Distributed File Download

Team: Doofinshmertz Evil Inc.

Anish Banerjee[*] Ankit Mondal[†] Piyush Chauhan[‡] Shankh Gupta[§]

August-September 2023

§1. Introduction

In this assignment, we implement a distributed file download system using a P2P connection with a Star Connection Topology(Master-Slave) between multiple clients. We have also done various analysis and comparisons with respect to time taken for downloading file, number of clients connected, etc. and reported our observations.

§2. Our Protocol

We have used a master-slave protocol to run this program. In this protocol, a central (master) server is connected to many different client (slave) computers. Any communication between clients is conducted through the master.

2.1. Client programs

In our design of the network, the client programs receive data from the server on vayu and send it to the master server. The client programs have 2 different sockets, one connected to vayu and the other with the master. Any data packet received from the vayu server is immediately passed on to the master server.

2.2. Master program

In our design of the network, the master programs receive data from the clients. It initializes 3 different connections with the client programs on 3 different threads. The master performs the computation intensive task of checking if the packet is broken or not and then waiting for multiple packets to arrive so as to get a complete line. The master program parses this by checking if the packet starts with a number or not, and if it does then how many new line characters are there in the packet. If there aren't 2 newline characters, it keeps receiving packets till it finds a newline character.

2.3. Master submission

Once all the lines in the database have been compiled, the master constructs the submission message and submits it to vayu.

2.4. Client submission

Once the master finishes its submission of the lines, it sends the submission message to the clients in one go. The client program receives the message and adds the necessary header code and submits the file to the vayu server.

[*] 2021CS10134

[†] 2021CS10229

[‡] 2021CS11010

[§] 2021CS50604

§3. Graphs

3.1. Running Individually

We ran the program on our individual systems and got the following results:

