COL334: Assignment 2

Shivam Kumar 2021CS10917, Suryanshu Verma 2021CS10962,
Raja Kumar 2021CS10915, Rishabh Kumar 2021CS10103

September 2023

Distributed P2P File Sharing System Assignment Report

In this assignment, we implemented a distributed P2P (Peer-to-Peer) lines sharing system. The primary objective was to design and implement a distributed algorithm that allows multiple clients to share and download files in a decentralized manner. This report provides an overview of the distributed algorithm, explains the scenarios we handled, and presents test results with different numbers of clients.

We have implemented a client-server network in which the server accepts connections from multiple clients simultaneously. The client program repeatedly sends SENDLINE requests to the vayu.iitd.ac.in server and receives the lines in the form of bytes. We used a maximum packet size of 1024 bytes to ensure that each line does not take more than 2 transmissions to reach the client.

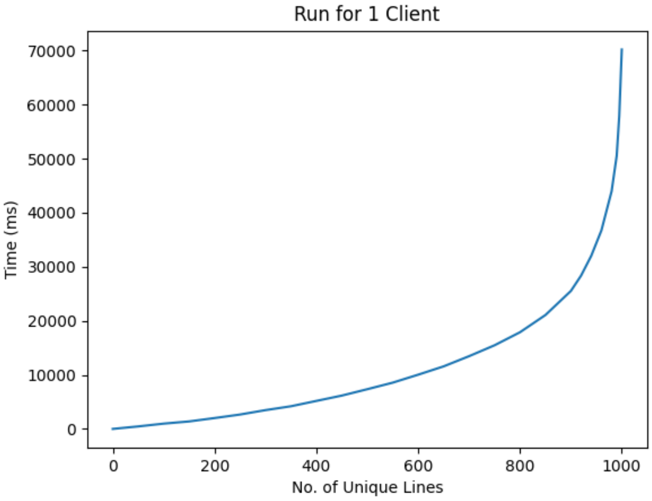
To optimize the client program, we used a dictionary of line numbers to ensure that only unique lines are sent to the server.

Additionally, the server program opens a parallel thread for each new client. Once the server receives all the lines, the next task was to efficiently distribute all those lines to all the clients. We implemented a handshake mechanism to ensure that each line is received by the client, which then sends back a confirmation signal.

We conducted tests with varying numbers of clients while keeping one server (except in the case of a single client where there was no need for a server). The tests included the following scenarios:

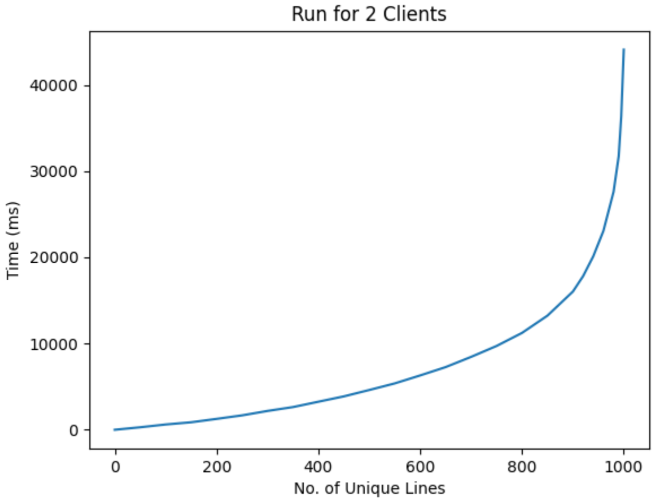
- **Single Client Download:**
  - Total Time Taken: 70209ms

Unique Lines	0	100	200	300	400	500	600	700	800	900	960	995	1000
Time (ms)	0	976	2008	3474	5179	7345	10035	13459	17886	25527	36742	57903	70209



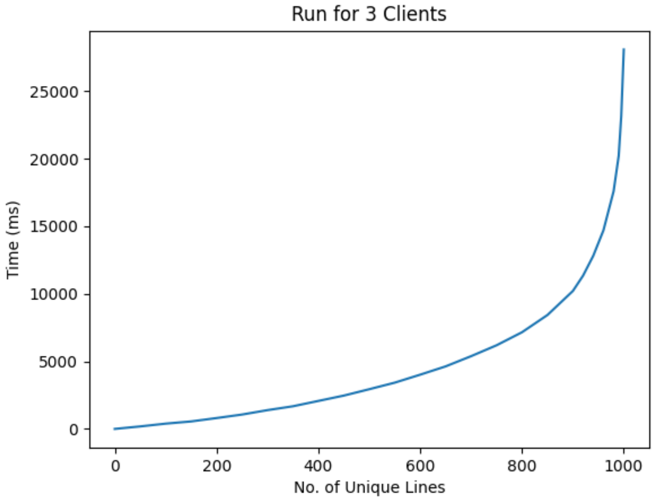
- **Two Clients Downloading:**
  - Time Taken: 44091ms

Unique Lines	0	100	200	300	400	500	600	700	800	900	960	995	1000
Time (ms)	0	612	1261	2181	3252	4612	6301	8452	11232	16030	23073	36363	44091



