

# CS 245: Database System Principles

## Notes 02: Hardware

Hector Garcia-Molina

CS 245

Notes 2

1

## Outline

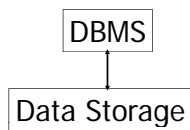
- Hardware: Disks
- Access Times
- Example - Megatron 747
- Optimizations
- Other Topics:
  - Storage costs
  - Using secondary storage
  - Disk failures

CS 245

Notes 2

2

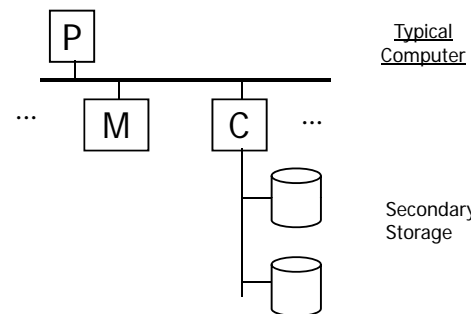
## Hardware



CS 245

Notes 2

3



CS 245

Notes 2

4

## Processor

Fast, slow, reduced instruction set,  
with cache, pipelined...  
Speed: 100 → 500 → 1000 MIPS

## Memory

Fast, slow, non-volatile, read-only,...  
Access time:  $10^{-6}$  →  $10^{-9}$  sec.  
 $1 \mu\text{s}$  → 1 ns

CS 245

Notes 2

5

## Secondary storage

Many flavors:

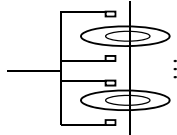
- Disk: Floppy (hard, soft)  
Removable Packs  
Winchester  
Ram disks  
Optical, CD-ROM...  
Arrays
- Tape Reel, cartridge  
Robots

CS 245

Notes 2

6

Focus on: "Typical Disk"



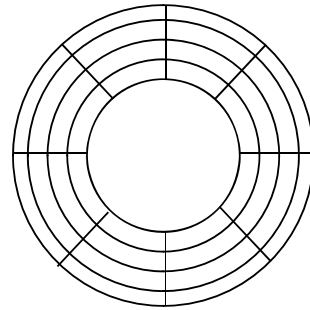
Terms: Platter, Head, Actuator  
Cylinder, Track  
Sector (physical),  
Block (logical), Gap

CS 245

Notes 2

7

Top View



CS 245

Notes 2

8

"Typical" Numbers

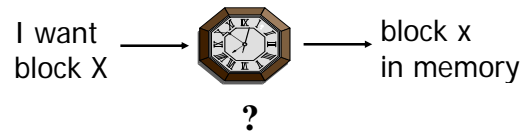
Diameter: 1 inch → 15 inches  
Cylinders: 100 → 2000  
Surfaces: 1 (CDs) →  
(Tracks/cyl) 2 (floppies) → 30  
Sector Size: 512B → 50K  
Capacity: 360 KB (old floppy)  
→ 400 GB (I use)

CS 245

Notes 2

9

Disk Access Time



CS 245

Notes 2

10

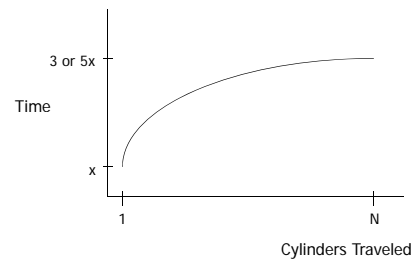
Time = Seek Time +  
Rotational Delay +  
Transfer Time +  
Other

CS 245

Notes 2

11

Seek Time



CS 245

Notes 2

12

### Average Random Seek Time

$$S = \frac{\sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \text{SEEKTIME}(i \rightarrow j)}{N(N-1)}$$

CS 245

Notes 2

13

### Average Random Seek Time

$$S = \frac{\sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \text{SEEKTIME}(i \rightarrow j)}{N(N-1)}$$

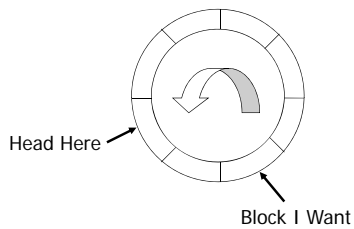
"Typical" S: 10 ms → 40 ms

CS 245

Notes 2

14

### Rotational Delay



CS 245

Notes 2

15

### Average Rotational Delay

R = 1/2 revolution

"typical" R = 8.33 ms (3600 RPM)

