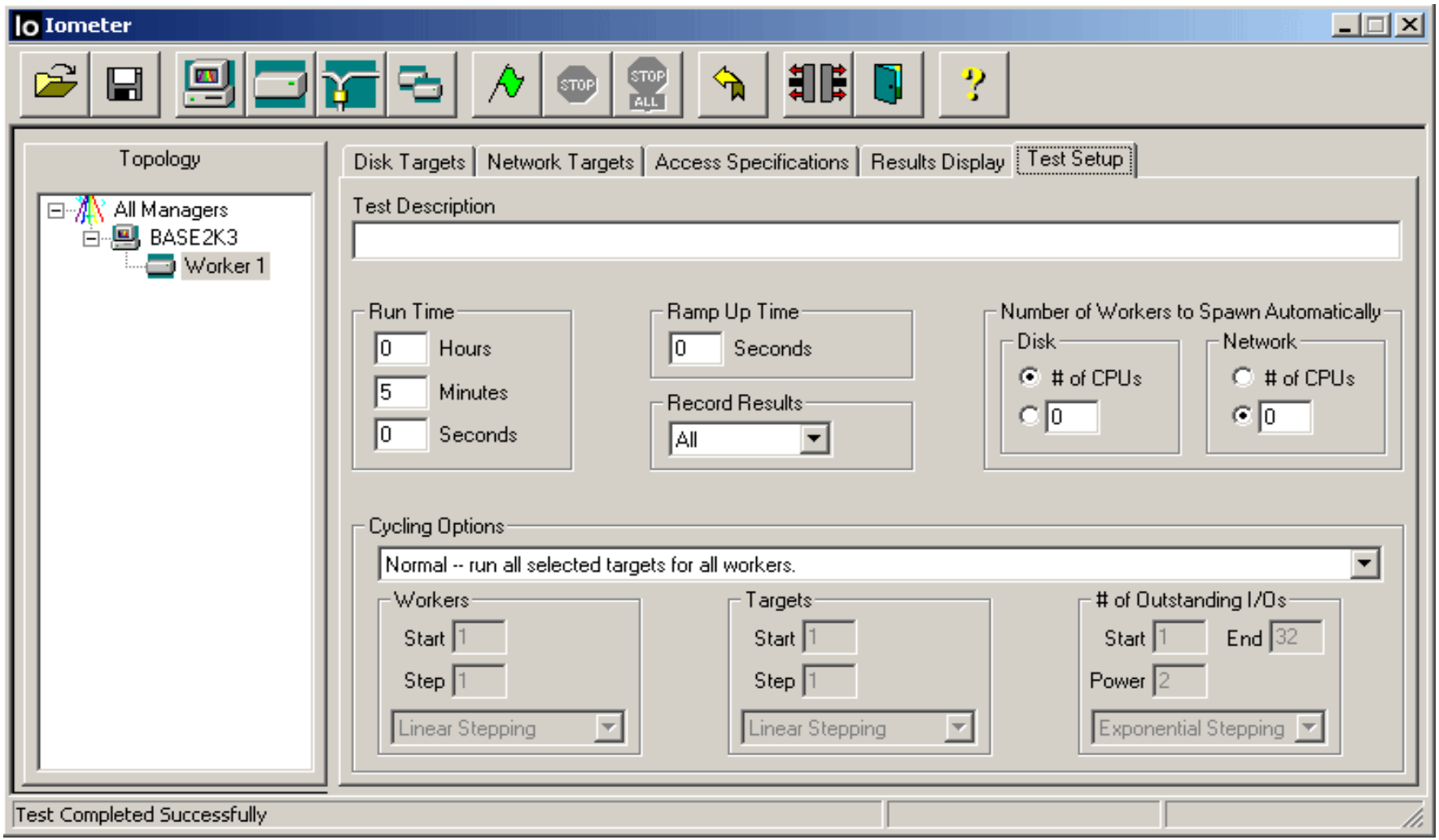
Access Specification



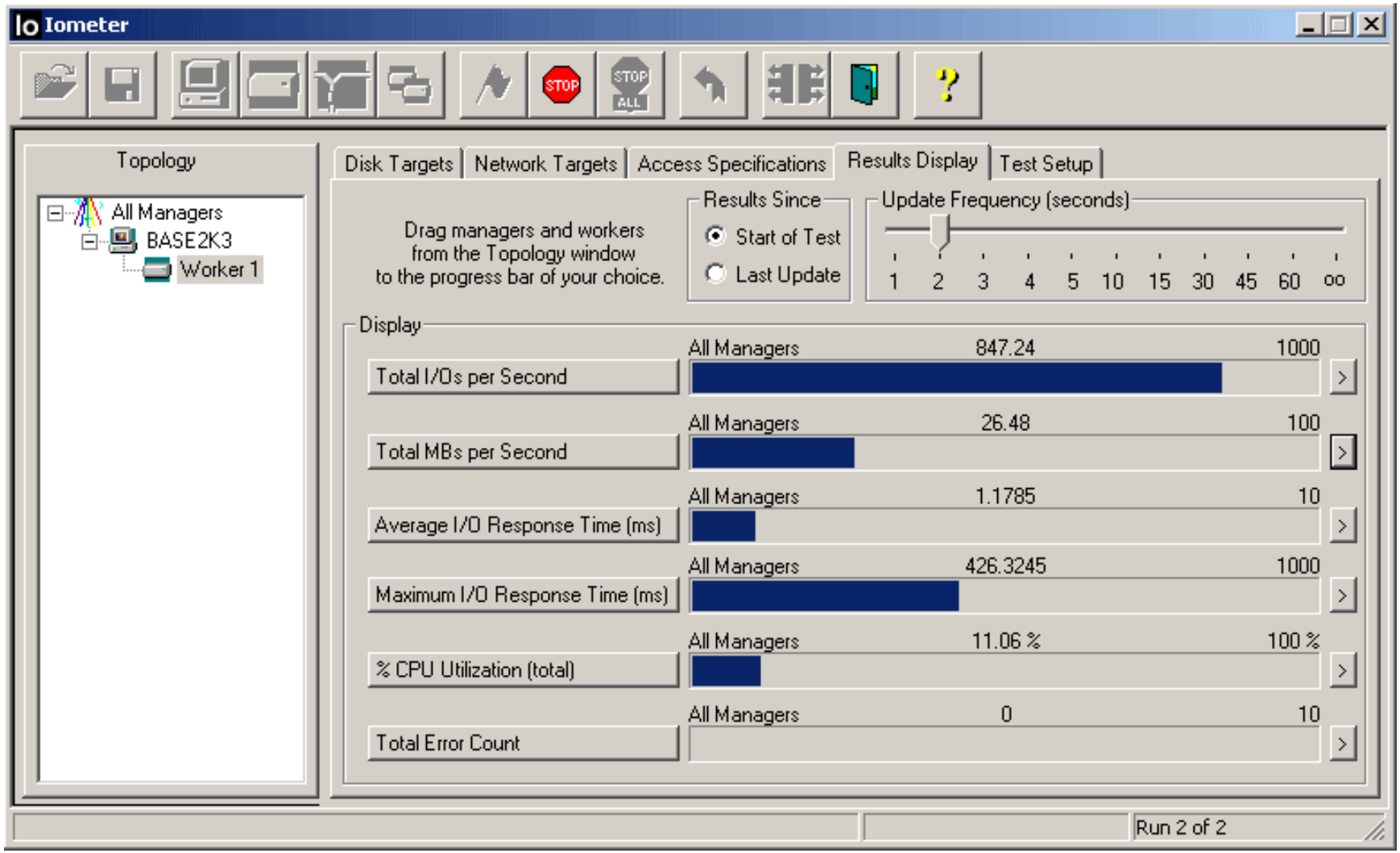
Access Specifications- Detailed



Test Setup


Iometer
Topology
All Managers
BASE2K3
Worker 1
Disk Targets
Network Targets
Access Specifications
Results Display
Test Setup
Test Description
Run Time
0 Hours
5 Minutes
0 Seconds
Ramp Up Time
0 Seconds
Record Results
All
Number of Workers to Spawn Automatically
Disk
# of CPUs
0
Network
# of CPUs
0
Cycling Options
Normal -- run all selected targets for all workers.
Workers
Start 1
Step 1
Linear Stepping
Targets
Start 1
Step 1
Linear Stepping
# of Outstanding I/Os
Start 1
End 32
Power 2
Exponential Stepping
Test Completed Successfully

Results display



Published Benchmarks (3)

