Review
- Protocol suites allow heterogeneous networking
  - Another form of principle of abstraction
  - Protocols \( \Rightarrow \) operation in presence of failures
  - Standardization key for LAN, WAN
- Integrated circuit revolutionizing network switches as well as processors
  - Switch just a specialized computer
- Trend from shared to switched networks to get faster links and scalable bandwidth

Magnetic Disks
- Purpose:
  - Long-term, nonvolatile, inexpensive storage for files
  - Large, inexpensive, slow level in the memory hierarchy (discuss later)

Disk Device Terminology
- Several platters, with information recorded magnetically on both surfaces (usually)
- Bits recorded in tracks, which in turn divided into sectors (e.g., 512 Bytes)
- Actuator moves head (end of arm, 1/surface) over track ("seek"), select surface, wait for sector rotate under head, then read or write
- "Cylinder": all tracks under heads

Disk Device Performance
- Disk Latency = Seek Time + Rotation Time + Transfer Time + Controller Overhead
  - Seek Time? depends on no. tracks move arm, seek speed of disk
  - Rotation Time? depends on speed disk rotates, how far sector is from head
  - Transfer Time? depends on data rate (bandwidth) of disk (bit density), size of request
Disk Device Performance

- Average distance sector from head?
- 1/2 time of a rotation
  - 7200 Revolutions Per Minute ⇒ 120 Rev/sec
  - 1 revolution = 1/120 sec ⇒ 8.33 milliseconds
  - 1/2 rotation (revolution) ⇒ 4.16 ms

- Average no. tracks move arm?
  - Sum all possible seek distances from all possible tracks / # possible
  - Assumes average seek distance is random
  - Disk industry standard benchmark

Data Rate: Inner vs. Outer Tracks

- To keep things simple, originally kept same number of sectors per track
  - Since outer track longer, lower bits per inch
- Competition ⇒ decided to keep BPI the same for all tracks ("constant bit density")
  - More capacity per disk
  - More of sectors per track towards edge
  - Since disk spins at constant speed, outer tracks have faster data rate
- Bandwidth outer track 1.7X inner track!

Disk Performance Model /Trends

- Capacity
  + 100%/year (2X / 1.0 yrs)
- Transfer rate (BW)
  + 40%/year (2X / 2.0 yrs)
- Rotation + Seek time
  - 8%/ year (1/2 in 10 yrs)
- MB/$
  > 100%/year (2X / <1.5 yrs)
  Fewer chips + areal density

State of the Art: Ultrastar 72ZX

- 73.4 GB, 3.5 inch disk
- 2¢/MB
- 10,000 RPM; 3 ms = 1/2 rotation
- 11 platters, 22 surfaces
- 15,110 cylinders
- 7 Gbit/sq. in. areal den
- 17 watts (idle)
- 0.1 ms controller time
- 5.3 ms avg. seek
- 50 to 29 MB/s(internal)

Disk Performance Example (will fix later)

- Calculate time to read 1 sector (512B) for UltraStar 72 using advertised performance; sector is on outer track

Disk latency = average seek time + average rotational delay + transfer time + controller overhead

= 5.3 ms + 0.5 * 1/(10000 RPM) + 0.5 KB / (50 MB/s) + 0.15 ms

= 5.3 ms + 0.5 / (10000 RPM)(60000ms/M) + 0.5 KB / (50 MB/s) + 0.15 ms

= 5.3 + 3.0 + 0.10 + 0.15 ms = 8.55 ms

Areal Density

- Bits recorded along a track
  - Metric is Bits Per Inch (BPI)
- Number of tracks per surface
  - Metric is Tracks Per Inch (TPI)
- Care about bit density per unit area
  - Metric is Bits Per Square Inch
  - Called Areal Density
  - Areal Density = BPI x TPI
**Disk History (IBM)**

