CMU 18-746/15-746 **Storage Systems** **20 April 2011**

Spring 2011 **Exam 2**

Name: ____________________________

Instructions

There are four (4) 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) Tim has a state-of-the-art 100GB Flash-based SSD on which he stores 20GB of active data. After he added a huge (70GB) backup file, he notices that the SSD's performance dropped significantly. Identify and briefly explain the most likely reason for this performance decrease.

(b) Zed has a desktop file system that implements snapshots. He configures the file system to take a snapshot each night, believing that it protects his desktop's data sufficiently. What is one problem that could cause Zed to lose data?

(c) Many distributed file systems (e.g., Google FS) and previous research systems (e.g., Zebra) use a dedicated server for metadata, rather than spreading file metadata across servers like they do with data. Identify and explain one primary reason for this design choice.

(d) Zim uses an incremental backup strategy in which a full backup is done each week, on Sunday, and a daily incremental backup is taken on other days. In his system, each day's incremental backup is relative to the previous day's. If his file system always holds 1TB, and he modifies exactly 100GB each day, what is the maximum amount of data he might need to read from his backup tapes in order to restore his file system after a failure? Justify your answer.

(e) Poe relies on a distributed log entry collection application. It consists of a process running on each node that periodically checks the local log for new records, opens the shared log file on a file server, appends the new records to the shared log file, and closes it. After having problems with NFS's weak consistency, Poe switched to using a bug-free AFS server. Explain the most likely reason that he still sees records being lost.

(f) Imagine a company that has a file system just like Google FS. Ted, who works at that company, has deployed a large-scale e-mail service atop it. For every e-mail message received, a separate small file is created. And, for each deleted message, the corresponding small file is deleted. Ted finds that the e-mail service's performance is limited by the file system, so he doubles the number of chunk servers. But, performance does not improve noticeable. What is the most likely reason that adding chunk servers would not improve performance? Explain.

Problem 2 : More short answer. [24 points]

(a) Identify a workload (i.e., access pattern and file characteristics) for which performance will be better with NFS than with AFS. Explain.

(b) Jed has decided to extend the a new append() operation to his NFS file system, in which a client specifies data that should be appended to the end of an existing file. His NFS file server crashes occasionally, but always restarts. His clients usually see no problems, though, because his RPC implementation retries each RPC until it receives an answer from the server. After one such crash, he finds multiple copies of the most recently append()'d data in an important file. Explain the most likely cause of this duplication.

(c) Fred has two file servers, one providing NFS service and the other AFS. He modifies them both to use non-volatile RAM for their caches. Which server would you expect to see a bigger performance boost, assuming that they serve identical clients? Explain.

Problem 3 : Layered File Systems (plus one other). [34 points]

(a) GIGA+ uses a distributed hash-table to store large directories. Suppose there is a large directory, ”/tmp”, managed by GIGA+, and that GIGA+ has split that directory into 1,100 partitions so far. An application (called foo) is started by the client boot sequence on each node and periodically runs a stat(”/tmp/foo.log”) to discover the values of its attributes (timestemps, length, etc). Suppose one node in the cluster reboots and restarts foo.

- If the number of servers available for partitions of GIGA+ is 2,000, what is the worst case number of tries of stat(”/tmp/foo.log”) that the newly booted node may have to issue?

- If the number of servers available for partitions of GIGA+ is 7, what is the worst case number of tries of stat(”/tmp/foo.log”) that the newly booted node may have to issue?

(b) Assume that you wrote a new FUSE file system, called myFS, that is mounted at the ”/tmp/mfs” directory in the underlying ext3 file system. Since creating this filesystem, your test codes have created exactly one file (”hello.txt”) in the root directory of the myFS file system and opened this file for writing.

- Before beginning to write the file, your debugging code prints the file handle of ”hello.txt”. What is the value of the handle printed by the debugging code? (Assume that the i-node number for ”hello.txt” in the underlying ext3 file system is 123456). Explain how this handle value is determined.

- Suppose another concurrent thread in your test code tries to repeat the above example; it tries to create the same file ”hello.txt” in the same directory after the thread in part (1) has already created this file. Obviously, a UNIX based file system must not allow duplicate file names in the same directory; it should reject the second create with the ”EEXIST” error code. Because the myFS FUSE file system is a layered on a traditional UNIX file system (ext3), your myFS code will get this error code from ext3. FUSE has a default way to propagate this error code to your test (application) code. What is the value FUSE returns?

(c) The most widely used version of GPFS (the parallel file system from IBM) has a few key properties: (1) each client has all the server code and all clients have access to all the disks, (2) other than the lock server, all data sharing is staged through the shared disks, (3) clients wanting to access a directory partition acquire a lock on that partition, inherently locking the child partition in case it wants to split the parent partition it has just locked. This version of GPFS creates large directories very fast provided that all concurrent creating threads are all running on the same one client, but it becomes very slow when threads on different clients are concurrently creating files at random in the same huge directory. Why do you think this version of GPFS was so slow at concurrent creates from multiple clients?

(d) Imagine a 100 GB SSD with 100,000 erase cycles. Also imagine that your always-accurate instructor tells you that this SSD can sustain writing at 100 MB/s, is overprovisioned by 50%, and achieves perfect wear leveling with a write amplification of 2.5. Approximately what is the minimum useful lifetime of the SSD (that is, until it wears out, neglecting random catastrophic failures)? Is the lifetime (i) less than a month, (ii) about one year, (iii) about 2 years, (iv) about 4 years, (v) at least 5 years. Show your calculations to explain your answer. (Note: 10,000,000 seconds is about 120 days.)

Problem 4 : Instructor trivia. [up to 2 bonus points]

(a) How does Garth pronounce the last letter of the alphabet?

(b) What beverage does Greg usually bring to class?

(c) Which TA is most likely to graduate soonest?

(d) Which TA will be doing an internship this summer?

(e) Where should Greg take his kids for a summer vacation? Why?