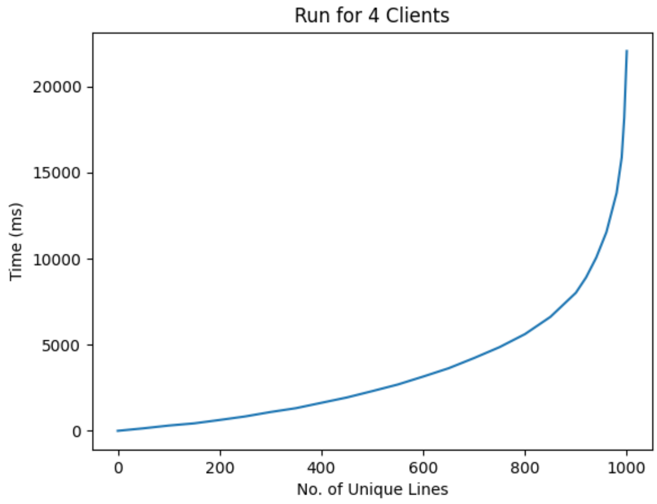
- **Three Clients Downloading:**
  - Time Taken: 28083ms

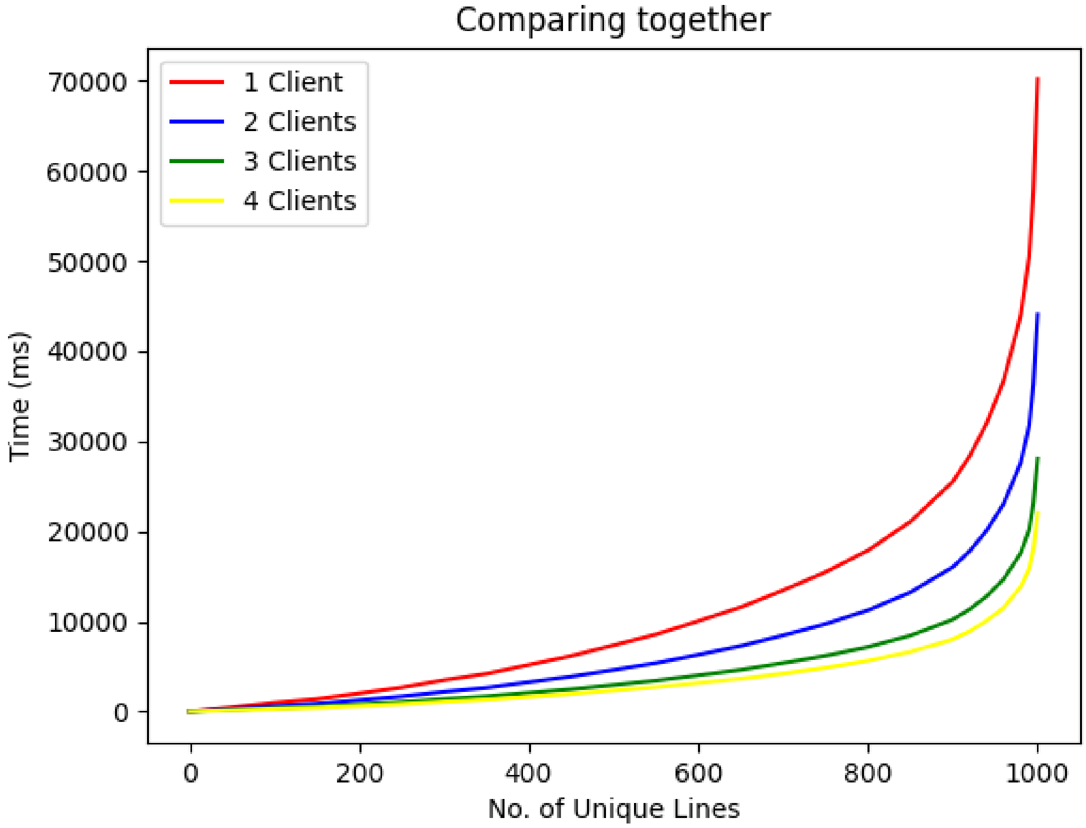
Unique Lines	0	100	200	300	400	500	600	700	800	900	960	995	1000
Time (ms)	0	390	803	1389	2071	2938	4014	5383	7154	10210	14696	23161	28083



- **Four Clients Downloading:**
  - Time Taken: 22059ms

Unique Lines	0	100	200	300	400	500	600	700	800	900	960	995	1000
Time (ms)	0	306	630	1091	1627	2307	3152	4228	5619	8020	11544	18193	22059





Analysis

From the test results, we observe that the download time reduces significantly but does not reduce linearly as more clients collaborate to download a file. This is due to various factors, including network congestion, client bandwidth limitations, and the availability of file-sharing peers. As more clients join the download, the overhead of coordinating the download may offset the benefits of additional bandwidth.

Test Environment

Client Hardware: 8 GB RAM with macOS processor
Network Conditions: IITD wifi network

Exception Handling

Client Disconnection and Reconnection:

Clients can gracefully disconnect and later reconnect. The system maintains the client's shared line index and continues normal operation upon reconnection. We have handled such cases using locks and mutexes, preventing parallel threads from writing into a cell where another thread is already writing or preparing to write.

Client-Server Connection Break:

If a client-server connection breaks, the client implements reconnection logic. The server keeps track of the last session where the client left off. When a client attempts to reconnect, the server checks if the IP address was encountered previously. If it finds an earlier session, it has a record of all the lines the client had in its last session.

Conclusion

In conclusion, we have implemented a distributed P2P file-sharing system that allows clients to share and download lines in a decentralized manner. We conducted tests to assess its performance in different scenarios and observed that download times do not necessarily reduce linearly as more clients collaborate. The system also handles various exception scenarios to ensure reliability.

This assignment has provided valuable insights into distributed systems and P2P networking, serving as a foundation for further exploration and improvement of P2P file-sharing algorithms.