COSC 6360 FINAL EXAMINATION AUGUST 11, 2005

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

1. Which techniques are used by FarSite to guarantee the availability and the integrity of (a) ***directory data*** and (b) ***file contents***? (2×5 points)

a) FarSite replicates directory and manage them through a Byzantine fault-tolerant protocol that ensures their integrity (as long as less than one third of the machines misbehave in any manner).

b) File contents are encrypted and replicated.

2. What are the respective advantages and disadvantages of journaling file systems using (a) ***buffered*** and (b) ***non-buffered log writes***? (2×5 points)

a) Buffered log writes do not require the file system to write individual updates to the log. This makes the server faster but does not guarantee the durability of updates as the last updates before a crash might be lost.

b) Non-buffered log writes ensure that no updates will be lost after a file crash but incur more overhead than buffered log writes.

3. Which steps must a log-structured file system take to find the disk addresses of all its ***i-node blocks*** after a system crash? (2×5 points)

a) The file system will go to its checkpoint area and fetch its i-node map. This map will contain the addresses of all i-node blocks at check point time.

b) It will discover the locations of the other i-node blocks during the roll-forward phase of the recovery process.

4. Which are the limitations of the recovery strategies of (a) FFS and (b) Sprite-LFS? (2×5 points) How does BSD-LFS address these two issues? (2×5 points)

a) FFS runs a comprehensive file system check (fschk) through the whole file system. This process is very thorough and very slow.

b) Sprite-LFS initializes first all the file structures from the most recent checkpoint then does a "roll forward" to incorporate all subsequent modifications. This process is very fast but does not check the overall integrity of the file system.

BSD-LFS combines two recovery strategies:

a) A quick roll forward from last checkpoint that is much faster than the FFS approach.

b) A complete consistency check of the file system that can be performed in the background.

2. Consider a file system using soft updates and assume that one directory block in its I/O buffer reflects the result of (a) one ***file delete*** and (b) one ***file creation***. Assuming that the i-nodes of the two files reside in the ***same i-node block***, describe each step the system will take to update the disk copies of the two blocks. (3×5 points) (***Hint:*** *treat this question as a problem.)*

The file system will perform three writes:

a) It will write first a version of the directory block reflecting the outcome of the delete but not that of the create.

b) It will write the new i-node block.

c) It will then write the final version of the directory block. This version will reflect the outcomes of both operations.

5. Consider an hypothetical version of NFS implementing ***leases***. What would be the main advantage of this modification? (5 points) Would that version of NFS still be ***stateless***? (5 points) Would the change affect the system's ability to recover from crashes? (5 points)

a) It would greatly improve the performance of NFS by eliminating all cache block revalidations while enforcing true Unix file sharing semantics.

b) This version of NFS would not be stateless.

c) The change would not affect the system's ability to recover from crashes because all leases would expire before the end of the recovery process.

3. Which ***guarantees*** does Coda provide, or not provide, about the ***consistency*** of its files? (2×5 points)

Coda does not guaranteed the consistency of users' files but it does guarantee that it will detect and report all file inconsistencies.

4. What criterion does Elephant use to decide that a specific version of a file should be kept ***forever***? (5 points)

Elephant looks at the time line of updates to the file in order to identify groups of updates separated by long periods of stability and keeps the last version of each group of updates

5. What is the main advantage of ***safe asynchronous updates*** that were introduced in the third version of NFS? (5 points)

Safe asynchronous updates improve the performance of NFS by eliminating nearly all blocking writes. (There might still be some blocking writes at commit time.)