CMU 18–746 **Storage Systems** **25 Feb 2009**

Spring 2009 **Exam 1**

Name: ________________________________

Instructions

There are four (4) questions on the exam. You may find questions that could have several answers and require an explanation or a justification. As we've said, many answers in storage systems are "It depends!". In these cases, we are more interested in your justification, so make sure you're clear. Good luck!

If you have several calculations leading to a single answer, please place a $\boxed{\text{box around your answer}}$.

Problem 1 : Short answer. [56 points]

(a) Imagine a disk with 500 KB per track and a rotation speed of 6000 RPM. Once positioning is complete, would you expect the disk to be able to read approximately 5 MB off the media in 100ms? ("Yes" or "No") Explain why or why not.

No. It would take longer because of cylinder and head switches.

(b) Modern disk drives have a vendor specified mean time between failures (MTBF) of over 1,000,000 hours. But, most disks are expected to be used for less than 45,000 hours (approximately 5 years). Given these facts, how is a disk whose MTBF is 2,000,000 hours better than one whose MTBF is 1,000,000 hours?

The expected rate of failures would be 50% lower.

(c) Dr. Smith saves every email he receives by appending it to a large file. Over the last year, the file has grown to 1 GB in size. In running a program to scan the file, from beginning to end, he notices that the read bandwidth is less that 5 MB/s. But, his system's specifications indicate that the disk can provide 50 MB/s. Give one likely explanation for the low observed bandwidth in reading the file.

The blocks of the file may not be located sequentially on the disk, because of fragmentation or poor allocation decisions. In this case, the scan on the file would result in multiple disk positioning delays, reducing the effective bandwidth.

(d) Some disks' firmware discards data from the on-board buffer/cache after it has been sent to the host that requested it, rather than retaining it in the cache. Explain how the firmware engineers might justify this approach.

Typically, host buffer caches are considerably larger than disk on-board buffer/caches, and hence almost all reuse hits occur there. Thus, retaining potential reuse hit data in the on-board buffer/cache does not have any added advantage.

(e) Most systems allow a program to create a "symbolic link" (a.k.a. a "soft link") to a file, even if that file does not currently exist. Why doesn't the target file have to exist in order to create the symbolic link?

A symbolic link file contains a pathname for a target, rather than any direct link to that target. So, the symbolic link contents are not affected by initial presence or absence of that target.

(f) Some modern disks perform write-back caching, wherein write requests are reported complete once the corresponding data is transferred from the host into the on-board RAM. Dr. Jones understands that this can result in the disk firmware's disk scheduler reordering the media writes in order to improve efficiency, which can reverse with the file system's ordering of metadata updates. But, he proposes that changing the file system to use write-ahead logging to protect its metadata integrity solves the problem. Is he correct? (”Yes” or ”No”) Explain (briefly).

No. With write-ahead logging, one must ensure that updates to the log are written to the disk before the corresponding actual updates. With the proposal, there is still a possibility that the disk might reorder updates and write the actual updates before writing the updates to the log.

(g) An engineer designing a new mobile phone is arguing that it should be equipped with a small disk drive, instead of a Flash-based SSD, since random access performance is not a concern for the expected workload. Give an argument in favor of the SSD, even in this environment.

Flash-based SSDs use considerably less energy. Some also argue that Flash-based SSDs are more robust to shock, though there is less data to support this position.

Problem 2 : Finding service time. [20 points]

Find the service time for the following request stream to a very simple disk that has 10000 cylinders, 8 surfaces, 200 sectors per track, and rotates at 10000 RPM. In this case, LBNs are mapped directly to PBNs, requests are serviced in FIFO order, and seek time is a linear function of cylinder distance.

$$\text{seek\_time(ms)} = 0.0006 \times \text{cylinder\_distance} + 3$$

(Remember that a linear seek curve is not realistic, but is a simplification for this problem.) Assume that the disk starts at LBN 0 and that head switches are instantaneous. The layout mapping is shown in Figure 1.

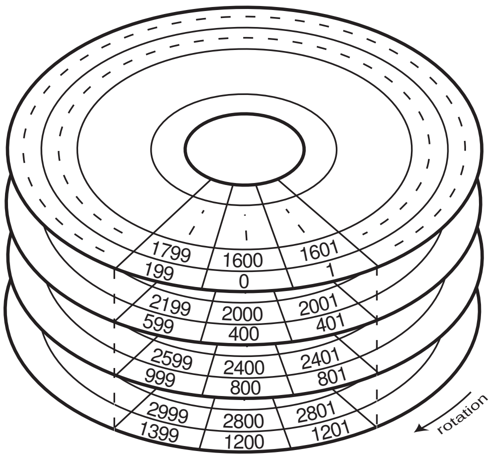

Figure 1: *LBN mappings onto physical sectors on the disk.*

The request stream for which you should compute the service times is shown in Table 1.

