Lecture 6: Storage Devices, Metrics, RAID, I/O Benchmarks, and Busses

Prof. Fred Chong
ECS 250A Computer Architecture
Winter 1999
(Adapted from Patterson CS252 Copyright 1998 UCB)

Motivation: Who Cares About I/O?

- CPU Performance: 60% per year
- I/O system performance limited by mechanical delays (disk I/O)
  - 10% per year (I/O per sec or MB per sec)
- Amdahl's Law: system speed-up limited by the slowest part!
  - 10% IO & 10x CPU => 5x Performance (lose 50%)
  - 10% IO & 100x CPU => 10x Performance (lose 90%)
- I/O bottleneck:
  - Diminishing fraction of time in CPU
  - Diminishing value of faster CPUs

Storage System Issues

- Historical Context of Storage I/O
- Secondary and Tertiary Storage Devices
- Storage I/O Performance Measures
- Processor Interface Issues
- Redundant Arrays of Inexpensive Disks (RAID)
- ABCs of UNIX File Systems
- I/O Benchmarks
- Comparing UNIX File System Performance
- I/O Busses

I/O Systems

Processor
Cache
Memory - I/O Bus
Main Memory
I/O Controller
Disk
Graphics
Network

Technology Trends

- Disk Capacity now doubles every 18 months; before 1990 every 36 months
- Today: Processing Power Doubles Every 18 months
- Today: Memory Size Doubles Every 18 months (4X/3yr)
- Today: Disk Capacity Doubles Every 18 months
- Disk Positioning Rate (Seek + Rotate) Doubles Every Ten Years!
- The I/O GAP

Storage Technology Drivers

- Driven by the prevailing computing paradigm
  - 1950s: migration from batch to on-line processing
  - 1990s: migration to ubiquitous computing
  - computers in phones, books, cars, video cameras, ...
  - nationwide fiber optical network with wireless tails
- Effects on storage industry:
  - Embedded storage
    - smaller, cheaper, more reliable, lower power
  - Data utilities
    - high capacity, hierarchically managed storage
Historical Perspective

- **1956 IBM Ramac — early 1970s Winchester**
  - Developed for mainframe computers, proprietary interfaces
  - Steady shrink in form factor: 27 in. to 14 in.

- **1970s developments**
  - 5.25 inch floppy disk form factor
  - early emergence of industry standard disk interfaces
    - ST506, SASI, SMD, ESDI

- **Early 1980s**
  - PCs and first generation workstations

- **Mid 1980s**
  - Client/server computing
  - Centralized storage on file server
  - Mass market disk drives become a reality
    - industry standards: SC59, IPI, IDE
  - 5.25 inch drives for standalone PCs, End of proprietary interfaces

Disk History

- **1973:**
  - 1.7 Mbit/sq. in
  - 140 MBytes

- **1979:**
  - 7.7 Mbit/sq. in
  - 2,300 MBytes

Source: New York Times, 2/23/98, page C3, "Makers of disk drives crowd even more data into even smaller spaces"

Historical Perspective

- **Late 1980s/Early 1990s:**
  - Laptops, notebooks, (palmtops)
  - 3.5 inch, 2.5 inch, (1.8 inch formfactors)
  - Form factor plus capacity drives market, not so much performance
  - Recently Bandwidth improving at 40%/ year
  - Challenged by DRAM, flash RAM in PCMCIA cards
  - Still expensive, Intel promises but doesn’t deliver
  - unattractive MBytes per cubic inch
  - Optical disk fails on performance (e.g., NEXT) but finds niche (CD ROM)

MBits per square inch: DRAM as % of Disk over time

- 1974: 3 v. 1.7 Mbits/sq
- 1987: 470 v. 3000 Mb/sq

Historical Perspective

- **1989:**
  - 63 Mbit/sq in
  - 1450 Mbit/sq in
  - 3090 Mbit/sq in
  - 2300 MBytes
  - 8100 MBytes

- **1997:**
  - 1200 MBytes
  - 33528 MBytes

Source: New York Times, 2/23/98, page C3, "Makers of disk drives crowd even more data into even smaller spaces"

Alternative Data Storage Technologies: Early 1990s

<table>
<thead>
<tr>
<th>Technology</th>
<th>Cap (MB)</th>
<th>BPI (Million)</th>
<th>TPI (Kbyte/s)</th>
<th>Data Xfer Access</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional Tape:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cartridge (.25&quot;)</td>
<td>150</td>
<td>12000</td>
<td>104</td>
<td>1.2</td>
<td>92 minutes</td>
</tr>
<tr>
<td>IBM 3490 (.5&quot;)</td>
<td>600</td>
<td>22600</td>
<td>38</td>
<td>0.3</td>
<td>3000 seconds</td>
</tr>
<tr>
<td>Helical Scan Tape:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video (8mm)</td>
<td>4600</td>
<td>43100</td>
<td>1638</td>
<td>71</td>
<td>492 45 secs</td>
</tr>
<tr>
<td>DAT (4mm)</td>
<td>1300</td>
<td>61000</td>
<td>1670</td>
<td>114</td>
<td>163 20 secs</td>
</tr>
<tr>
<td>Magnetic &amp; Optical Disk:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hard Disk (5.25&quot;)</td>
<td>1200</td>
<td>33528</td>
<td>1880</td>
<td>63</td>
<td>3000 18 ms</td>
</tr>
<tr>
<td>IBM 3390 (10.5&quot;)</td>
<td>3600</td>
<td>27940</td>
<td>2235</td>
<td>62</td>
<td>4200 20 ms</td>
</tr>
<tr>
<td>Sony MO (5.25&quot;)</td>
<td>640</td>
<td>24130</td>
<td>18796</td>
<td>454</td>
<td>88 100 ms</td>
</tr>
</tbody>
</table>

