Distributed Network Application Development, Spring 2015

Bucăţea Maria Cristina and Cârlig Radu Victor

June 23, 2015

Professor: Costin Bădică
Year of Study: Third Year
Group: 10305
Section: Computers in English

1 Problem statement

Mobile Code: Distributed calculus with resources from volunteers

It is considered the situation where a research group needs a lot of calculation power and their supporters want to give them acces to their home resources.

Implement a software client that can be installed by anyone to offer their personal computer resources. People who make their computers available to the research group will be named ‚volunteers'.

Implement a server to which the clients can connect automatically. This server will distribute tasks to clients as code that is going to be run on the volunteer's computer. After the code is run, the client will send to the server the result.

2 Pseudocode

Server:

StartListening()
1. **while** tasks are available **do**
2. switch clientRequest
3. case requestTasksList
4. SendTasksList
5. case chooseTask
6. SendTask
7. case fileOutput
8. print result

SendTask()
1. **if** tasks are available **do**
2. send fileInfo
3. send javaFile
4. remove task from available tasks list

SendTasksList()
1. GetTasksList
2. **if** tasks are available **do**
3. send tasksList
4. **else do**
5. send tasksUnavailable response

Client:

StartListening()
1. **while** tasks are available **do**
2. **if** not busy **do**
3. getTasksList

getTasksList()
2. **switch** serverResponse
3. **case** tasks available
4. receive list of tasks
5. sort tasks
6. choose task
7. getFile
8. **case** tasks unavailable
9. return

getFile()
1. receive fileInfo
2. receive java file
3. executeFile

executeFile(filename, extension, arguments)
1. compile file
2. run file
3. get result
4. sendResult
5. delete file

3 Application design

The application is divided into two main components: server and client. Each of these components are java files (.java). The project contains some additional java files which represent the tasks that the server can send to clients.

The clients are able to connect to the server and send requests. The supported requests are: request the list of available tasks, request a certain task, give the result of the task to server.

The server can respond to clients. The responses that are supported are: the requested task is available, the requested task is unavailable; the server is also able to send the list of available tasks and to send the file that represents the task requested if available.

Server functionality:

- void StartListening();
  This method sets up the connection between server and clients. It receives the requests from clients and calls the appropriate functions to deal with the requests. The method also receives the result of the taks sent by a client and prints it in the console.

- void CreateTasks();
  This is the method where the list of tasks is created and mapped. For each task available there is a java file that represents the code that will be sent to clients.

- String GetTasksList();
  This method concatenates all the available tasks; on a line are the three characteristics of a task: id, estimated complexity and the date when it was added. Is is used when a client requests the list of tasks to obtain the concatenated string in order to be sent to the client.

- void SendTasksList();
  This method is responsible if sending the list of available tasks to the client that requested it. It verifies if there are tasks available and sends the appropriate response to the client and if there are tasks available, it gets the concatenated string by calling the GetTasksList method and sends it through a DataOutputStream to the client.

- void SendTask();
  This method is responsible for sending the java file for the corresponding task that was requested by a client. It first receives the id of the task requested and then it retrieves and sends to the client the information about that task (name of the file corresponding to the task, the extension of the file and the arguments needed to run the file). Afterwards, the file is sent to the client and it is removed from the list of available tasks.

- Task getTaskFromString (String task);
  This method receives an available tasks as a concatenated string. It transforms the data into a Task object and returns the task.

- void main(String args[]);
  The main function creates a server object and starts listening to requests from clients.

Client functionality:

- void startListening();
  This method communicates with the server. When the client is not busy, it requests makes requests while there are still tasks available on the server. Requests are made by calling the getTasksList method that receives the list of tasks available, chooses a task, runs the code and returns the result. When there are no more tasks available, the method returns and it means that all the tasks were processed and the results were sent to the server.

- void restartStreams();
  This method restarts the streams for the client-server communication.

