# Documentation

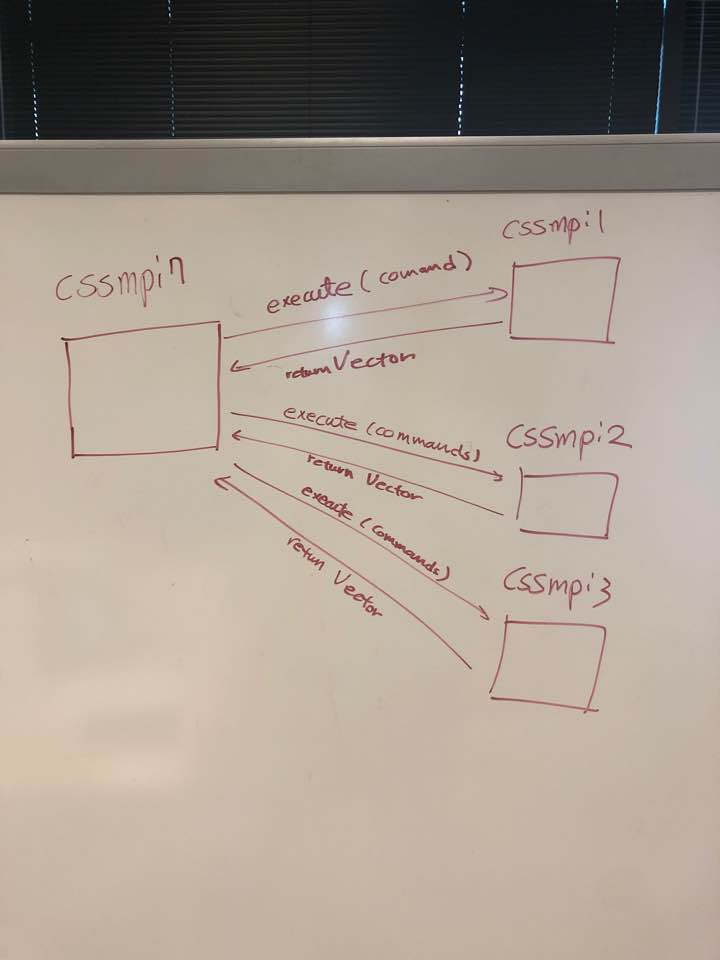
**UnixClient.java**

This program executes a list of commands in each of the server in the given server list by utilizing Remote Method Invocation technology. To begin with, the program takes arguments: print option, server port number, the number of servers, a list of server IP names, the number of Unix commands, and a list of Unix commands. Then it starts the timer. As illustrated in figure 1. For each server in the server list, this client program connects to the server and executes all the commands by remotely invoking the method (RMI). For every execution method call, server returns the result and the client saves it. After execution cycle, this program prints result or count of result after checking the requested option. Then it stops the timer and report execution time.

