CS143: Disks and Files
System Architecture

CPU

Main Memory

System Bus

Disk Controller

Disk

Word (1B – 64B)
~ 10 GB/sec

Block (512B – 50KB)
~ 100 MB/sec
Magnetic disk vs SSD

• Magnetic Disk
  – Stores data on a magnetic disk
  – Typical capacity: 1TB – 10TB

• Solid State Drive
  – Stores data in NAND flash memory
  – Typical capacity: 100GB – 1TB
  – Much faster and more reliable than magnetic disk
  – But, x10 more expensive and limited write cycles (~2000)
Structure of a Platter

- Track, cylinder, sector (=block, page)
Typical Magnetic Disk

- Platter diameter: 2.5-5.25 in
- Platters: 1 – 20
- Tracks: 1000 – 50,000
- Sectors per track: 1000 – 50,000
- Sector size: 512 – 50K
- Rotation speed: 1000 – 15000 rpm
- Overall capacity: 1TB – 10TB

Q: 2 platters, 2 surfaces/platter, 20,000 tracks/surface, 20,000 sectors(track, 1KB/sector. What is the overall capacity?
Capacity of Magnetic Disk

- Capacity keeps increasing, but what about speed?
Access Time

• Q: How long does it take to read a page of a disk to memory?

• Q: What needs to be done to read a page?
Access Time

- Access time = 
  (seek time) + (rotational delay) + 
  (transfer time)
Seek Time

• Time to move a disk head between tracks

• Track to track ~ 1ms
• Average ~ 10 ms
• Full stroke ~ 20 ms
Rotational Delay

- Typical disk:
  - 1000 rpm – 15000 rpm
- Q: For 6000 RPM, average rotational delay?
Transfer Rate

6000 RPM, 10,000 sectors/track, 1KB/sector

• Q: How long to read one sector?

• Q: What is the transfer rate (bytes/sec)?
(Burst) Transfer Rate

• (Burst) Transfer rate =
  \((\text{RPM} / 60) \times (\text{sectors/track}) \times (\text{bytes/sector})\)
Sequential vs. Random I/O

• Q: How long to read 3 sequential sectors?

- 6000 RPM
- 10,000 sectors/track
- Assume the head is above the first sector
Sequential vs. Random I/O

Q: How long to read 3 random sectors?

- 6000 RPM
- 10,000 sectors/track
- 10ms seek time
- Assume the head is above the first sector
Random I/O

• For magnetic disks:
  – Random I/O is VERY expensive compared to sequential I/O
  – Avoid random I/O as much as we can
## Magnetic Disk vs SSD

<table>
<thead>
<tr>
<th></th>
<th>Magnetic</th>
<th>SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random IO</td>
<td>~100 IOs/sec</td>
<td>~100K IOs/sec</td>
</tr>
<tr>
<td>Transfer rate</td>
<td>~100MB/sec</td>
<td>~500MB/sec</td>
</tr>
<tr>
<td>Capacity/$</td>
<td>~1TB/$50 (in 2015)</td>
<td>~100GB/$50 (in 2015)</td>
</tr>
</tbody>
</table>

SSD speed gain is mainly from high random IO rate
RAID

• Redundant Array of Independent Disks
  – Create a large-capacity “disk volumes” from an array of many disks

• Q: Possible advantages and disadvantages?
RAID Pros and Cons

• Potentially high throughput
  – Read from multiple disks concurrently

• Potential reliability issues
  – One disk failure may lead to the entire disk volume failure
  – How should we store data into disks?

• Q: How should we organize the disks and store data to maximize benefit and minimize risks?
RAID Levels

• RAID 0: striping only (no redundancy)
• RAID 1: striping + mirroring
• RAID 5: striping + parity block
Data Modification

• Byte-level modification not allowed
  – Can be modified by blocks

• Q: How can we modify only a part of a block?
Abstraction by OS

- Sequential blocks
  - No need to worry about head, cylinder, sector
- Access to non-adjacent blocks
  - Random I/O
- Access to adjacent blocks
  - Sequential I/O

(head, cylinder, sector)
Buffers, Buffer pool

• Temporary main-memory “cache” for disk blocks
  – Avoid future read
  – Hide disk latency
  – Most DBMS let users change buffer pool size
Reference

• Storage review disk guide
Files: Main Problem

- How to store tables into disks?

