CSE4/586 Distributed Systems Spring 2011
Handout#1: Remote Method Invocation

Prepared by Manavendar Reddy (mrm42@buffalo.edu)

1. Link for the RMI tutorial http://download.oracle.com/javase/tutorial/rmi/overview.html

2. Eclipse download link http://www.eclipse.org/downloads/

3. JDK download link http://www.oracle.com/technetwork/java/javase/downloads/index.html

4. Please go through above RMI tutorial to better understand step 5.

5. Steps to create and run RMI on Eclipse

   a. Create java project in eclipse (File -> New -> Project -> Java -> Java Project). Let's assume C:/Weather is the root directory for the project.
   b. Weather/src should contain source files and Weather/bin should contain all the class files.
   c. Create interface files Compute.java and Task.java inside compute package (Weather/src/compute/Compute.java and Weather/src/compute/Task.java). *Compute* Remote interface enables clients to submit tasks to the server. The client interface *Task* defines how the server executes a submitted task.
   d. Implement remove interface ComputeEngine.java inside server package (Weather/src/server/ComputeEngine.java). This file should implement Compute interface to create and install security manager, export remote objects and registry remote objects with RMI Registry.
   e. Create client files (ComputeWeather.java, Weather.java, WeatherInfo.java) inside client package (Weather/src/client/*). ComputeWeather.java looks up and invokes remote object. Weather.java implements Task.java and defines the work to be done by the server (i.e parse weather feeds and store the data in mysql database). WeatherInfo.java is the object created by server to return it to client, it stores the data obtained by parsing weather feed. This class is analogous to BigDecimal class in the tutorial.
   f. Download mysql connector from http://dev.mysql.com/downloads/connector/j/ and copy the jar file to Weather/lib/ directory (Create lib directory)
   g. Create server.policy and client.policy files in Weather directory.
   h. server.policy should grant access to classes in the Weather directory including the mysql-connect.jar. Here is an example of server.policy
      ```
      grant codeBase "file:C:/Weather/-"{
         permission java.security.AllPermission;
      };
      ```
      In the above example the trailing /- grants access to all the classes in the Weather directory including all sub-directories.
   i. client.policy should grant access to classes in Weather/bin. Here is an example.
      ```
      grant codeBase "file:C:/Weather/bin/" {
      ```

```
                permission java.security.AllPermission;
            };
```

j. Eclipse will automatically compile the programs and place the class files in the bin directory. After successful completion of above steps, its time to run the server and client. Make sure to start rmiregistry and mysql server before starting the rmi server.

k. Enter into Run/Debug tab under Weather properties to create an instance of ComputeEngine. Select New->Java Application and specify ComputeEngine as the name of the configuration. Specify Weather as the project and server.ComputeEngine as the Main Class. Goto Arguments tab in the same window and specify the VM arguments as

   ```
   Djava.rmi.server.codebase=file:C:/Weather/bin/ -
   Djava.rmi.server.hostname=localhost -Djava.security.policy=server.policy
   ```

   Press ok and start the rmi server (ComputeEngine) by running ComputeEngine as Java Application.

l. Similarly create an instance of ComputeWeather and in VM arguments tab specify the following commands.

   ```
   -Djava.rmi.server.codebase="file:C:/Weather/lib/mysql-connector-java.jar
   file:C:/Weather/bin/" -Djava.security.policy=client.policy
   ```

   and in Program arguments tab specify arguments needed by the main method of ComputeWeather class. Press ok and start the client by running ComputeWeather as Java Application.