CS 455: Principles of Database Systems

Review Guide 4: Performance

Your Name:

1. Consider a disk with a sector size of 512 bytes, 2000 tracks per surface, 50 sectors per track, a single 1-sided platter, and average seek time of 10 msec.
   (a) What is the capacity of a track in bytes? What is the capacity of the disk?
   (b) Give examples of valid block sizes. Is 256 bytes a valid block size? 2048? 51200?
   (c) If the disk platters rotate at 5400 rpm (revolutions per minute), what is the maximum rotational delay?
   (d) If one track of data can be transferred per revolution, what is the transfer rate?
2. Consider again the disk specifications from the previous question, and suppose that a block size of 1024 bytes is chosen. Suppose that a file containing 100,000 records of 100 bytes each is to be stored on such a disk and that no record is allowed to span two blocks.
   (a) How many records fit onto a block?
   (b) How many blocks are required to store the entire file?
   (c) How many records of 100 bytes each can be stored using this disk?
   (d) Assuming decode time and transfer time are negligible, what time is required to read a file containing 100,000 records of 100 bytes each sequentially?
   (e) Assuming decode time and transfer time are negligible, what is the time required to read a file containing 100,000 records of 100 bytes each in a random order? To read a record, the block containing the record has to be fetched from disk. Assume that each block request incurs the average seek time and rotational delay.

3. **[Disk Scheduling]** None of the disk-scheduling policies, except FCFS, is truly fair (starvation may occur). Explain why this assertion is true.

4. **[Disk Scheduling]** Suppose that a disk drive has 5,000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 2150, and the previous request was at cylinder 1805. The queue of pending requests, in FIFO order, is: 2069, 1212, 2296, 2800, 544, 1618, 356, 1523, 4965, 3681

   Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

   - FCFS
   - SSTF
   - LOOK
   - C-LOOK

5. **[Hashing]** Consider a hash file that using the function $h(k) = k \; mod \; n$, where n is an arbitrary positive constant and k is the search key for a tuple. Answer the following questions:

   (a) With regards to $h(k)$, discuss the pros and cons of a small n (as it approaches 1) versus a large n (as it approaches the total number of blocks taken up by this file).

   (b) A follow up to the previous question, the maximum length of bucket overflow is often an indication of the "goodness" of a hash function. Explain why.

6. **[Disk]** Hard disks store more sectors on the outer tracks than the inner tracks. Since rotation speed is constant, the sequential data transfer rate is also higher on the outer tracks. Seek time and rotational latency are unchanged. Considering this information, explain good strategies for placing files on disk with the following kinds of access patterns:

   (a) Frequent, random accesses to a small file.

   (b) Sequential scans of a large file.

   (c) Sequential scans of a small file.

7. **[Joins]** Assume that you want to join two relations R(A,B) and S(B,C). Relation R has attributes A and B, and relation S has attributes B and C. The two relations are stored as simple (unsorted) heap files. When would you prefer a hash join to a sort-merge join, and when would you prefer a sort-merge join to a hash-join?

8. **[B+Trees]** Show that the height of a B+Tree with degree d is worst-case $O(log_{\lceil d/2 \rceil} N)$, where N is the number of keys stored.

9. **[B+Trees]** ** Consider a B^+-Tree with $d = 4$, and suppose you're given a list of keys to insert: $K = (3, 5, 7, 9, 11, 13, 15, 17, 19)$. Suppose we want to try out a new splitting strategy: keep the first $\lceil d/2 \rceil - 1$ keys in the original node, and split off the rest. Show the final B+Tree if the keys in K were inserted in the given order. How does the tree differ from the one built using the splitting strategy we used in class?

10. **[Dynamic Hashing]** ** Repeat the above insertion sequence on an extendible-hash index. Each bucket holds only 2 tuples. Assume we use 5-bit keys, and that $h(k) = k$.

11. **[Bitmap Index]** Explain the significance of an attribute's *bin cardinality*. Discuss the pros and cons of low vs. high cardinality.

12. **[Bitmap Index]** Consider the 32-bit WAH bitmap-encoding scheme. Decode the following word: `0xC00000FF`.