Concurrent Systems

3D4 ⟷ CS2016

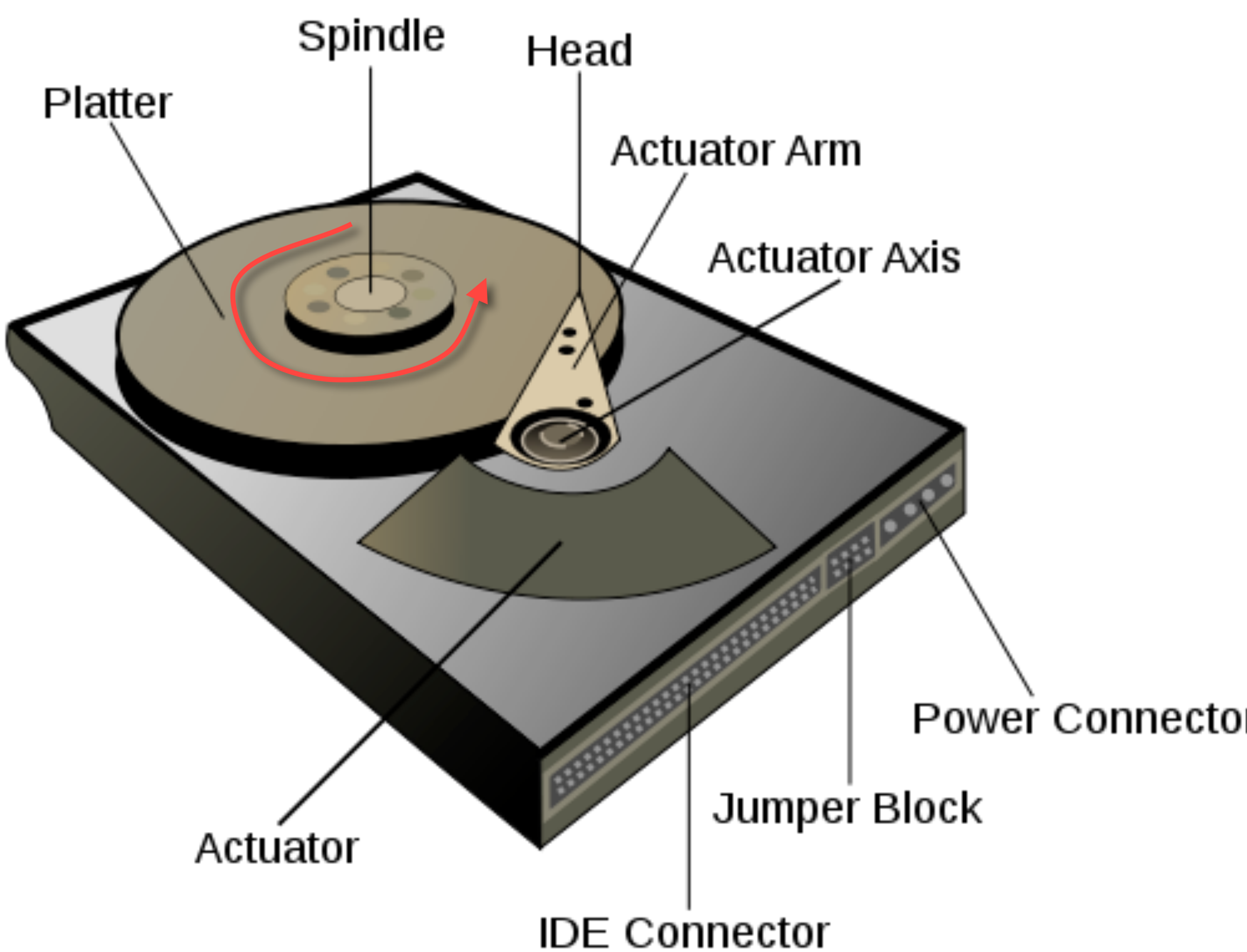
Operating Systems

Andrew Butterfield
ORI.G39, Andrew.Butterfield@scss.tcd.ie

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

with thanks to Mike Brady

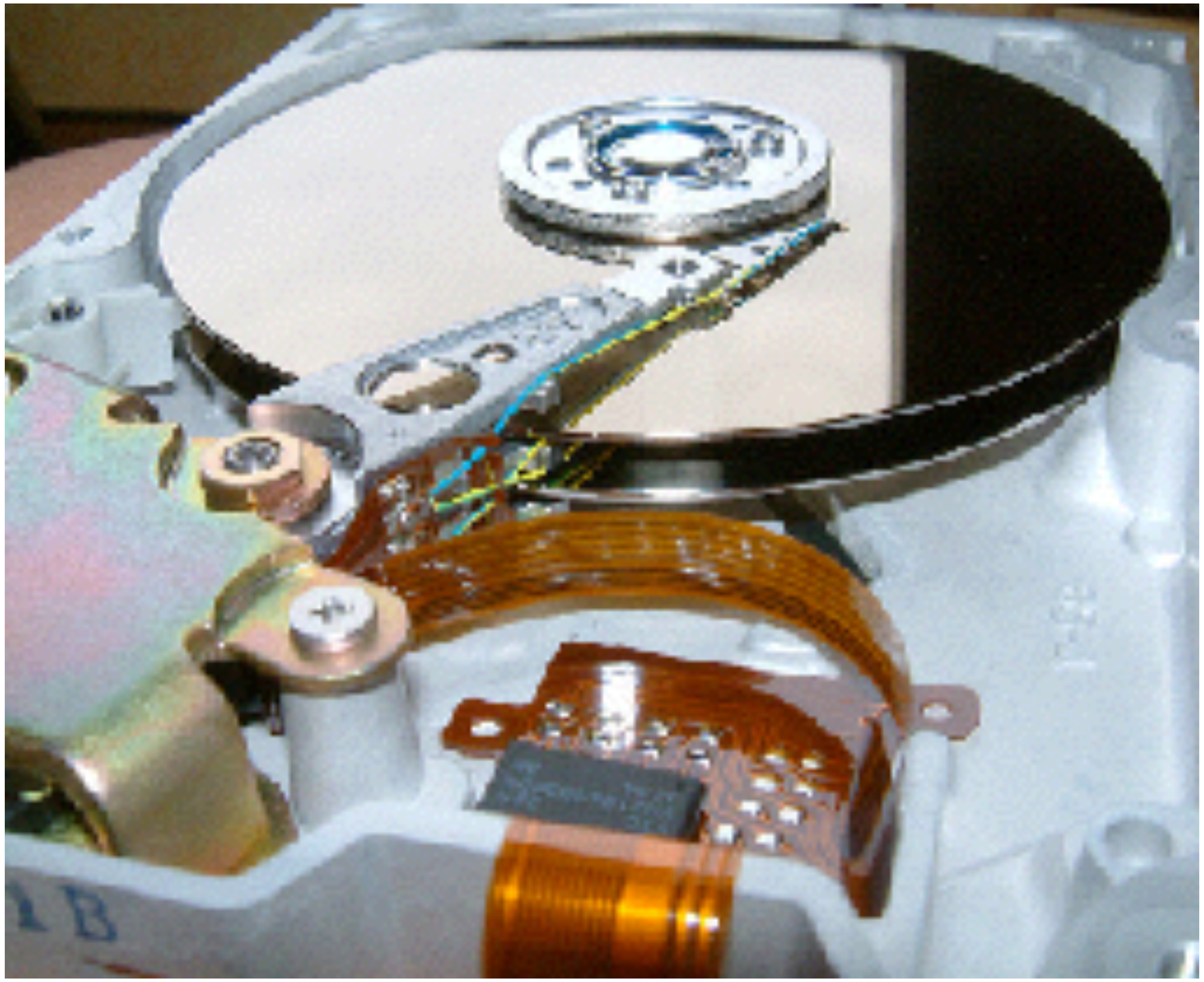
Disk Drive



Source: http://en.wikipedia.org/wiki/File:Hard_drive-en.svg

- Rotation rate, typically 5400, 7200, 10000, 15000 rpm.
- Each platter can be considered to hold a large number of concentric tracks.
- Each track is broken up into sectors.
- Sectors typically 512 bytes.
- Worst case time for a byte on a track to reappear under the read/ write head:
  - 1/120 second at 7200 rpm.

