SHARED FILE SYSTEM

Nitin Anvesh

24M0766 – Nitin Chakravarthy

24M0799 – Anvesh Merugu

https://github.com/24M0766/CS744_Project

Project Presentation CS744 Autumn 2024

CONTEXT

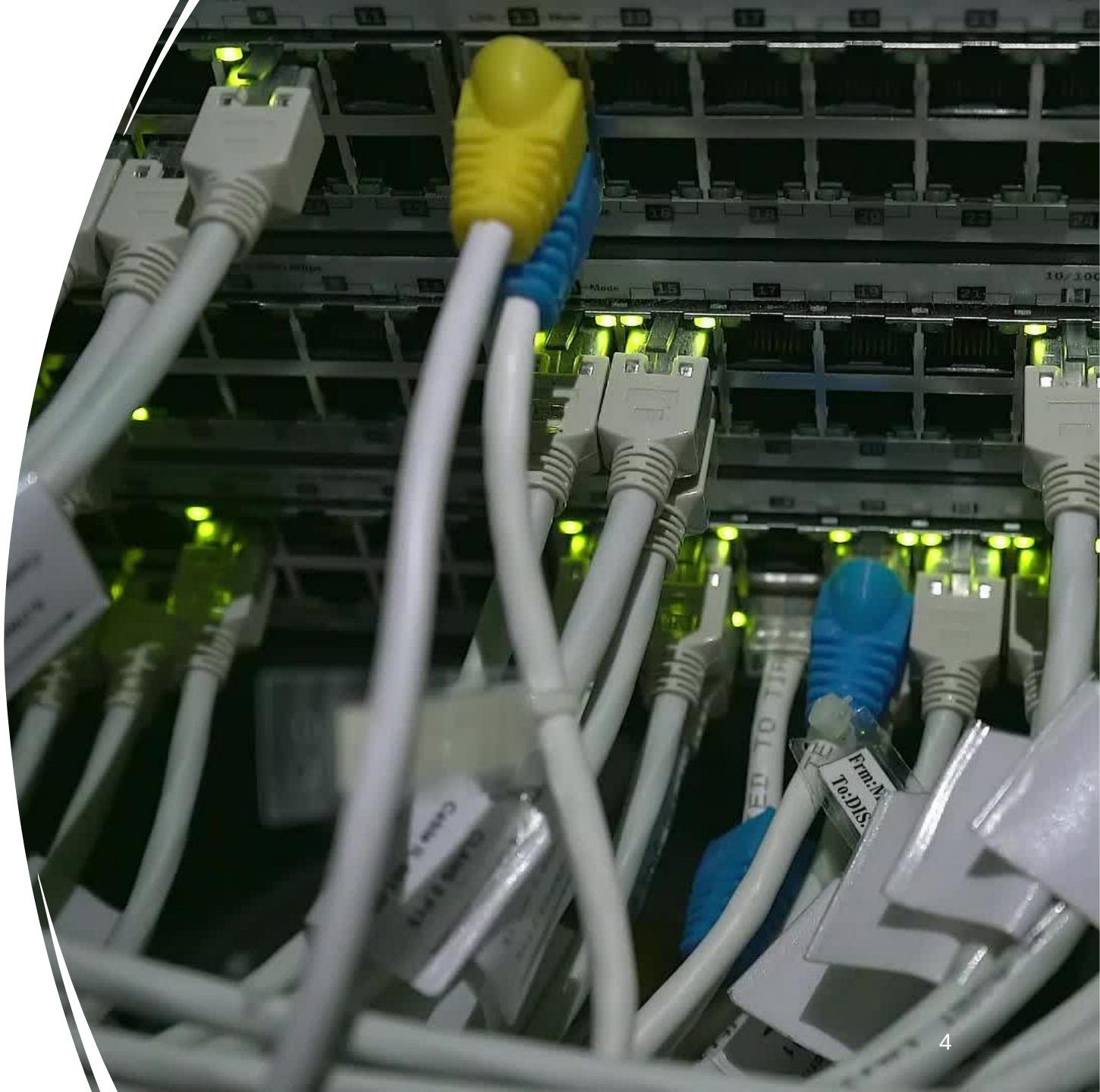
- This project falls under Cloud Computing
- We are trying to build a Shared File System
- Multiple Users can access the File System concurrently

Problem Statement

- Statement: Design a file-sharing service that can be accessed by multiple users for read, write, create and delete. Users connect to the service via a client process and can browse, checkout, commit, add, delete, writelock files to the repository.

