Distributed Systems 2016 Project 2

Due date: December 12th, 2016.

Implement a consistency test for the key-value store we built in project 1. On the command line, your test should accept the host and port of the server. Your test should run for no longer than one minute. The exit code of your test should be 0 if the test passed, 1 if there were consistency errors, and 2 if the test was inconclusive (e.g., the server stopped responding). Stdout and stderr of your test will be ignored for grading purposes. The keys and values that your test uses can be arbitrary strings, but should be under 1KB.

Note: Don't mix up exit code and stdout. For reference see this Wikipedia article and this Stack Overflow discussion.

Example invocation in Linux (test passes):

```
$ ./consistency_test -server 192.168.1.5:9634
$ echo $?
0
```

Example invocation in Linux (test fails):

```
$ ./consistency_test -server 192.168.1.5:9634
$ echo $?
1
```

In your submission, include a README.txt file that explains how your program works internally. Include the source code in your submission. Do not include compiled code in your submission.

Be sure your consistency test issues concurrent RPCs. Some bugs will not manifest unless there is concurrency. To get concurrency, there are several options depending on your programming language: multiple threads, multiple processes, asynchronous I/O, goroutines, etc.

Note: Please refer to the paper "What consistency does your key-value store actually provide?" as a reference regarding what is meant by "consistency".

Grading: Your test will be run 25 times, against buggy and non-buggy servers. Your grade will be the number of runs where your exit code was correct. If your test has an exit code of 2, it will not count toward the 25 runs. Only the runs where the exit code was 0 or 1 count.