



# PARALLEL AND CLOUD COMPUTING

## REPORT

### LAB ASSIGNMENT: 2

Student Name: Yaohui Han (韩耀辉)

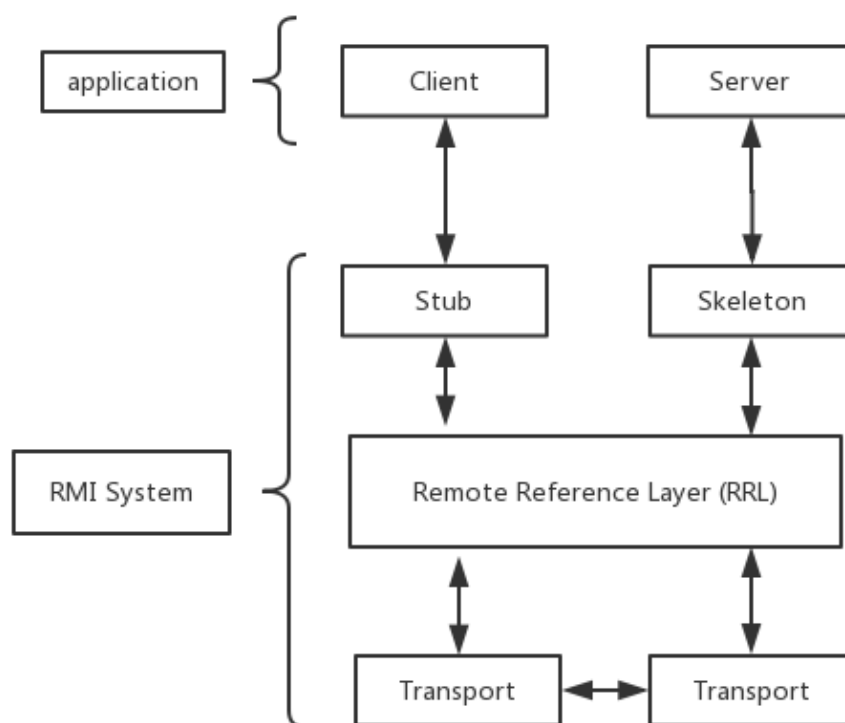
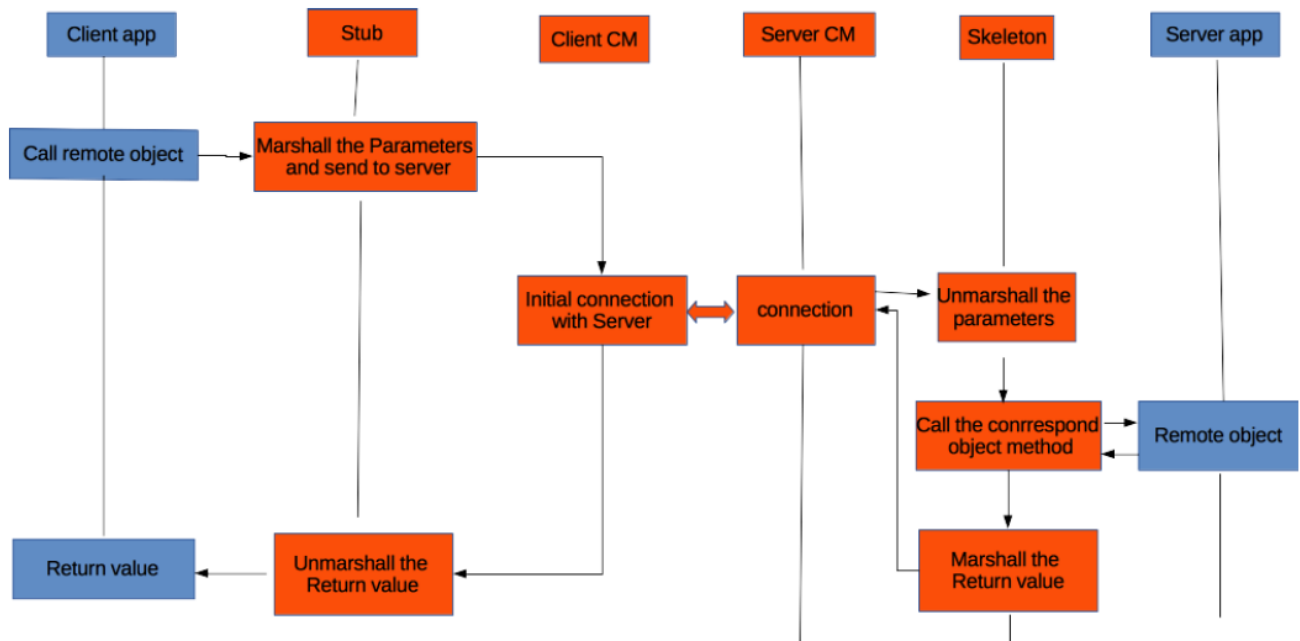
Student ID: 11611223

