

This exam is **closed book**. You can have **two pages** of notes. UH expels cheaters.

1. Which of the following statements are *true* or *false* (2 points) and *why*? (3 points)

a) Coda servers are not stateless.

True, the server maintains callbacks.

b) A pessimistic replication control protocol assumes that data can and will often be in an inconsistent state.

False, a pessimistic replication control protocol never allows inconsistent updates.

c) A journaling file system stores its i-node map into contiguous blocks

False, only log-structured file systems have i-node maps. (*Other file systems keep their i-node tables in contiguous blocks. As a result, the i-node number suffices to access a particular i-node.*)

d) All journaling file systems make the same guarantees about the durability of metadata updates.

False, journaling file systems can either use synchronous metadata updates and offer the same durability guarantees as FFS or use delayed updates and offer the same weaker durability guarantees as soft updates.

e) Elephant lets users specify which versions of their files should be kept forever.

True, it offers that option.

f) Unlike NFS, Coda does not support diskless clients.

True, a diskless client could not (a) cache entire files and (b) operate in disconnected mode.

2. Under which circumstances will soft updates write *two times* the same directory block to the disk? (10 points)

Soft updates will write two times any directory block that contains file deletions and file insertions, whenever the i-nodes of at least one deleted file and one new file are in the same i-node block.

3. Which functions are performed by the Sprit-LFS *roll forward* mechanism? (10 points)

The roll forward operation looks at the portion of the log that was written after the last checkpoint and accepts them if they leave the file system in a consistent state. In addition, updates the system i-node map whenever a summary block indicates the presence of a new i-node.

4. Why do NFS servers implement a *write-through* policy? (5 points) What is the main drawback of this policy? (5 points) Discuss two solutions that address that issue. (2×5 points)

NFS servers implement a write-through policy in order to remain stateless. The main drawback of this policy is that it slows down the system by requiring the server to write data to disk before replying to any client write request. Two techniques have been proposed to address that issue:

- a) Using non-volatile RAM in the server: this is the solution adopted by Network Appliance's NFS servers. Its sole drawback is its cost.
  - b) Using safe asynchronous updates: the server does not have then to write data to disk until it receives a *COMMIT* request from the client. If the *COMMIT* fails, the client must resend its copy of the data being written.
5. Consider a Coda file system having two copies *A* and *B* of the same file *X*. Which is the state of file *X* if the CVV's of copies *A* and *B* respectively are:

$$CVV_A = \{5, 4\} \quad CVV_B = \{4, 4\}$$

(5 points)

Replica *A* dominates replica *B* as its CVV contains higher values than that of replica *B*.

6. Somebody proposes to you to modify the *LOCKSS* system by letting the initiator of a poll notify all peers that participated in the poll about the poll outcome. He adds that it would allow the peers that have incorrect versions of an AU to repair these without having to initiate a new poll. What do you think of his proposal? (10 points for a well-justified answer)

This is a terrible idea as a malicious poll initiator could notify peers with good copies that their copies are not correct and convince them to replace it by a bad copy. With the current system, peers are required to check by themselves if the copies they have are good or bad; they should trust a single peer for that!

7. We know that Farsite uses a Byzantine agreement protocol in its directory host. Which steps takes Farsite to protect user files against malicious behaviors by its file hosts? (3×5 points)
- a) File blocks are encrypted so that file hosts cannot access their contents.
  - b) File blocks are also replicated on different hosts so that a single file host cannot maliciously destroy a file.
  - c) Farsite ensures that all copies of a given file block will be spread over machines controlled by different owners.

The paper we discussed in Spring 2005 also explains that Farsite stores a copy of a Merkle hash tree over the file data blocks in the directory group that manages the file's metadata.