

15-440: Distributed Systems

Problem Solving Assignment 4

School of Computer Science
Carnegie Mellon University, Qatar

Fall 2011

Due date: December 7, 2011

1. Suppose we have a file server that connects to a storage server via a storage area network (SAN). The storage server provides a collection of disk blocks for use by the file server. The file server uses these disk blocks to store files, and provides these files to its clients. Since its clients are fully capable of implementing a file system from the disk blocks provided by the file server, is the file server really necessary? Can't the clients deal directly with the storage server? Explain.

2. In most operating systems file access permissions are checked only when the file is opened and not with each read or write. Thus if you open a file successfully for both reading and writing, obtaining file descriptor f , and if you subsequently change the file's access permissions to read-only, you can still write to the file using f .
 - a. Explain why supporting this feature is difficult in distributed file systems whose servers do not hold open-file state, such as NFSv2.

 - b. How might you "approximate" this feature so that it's supported on NFSv2 for what's probably the most common case: the file's owner is accessing the file?

3. An approach for implementing file locking in distributed file systems is to maintain the locks on the server in volatile storage (i.e., storage whose contents disappear in the event of a crash). Thus, after a server crash, the server must recover its prior state by obtaining information from its clients about who had which files locked. If servers maintained locks in non-volatile storage, then, it would seem, state recovery after a crash would not be necessary. Explain why NFS does not maintain such lock information in non-volatile storage. (Hint: consider, among other things, client crashes.)

4. Read the paper entitled "***The Google File System***" by Sanjay Ghemawat *et. al.* and answer the following questions:
 - Summarize the paper in 2 paragraphs.
 - Despite that the Google File System scales well, it could be argued that the master is still a potential bottleneck. What would be a reasonable alternative to replace it?
 - State 3 strengths and 3 weaknesses in the paper.