Request Stream	
LBN	Size (sectors)
0	10
4800600	20
12800200	10
6400400	20

Table 1: *Request stream.*

(a) Compute the cylinder, surface and offset for the first LBN of each request in the request stream (i.e. for LBN 0, 4800600, etc.)

With some simple formulae we can find for any LBN its surface, cylinder, and offset.

$$\begin{aligned} \text{surface} &= \frac{LBN}{200} \,\%\, 8 \\ \text{cylinder} &= \frac{\text{LBN}}{1600} \\ \text{offset} &= \text{LBN} \,\%\, 200 \end{aligned}$$

Now we can assign a physical location to each LBN:

LBN	surface	cylinder	offset
0	0	0	0
4800600	3	3000	0
12800200	1	8000	0
6400400	2	4000	0

(b) Calculate the total service time for the request stream by filling in the table 2. Time (ms) shows the time at which the head is at the given LBN. The LBN is mapped to a cylinder, surface, and offset within the track. The “next event” field shows which part of servicing the request (seek, rotation, or transfer) takes place next.

We note that a 10000 RPM disk takes $\frac{1}{10000} \times 60 \times 1000 = 6$ *milliseconds to make a full rotation.*

Now we can apply the service time formula:

$$\begin{aligned}
T_{service} &= T_{seek} + T_{rotation} + T_{transfer} \\
T_{seek} &= 0.0006 \times \text{cylinder\_distance} + 3 \\
T_{rotation} &= \frac{\text{sectors to rotate}}{200} \times 6\text{ms} \\
T_{transfer} &= \frac{\text{sectors to transfer}}{200} \times 6\text{ms (same as rotation!)}
\end{aligned}$$

It is important to remember that during the seek, the disk continues to rotate and so we need to take into account the amount that the disk has rotated during the seek and add in any additional rotational latency this introduces.

time (ms)	LBN	cylinder	surface	offset	next event
0	0	0	0	0	transfer 10 sectors
0.3	10	0	0	10	3000 cylinder seek
5.1	4800770	3000	3	170	rotational latency
6	4800600	3000	3	0	transfer 20 sectors
6.6	4800620	3000	3	20	5000 cylinder seek
12.6	12800220	8000	1	20	rotational latency
18	12800200	8000	1	0	transfer 10 sectors
18.3	12800210	8000	1	10	4000 cylinder seek
23.7	6400590	4000	2	190	rotational latency
24	6400400	4000	2	0	transfer 20 sectors
24.6	6400420	4000	2	20	done

Table 2: *Timetable of events.*

Problem 3 : More short answer. [24 points]

(a) A file system engineer implementing a log-structured file system noticed that using 128 KB segments instead of 64 KB segments nearly doubles file system write throughput. Explain why.

Writing 128 KB segments instead of 64 KB segments increases the disk transfer time slightly, but not the positioning delays that constitute most of the disk service time. Hence, the positioning delays get amortized over twice data transfer, roughly doubling write throughput.

(b) Imagine a new file system designed for a hybrid storage system that consists of one disk and one Flash SSD. Imagine that the workload consists of two types of files: (1) files that are randomly read in small units with no locality, and (2) large files that are read and written sequentially. Which files would you suggest assigning to each device? Explain.

For (1), Flash SSD. Absence of mechanical components gives Flash SSD much higher throughput for small random reads.

For (2), disk. The read/write bandwidth of typical Flash SSDs is comparable to that of disks, but the cost per byte is considerably higher.

(c) Imagine an inode structure that uses 12 direct block pointers, 1 indirect block pointer, and 2 double indirect block pointers. With a 4 KB block size and 64-bit unsigned integers for block pointers, what is the largest file size?

There are (4 KB / 8 bytes) = 512 block pointers in each indirect block. So:

- Direct: 12 blocks $\times$ 4 KB/block = 49,152 bytes = 48 KB;
- Single: 512 $\times$ 4 KB = 2,097,152 bytes = 2 MB;
- Double: 2 $\times$ 512 $\times$ 512 $\times$ 4 KB = 2,147,483,648 bytes = 2 GB;

For a total of 2,149,629,952 bytes (2.002 GB).

Problem 4 : Instructor trivia. [up to 2 bonus points]

(a) Which instructor has kids?

Greg

(b) Which instructor is Canadian?

Garth

(c) Which instructor will take a new job in May?

Zoheb. (The others do not plan to take new jobs, to be precise.)

(d) Which instructor most needs an obnoxious stopwatch alarm to use during their lectures?

Multiple correct answers here ;).

(e) Where (city, not company) should Zoheb take a job after completing his INI M.S. program?

Any of the following: Nirvanna, MI; Paradise, MI; Bird-in-Hand, PA; Happyland, OK; Happy, TX; Celebration, FL; Panacea, FL.... or, glorious Pittsburgh ;).