Review: Major Components of a Computer

Magnetic Disk

- **Purpose**
  - Long term, nonvolatile storage
  - Lowest level in the memory hierarchy
  - Slow, large, inexpensive

- **General structure**
  - A rotating platter coated with a magnetic surface
  - A movable read/write head to access the information on the disk

- **Typical numbers**
  - 1 to 4 (1 or 2 surface) platters per disk of 1" to 5.25" in diameter (3.5" dominate in 2004)
  - Rotational speeds of 5,400 to 15,000 RPM
  - 10,000 to 50,000 tracks per surface
    - cylinder - all the tracks under the head at a given point on all surfaces
  - 100 to 500 sectors per track
    - The smallest unit that can be read/written (typically 512B)

Magnetic Disk Characteristic

- **Disk read/write components**
  1. Seek time: position the head over the proper track (3 to 14 ms avg)
     - Due to locality of disk references the actual average seek time may be only 25% to 33% of the advertised number
  2. Rotational latency: wait for the desired sector to rotate under the head (½ of 1/RPM converted to ms)
    - 0.5/5400RPM = 5.6ms to 0.5/15000RPM = 2.0ms
  3. Transfer time: transfer a block of bits (one or more sectors) under the head to the disk controller’s cache (30 to 80 MB/s are typical disk transfer rates)
    - The disk controller’s “cache” takes advantage of spatial locality in disk accesses
    - Cache transfer rates are much faster (e.g., 320 MB/s)
  4. Controller time: the overhead the disk controller imposes in performing a disk I/O access (typically < .2 ms)

Typical Disk Access Time

- The average time to read or write a 512B sector for a disk rotating at 10,000RPM with average seek time of 6ms, a 50MB/sec transfer rate, and a 0.2ms controller overhead

  \[
  \text{Avg disk read/write} = 6.0 + 0.5(10000RPM/60) + 0.5(50MB/sec) + 0.2 = 9.21ms
  \]

  If the measured average seek time is 25% of the advertised average seek time, then

  \[
  \text{Avg disk read/write} = 1.5 + 3.0 + 0.01 + 0.2 = 4.71ms
  \]

- The rotational latency is usually the largest component of the access time
### Magnetic Disk Examples (www.seagate.com)

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Seagate ST37</th>
<th>Seagate ST122</th>
<th>Seagate ST184</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk diameter (inches)</td>
<td>3.5</td>
<td>3.5</td>
<td>2.5</td>
</tr>
<tr>
<td>Capacity (GB)</td>
<td>73.4</td>
<td>200</td>
<td>40</td>
</tr>
<tr>
<td># of surfaces (heads)</td>
<td>8</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Rotation speed (RPM)</td>
<td>15,000</td>
<td>7,200</td>
<td>5,400</td>
</tr>
<tr>
<td>Transfer rate (MB/sec)</td>
<td>57-86</td>
<td>32-58</td>
<td>34</td>
</tr>
<tr>
<td>Minimum seek (ms)</td>
<td>0.2r-0.4w</td>
<td>1.0r-1.2w</td>
<td>1.5r-2.0w</td>
</tr>
<tr>
<td>Average seek (ms)</td>
<td>3.6r-3.9w</td>
<td>8.5r-9.5w</td>
<td>12r-14w</td>
</tr>
<tr>
<td>MTTF (hours@25°C)</td>
<td>1,200,000</td>
<td>600,000</td>
<td>330,000</td>
</tr>
<tr>
<td>Dimensions (inches)</td>
<td>1&quot;x4&quot;x5.8&quot;</td>
<td>1&quot;x4&quot;x5.8&quot;</td>
<td>0.4&quot;x2.7&quot;x3.9&quot;</td>
</tr>
<tr>
<td>GB/cu.inch</td>
<td>3</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>Power: op/idle/sb (watts)</td>
<td>20/12/0.2</td>
<td>12/8/1</td>
<td>2.4/1/0.4</td>
</tr>
<tr>
<td>GB/watt</td>
<td>4</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>Weight (pounds)</td>
<td>1.9</td>
<td>1.4</td>
<td>0.2</td>
</tr>
</tbody>
</table>