Source: New York Times, 2/23/98, page C3, "Makers of disk drives crowd even more data into even smaller spaces"
**Devices: Magnetic Disks**

- **Purpose:**
  - Long-term, nonvolatile storage
  - Large, inexpensive, slow level in the storage hierarchy
- **Characteristics:**
  - Seek Time (~8 ms avg)
  - Positional latency
  - Rotational latency
- **Transfer rate**
  - About a sector per ms (5-15 MB/s)
- **Blocks**
- **Capacity**
  - Gigabytes
  - Quadruples every 3 years (aerodynamics)

**Purpose:**

- Long-term, nonvolatile storage
- Large, inexpensive, slow level in the storage hierarchy

**Characteristics:**

- Seek Time (~8 ms avg)
- Positional latency
- Rotational latency

**Transfer rate**

- About a sector per ms (5-15 MB/s)

**Capacity**

- Gigabytes
- Quadruples every 3 years (aerodynamics)

**Tape vs. Disk**

- Longitudinal tape uses same technology as hard disk; tracks its density improvements
- Disk head flies above surface, tape head lies on surface
- Disk fixed, tape removable
- Inherent cost-performance based on geometries:
  - Fixed rotating platters with gaps
  - Removable long strips wound on spool
- New technology trend:
  - Helical Scan (VCR, Camcorder, DAT)
  - Spins head at angle to tape to improve density

**Disk Device Terminology**

- Disk Latency = Queuing Time + Controller time + Seek time + Rotation time + Xfer Time

**Order of magnitude times for 4K byte transfers:**

- Seek: 8 ms or less
- Rotate: 4.2 ms @ 7200 rpm
- Xfer: 1 ms @ 7200 rpm

**Advantages of Small Formfactor Disk Drives**

- Low cost/MB
- High MB/volume
- High MB/watt
- Low cost/Actuator

**Tape vs. Disk**

- Longitudinal tape uses same technology as hard disk; tracks its density improvements
- Disk head flies above surface, tape head lies on surface
- Disk fixed, tape removable
- Inherent cost-performance based on geometries:
  - Fixed rotating platters with gaps (random access, limited area, 1 media / reader)
  - Removable long strips wound on spool (sequential access, “unlimited” length, multiple / reader)
- New technology trend:
  - Helical Scan (VCR, Camcorder, DAT)
  - Spins head at angle to tape to improve density

**Current Drawbacks to Tape**

- Tape wear out:
  - Helical 100s of passes to 1000s for longitudinal
- Head wear out:
  - 2000 hours for helical
- Both must be accounted for in economic / reliability model
- Long rewind, eject, load, spin-up times; not inherent, just no need in marketplace (so far)
- Designed for archival

**Automated Cartridge System**

- STC 4400
- 6000 x 0.8 GB 3490 tapes = 5 TBytes in 1992
- $500,000 O.E.M. Price
- 6000 x 10 GB D3 tapes = 60 TBytes in 1998
- Library of Congress: all information in the world; in 1992, ASCII of all books = 30 TB
Library vs. Storage

- Getting books today as quaint as the way I learned to program
  - punch cards, batch processing
  - wander thru shelves, anticipatory purchasing
- Cost $1 per book to check out
- $30 for a catalogue entry
- 30% of all books never checked out
- Write only journals?
- Digital library can transform campuses
- Will have lecture on getting electronic information

Relative Cost of Storage Technology—Late 1995/Early 1996

<table>
<thead>
<tr>
<th>Technology</th>
<th>5.25&quot;</th>
<th>4.3 GB</th>
<th>3.5&quot;</th>
<th>2.5&quot;</th>
<th>1.1 GB</th>
<th>5.25&quot;</th>
<th>4.6 GB</th>
<th>3.5&quot;</th>
<th>4.3 GB</th>
<th>3.5&quot;</th>
<th>2.5&quot;</th>
<th>1.1 GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magnetic Disks</td>
<td>$2129</td>
<td>$1995</td>
<td>$199</td>
<td>$899</td>
<td>$545</td>
<td>$1695</td>
<td>$1499</td>
<td>$1099</td>
<td>$999</td>
<td>$514</td>
<td>$289</td>
<td>$545</td>
</tr>
<tr>
<td>Optical Disks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$1695</td>
<td>$1499</td>
<td>$1099</td>
<td>$999</td>
<td>$514</td>
<td>$289</td>
<td>$545</td>
</tr>
<tr>
<td>PCMCIA Cards</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$1695</td>
<td>$1499</td>
<td>$1099</td>
<td>$999</td>
<td>$514</td>
<td>$289</td>
<td>$545</td>
</tr>
</tbody>
</table>

Outline

- Historical Context of Storage I/O
- Secondary and Tertiary Storage Devices
- Storage I/O Performance Measures
- Processor Interface Issues
- Redundant Arrays of Inexpensive Disks (RAID)
- ABCs of UNIX File Systems
- I/O Benchmarks
- Comparing UNIX File System Performance
- I/O Busses

Disk I/O Performance

- 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 shrink system response time 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

Response Time vs. Productivity

- 0.7sec off response saves 4.9 sec (34%) and 2.0 sec (70%) total time per transaction => greater productivity
- Another study: everyone gets more done with faster response, but novice with fast response = expert with slow
### Disk Time Example

- **Disk Parameters:**
  - Transfer size is 8K bytes
  - Advertised average seek is 12 ms
  - Disk spins at 7200 RPM
  - Transfer rate is 4 MB/sec
- Controller overhead is 2 ms
- Assume that disk is idle so no queuing delay
- What is Average Disk Access Time for a Sector?
  - Ave seek + Ave rot delay + transfer time + controller overhead
  - 12 ms + 0.5/(7200 RPM/60) + 8 KB/4 MB/sec + 2 ms
  - 12 + 4.15 + 2 + 2 = 20 ms