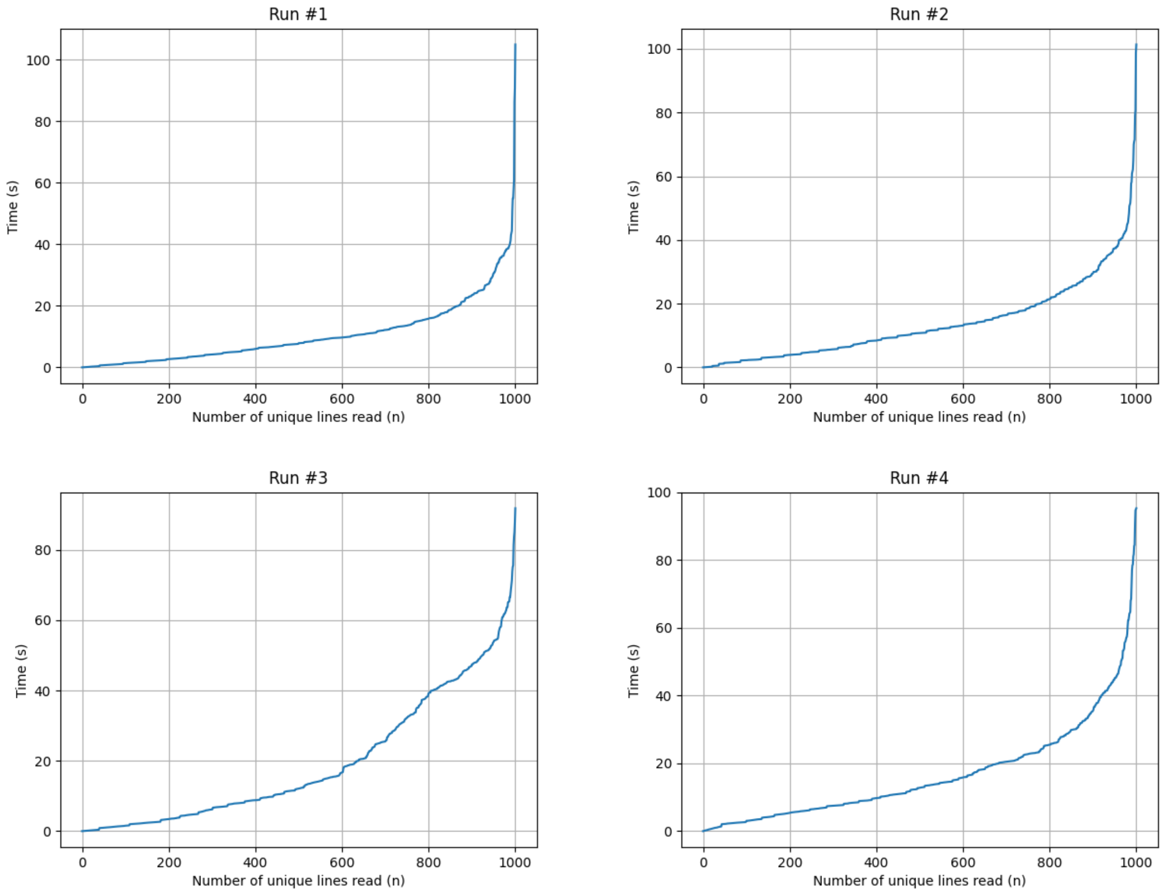

Figure 1: Runs in which a single client is connected to the server

The approximate total time taken for the above runs is as follows:

- **Run #1** 104.93s
- **Run #2** 101.37s
- **Run #3** 91.87s
- **Run #4** 95.31s

The average time is around 98.37s.

3.2. Two Combined

The approximate total time taken for the above runs are:

- **Run #1+2** 39.63s
- **Run #3+4** 51.39s

The average time is around 45.51s.

