CMPT 454 Assignment 1: HDDs and RAID

In this assignment you are to answer questions related to DBMS and secondary memory. This assignment is worth 6% of your final grade. If you feel you have to make any assumptions to answer any of the questions, please state these assumptions.

Marks allocated to questions are shown in []s.

Question 1: Hard Disks

Consider a hard disk with the following statistics.

- RPM – 5,400 = 0.09 rp*ms*, i.e. 11.11 ms for one revolution; there are 20 blocks per track (each block is 2 sectors), it therefore takes 11.11/20 = 0.56 ms to read a single block
- Time for the disk head to move from the innermost to outermost track - 12ms, assume that the disk head speed is constant
- Number of tracks on each surface – 12,000
- Number of (double sided) platters - 4
- Sector size – 4,096 bytes
- Sectors per Track - 40
- Block size - 2 sectors
- Tracks are labeled 1 to 12,000, with 1 being the outermost and 12,000 the innermost
- Platters are labeled 1 to 4 and their surfaces are labeled A and B.

1. What is the storage capacity of the disk (in bytes)? – 40 (sectors) * 12,000 (tracks) * 8 (surfaces) * 4,096 (bytes) = 15,728,640,000 bytes [1]
2. How many disk heads does the disk head array contain? – 1 per surface = 8 [1]
3. How many cylinders are there on the disk? – same as the number of tracks = 12,000 [1]
4. What is the maximum time to read a single block? Show the values for each of the components of the read time. – 12 (seek time) + 11.11 (rotational delay for one revolution) + 0.56 (transfer time) = 23.67 [3] (1 mark for each component)
5. What is the average time to read a single block? – 4 (seek) + 5.56 (rotational delay) + 0.56 (transfer) = 10.12 [3] (1 mark for each component)
6. What is the average time to read an entire track, assume that the track must be read starting from a particular block? Show the values for each of the three components of the read time. – Seek time as above, plus average rotational delay to wait for the start block then an entire rotation = 4 + 5.56 + 11.11 = 20.67 [3] (1 mark for each component)
7. What is the maximum time to read two (unrelated) blocks? – The blocks are on the inner and outermost tracks, and the disk head has to travel across the entire disk to service the first request, and in both cases a full revolution of the disk is required before the block can be read: (12 + 11.11 + 0.56) * 2 = 47.34 [2] (1 mark for answer, 1 mark for explanation)
8. What is the minimum time to read two blocks? – The blocks are adjacent on the same track, the disk head is on that track and at the position of the first block: 0.56 * 2 = 1.12 [2] (1 mark for answer, 1 mark for explanation)

9. What is the size in bytes of the largest data file that could be read without having to move the disk head? – the largest such file takes up all the tracks of the same cylinder of the disk: 8 * 8,192 * 20 = 1,310,720 bytes [1]
10. Why does defragmenting a hard drive improve performance? – because data is reorganized on the disc so that related data are located close to each other [1]
11. The question specifies that the disk head speed is assumed to be constant. Why is this assumption unrealistic? – because the disk head has to accelerate to reach its maximum speed and then decelerate to stop at the desired track [1]

Question 2 – RAID

Complete the table at the end of the question by inserting the time taken for the noted RAID systems and file access requests. Each RAID system is to contain ***four disks worth of data***. This means that each system will consist of *at least* four disks and, in some cases, ***one or more additional disks*** containing redundant data.

RAID Systems
RAID 0
RAID 1+0
RAID 4
RAID 5

Striping

- For RAID 0, RAID 1+0 and RAID 4, blocks of the file are striped in a *round robin* fashion across the data disks starting with disk 0

    - disk 0 - block 0 contains block 0 of the file
    - disk 1 - block 0 contains block 1 of the file
    - disk 2 - block 0 contains block 2 of the file
    - …

- For RAID 5, blocks of the file are striped in a *round robin* fashion across the system except that, for each set of same numbered blocks in the system's disks, one disk contains parity data. The parity data is also distributed across the disks in a *round robin fashion starting with disk 0*

    - disk 0 - block 0 contains parity data
    - disk 1 - block 0 contains block 0 of the file
    - disk 2 - block 0 contains block 1 of the file
    - …

Requests

Requests are specified as blocks of a file (whose first block is block 0) that is striped across the RAID systems.

1. Read blocks 0 to 31
2. Read blocks 4, 11, 17 and 22
3. Read blocks 1, 9, 14 and 17

4. Write blocks 0 to 31
5. Write blocks 4, 11, 17 and 22
6. Write blocks 1, 9, 14 and 17

All writes must read the data before writing it except for writes to redundant data which should only read data when necessary (that is, necessary to allow for the greatest amount of parallelism).

