CS 351: DATA ORGANIZATION AND MANAGEMENT - FALL 2011

HOMEWORK 1

QUESTION 1)

a) The hard disk that I choose for this question is the one I use in my personal desktop computer. It is Samsung HD502HJ and desired technical specifications are below:

Rotational Speed: 7200 rpm

Seek Time: 8,9ms

Rotational Latency: 4,14ms

Disk capacity: 500GB

Bytes per sector: 512 Bytes

b) The smallest unit of storing is block or sometimes called sector which constitutes tracks. Each sector of Samsung HD502HJ contains 512 bytes and a surface of the platter contains tens of thousands of tracks. The same track of each surface constitutes the cylinders. A cylinder contains the same indexed tracks of the different platter surfaces. The addressing modes for the sectors have been changed within years. CHS (Cylinder-Head-Sector) mode was the first addressing mode and then today LBA (Logical Block Addressing) is being used. Samsung HD502HJ use 48-bit LBA which allows a capacity of 144 petabytes for a hard disk drive. To sum up, the data are stored in sector in track which constitutes cylinders but the concept of cylinder is recently not used because LBA addresses each block sequentially so we don't need to think a physical cylinder structure.

c) Solid-state disk (SSD) drives don't contain any moving or rotating part. Instead, they use microchips. So they have an advantage in terms of loudness and resistance. Moreover, the rotational latency time for magnetic disk drives could not be observed in SSDs. The access time in SSD drives are less than magnetic disk drives, but the cost is higher in SSDs per gigabyte.

d) Samsung HD502HJ disk drive has a maximum transfer rate of 300MB/sec which I have never observed in my personal computer. The IBM 3380 has 3MB/sec which is hundred times less than Samsung's hard disk drives. In terms of access time, IBM 3380 has the average access time of 16ms and Samsung HD502HJ's access time is about 9ms. In 30 years, this value has been improved by about %56. Another huge difference can be observed physically since the size of IBM 3380 is very large and looks like an engine of a car. Also the price of the IBM 3380 was between $81,000 and $142,000. The price of the Samsung HD502HJ is differentiating between $50 and $90.

QUESTION 2)

For a sequential reading;

For bucket size of 50 blocks;

Sequential read time = s + r + 50*10*(ebt)

16 + 8.3 + 50*10*(0.84) = 444.3 ms = 0.44 s

For bucket size of 1,000,000 blocks;

Sequential read time = s + r + 10,000,000*(ebt)

16 + 8.3 + 10^7*(0.84) = 840,024.3 ms ~= 140 mins

For random reading;

For bucket size of 50 blocks;

Random read time = 50*(s + r + 10*ebt)

50*(32.7) = 1635ms = 1.63 s

For bucket size of 1,000,000 blocks;

Random read time = 1,000,000*(s + r +10* ebt)

10^6*(32.7) = 32,700,000ms ~= 9h

For infinitely large number of blocks per bucket:

Tr / Ts = Random read / Sequential Read = INF*(32.66) / (24.3 + INF*10*(0,84))

= 32.66 / 0.84 = 38.8

QUESTION 3)

250,000 * 200 = 50,000,000 bytes = 50MB is the files' size.

a) First, read/write head will move to F1 in order to read 10MB of the file. In total, head will move to read F1 for **5 times**, so there should be 5(s+r)'s caused by reading of F1. For each active record in F1, F2 should be traversed. So in each traversal first the head should move to disk to the F2. This will occur **200,000 times** since 200,000 of the records in F1 will be compared with F2. This process will repeat times **5 times** since there are 5 portions of records each 10MB in a 50MB file.

b) Since we read 10MB portions each time, we should calculate the block number of single 10MB portion.

1,000,000 / 2400 = 4166,666 is the block number for 10MB of data.

We repeat the same action for 5 times so;

5*(s+r+ 4166,666*ebt) = 5*(24.3 + 4166,66*0,84) = 17621,5 msec = 17,6 s

c) %20 of the records of F1 are marked as deleted and %50 of the active records are common with F2, therefore for the %40 of the data the whole F2 file should be scanned. T_F is the average time to reach to correct data. So;

(250,000*200 / 2400)*ebt) = 17.5s => T_F = 17,5 / 2 = 8,75

100,000 * 8.75 + 100,000 *17,5 = 30 days 9 hours

d) The size of each file is 50MB and %50 of the active records is common between these files. So 20MB will be written to the intersection file.