### Outline

- Historical Context of Storage I/O
- Secondary and Tertiary Storage Devices
- Storage I/O Performance Measures
- Processor Interface Issues
  - Redundant Arrays of Inexpensive Disks (RAID)
  - ABCs of UNIX File Systems
  - I/O Benchmarks
  - Comparing UNIX File System Performance
  - I/O Busses

- Processor Interface Issues
  - Processor interface
    - Interrupts
    - Memory mapped I/O
  - I/O Control Structures
    - Polling
    - Interrupts
    - DMA
    - I/O Controllers
    - I/O Processors
  - Capacity, Access Time, Bandwidth
  - Interconnections
    - Busses

### I/O Interface

- Independent I/O Bus
- CPU
- Memory bus
- Peripheral
  - Separate I/O instructions (in/out)

### Memory Mapped I/O

- Single Memory & I/O Bus
  - No Separate I/O Instructions
  - CPU
  - Memory
  - Interface
  - Peripheral
  - Lines distinguish between I/O and memory transfers
  - VME bus Optimistically
  - Multibus-II Nubus

### Programmed I/O (Polling)

- CPU
- Memory
- Interface
- Peripheral
- Busy wait loop not an efficient way to use the CPU
- But checks for I/O completion can be dispersed among computationally intensive code
Interrupt Driven Data Transfer

CPU

Memory

IOC

device

User program progress only halted during actual transfer

Interrupts:
- Stack replaced by shadow registers
- Handler saves registers and re-enables higher priority int’s
- Interrupt types reduced in number; handler must query interrupt controller

1000 transfers at 1 ms each:
1000 interrupts @ 2 ms per interrupt
1000 interrupt service @ 50 usec each = 0.1 CPU seconds
1000 transfers @ 100 usec + 100 usec + 50 usec...

Still far from device transfer rate of 10 MBytes/sec

Direct Memory Access

CPU

IOC

device

Memory

DMAC

Device

DMAC provides handshake signals for Peripheral Controller, and Memory Addresses and handshake signals for Memory.

Device xfer rate = 10 MBytes/sec => 0.1 x 10^6 sec/byte => 0.1 µsec/byte
=> 1000 bytes = 100 µsec
1000 transfers x 100 µsecs = 100 ms = 0.1 CPU seconds

-6

Still far from device transfer rate! 1/2 in interrupt overhead

Input/Output Processors

CPU

IOP

Mem

D1

D2

Dn

DMAC

main memory

bus

I/O

bus

CPU

issues instruction to IOP
interrupts when done

IOP Device Address

OP   Device   Address

OP   Addr   Cnt   Other

what

where
to put

data

how

special

requests

What is the relationship between processor architecture and disk industry?

- Disk industry growing rapidly, improves:
  - bandwidth 40%/yr
  - area density 60%/year, 5MB faster?
- queue + controller + seek + rotate + transfer
- Advertised average seek time benchmark much greater than average seek time in practice
- Response time vs. Bandwidth tradeoffs
- Value of faster response time:
  - 0.7sec off response saves 4.9 sec and 2.0 sec (70%) total time per transaction => greater productivity
  - everyone gets more done with faster response, but novice with fast response = expert with slow
- Processor Interface: today peripheral processors, DMA, I/O bus, interrupts

Summary
Summary: Relationship to Processor Architecture

- I/O instructions have disappeared
- Interrupt vectors have been replaced by jump tables
- Interrupt stack replaced by shadow registers
- Interrupt types reduced in number
- Caches required for processor performance cause problems for I/O
- Virtual memory frustrates DMA
- Load/store architecture at odds with atomic operations
- Stateful processors hard to context switch

Outline

- Historical Context of Storage I/O
- Secondary and Tertiary Storage Devices
- Storage I/O Performance Measures
- Processor Interface Issues
- Redundant Arrays of Inexpensive Disks (RAID)
- ABCs of UNIX File Systems
- I/O Benchmarks
- Comparing UNIX File System Performance
- I/O Busses

Network Attached Storage

Decreasing Disk Diameters

14" » 10" » 5.25" » 3.5" » 2.5" » 1.8" » 1.3" ... high bandwidth disk systems based on arrays of disks

Network provides well defined physical and logical interfaces: complete storage system!

3 Mb/s » 10Mb/s » 50 Mb/s » 100 Mb/s » 1 Gb/s » 10 Gb/s networks capable of sustaining high bandwidth transfers

Network File Services

High Performance Storage Service on a High Speed Network

OS structures supporting remote file access

Manufacturing Advantages of Disk Arrays

<table>
<thead>
<tr>
<th>Disk Product Families</th>
<th>Conventional: 4 disk designs</th>
<th>Disk Array: 1 disk design</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5&quot;</td>
<td>5.25&quot;</td>
<td>10&quot;</td>
</tr>
<tr>
<td>Low End</td>
<td>High End</td>
<td></td>
</tr>
</tbody>
</table>

Replace Small # of Large Disks with Large # of Small Disks! (1988 Disks)