### Disk Latency & Bandwidth Milestones

<table>
<thead>
<tr>
<th>Interface</th>
<th>CDC When</th>
<th>SG ST41</th>
<th>SG ST16</th>
<th>SG ST39</th>
<th>SG ST37</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity (Gb)</td>
<td>8.03</td>
<td>1.4</td>
<td>4.3</td>
<td>9.1</td>
<td>73.4</td>
</tr>
<tr>
<td>Diameter (inches)</td>
<td>5.25</td>
<td>5.25</td>
<td>3.8</td>
<td>3.8</td>
<td>2.8</td>
</tr>
<tr>
<td>Transfer Rate (MB/s)</td>
<td>ST-412</td>
<td>SCsi</td>
<td>SCsi</td>
<td>SCsi</td>
<td>SCsi</td>
</tr>
<tr>
<td>Bandwidth (MB/s)</td>
<td>0.6</td>
<td>4</td>
<td>9</td>
<td>24</td>
<td>86</td>
</tr>
<tr>
<td>Latency (msec)</td>
<td>48.3</td>
<td>17.1</td>
<td>12.7</td>
<td>8.8</td>
<td>5.7</td>
</tr>
</tbody>
</table>

Patterson, CACM Vol 47, #10, 2004

- Disk latency is one average seek time plus the rotational latency.
- Disk bandwidth is the peak transfer time of formatted data from the media (not from the cache).

### Latency & Bandwidth Improvements

- In the time that the disk bandwidth doubles the latency improves by a factor of only 1.2 to 1.4

![Bandwidth Latency Graph](image)

### Aside: Media Bandwidth/Latency Demands

- Bandwidth requirements
  - High quality video
    - Digital data = (30 frames/s) × (640 x 480 pixels) × (24-b color/pixel) = 231 Mbits (227 MB/s)
  - High quality audio
    - Digital data = (44,100 audio samples/s) × (16-b audio samples) × (2 audio channels for stereo) = 1.4 Mbits (0.175 MB/s)
  - Compression reduces the bandwidth requirements considerably

- Latency issues
  - How sensitive is your eye (ear) to variations in video (audio) rates?
  - How can you ensure a constant rate of delivery?
  - How important is synchronizing the audio and video streams?
    - 15 to 20 ms early to 30 to 40 ms late is tolerable

### Dependability, Reliability, Availability

- Reliability – measured by the mean time to failure (MTTF). Service interruption is measured by mean time to repair (MTTR)
- Availability – a measure of service accomplishment
  \[ \text{Availability} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \]

- To increase MTTF, either improve the quality of the components or design the system to continue operating in the presence of faulty components
  1. Fault avoidance: preventing fault occurrence by construction
  2. Fault tolerance: using redundancy to correct or bypass faulty components (hardware)
     - Fault detection versus fault correction
     - Permanent faults versus transient faults

### RAIDs: Disk Arrays

- Redundant Array of Inexpensive Disks
  - 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 at a time
  - 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
### RAID: Level 0 (No Redundancy; Striping)

- **Multiple smaller disks as opposed to one big disk**
  - Spreading the blocks over multiple disks — striping — means that multiple blocks can be accessed in parallel increasing the performance.
  - A 4 disk system gives four times the throughput of a 1 disk system.
- **Same cost as one big disk** — assuming 4 small disks cost the same as one big disk.
- **No redundancy, so what if one disk fails?**
  - Failure of one or more disks is more likely as the number of disks in the system increases.

