Chapters 6
Interfacing Processors and Peripherals

The Big Picture: Where are We Now?

- Today's Topic: I/O Systems
Main components of Intel Chipset: Pentium III

- Northbridge: a DMA controller, connecting the processor to memory, the AGP graphic bus, and the south bridge chip.

- Southbridge: I/O
  - PCI bus
  - Disk controllers
  - USB controllers
  - Audio
  - Serial I/O
  - Interrupt controller
  - Timers

- DMA gives external device ability to access memory directly: much lower overhead than having processor request one word at a time.

- Issue: Cache coherence:
  - What if I/O devices write data that is currently in processor cache?
    - The processor may never see new data!
  - Solutions:
    - Flush cache on every I/O operation (expensive)
    - Have hardware invalidate cache lines (remember 'Coherence' cache misses?)

- Northbridge: a DMA controller, connecting the processor to memory, the AGP graphic bus, and the south bridge chip.

- Southbridge: I/O
  - PCI bus
  - Disk controllers
  - USB controllers
  - Audio
  - Serial I/O
  - Interrupt controller
  - Timers

Main components of Intel Chipset: Pentium 4

- System Bus (“Front Side Bus”): 64 bits x 400, 533, 800 MHz

- Gbit Ethernet: 125 MB/s

- Hub bus: 8 bits x 266 MHz

- 2 Serial ATA: 150 MB/s

- 10/100 Mbit Ethernet: 1.25 - 12.5 MB/s

- Parallel ATA: 100 MB/s

- 8 USB: 60 MB/s

- 1 PCI: 32b x 33 MHz
### Table of Contents

- I/O Performance Measures
- Queueing Theory
- Magnetic Disks – Example
- Other Measures
  - Reliability and Availability
  - RAID (Redundant Array of Inexpensive Disks) - Example
- Bus and Bus Arbitration

### I/O Performance Measures

#### I/O System Design Issues
- Performance
- Expandability
- Resilience in the face of failure

![Diagram of I/O System Design](image-url)
I/O System Performance

- I/O System performance depends on many aspects of the system ("limited by weakest link in the chain"):
  - The CPU
  - The memory system:
    - Internal and external caches
    - Main Memory
  - The underlying interconnection (buses)
  - The I/O controller
  - The I/O device
  - The speed of the I/O software (Operating System)
  - The efficiency of the software’s use of the I/O devices

- Two common performance metrics:
  - Throughput: I/O bandwidth
  - Response time: Latency
Throughput = 3 patients/25 minutes = 0.12 patients/second
Response time = (10 + 8 + 10)/3 = 9.33 seconds
Service time = (10 + 8 + 5)/3 = 7.67 seconds
Queueing delay = (0 + 0 + 5)/3 = 1.67 seconds
Utilization = 23 minutes/25 minutes = 92%

Table of Contents

° I/O Performance Measures
° Queueing Theory
° Magnetic Disks – Example
° Other Measures
  • Reliability and Availability
  • RAID (Redundant Array of Inexpensive Disks) - Example
° Bus and Bus Arbitration
Simple Producer-Server Model

Throughput:
- The number of tasks completed by the server in unit time
- In order to get the highest possible throughput:
  - The server should never be idle
  - The queue should never be empty

Response time:
- Begins when a task is placed in the queue
- Ends when it is completed by the server
- In order to minimize the response time:
  - The queue should be empty
  - The server will be idle

Throughput versus Respond Time
Throughput Enhancement

In general throughput can be improved by:
- Throwing more hardware at the problem
- Reduces load-related latency

Response time is much harder to reduce:
- Ultimately it is limited by the speed of light (but we’re far from it)

Queueing Model

Queue: a useful abstraction of a shared resource
- A cpu in your PC
- The memory bus in your PC
- An Ethernet line in a local area network
- Wireless medium in a wireless LAN
- An output port in a network router
- A disk in a file server
- A semaphore in a multithreaded (multi-process) program

A queueing system consists of
- Customers arrive at unpredictable times
- Each customer has a service requirement
- Specified number of servers and scheduling discipline
Throughput = 3/25 patients/minute (X)

