CMU 18-746/15-746 **Storage Systems** **23 Feb 2011**

Spring 2012 **Exam 1**

Name: _____

Instructions

There are three (3) 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 box around your answer.

Problem 1 : Short answer. [48 points]

(a) Imagine a system in which two ext2 file systems are mounted at /home and /home2. In such a system, can a hard link from /home/foo to /home2/foo be created? Explain your answer.

No. A hard link associates a name within a directory with an inode number within the same file system.

(b) Fred modifies his file system software to use the TRIM command whenever a file is deleted. Explain why Fred expects to observe higher write performance when using his modified file system (instead of the pre-modification version) on a Flash-based SSD.

The TRIM command tells the SSD to treat specified blocks as "not yet written", such that it is unnecessary to preserve their values when cleaning regions to allow subsequent writes. Thus, the SSD has less work to do in supporting writes.

(c) A good friend who administers a small storage system tells you that he has never had a disk fail, and he explains that it is because he buys disks with MTBF values of 3 years and always replaces them after only two years. Is his explanation correct or has he simply been lucky to not (yet) have a disk failure? Explain.

Just lucky. The MTBF is not an expected lifetime, but a statistical indication of failure rate for devices of the given type.

(d) Imagine a file system that uses synchronous writes (for update ordering) to protect the integrity of its metadata. Explain why, after a crash occurred during a rename operation, a file that I requested be renamed from one directory to another could end up with two names: one in each of the directories.

Most file systems will ensure that the new name is persistent before removing the old name in order to avoid having neither name after a crash.

(e) Imagine an inode with NULL in its first 9 block pointers and a non-NULL value in the tenth. Using only standard system calls, how could a user have caused the file's inode to be in this state? Explain.

The user could have used the seek() system call to move the "current file offset" to the tenth block and then the write() system call to write one or more bytes.

Just as a side note: one person answered that it might be the data of a very small file being stored in the block pointer space of the inode, which is a concept known as "immediate files". This was not the answer we were anticipating, but we did accept it.

(f) Julio implemented a RAID-4 disk array system for use in digital video recorders (DVRs) whose primary workload is writes of 128 KB in size. If there are 9 disks in the array, what is a good configuration value for the stripe unit size? Explain.

One valid answer would be "16KB", if justified by saying that every 128KB write would then update an entire stripe and thus require no reading of unmodified data in order to compute new parity data.

Another valid answer would be "a multiple of 128KB", if justified by saying that one should ensure that all writes go to a single one of the data disks in order to amortize the positioning time as much as possible for these "small" writes.

Problem 2 : More short answer. [48 points]

(a) In designing a Flash-based storage system for your data center, you are told by your boss that you should go with triple level cell (TLC)-based Flash products, because they are much less expensive. Explain why that might be a bad decision.

TLC-based Flash is also less reliable and less durable (fewer writes before no longer functional). As such, it may require higher levels of fault-tolerance and higher rates of replacement.

(b) Your boss is excited about a new line of disk drives that support write-back caching, expecting that they will significantly enhance performance. Is he likely to be correct? Explain your reasoning (and specify any relevant assumptions).

Expected answer: No, because higher levels of the system (e.g., the file system or database) will do write-back caching and thereby make writes a background activity for which the reduced latency is not important.

We also accepted: Yes, because the on-disk write-back caching can lead to effectively longer queues that in-firmware disk request schedulers exploit to increase write throughput.

(c) Disk drives use very powerful ECC on each sector to ensure that incorrect data will not be returned from the media. Describe what a disk tells its host system if the ECC check and correction computation fails.

The disk reports that the logical block number (LBN) cannot be read. It will continue to respond to reads of that LBN with the same error until the next time that that LBN is written.

(d) Imagine a mirrored pair of disks, where each disk can service 100 I/Os per second. Given a workload that issues requests at an exponentially distributed rate w/mean 20 I/Os per second, what is the average disk response time if every request is a read? (State any assumptions.)

Assuming that the read requests are divided between the two disks evenly and blindly (e.g., without regard to current queue length or disk head position), the average response time will be 11.1ms.

$T_s(servicetime) = \frac{1000ms}{100} = 10ms$

$util = \frac{20}{2 \times 100} = 0.1$

$T_q(queuetime) = T_s \cdot \frac{util}{1-util} = 1.1ms$

$T_r(responsetime) = T_q + T_s = 11.1ms$

(e) Some modern disk-based file systems prefetch 64 KB of data when they observe sequential read behavior for an open file. Your boss argues that the prefetch size should be raised to 1 MB. Do you agree or disagree? Explain your position.

Agree. The data transfer time for 1MB is still less than an average disk positioning time and would much more effectively amortize positioning overheads.

Few students got this question (which means that it will mostly be handled by a grading curve), and there were some recurring issues worth highlighting.

Many students seemed to focus on in-disk prefetching rather than the filesystem-level (FS-level) prefetching discussed in this question. This resulted answers relating to disk buffers being too small (the FS prefetches data into main memory) or going past the end of a track, neither of which are particularly relevant to FS-level prefetching. It also seemed responsible for answers suggesting that useless data would get prefetched in the case of small files, which does not happen for FS-level prefetching because the FS knows the exact file size and only prefetches that file's data (in most FSs).

We did give partial credit for answers that talked about indirect blocks not being sequential to direct blocks. A file system will generally try hard to have lengthy sequential runs of disk locations for data within sizable files, and much more than 1MB is usually referenced by a given indirect block of pointers. When it fails to do so, most FSs will prefetch only until the next non-contiguous LBN, rather than generating multiple reads of non-contiguous disk regions.

(f) You are designing a redundant disk array using Flash-based SSDs instead of traditional disk drives. If the primary workload to be supported is small random writes, will the performance impact of choosing RAID-5 instead of mirroring be greater than or less than when using traditional disks? Explain you answer.

Less. Unlike with traditional disks, SSD reads are significantly faster (on average) than SSD writes. As such, it is not as much more work to do two read-modify-write sequences to update data and parity (for RAID 5) than to just do two writes (for replication).

Many students incorrectly pointed at write amplification generally as an issue. But, as some students correctly noted, both mirroring and RAID-5 will involve writes to two SSDs (for a small write), with those two writes being the same size. So, the write amplification would be the same, for small writes.

A couple of students offered this acceptable specific answer related to write amplification: Greater, because the striping of RAID-5 could cause more SSDs to be involved in writing smaller (stripe unit sized) chunks of data, perhaps inducing more write amplification.

Note: this question asks about mirrored-SSDs vs. RAID5-SSDs, not mirrored-traditional vs. RAID5-SSDs. A few students made this mistake, leading to answers that did not relate to the question being asked.

Problem 3 : Bonus questions. [up to 2 bonus points]

(a) Name one instructor who is not present in class today (3/5/2012).
Prof. Garth Gibson

(b) Name two 746 TAs for this semester.
Kai Ren, Wittawat Tantisiriroj, Yifan Wang

(c) What is Carnegie Mellon's mascot?
Scotty the Scottish Terrier

(d) How many projectors does Prof. Ganger use when giving 746 lectures this semester?
Two