### RAID: Level 1 (Redundancy via Mirroring)

- **Uses twice as many disks as RAID 0** (e.g., 8 smaller disks with second set of 4 duplicating the first set) so there are always two copies of the data.
  - If a disk fails, the system just goes to the “mirror” for the data.
- **No performance improvement over RAID 0** because the same amount of data is written twice.
- **Cost of higher availability is reduced to 1/N where N is the number of disks in a protection group**.
  - If redundant disks = # of data disks, so twice the cost of one big disk.
  - Writes have to be made to both sets of disks, so writes would be only 1/2 the performance of RAID 0.
- **What if one disk fails?**
  - If a disk fails, the system just goes to the “mirror” for the data.

### RAID: Level 0+1 (Striping with Mirroring)

- **Combines the best of RAID 0 and RAID 1**, data is striped across four disks and mirrored to four disks.
  - Four times the throughput (due to striping).
  - # redundant disks = # of data disks so twice the cost of one big disk.
  - Writes have to be made to both sets of disks, so writes would be only 1/2 the performance of RAID 0.
- **What if one disk fails?**
  - If a disk fails, the system just goes to the “mirror” for the data.

### RAID: Level 3 (Bit-Interleaved Parity)

- **Cost of higher availability is reduced to 1/N but the parity is stored as blocks associated with sets of data blocks**.
  - Four times the throughput (striping).
  - # redundant disks = # of data disks so twice the cost of one big disk.
  - Supports “small reads” and “small writes” (reads and writes that go to just one (or a few) data disk in a protection group).
  - Can tolerate limited disk failure, since the data can be reconstructed.
    - The parity disk must be updated on every write, so it is a bottleneck for back-to-back writes.
- **What if one disk fails?**
  - If a disk fails, the system just goes to the “mirror” for the data.

### RAID: Level 4 (Block-Interleaved Parity)

- **Cost of higher availability still only 1/N but the parity is stored as blocks associated with sets of data blocks**.
  - Four times the throughput (striping).
  - # redundant disks = # of data disks so twice the cost of one big disk.
  - Supports “small reads” and “small writes” (reads and writes that go to just one (or a few) data disk in a protection group).
  - Can tolerate limited disk failure, since the data can be reconstructed.
Small Writes

- RAID 3 small writes
  - New D1 data
  - 3 reads and 2 writes involving all the disks

- RAID 4 small writes
  - New D1 data
  - 2 reads and 2 writes involving just two disks

RAID: Level 5 (Distributed Block-Interleaved Parity)

- Cost of higher availability still only 1/N but the parity block can be located on any of the disks so there is no single bottleneck for writes
  - Still four times the throughput (striping)
  - If redundant disks = 1 × # of protection groups
  - Supports “small reads” and “small writes” (reads and writes that go to just one (or a few) data disk in a protection group)
  - Allows multiple simultaneous writes as long as the accompanying parity blocks are not located on the same disk
- Can tolerate limited disk failure, since the data can be reconstructed

Summary

- Four components of disk access time:
  - Seek Time: advertised to be 3 to 14 ms but lower in real systems
  - Rotational Latency: 5.6 ms at 5400 RPM and 2.0 ms at 15000 RPM
  - Transfer Time: 30 to 80 MB/s
  - Controller Time: typically less than .2 ms

- RAIDS can be used to improve availability
  - RAID 0 and RAID 5 – widely used in servers, one estimate is that 80% of disks in servers are RAIDs
  - RAID 1 (mirroring) – EMC, Tandem, IBM
  - RAID 3 – Storage Concepts
  - RAID 4 – Network Appliance

- RAIDS have enough redundancy to allow continuous operation, but not hot swapping

Input and Output Devices

- I/O devices are incredibly diverse with respect to
  - Behavior – input, output or storage
  - Partner – human or machine
  - Data rate – the peak rate at which data can be transferred between the I/O device and the main memory or processor

