

Distributed Systems Assignment 3

Java RMI

Due: March 11, 11:55PM

1 Overview

The aim of the assignment is to get familiar with Java RMI. RMI stands for Remote Method Invocation. It is a mechanism that allows an object residing in one system (JVM) to access/invoke an object running on another JVM. RMI is used to build distributed applications, it provides remote communication between Java programs.

2 Important Points

- The assignment has to be done in Java only.
- Java RMI tutorial1: https://www.tutorialspoint.com/java_rmi/java_rmi_introduction.htm
- Java RMI tutorial2: <https://www.geeksforgeeks.org/how-to-run-java-rmi-application/>

3 Problem

For this assignment, you will be using RMI(Remote Method Invocation) in Java to implement a simple single server architecture with support for multiple clients. The details are as follows:

- The server maintains a list of graphs each associated with a distinct identifier.
- Clients can request to add a new graph, update an existing graph and query for the total weight of the minimum weight spanning tree of a given graph.
- Clients can request to add a new graph using ‘add_graph <graph_identifier> n’. This command will add a new graph on the server with the identifier graph identifier and n number of nodes. The graph identifier is a string with a maximum length of 10 and it won’t already exist. n will be in the range: $1 \leq n \leq 100,000$.
- Clients can request to add a new edge in a graph using ‘add_edge <graph_identifier> <u> <v> <w>’. This will add an undirected edge between the nodes u and v with weight w. u and v are the node numbers of the endpoints of the edge such that $1 \leq u, v \leq n$ and $0 \leq w \leq 10,000$. n is the number of nodes in the specified graph. A graph with identifier graph identifier will already exist. There can be multiple edges and self-loops added to the graph.
- Clients can request for the total weight of the minimum weight spanning tree in a graph from the server using ‘get_mst <graph_identifier>’. The client will print the solution the server returns. In case the graph does not have a spanning tree, -1 should be printed. A graph with identifier graph identifier will already exist.
- All values should fit in 32-bit signed integers.
- The server should be able to handle multiple clients simultaneously and should also work with clients on other machines.
- You are free to use any algorithm for MST (can be sequential).

4 Submission Instructions

Your submission is expected to be a <RollNumber>.zip file containing a directory with the same name as your roll number that holds the following files:

1. 2 Java source files - **Client.java** and **Server.java**. Your code will be run using:
 - The files would be compiled with ‘javac *.java’.
 - The server is run using with ‘java Server <server_port>’.
 - The clients are run using ‘java Client <server_ip> <server_port>’.
2. A report for the problem - **README.md**. The report should contain details of the architecture, algorithm implementation, results and observations.

NOTE: Strict actions would be taken against anyone found involved in any kind of plagiarism either from the internet or from other students.