<table>
<thead>
<tr>
<th>Disk Array: 1 disk design</th>
<th>IBM 3190 (K)</th>
<th>IBM 3.5&quot; 0061</th>
<th>x?/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Capacity</td>
<td>20 GBytes</td>
<td>320 MBytes</td>
<td>23 GBytes</td>
</tr>
<tr>
<td>Volume</td>
<td>97 cu. ft.</td>
<td>0.1 cu. ft.</td>
<td>17 cu. ft.</td>
</tr>
<tr>
<td>Power</td>
<td>3 KW</td>
<td>11 W</td>
<td>7 KW</td>
</tr>
<tr>
<td>Data Rate</td>
<td>15 MB/s</td>
<td>1.5 MB/s</td>
<td>120 MB/s</td>
</tr>
<tr>
<td>I/O Rate</td>
<td>600 I/Os/s</td>
<td>55 I/Os/s</td>
<td>3000 I/Os/s</td>
</tr>
<tr>
<td>MTTF</td>
<td>250 Khrs</td>
<td>50 Khrs</td>
<td>??? Khrs</td>
</tr>
<tr>
<td>Cost</td>
<td>$250K</td>
<td>$2K</td>
<td>$150K</td>
</tr>
</tbody>
</table>

Array Reliability

- Reliability of N disks = Reliability of 1 Disk + N
  - 50,000 Hours + 70 disks = 700 hours
  - Disk system MTTF: Drops from 6 years to 1 month!
- Arrays (without redundancy) too unreliable to be useful!

Hot spares support reconstruction in parallel with access: very high media availability can be achieved
Redundant Arrays of Disks

- Files are "striped" across multiple spindles
- Redundancy yields high data availability

Disks will fail
Contents reconstructed from data redundantly stored in the array
Bandwidth penalty to update

Techniques:
- Mirroring/Shadowing (high capacity cost)
- Horizontal Hamming Codes (overkill)
- Parity & Reed-Solomon Codes
  - Failure Prediction: no capacity overhead!
  - VaxSimPlus — Technique is controversial

RAID 1: Disk Mirroring/Shadowing

- Each disk is fully duplicated onto its "shadow"
  - Very high availability can be achieved
- Bandwidth sacrifice on write:
  - Logical write = two physical writes
- Reads may be optimized
- Most expensive solution: 100% capacity overhead
  - Targeted for high I/O rate, high availability environments

RAID 3: Parity Disk

- Parity computed across recovery group to protect against hard disk failures
  - 33% capacity cost for parity in this configuration
  - Wider arrays reduce capacity costs, decrease expected availability, increase reconstruction time
  - Arms logically synchronized, spindles rotationally synchronized logically a single high capacity, high transfer rate disk
  - Targeted for high bandwidth applications: Scientific, Image Processing

RAID 5+: High I/O Rate Parity

- A logical write becomes four physical I/Os
- Independent writes possible because of interleaved parity
- Redundant write: possible because of parity
  - “Q” for protection during reconstruction
- Targeted for mixed applications

Problems of Disk Arrays: Small Writes

RAID-5: Small Write Algorithm

1 Logical Write = 2 Physical Reads + 2 Physical Writes

- New data
- Old data

K.S.D. 13 14 15

Striped physical records

10010011
11001101
10010011

Logical record

D0  D1  D2  D3
P
D4  D5  D6  P
D7

D8  D9  P  D10  D11
D12  P  D13  D14  D15
D16  D17  D18  D19
D20  D21  D22  D23
P

Increasing Logical Disk Addresses

Disk Columns

Stripe

Stripe Unit

10010011
11001101
10010011

Capacity of RAID 1 = 3 drives x 400 MB = 1.2 GB

Comparison of disk drives with RAID 1 and RAID 2

RAID 1:

- Disk: 12 MB
- I/O Rate: 10 MB/sec
- Controller: 100 MB/sec
- Average Cost: $5,000

RAID 2:

- Disk: 12 MB
- I/O Rate: 10 MB/sec
- Controller: 100 MB/sec
- Average Cost: $5,000

RAID 5:

- Disk: 12 MB
- I/O Rate: 10 MB/sec
- Controller: 100 MB/sec
- Average Cost: $5,000
Subsystem Organization

- Host
- Host adapter
- Array controller
- Single board disk controller
- Single board disk controller
- Single board disk controller
- Single board disk controller
- Striping software off-loaded to array controller
- No applications modifications
- No reduction of host performance
- Often piggy-backed in small format devices

System Availability: Orthogonal RAIDs

- Array controller
- System availability

- Fully redundant
- End to end data integrity
- Internal parity protected data paths
- Data recovery group
- Redundant support components: fans, power supplies, controller, cables
- Goal: No single points of failure
- Recovery group

System-Level Availability

- Host
- I/O controller
- Array controller
- Fully dual redundant
- End to end data integrity

Summary: Redundant Arrays of Disks (RAID) Techniques

- Disk Mirroring, Shadowing (RAID 1)
  - Each disk is fully duplicated onto its "shadow"
  - Logical write = two physical writes
  - 100% capacity overhead

- Parity Data Bandwidth Array (RAID 3)
  - Parity computed horizontally
  - Logically a single high data bw disk

- High I/O Rate Parity Array (RAID 5)
  - Interleaved parity blocks
  - Independent reads and writes
  - Logical write = 2 reads + 2 writes
  - Parity + Reed-Solomon codes

Outline

- Historical Context of Storage I/O
- Secondary and Tertiary Storage Devices
- Storage I/O Performance Measures
- Processor Interface Issues
- Redundant Arrays of Inexpensive Disks (RAID)
- ABCs of UNIX File Systems
- I/O Benchmarks
- Comparing UNIX File System Performance
- I/O Buses

ABCs of UNIX File Systems

- Key Issues
  - File vs. Raw I/O
  - File Cache Size Policy
  - Write Policy
  - Local vs. Server Disk
  - File vs. Raw:
    - File system access is the norm: standard policies apply
    - Raw: alternate I/O system to avoid file system, used by data bases
  - File Cache Size Policy
    - % of main memory dedicated to file cache is fixed at system generation (e.g., 10%)
    - % of main memory for file cache varies depending on amount of file I/O (e.g., up to 80%)
**ABCs of UNIX File Systems**