<table>
<thead>
<tr>
<th>Device</th>
<th>Behavior</th>
<th>Partner</th>
<th>Data rate (Mb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyboard</td>
<td>input</td>
<td>human</td>
<td>0.0001</td>
</tr>
<tr>
<td>Mouse</td>
<td>input</td>
<td>human</td>
<td>0.0038</td>
</tr>
<tr>
<td>Laser printer</td>
<td>output</td>
<td>human</td>
<td>3.2000</td>
</tr>
<tr>
<td>Graphics display</td>
<td>output</td>
<td>human</td>
<td>800.0000-8000.0000</td>
</tr>
<tr>
<td>Network/LAN</td>
<td>input or output</td>
<td>machine</td>
<td>100.0000-1000.0000</td>
</tr>
<tr>
<td>Magnetic disk</td>
<td>storage</td>
<td>machine</td>
<td>240.0000-2560.0000</td>
</tr>
</tbody>
</table>
I/O Performance Measures
- I/O bandwidth (throughput) — amount of information that can be input (output) and communicated across an interconnect (e.g., a bus) to the processor/memory (I/O device) per unit time
  1. How much data can we move through the system in a certain time?
  2. How many I/O operations can we do per unit time?
- I/O response time (latency) — the total elapsed time to accomplish an input or output operation
  - An especially important performance metric in real-time systems
- Many applications require both high throughput and short response times

I/O System Performance
- Designing an I/O system to meet a set of bandwidth and/or latency constraints means
  1. Finding the weakest link in the I/O system — the component that constrains the design
    - The processor and memory system?
    - The underlying interconnection (e.g., bus)?
    - The I/O controllers?
    - The I/O devices themselves?
  2. (Re)configuring the weakest link to meet the bandwidth and/or latency requirements
  3. Determining requirements for the rest of the components and (re)configuring them to support this latency and/or bandwidth

I/O System Performance Example
- A disk workload consisting of 64KB reads and writes where the user program executes 200,000 instructions per disk I/O operation and
  - a processor that sustains 3 billion instr/s and averages 100,000 OS instructions to handle a disk I/O operation
  - a memory-I/O bus that sustains a transfer rate of 1000 MB/s
  - SCSI disk I/O controllers with a DMA transfer rate of 320 MB/s that can accommodate up to 7 disks per controller
  - disk drives with a read/write bandwidth of 75 MB/s and an average seek plus rotational latency of 6 ms

  what is the maximum sustainable I/O rate and what is the number of disks and SCSI controllers required to achieve that rate?

  Each disk I/O reads/writes 64 KB so the maximum I/O rate of the bus is
  \[ \frac{1000 \times 10^6}{64 \times 10^3} = 15,625 \text{ I/O's/s} \]

  The maximum disk I/O rate (# I/O's/sec) of the processor is
  \[ \frac{3 \times 10^9}{(200 + 100) \times 10^3} = 10,000 \text{ I/O's/sec} \]

  what is the maximum sustainable I/O rate and what is the number of disks and SCSI controllers required to achieve that rate?

  Each disk I/O reads/writes 64 KB so the maximum I/O rate of the bus is
  \[ \frac{15,625 \text{ I/O's/s}}{64 \times 10^3} = 242 \text{ I/O's/sec} \]

  what is the maximum sustainable I/O rate and what is the number of disks and SCSI controllers required to achieve that rate?
I/O System Performance Example, Con’t

So the processor is the bottleneck, not the bus
- disk drive with a read/write bandwidth of 75 MB/s and an average seek plus rotational latency of 6 ms

Disk I/O read/write time = seek + rotational time + transfer time = 6ms + 64KB/(75MB/s) = 6.9ms

Thus each disk can complete 1000ms/6.9ms or 146 I/O’s per second. To saturate the processor requires 10,000 I/O’s per second or 10,000/146 = 69 disks.

Disk transfer rate = (transfer size)/(transfer time) = 64KB/6.9ms = 9.56 MB/s

Thus 7 disks won’t saturate either the SCSI controller (with a maximum transfer rate of 320 MB/s) or the memory-I/O bus (1000 MB/s). This means we will need 69/7 or 10 SCSI controllers.