CS 245

Notes 2

16

### Transfer Rate: t

- "typical" t: 1 → 3 MB/second
- transfer time:  $\frac{\text{block size}}{t}$

CS 245

Notes 2

17

### Other Delays

- CPU time to issue I/O
- Contention for controller
- Contention for bus, memory

CS 245

Notes 2

18

## Other Delays

- CPU time to issue I/O
- Contention for controller
- Contention for bus, memory

"Typical" Value: 0

CS 245

Notes 2

19

- So far: Random Block Access
- What about: Reading "Next" block?


CS 245

Notes 2

20

If we do things right (e.g., Double Buffer, Stagger Blocks...)

Time to get block =  $\frac{\text{Block Size}}{t}$  + Negligible

- 
- skip gap
  - switch track
  - once in a while, next cylinder

CS 245

Notes 2

21

|                      |  |
|----------------------|--|
| <b>Rule of Thumb</b> | Random I/O: Expensive<br>Sequential I/O: Much less |
|----------------------|--|

- Ex: 1 KB Block
  - » Random I/O: ~ 20 ms.
  - » Sequential I/O: ~ 1 ms.

CS 245

Notes 2

22

Cost for Writing similar to Reading

.... unless we want to verify!  
need to add (full) rotation +  $\frac{\text{Block size}}{t}$

CS 245

Notes 2

23

- To Modify a Block?

CS 245

Notes 2

24

- To Modify a Block?

To Modify Block:

- (a) Read Block
- (b) Modify in Memory
- (c) Write Block
- [(d) Verify?]

CS 245

Notes 2

25

Block Address:

- Physical Device
- Cylinder #
- Surface #
- Sector

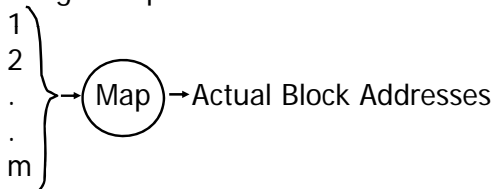
CS 245

Notes 2

26

Complication: Bad Blocks

- Messy to handle
- May map via software to integer sequence



CS 245

Notes 2

27

An Example Megatron 747 Disk (old)

- 3.5 in diameter
- 3600 RPM
- 1 surface
- 16 MB usable capacity ( $16 \times 2^{20}$ )
- 128 cylinders
- seek time: average = 25 ms.  
adjacent cyl = 5 ms.

CS 245

Notes 2

28

- 1 KB blocks = sectors
- 10% overhead between blocks
- capacity = 16 MB =  $(2^{20})16 = 2^{24}$
- # cylinders =  $128 = 2^7$
- bytes/cyl =  $2^{24}/2^7 = 2^{17} = 128 \text{ KB}$
- blocks/cyl =  $128 \text{ KB} / 1 \text{ KB} = 128$

CS 245

Notes 2

29

3600 RPM  $\rightarrow$  60 revolutions / sec  
 $\rightarrow$  1 rev. = 16.66 msec.

One track:



CS 245

Notes 2

30

3600 RPM → 60 revolutions / sec  
 → 1 rev. = 16.66 msec.



Time over useful data:  $(16.66)(0.9) = 14.99$  ms.  
 Time over gaps:  $(16.66)(0.1) = 1.66$  ms.  
 Transfer time 1 block =  $14.99/128 = 0.117$  ms.  
 Trans. time 1 block+gap =  $16.66/128 = 0.13$  ms.

### Burst Bandwidth

1 KB in 0.117 ms.

$$BB = 1/0.117 = 8.54 \text{ KB/ms.}$$

or

$$BB = 8.54 \text{ KB/ms} \times 1000 \text{ ms/1sec} \times 1 \text{ MB}/1024 \text{ KB} \\ = 8540/1024 = 8.33 \text{ MB/sec}$$

Sustained bandwidth (over track)  
 128 KB in 16.66 ms.

$$SB = 128/16.66 = 7.68 \text{ KB/ms}$$

or

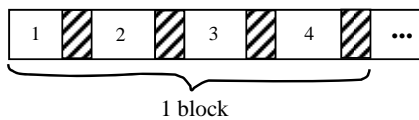
$$SB = 7.68 \times 1000/1024 = 7.50 \text{ MB/sec.}$$