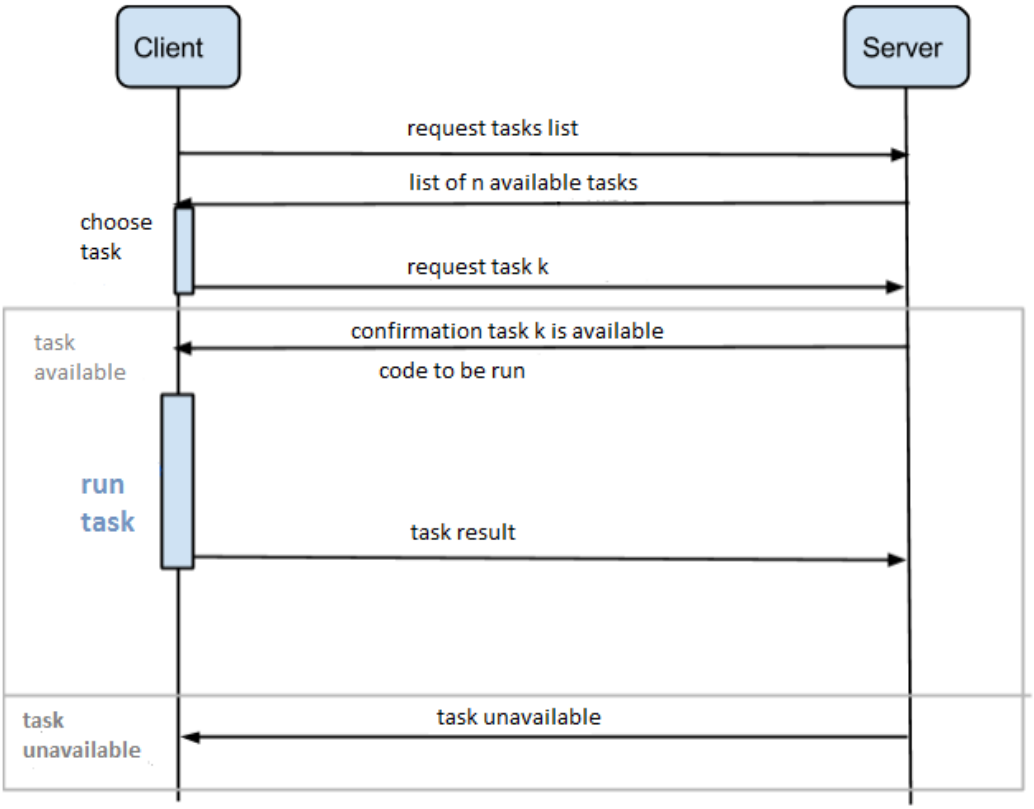
- void getTasksList();
  This method collects the list of available tasks from server. It also sorts the tasks and chooses the apropiate task to be requested from the server by calling the getFile method. The client is now set to ‚busy' until the task is done.

- void getFile();
  This method requests from server the chosen task to be resolved. Then, it receives the info about the task and the file to run. After it finishes receiving, it calls the executeFile method where the code is executed and the result is received.

- void executeFile(String filename, String extension, String arguments);
  This method compiles and runs the file corresponding to the current task. After it receives the result, it calls the sendResult method which sends the result to the server. It also deletes the file received for the current task, since the result was received and the file is no longer needed.

- void sendResult(String result);
  This method receives the result of the current task and sends it to the server from which it was requested.

- static void main(String args[]);
  The main function creates a new client object and then it call the startListening method in order to communicate with the server.



In the above picture is explained the communication protocol between the server and a client for allocating a task.

The list of tasks consists of a number of n tasks represented by an id, complexity and the date when it was added. After receiving the list of tasks, the client sorts it and chooses the preferred task. The client then requests the chosen task from the server which responds by telling if the task is available or not.

If the requested task is available, the client receives the file containing the code to be run, so it compiles and runs the file in order to obtain the result. When the result is complete, it is send back to the server.

The client continues to request tasks from the server until there are no more tasks available (the server will send a response telling the client when the list of tasks is empty).

4 Experiments and results

Inputs:

