CS/CE/TE 6378: Project II

Instructor: Ravi Prakash

Assigned on: Oct 18, 2012
Due date and time: Nov 8, 2012, 11:59 pm

You are required to implement a replicated file system using the distributed mutual exclusion exercise performed in Project 1. The critical section execution (in project-1) has to be replaced with *WRITE* operation on the replicated file system. The replicated file system consists of three nodes, each hosts a copy of some file. Initially, all copies of the file are identical, $i.e.$, the replicas are consistent. Your task is to make sure they are consistent after every *WRITE*.

This is an individual project and you are expected to demonstrate its operation to the instructor and/or the TA.

1 Requirements

1. There are nine nodes in the system, numbered zero through eight. Each node executes on a different machine. You can choose the machines from the fifty machines set aside for network programming (net01.utdallas.edu - net50.utdallas.edu). Nodes 0, 1 and 2 act as servers hosting the replicated file system. Rest of the nodes are clients. One example of such network topology is shown in Figure. 1. Your program should be easily extensible for any number of servers and clients.

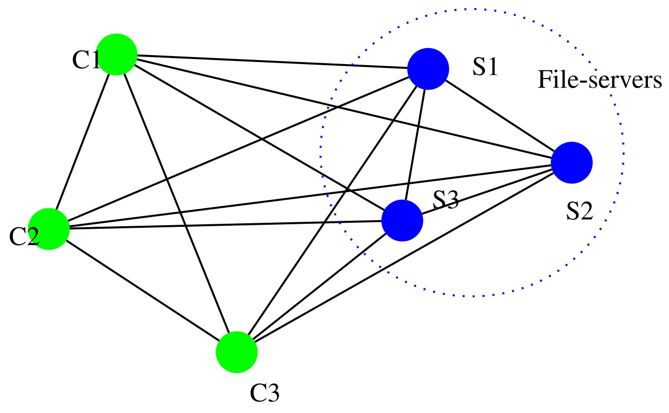

Figure 1: Network Topology with three clients

2. There are reliable socket connections (TCP) between each pair of nodes. The algorithm messages are sent over these connections.

3. All client nodes go through the following sequence of operations until each node has successfully generated *WRITE* requests 40 times:

   (a) Waits for a period of time that is uniformly distributed in the range [10, 50] time units before requesting next *WRITE*.

   (b) Acquire access to the critical section using the mutual exclusion protocol.

   (c) On entry into the critical section, the node writes the message $< ID, SEQ\_NUM, STRING >$ (where ID is the client number, SEQ_NUM is the sequence number of the client write operation (initialized to zero), and $STRING$ is the client's hostname) to each of the replica servers. Each message should be appended to the file in a separate line.

4. Each replica server must perform *WRITEs* to the file in the order they were received.

Your program must display informative log messages on console. You may write a program/script to test the correctness of the program.

2 Submission Information

The submission should be through WebCT in the form of an archive consisting of:

1. File(s) containing the source code.
2. The README file, which describes how to run your program.

DO NOT submit unnecessary files.