<table>
<thead>
<tr>
<th>Year</th>
<th>Data Density</th>
<th>Capacity of Unit Shown</th>
</tr>
</thead>
<tbody>
<tr>
<td>1973</td>
<td>1.7 Mbit/sq in</td>
<td>140 MBytes</td>
</tr>
<tr>
<td>1979</td>
<td>7.7 Mbit/sq in</td>
<td>2,300 MBytes</td>
</tr>
<tr>
<td>1989</td>
<td>63 Mbit/sq in</td>
<td>60,000 MBytes</td>
</tr>
<tr>
<td>1997</td>
<td>1450 Mbit/sq in</td>
<td>2300 MBytes</td>
</tr>
</tbody>
</table>


**Areal Density**

- Areal Density = BPI x TPI
- Change slope 30%/yr to 60%/yr about 1991

**Historical Perspective**

- Form factor and capacity drives market, more than performance
- 1970s: Mainframes ➞ 14 inch diameter disks
- 1980s: Minicomputers, Servers ➞ 8”, 5.25” diameter disks
- Late 1980s/Early 1990s:
  - Pizzabox PCs ➞ 3.5 inch diameter disks
  - Laptops, notebooks ➞ 2.5 inch disks
  - Palmtops didn’t use disks, so 1.8 inch diameter disks didn’t make it

**1 inch disk drive!**

- 2000 IBM MicroDrive:
  - 1.7” x 1.4” x 0.2”
  - 1 GB, 3600 RPM, 5 MBytes, 15 ms seek
  - Digital camera, PalmPC?
- 2006 MicroDrive?
  - 9 GB, 50 MBytes/s!
  - Assuming it finds a niche in a successful product
  - Assuming past trends continue

**Administrivia**

- Midterm Review Sunday Oct 22 starting 2 PM in 155 Dwinelle
- Midterm will be Wed Oct 25 5-8 P.M.
  - 1 Pimintel
  - Midterm conflicts? Talk to TA about taking early midterm (“beta tester”)
  - Pencils
  - 2 sides of paper with handwritten notes
  - no calculators
  - Sample midterm online, old midterms online
Fallacy: Use Data Sheet “Average Seek” Time

- Manufacturers needed standard for fair comparison (“benchmark”)
  - Calculate all seeks from all tracks, divide by number of seeks \(\Rightarrow \) “average”

- Real average would be based on how data laid out on disk, where seek in real applications, then measure performance
  - Usually, tend to seek to tracks nearby, not to random track

- Rule of Thumb: observed average seek time is typically about 1/4 to 1/3 of quoted seek time (i.e., 3X-4X faster)
  - UltraStar 72 avg. seek: 5.3 ms \(\Rightarrow 1.7 \text{ ms} \)

Fallacy: Use Data Sheet Transfer Rate

- Manufacturers quote the speed off the data rate off the surface of the disk
  - Sectors contain an error detection and correction field (can be 20% of sector size) plus sector number as well as data

- There are gaps between sectors on track

- Rule of Thumb: disks deliver about 3/4 of internal media rate (1.3X slower) for data
  - For example, UltraStar 72 quotes 50 to 29 MB/s internal media rate

\[ \Rightarrow \text{Expect 37 to 22 MB/s user data rate} \]

Disk Performance Example

- Calculate time to read 1 sector for UltraStar 72 again, this time using 1/3 quoted seek time, 3/4 of internal outer track bandwidth; \(6.55 \text{ ms before}\)

- Disk latency = average seek time + average rotational delay + transfer time + controller overhead

\[
= (0.33 \times 5.3 \text{ ms}) + 0.5 \times \frac{1}{(10000 \text{ RPM})} + 0.5 \text{ KB} / \left( \frac{0.75}{50 \text{ MB/s}} \right) + 0.15 \text{ ms}
\]

\[
= 1.77 \text{ ms} + 0.5 \times \frac{1}{(10000 \text{ RPM}) (60000 \text{ ms/M})} + 0.5 \text{ KB} / (37 \text{ KB/ms}) + 0.15 \text{ ms}
\]

\[
= 1.73 + 3.0 + 0.14 + 0.15 \text{ ms} = 5.02 \text{ ms}
\]

Future Disk Size and Performance

- Continued advance in capacity (60%/yr) and bandwidth (40%/yr)

- Slow improvement in seek, rotation (8%/yr)

- Time to read whole disk

<table>
<thead>
<tr>
<th>Year</th>
<th>Sequentially</th>
<th>Randomly</th>
</tr>
</thead>
<tbody>
<tr>
<td>1990</td>
<td>4 minutes</td>
<td>6 hours</td>
</tr>
<tr>
<td>2000</td>
<td>12 minutes</td>
<td>1 week(!)</td>
</tr>
</tbody>
</table>

- 3.5” form factor make sense in 5-7 yrs?

