

Example: Index Files

Consider a table with 2 Million records, stored using a fixed length format of 120 bytes/record, unspanned organization, Key field = 9 bytes, on a disk with block size = 8192 bytes = 8KB, seek time $s = 3$ ms (average), rotational delay $rd = 3$ ms (10k RPM) and block transfer rate of 1 ms/block, block pointer size = 8 bytes = 64 bits. Suppose the files are reorganized nightly so that the main file blocks are kept 70% full, and index blocks at 90%.

Main file: Blocking factor, $bfr = \left\lfloor \frac{8192}{120} \right\rfloor = 68$. 70% of 68 ≈ 48

Number of blocks of main file = $\left\lceil \frac{2M}{48} \right\rceil = 41667$.

If file occupies contiguous blocks of the disk, time to access the whole file (linear search) = $3 + 3 + 41667 \times 1$ ms ≈ 42 secs

If file is fragmented, then worst case time to access file

Assume contiguous blocks from head = $(3 + 3 + 1) \times 41667$ ms ≈ 4.8 mins

If file is sorted by key, time for binary search on key =

$$\left\lceil \log_2(41667) \right\rceil * (3 + 3 + 1) = 128 \text{ ms} = 0.128 \text{ sec.}$$

Primary index on key: Record size = $\underset{\text{(key)}}{9} + \underset{\text{(block ptr)}}{8} = 17$ bytes

$$bfr_i = \left\lfloor \frac{8192}{17} \right\rfloor = 481, \quad 90\% \text{ of } 481 \approx 433$$

No. of records of primary index = No. of records of main file = 41667

$$\text{No. of blocks of primary index} = \left\lceil 41667 / 433 \right\rceil = 97$$

Binary search using primary index, time to find a record based on key:

$$= \underbrace{\left\lceil \log_2 97 \right\rceil * (3 + 3 + 1)}_{\text{(Index file blocks)}} + \underbrace{1 * (3 + 3 + 1)}_{\text{(main file block)}} = 56 \text{ ms}$$

2nd level index has only 97 records, so it fits in 1 block of file.

Finding a record for a given key using a 2-level index:

$$\underset{\text{2nd index}}{1 * (3+3+1)} + \underset{\text{primary index}}{1 * (3+3+1)} + \underset{\text{main file}}{1 * (3+3+1)}$$

$$= 21 \text{ ms.}$$

Summary: If main file is ordered by key, time for search:
(and stored in contiguous blocks)

128 ms (binary search), 56 ms (primary index), 21 ms (2-level index).

If file is not sorted by key, we can build a secondary index for it.

Secondary index: No. of records = No. of records of main file = $2M$

$$bfr_i = 433 \text{ (calculated earlier). No. of blocks} = \left\lceil \frac{2M}{433} \right\rceil = 4619$$

Assuming that the file is stored in contiguous blocks of disk,

Linear search takes 42 sec.

$$\text{Searching with secondary index on key: } \left\lceil \log_2(4619) \right\rceil * (3+3+1) + 1 * (3+3+1) \quad \begin{matrix} \text{(index file)} \\ \text{(main file)} \end{matrix}$$
$$= 98 \text{ ms}$$

2nd level index is a primary index to the 1st level index.

$$\text{So No. of records of 2nd index} = \text{No. of blocks of 1st index} = 4619$$

$$\text{No. of blocks of 2nd index} = \left\lceil 4619 / 433 \right\rceil = 11$$

Time for searching on key using second-level index:

$$\left\lceil \log_2(11) \right\rceil * (3+3+1) + 1 * (3+3+1) + 1 * (3+3+1) \quad \begin{matrix} \text{(2nd index)} & \text{(1st index)} & \text{main file} \end{matrix}$$

$$= 42 \text{ ms}$$

3rd level index has only 11 records and fits in 1 block.

Search with 3rd level index accesses 4 blocks and takes 28 ms

Summary: 42 sec (linear search), 98 ms (secondary index), 42 ms (2nd index), 28 ms (3rd index)