CMU 18-746/15-746 **Storage Systems** **4 Mar 2013**

Spring 2013 **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 file system that uses synchronous writes (for update ordering) to protect the integrity of its metadata. If such a system crashes in the middle of a rename operation that moves a file from one directory to another, is it possible that both names exist and refer to the same file? Explain why or why not.

Yes. The traditional approach proceeds in four steps: increment the file's link count, write the new entry into the target directory, remove the old entry from the source directory, and decrement the link count. If the system crashes after writing the new entry but before removing the old entry, both will be present.

(b) Fred modifies his file system software to use the TRIM command whenever a file is deleted. Explain why his colleague Alice tells him to expect no performance improvement when using his modified file system (instead of the pre-modification version) on a traditional disk.

A traditional disk does not have to do background cleaning to coalesce free space into regions that can be erased (for use by subsequent writes). So, it doesn't help to use TRIM to inform the device that some written data will never be read (and can therefore be discarded instead of copied during cleaning).

Although it is not what we were looking for, we gave partial credit for answers that noted that some implementations of TRIM take a variable and sometimes long time to execute the TRIM command, so any benefit of more effective background cleaning is perhaps more than lost by the extra, slow commands.

(c) Imagine that you work for a large Internet services company that has 100,000 disks in its data center. If each set of 10 disks is maintained as a RAID-5 array, and each disk has an MTBF of 100 years, how many data loss events would you tell your boss to expect in a one-year period (assume no rebuild)?

$MTBF_array = (100 years/10) + (100 years/9) = 21 years$ (Recall that the question specifies "assume no rebuild".)

The expected number of array failures (data loss events) in one year would be $10000/21$*, which is approximately 500.*

(d) Imagine a redundant disk array that performs regular scrubbing to find defective sectors. When a defect is encountered, how can the disk array controller fix the problem?

Use the redundancy to determine the value of the defective sector(s) and then write those disk blocks to the same LBNs. The disk can then remap those LBNs to different, non-defective sectors.

(e) Which change would you expect to more significantly improve storage response times: doubling the number of storage devices or replacing the existing devices with the same number of devices that each perform commands at twice the speed? Explain your answer.

Using the faster devices. More servers only reduces queueing delays, while faster devices reduces both queueing delays and actual service times, and always reduces queueing delays at least as much as the "double number of servers" option. (If only it were always an option ;))

(f) Fred suggests doubling the file system block size as a way of increasing the maximum file size without changing the existing inode structure. Why would such a change result in the maximum file size being *much larger than double* the original maximum file size?

Assuming that the file system uses indirect blocks, each indirect block will hold twice as many pointers to blocks that are each twice as big. Double indirect blocks could point to twice as many single indirect blocks that each point to twice as many blocks that are twice the size, and so on.

Problem 2 : More short answer. [40 points]

(a) Imagine that you have implemented a file server that uses Flash-based SSDs to store its files. As you test it, you observe that request service times are usually very good, but occasionally an SSD takes a much longer than average time to complete a request. What is the most likely cause of such slow requests?

The "clean and erase" process can delay some requests, especially requests that are writes and need to be written to the region being erased.

(b) Imagine that you have designed a disk with 16MB a cache. If the disk performs no prefetching into its cache, but does retain data requested by the host, approximately what read hit rate would you expect in the cache? Explain your answer.

Zero. I would expect the host cache to be much bigger than 16 MB, and any reads that could hit in the disk's cache would instead be handled by the host cache. Only host cache misses will get sent to the disk.

(c) Fred has devised a clever file placement algorithm that reduces disk seek time by one-half. Alice warns him, however, that he should expect disk service times will only decrease by approximately 25%. Assuming small I/O requests, is Alice correct? Explain your answer.

Alice. Disk service time for small I/Os is seek + rotational latency, which are approximately equal when there is no locality. Cutting half (the seek) of the service time in half will yield approximately a 25% improvement.

(d) Imagine a RAID-5 disk array with 11 disks that uses a 16KB stripe unit. If you are configuring a log-structured file system to use the array, what is a good setting for the log segment size? Explain your answer.

16000 KB might be one good choice. A good setting should be a multiple of the amount of data in one full stripe, which is 160KB. Since entire log segments are written as a unit, a value large enough to amortize disk positioning times is also important, so long as the large segment size is not a problem for the higher-level software.

(e) Imagine a RAID-5 disk array that has spare disks and can rebuild a failed disk's contents onto a spare, after a disk failure. Would you expect it to be a major reliability concern if you configured the rebuild process to take 2 days, instead of allowing it to proceed as fast as possible (2 hours)? Explain your answer.

The reasoning is the key here.

No, because the probability of a second failure occurring during the longer rebuild period is still much lower than the requirements of many deployments, when faced with the performance consequences of the faster rebuild.

Yes, because in an environment with many disk arrays, the increased probability can be significant enough to cause more data loss events than are acceptable.

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

(a) Name one instructor who is not present in class today (3/4/2013).
Greg. (Prof. Ganger)

(b) Name two 746 TAs for this semester.
Any two of Chinmay Kamat, Jin Kyu Kim, and Pavan Alampalli

(c) Does Carnegie Mellon have a football (”American football”) team?
Yes.

(d) What mailing list should you use for technical questions about a 746 lab?
746-staff@lists.andrew.cmu.edu