Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.
Fourth Year
Spring 2018

CS432: Distributed Systems and Net-Centric Computing
Assigned: Tuesday, Feb 26th 2018
Due: Monday, March 12th 2018

Bulletin Board System V2.0

Objective

This assignment aims to familiarize yourself with either RPC or RMI.

Description

In this assignment you will implement news bulletin board system, like we did in Assignment 1. But this time we will use RPC/RMI instead of sockets. So, we will be developing another distributed client/server version of the solution using a service-oriented request/reply scheme.

Typically, we will use the same code we have written for Assignment 1 but using RMI for all necessary communication instead of the Sockets. Also, while RMI will take care of the client threads (we don't need to handle it like sockets), we still need to provide the necessary synchronization. In sum, the major difference from Assignment 1 is that Read and Write requests of clients will be remote method invocations.

Unless otherwise specified in this document, Assignment 1 specification is valid for this assignment.

RMI Implementation

RMI Registry

RMI registry is a Naming Service that acts like a broker. RMI servers registers their objects with the RMI registry to make them publicly available. RMI clients look up the registry to locate an object they are interested in and then obtain a reference to that object in order to use its remote methods. Of course, servers register only their remote objects, which have remote methods.

The Remote Interface

Remote methods that will be available to clients over the network should be declared. This is accomplished by defining a remote interface. This interface must extend java.rmi.Remote class and must be defined as public to make it accessible to clients. The body of this interface definition consists of only the declaration of remote methods. The declaration is nothing more than the signatures of methods namely method name, parameters and return type. Each method declaration must also declare java.rmi.RemoteException as the throws part.

Dr. Ahmed El-Sayed
Eng. Alaa El-Ebshihy
Eng. Reham Mohamed

System Configuration and Utility

The only change in system.properties is that it specifies the port of rmiregistry (we assume that the rmiregistry runs on the same host as the reader/writer sever).

For example:
RW.rmiregistry.port=1099

Implementation of Remote Interface

After defining the remote interface, you need to implement this interface. Implementing the interface means writing the actual code that will form the bodies of your remote methods. For this purpose you define a class from which you create your remote objects that will serve the clients. This class must extend RemoteObject provided by the java.rmi.server package.

There is also a subclass UnicastRemoteObject which is also provided by the same package that provide sufficient basic functionality for our purpose.
When you call its constructor, necessary steps are taken for you so that you will not need to deal with them.
An RMI server registers its remote objects by using bind() or rebind() method of Naming class.

RMI Clients

RMI clients are mostly ordinary Java programs. The only difference is that they use the remote interface. As you now know remote interface declares the remote methods to be used. In addition, the clients need to obtain a reference to the remote object, which includes the methods declared in remote interface. The clients obtain this reference by using lookup() method of Naming class.

Readings

- An Overview of RMI Applications: http://docs.oracle.com/javase/tutorial/rmi/overview.html

Server-Client Specifications Reminder

- Similar to V1.0 of the bulletin board system, our news is a one shared integer where writers modify it with their ID value and readers read it.
- The server provides two numbering sequences for the clients: (1) the Request Sequence (rSeq) is the sequence number for the incoming requests and (2) the Service Sequence (sSeq) is the sequence number of the request being serviced. They are different due to the access policy that you will implement.
- The server log format is specified in assignment 1.
- The client logs should be at the clients' machines and their format is specified in assignment 1.

Notes

- Develop this assignment in Java or C/C++.
- You should deliver a report explaining your design and implementation.
- We will test the assignment using multiple machines.
- The assignment is based on assignments from UFL.

Grading Policies

- You should work in groups of 2 or 3 students.
- The penalty of late submission is 20% of delivery grades per week and after 4 weeks of the deadline you will get no grades for the delivery.
- Plagiarizing is not acceptable. Sharing code fragments between groups is prohibited and all the groups that are engaged in this action will be severely penalized. Not delivering the assignment will be much better than committing this offence.

Dr. Ahmed El-Sayed

Eng. Alaa El-Ebshihy
Eng. Reham Mohamed