Student E-mail: 11611223@mail.sustc.edu.cn

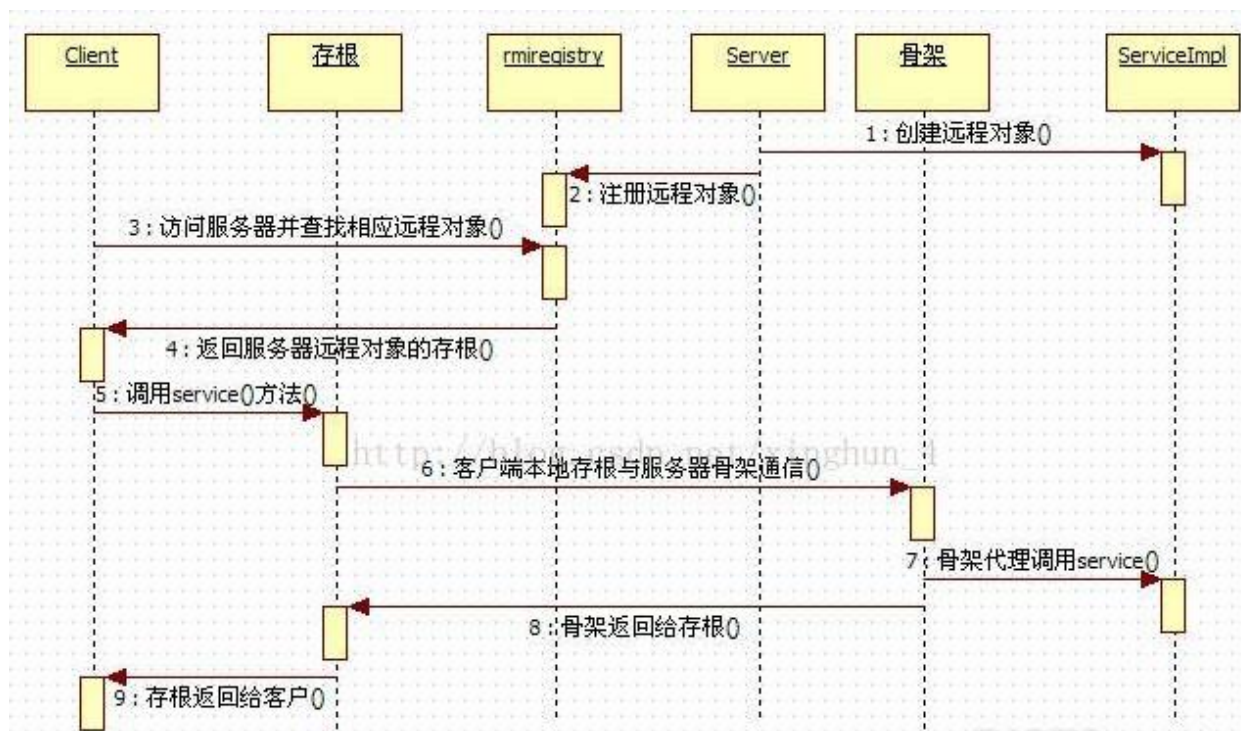


## DESIGN

Describe the design of your system by providing the following information:



