Chat

Distributed Systems Paradigms
Lab Guide 1

2021/2022

Consider a simple multi-threaded chat server using Java and NIO sockets, where lines sent by any client are broadcast to all currently connected clients.

Steps

1. Implement the server using a simple thread-per-connection strategy.
2. Implement an interactive client (or use `nc`) to interact with the server.
3. Implement a non-interactive client to generate load (*bot*) that sleeps a configurable amount of time between sending or receiving messages.
4. Run clients with different delay configurations.
5. Reconsider threading strategy to avoid blocking writers.

Questions

1. How does one client affect other clients?
2. How do clients affect server memory usage as observed with jconsole?

Learning Outcomes Recall basic distributed systems programming with Java, sockets and threads. Relate interactive performance and memory usage with server programming. Apply NIO sockets and byte buffers reduce memory usage.