- IOZone
  - A file system benchmark
  - Broad OS support
  - Available free from www.iozone.org.
  - Multiple stream measurement
  - Distributed fileserver measurements (Cluster)
  - Multi-process measurement

Tests using IOZone

- IOZone tests file I/O performance for the following operations
  - read, write, re-read, re-write, read backwards,
  - read strided, fread, fwrite, random read/write,
  - aio_read, aio_write, mmap, …

IOZone Usage

- **$ iozone [option]**

Usage: iozone [-s filesize_Kb] [-r record_size_Kb] [-f [path]filename]
[-i test] [-E] [-p] [-a] [-A] [-z] [-Z] [-m] [-M] [-t children] [-h] [-o]
[-l min_number_procs] [-u max_number_procs] [-v] [-R] [-x]
[-d microseconds] [-F path1 path2...] [-V pattern] [-j stride]
[-T] [-C] [-B] [-D] [-G] [-I] [-H depth] [-k depth] [-U mount_point]
[-S cache_size] [-O] [-K] [-L line_size] [-g max_filesize_Kb]
[-n min_filesize_Kb] [-N] [-Q] [-P start_cpu] [-c] [-e] [-b filename]
[-J milliseconds] [-X filename] [-Y filename] [-w] [-W]
[-y min_recordsize_Kb] [-q max_recordsize_Kb] [-+m filename]
[-+u] [-+d] [-+p percent_read] [-+r] [-+t] [-+A #]

Example: IOZone (1)

```
$ ./iozone -a
```

```
Iozone: Performance Test of File I/O
        Version $Revision: 3.394 $
        Compiled for 32 bit mode.
        Build: linux

Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
             Al Slater, Scott Rhine, Mike Wisner, Ken Goss

Run began: Sat Apr 23 12:25:34 2011

Auto Mode
Command line used: ./iozone -a
Output is in Kbytes/sec
Time Resolution = 0.000001 seconds.
Processor cache size set to 1024 Kbytes.
Processor cache line size set to 32 bytes.
File stride size set to 17 * record size.
```

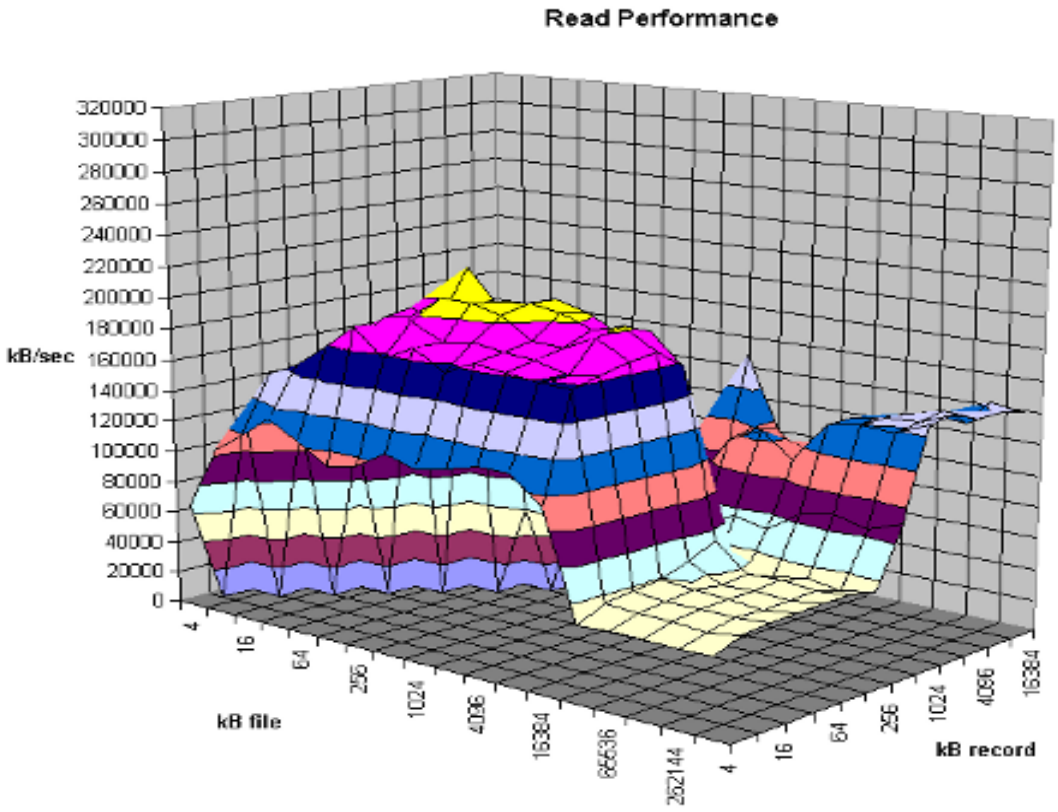
Example: IOZone (2)

