Assignment 2: Implementing Your Own RPC/RMI Framework

CS328 - Distributed and Cloud Computing

DDL: 23:59, ~~November 20~~, 2023
November 28

1 Introduction

You are required to design and code a working program and submit well-commented code together with a lab report. Any text should be in English only. Plagiarism is strictly prohibited.

2 Preamble

Before working on this assignment, you should be familiar with: the interface of Java RMI framework, Java multi-threading, Java object serialization/Protobuf, Java reflection and proxy. You may find some useful information in the lab slides.

3 Specification

In this assignment, you need to implement a simple but fully functional Java RPC/RMI library. A code repository `MyRMI.zip` will be provided to you for reference.

There are four main parts of implementing your own RPC/RMI framework based on the existing code:

1. Design communication protocols for **stub/skeleton** communication. Use any communication and serialization methods you prefer.

2. Implement the **skeleton generation** for remote objects on the server side. When exporting a remote object, a skeleton should be generated to allow incoming method invocation on that object. You also need to implement a **multi-threaded skeleton class** (for example, one thread per connection or one thread per request, or using a thread pool) to handle the incoming requests.

3. Implement the **stub generation**, which provides transparency for remote invocation initiated by clients. This part includes creating proxies and appropriate invocation handlers for proxies.

4. Implement the **registry**, which also utilises its own skeleton and stub to handle incoming remote method invocation by the server and client, but many require special processing when invoking certain methods. The infrastructure implemented in 1, 2 and 3 can be reused. Or, you could try to use some existing libraries such as *Apache Zookeeper*[1] and integrate them into your implementation.

The reference code is provided in a `.zip` file named `MyRMI.zip`, with configurations for *IntelliJ IDEA* and *Maven*. **The reference code was previously designed for the RMI framework (the same structure as Java's RMI), so you may change it as needed.**

You are free to implement any **bonus** you want to, such as load balancing, serialization with libraries such as Protobuf, or the web service design style. You will get up to 10 points for the bonus you implement, and the bonus content of each submission will be evaluated individually.

4 Submission

Be sure to include in your submission: source code files and a report (using the provided template) in PDF format. Pack all files into **`SID_NAME_A2.zip`**, where `SID` is your student ID and `NAME` is your name (e.g., 11710106_ 张三 _A2.zip).

5 Assessment

The full mark for this assignment is 100, and is assigned according to Table 1 and 2.

The maximum bonus you could get is 10 points. **If you get more than 100 points, your bonus will be added to the overall score of your homework. Please explicitly document in the report any bonus that you have implemented.**

[1] `https://zookeeper.apache.org/`

Code	
Functionalities	**Marks**
Overall structure and interface	30
Stub/skeleton generation and communication	20
Serialization	10
Registry	10
Quality of code (comments, naming, etc.)	5
Total	**75**

Table 1: Assessment for code.

Report	
Items	**Marks**
Explanation of your design	10
Tests of your design (with screenshots): use simple client and server codes to show how to use your framework and the execution results.	10
Difficulties and solutions	5
Total	**25**

Table 2: Assessment of the report.

6 Useful Materials

The source code of RMI in jdk5.0: https://github.com/eagle518/jdk-source-code/tree/master/jdk5.0_src/j2se/src/share/classes/java/rmi