International Technological University

Department of Computer Science

Distributed Systems (CSC 502), Section 1

Signature Project: Stateless File Server with Client Side Cache

Project Overview:

Each student will work within a group on this project. This project concerns the design and implementation of a client-Server architecture. The server is a simple stateless file server and the client access file over the network from the file server. The client side will support client side file caching (not block-level caching). This project is meant to enrich your experience in distributed systems. Please contact the instructor if you are having difficulty joining a group or if you have ideas that you would like to discuss/validate.

Goals:

The main goal is to allow client(s) be able to access (Create / Read / Write / Delete), assuming the user has the appropriate access permissions, remote file on a remote server over the network using Java or CORBA. **Notice, no access to portion of a file**. Understanding Caching is an important component in distributed systems architecture. The client side implementation will support simple client –side caching (caching to be implemented as a library linked to the client application).

Architecture Requirements:

1. Either use Java/RMI or CORBA for communication between the client and the server.
2. Assume that the file server has a process serving client requests at a known, predetermined, socket address (IP, port>.
3. The Server will support concurrency control, i.e., locking at the file-level.
4. To simplify things, client will **access only the whole file for simplicity**. As a result files on the server used in testing should be small.

5. Client caching is write-through, i.e., no optimizations for lazy writes.
6. Server will maintain which client accessed what and sends invalidate cache message to the appropriate clients when needed.
7. Client side cache is structured as <key, valid, value> table where key is the file-name string, valid is a tag to indicate if this entry is having valid data or not, and value is the file content as a string assuming the entry is valid. Will use hashing based on the file-name to get to the right <Key, Valid, Value> entry in the cache table.

Each group will need to turn in (~3-pages) proposal for the instructor feedback that describes the project ideas and any choices that are made?

Expected Outcome:

Each group on the project due date needs to turn-in the following:

1. Project Report (word document ~10 pages) which clearly describes the design of your software including your architecture, design decisions, cache features, Server concurrency control, error handling, etc.
2. Hardcopy of the source code; ensure that your code is well commented.
3. Ensure you have Makefile to clean and build your client/server application.
4. If you made effort on design and implementation but have not been able to make your code run, you can still submit a report describing your design as concretely as possible. Although you will not receive high marks.
5. Send softcopy of the code and Makefile to the instructor before the demo date.
6. The instructor will schedule time to see and discuss the demo.
7. 50% of the grade (Group-level grade) to the project depends on the project run correctly or not. The other 50% of the project is on individual basis depending on the answer you give to questions from the instructor.