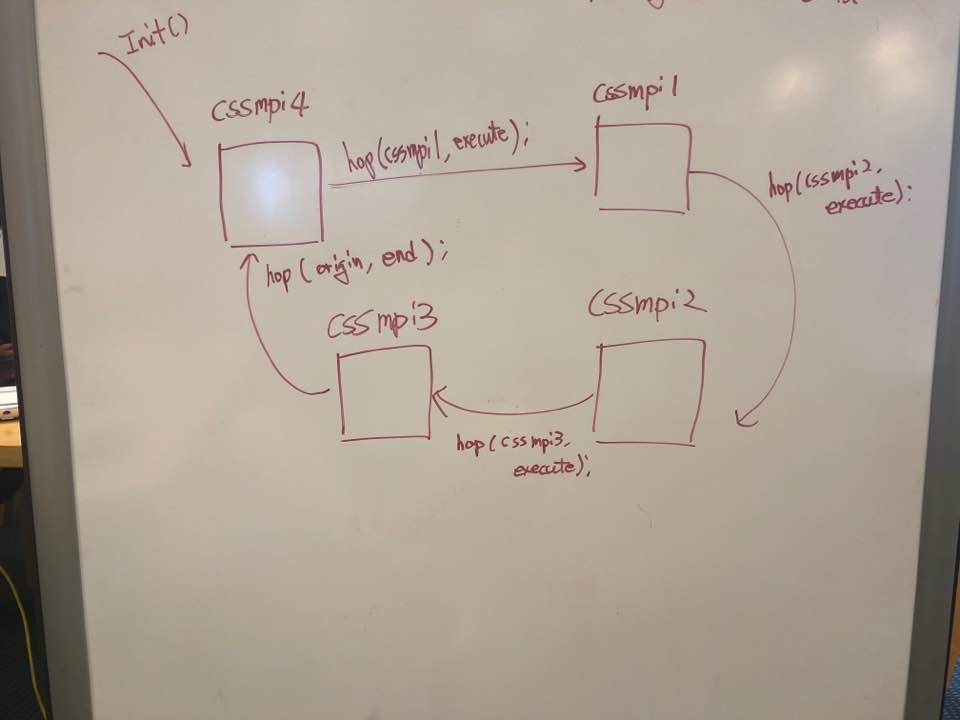
**UnixAgent.java**

This program executes a list of commands in each of the server in the given server list by utilizing Mobile Agent technology. To begin with, the program takes arguments: print option, the number of servers, a list of server IP names, the number of Unix commands and a list of Unix commands. Then, as illustrated in figure 2, agent is initiated to the initial computing node (“cssmpi4”). Once initiated, the agent will start the timer and hop to the next computing node (“cssmpi1”) and run execute() function with carrying list of commands and save the result to agent’s carrying output list. After hopping to all the servers in the given server list, agent return to the original computing node and print result or count as requested by option. Then it stops the timer and report execution time.

**Figure 1. Illustration of UnixClient.java algorithm.**



**Figure 2. Illustration of UnixAgent.java algorithm.**



# Execution output

**“rmiOutput.txt”** (full text included in the folder)

**Test 1: Executing multiple commands at remote servers**

1. Print/Count: print, Port: 34798, nServers: 1, Server1: cssmpi1, Command1: who, Command2: ls, Command3: ps, Command4: df

…

Execution Time: 224

1. Print/Count: print, Port: 34798, nServers: 2, Server1: cssmpi1, Server2: cssmpi2, Command1: who, Command2: ls, Command3: ps, Command4: df

…

Execution Time: 257

1. Print/Count: print, Port: 34798, nServers: 3, Server1: cssmpi1, Server2: cssmpi2, Server3: cssmpi3, Command1: who, Command2: ls, Command3: ps, Command4: df

…

Execution Time: 307

1. Print/Count: print, Port: 34798, nServers: 1, Server1: cssmpi1, Command1: who, Command2: ls, Command3: ps, Command4: df, Command5: who, Command6: ls, Command7: ps, Command8: df, Command9: who, Command10: ls, Command11: ps, Command12: df

…

Execution Time: 298

1. Print/Count: print, Port: 34798, nServers: 2, Server1: cssmpi1, Server2: cssmpi2, Command1: who, Command2: ls, Command3: ps, Command4: df, Command5: who, Command6: ls, Command7: ps, Command8: df, Command9: who, Command10: ls, Command11: ps, Command12: df

…

Execution Time: 404

1. Print/Count: print, Port: 34798, nServers: 3, Server1: cssmpi1, Server2: cssmpi2, Server3: cssmpi3, Command1: who, Command2: ls, Command3: ps, Command4: df, Command5: who, Command6: ls, Command7: ps, Command8: df, Command9: who, Command10: ls, Command11: ps, Command12: df

…

Execution Time: 474

**Test 2: Executing a grep at remote servers**

1. Print/Count: count, Port: 34798, nServers: 1, Server1: cssmpi1, Command1: grep -o 123 ../files/text1.txt

Count: 359

Execution Time: 266

1. Print/Count: count, Port: 34798, nServers: 2, Server1: cssmpi1, Server2: cssmpi2, Command1: grep -o 123 ../files/text1.txt

Count: 718

Execution Time: 430

1. Print/Count: count, Port: 34798, nServers: 3, Server1: cssmpi1, Server2: cssmpi2, Server3: cssmpi3, Command1: grep -o 123 ../files/text1.txt