Utilization = 23/25 (U)

Average service time = (10 + 8 + 5)/3 minutes (Ts)

Average waiting time = (0 + 0 + 5)/3 minutes (Tq)

Average number of customers in queue = (0*20 + 1*5)/25 = 1/5 (Nq)

Average number of customers in service = (0*2 + 1*23)/25 = 23/25 (Ns)

Customer incoming rate = 3/25 patients/minute (λ)

Service rate = 3/23 patients/minute (µ or 1/Ts)

Stability condition: λ < µ

λ/µ = λ/µ = 3/25 / 3/23 = 23/25 (U or ρ)

N = Ns = 1/5

Nq = 1/5

Ns = 23/25

λTq = 3/25 * 5/3 = 1/5

λTs = 3/25 * 23/3 = 23/25

Little’s Law (1961)

Queue Metrics

X = throughput (i.e., departure rate)

U = server utilization (i.e., fraction of time the server is busy)

T = average time a customer spends in the queue (i.e., average waiting time + average service time)

Time in service (Ts) vs in queue (Tq): T = Tq + Ts

N = average number of customers in the queue

Number in service (Ns) vs number in queue (Nq): N = Ns + Nq

Rules for all queues

Stability condition: λ < µ (for infinite buffer and infinite population system)

Number vs Time: Little’s Law (1961) if no creation/no lost of jobs

- mean # in system = arrival rate x mean response time (N = λT or Nq = λTq)

T: response time

Ts: service time per job = 1/µ

Tq: waiting time
Analysis of a Single Queue

- Interests in Queueing Systems
  - Traffic Intensity ($\lambda/\mu$)
  - Server Utilization
  - Probability that $N$ customers are in the system at time $t$

- Relationships (Little’s Law)
  - $N = \lambda T$ \hspace{0.5cm} ($N$: avg # in the system)
  - $N_q = \lambda T_q$ \hspace{0.5cm} ($N_q$: avg # in queue)
  - $T = T_q + 1/\mu$ \hspace{0.5cm} ($T$: avg waiting time in sys.)
    \hspace{1cm} ($T_q$: avg waiting time in queue)

What is the average response time and average number of patients?

To answer this, we need to answer what is the chance that the hospital has 10 patients

\[ N = 0*P_0 + 1*P_1 + 2*P_2 + \ldots \]
\[ T = N/\lambda = (0*P_0 + 1*P_1 + 2*P_2 + \ldots)/3/25 \]

To know $p_k$, consider the following state transition diagram and “load balance equation”
M/M/1: Example

Engineers use one terminal for serious computation. Arrival pattern is Poisson with a mean of 10 people each day. The distribution of time spent at a terminal is exponential with a mean of 30 minutes. Engineers complain about the terminal service but the manager finds the terminal is in use only 5 hours out of an 8-hour working day.

- \( \lambda = 10 \text{ persons/day} \times 1 \text{ day/8hours} \times 1 \text{ hour/60minutes} = 1/48 \text{ persons/min} \)
- \( \mu = 1/30 \text{ persons/min} \)
- \( \rho = \frac{\lambda}{\mu} = 30/48 = 0.625 \) : server utilization
- \( N = \rho/(1-\rho) = 1.667 \text{ persons} \) : average number of customers in system
- \( N_q = 1.667 - 0.625 = 1.042 \) : average number of customers in queue
- \( T = 80 \text{ minutes} \) : average time spent in the system
- \( T_q = 80 - 30 = 50 \text{ minutes} \) : average time wasted in queue

\[ T_q = N_q / \lambda \quad \text{Little’s Law can be applied in the subsystem} \]

Table of Contents

- I/O Performance Measures
- Queueing Theory
- Magnetic Disks – Example
- Other Measures
  - Reliability and Availability
  - RAID (Redundant Array of Inexpensive Disks) - Example
- Bus and Bus Arbitration
Magnetic Disk

° Purpose:
  • Long term, nonvolatile storage
  • Large, inexpensive, and slow
  • Lowest level in the memory hierarchy