- **Write Policy**
  - File storage should be permanent; either write immediately or flush file cache after fixed period (e.g., 30 seconds)
  - Write Through with Write Buffer
  - Write Back
- Write Buffer often confused with Write Back
  - Write Through with Write Buffer, all writes go to disk
  - Write Through with Write Buffer, writes are asynchronous, so processor doesn’t have to wait for disk write
  - Write Buffer will combine multiple writes to same page; hence can be called Write Cancellation

- **Local vs. Server**
  - Unix file systems have historically had different policies (and even file systems) for local client vs. remote server
  - NFS local disk allows 30 second delay to flush writes
  - NFS server disk writes through to disk on file close
  - Cache coherency problem if allow clients to have file caches in addition to server file cache
  - NFS just writes through on file close
  - Stateless protocol: periodically get new copies of file blocks
  - Other file systems use cache coherency with write back to check state and selectively invalidate or update

**Network File Systems**

- **AUSPEX NS5000 File Server**
  - Special hardware/software architecture for high performance NFS I/O
  - Functional multiprocessing

- **AUSPEX Software Architecture**
  - Limited control interfaces
  - Primary data flow
  - Primary control flow

**Typical File Server Architecture**

- Limits to performance: data copying
  - Read data staged from device to primary memory
  - Copy again into network packet templates
  - Copy yet again to network interface
  - No specialization for fast processing between network and disk

**Page 10**
Berkeley RAID-II Disk Array File Server

I/O Benchmarks

- For better or worse, benchmarks shape a field.
  Processor benchmarks classically aimed at response time for fixed sized problem.
  I/O benchmarks typically measure throughput, possibly with upper limit on response times (or 50% of response times).
- What if fix problem size, given 60%/year increase in DRAM capacity?

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Size of Data</th>
<th>% Time I/O</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Stones</td>
<td>1 MB</td>
<td>26%</td>
<td>1990</td>
</tr>
<tr>
<td>Andrew</td>
<td>4.5 MB</td>
<td>4%</td>
<td>1988</td>
</tr>
</tbody>
</table>

- Not much time in I/O
- Not measuring disk (or even main memory)

I/O Benchmarks: Transaction Processing

- Transaction Processing (TP) (or On-line TP=OLTP)
  - Changes to a large body of shared information from many terminals, with the TP system guaranteeing proper behavior on a failure.
  - If a bank's computer fails when a customer withdraws money, the TP system would guarantee that the account is debited if the customer received the money and that the account is unchanged if the money was not received.
- Airlines reservation systems & banks use TP.
- Atomic transactions makes this work.
- Each transaction => 2 to 10 disk I/Os & 5,000 and 20,000 CPU instructions per disk I/O.
  - Efficiency of TP SW & avoiding disks accesses by keeping information in main memory.
- Classic metric is Transactions Per Second (TPS).
  - Under what workload? How machine configured?

I/O Benchmarks: TP by Anon et. al

- Proposed 3 standard tests to characterize commercial OLTP.
  - TP1:OLTP test, Debit/Credit, simulates ATMs (TP1).
  - Batch sort.
  - Batch scan.
- Debit/Credit:
  - One type of transaction: 100 bytes each.
  - Recorded 3 places: account file, branch file, teller file + events recorded in history file (50 days).
  - 15% requests for different branches.
  - Under what conditions, how report results?
**I/O Benchmarks: TP1 by Anon et al.**

- **DebitCredit Scalability:** size of account, branch, teller, history function of throughput
  - TPS  Number of ATMs Account-file size
  - 10  1,000  0.1 GB
  - 100 10,000 1.0 GB
  - 1,000 100,000 10.0 GB
  - 10,000 1,000,000 100.0 GB
  - Each input TPS => 100,000 account records, 10 branches, 100 ATMs
  - Accounts must grow since a person is not likely to use the bank more frequently just because the bank has a faster computer!

- **Response time:** 95% transactions take ≤ 1 second

- **Configuration control:** just report price (initial purchase price + 5 year maintenance = cost of ownership)
  - By publishing, in public domain

---

**I/O Benchmarks: Old TPC Benchmarks**

- **TPC-A:** Revised version of TP1/DebitCredit
  - Arrivals: Random (TPC) vs. uniform (TP1)
  - Terminals: Smart vs. dumb (affects instruction path length)
  - ATM scaling: 10 terminals per TPS vs. 100
  - Branch scaling: 1 branch record per TPS vs. 10
  - Response time constraint: 90% ≤ 2 seconds vs. 95% ≤ 1
  - Full disclosure, approved by TPC
  - Complete TPS vs. response time plots vs. single point

- **TPC-B:** Same as TPC-A but without terminals—batch processing of requests
  - Response time makes no sense: plots tps vs. residence time (time of transaction resides in system)

- These have been withdrawn as benchmarks

---

**I/O Benchmarks: TPC-C Complex OLTP**

- Models a wholesale supplier managing orders
- Order-entry conceptual model for benchmark
- Workload = 5 transaction types
- Users and database scale linearly with throughput
- Defines full-screen end-user interface
- Metrics: new-order rate (tpmC) and price/performance ($/tpmC)
- Approved July 1992