Count: 1077

Execution Time: 488

**Test 3: Downloading a file from remote servers and thereafter executing grep locally**

1. Print/Count: print, Port: 34798, nServers: 1, Server1: cssmpi1, Command1: cat ../files/text1.txt

Execution Time: 5748

359

1. Print/Count: print, Port: 34798, nServers: 2, Server1: cssmpi1, Server2: cssmpi2, Command1: cat ../files/text1.txt

Execution Time: 12211

718

1. Print/Count: print, Port: 34798, nServers: 3, Server1: cssmpi1, Server2: cssmpi2, Server3: cssmpi3, Command1: cat ../files/text1.txt

Execution Time: 19127

1077

**“agentOutput.txt”** (full text included in the folder)

**Test 1: Executing multiple commands at remote servers**

1. Print/Count: print, nServers: 1, Server1: cssmpi1, Command1: who, Command2: ls, Command3: ps, Command4: df

…

Execution Time: 85

1. Print/Count: print, nServers: 2, Server1: cssmpi1, Server2: cssmpi2, Command1: who, Command2: ls, Command3: ps, Command4: df

…

Execution Time: 101

1. Print/Count: print, nServers: 3, Server1: cssmpi1, Server2: cssmpi2, Server3: cssmpi3, Command1: who, Command2: ls, Command3: ps, Command4: df

…

Execution Time: 129

1. Print/Count: print, nServers: 1, Server1: cssmpi1, Command1: who, Command2: ls, Command3: ps, Command4: df, Command5: who, Command6: ls, Command7: ps, Command8: df, Command9: who, Command10: ls, Command11: ps, Command12: df

…

Execution Time: 103

1. Print/Count: print, nServers: 2, Server1: cssmpi1, Server2: cssmpi2, Command1: who, Command2: ls, Command3: ps, Command4: df, Command5: who, Command6: ls, Command7: ps, Command8: df, Command9: who, Command10: ls, Command11: ps, Command12: df

…

Execution Time: 173

1. Print/Count: print, nServers: 3, Server1: cssmpi1, Server2: cssmpi2, Server3: cssmpi3, Command1: who, Command2: ls, Command3: ps, Command4: df, Command5: who, Command6: ls, Command7: ps, Command8: df, Command9: who, Command10: ls, Command11: ps, Command12: df

…

Execution Time: 234

**Test 2: Executing a grep at remote servers**

1. Print/Count: count, nServers: 1, Server1: cssmpi1, Command1: grep -o 123 ../files/text1.txt

Count: 359

Execution Time: 130

1. Print/Count: count, nServers: 2, Server1: cssmpi1, Server2: cssmpi2, Command1: grep -o 123 ../files/text1.txt

Count: 718

Execution Time: 250

1. Print/Count: count, nServers: 3, Server1: cssmpi1, Server2: cssmpi2, Server3: cssmpi3, Command1: grep -o 123 ../files/text1.txt

Count: 1077

Execution Time: 344

# Discussion

**Programmability**

* UnixClient(111) + UnixServer(48) = 159 (LOC)
* UnixAgent = 163 (LOC)

In this experiment in RMI and Mobile Agent programming, the number of lines of code is about similar. However, in mobile agent programming, it was simpler than the server/client model programming. I did not have to worry about connecting to nodes in the programming level. In addition, it felt that mobile agent programming is similar to recursion while the server/client model is similar to iteration programming. Therefore, I did not have any difficulty in programming in both ways. Mobile agent was simple and server/client model was the one that I was familiar.

**Performance**

In terms of performance, as shown in the table 1 and 2, mobile agent model performs better in all testing cases. Based on the experiment results, the exact reason cannot be discussed. The difference between the two model is communication method; Agent carrying the data while traveling vs. client saving the result locally. Therefore, presumably, this is due to the heavier communication overheads in client/server model.

**Table 1. Execution time of tests with UnixClient**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **4 Commands** | **12 Commands** | **Grep** | **DownloadFile** |
| **1** | 224 | 298 | 266 | 5748 |
| **2** | 257 | 404 | 430 | 12211 |
| **3** | 307 | 474 | 488 | 19127 |

**Table 2. Execution time of test with UnixAgent**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **4 Commands** | **12 Commands** | **Grep** |
| **1** | 85 | 103 | 139 |
| **2** | 101 | 173 | 250 |
| **3** | 129 | 234 | 344 |