° Two major types:
  • Floppy disk
  • Hard disk

° Both types of disks:
  • Rely on a rotating platter coated with a magnetic surface
  • Use a moveable read/write head to access the disk

° Advantages of hard disks over floppy disks:
  • Platters are more rigid (metal or glass) so they can be larger
  • Higher density because it can be controlled more precisely
  • Higher data rate because it spins faster
  • Can incorporate more than one platter

Organization of a Hard Magnetic Disk

° Typical numbers (depending on the disk size):
  • 500 to 2,000 tracks per surface
  • 32 to 128 sectors per track
    - A sector is the smallest unit that can be read or written

° Traditionally all tracks have the same number of sectors:
  • Constant bit density: record more sectors on the outer tracks
  • Recently relaxed: constant bit size, speed varies with track location
Magnetic Disk Characteristic

° Cylinder: all the tracks under the head at a given point on all surface

° Read/write data is a three-stage process:
  • Seek time: position the arm over the proper track
  • Rotational latency: wait for the desired sector to rotate under the read/write head
  • Transfer time: transfer a block of bits (sector) under the read-write head

° Average seek time as reported by the industry:
  • Typically in the range of 8 ms to 12 ms
  • (Sum of the time for all possible seek) / (total # of possible seeks)

° Due to locality of disk reference, actual average seek time may:
  • Only be 25% to 33% of the advertised number

Typical Numbers of a Magnetic Disk

° Rotational Latency:
  • Most disks rotate at 3,600 to 7200 RPM
  • Approximately 16 ms to 8 ms per revolution, respectively
  • An average latency to the desired information is halfway around the disk: 8 ms at 3600 RPM, 4 ms at 7200 RPM

° Transfer Time is a function of:
  • Transfer size (usually a sector): 1 KB / sector
  • Rotation speed: 3600 RPM to 7200 RPM
  • Recording density: bits per inch on a track
  • Diameter typical diameter ranges from 2.5 to 5.25 in
  • Typical values: 2 to 12 MB per second
**Disk I/O Performance**

\[ \text{Disk Access Time} = \text{Seek time} + \text{Rotational Latency} + \text{Transfer time} + \text{Controller Time} + \text{Queueing Delay} \]

**Estimating Queue Length:**
- Utilization \( U = \frac{\text{Request Rate}}{\text{Service Rate}} \)
- Mean Queue Length \( \frac{U}{1 - U} \)
- As Request Rate \( \rightarrow \) Service Rate
  - Mean Queue Length \( \rightarrow \) Infinity

**Example**

- 512 byte sector, rotate at 5400 RPM, advertised seeks is 12 ms, transfer rate is 4 GB/sec, controller overhead is 1 ms, queue idle so no service time

\[ \text{Disk Access Time} = 12 \text{ ms} + 0.5 \text{ ms} + 0.125 \text{ ms} + 1 \text{ ms} + 0 \text{ ms} \]

- If real seeks are 1/3 advertised seeks, then its 10.6 ms, with rotation delay at 50% of the time!
### Magnetic Disk Examples

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>IBM 3090</th>
<th>IBM UltraStar</th>
<th>Integral 1820</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk diameter (inches)</td>
<td>10.88</td>
<td>3.50</td>
<td>1.80</td>
</tr>
<tr>
<td>Formatted data capacity (MB)</td>
<td>22,700</td>
<td>4,300</td>
<td>21</td>
</tr>
<tr>
<td>MTTF (hours)</td>
<td>50,000</td>
<td>1,000,000</td>
<td>100,000</td>
</tr>
<tr>
<td>Number of arms/box</td>
<td>12</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Rotation speed (RPM)</td>
<td>3,600</td>
<td>7,200</td>
<td>3,800</td>
</tr>
<tr>
<td>Transfer rate (MB/sec)</td>
<td>4.2</td>
<td>9-12</td>
<td>1.9</td>
</tr>
<tr>
<td>Power/box (watts)</td>
<td>2,900</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>MB/watt</td>
<td>8</td>
<td>102</td>
<td>10.5</td>
</tr>
<tr>
<td>Volume (cubic feet)</td>
<td>97</td>
<td>0.13</td>
<td>0.02</td>
</tr>
<tr>
<td>MB/cubic feet</td>
<td>234</td>
<td>33000</td>
<td>1050</td>
</tr>
</tbody>
</table>

