Database Development and Design (CPT201)

Tutorial 1

Wei Wang
Department of Computing

Q1

- Given a disk with the following characteristics:
  - There are 2^{14}=16384 tracks per surfaces
  - There are 2^{7}=128 sectors per track
  - There are 2^{12}=4096 byte per sector
  - The disk rotates at 7200rpm; i.e., it makes one rotation in 8.33 milliseconds
  - To move the head arm between cylinders (tracks) take one milliseconds to start and stop, plus one additional millisecond for every 1000 cylinders travelled.
- Questions:
  - 1. what is the time to take one track movement?
  - 2. what is the time to move the head from innermost track to outmost track?

Q1 Answer

- 1.001ms: move head arm between tracks needs 1 ms plus 0.001ms for cylinder travelling

- 17.384 millisecond: 16384 tracks on a surface, so total cylinder travelling is 16.384ms, plus 1 ms, which is 17.384ms

Q2

- Given a disk with the following characteristics:
  - There are 2^{14}=16384 tracks per surfaces
  - There are 2^{7}=128 sectors per track
  - There are 2^{12}=4096 byte per sector
  - The disk rotates at 7200rpm; i.e., it makes one rotation in 8.33 milliseconds
  - To move the head arm between cylinders (tracks) take one milliseconds to start and stop, plus one additional millisecond for every 1000 cylinders travelled.
- Questions:
  - 1. Assume that there is no gap between sectors and each block occupies 4 sectors. what is the minimum time to read a block ?
  - 2. What is the maximum time to read a block?
  - 3. What is the average time?

Q2 Answer

- Best case: seek time = 0; rotational latency = 0
  - Time taken for one complete rotation: 1/(7200/60) = 0.00833s = 8.33ms
  - Reading 4 sectors only needs to rotate 4/128
  - *So the time is 8.33*4/128 = 0.2603 ms*

Q2 Answer cont'd

- Worst case:
  - Worst time: travel from innermost to outermost track; then rotate the whole track; then plus the read time.
  - 17.38 + 8.33 + 0.2603 (rotation latency: 8.33; transfer time: 0.2603)
- Average case:
  - Average time is "$\frac{1}{2}$" of the worst time except for the actual read time
  - (17.38+8.33)/2 + 0.2603

Q3

- Suppose that a relation called student holds 25,000 tuples, which are stored as fixed length and fixed format records. The length of each tuple is 350 bytes. The key attribute, student_ID, occupies 10 bytes and another attribute address occupies 50 bytes. The records are sequentially ordered by student_ID and stored in a number of blocks. Each block has the size of 4,096 bytes (i.e., 4 Kilobytes). Assume that a complete record or an index entry must be stored in one block.
  - How many blocks are needed to store the relation student?
  - Consider creating a primary index on the *student_ID* attribute. Each index entry contains a search key and a 10-byte long pointer to the records. Suppose the primary index is sparse (i.e., one index entry for one block), compute the number of blocks needed to store the index.

Q3 Answers

- Each tuple of student is 350 bytes. Each block at most holds ⌊4,096 bytes/350 bytes⌋ = 11 tuples (where ⌊ ⌋ indicates round down).
- There are 25,000 tuples, so ⌈25000 tuples/11 tuples per block⌉ = 2,273 blocks required (where ⌈ ⌉ indicates round up).

Q3 Answers cont'd

- Each index entry is 10 bytes for the key plus 10 bytes for the pointer (20 bytes in total).
- Each block at most can store ⌊4096 bytes/20 bytes⌋ = 204 index entries.
- There are 2,273 blocks in the *student* relation (answer of the previous question), so ⌈2,273/204⌉ = 12 blocks are needed.