20000000 / 2400 = 8333,3 is the block size for 20MB data.

There will be 5 s+r's since writing will occur after processing every 10MB data of the files.

So the total time for writing to the intersection file is;

5*(s+r) + 8333,3*ebt = 5*24,3 + 8333,3*0,84 = 7121.5 ms = 7,12 s

e) If we have two disks while we can read from F1, we can also write to the F12 at the same time and this overlapping may gain us the time needed for writing to the intersection file, but since we have to traverse all the records in F2 one by one it will still take 30 days to do it. In this case, two disk drives may not differ from one disk drive. If we don't have any memory concern, we can first copy F2 to the second disk drive and we can delete the marked nodes as well. This will take;

(250,000*200/2400)*ebt + (200,000*200/2400)*ebt = 31,5. After this operation we may search two different records from F1 in different disk drives at the same time. It means we can decrease the time spent on traversal of F2 to half which will be 15 days. (by using the answer of **3c**) The extra operation time and the other times are negligible when compared to 15 days.

QUESTION 4)

a) To reorganize files, they should be read first and then written back without the data marked as deleted. So

(250,000*200/2400)*ebt + (200,000*200/2400)*ebt = 31,5 s. This value is the same for both of the files.

b1) With 200,000 records file size has been decreased to 40MB, so there should be **4** (s+r)'s for reading F1 and **4** (s+r)'s for writing.

b2) b = (200,000*200/2400) = 16,666

4*(s+r+4166,6*ebt) = 14 s

b3) (200,000*200 / 2400)*ebt) = 14s

100,000 * 7 + 100,000 *14 = 24 days 7 hours

b4) 4*(s+r) + 8333,3*ebt = 7,1 s

b5) It will behave as same as I explained in **3e.** The only change will be observed in the reading of F1. 200,000 records which are 40MB will be read from disk but it will be negligible when compared to reading of F2. For my suggestion about copying F2 to the second disk, the copying time will be smaller, but it is still negligible compared to reading time of F2.

QUESTION 5)

a) **1.** Read **M** MB of records from F1 to memory

2. For each record of F1, first check F2 if the same record is present there

If F2 contains this record do not write this record to union file.

If F2 does not contain the record, keep the record to be written to union file and repeat **step 2** for the next record of F1.

3. When all this process has finished for a **M** MB portion of F1, write the data corresponding data to union file on disk and read another **M** MB portion from F1 until all data is completed.

4. After the end of F1, directly write F2 to union file.

b) The equation of the whole operation is:

3*(n/Bfr)*ebt + ((1-p)*n/Bfr)*ebt + p*n*(n/Bfr*ebt/2) + ((1-p)*n*n/Bfr*ebt)

-3*(n/Bfr)*ebt: This term comes from one read of F1, one read of F2 and one write of F2 to union

-((1-p)*n/Bfr)*ebt: This term comes from write of F1 to union but only the non-common records

- p*n*(n/Bfr*ebt): This term comes from the search for common records between F1 and F2 in file F2.

-((1-p)*n*n/Bfr*ebt): This term is derived from the non-common searches of records of F1 in file F2.

While deriving this equation, s+r's are neglected since they are small numbers when compared to other terms. The better form of the equation is:

n*ebt / Bfr (4-p + p*n/2 + (1-p)*n)

c) We may overlap the writing time with the reading and we can increase the speed of the operation, but we still should make comparisons between two files so it would not be a huge gain, it is even a negligible gain.

References:

- http://www.samsung.com/latin_en/consumer/monitor-peripherals-printer/memory-storage/internal-storage/HD502HJ/AMC/index.idx?pagetype=prd_detail&tab=specification
- http://en.wikipedia.org/wiki/Cylinder-head-sector
- http://en.wikipedia.org/wiki/Logical_block_addressing
- http://etutorials.org/Linux+systems/red+hat+linux+9+professional+secrets/Part+VI+Appendixes/Appendix+B+Disk+Drives/Disk+Drive+Concepts/
- http://en.wikipedia.org/wiki/Solid-state_drive
- http://en.wikipedia.org/wiki/History_of_IBM_magnetic_disk_drives#IBM_3380[1]

[1] Solutions are due to Övünç Sezer.