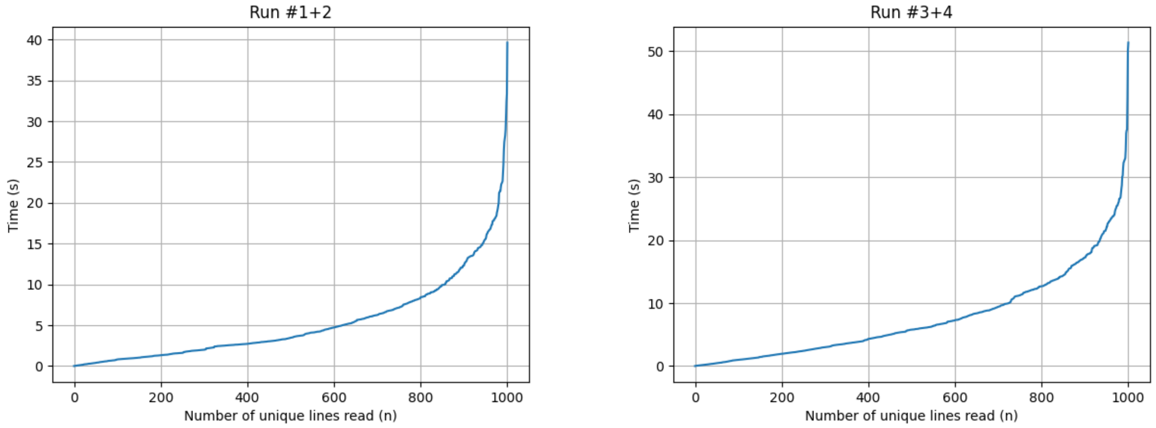

Figure 2: Runs with 2 clients connected to the server

3.3. Three Combined

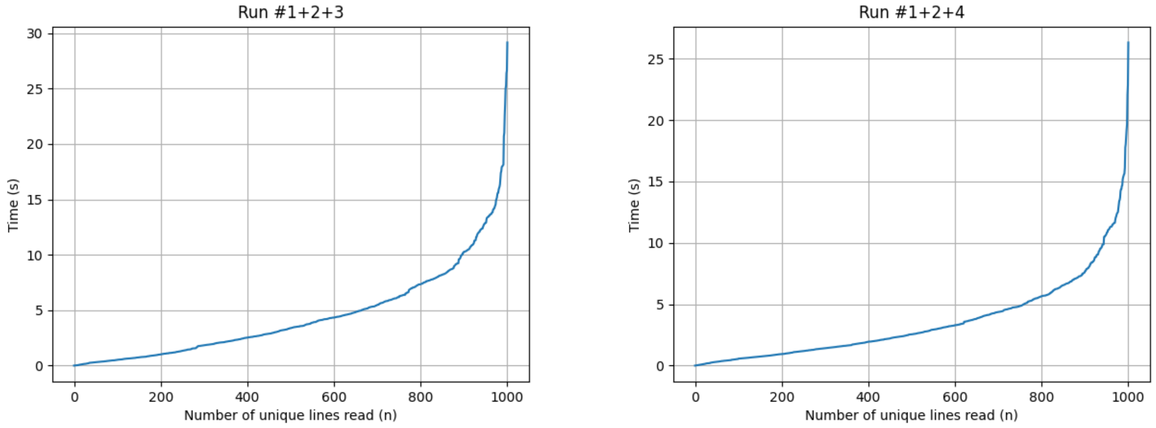

Figure 3: Runs with 3 clients connected to the server

The approximate total time taken for the runs are:

- **Run #1+2+3** 26.32s
- **Run #1+2+4** 29.15s

The average time is around 27.74s.

3.4. All 4

The approximate total time taken for the run is 20.28s.

