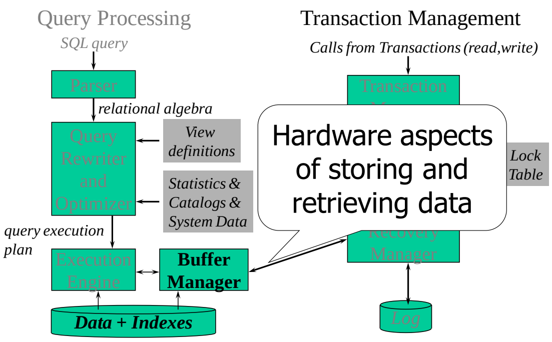
CSE232: Database System Principles

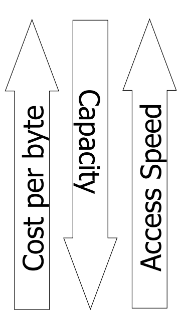
Hardware

Database System Architecture



Memory Hierarchy

- Cache memory
  - On-chip and L2
  - Caching outside control of DB system
  - Increasingly important - comments
- RAM (controlled by db system)
  - Addressable space includes virtual memory but DB systems avoid it
- SSDs
  - Block-based storage
- Disk
  - Block
  - Preference to sequential access
- Tertiary storage for archiving
  - Tapes, jukeboxes, DVDs
  - Does not matter any more



Non-Volatile Storage is important even when RAM is large

- 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

4

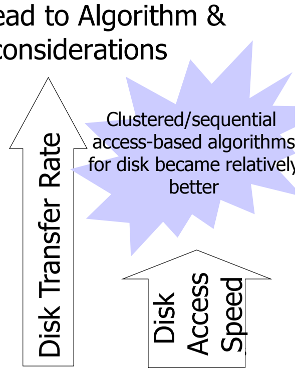
Peculiarities of storage mediums affect algorithm choice

- Block-based access:
  - Access performance: How many blocks were accessed
  - ~~How many objects~~
  - Flash is different on reading Vs writing
- Clustering for sequential access:
  - Accessing consecutive blocks costs less on disk-based systems
- We will only consider the effects of block access

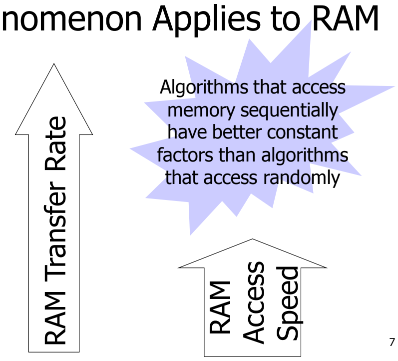
5

Moore's Law: Different Rates of Improvement Lead to Algorithm & System Reconsiderations

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

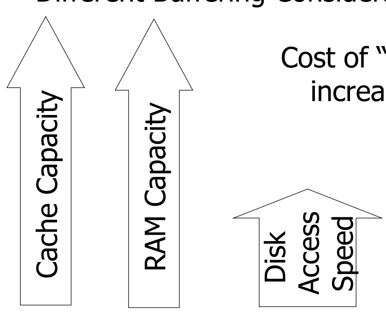


Moore's Law: Same Phenomenon Applies to RAM



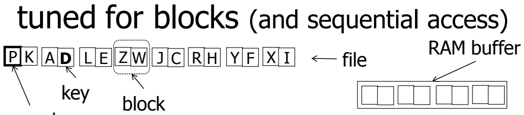
7

Moore's Law: Different Rates of Improvement => Different Buffering Considerations



8

2-Phase Merge Sort: An algorithm tuned for blocks (and sequential access)

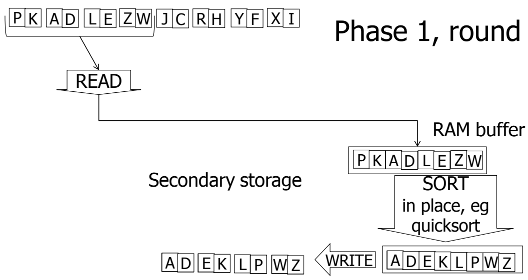


Assume a file with many records.
Each record has a key and other data.
For ppt brevity, the slide shows only the key of each record and not its data.
Assume each block has 2 records.
Assume RAM buffer fits 4 blocks (8 records)
In practice, expect many more records per block and many more records fitting in buffer.