$T_1$  = Time to read one random block

$$T_1 = \text{seek} + \text{rotational delay} + TT$$

$$= 25 + (16.66/2) + .117 = 33.45 \text{ ms.}$$

Suppose OS deals with 4 KB blocks



$$T_4 = 25 + (16.66/2) + (.117) \times 1 \\ + (.130) \times 3 = 33.83 \text{ ms}$$

[Compare to  $T_1 = 33.45$  ms]

$T_T$  = Time to read a full track  
 (start at any block)

$$T_T = 25 + (0.130/2) + 16.66^* = 41.73 \text{ ms}$$

↑  
 to get to first block

\* Actually, a bit less; do not have to read last gap.

## The NEW Megatron 747 (Example 2.1 book)

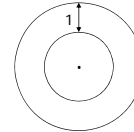
- 8 Surfaces, 3.5 Inch diameter
  - outer 1 inch used
- $2^{13} = 8192$  Tracks/surface
- 256 Sectors/track
- $2^9 = 512$  Bytes/sector

CS 245

Notes 2

37

- 8 GB Disk
- If all tracks have 256 sectors
  - Outermost density: 100,000 bits/inch
  - Inner density: 250,000 bits/inch



CS 245

Notes 2

38

- Outer third of tracks: 320 sectors
- Middle third of tracks: 256
- Inner third of tracks: 192
  
- Density: 114,000 → 182,000 bits/inch

CS 245

Notes 2

39

## Timing for new Megatron 747 (Ex 2.3)

- Time to read 4096-byte block:
  - MIN: 0.5 ms
  - MAX: 33.5 ms
  - AVE: 14.8 ms

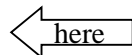
CS 245

Notes 2

40

## Outline

- Hardware: Disks
- Access Times
- Example: Megatron 747
- Optimizations
- Other Topics
  - Storage Costs
  - Using Secondary Storage
  - Disk Failures



CS 245

Notes 2

41

## Optimizations (in controller or O.S.)

- Disk Scheduling Algorithms
  - e.g., elevator algorithm
- Track (or larger) Buffer
- Pre-fetch
- Arrays
- Mirrored Disks
- On Disk Cache

CS 245

Notes 2

42

## Double Buffering

Problem: Have a File

- » Sequence of Blocks B1, B2

Have a Program

- » Process B1
- » Process B2
- » Process B3

⋮

CS 245

Notes 2

43

## Single Buffer Solution

- (1) Read B1 → Buffer
- (2) Process Data in Buffer
- (3) Read B2 → Buffer
- (4) Process Data in Buffer ...

CS 245

Notes 2

44

Say  $P$  = time to process/block  
 $R$  = time to read in 1 block  
 $n$  = # blocks