Tasks.txt

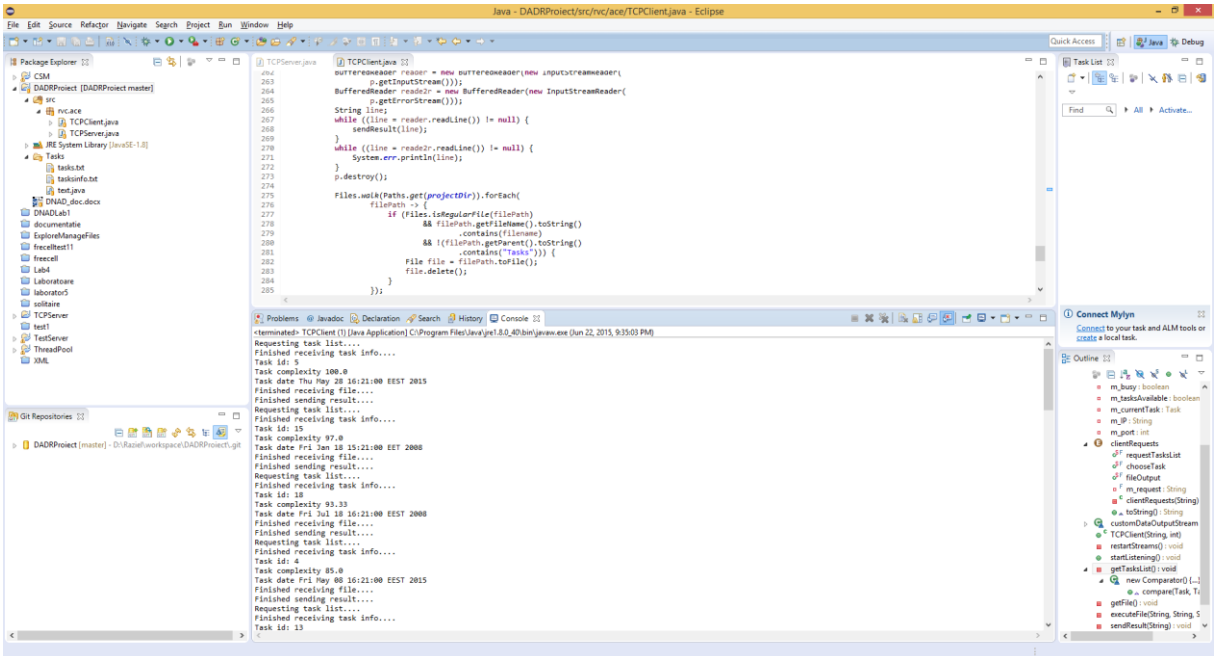
1 15 Tue May 18 16:21:00 EEST 2015
2 40 Mon May 15 16:21:00 EEST 2015
3 55 Wed May 12 16:21:00 EEST 2015
4 85 Mon May 8 16:21:00 EEST 2015
5 100 Mon May 28 16:21:00 EEST 2015
6 0.67 Wed May 10 16:21:00 EEST 2015
7 13.12 Tue May 18 16:21:00 EEST 2015
8 23.43 Mon May 18 12:21:00 EEST 2015
9 67.67 Mon May 18 1:21:00 EEST 2015
10 16 Mon May 18 16:21:00 EEST 2014
11 23 Mon May 18 16:21:00 EEST 2012
12 54 Wed May 18 16:21:00 EEST 2008
13 69 Mon May 18 16:21:00 EEST 2008
14 5 Mon Jun 18 16:21:00 EEST 2008
15 97 Tue Jan 18 16:21:00 EEST 2008
16 43 Mon Sep 18 16:21:00 EEST 2008
17 45.67 Mon Oct 18 16:21:00 EEST 2008
18 93.33 Wed Jul 18 16:21:00 EEST 2008
19 66.67 Tue Jul 18 16:21:00 EEST 2008