Metrics

- Your answer should specify time as a fraction of the time that a single disk would take to perform the same request. For example, a four data disk RAID 0 system can read blocks 0 to 31 of the file four times as fast as a single disk so write ¼ in the appropriate place in the table.
- Assume that it takes the same time to read or write any block, regardless of its position on the disk, such that time is to be measured in units of disk reads or writes. The components of those reads or writes (i.e. seek time, rotational delay and transfer time) should not be considered. To elaborate the previous example of reading blocks 0 to 31, a single disk would require 32 reads. The equivalent RAID 0 system would require 4 parallel reads of a sequence of 8 blocks each and would therefore require ¼ of the time as the single disk.
- The cost is to be specified as a multiple of the cost of a single disk, and only considers the number of disks required for the RAID system (not the RAID controller). For example, a four data disk RAID 0 requires four disks and is therefore four times the cost of a single disk so write 4 in the appropriate place in the table.
- When assessing the time taken to perform writes your answer should comply with the following rule: *data disks may not be written to at the same time as check disks*. This means that for writes no check disk should be written to during the write of any data disk though data disks may be written to in parallel with other data disks and check disks may be written to in parallel with other check disks. This rule does not mention reads so any read may be performed in parallel with any read *or write*. RAID systems with distributed parity do not have dedicated check disks. For such systems if a disk is writing (DB) data it is counted as a data disk, whereas if it is writing parity (check) data it is counted as a check disk.

Solution

Striping – not part of the solution (random requests – **4**, **11**, **17**, **22** and **1**, **9**, **14** and **17**)

RAID 0, 1+0, 4						RAID 5				
disk 0	**disk 1**	**disk 2**	**disk 3**	**disk 4**		**disk 0**	**disk 1**	**disk 2**	**disk 3**	**disk 4**
0	**1**	2	3	**check**		check	0	**1**	2	3
4	5	6	7	**check**		4	**check**	5	6	7
8	**9**	10	**11**	**check**		8	**9**	**check**	10	**11**
12	13	**14**	15	**check**		12	13	**14**	**check**	15
16	**17**	18	19	**check**		16	**17**	18	19	**check**
20	21	**22**	24	**check**		**check**	20	21	**22**	23

RAID 5 – sequential write, read all disks and write check disks

disk 0	disk 1	disk 2	disk 3	disk 4		disk 0	disk 1	disk 2	disk 3	disk 4
check	0	1	2	3		check	0	1	2	3
4	check	5	6	7		4	check	5	6	7
8	9	check	10	11		8	9	check	10	11
12	13	14	check	15		12	13	14	check	15
16	17	18	19	check		16	17	18	19	check
check	20	21	22	23		check	20	21	22	23
24	check	25	26	27		24	check	25	26	27
28	29	check	30	31		28	29	check	30	31

1 2 3 4 5 6 7 8 9 - 10 11 12 13 14 15 16

Solution

Request	RAID 0	RAID 1+0	RAID 4	RAID 5
Read 0 to 31	¼	1/8	¼	¼
Read 4, 11, 17 and 22	¼	¼	¼	¼
Read 1, 9, 14 and 17	¾	½	¾	½
Write 0 to 31	¼	**17/64**[1]	17/64[2]	**1/4**[3]
Write 4, 11, 17 and 22[4]	¼	3/8[5]	1[6]	½[7]
Write 1, 9, 14 and 17[4]	¾	7/8[8]	1[9]	¾[10]
Cost	4	8	5	5

½ mark for each correct answer

1. Read data disks and mirror disks to read all 8 blocks (1), then write 6 sets of 4 data blocks while reading from mirror disks (6). Write remaining 2 sets of 4 data blocks then all mirror disks (10)
2. Read first 4 blocks (1), read second 4 blocks and write check data relating to previous blocks – repeat (7), write data blocks (8), write remaining check block (1).
3. Read data (8), can write to check data block for 7 of these (see picture), then write last check block and 7 sets of data blocks (five at a time)
4. For the random writes check blocks have to be read to compute the new parity. Note that check disk reads can be performed in parallel with data disk reads and writes.
5. Read then write all data in parallel, repeat for mirror disks. Perform reads of mirrors in parallel with operations on data disks.
6. Read then write all data in parallel, read and write 4 check blocks, 2 of the reads can be done in parallel with initial read and write.
7. Read then write all data in parallel, read and write check blocks in parallel.
8. Read 1, 9 (mirror) and 14 (1), write 1, 14 and read 17 (mirror) (1), write to data disks twice (2), then mirror disks three times (3).
9. Data is read and written on two disks (1 and 2) and is on three blocks of disk 1. The constraint is that the check disk must be read and written to four times (8). However, all reads and two of the writes can be combined with the data disk operations – the writes being paired only with reads.
10. Data is read and written on two disks (1 and 2) (2 + 2). Check data on disks 0, 3 and 4 can be read in parallel but check data on 2 cannot. So requires and additional (1 + 1) to read and write check data.

Question 3 – Disk Scheduling

There are four algorithms that are variations of the elevator algorithm, SCAN, LOOK, C-SCAN and C-LOOK.

SCAN – goes to last track on disc (regardless of whether or not there is a request on it), satisfying requests en route. Then returns to first track, satisfying requests. [2]

C-SCAN – as SCAN except that it does not satisfy requests on the return trip to the first track – this movement is therefore very fast as it can accelerate to max speed. [1] This makes the algorithm fairer in that it satisfies requests from any track at approximately the same frequency [1]

LOOK – as SCAN except that it stops at the track with the last request before reversing. [1] Therefore wastes less time. [1]

C-LOOK – as C-SCAN except that it stops at the track with the last request before reversing. [2]

Briefly describe each algorithm, noting the differences between the other algorithms and the effect on performance. [2 marks each]

Assessment

The assignment is out of 40. Marks are assigned as follows:

- Question 1 – 19
- Question 2 – 13
- Question 3 – 8