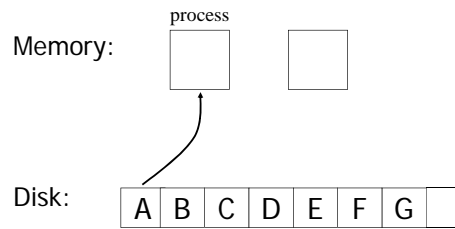
Single buffer time =  $n(P+R)$

CS 245

Notes 2

45

## Double Buffering

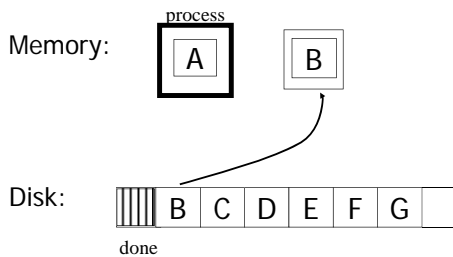


CS 245

Notes 2

46

## Double Buffering

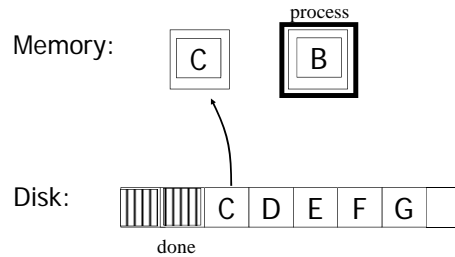


CS 245

Notes 2

47

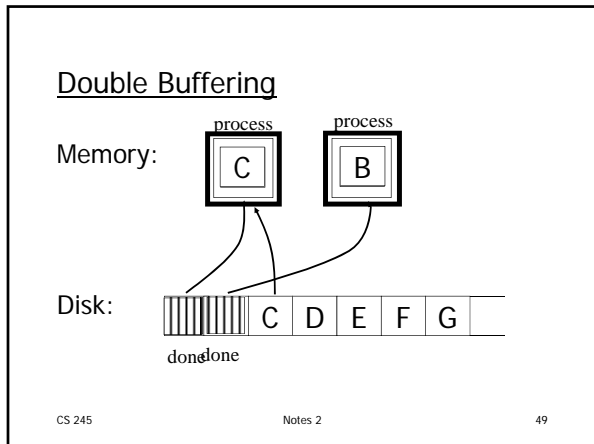
## Double Buffering



CS 245

Notes 2

48



Say  $P \geq R$

$P$  = Processing time/block  
 $R$  = IO time/block  
 $n$  = # blocks

What is processing time?

CS 245 Notes 2 50

Say  $P \geq R$

$P$  = Processing time/block  
 $R$  = IO time/block  
 $n$  = # blocks

What is processing time?

- Double buffering time =  $R + nP$
- Single buffering time =  $n(R+P)$

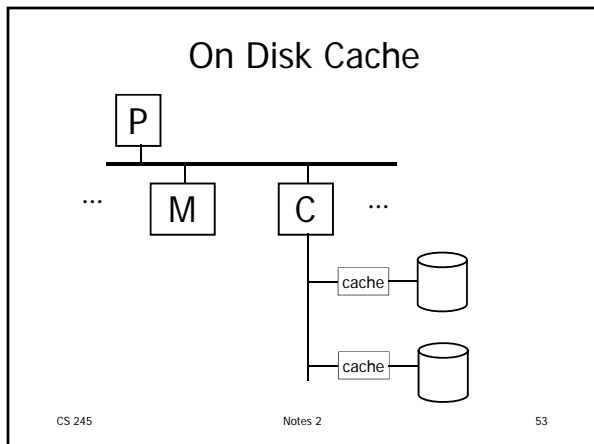
CS 245 Notes 2 51

### Disk Arrays

- RAIDs (various flavors)
- Block Striping
- Mirrored

logically one disk

CS 245 Notes 2 52



Block Size Selection?

- Big Block  $\rightarrow$  Amortize I/O Cost

Unfortunately...

- Big Block  $\Rightarrow$  Read in more useless stuff!  
and takes longer to read

CS 245 Notes 2 54

### Trend

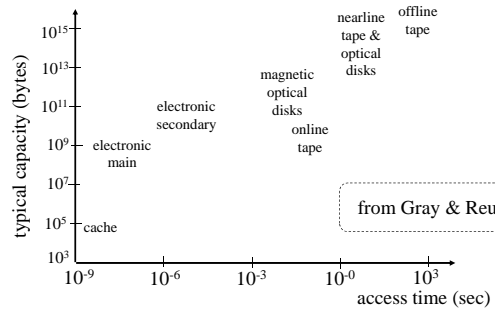
- As memory prices drop, blocks get bigger ...

CS 245

Notes 2

55

### Storage Cost

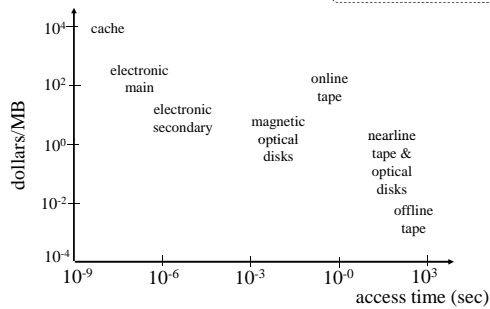


CS 245

Notes 2

56

### Storage Cost



CS 245

Notes 2

57

### Using secondary storage effectively (Sec. 2.3)

- Example: Sorting data on disk
- Conclusion:
  - I/O costs dominate
  - Design algorithms to reduce I/O
- Also: How big should blocks be?

