CSE232A: Database System Principles

Hardware

Database System Architecture

Query Processing

Transaction Management

SQL query

Hardware aspects of storing and retrieving data

relational algebra

view definitions

Statistics & Catalogs & System Data

Buffer Manager

Data + Indexes

Query execution plan

Concurrency Controller

Lock Table

Cost per byte

Capacity

Access Speed

Memory Hierarchy

- Cache memory
  - On-chip and L2
  - Caching outside control of DB system
- RAM
  - Addressable space includes virtual memory but DB systems avoid it
- Disk
  - Access speed & Transfer rate
  - Winchester, arrays,...
- Tertiary storage
  - Tapes, jukeboxes, DVDs
Volatile Vs Non-Volatile Storage

- Persistence important for transaction atomicity and durability
- Even if database fits in main memory changes have to be written in non-volatile storage
- Hard disk
- RAM disks w/ battery
- Flash memory
Cost of Disk Access: Non-trivial part of estimating performance on secondary storage

- How many blocks were accessed?
- Clustered/consecutive?

- Such complexities also apply to flash, even main memory
- Learn to analyze them when you make the next generation of secondary storage data structures

Moore’s Law: Different Rates of Improvement Lead to Reconsiderations

- Processor speed
- Main memory bit/$
- Disk bit/$
- RAM access speed
- Disk access speed
- Disk transfer rate

Clustering/sequential access-based algorithms become relatively better

Moore’s Law: Same Phenomenon Applies to RAM

Algorithms that access memory sequentially have better constant factors than algorithms that access randomly
Moore’s Law: Different Rates of Improvement

- Cache Capacity
- RAM Capacity
- Disk Access Time

Cost of “miss” increases

Focus on: “ Typical Disk”

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

Top View

Often different numbers of sectors per track

Block (typically multiple sectors)
"Typical" Numbers

- Diameter: 1 inch → 15 inches
- Cylinders: 100 → 20000
- Surfaces: 1 (CDs) → 2 (floppies) → 5 (typical hd) → 30
- Sector Size: 512B → 50K
- Capacity: 360 KB (old floppy) → 200 GB

Key performance metric: Time to fetch block

I want block X in memory?

Time = Seek Time (locate track) + Rotational Delay (locate sector) + Transfer Time (fetch block) + Other (disk controller, ...)

Seek Delay

Track Where Head must go

Track Where Head is
Rotational Delay

Seek Time

Average Random Seek Time

\[
S = \frac{\sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \text{SEEKTIME (i \rightarrow j)}}{N(N-1)}
\]

“Typical” \( S \): 10 ms → 40 ms
Average Rotational Delay

\[ R = \frac{1}{2} \text{ revolution} \]

"typical" \( R = 8.33 \text{ ms (7200 RPM)} \)

Assume we have to start reading from start of first sector

Transfer Rate: \( t \)

- "typical" \( t: 1 \rightarrow 3 \text{ MB/second} \)
- transfer time: \( \text{block size} \)/\( t \)

Other Delays

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

"Typical" Value: 0
Homework Practice Problem

• Single surface
• Rotation speed 7200rpm
• 16,384 tracks
• 128 sectors/track
• 4096 bytes/sector
• 4 sectors/block (16,384 bytes/block)
• SEEKTIME (i \rightarrow j) = [1000 + (j-i)] \mu s
• Neglect gaps
• Calculate minimum, maximum, average time to fetch one block

Practice Problem: Minimum Time

• Head is at the start of the first sector of the block
• Just compute transfer time
• 4 sectors cover 4/128 of a track
• 1 full rotation takes 60/7200=8.33ms
• Transfer time is 8.33 * 4 /128 = 0.26ms

Practice Problem: Maximum Time

• Assume read must start at the first sector
• Head is at innermost, required track is the outermost
• Seek time = ...
• Head just missed the beginning
• Rotational delay = ...
• Transfer time = ...
Practice problem: Average time

- Solve...

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

Time to get \( t \) = Block Size \( + \) Negligible

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

Rule of Thumb

Random I/O: Expensive
Sequential I/O: Much less

- Ex:
  - 1 KB Block
    - Random I/O: ~ 20 ms.
    - Sequential I/O: ~ 1 ms.
Practice Problem cont’d: Sustained Bandwidth over Track

• Assume required blocks are consecutive on single track
• What is the approximate sustained bandwidth of fetching consecutive blocks?
• 128 sectors/track * 4KB/sector in 8.33ms/track full rotation = 512KB/8.33ms = 61.46KB/ms

Suggested optimization

• Cluster data in consecutive blocks
• Give an extra point to algorithms that
  – exploit data clustering by avoiding "random" accesses
  – Read/write consecutive blocks

An Algorithm with Little Random Access: 2-Phase Merge Sort

Main Memory: 4 blocks

MERGE

Acdf ...

Write ADEKPMZ

READ ADEKPMZ

Sort

READ

Write ADEKPMZ

Write CFRJXBY

Improve by bringing max number of blocks in memory in Phase 2
Cost for Writing similar to Reading

.... unless we want to verify!
   need to add (full) rotation + Block size
   t

• To Modify a Block?

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

Block Address:

• Physical Device
• Cylinder #
• Surface #
• Sector

Once upon a time DBs had access to such – now it is the OS’s domain
Optimizations (in controller or O.S.)

- Disk Scheduling Algorithms
  - e.g., elevator algorithm
- Pre-fetch
- Arrays

Double Buffering

Problem: Have a File
  » Sequence of Blocks B1, B2

Have a Program
  » Process B1
  » Process B2
  » Process B3
  ...

Single Buffer Solution

1. Read B1 → Buffer
2. Process Data in Buffer
3. Read B2 → Buffer
4. Process Data in Buffer ...
Say  \( P = \) time to process/block  
\( R = \) time to read in 1 block  
\( n = \) # blocks  

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

---

**Double Buffering**

Memory:

```
C
B
```

Disk:

```
C D E F G
```

done

done

---

Say \( P \geq R \)

\[
P = \text{Processing time/block} \\
R = \text{IO time/block} \\
n = \text{# blocks}
\]

What is processing time?

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

Improvement much more dramatic if consecutive blocks: ...
Block Size Selection?

• Big Block $\Rightarrow$ Amortize I/O Cost

Unfortunately...

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

Trend

• memory prices drop and memory capacities increase,
• transfer rates increase
• Disk access times do not increase that much

$\Rightarrow$ blocks get bigger ...

Summary

• Secondary storage, mainly disks
• I/O times
• I/Os should be avoided,
  especially random ones.....