


External Sorting

Chapter 13


3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 1



Why Sort?

- User wants query answers in some order
 - E.g., ORDER BY Age DESC
- First step to bulk-loading B+ Tree
- Eliminate duplicate records
 - SELECT DISTINCT
- Evaluate GROUP BY aggregate queries
- Sort-merge join algorithm (later)

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 2



Why bother?

- But we already know how to sort...
- New Problem: How to sort 100 GB of data with 1 GB RAM?
 - External Sort – Minimize disk access cost
 - Why not use virtual memory?

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 3

Cost of External Merge Sort

- Number of passes: $1 + \lceil \log_{B-1} \lceil N/B \rceil \rceil$
- Cost: $= 2N * (\# \text{ of passes})$
- E.g., with 11 buffer pages, to sort 1000 page file:
 - Pass 0: $\lceil 1000/11 \rceil = 91$ runs of 11 pages each, except the last run which is 10 pgs
 - Pass 1: $\lceil 91/10 \rceil = 10$ sorted runs of 110 pages each, last run is only 10 pages
 - Pass 2: **Sorted File!**

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 7

Number of Passes of External Sort

N (# of pages)	B=3	B=17	B=257
100	7	2	1
10,000	13	4	2
1,000,000	20	5	3
10,000,000	23	6	3
100,000,000	26	7	4
1,000,000,000	30	8	4

32K pg size, 32TB relation

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 8

Internal Sorting

- How to do the in-memory part of the sort?
- One idea:** Sort data in buffers using your favorite sorting algorithm (e.g., Quicksort)
- Alternative:** Try to increase the length of output sorted runs using replacement sort
 - Longer runs may mean fewer passes (less I/O)

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 9

Replacement Sort

- Start by reading pages from file until buffer is full
 - Maintain *current set* in memory
- Repeatedly pick smallest value from current set that is greater than largest value in output buffer
 - Write to output buffer (run)
- Start a new run when no value in current set larger than all values in output
- On average, produces runs of size $2B$

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 10

Replacement Sort (Example)

Input Current (3) Output

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 11

Blocked I/Os

- Single request to read a *block* of pages often cheaper than independent requests for each page – *Why?*
- Make each buffer a block of pages instead
 - Reduces cost per page I/O
 - Reduces fan-out during merge passes – Side effect?
 - First Pass: Each run $2B$ pages, $\lceil N/2B \rceil$ runs (where B is the size of the buffer pool in #pages)
 - Assuming we use replacement sort optimization...
 - Merge Tree Fanout: $F = \lceil B/b \rceil - 1$, b is block size
 - # passes: $\lceil \log_F \dots \rceil + 1$
 - In practice, buffer pools are large
 - most files still sorted in 2-3 passes

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 12

Double Buffering

Reduces response time. What about throughput?

- Overlap CPU and IO processing
- Prefetch* into shadow block.
 - Potentially, more passes; in practice, 2-3 passes.

B main memory buffers, k-way merge

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 13

Using B+ Trees for Sorting

- Scenario: Table to be sorted has B+ tree index on sorting column(s).
- Idea*: Can retrieve records in order by traversing leaf pages.
- Is this a good idea?**
- Cases to consider:
 - B+ tree is *clustered* **Good idea!**
 - B+ tree is *not clustered* **Could be a very bad idea!**

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 14

Clustered B+ Tree Used for Sorting

- Go to the left-most leaf, then retrieve all leaf pages
- Alt 1: Done!
 - # pages?
- Alt 2: Retrieving data records, each page fetched just once

> Faster than external sorting!

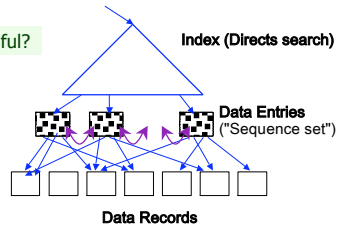
Why not scan the data file directly?

3/8/09 EECS 484: Database Management Systems, Kristen LeFevre 15

Unclustered B+ Tree Used for Sorting

- Alternative (2): In general, **one I/O per data record!**

When can this be useful?



3/8/09

EECS 484: Database Management Systems, Kristen LeFevre

16

Summary, External Sort

- Important operation
- Minimize disk I/O cost, use the (large) buffer pool:
 - Larger runs
 - Fewer merges
 - Blocked IOs
 - Double Buffering
- Choice of internal sort algorithm may matter
 - Pass 0: Run size B or 2B
- Can use indices
 - Clustered Index: Great! Always better than external sort
 - Unclustered Index: Use with caution
- Exercises: 13.1, 13.3

3/8/09

EECS 484: Database Management Systems, Kristen LeFevre

17