Two-Platter Drive



Disk I/O

- Almost invariably, disk I/O is interrupt-driven.
- A request is made to the disk driver, and when complete, an interrupt is generated.
- The disk interrupt handler interacts with the device queue, and when appropriate, will call a specialised disk I/O scheduler.
- Disk I/O Schedulers try to improve performance by taking device characteristics into account:
  - Latency
  - Transfer Rate
  - Seek Time
  - Position of the data required.

Hard Disk Characteristics

- Sources of Hard Disk Delays:
  - Seek Time: the head must be positioned at the correct track on the platter
  - Rotational Delay: the transfer cannot begin until the required sector is below the head (sometimes expressed as latency)
  - Data Transfer: the read or write operation is performed

Typical Hard Disk Characteristics

- Average Latency = 4.17ms at 7,000 revolutions per minute (RPM)
- Average Seek Time ~ 8 ms for standard drives, less for some server-grade drives.
- Track-To-Track seek times ~ 1 ms
- 512-byte Sectors
- Around 80 MB/sec transfer rates, up to about 150 MB/sec for 15,000 RPM drives.

Hard Disk Access Time Calculation

- Average access time:

$$T_a = \underbrace{T_s}_{\text{seek}} + \underbrace{\frac{1}{2r}}_{\text{latency}} + \underbrace{\frac{b}{rN}}_{\text{transfer}}$$

r = revs per second

b = number of sectors to be read

N = sectors per track

Typical Solid State Drive

- Typical Access Time: 75 μs
- 512-byte Sectors
- More than 500 MB/sec transfer rates

Access time example

- Suppose we have a disk with these characteristics …
  - Rotational speed: 7,200 RPM (latency = 4.17ms)
  - Average seek time: 8.5 ms
  - 512-byte sectors
  - 320 sectors per track

Access time example

- Using these average delays, how long does it take to read 2,560 sectors from a file?
  - Time to read one sector = 8.5 + 4.17 + 0.0261 = 12.6961 ms
  - Time to read 2,560 sectors = 32.5 seconds (that's slow)

Access time example – contiguous sectors

- However, if we assume the file is contiguous on disk
  - and occupies exactly 8 tracks
  - Time to seek to first track = 8.5 ms
  - Time to read track = 4.17 ms + 8.34 ms
  - Suppose seek time between tracks is 1.2 ms
  - Total read time = 8.5 + 4.17 + 8.34 + 7 x (1.2 + 4.17 + 8.34) = 117 ms

Disk Based Virtual Memory System

- Imagine we have a Virtual Memory system using a hard disk as a backing store.
- What is the average memory access time?

Page Fault Probability	0.0001	Proportion of accesses resulting in a page fault
Memory Access Time	10 ns	Main Memory Access Time
Time to Read a Page from Disk	10 ms	Read from Hard Disk

We denote RAM and Disk access time by t_R and t_D respectively.

Let p denote the probability of a page fault.

Average memory access time t_M is given by:

$$t_M = p\, t_D + (1-p)\, t_R$$

With above figures we get $t_M = 1.01us = 1010ns$ **!**

Disk Based Virtual Memory System

Memory Access Time	10 ns	Main Memory Access Time
Time to Read a Page from Disk	10 ms	Read from Hard Disk

Define *slowdown* as the ratio of average time to RAM time (t_M/t_R) - from previous example, this is *101*

Define *overhead (oh)* as the *slowdown* minus one - from previous example, this is *100* - clearly too high!

How is overhead related to page fault probability?

$p = (oh\ t_R)/t_D$ [Exercise]

If we want an overhead of *1*, (or a slowdown of 2), then we get $p_2 = t_R/t_D$

With the above figures we get $p_2 = 0.000001$! (.0001%)

Virtual Memory and (Hard) Real-Time

- They don't mix
  - Real-Time systems do scheduling very conservatively
    - based on concept of Worst-Case Execution Time (WCET)
    - Given the vast difference in time between RAM and Disk access, a WCET analysis would be far too conservative
  - If page-fault probability is low, can we just ignore it?
    - the probability is per memory access
    - even if low, in a long mission it will occur eventually
    - inevitably losing a mission due to eventually missing a hard deadline due to a page fault is not OK
  - So most RT systems only use the MMU for memory protection
    - Every program has a fixed pre-determined address space.