Peer-to-Peer Chat

Foundations of Distributed Systems
Lab Guide 5

2020/2021

Consider a simple peer-to-peer chat program using Java and Atomix, where lines are sent directly to all other peers.

Steps

1. Implement the chat peer using the simplest strategy possible.
2. Add causal delivery to chat messages.
3. (Optional) Make each peer accept client connections, turning it into a super-peer (i.e. both a peer and a chat server).
4. Change the chat to totally order messages.
5. Refactor the ordering code in 2. and 4. as an encapsulated and interchangeable layer.

Questions

1. Could this application be implemented with scalar clocks?
2. Consider using the peer-to-peer chat system for online auctions. How to determine the winner with equal bids?
3. In what cases does each version block (i.e., stops delivering messages)?

Learning Outcomes Identify inconsistent state observation in a distributed system. Recognize the importance of logical time in achieving consistency. Employ vector clocks and causal delivery in a distributed application. Relate causal to total order and a replicated state machine.