

Executive Summary

Distributed Chat Application

Project Overview:

The main goal of this project is to build a distributed chat application using the PAXOS protocol and group communication in which the users are able to chat with each other and in groups. They are able to exchange text messages and files with each other. The user can join a group, create a new group and can come back online after going away for a while. We have used PAXOS protocol to gain consensus when the user tries to join a group or tries to come back online after being offline for a while. In PAXOS protocol, there is Acceptor, Learner and Proposer which exchange several messages between them in order to make a decision and reach consensus. Since, we have used Java, it has inbuilt RMI which we have leveraged here for the project. Further, the other goal of this project is to build highly scalable fault tolerant distributed chat application and gain hands on experience with building scalable distributed systems. Our application also has a simple yet robust user interface which helps the user to perform necessary applications. For this project, we built a peer to peer network which connects users together and form groups. Further, consistency throughout the application and being highly fault tolerant is achieved by the PAXOS protocol.

Technical Impression:

This project aims to implement the PAXOS and group communication protocols at core for distributed group chat. It involved understanding of how PAXOS works behind the scenes and how it is superior to 2-phase commit protocol. Further, the concepts of multi-threading and mutual exclusion were necessary to implement the requirements for the project. PAXOS has several advantages over 2PC in terms of availability, fault-tolerance, latency, scalability, and flexibility. The protocol works by having nodes send messages to each other proposing values and acknowledging receipt of proposals. If a majority of nodes acknowledge a proposal, it is considered accepted, and the value is chosen. If multiple proposals are made, nodes will choose the proposal with the highest sequence number.

The PAXOS protocol has three distinct phases: proposal, acceptance, and learning. In the proposal phase, a node proposes a value to be chosen. In the acceptance phase, nodes acknowledge the proposal and agree to choose the proposed value if it receives enough acknowledgments. In the learning phase, nodes inform each other of the chosen value. This is how PAXOS protocol works. We have used multiset library of java to achieve group communication. The UI is designed using Java Swing. Further, we leveraged Java RMI to achieve RPC and multithreading.

We learned a lot while implementing this project. The multiset part of the project was new to us even though we had implemented PAXOS for project 4. We also got a chance to learn about UI libraries in Java.