---

**I/O Benchmarks: TPC-D Complex Decision Support Workload**

- OLTP: business operation
- Decision support: business analysis (historical)
- Workload = 17 adhoc transactions
  - e.g., Impact on revenue of eliminating company-wide discount?
- Synthetic generator of data
- Size determined by Scale Factor:
  - 100 GB, 300 GB, 1 TB, 3 TB, 10 TB
- Metrics: “Queries per Gigabyte Hour” Power (QppG@Size) = 3600 x SF / Geo. Mean of queries Throughput (Qthd@Size) = 17 x SF / (time/3600)
- Price/Performance ($/QphD@Size) = $ / Geo. mean(QppD@Size, QthD@Size)
- Report time to load database (indices, stats) too
- Approved April 1995

---

**I/O Benchmarks: TPC-W Transactional Web Benchmark**

- Represent any business (retail store, software distribution, airline reservation, electronic stock trades, etc.) that markets and sells over the Internet/Intranet
- Measure systems supporting users browsing, ordering, and conducting transaction oriented business activities.
- Security (including user authentication and data encryption) and dynamic page generation are important
- Before: processing of customer order by terminal operator working on LAN connected to database system
- Today: customer accesses company site over Internet connection, browses both static and dynamically generated Web pages, and searches the database for product or customer information. Customer also initiates, finalizes and checks on product orders and deliveries.
- Started 1/97; hope to release Fall, 1998
### TPC-C Performance tpm(c)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Config</th>
<th>tpmC</th>
<th>$/tpmC</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IBM RS/6000 SP (12 node x 8-way)</td>
<td>57,033.80</td>
<td>$147.80</td>
<td>Oracle8 8.0.4</td>
</tr>
<tr>
<td>2</td>
<td>HP HP 9000 V2210 (16-way)</td>
<td>52,177.80</td>
<td>$117.17</td>
<td>Sybase ASE</td>
</tr>
<tr>
<td>3</td>
<td>Sun Ultra E6000 c (2 node x 2-way)</td>
<td>51,671.62</td>
<td>$134.46</td>
<td>Oracle8 6.0.3</td>
</tr>
<tr>
<td>4</td>
<td>HP HP 9000 V2220 (16-way)</td>
<td>39,469.47</td>
<td>$94.18</td>
<td>Sybase ASE</td>
</tr>
<tr>
<td>5</td>
<td>Fujitsu GRANPOWER 7000 Model 800</td>
<td>34,116.83</td>
<td>$57,603.00</td>
<td>Oracle8</td>
</tr>
<tr>
<td>6</td>
<td>Sun Ultra E6000 c (24-way)</td>
<td>31,147.04</td>
<td>$94.18</td>
<td>Sybase ASE</td>
</tr>
<tr>
<td>7</td>
<td>Digital Alpha/S400 (4 node x 8-way)</td>
<td>30,390.00</td>
<td>$305.00</td>
<td>Oracle7 V7.3</td>
</tr>
<tr>
<td>8</td>
<td>Digital Origin/2000 Server c (2-way)</td>
<td>25,206.20</td>
<td>$139.04</td>
<td>INFORMIX</td>
</tr>
<tr>
<td>9</td>
<td>IBM AS/400e Server (12-way)</td>
<td>25,149.75</td>
<td>$128.00</td>
<td>DB2</td>
</tr>
<tr>
<td>10</td>
<td>Digital Alpha/S400 5/25 (10-way)</td>
<td>24,577.00</td>
<td>$110.48</td>
<td>Sybase SQL</td>
</tr>
</tbody>
</table>

### TPC-C Price/Performance $/tpm(c)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Config</th>
<th>Qppd</th>
<th>QthD</th>
<th>$/QphD</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Acer AcerAltos 1900Pro4</td>
<td>11,972.07</td>
<td>$27.25</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Dell PowerEdge 6100 c/c</td>
<td>10,984.07</td>
<td>$29.55</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Compaq ProLiant 5500 c/c</td>
<td>10,526.90</td>
<td>$33.37</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ALR Revolution 6x6 c/c</td>
<td>13,089.30</td>
<td>$35.44</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>HP NetServer LX Pro</td>
<td>10,505.97</td>
<td>$35.82</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Fujitsu teamserver M791</td>
<td>13,391.13</td>
<td>$37.62</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Fujitsu GRANPOWER 5000 Model 670</td>
<td>13,391.13</td>
<td>$37.62</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Unisys Aquanta HS/6 c/c</td>
<td>13,089.30</td>
<td>$37.96</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Compaq ProLiant 7000 c/c</td>
<td>11,055.70</td>
<td>$39.25</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Unisys Aquanta HS/6 c/c</td>
<td>12,026.07</td>
<td>$39.39</td>
<td>M/S SQL 6.5</td>
<td></td>
</tr>
</tbody>
</table>

### SPEC SFS/LADDIS Predecessor: NFSstones

- NFSStones: synthetic benchmark that generates series of NFS requests from single client to test server: reads, writes, & commands & file sizes from other studies
  - Problem: 1 client could not always stress server
  - Files and block sizes not realistic
  - Clients had to run SunOS
1993 Attempt by NFS companies to agree on standard benchmark: Legato, Auspex, Data General, DEC, Interphase, Sun. Like NFSstones but
- Run on multiple clients & networks (to prevent bottlenecks)
- Reads: 50% full block & 15% partial blocks
- Writes: 50% full block & 50% partial blocks
- Average response time: 50 ms
- Scaling: for every 100 NFS ops/sec, increase capacity 1GB

Results: plot of server load (throughput) vs. response time & number of users

Assumes: 1 user => 10 NFS ops/sec

Example SPEC SFS Result: DEC Alpha
- 200 MHz 21064: 8KI + 8KD + 2MB L2; 512 MB: 1 Gigaswitch
- DEC OSF1 v2.0
- 4 FDDI networks; 32 NFS Daemons, 24 GB file size
- 88 Disks, 16 controllers, 84 file systems