I/O System Interconnect Issues
- A bus is a shared communication link (a single set of wires used to connect multiple subsystems) that needs to support a range of devices with widely varying latencies and data transfer rates
  - Advantages
    - Versatile: new devices can be added easily and can be moved between computer systems that use the same bus standard
    - Low cost: a single set of wires is shared in multiple ways
  - Disadvantages
    - Creates a communication bottleneck: bus bandwidth limits the maximum I/O throughput
  - The maximum bus speed is largely limited by
    - The length of the bus
    - The number of devices on the bus

Bus Characteristics

- Control lines
  - Master initiates requests
  - Communicate what type of information is on the data lines
- Data lines
  - Data, addresses, and complex commands
- Bus transaction consists of
  - Master issuing the command (and address) – request
  - Slave receiving (or sending) the data – action
  - Device issuing data to the memory – output
- Defined by what the transaction does to memory
  - Input – inputs data from the I/O device to the memory
  - Output – outputs data from the memory to the I/O device

Types of Buses

- Processor-memory bus (proprietary)
  - Short and high speed
  - Matched to the memory system to maximize the memory-processor bandwidth
  - Optimized for cache block transfers
- I/O bus (industry standard, e.g., SCSI, USB, Firewire)
  - Usually is lengthy and slower
  - Needs to accommodate a wide range of I/O devices
  - Connects to the processor-memory bus or backplane bus
- Backplane bus (industry standard, e.g., ATA, PCIe Express)
  - The backplane is an interconnection structure within the chassis
  - Used as an intermediary bus connecting I/O buses to the processor-memory bus

Synchronous and Asynchronous Buses

- Synchronous bus (e.g., processor-memory buses)
  - Includes a clock in the control lines and has a fixed protocol for communication that is relative to the clock
  - Advantage: involves very little logic and can run very fast
  - Disadvantages:
    - Every device communicating on the bus must use same clock rate
    - To avoid clock skew, they cannot be long if they are fast
- Asynchronous bus (e.g., I/O buses)
  - Not clocked, so requires a handshaking protocol and additional control lines (ReadReq, Ack, DataRdy)
  - Advantage:
    - Can accommodate a wide range of devices and device speeds
    - Can be lengthened without worrying about clock skew or synchronization problems
  - Disadvantages: slow(er)
The Need for Bus Arbitration
- Multiple devices may need to use the bus at the same time so must have a way to arbitrate multiple requests
- Bus arbitration schemes usually try to balance:
  - Bus priority – the highest priority device should be serviced first
  - Fairness – even the lowest priority device should never be completely locked out from the bus
- Bus arbitration schemes can be divided into four classes
  - Daisy chain arbitration – see next slide
  - Centralized, parallel arbitration – see next-next slide
  - Distributed arbitration by self-selection – each device wanting the bus places a code indicating its identity on the bus; Each device sees all requestors; Priority scheme allows each to know if they get bus
  - Distributed arbitration by collision detection – device uses the bus when its not busy and if a collision happens (because some other device also decides to use the bus) then the device tries again again

Daisy Chain Bus Arbitration
- Advantage: simple
- Disadvantages:
  - Cannot assure fairness – a low-priority device may be locked out indefinitely
  - Slower – the daisy chain grant signal limits the bus speed

Centralized Parallel Arbitration
- Advantages: flexible, can assure fairness
- Disadvantages: more complicated arbiter hardware
- Used in essentially all processor-memory buses and in high-speed I/O buses

Bus Bandwidth Determinates
- The bandwidth of a bus is determined by:
  - Whether it is synchronous or asynchronous and the timing characteristics of the protocol used
  - The data bus width
  - Whether the bus supports block transfers or only word at a time transfers

Example: The Pentium 4’s Buses
- System Bus (“Front Side Bus”): 64b x 800 MHz (6.4GB/s), 533 MHz, or 400 MHz
- 2 serial ATAs: 150 MB/s
- 2 parallel ATA: 100 MB/s
- Hub Bus: 6b x 266 MHz
- 10/100 LAN
- 1.6GB/s 32b x 33 MHz
- Memory Controller Hub (“Northbridge”)
- Graphics output: 2.0 Gb/s
- Gbit ethernet: 0.290 Gb/s
- PCI: 32b x 33 MHz
- I/O Controller Hub (“Southbridge”)