KB	reclen	write	rewrite	read	reread	random read	random write	bkwd read	record rewrite	stride read	fwr
64	4	495678	152376	1824993	2065601	2204215	875739	582008	971435	667351	383
64	8	507650	528611	1051124	1563289	2071399	1084570	1332702	1143842	2138827	1066
64	16	587283	1526887	2560897	2778775	2366545	1122734	1254016	593214	1776132	463
64	32	552203	402223	1121909	1388380	1162129	415722	666360	1163351	1637488	1876
64	64	551580	1122912	2895401	4911206	2782966	1734491	1825933	1206983	2901728	1207
128	4	587259	1525366	1801559	3366950	1600898	1391307	1348096	547193	666360	458
128	8	292218	1175381	1966197	3451829	2165599	1601619	1232122	1291619	3273329	1827
128	16	650008	510099	4120180	4003449	2508627	1727493	1560181	1307583	2203579	1229
128	32	703200	1802599	2842966	2974289	2777020	1331977	3279734	1347551	1152291	684
128	64	848280	1294308	2288112	1377038	1345725	659686	1997031	1439349	2903100	1267
128	128	902120	551579	1305206	4727881	3046261	1405509	1802090	1085124	3649539	2066

...

Example: IOZone (3)

```
$ ./iozone -a -b output.xls
```



Benchmarking Software (4)

- Benchmark Factory for Databases by Quest Software
  - TPC-B, TPC-C, TPC-D
- Bonnie++
  - Performs a number of simple tests of hard drive and file system performance
  - Supports Linux
  - Available free from http://www.coker.com.au/bonnie++/

Application Simulation Testing

- One type of test does not represent all applications
- One type of application does not represent all uses for a storage product
- Common types of application simulation testing
  - Test storage latency for messaging or other single-threaded applications
  - Test peak storage bandwidth for data acquisition or data streaming environments
  - Test peak storage IOPS for databases

