CMU 18-746/15-746 **Storage Systems** **25 April 2012**

Spring 2012 **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) Joe has implemented a simple file server that uses two threads: one to service cache hits and one to service cache misses. In his code, each thread finishes processing one request fully before starting the next. His server hardware includes a very fast network and a single traditional disk. Should he expect significantly improved throughput if he changes his server code to use multiple threads for servicing cache misses? Explain your answer.

Assuming that not all requests hit in the server cache and there is significant load on the server, he should see a reasonable amount of improvement due to disk scheduling being possible with multiple disk requests (in response to cache misses) where it wasn't with a single disk request at a time. Specifically, each cache miss processing thread would have one disk request pending, so multiple threads would mean multiple requests, giving the disk scheduler an opportunity to choose the order in which they are serviced by the disk.

While it is true that no significant improvement would be seen from parallelism, if the server load is insignificant, such an answer provides ignores most of the question.

It is also true that no significant improvement would be seen if all requests hit in the cache. Again, though, ignores most of the question.

(b) Many modern distributed file systems have a centralized metadata service but store file contents across many server machines. Joe is bothered that having the metadata service be centralized creates a bottleneck and a single point of failure, so he has designed an alternative in which each of 10 metadata servers manage one-tenth of the directories in the system. Identify a file system operation that will be particularly difficult to implement and debug. Explain your answer.

Rename. In particular, renaming a file from one directory to another requires updating both. Doing so consistently, when the two directories are managed by different metadata servers, will require extra work and will involve distributed systems challenges. Another such multi-server operation would be a

hard link creation or deletion, when two of the links for the file are in directories managed by different servers.

When writing the question, we intended for it to be interpreted as describing a system where each directory is managed by a single server, but many students made clear that they interpreted it as a system in which each directory is spread across all servers. So long as the explanation is clear that the multi-server coordination is the primary issue, many file system operations are valid answers under this interpretation.

(c) The original AFS file system design was much less efficient than NFS (version 3) for workloads in which a client opens a large file, appends a small amount of data, and closes the file. Explain why.

When the AFS client closes the file, the entire new version of the file is sent to the server, even if only a small change was made. With NFS, only the modified file blocks need to be sent to the server. The same whole file issue holds for reads, if the client does not already have the data cached.

(d) Two users (Jack and Jill) access data stored on two NFS file servers from separate desktop computers. If Jack tells Jill the pathname that he uses for a file, is Jill guaranteed to find the same file via the same pathname? Explain your answer.

No. Each client computer mounts each NFS file server within its namespace hierarchy independently, and nothing within NFS ensures that the two computers choose to do so the same way.

(e) In most disk array systems, all reads and writes are processed and distributed among disks by a single disk array controller. Joe has implemented a new multi-server distributed file system using the same concept: all client requests go through one ”controller” server that distributes them among other servers and forwards the responses (including data, for reads) back to the clients. But, Joe notices that performance is actually worse than when he simply stores all of the data on the one ”controller” server. Assuming that the CPU is not a bottleneck, what is the most likely reason that using additional servers in this architecture does not increase performance?

The ”controller” server’s network interface. When data is stored on the ”controller” server’s local disks, data written from (and read by) clients passes over its network interface once. In the new arrangement, that same data passes over that same network interface twice. Using multiple ”backend” file servers does not increase the bandwidth of that network interface bottleneck, because clients still go through the ”controller” server.

We were expecting people to focus on throughput, which would help in identifying the network as the most likely bottleneck, but some students focused on latency. With respect to latency, performance could decrease (i.e., latency increase) because each request would go from one round-trip (client to server) to two or more (client to ”controller” server to one or more ”backend” servers).

(f) Most modern file servers allow multiple snapshots to be retained online, and most IT administrators configure their servers to do so even though they also make backups on tape and/or remote mirrors. Give one reason why such server snapshots are kept.