Components of Project

- We have two processes: Server and Client
- Building Blocks: Threads, Concurrency Control, Server-Client Communication



Design

- The entire design revolves around how the Server and Client communicate
- Every New connection from a Client will be handled by a new Thread in the server
- All communication will be initiated by the Client while the Server Thread will wait

Implementation Details

Upon starting, the server waits for incoming connections

Communication will be initiated by a client

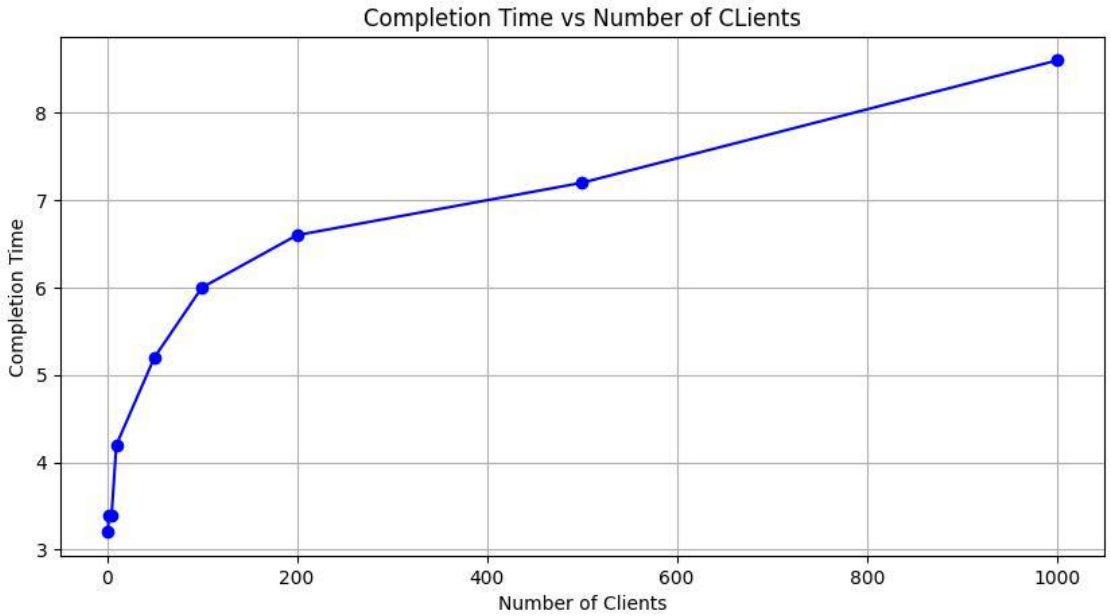
Every action starts with the client sending a request and the server responding accordingly

We will use locking mechanisms to ensure the consistency of the files

Evaluation

- What is the completion time for a request when multiple users are connected?
- Setup: We will increase the number of connected clients from 1 to 1000 and note the completion time for a request.
- Independent Variables: Number of Clients connected to Server
- Metrics: The completion time
- The request that we will focus on is READ

Completion Time vs Number of Clients



Summary of Results

- The results show that with an increasing number of clients, the completion time also increases
- This is the expected behavior as the server gets more busy with an increased number of clients
- Unable to increase number of clients as server is refusing connection beyond 1000 clients

Unfinished Scope

- Add more robust Consistency Measures
- Make the User Authentication a little more sophisticated
- Add a better notification system to intimate clients of changes

Challenges

- Making the communication between the Server and Client to be in sync
- The handling of book-keeping information for smooth and easy communication

Reflection

The aspect of inter-process communication and how simple yet complex it is

The amount of additional information that is being handled that is unknown to the users

Try to implement more robust locking mechanisms and synchronization techniques

Implement a more efficient client handling techniques

Conclusion

- Made us realize how important synchronization is in inter-process communication
- Understood how efficient and easy-to-use threads can be
- Shed light on how just two programs can create a full-fledged file system