CMU 18-746/15-746 **Storage Systems** **24 April 2013**

Spring 2013 **Exam 2**

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 $\boxed{\text{box around your answer}}$.

Problem 1 : Short answer. [48 points]

(a) If the most common quorum consensus replicated database size is 3, what is the second most common quorum consensus replicated database size? Explain your answer.

ANSWER: 5, so that two failures can be tolerated. (Reminder: Three is the minimum size to tolerate one failure.)

(b) Early Hadoop file system (HDFS) developers are very happy that they used only replication for stored files, even though the system primarily supports large files that are read-only. Why are they so happy, despite the fact that HDFS workloads are a great match for parity-based protection (e.g., RAID-5), which would be more space-efficient and likely no slower? Explain your answer.

ANSWER: The implementation complexity is lower. Parity-based protection is well-known, but there are many corner cases in its implementation, especially in a distributed system setting. The relatively small Hadoop team was happy to have the simpler replication-only implementation to develop (and debug ;)).

(c) Imagine a workload in which a client opens a large file, modifies every block of that file once but does so in a random order, and then closes the file. Would you expect performance for this workload to be higher for AFS or for NFS (version 3)? Explain your answer.

ANSWER: The explanation is the key to getting credit for a question like this, and no answer is accepted without a clear and well-reasoned explanation.

The answer we had in mind was AFS, because it avoids two issues faced by NFS version 3 (according to the specification): per-block write-through to the server and synchronous per-write disk writes (which would be to non-sequential disk locations, in this case). An AFS client would not send the new data to the server until the file is closed, and it could send it all together and sequentially.

Another answer that works is that NFS version 3 could be faster, if the server uses NVRAM (as many do) or just ignores the synchronous write requirement (as some do). In this case, the disk updates could be overlapped better with the other work, by starting sooner, rather than waiting to do it all at the end (with the close).

(d) Imagine a Google file system instance configured to use the default of 3 replicas for each block. After a server fails, there will only be two replicas of many blocks. Joe claims that the Google file system can restore full redundancy of all blocks faster than would be the case with traditional 3-way replication (in which sets of three servers contain identical sets of blocks). Is he right? Explain your answer.

ANSWER: Yes, he is right. Since the each block is replicated on a different three servers, the two remaining replicas of the blocks that were stored on the failed server are on different pairs of still-functioning servers. Thus, recovery of full replication (i.e., a third replica) can be done for many blocks in parallel... each going from a different source server and to a different destination server. With traditional 3-way replication, only the two partner servers have copies of blocks that were stored on the failed server, meaning that only they can send data to the one replacement server.

(e) Imagine a distributed file system in which every client request first goes to a dedicated metadata server (MDS) and then to one or more data servers. If the workload consists of requests that take 10ms at the MDS and 90ms divided evenly among all data servers, what is the maximum speedup that can be achieved by increasing the number of data servers beyond the default setting of 1?

ANSWER: 10X. The data server time can be made arbitrarily small, but the MDS time remains unchanged. As per Amdahl's law, that non-parallelized portion (10ms out of 100ms total) bounds the maximum speedup.

(f) Joe has discovered that his file server supports snapshots, and so he has decided that regular backups onto magnetic tape are no longer required. Do you agree? Explain your answer.

ANSWER: No, he should keep doing backups. The snapshots will provide no benefit if the file server itself fails, since they are stored on the same storage device(s) as the data they protect... both would be lost if the device(s) fail, unlike backups stored elsewhere (e.g., tape).

Problem 2 : More short answer. [48 points]

(a) Joe recently discovered that the Carnegie Mellon system administrators are considering moving half of the AFS volumes in his home directory to a different server. He is very concerned that he won't be able to find all of his files. Should he be concerned? Explain your answer.

ANSWER: No, no reason for concern. AFS uses a volume location database, whenever traversing an AFS volume mount point, to determine which AFS server stores the volume's files. So, Joe can continue to access his files exactly as he had been.

(b) Imagine two users trying to use a shared AFS file server to allow one of them to talk to the other: both users open the same file, then one user writes text to it, and the other user reads the contents of the opened file. After the first user finishes typing, he tell the second user to read the contents, but the second user doesn't see the new text. The first user realizes that he needs to close the file, so he does. Explain what additional action the second user needs to take, before he will see the new text, and why.

ANSWER: The second user needs to re-open the file, since AFS's semantics mean that he will not see the new content in his existing open file session. (Assuming that it is a smallish file, in the case of more recent versions of AFS.)

(c) Imagine a distributed storage system that provides virtual disks to iSCSI clients by having a metadata server maintain a block map for each virtual disk, indicating which servers store each of its blocks. Upon discovering that the metadata server load is too high, Joe has decided to change the system to support multiple metadata servers. How could Joe distribute responsibilities among the metadata servers so as to avoid the need for synchronizing them via locks? Explain your answer.

ANSWER: Assuming that the metadata server allocates the blocks in addition to storing the block maps, then both responsibilities must be partitioned. Since each virtual disk would be separate, responsibility for storing the block maps can be partitioned by simply giving each metadata server responsibility for a different set of virtual disks. Responsibility for space allocation can be partitioned by giving each metadata server ”ownership” of a distinct fraction of the allocatable blocks.

If one assumed that the data servers allocate blocks based on a tuple of virtual disk ID plus logical block number, then only the second responsibility (storing the block maps) is necessary. But, to receive credit, one would need to have stated this assumption.

(d) Joe writes a distributed application in which each process consists of a loop in which it opens the one shared file, obtains a lock (via a special lock server dedicated to his application), reads the file's contents, updates the contents, releases the lock, and closes the file. When he runs all of the processes on the same machine, it works fine. When he runs the processes on different machines, using an NFS (version 3) file, it sometimes produces the wrong final result. Explain the problem for the NFS case.

ANSWER: NFS version 3 guarantees only that other clients will see a given client's updates within 30 seconds. So, even though the application uses proper locking to ensure that processes will read-then-update the file, one at a time, it cannot guarantee that processes will read the most recent updates from processes on other machines.

(e) Deduplication is a popular feature in disk-based backup systems. Joe insists that the benefits of deduplication are much larger for physical backups than for logical backups. Do you agree? Explain your answer.

ANSWER: Yes, because a physical backup makes a copy of the raw storage, independent of which blocks have been modified, as opposed to just modified files like a logical backup. So, a physical backup will usually include much more unmodified capacity, which will be identical to the previous backup (and the one before that and the one before that...).

(f) The GFS master keeps chunk location information only in main memory, even though it uses write-ahead logging and checkpoints on persistent storage for other file system metadata (like the namespace and inodes). So, the master does not retain the chunk location information across reboots. After a reboot, how can the GFS master figure out where a given chunk is stored?

ANSWER: It can ask all of the chunk servers. The chunk servers keep track of which chunks they store, and they periodically tell the GFS master. When the GFS master starts up, it queries the chunk servers to rebuild its chunk location mapping information.

Problem 3 : Instructor trivia. [up to 2 bonus points]

(a) Which professor is not present in class today (April 24, 2013)?

Gibson

(b) What should Professor Ganger do about the sleep deprivation he suffered this semester?

Lots of fun answers here. Sleep? Take a vacation? More diet coke?

(c) How many TAs did we have for 746 this semester?

Three

(d) Which instructor enjoyed the NCAA men's basketball tournament most?

Ganger

(e) When and where should Greg first take his kids to Asia? Why?

Lots of fun answers... lets just hope he finally gets the party started.