

Distributed Chat Room

Tanay Ghirnikar
IIT2017107

Shivam Agrawal
IIT2017110

Prashik Raut
IIT2017137

Pranav Jhavar
IIT2017141

Mohit Dhillon
IIT2017029

Abstract — The paper presents an approach to develop a distributed chat room. The chat room consists of various functionalities such as joining a new group , create a new group , admin privileges to accept or reject group joining requests and main focus has been made to implement group chat with proper ordering of messages to each user in the chat.

Keywords : Distributed, client, server,

I. INTRODUCTION

Important applications in our lives now, with the widespread use of the Internet is the chat, because of the use in many sectors, scientific, educational, training, and social networking. To understand the idea of the chat, explaining the basic chat model:

1. Client-server computing or networking
2. Socket:A simple chat application is based on Client-Server chat where the client registers itself using a username on the network and two or more users (Client) communicate over the network The server is always ready to listen to any request and waits for the clients to connect. The clients connect to the server and start communicating with each other clients, but no one knows the other client's IP address, only the server IP address they know.

The whole system is divided into 2 parts: Client-side and Server-side. A server runs an infinite loop to keep accepting incoming requests. The server also stores the client's name into a data structure to keep a track of connected devices. The data structure stores the thread object corresponding to the current request. This data structure is further used to find where a message is to be delivered. A client should readily receive a message whenever it is delivered to it. In the client-side, there are majorly 2 threads: SendMessage
Read Message
Communication occurs with the help of the read message and sendMessage threads. Separate threads for reading and writing ensures simultaneous sending and receiving of messages.

This report further contains -
II Problem Definition
III Experimental Setup
IV Approach
V Result
VI Conclusion
VII Scope

VIII Acknowledgements

IX References

II. PROBLEM DEFINITION

The aim is to create a distributed chat facility which would be able to implement the system to handle multiples clients(users) into the chat room . The order of messages received by the users in the chat has to be same. Also additional facilities for the admin to handle the group joining requests is to be implemented. User must also be able to join or leave a group as per user's wish.

III. EXPERIMENTAL SETUP

Language used : Java

System Pre-requisites : The system must contain Java Environment and MySQL database installed.

Protocol used : Java RMI for RPC.

IV. APPROACH

4.1 : Architectural Design

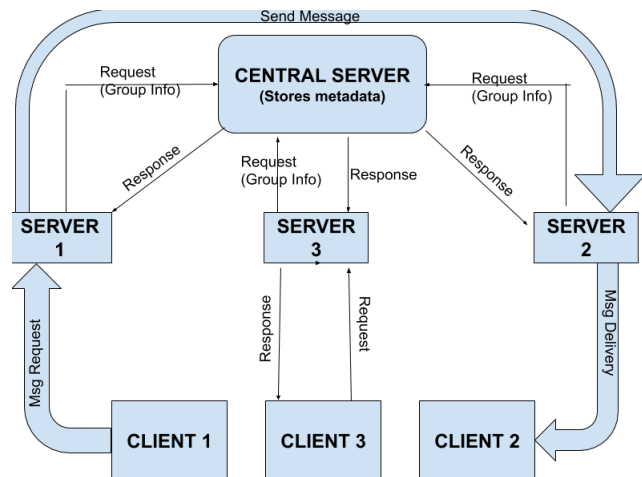


Fig. 1. Architecture

The architecture of Chat room consists of the following components :

- 1) Main Server
- 2) Side Server
- 3) Client

Main Server

The main server contains the meta data . It has the

information of which side server handles which client. The Main Server is responsible to implement the overall execution of the distributed chat facility, it is the job of main server to connect the clients(users) to the side servers which eventually connects to the main server to exchange the messages. Thus, main server is directly or indirectly connected to the other components and coordinates with all of them to give group chat facility.

Side Server

The side servers act as intermediates between the client and the main server. Each side server is one to one connected to the clients and helps the main server to identify each client by coordinating the messages given by the clients in group chat system. Also order of messages is maintained by main server using the side servers to update their clients message box each time a user sends any message.

Client

The client takes care of the message sending and receiving utilities by the user. One client per user maintains chat record of the user and coordinates with the side server to connect to the group chat implemented by the main server.

MySQL Database

The database stores the information regarding the person and the groups the person is present in, also admin details are stored. Database consists of three tables:

Info	
id	password

Fig. 2. Table Info

Table Info :

This table consists of the usernames and passwords of the people to login to the system developed.

Persongroup		
group name	person	admin

Fig. 3. Table Person group

Table Persongroup :

It consists of the group details, it has name of the group, the persons present in the group and also the information of admin of the group.

Requests	
id	group name

Fig. 4. Table Requests

Table Requests :

The requests keep tracks of the requests made by a user to enter a particular group, it has user id and the group name requested for.

4.2 Algorithmic Flow And Functionality

The client first requests the main server which assigns a side server to the client, thus the main server contains the information of each client and its group and also the one to one mapping of the side server to client.

Now, one to one chat is implemented as a special case of group chat with 2 users, the client sends messages to the assigned side server which takes the message to the main server, the main server sends the messages to the side server of the client to which message has to be sent. Finally the side server sends the message to the assigned client.

To implement the group chat multiple servers are one to one connected to the main server by the process discussed earlier. At each point a user sends a message in a group chat and the messages is delivered to the main server by the respective side server, the main server sends the messages to each client in the group through the respective side servers. Also, to maintain the order of messages has been timestamped with the first server contact (using server's time) to ensure the same order to all clients.

