COMP90015 Distributed Systems
Project 2 - Extended Multi-Server Chat System

Yuchen Qiao 748663
Kaixuan Ni 781111
Kaili Wang 739079
Chang Liu 741011

October 20, 2016

1 Introduction

The objective of Project 2 is to extend the Multi-Server Chat System in the last assignment. The project primarily focuses on addressing the following challenges of the system: Security, Failure and Scalability. Bisides, the program of this project achieved a Graphical User Interface (GUI) for the Chat Client. In this report, all part of this project will be described, and the relevant diagrams will be displayed.

2 Security

Kaixuan Ni is in charge of design and implementation of Security.
The technology which mostly used in the security section is the SSLSocket, which is a kind of security technology using an encryption strategy. In this project, the connection between server and client is required to be secure and protected so that the SSL connections would be the most secure channel. Just using the KeyStore command to generate the KeyStore file, then set the path and the password as an argument in Java code which using the KeyStore as SSL certificate. Therefore, all the established connections are based on the KeyStore and only the terminal that set the KeyStore could connect to each other. Another issue is the authentication mechanism, which means login checking. A file has been built to maintain the registered users. When a user wants to establish a connection to the server, the user needs to enter the username and password first then the server verifies its correctness. If the user existed, it is allowed to choose a port and an identity to log into the different chat server. Change all the communication via the secure channel; all the Socket are changed to SSLSocket. Add the code for the authentication mechanism and using a file to maintain it. Change client code to support the authentication mechanism. Draw the sequence diagram. Helping debugs and merge the code.

3 Failure Handling

Kaixuan Ni, Kaili Wang, Chang Liu are in charge of design and implementation

of Failure Handling. On the other hand, this statement is writen by ***Chang Liu***.
In this part, with the help of my group mates I implement the following functions.
First, using heartbeat signals generated at regular intervals to check if the other servers are working correctly. Second, if a chat server crashes or stops responding, all information of this server will be removed and other chat servers will print the error message. At last, all chatrooms on that server should be removed from the list and clients will not be redirected to that server anymore.
In this program, all server information is stored in the all_ServerInfo. And I use this as the initial server state. In order to get the server which is crashed, I write a heartbeat method to send heartbeat single. In heartbeat method, server will send heartbeat message to other servers every 5 seconds. And if there is ConnectException happen when sending the heartbeat signal, this exception will be caught. The information of this server will be removed from the all_ServerInfo, and the chatrooms located on this server will be removed from the remote room list.
Clients will not be redirected to that server because the information of this server was removed from the all_ServerInfo.

4 Scalability

Kaili Wang is in charge of design and implementation of Scalability.
For the sake of scalability, my task is that the system can add new servers to the system manually. Therefore, my operation is:

(1) Change serverInfo class:
In the project1, server has attributes: server id, address, client port and coordination port. Besides, I add new attribute of server: server status. When the server status is on, it means the server has entered the system already. Otherwise, the server does not introduce itself to the system yet.

(2) Send JSON message:
When new server introduce itself to the system, if the servers information already exists in the configuration file, its server status change from off to on. Otherwise, User input new servers information manually to the system through keyboard, like server id, host, client port, and coordinate port. Then, the server sends heart beat message to all other servers to check whether other servers status is on or off. Furthermore, it sends JSON message with server id to other existing servers. Existing servers receive message, add remote chatroom main-newserverid. At the same time, existing servers reply JSON message to new server with their allroomlist information including local rooms and remote rooms. New server receives the information of other servers all room information and add them to its remote chatroom. In addition, existing servers reply current all server information to new server.

(3) Change servers connection:
As for connection among servers, when one server sends JSON message to other servers, it checks the status of other servers firstly and send JSON message to servers whose server status is on.

(4) Failure handling:
Once certain server crashes or stop responding, other servers should detect the situation because of heartbeat class. The server status of crashed server should change from on to off. Furthermore, active servers should delete the crashed servers room from remote room list.

5 GUI

Yuchen Qiao is in charge of design and implementation of GUI.
As for the selected technologies, packages inside java: AWT and SWING are used.