CS 245

Notes 2

58

### Five Minute Rule

- THE 5 MINUTE RULE FOR TRADING MEMORY FOR DISC ACCESSES  
Jim Gray & Franco Putzolu  
May 1985
- The Five Minute Rule, Ten Years Later  
Goetz Graefe & Jim Gray  
December 1997

CS 245

Notes 2

59

### Five Minute Rule

- Say a page is accessed every X seconds
- CD = cost if we keep that page on disk
  - $\$D$  = cost of disk unit
  - I = numbers IOs that unit can perform
  - In X seconds, unit can do XI IOs
  - So  $CD = \$D / XI$

CS 245

Notes 2

60

## Five Minute Rule

- Say a page is accessed every X seconds
- CM = cost if we keep that page on RAM
  - \$M = cost of 1 MB of RAM
  - P = numbers of pages in 1 MB RAM
  - So CM = \$M / P

CS 245

Notes 2

61

## Five Minute Rule

- Say a page is accessed every X seconds
- If CD is smaller than CM,
  - keep page on disk
  - else keep in memory
- Break even point when CD = CM, or

$$X = \frac{\$D P}{I \$M}$$

CS 245

Notes 2

62

## Using '97 Numbers

- P = 128 pages/MB (8KB pages)
- I = 64 accesses/sec/disk
- \$D = 2000 dollars/disk (9GB + controller)
- \$M = 15 dollars/MB of DRAM
  
- X = 266 seconds (about 5 minutes)  
(did not change much from 85 to 97)

CS 245

Notes 2

63

## Disk Failures (Sec 2.5)

- Partial → Total
- Intermittent → Permanent

CS 245

Notes 2

64

## Coping with Disk Failures

- Detection
  - e.g. Checksum
  
- Correction
  - ⇒ Redundancy

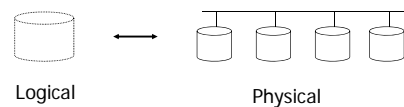
CS 245

Notes 2

65

## At what level do we cope?

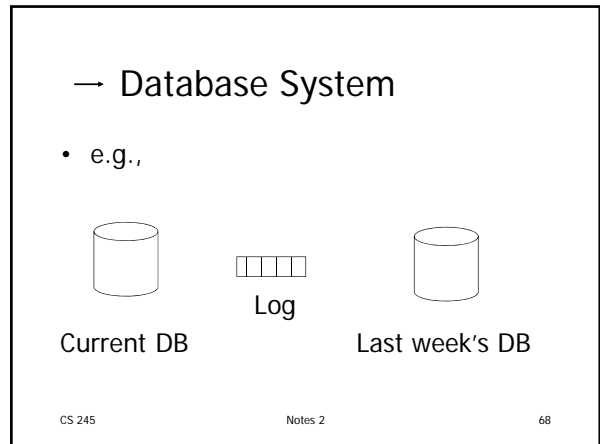
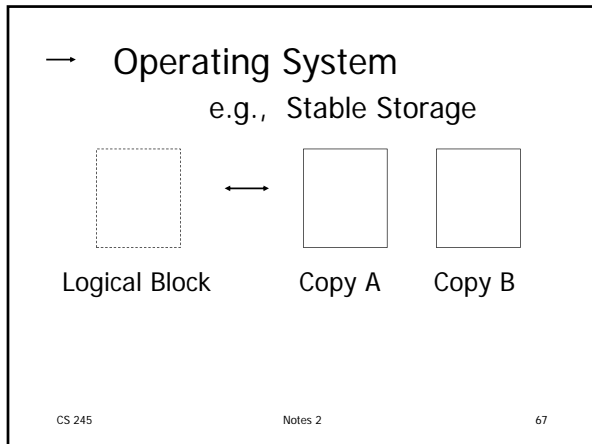
- Single Disk
  - e.g., Error Correcting Codes
- Disk Array



CS 245

Notes 2

66



- Summary**
- Secondary storage, mainly disks
  - I/O times
  - I/Os should be avoided, especially random ones.....
- CS 245      Notes 2      69

- Outline
- Hardware: Disks
  - Access Times
  - Example: Megatron 747
  - Optimizations
  - Other Topics
    - Storage Costs
    - Using Secondary Storage
    - Disk Failures
- ← here
- CS 245      Notes 2      70