### Table of Contents

- I/O Performance Measures
- Queueing Theory
- Magnetic Disks – Example
- Other Measures
  - Reliability and Availability
  - RAID (Redundant Array of Inexpensive Disks) - Example
- Bus and Bus Arbitration
Reliability and Availability

- Two terms that are often confused:
  - Reliability: Is anything broken?
  - Availability: Is the system still available to the user?

- Availability can be improved by adding hardware:
  - Example: adding ECC on memory

- Reliability can only be improved by:
  - Bettering environmental conditions
  - Building more reliable components
  - Building with fewer components
    - Improve availability may come at the cost of lower reliability

Disk Arrays

- A new organization of disk storage:
  - Arrays of small and inexpensive disks
  - Increase potential throughput by having many disk drives:
    - Data is spread over multiple disk
    - Multiple accesses are made to several disks

- Reliability is lower than a single disk:
  - But availability can be improved by adding redundant disks (RAID):
    - Lost information can be reconstructed from redundant information
  - MTTR: mean time to repair is in the order of hours
  - MTTF: mean time to failure of disks is tens of years
Replace Small # of Large Disks with Large # of Small Disks! (1988 Disks)

<table>
<thead>
<tr>
<th></th>
<th>IBM 3390 (K)</th>
<th>IBM 3.5” 0061</th>
<th>x70</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Capacity</strong></td>
<td>20 GBytes</td>
<td>320 MBytes</td>
<td>23 GBytes</td>
</tr>
<tr>
<td><strong>Volume</strong></td>
<td>97 cu. ft.</td>
<td>0.1 cu. ft.</td>
<td>11 cu. ft.</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>3 KW</td>
<td>11 W</td>
<td>1 KW</td>
</tr>
<tr>
<td><strong>Data Rate</strong></td>
<td>15 MB/s</td>
<td>1.5 MB/s</td>
<td>120 MB/s</td>
</tr>
<tr>
<td><strong>I/O Rate</strong></td>
<td>600 I/Os/s</td>
<td>55 I/Os/s</td>
<td>3900 I/Os/s</td>
</tr>
<tr>
<td><strong>MTTF</strong></td>
<td>250 Khrs</td>
<td>50 Khrs</td>
<td>??? Khrs...</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>$250K</td>
<td>$2K</td>
<td>$150K</td>
</tr>
</tbody>
</table>

Disk Arrays have potential for large data and I/O rates

high MB per cu. ft., high MB per KW

reliability?

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
  
  - Capacity penalty to store it
  - 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*

---

**Redundant Arrays of Disks**  
**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*
**Redundant Arrays of Disks 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*

**Redundant Arrays of Disks RAID 5+: High I/O Rate Parity**

- A logical write becomes four physical I/Os
- Independent writes possible because of interleaved parity
- Reed-Solomon Codes ("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

Subsystem Organization

- host
  - manages interface to host, DMA
  - control, buffering, parity logic
  - physical device control
  - striping software off-loaded from host to array controller
    - no applications modifications
    - no reduction of host performance

- host adapter
- array controller
- single board disk controller
- single board disk controller
- single board disk controller
- single board disk controller

often piggy-backed in small format devices
**System Availability: Orthogonal RAIDs**

**Array Controller**

- String Controller
- String Controller
- String Controller
- String Controller
- String Controller
- String Controller
- String Controller

*Data Recovery Group:* unit of data redundancy

*Redundant Support Components:* fans, power supplies, controller, cables

*End to End Data Integrity:* internal parity protected data paths

---

**System-Level Availability**

**Host**

- I/O Controller

*Fully dual redundant*

- I/O Controller

**Array Controller**

*Goal: No Single Points of Failure*

with duplicated paths, higher performance can be obtained when there are no failures
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