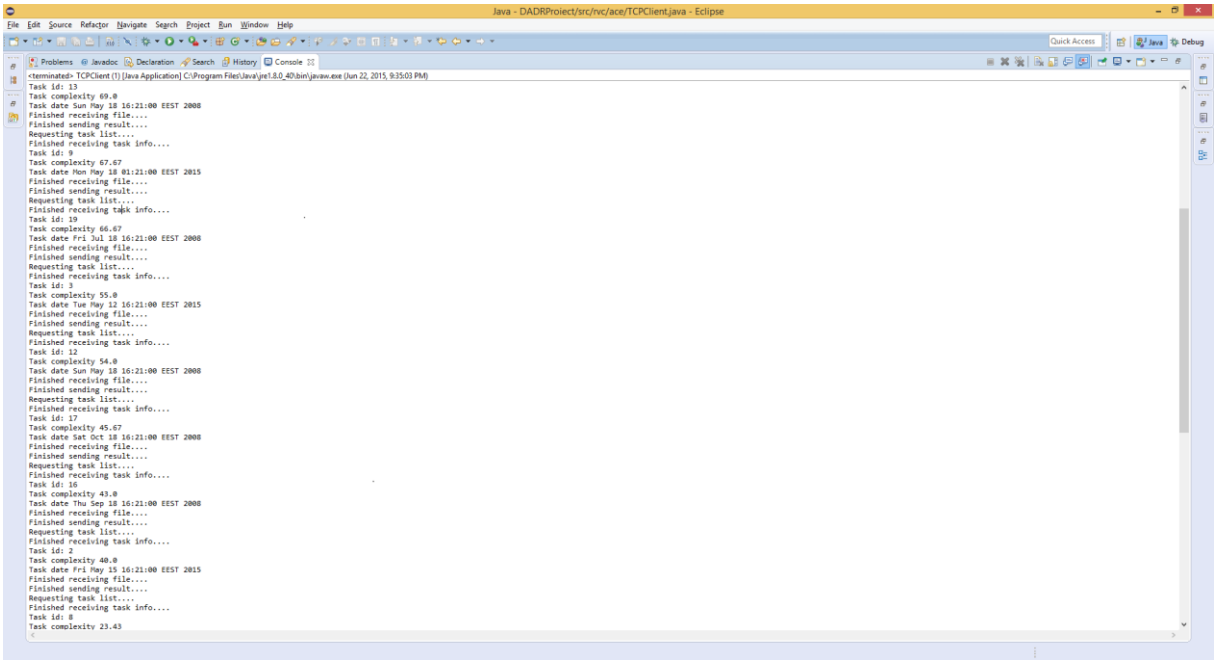
TasksInfo.txt

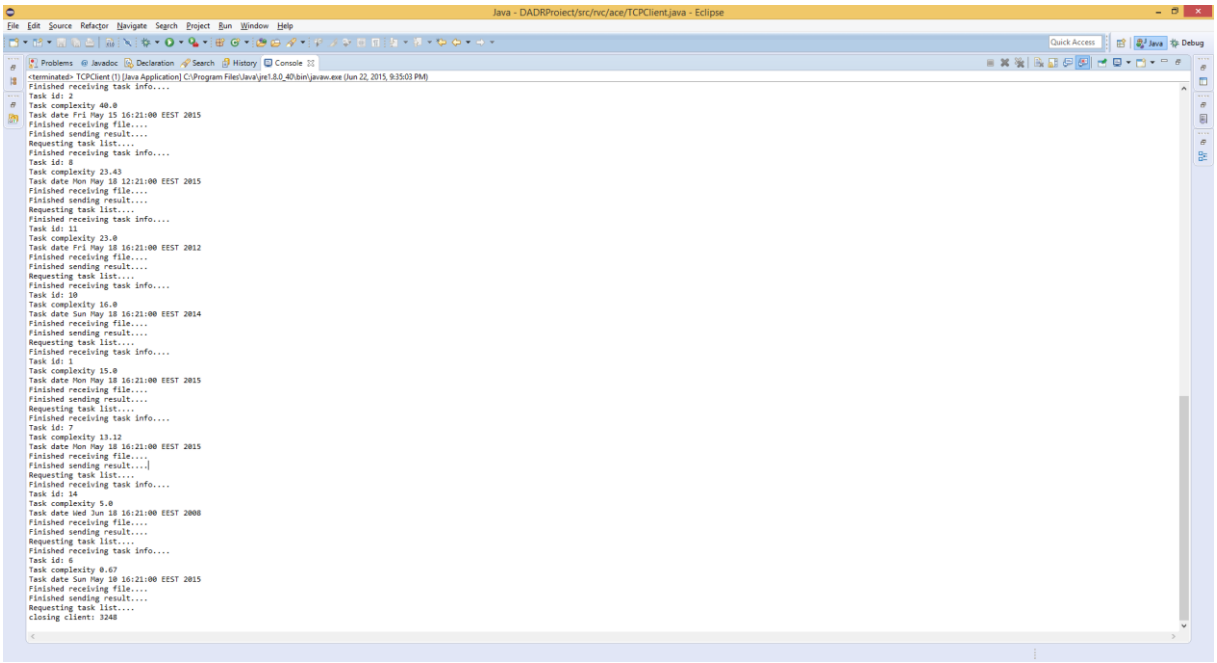
1 password java ef797c8118f02dfb649607dd5d3f8c7623048c9c063d532cc95c5ed7a898a64f
2 password java 5db1fee4b5703808c48078a76768b155b421b210c0761cd6a5d223f4d99f1eaa
3 password java bb7e624ad2dd0c99c98b7fda286892a69b4db6d52806f4fa7f7603d347cfd412
4 text java f10d91a7596bf5a6773579ff1306afdc363b0be08602c768907c09261cad3a56
5 text java 5994471abb01112afcc18159f6cc74b4f511b99806da59b3caf5a9c173cacfc5
6 text java 82d5de89e8420ba9c0baf51d1aec8c4322de27065c445dfa180dedabf44b3a8f
7 text java 3538a1ef2e113da64249eea7bd068b585ec7ce5df73b2d1e319d8c9bf47eb314
8 text java 3f08d8fadb4b67fb056623565edbbc2c788091d78fd24cbc473fce3043ce3473
9 text java d17f25ecfbcc7857f7bebea469308be0b2580943e96d13a3ad98a13675c4bfc2
10 text java d49b4ad6f32ed006d77d7381a405adaa3d71c890b06c9476806da3c172930131
11 text java bea9bd1ca4843718af9d158e7f70acdcf23e6cbe486a3ae6730baa396058664e
12 text java f1a86993bd8fb1a254cf95ea0ca37d98057d6e6e0aa6cd74764a55bf57ccc9a9
13 text java f8dbd9a22b5bd7848b6fc0a3f52543e190a5d6270fb0321aac80a795baaf6edc
14 text java 4fe5b89dd1cab5383b8484593b81a55e22a368144a3efe812d5ab58a0740b668
15 text java b85bf0f7330be07933314afcfc04aa8e8bb33827eb03bdf2f65ff26fd32444f5
16 text java dcac43c6d223b17998d10876edc0acd69128bb3fc90bf749dc942d440d74571c
17 text java f14f286ca435d1fa3b9d8041e8f06aa0af7ab28ea8edcd7e11fd485a100b632b
18 text java 4e20613ec3a63732f98630b9429e1600d47a08f059496d39ffd353f0c3038f39
19 text java cc2e018aa6eb9612ccd027bbdcdc9b8c8d351789f14cae4d688a876c18938235