Willy

UNIX File System Benchmark that gives insight into I/O system behavior (Chen and Patterson, 1993)
- Self scaling to automatically explore system size
- Examines five parameters
  - Unique bytes touched: data size; locality via LRU
  - Percentage of reads: %reads = 1 – % writes; typically 50%
  - Average I/O Request size: Bernoulli, C=1
  - Percentage sequential requests: typically 50%
  - Number of processes: concurrency of workload (number of processes issuing I/O requests)
- Fix four parameters while vary one parameter
- Searches space to find high throughput

Example Willy: DS 5000

<table>
<thead>
<tr>
<th></th>
<th>Sprite</th>
<th>Ultrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Access Size</td>
<td>32 KB</td>
<td>13 KB</td>
</tr>
<tr>
<td>Data touched (file cache)</td>
<td>2MB, 15 MB</td>
<td>2 MB</td>
</tr>
<tr>
<td>Data touched (disk)</td>
<td>36 MB</td>
<td>6 MB</td>
</tr>
<tr>
<td>% reads ≥ 50%, % sequential ≥ 50%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DS 5000 32 MB memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ultrix: Fixed File Cache Size, Write through</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprite: Dynamic File Cache Size, Write back (Write cancelling)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sprite’s Log Structured File System

Large file caches effective in reducing disk reads
Disk traffic likely to be dominated by writes
Write-Optimized File System
- Only representation on disk is log
- Stream out files, directories, maps without seeks

Advantages:
- Speed
- Striping across several disks
- Fast recovery
- Simple, Scalable
- Versioning

Problems:
- Random access retrieval
- Log wrap
- Disk space utilization

Willy: DS 5000 Number Bytes Touched

Log Structured File System: effective write cache of LFS much smaller (5-6 MB) than read cache (20 MB)
- Reads cached while writes are not ≥ 3 plateaus
Summary: I/O Benchmarks

• Scaling to track technological change
• TPC: price performance as normalizing configuration feature
• Auditing to ensure no foul play
• Throughput with restricted response time is normal measure

Outline

• Historical Context of Storage I/O
• Secondary and Tertiary Storage Devices
• Storage I/O Performance Measures
• Processor Interface Issues
• A Little Queuing Theory
• Redundant Arrays of Inexpensive Disks (RAID)
• ABCs of UNIX File Systems
• I/O Benchmarks
• Comparing UNIX File System Performance
• I/O Busses

Interconnect Trends

• Interconnect = glue that interfaces computer system components
• High speed hardware interfaces + logical protocols
• Networks, channels, backplanes

<table>
<thead>
<tr>
<th>Distance</th>
<th>Network</th>
<th>Channel</th>
<th>Backplane</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;1000 m</td>
<td>high</td>
<td>low</td>
<td>Byte Parity</td>
</tr>
<tr>
<td>10 - 100 m</td>
<td>medium</td>
<td>medium</td>
<td>Byte Parity</td>
</tr>
<tr>
<td>1 m</td>
<td>low</td>
<td>high</td>
<td>Extensive CRC</td>
</tr>
</tbody>
</table>

Interconnect

• Two generic types of busses:
  – I/O busses: lengthy, many types of devices connected, wide range in data bandwidth, and follow a bus standard (sometimes called a "channel")
  – CPU-memory busses: high speed, matched to the memory system to maximize memory-CPU bandwidth, single device (sometimes called a "backplane")
• To lower costs, low cost (older) systems combine together
• Bus transaction
  – Sending address & receiving or sending data

Backplane Architectures

Bus-Based Interconnect

• Bus: a shared communication link between subsystems
  – Low cost: a single set of wires is shared multiple ways
  – Versatility: Easy to add new devices & peripherals may even be ported between computers using common bus
• Disadvantage
  – A communication bottleneck, possibly limiting the maximum I/O throughput
• Bus speed is limited by physical factors
  – the bus length
  – the number of devices (and, hence, bus loading)
  – these physical limits prevent arbitrary bus speedup.
Bus Protocols

- **Master**: has ability to control the bus, initiates transaction
- **Slave**: module activated by the transaction

Bus Communication Protocol: specification of sequence of events and timing requirements in transferring information.

Asynchronous Bus Transfers: control lines (req., ack.) serve to orchestrate sequencing

Synchronous Bus Transfers: sequence relative to common clock

Multibus: 20 address, 16 data, 5 control, 50ns Pause

**Bus Master**: can control the bus, initiates transactions

**Bus Slave**: activated by the transaction

**Bus Communication Protocol**: specifies sequence of events and timing requirements in transferring information.

Asynchronous Bus Transfers: control lines (req., ack.) serve to orchestrate sequencing

Synchronous Bus Transfers: sequence relative to common clock

**Bus Options**

- **Option**: High performance, Low cost
- **Bus width**: Separate address & data lines, Multiplex address & data lines
- **Data width**: Wider is faster (e.g., 32 bits), Narrower is cheaper (e.g., 8 bits)
- **Transfer size**: Multiple words has less bus overhead, Single-word transfer is simpler
- **Bus masters**: Multiple (requires arbitration), Single master (no arbitration)
- **Split transaction?**: Yes—separate Request and Reply packets gets higher bandwidth (needs multiple masters), No—continuous connection is cheaper and has lower latency
- **Clocking**: Synchronous, Asynchronous

**Write Transaction**

1. Master asserts address
2. Slave asserts data
3. Wait a specified amount of time for slaves to decode target
4. Master asserts request line
5. Slave asserts ack, indicating data received
6. Master releases req
7. Slave releases ack

**Read Transaction**

1. Master has obtained control and asserts address, direction, data
2. Waits a specified amount of time for slaves to decode target
3. Master asserts request line
4. Slave asserts ack, indicating ready to transmit data
5. Master releases req, data received
6. Slave releases ack

