CMPS 4760/6760 Distributed Systems – Spring 21

Lab 1 (20 points)

Due 02/11/21 before class

In this lab, you will implement a simple election service using Java RMI. You need to implement:

An *Election* interface, a remote interface that provides two remote methods:

vote: This method has two parameters through which the client supplies the name of a candidate (a string) and the 'voter's number' (an integer used to ensure each user votes once only). The voter's numbers are allocated sparsely from the range of integers to make them hard to guess.

result: This method has two parameters through which the server supplies the client with the name of a candidate and the number of votes for that candidate.

A *Server* program that (1) implements the *Election* interface, and (2) contains a *main* method that creates a remote object and binds it to a name in the RMIregistry.

A *Client* program that invokes the remote object to submit a vote or query the voting result.

Some additional requirements:

1. Ensure that each user votes only once.
2. The voting records remain consistent even when it is accessed concurrently by multiple clients.
3. All votes are saftely stored even when the server process crashes.

In addition to the source code, you need to turn in (1) a document that describes how you address the above requirements in your implementation; (2) screenshots of the command windows for running the server and the client, respectively.

References:

1. Getting Started Using Java RMI: `https://docs.oracle.com/javase/8/docs/technotes/guides/rmi/hello/hello-world.html`.
2. Java Concurrency: `https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html`.