<table>
<thead>
<tr>
<th>Name</th>
<th>Major</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jane</td>
<td>CS</td>
<td>3.7</td>
</tr>
<tr>
<td>Susan</td>
<td>ME</td>
<td>1.8</td>
</tr>
<tr>
<td>June</td>
<td>EE</td>
<td>2.6</td>
</tr>
<tr>
<td>Tony</td>
<td>CS</td>
<td>3.1</td>
</tr>
</tbody>
</table>
Spanned vs Unspanned

• Q: 512Byte block. 80Byte tuple. How to store?
Spanned vs Unspanned

- **Unspanned**

  ![Unspanned Diagram]

  - T1
  - T2
  - T3
  - T4
  - T5
  - T6

- **Spanned**

  ![Spanned Diagram]

  - T1
  - T2
  - T3
  - T4
  - T5
  - T6

- **Q:** Maximum space waste for unspanned?
Variable-Length Tuples

• How do we store them?
Reserved Space

• Reserve the maximum space for each tuple

• Q: Any problem?
Variable-Length Space

<table>
<thead>
<tr>
<th>R1</th>
<th></th>
<th>R2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R3</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td></td>
<td>R5</td>
</tr>
<tr>
<td></td>
<td>R6</td>
<td></td>
</tr>
</tbody>
</table>

- Pack tuples tightly
- Q: How do we know the end of a record?
- Q: What to do for delete/update?
- Q: How can we “point to” to a tuple?
Q: How can we point to a tuple?
Long Tuples

• ProductReview(
  pid INT,
  reviewer VARCHAR(50),
  date DATE,
  rating INT,
  comments VARCHAR(1000))

• Block size 512B

• How should we store it?
Long Tuples

• Spanning
• Splitting tuples

Block with short attributes. 

Block with long attrs. 

This block may also have fixed-length slots.
Sequential File

- Tuples are ordered by certain attribute(s) (search key)

<table>
<thead>
<tr>
<th>Name</th>
<th>Department</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elaine</td>
<td>CS</td>
<td>3.7</td>
</tr>
<tr>
<td>James</td>
<td>ME</td>
<td>2.8</td>
</tr>
<tr>
<td>John</td>
<td>EE</td>
<td>1.8</td>
</tr>
<tr>
<td>Peter</td>
<td>EE</td>
<td>3.9</td>
</tr>
<tr>
<td>Susan</td>
<td>CS</td>
<td>1.0</td>
</tr>
<tr>
<td>Tony</td>
<td>EE</td>
<td>2.4</td>
</tr>
</tbody>
</table>

- Search key: Name
Sequencing Tuples

• Inserting a new tuple
  – Easy case

```
+---+---+---+---+
| T1| T3| T6| T8 |
+---+---+---+---+
```

? → T7
Sequencing Tuples

Two options

1) Rearrange

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>T3</td>
<td>T6</td>
<td>T7</td>
<td>T8</td>
</tr>
</tbody>
</table>

2) Linked list

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>T7</td>
<td>T3</td>
<td>T6</td>
<td>T8</td>
</tr>
</tbody>
</table>
Sequencing Tuples

- Inserting a new tuple
  - Difficult case

<table>
<thead>
<tr>
<th>T1</th>
<th>T4</th>
<th>T5</th>
<th>T8</th>
<th>T9</th>
</tr>
</thead>
</table>

T7
Sequencing Tuples

- **Overflow page**

<table>
<thead>
<tr>
<th>header</th>
<th>T1</th>
<th>T2</th>
<th>T4</th>
<th>T6</th>
<th>T7</th>
</tr>
</thead>
</table>

- Reserving free space to avoid overflow
  - PCTFREE in DBMS
  ```sql
  CREATE TABLE R(a int) PCTFREE 40
  ```
Things to Remember

• Disk
  – Platter, track, cylinder, sector, block
  – Seek time, rotational delay, transfer time
  – Random I/O vs Sequential I/O

• Files
  – Spanned/unspanned tuples
  – Variable-length tuples (slotted page)
  – Long tuples
  – Sequential file and search key
    • Problems with insertion (overflow page)
    • PCTFREE