Buses in Transition
- Companies are transitioning from synchronous, parallel, wide buses to asynchronous narrow buses
- Reflection on wires and clock skew makes it difficult to use 16 to 64 parallel wires running at a high clock rate (e.g., ~400 MHz) so companies are transitioning to buses with a few one-way wires running at a very high “clock” rate (~2 GHz)
ATA Cable Sizes
- Serial ATA cables (red) are much thinner than parallel ATA cables (green).

Communication of I/O Devices and Processor
- How the processor directs the I/O devices
  - Special I/O instructions
    - Must specify both the device and the command
  - Memory-mapped I/O
    - Portions of the high-order memory address space are assigned to each I/O device
    - Read and writes to those memory addresses are interpreted as commands to the I/O devices
    - Load/store to the I/O address space can only be done by the OS
- How the I/O device communicates with the processor
  - Polling – the processor periodically checks the status of an I/O device to determine its need for service
    - Processor is totally in control – but does all the work
    - Can waste a lot of processor time due to speed differences
  - Interrupt-driven I/O – the I/O device issues interrupts to the processor to indicate that it needs attention

Interrupt-Driven I/O
- An I/O interrupt is asynchronous w.r.t instruction execution
  - Is not associated with any instruction so doesn’t prevent any instruction from completing
    - You can pick your own convenient point to handle the interrupt
- With I/O interrupts
  - Need a way to identify the device generating the interrupt
  - Can have different urgencies (so may need to be prioritized)
- Advantages of using interrupts
  - Relieves the processor from having to continuously poll for an I/O event; user program progress is only suspended during the actual transfer of I/O data to/from user memory space
- Disadvantage – special hardware is needed to
  - Cause an interrupt (I/O device) and detect an interrupt and save the necessary information to resume normal processing after servicing the interrupt (processor)

Direct Memory Access (DMA)
- For high-bandwidth devices (like disks) interrupt-driven I/O would consume a lot of processor cycles
- DMA – the I/O controller has the ability to transfer data directly to/from the memory without involving the processor
  - The processor initiates the DMA transfer by supplying the I/O device address, the operation to be performed, the memory address destination/source, the number of bytes to transfer
  - The I/O DMA controller manages the entire transfer (possibly thousand of bytes in length), arbitrating for the bus
  - When the DMA transfer is complete, the I/O controller interrupts the processor to let it know that the transfer is complete
- There may be multiple DMA devices in one system
  - Processor and I/O controllers contend for bus cycles and for memory

The DMA Stale Data Problem
- In systems with caches, there can be two copies of a data item, one in the cache and one in the main memory
  - For a DMA read (from disk to memory) – the processor will be using stale data if that location is also in the cache
  - For a DMA write (from memory to disk) and a write-back cache – the I/O device will receive stale data if the data is in the cache and has not yet been written back to the memory
- The coherency problem is solved by
  1. Routing all I/O activity through the cache – expensive and a large negative performance impact
  2. Having the OS selectively invalidate the cache for an I/O read or force write-backs for an I/O write (flushing)
  3. Providing hardware to selectively invalidate or flush the cache – need a hardware snooper

I/O and the Operating System
- The operating system acts as the interface between the I/O hardware and the program requesting I/O
  - To protect the shared I/O resources, the user program is not allowed to communicate directly with the I/O device
- Thus OS must be able to give commands to I/O devices, handle interrupts generated by I/O devices, provide equitable access to the shared I/O resources, and schedule I/O requests to enhance system throughput
  - I/O interrupts result in a transfer of processor control to the supervisor (OS) process
Next Lecture and Reminders

Reminders

- Final exam in two weeks