Problem Set 4

15-440/15-640 Distributed Systems Spring 2015

Assigned: Tuesday April 14, 2015

Due: 5pm, Thursday April 23, 2015

Submission: Submit .pdf file via Autolab. ***No other file format will be accepted.*** The .pdf file can be a scan of handwritten answers, or .pdf created from a word processor such as Word, Latex, LibreOffice, Google Docs, etc.

Question 1 (25 points)

Briefly (i.e., few bullet points or few short sentences) answer the following questions about two-phase commit, Paxos, and eventual consistency.

A. When is it acceptable to use eventual consistency rather than strong consistency?

B. What safety properties does Paxos strive to enforce?

C. Two-phase commit is prone to indefinite blocking because of coordinator failure. How does Paxos address this problem? What does it sacrifice in return?

D. How does Paxos handle the situation where there are multiple concurrent proposers that are unaware of each other?

E. In Paxos, it is essential that acceptors never recant. In other words, an acceptor never changes the value v ($\equiv a_k$) after it is written to its stable storage. Explain why this is important.

Question 2 (20 points)

In the context of a POSIX-like distributed file system that uses optimistic replication (like Coda), consider violations of one-copy semantics represented by each of the following scenarios.

A. John, who is experiencing no networking or server failures, is working in the directory `/coda/coda.cs.cmu.edu/project/paxos/BUILD`. His test program crashes and creates a file named `core`. Jane, who is John's project partner, is working on a disconnected laptop and debugging the same program in the same directory. Her test program also crashes and creates a file named `core`. Just after this, Jane regains wireless network connectivity and is reconnected to the servers. What kind of conflict will Jane discover? In practice, what simple (almost trivial) feature of modern Linux systems reduces the likelihood of this scenario leading to a conflict in Coda-like systems?

B. Shortly after she discovers the conflict resulting from scenario A, but before she can do anything about it, Jane again loses wireless network connectivity. In total disgust, she types "make clean" in the BUILD directory. Meanwhile, John is still fully connected and experiencing no server or network failures. He edits a C file, then types “make”, and is now testing out the newly-built executable. At this point, Jane regains wireless connectivity. What kind of conflict will she discover? What is her simplest option to repairing this conflict?

Question 3 (25 points)

You have been hired to design a file-sharing service by a construction company. Project managers and engineers intend to use the service to collaborate on files using mainly mobile devices. For example, a project manager could be updating a building status file from the construction site, while the chief engineer may be viewing and updating that file from the office. Assume that they are likely to be updating different aspects of the same files. The construction site is located in the wilderness, and wireless coverage is spotty. Would you use a pessimistic or an optimistic replica control policy? Justify your decision. If you choose a pessimistic approach, explain why you believe its shortcomings will not hurt the productivity of the construction company. If you choose an optimistic approach, briefly describe your design for an application-specific conflict resolver.

(Hint: this is a deliberately open-ended question. Clearly state any assumptions you make, and then present your solution. The realism of your assumptions is important.)

Question 4 (15 points)

In the context of pessimistic replica control of byte ranges within files (not whole files), answer the following questions.

A. Give a use case in which read-one, write-all is a reasonable strategy.

B. Give a use case in which write-one, read-all is a reasonable strategy.

C. Suppose your boss has mandated use of a read-one, write-all strategy for all deployments. In the workload of the particular customer you are working with, writes are much more frequent than reads. You have 3 sets of servers that you can deploy as replicas. Choice-1 has one server with a read/write latency of 100 ms, and two servers with read/write latency of 400 ms. Choice-2 has one server with read/write latency of 100 ms, one server with read/write latency of 225 ms, and one server with read/write latency of 250 ms. Choice-3 has three servers with read/write latency of 225ms.

   Which set of servers would you use for this customer? Justify your answer.

Question 5 (15 points)

In the context of Gifford-style voting-based pessimistic replication of whole files, state the read and write quorums for each of the following system configurations. Briefly explain each answer.

A. 5 replicas, maximum read availability

B. 8 replicas, ability to update with at least two replicas up

C. 8 replicas, ability to update with at least five replicas up