Functionalities Developed

- 1) Create a group as admin
- 2) Join / Leave a group
- 3) One to One chat facility
- 4) Group chat facility
- 5) As an Admin, Reject/Accept request by user to join group.

V. RESULTS

The Project was successfully able to implement the functionalities discussed, also to enhance the user interface, Java based GUI was developed.

The login screen takes the username and password.

The home screen gives user to get added in a group, leave a entered group and also accept and reject group requests if he is an admin.

It consists option to request a group to be added and to start a group chat.

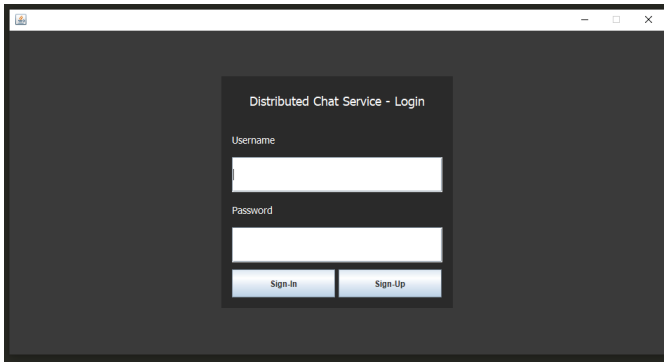


Fig. 5. Login Page

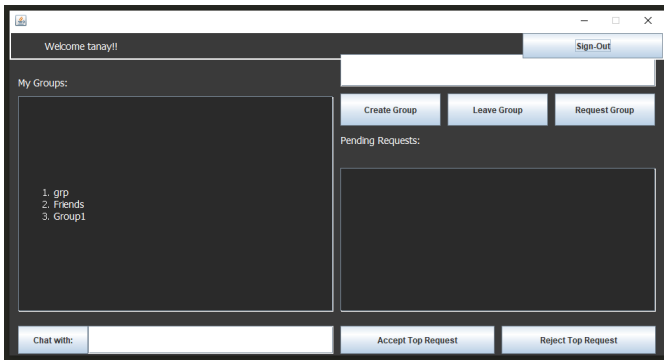


Fig. 6. Home Page

The chat screen shows the Group chat box to interact with the other users in the group and message sending option.

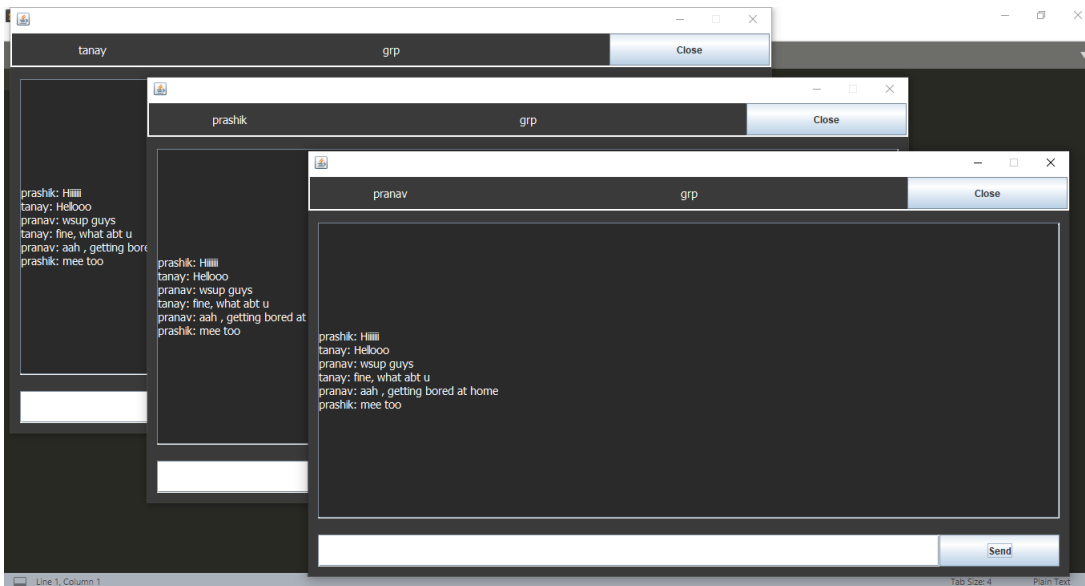


Fig. 7. Group Chat Screen

The the project successfully implements one to one and group chat utilities with proper messages ordering and a very user friendly interface.

VI. CONCLUSION

The chat utility developed uses the distributed one to one server client approach to implement group chat .The main server handles important functionalities such as similar order of messages for each user and contains information of each client , its groups and respective servers.Thus, the architecture effectively implements the distributed chat features.

VII. SCOPE

The developed system is able to implement basic one to one and group chat facility. Still a lot of work can be done to improve the scalability of the developed project , handling a lot of users needs improvement. Also, security improvements can be introduced as users privacy needs to be maintained . Video and audio messages can be added.

VIII. ACKNOWLEDGMENTS

We sincerely thank Dr. Anshu S. Anand sir for providing us the opportunity to work on Distrubuted Chat Room.

REFERENCES

- [1] <https://thescipub.com/PDF/jcssp.2015.723.729.pdf>
- [2] http://www.skjegstad.com/papers/milcom11_distributed_chat.pdf