To be precise, creating two subclasses WelcomePanel and MainPanel which inherit from the superclass JFrame. The use of the WelcomePanel is creating the panel for the user to login, and the use of the MainPanel is creating the panel for chatting. To be identifiable, all panels use JLabel as the label of the input text field. After that, using JTextField to accept input from the user. It is worthy to note that the WelcomePanel used JPasswordField to accept the input of the pin, which can hide the real password while typing. Besides, there are three buttons on the WelcomePanel. First is login, if one user clicks this button, it will switch the user to the chat panel and start connecting the target server base on the information from all text fields. Second is exit, the program will shut down. And the button with label debug is settled as an option of the debug mode.
As for the MainPanel, an object of JTextArea has been used for display in the message. And an object of JTextField is under the responsibility of providing a field which can accept the input order from the user. Moreover, there are three buttons lying on this panel. If user clicks send, it will go to get the message from the text field and start send it to server. If user clicks clear, all message on the text area will be eliminated. What's more, if user clicks exit, this panel will treat this behavior as sending quit to server, then the server will deal with this issue.

6 Conclusion

This project primarily focuses on addressing the following challenges of the system: Security, Failure and Scalability. Bisides, the program of this project achieved a Graphical User Interface (GUI) for the Chat Client. The results from test shown that all components work well. As for the feature study, the main target the next stage should focus on how to build a central authentication server. Besides, it is worthy to achieve the GUI of server to raise the user performance to the next level.

