**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 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.

**ANSWER:**
SSD defragmentation becomes less efficient as SSD starts storing more data (higher cost of cleaning and write amplication).

(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?

**ANSWER:**
If the disk with the data and the snapshot has a "device" failure, everything is lost (and cannot be recovered without any additional copies available)

(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.

**ANSWER:**
Many reasons: simple design/implementation, error handling and debugging is easier, serialization and synchronization overhead is not much, etc.

(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.

**ANSWER:**
1.6 TB (1 TB on day 1 and 100 GB per day from day 2-7 of the week)

(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.

**ANSWER:**
AFS uses "last writer" wins semantics together with "write on close". So, if two of the nodes open the file before either closes it, then they will both start with and modify the same original. The second one to close will overwrite the changes of the first.

(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.

**ANSWER:**
GoogeFS namenode (or master or single MDS) is likely the bottleneck – every file create and delete has to this centralized server which may become the bottleneck to achieve high scalability.

**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.

**ANSWER:**
Several answers can work. The clearest cases relate to the whole file caching of AFS: reading the first few bytes of many large files will result in much more data transfer from clients to servers than necessary. Likewise for writing small amounts of data to large files.

(b) Jed has decided to add 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.

**ANSWER:**
Appends are not idempotent. So, the same value may be appended a second (or third or ...) by the retries.

(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.

**ANSWER:**
NFS benefits more because the servers supposed to synchronously writes updates to persistent storage, so using NVRAM can improve that latency.

**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?
  **ANSWER:**
  11. GIGA+ uses binary hash partition splitting to grow the index. If the number of servers is greater than the number of partitions, each server will have at most one partition on it, then it takes $log_2(number\_of\_partitions)$ tries because the split histories will tell you about the next partition on the next level of the index tree (and log(1100) = 11 partitions).

- 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?
  **ANSWER:** all 7 servers
  When number of servers is much smaller than the number of partitions, each server may have multiple partitions per server and hitting one server gets you all that information. So, in the worst case, the client may have to hit all servers to have an accurate state of the system.

(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.
  **ANSWER:**
  0x00000002. FUSE gives the root directory the handle 0x00000001 and assigns every subsequent handle by incrementing a global next handle number.

- 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?

  **ANSWER:**
  ”-EEXIST”. FUSE negates error codes it gets from the underlying file system as it propagates the error.

(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?

**ANSWER:**
Each client has to lock a partition in order to insert into it, and must get it from disk, so when it asks for the lock on that partition, the lock server forces another client to release its lock and write its changes back to disk before this client can read the partition from disk. So, concurrent creates may involved two disk operations per client create.

(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.)

**ANSWER:** (iii) about 2 years

```
- WA of 2.5, means 100 MB/s is really causing the flash to see 250 MB/s
- overprovisioning of 50% means that this 100 GB SSD is really 150 GB
- 150 GB is written once at 250 MB/s in 600 seconds
- 100,000 erase cycles means 60,000,000 seconds
= 694 days
```

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

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

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

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

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

(e) Where should Greg take his kids for a summer vacation? Why?
*Lots of fun answers... lets just hope he does* **something** *for a change*