The method call is passed from the client object via the Stub, Remote Reference Layer, and Transport Layer, down to the host, and then through the Transport Layer, up through the Remote Reference Layer and Skeleton, to the server object. The stub acts as a proxy for the remote server object, making the object available to the client for activation. The Remote Reference Layer handles semantics, manages communication for single or multiple objects, and determines whether calls should be sent to one server or multiple. The Transport Layer manages the actual connections and tracks remote objects that can accept method calls. The server-side Skeleton completes the actual method call to the server object and gets the return value. The return value is passed back to the client via the remote reference layer, the server-side transport layer, and then back up through the transport layer and the remote call layer. Finally, the stub gets the return value.



1. Server creates a remote object , registry and stub.

```

Impl robj = new Impl();
InterRL stub = (InterRL) UnicastRemoteObject.exportObject(robj,1098);
RemoteObjectRef ror = new RemoteObjectRef(host, port, rortbl.counter,
"InterRL");
LocateRegistry.createRegistry(1098);
SimpleRegistry sr = LocateSimpleRegistry.getRegistry(host, port);
sr.rebind(ServiceName, ror);
    
```

2. The server registered a remote object

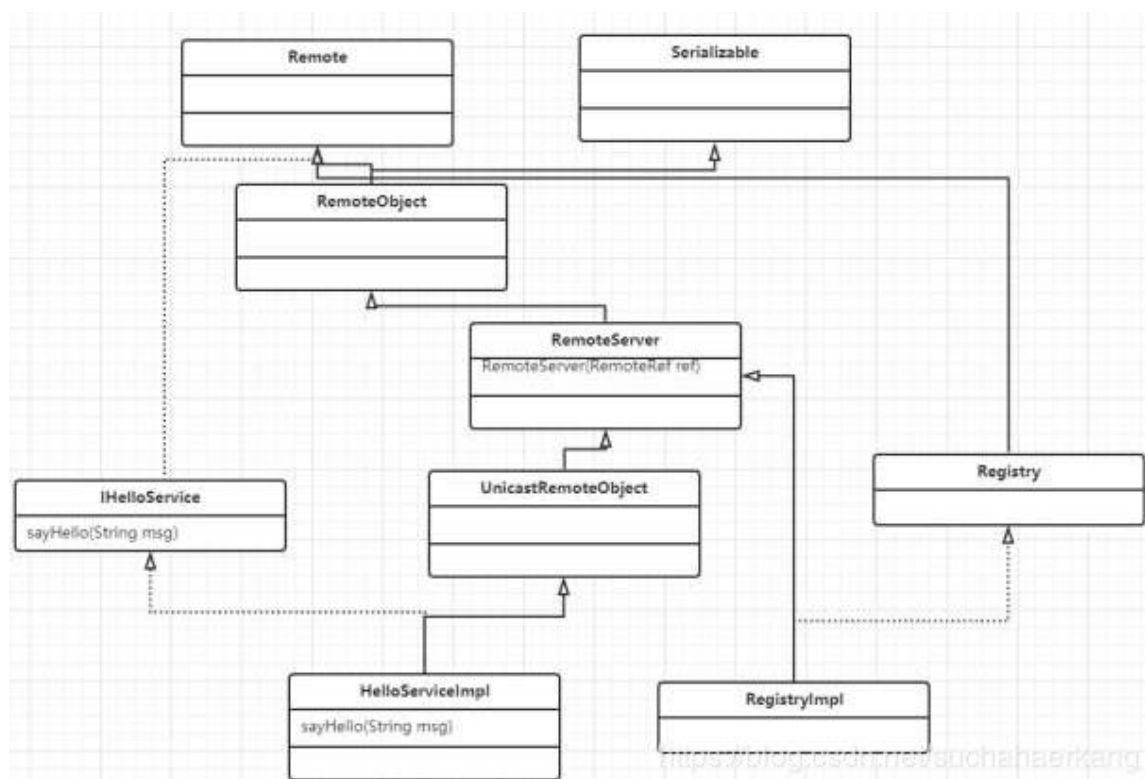
```

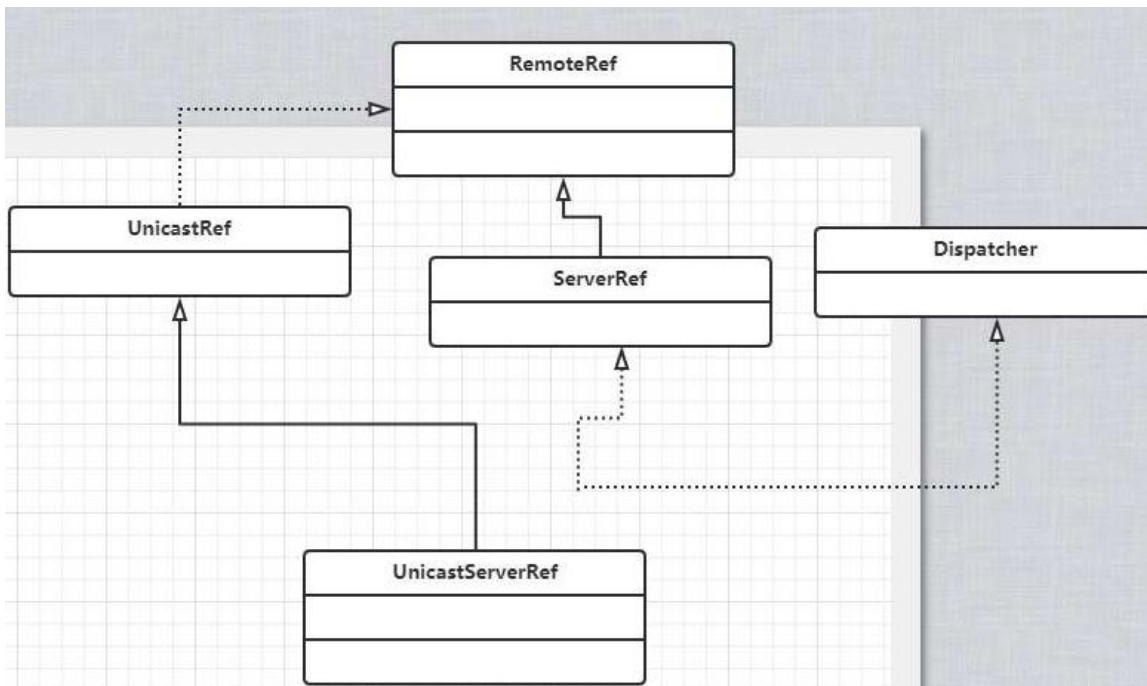
RemoteObjectRef ror = new RemoteObjectRef(host, port, rortbl.counter,
"interName");
    
```

3. The client accesses the Registry to create a stub, Registry.lookup

```
SimpleRegistry sr = LocateSimpleRegistry.getRegistry(host, port);
RemoteObjectRef ror = sr.lookup(ServiceName);
stub = (InterRL)ror.localise();
```

4. Return remote objects and stubs
5. Calling a remote method
6. Stub and skeleton communicate
7. Skeleton call method
8. Skeleton returned to stub
9. Stub is returned to the client





## PROBLEMS

Server creates a remote object , registry and stub. The server registered a remote object. The client accesses the Registry to create a stub, Registry.lookup . Return remote objects and stubs. Calling a remote method. Stub and skeleton communicate. Skeleton call method. Skeleton returned to stub. Stub is returned to the client.

