CMU 18-746/15-746 **Storage Systems** **23 April 2014**

Spring 2014 **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 box around your answer.

Problem 1 : Short answer. [48 points]

(a) Many scalable distributed file system designs use multiple data servers but just a single metadata server. Why is this a reasonable approach, despite the fundamental scalability limitation that the single metadata server represents? Explain your answer.

ANSWER:
It is a much simpler design, which makes it easier to get right. Until the bottleneck becomes a problem, this is the right choice.

In general, one should only use complex distributed implementations when necessary. In this case, bypassing the server for data transfers reduces the server load, and the single server may not be a significant bottleneck for the large files often present in such environments.

(b) The corporate policy at Joe's company requires regular (daily) backups of the file server contents onto magnetic tape. So, each evening, a snapshot is made on the file server and saved to tape. Joe's colleague wants to configure the server to immediately delete the snapshot to free the space, once the backup is fully saved on tape, but Joe suggests keeping each snapshot until the next one is created. What is Joe's best argument for not deleting the snapshot right away? Explain your answer.

ANSWER:
Keeping the online snapshot allows users to restore recently-lost files quickly, without needing to mount and access the backup tape.

(c) Imagine a workload in which a client opens a large file, modifies a single block of that file, 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:
NFS. An AFS client would have to load and then write back the entire file, whereas an NFS client can read and update just the modified block.

(d) File systems like HDFS and GFS keep multiple replicas of each block, stored on different servers. Joe argues that one advantage of this approach is that the expensive scrubbing performed in most disk array systems is unnecessary. Do you agree? Explain your answer.

ANSWER:
No. The replication enables one to tolerate faults, but will not help one detect a corrupted block. Undetected faults reduce reliability, because the system cannot tolerate as many additional faults as expected.

(e) In many data centers, the aggregate bandwidth among machines within each rack is higher than the bandwidth between one rack and the next. Nonetheless, HDFS follows the GFS design of putting at least one replica of each newly written block on a server in a different rack than the writer, even though doing so is generally slower than putting all replicas on the same rack. Explain the main reason for this design choice.

ANSWER:
Rack failures (e.g., due to switch failure or power distribution unit failure) result is all machines in the rack being unavailable, thereby making all replicas in that rack unavailable. So, placing a replica on a different rack increases reliability.

(f) A FUSE-based file system implementation is almost always slower than an implementation in the kernel. Why is FUSE ever used? Explain your answer.

ANSWER:
Simplicity. Several ways in which it is simpler: user level debugging is easier than kernel debugging; the cost of supporting different kernel versions is reduced; some kernels are proprietary or have unacceptible licenses.

Problem 2 : More short answer. [48 points]

(a) One of the things things that Google changed, in moving from the original Google FS design to Colossus, was the way in which the three replicas are conveyed from the client to the three chunkservers that will store those replicas. The original Google FS used a daisy-chaining approach, with the client sending the data to the first server, which sends it to the second, which sends it to the third, and so on. The newer system has the client send to the three replicas directly. How does this design change reduce tail latency for write requests?

ANSWER:
A single slow server does not delay the entire write. Instead, the client can proceed as soon as it knows that the written block is stored on at least two of the servers.

(b) Joe operates a file server that supports a private cloud in which each virtual machine's virtual disk is stored as a file on the file server. Joe's file server supports logical backups to a disk-based backup system, and each nightly backup saves only the (whole) files that were modified during the day. Would you expect enabling deduplication in the backup system to provide significant capacity benefits? Explain your answer.

ANSWER:
Yes. Whole file backup of virtual disk images is the same as physical backups in traditional non-virtual environments. Most of the virtual disk blocks will not change from one backup to the next.

(c) Many large-scale web properties are built atop a simple get-put object interface, rather than the traditional distributed file system interface, in part to simplify implementation of scalable storage. Why is the implementation easier to scale? Explain your answer.

ANSWER:
The metadata is much simpler, with each object being independent. Complex multi-object semantics, such as rename between directories, do not need to be addressed.

Another reasonable answer is that the need to deal with complex concurrent access patterns to an object (e.g., two clients having a file open) is avoided, since each operation is atomic and idempotent.

(d) Imagine a 300-server scalable distributed file system that keeps 3 replicas of each data block, in one of two configuration options: (1) each data block is replicated on a random 3 of the 300 servers, (2) the 300 servers are treated as 100 groups, with each data block assigned to one group and replicated across the 3 servers of that group. The second option could be expected to be more reliable, in terms of time to first data loss, since it can tolerate many more failures than the first option (assuming they are in different groups). What would be your most compelling argument for using the first option? Explain your answer.

ANSWER:
There are at least two acceptable answers, both related to what happens when a failure occurs: (1) better load balancing, because the client requests that would have been served by the failed server are spread among the remaining 299, rather than only the other two of a 3-server group; (2) the reliability may actually be better for option 1, because restoration of the lost replicas can be immediate (onto free space on the remaining 299 servers) and proceed much faster (as 299 machines read replicas of lost data in parallel).

(e) Most scalable distributed file system designs that directly implement software parity-based redundancy, rather than replication, do so on a per-file basis. What is the primary reason that they don't do so at the block level, like in traditional disk array systems? Explain your answer.

ANSWER:
Two good answers: (1) most files are written in their entirity, which means that the parity can be computed from new data available at the time of the write; (2) if the client does the parity computation, the parity can be computed without needing to potentially read blocks from other files to which the client does not have permission.

(f) Joe has a directory called /user/joe/wierdfiles on his desktop machine, but he can't find it after mounting his corporate NFS server directory at /user/joe. How can he get access to that directory? Explain your answer.

ANSWER:
Unmount the NFS directory. The local directory is still there, even though the name was hidden by the NFS mount.

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

(a) How many diet cokes does Professor Ganger usually bring to class?
Two. Just in case.

(b) Which instructor developed the RAID taxonomy as part of his Ph.D. dissertation research?
Garth Gibson.

(c) Which question on this exam was your favorite and why?
Best answer: this one because this answer results in circular logic

(d) Name two TAs.
Any two will do.

(e) What great site in the world should Greg take his kids to see? Why?
Lots of fun answers... lets just hope he does **something** *for a change*