Client results

The client takes the most complex task from the list received from the server and executes them.




Java - DADRProiect/src/rvc/ace/TCPClient.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Search History Console
<terminated> TCPClient (1) [Java Application] C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (Jun 22, 2015, 9:35:03 PM)
Task id: 13
Task complexity 69.0
Task date Sun May 18 16:21:00 EEST 2008
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 9
Task complexity 67.67
Task date Mon May 18 01:21:00 EEST 2015
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 19
Task complexity 66.67
Task date Fri Jul 18 16:21:00 EEST 2008
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 3
Task complexity 55.0
Task date Tue May 12 16:21:00 EEST 2015
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 12
Task complexity 54.0
Task date Sun May 18 16:21:00 EEST 2008
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 17
Task complexity 45.67
Task date Sat Oct 18 16:21:00 EEST 2008
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 16
Task complexity 43.0
Task date Thu Sep 18 16:21:00 EEST 2008
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 2
Task complexity 40.0
Task date Fri May 15 16:21:00 EEST 2015
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 8
Task complexity 23.43


Java - DADRProiect/src/rvc/ace/TCPClient.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Search History Console
<terminated> TCPClient (1) [Java Application] C:\Program Files\Java\jre1.8.0_40\bin\javaw.exe (Jun 22, 2015, 9:35:03 PM)
Finished receiving task info....
Task id: 2
Task complexity 40.0
Task date Fri May 15 16:21:00 EEST 2015
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 8
Task complexity 23.43
Task date Mon May 18 12:21:00 EEST 2015
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 11
Task complexity 23.0
Task date Fri May 18 16:21:00 EEST 2012
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 10
Task complexity 16.0
Task date Sun May 18 16:21:00 EEST 2014
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 1
Task complexity 15.0
Task date Mon May 18 16:21:00 EEST 2015
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 7
Task complexity 13.12
Task date Mon May 18 16:21:00 EEST 2015
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 14
Task complexity 5.0
Task date Wed Jun 18 16:21:00 EEST 2008
Finished receiving file....
Finished sending result....
Requesting task list....
Finished receiving task info....
Task id: 6
Task complexity 0.67
Task date Sun May 10 16:21:00 EEST 2015
Finished receiving file....
Finished sending result....
Requesting task list....
closing client: 3248

Server results