Northeastern University

CS 6650 Scalable Dist Systems

Project #2 [200 points]

Multi-threaded Key-Value Store using RPC

Guidelines

Project #1 should be electronically submitted to Blackboard by midnight on the due date. A submission link is provided.

Assignment Overview

For this project, you will extend Project #1 in two distinct ways.

1) You need to enable your client and server to communicate using Remote Procedure Calls (RPC) instead of sockets. If you've implemented Project #1 in Java, you may want to look into and leverage Java RMI for RPC communication. However, there are multiple other RPC frameworks you can leverage (with their own IDLs) to provide the stubs/skeletons necessary across the network. An additional example that enables the use of multiple languages is Apache Thrift (http://thrift.apache.org/ (Links to an external site.)

2) You need to make your server multi-threaded such that you can handle multiple outstanding client requests at once. You may decide how to thread your server. One approach may be to use thread pools similar to other servers, although there are certainly many ways to do this. The key result is that your servers should be able to handle requests from multiple running instances of you client doing concurrent PUT, GET, and DELETE operations. Due to the addition of multi-threading, you will need to handle mutual exclusion.

As in project #1, you should use your client to pre-populate the Key-Value store with data and a set of keys. The composition of data is up to you in terms of what you want to store. Once the key-value store is populated, your client must do at least 5 of each operation: 5 PUTs, 5 GETs, 5 DELETEs.

Evaluation

Your multi-threaded Key-Value Store server will be evaluated on how well they interoperate with each other using RPC while doing concurrent operations. The grade for your executive summary is based on the effort you put into the assignment overview and technical impression. In general, if you put some effort into your writing, you will receive full credit for your executive summary (provided that it is properly formatted and submitted as a plain text file).

Executive Summary

Part of your completed assignment submission should be an executive summary containing an "assignment overview" (1 paragraph, up to about 250 words) explaining what you understand to be the purpose and scope of the assignment and a "technical impression" (1–2 paragraphs, about 200–500 words) describing your experiences while carrying out the assignment. The assignment overview shows how well you understand the assignment; the technical impression section helps to determine what parts of the assignment need clarification, improvement, etc., for the future.

The grade for your executive summary is based on the effort you put into the assignment overview and

technical impression. In general, if you put some effort into your writing, you will receive full credit for your executive summary.

Project Deliverables

The following items should be archived together, e.g., placed in a .zip file or tarball file (*.tgz or *.tar.gz), and electronically submitted via the link is provided on the course Moodle page.

1. All novel Java, C, and/or other source code files implementing the client and server programs, i.e., plus any additional support code.
2. A simple README that includes
    i. How to build your server and client codes (including any external libraries necessary)
    ii. How to run your server and client programs
    iii. Your executive summary

Other notes:

You should use your client to pre-populate the Key-Value store with data and a set of keys. The composition of the data is up to you in terms of what you want to store there. Once the key-value store is populated, your client must do at least five of each operation: 5 PUTs, 5 GETs, 5 DELETEs.