The code given at the beginning does not understand. After searching and consulting, I understand the meaning of each class.



## RUNNING RESULT

---

Environment: Win 7

Java: jdk1.8.0\_121

JDBC: mysql-connector-java-5.1.44-bin.jar

Mysql Server version: 5.7.18-log MySQL Community Server (GPL)

### Database construction statement :

```
drop database if exists Cloud_Computing_Assignment;
```

```
create database Cloud_Computing_Assignment;
```

```
use Cloud_Computing_Assignment;
```

```
create table User_Login(  
id int(5) primary key not null auto_increment,  
username varchar(40),  
passwd varchar(40)  
);
```

So:

```
mysql> desc user_login;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| id    | int(5) | NO | PRI | NULL | auto_increment |  
| username | varchar(40) | YES | | NULL | |  
| passwd | varchar(40) | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```



## Start:

```
cd 11611223
javac -d . *.java
java SimpleRegistryServer
java Server 1099
java Client 1099
```

```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java SimpleRegistryServer
server socket created.
```

```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java SimpleRegistryServer
server socket created.
```

```
accepted the request.
I was asked who I am, so I answered.
```

```
accepted the request.
it is rebind request.
the service name is RL_Server.
I got the following ror:
IP address: localhost
port num:1099
object key:0
Interface Name:InterRL
ROR is put in the table.
```

```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java Server 1099
RL_Server is ready to listen on Think-PC at port 1099
```

```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java SimpleRegistryServer
server socket created.
```

```
accepted the request.
I was asked who I am, so I answered.
```

```
accepted the request.
it is rebind request.
the service name is RL_Server.
I got the following ror:
IP address: localhost
port num:1099
object key:0
Interface Name:InterRL
ROR is put in the table.
```

```
accepted the request.
I was asked who I am, so I answered.
```

```
accepted the request.
it is lookup request.
the service name is RL_Server.
the service found.
ROR is localhost,1099,0,InterRL.
ROR was sent.
```

```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java Server 1099
RL_Server is ready to listen on Think-PC at port 1099
```

```
MINGW64:/c/Users/Think/Desktop/e

Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java Client 1099
socket made.
stream made.
command and service name sent.
it is found!.
localhost
1099
0
InterRL
Hello, welcome!
Do you want to Login, Register or Quit? (Please enter "L", "R" or "Q")
```



```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java SimpleRegistryServer
server socket created.
```

```
accepted the request.
I was asked who I am, so I answered.
```

```
accepted the request.
it is rebind request.
the service name is RL_Server.
I got the following ror:
IP address: localhost
port num:1099
object key:0
Interface Name:InterRL
ROR is put in the table.
```

```
accepted the request.
I was asked who I am, so I answered.
```

```
accepted the request.
it is lookup request.
The service name is RL_Server.
the service found.
ROR is localhost,1099,0,InterRL.
ROR was sent.
```

```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java Server 1099
RL_Server is ready to listen on Think-PC at port 1099
```

MINGW64:/c/Users/Think/Desktop/e

```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ java Client 1099
socket made.
stream made.
command and service name sent.
it is found!.
localhost
1099
0
InterRL
Hello, welcome!
Do you want to Login, Register or Quit? (Please enter "L", "R" or "Q")
R
Please enter user name
11611223
Please enter password
11611223
Congratulations.
Do you want to Login, Register or Quit? (Please enter "L", "R" or "Q")
L
Please enter user name
11611223
Please enter password
11611223
Congratulations, you have successfully entered the system.
Enter repeat mode. (Press "Q" to exit)
Q
Q
Do you want to Login, Register or Quit? (Please enter "L", "R" or "Q")
Q
bye
```

```
Think@Think-PC MINGW64 ~/Desktop/e (master)
$ |
```