wordscount: 1213

Appendices

A Sequence Diagram

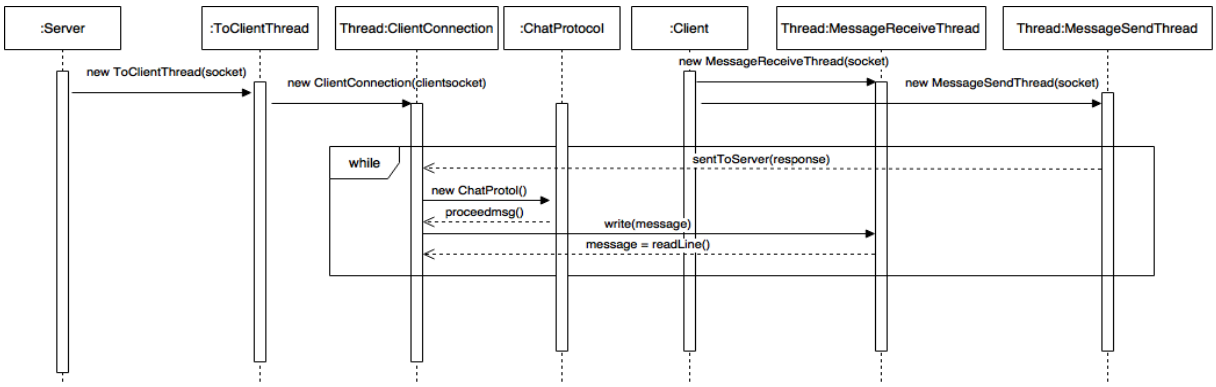

Figure 1: The Sequence Diagram indicates the process of Server to Client

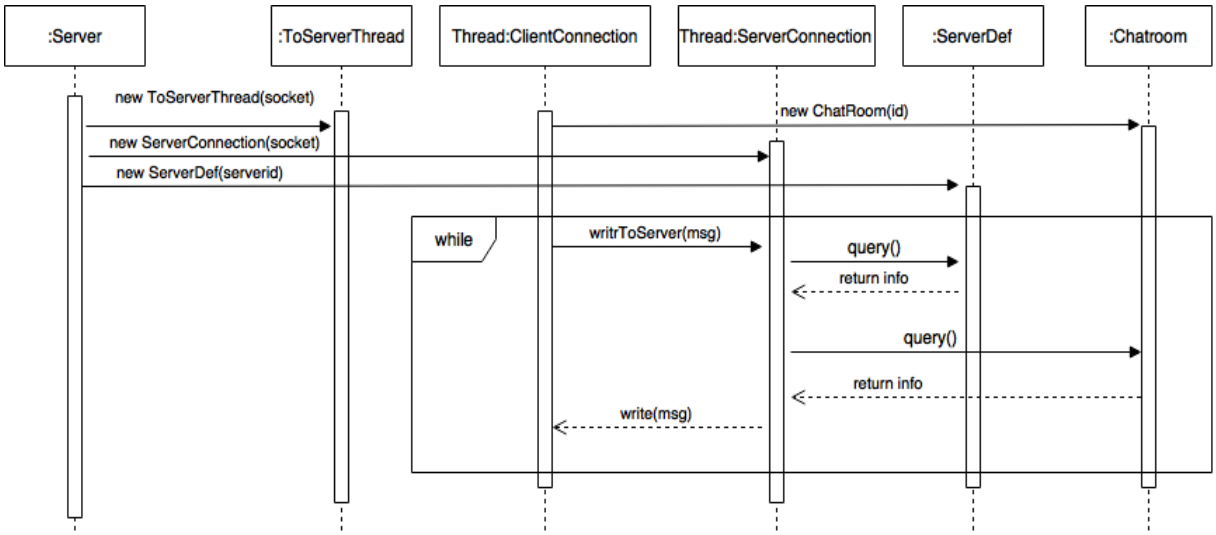

Figure 2: The Sequence Diagram indicates the process of Server to Server

B Architectural Model

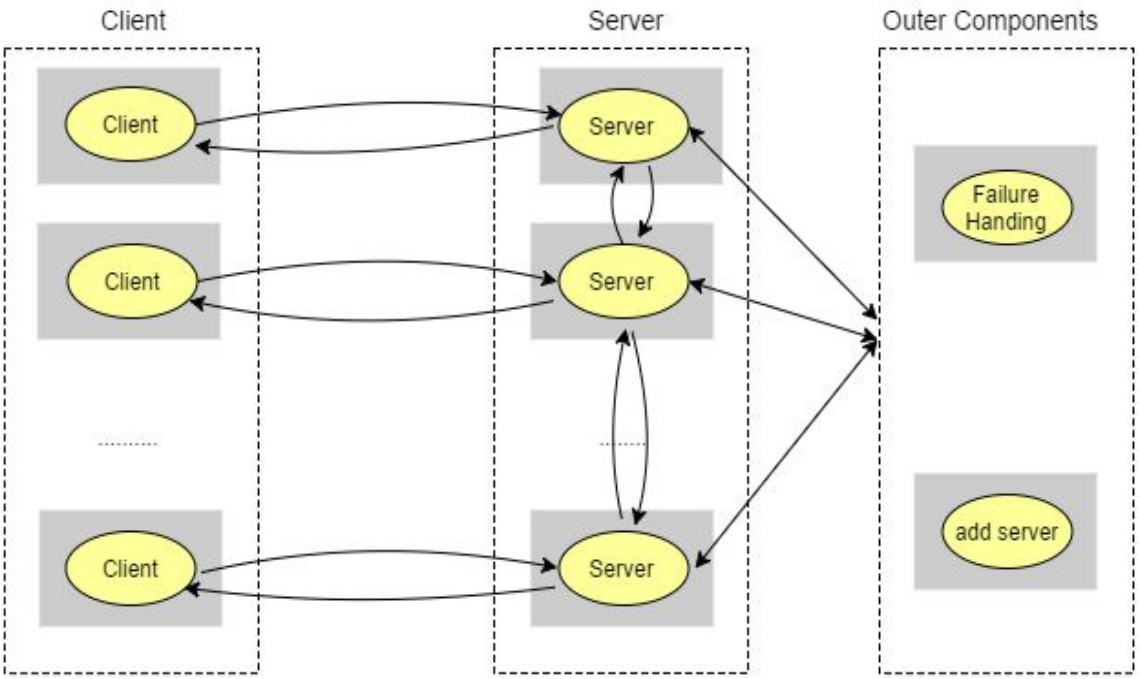

Figure 3: The architectural model of this program