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

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

Disk Arrays have potential for large data and I/O rates, high MB per cu. ft., high MB per KW, but what about reliability?
Array Reliability

- **Reliability** - whether or not a component has failed
  - measured as Mean Time To Failure (MTTF)
- Reliability of N disks
  = Reliability of 1 Disk ÷ N (assuming failures independent)
  - 50,000 Hours ÷ 70 disks = 700 hour
- Disk system MTTF: Drops from 6 years to 1 month!
- Arrays too unreliable to be useful!

Redundant Arrays of (Inexpensive) Disks

- Files are "striped" across multiple disks
- Redundancy yields high data availability
  - **Availability**: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array
  ⇒ Capacity penalty to store redundant info
  ⇒ Bandwidth penalty to update redundant info

Redundant Arrays of Inexpensive Disks

RAID 1: Disk Mirroring/Shadowing

- Each disk is fully duplicated onto its “mirror”
  - 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
- (RAID 2 not interesting, so skip)

RAID 3

- Sum computed across recovery group to protect against hard disk failures, stored in P disk
- Logically, a single high capacity, high transfer rate disk: good for large transfers
- Wider arrays reduce capacity costs, but decreases availability
- 33% capacity cost for parity in this configuration

Inspiration for RAID 4

- RAID 3 relies on parity disk to discover errors on Read
- But every sector has an error detection field
- Rely on error detection field to catch errors on read, not on the parity disk
- Allows independent reads to different disks simultaneously
Redundant Arrays of Inexpensive Disks

**RAID 4: High I/O Rate Parity**

<table>
<thead>
<tr>
<th>D0</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>D7</td>
<td>P</td>
</tr>
<tr>
<td>D8</td>
<td>D9</td>
<td>D10</td>
<td>D11</td>
<td>P</td>
</tr>
<tr>
<td>D12</td>
<td>D13</td>
<td>D14</td>
<td>D15</td>
<td>P</td>
</tr>
</tbody>
</table>

Example: small read D0 & D5, large write D12-D15

Inspiration for RAID 5

- RAID 4 works well for small reads
- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk
  - Option 2: since P has old sum, compare old data to new data, add the difference to P
- Small writes are limited by Parity Disk: Write to D0, D5 both also write to P disk

Redundant Arrays of Inexpensive Disks

**RAID 5: High I/O Rate Interleaved Parity**

<table>
<thead>
<tr>
<th>D0</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>D7</td>
<td>P</td>
</tr>
<tr>
<td>D8</td>
<td>D9</td>
<td>D10</td>
<td>D11</td>
<td>P</td>
</tr>
<tr>
<td>D12</td>
<td>D13</td>
<td>D14</td>
<td>D15</td>
<td>P</td>
</tr>
<tr>
<td>D16</td>
<td>D17</td>
<td>D18</td>
<td>P</td>
<td>D19</td>
</tr>
</tbody>
</table>

Example: write to D0, D5 uses disks 0, 1, 3, 4

Berkeley History: RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software
  - Today RAID is $19 billion dollar industry, 80% nonPC disks sold in RAIDs

And in Conclusion... 1/1

- Magnetic Disks continue rapid advance: 60%/yr capacity, 40%/yr bandwidth, slow on seek, rotation improvements, MB/$ improving 100%/yr?
  - Designs to fit high volume form factor
  - Quoted seek times too conservative, data rates too optimistic for use in system
- RAID
  - Higher performance with more disk arms per $
  - Adds availability option for small number of extra disks