Problem: Sort the records according to the key.
Morale: What you learnt in algorithms and data structures is not always the best when we consider block-based storage

2-Phase Merge Sort

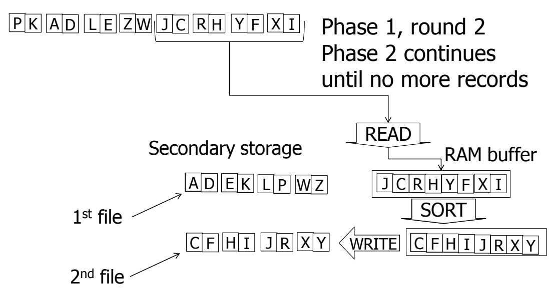
Phase 1, round 1



10

2-Phase Merge Sort

Phase 1, round 2
Phase 2 continues
until no more records



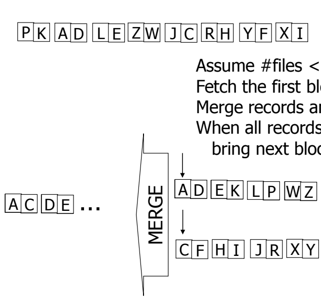
In practice, probably many more Phase 1 rounds and many respective output files

11

2-Phase Merge Sort

Phase 2

Assume #files < #blocks that fit in RAM buffer.
Fetch the first block of each file in RAM buffer.
Merge records and output.
When all records of a block have been output, bring next block of same file



Improvement: Bring max number of blocks in memory.

12

2-Phase Merge Sort: Most files can be sorted in just 2 passes!

Assume
- *M* bytes of RAM buffer (eg, 8GB)
- *B* bytes per block (eg, 64KB for disk, 4KB for SSD)

Calculation:
- The assumption of Phase 2 holds when *#files < M/B*

=> there can be up to *M/B* Phase 1 rounds
- Each round can process up to *M* bytes of input data

=> 2-Phase Merge Sort can sort $\boldsymbol{M^2/B}$ bytes
  - eg $(8GB)^2/64KB = (2^{33}B)^2 / 2^{16}B = 2^{50}B = 1PB$

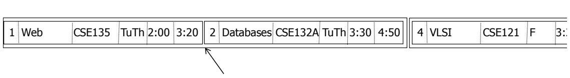
Horizontal placement of SQL data in blocks

Relations:
- Pack as many tuples per block
  - improves scan time
- Do not reclaim deleted records
- Utilize overflow records if relation must be sorted on primary key
- A novel generation of databases features column storage
  - to be discussed late in class

Pack maximum #records per block

Classes

id	name	number	date_code	start_time	end_time
1	Web	CSE135	TuTh	2:00	3:20
2	Databases	CSE132A	TuTh	3:30	4:50
4	VLSI	CSE121	F	*null*	*null*



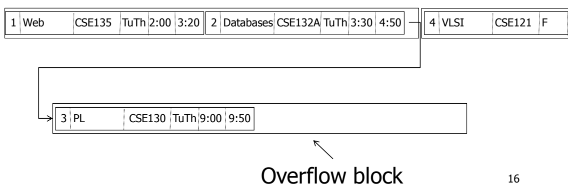
"pack" each block with maximum # records

Utilize overflow blocks for insertions with "out of order" primary keys

Classes

id	name	number	date_code	start_time	end_time
1	Web	CSE135	TuTh	2:00	3:20
2	Databases	CSE132A	TuTh	3:30	4:50
3	**PL**	**CSE130**	**TuTh**	**9:00**	**9:50**
4	VLSI	CSE121	F	*null*	*null*

just inserted tuple



Block Size Selection?

- Big Block → Amortize I/O Cost

Unfortunately...

- Big Block ⇒ Read in more useless stuff! and takes longer to read

The future
(which has partly arrived)

- Entire (analytics) databases in RAM
- Multi-core CPUs
  - case of parallel query processing
- Non-Volatile RAM