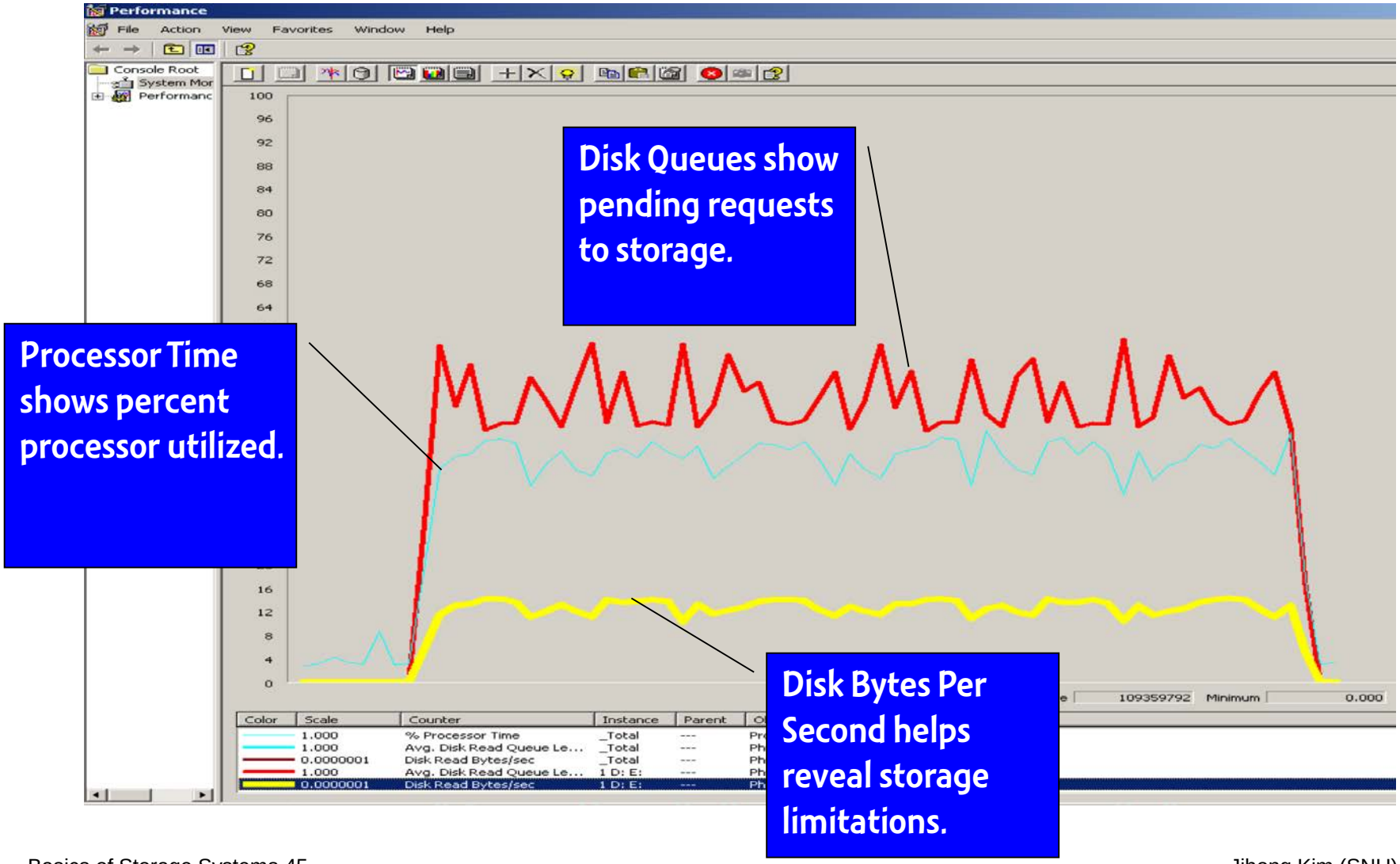
Real Application Testing

- Testing with the actual application is the best way to measure storage performance.
  - Production-like environment that can stress storage limits is desirable
  - Operating system and application tools can help monitor storage performance
    - IOStat in Linux
    - Windows Performance Monitor

Monitoring Storage Performance with Windows

- Windows Performance Monitor can be used to monitor storage performance.
- Capture the following key variables over the duration of a peak processing period or test run:
  - Processor: % processor time (total and by processor).
  - Physical disk: average disk queue (total, read and write by disk/array).
  - Physical disk: disk bytes/second (total, read and write by disk array).

Example: Windows Performance Monitor



Monitoring Storage Performance with Linux

- IOStat results show read and write bytes per second:

```
Device:     r/s     w/s     rkB/s     wkB/s   avgrq-sz   avgqu-sz   await  svctm    %util
/dev/sdb    0.00  10619.39  0.00    85570.91    16.12    4636.79    43.52   0.10   101.21
/dev/sdc    0.00  10678.79  0.00    85570.91    16.07    2438.06    22.75   0.10   107.27

avg-cpu:  %user   %nice    %sys   %idle
          13.04    0.33   68.15   18.48
```

- TOP shows CPU utilization including I/O Wait.

```
load averages:  0.09,  0.04,  0.03                                          16:31:09
66 processes:  65 sleeping, 1 on cpu
CPU states: 69.2% idle, 18.9% user, 11.9% kernel, 0.0% iowait, 0.0% swap
Memory: 128M real, 4976K free, 53M swap in use, 542M swap free
```

Performance Metrics for Storage Systems

- Performance metrics and measurement methods depend on a target storage type
  - Legacy performance metrics such as write throughput and response time are not enough to understand performance of all kinds of storage systems
  - e.g., NAND Flash memory internal behavior
    - # of erasure counts
    - Extra read/write operations for garbage collection

References

- Woody Hutsell, "Storage Performance Testing, Storage Networking Industry Association," Texas Memory Systems, 2006
- Peter M. Chen and David A. Patterson, "Storage Performance-Metrics and Benchmarks," 1993
- Sohail Sarwar and Raihan Ur Rasool, "Iometer: Why, What, Where, and How?"
- Bill Pierce, "Measuring I/O Channel Performance on Windows and UNIX Systems," 2004
- http://www.iozone.org/
- http://www.thegeekstuff.com/2011/05/iozone-examples/

References