**Bus Arbitration**

- **Parallel (Centralized) Arbitration**
  - **Bus Request**
  - **Bus Grant**

- **Serial Arbitration (daisy chaining)**
  - **Bus Request**
  - **Bus Grant**

**Asynchronous Handshake**

- **Write Transaction**
  - **Address**
  - **Data**
  - **Read**
  - **Wait**
  - **Req.**
  - **Ack.**

- **Read Transaction**
  - **Address**
  - **Data**
  - **Read**
  - **Req**
  - **Ack**

- **Bus Options**
  - **Option**: High performance, Low cost
  - **Bus width**: Separate address & data lines, Multiplex address & data lines
  - **Data width**: Wider is faster (e.g., 32 bits), Narrower is cheaper (e.g., 8 bits)
  - **Transfer size**: Multiple words has less bus overhead, Single-word transfer is simpler
  - **Bus masters**: Multiple (requires arbitration), Single master (no arbitration)
  - **Split transaction?**: Yes—separate Request and Reply packets gets higher bandwidth (needs multiple masters), No—continuous connection is cheaper and has lower latency
  - **Clocking**: Synchronous, Asynchronous
### SCSI: Small Computer System Interface

- **Clock rate:** 5 MHz / 10 MHz (fast) / 20 MHz (ultra)
- **Width:** n = 8 bits / 16 bits (wide); up to n – 1 devices to communicate on a bus or “string”
- **Devices can be slave ("target") or master("initiator")**
- **SCSI protocol:** a series of “phases”, during which specific actions are taken by the controller and the SCSI disks
  - **Bus Free:** No device is currently accessing the bus
  - **Arbitration:** When the SCSI bus goes free, multiple devices may request (arbitrate for) the bus; fixed priority by address
  - **Selection:** informs the target that it will participate (Reselection if disconnected)
  - **Command:** the initiator reads the SCSI command bytes from host memory and sends them to the target
  - **Data Transfer:** data in or out, initiator: target
  - **Message Phase:** message in or out, initiator: target (identify, save/restore data pointer, disconnect, command complete)
  - **Status Phase:** target, just before command complete

### SCSI “Bus”: Channel Architecture

![SCSI Bus Architecture Diagram](image)

### 1993 I/O Bus Survey (P&H, 2nd Ed)

<table>
<thead>
<tr>
<th>Bus</th>
<th>Originator</th>
<th>SBus</th>
<th>TurboChannel</th>
<th>MicroChannel</th>
<th>PCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Originator</td>
<td>Sun</td>
<td>DEC</td>
<td>IBM</td>
<td>Intel</td>
<td></td>
</tr>
<tr>
<td>Clock Rate (MHz)</td>
<td>16-25</td>
<td>12.5-25</td>
<td>async</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Addressing</td>
<td>Virtual</td>
<td>Physical</td>
<td>Physical</td>
<td>Physical</td>
<td></td>
</tr>
<tr>
<td>Data Sizes (bits)</td>
<td>8,16,32</td>
<td>8,16,24,32</td>
<td>8,16,24,32,64</td>
<td>8,16,24,32,64</td>
<td></td>
</tr>
<tr>
<td>Master</td>
<td>Multi</td>
<td>Single</td>
<td>Multi</td>
<td>Multi</td>
<td></td>
</tr>
<tr>
<td>Arbitration</td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
<td></td>
</tr>
<tr>
<td>32 bit read (MB/s)</td>
<td>33</td>
<td>25</td>
<td>20</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>Peak (MB/s)</td>
<td>89</td>
<td>64</td>
<td>75</td>
<td>111 (222)</td>
<td></td>
</tr>
<tr>
<td>Max Power (W)</td>
<td>16</td>
<td>26</td>
<td>13</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>

### 1993 MP Server Memory Bus Survey

<table>
<thead>
<tr>
<th>Bus</th>
<th>Summit</th>
<th>Challenge</th>
<th>XDBus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Originator</td>
<td>HP</td>
<td>SGI</td>
<td>Sun</td>
</tr>
<tr>
<td>Clock Rate (MHz)</td>
<td>60</td>
<td>68</td>
<td>65</td>
</tr>
<tr>
<td>Split transaction?</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes?</td>
</tr>
<tr>
<td>Address lines</td>
<td>48</td>
<td>40</td>
<td>??</td>
</tr>
<tr>
<td>Data lines</td>
<td>128</td>
<td>256</td>
<td>144 (parity)</td>
</tr>
<tr>
<td>Data Sizes (bits)</td>
<td>512</td>
<td>1024</td>
<td>512</td>
</tr>
<tr>
<td>Clocks/transfer</td>
<td>4</td>
<td>5</td>
<td>4?</td>
</tr>
<tr>
<td>Peak (MB/s)</td>
<td>960</td>
<td>1200</td>
<td>1056</td>
</tr>
<tr>
<td>Master</td>
<td>Multi</td>
<td>Multi</td>
<td>Multi</td>
</tr>
<tr>
<td>Arbitration</td>
<td>Central</td>
<td>Central</td>
<td>Central</td>
</tr>
<tr>
<td>Addressing</td>
<td>Physical</td>
<td>Physical</td>
<td>Physical</td>
</tr>
<tr>
<td>Slots</td>
<td>16</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Busses/system</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Length</td>
<td>13 inches</td>
<td>127 inches</td>
<td>17 inches</td>
</tr>
</tbody>
</table>

### Summary: I/O Benchmarks

- **Scaling to track technological change**
- **TPC:** price performance as normalizing configuration feature
- **Auditing to ensure no foul play**
- **Throughput with restricted response time is normal measure**