Online snapshots enable quick and easy recovery from many user mistakes, like accidentally deleting or over-writing files, without needing to go to the tape or remote mirror backups. This convenience can save substantial administrator time.

Problem 2 : More short answer. [48 points]

(a) Joe argues that using RAID-per-file is a particularly good match for a distributed file system supporting AFS semantics. Do you agree or disagree? Explain your answer.

Agree. Original AFS semantics call for the entire file to be written on close, if the file changed at all, which allows for parity information to always be computed from the new version of the file without needing to separately pre-read existing data just for parity computation.

(b) Joe says that declustering of redundancy groups in a distributed file system (like Panasas PanFS or Google GFS) makes the rebuild time for a failed server's contents decrease as the number of servers grows. Do you agree or disagree? Explain your answer.

Agree. Declustering spreads the blocks of each RAID group over different sets of other servers/disks, so the rebuild of a failed disk will need a subset of the blocks from all servers/disks, rather than all the blocks from a subset of the servers/disks servers; it also spreads the space of the replacement server/disk over all servers/disks as free space that cannot be allocated, allowing the writing of rebuilt data to be spread a little to all servers/disks. With declustering, the fixed total work (read and write) that must be done to rebuild a failed disk is evenly spread over all servers/disks, and every RAID group can be recovered independently and at the same time (up to the total throughput of the network and XOR servers) so more servers means less work per server/disk, which means faster parallel rebuild.

(c) Joe's disk array regularly performs scrubbing on its 5 TB disks. Give one reason that he might want to use RAID-6 redundancy, even though he expects disk crashes to be relatively rare. Explain your answer.

A 5 TB disk has so many individual sectors that there is a non-trivial chance that one could become unreadable during the rebuild of a failed disk's contents, effectively resulting in a double-disk failure for a single stripe.

(d) Joe writes a program that opens a file and enters a loop that reads the first block over and over, checking to see if it changes. If his program finds that the first block changed, which of the following file systems could he be using: ext2fs, NFS (version 3), or AFS. (Note: it can be more than one.) Explain your answer.

ext2fs and NFSv3. For both, individual updates are visible to other clients/applications as soon as they are submitted to the file system, though other NFSv3 clients may not see them for up to 30 seconds. For AFS, the program should not see updates from other clients, because it sees only the version from the moment of the open, which it caches at the time of the open.

(e) Joe has a file system on which he regularly performs both a logical and a physical backup, because he is interested in seeing which one is faster. Explain the most likely reason that he observes logical backup being slower than physical after the tenth month, even though he observed the opposite (i.e., physical slower than logical) after the first month.

The number of files in the file system probably increased. In the first month, most of the disk capacity might have been unused, so a logical backup might have only needed to copy a few files (rather than the whole disk, like with a physical backup). In the tenth month, there might be a lot of files, and the extra seeks from reading them individually could be slower than just reading the full disk sequentially.

(f) The Zebra file manager stores the metadata across the same servers as regular client's store their file data. Explain how this enables Zebra to recover after a file manager crash, even if the machine on which the file manager previously executed becomes completely unusable.

Because the data is stored redundantly across the servers, the file manager can just be restarted on another machine. The restarted file manager can read the saved state from the remaining servers. This is in contrast to a file manager that stores all of its metadata on local disk, which cannot be recovered without accessing that disk (or a backup copy).

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

(a) What answer did Prof. Ganger indicate for problem #3, when noticing poor attendance in class one day?

"Yes"

(b) Name one guest lecturer from this semester.

Erik Riedel, Nitin Agrawal, Mike Kazar, or Hugo Patterson

(c) Which 746 instructor has the most 'g's in his name?

Greg Ganger (4 glorious 'g's)

(d) Which instructor continues to vacation at a cabin in Canada, every year?

Garth Gibson

(e) To which non-North America country should Greg take his kids first? Why?

Lots of fun answers... lets just hope he does **something** *for a change*