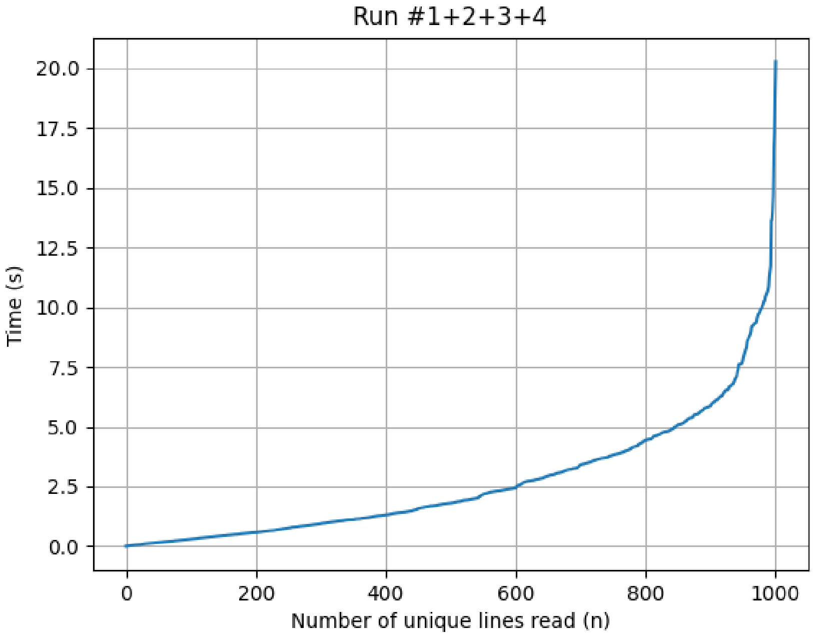

Figure 4: Runs with 3 clients connected to the server

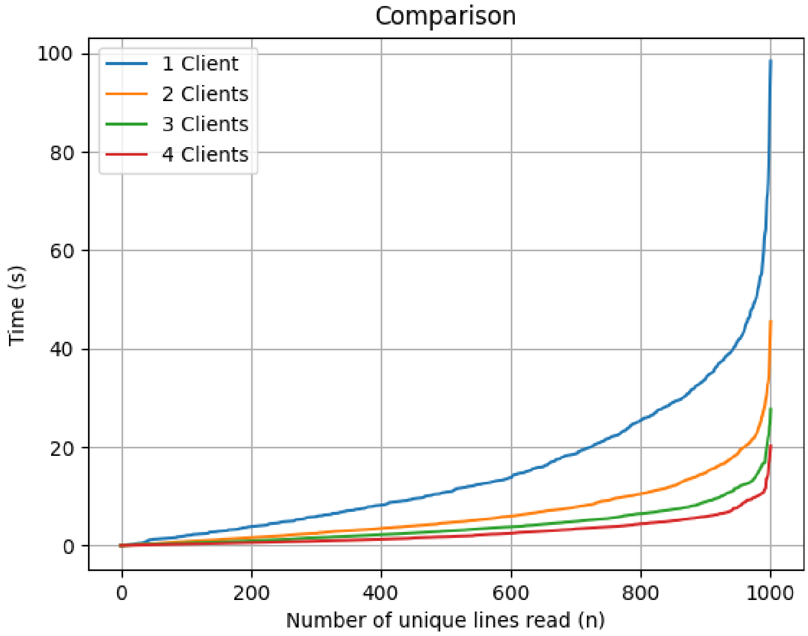

Figure 5: Comparison of rates of file download with increasing number of clients

3.5. Observations

1. **Rate of Receiving Lines from Vayu:**
   - Initially, the rate of receiving new lines from Vayu is rapid due to a high probability of obtaining unique lines.
   - Over time, the rate gradually decreases as the likelihood of receiving lines already in possession increases. This decrease in the rate of unique lines received is evident.
2. **Impact of the Number of Clients:**
   - As the number of clients increases, the total time required to receive the files decreases noticeably.
   - This improvement is due to the higher probability of acquiring unique lines from multiple clients compared to a single client.
3. **Time of file download with increasing clients:**
   - The reduction in time is not linear; it exhibits diminishing returns. This means that while adding more clients initially results in a significant time reduction, the subsequent improvements become less pronounced.

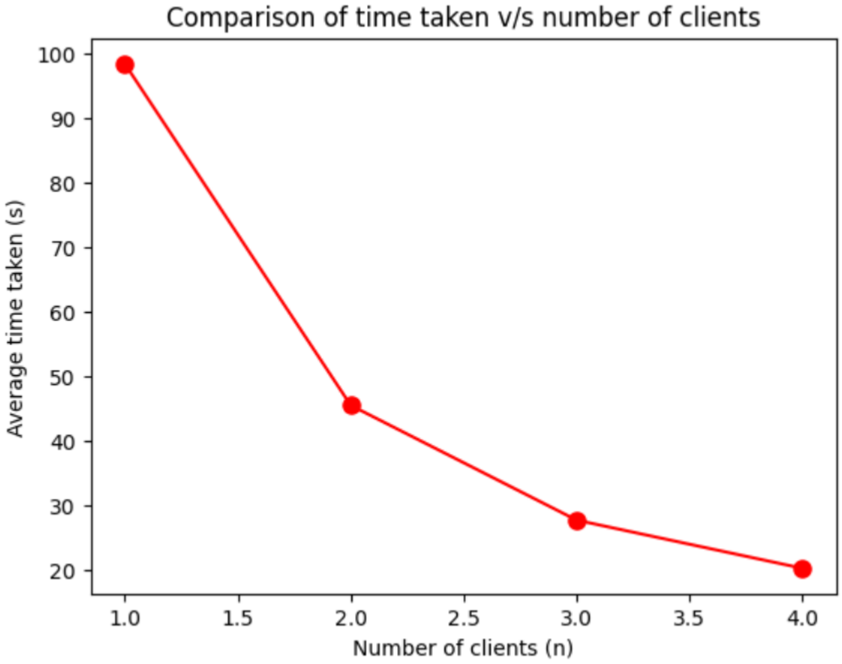

Figure 6: Comparison of time taken v/s number of clients

   - The average time decreases progressively, transitioning from 98.37 to 45.51, then to 27.74, and finally to 20.28. However, the initial halving of time is followed by smaller incremental decreases. It resembles a hyperbolic decrease as expected.
   - In a P2P connected distributed file download, the time taken to download a file is given by:

$$D_{P2P} > \max\{\frac{F}{u_s}, \frac{F}{d_{min}}, \frac{F}{u_s + \Sigma u_i}\}$$

     Here, the bottleneck will be $\frac{F}{u_s+\Sigma u_i}\}$ which decreases with the increase in number of clients. However, the time never goes to zero since it is limited by the server's upload speed and the clients' download speed, too.

§4. Exceptions Handled

We did the following exception handling :

- **Connection of Master with Vayu**: The master server's connection with Vayu, if broken, gets reestablished again. We handled this using a try-except block.
- **Connection of Slaves with Vayu**: Similar to the Master-Vayu connection, If the connection of the slave with the Vayu server gets broken, it gets reestablished using a try-except block.
- **Slave connection with Master**: If the slave loses connection with the master, it closes the master connection, creates its database, saves the data received from the Vayu server, and eventually submits as soon as it receives all the data. So even if the slave node loses connection with the master server, it will still be able to submit successfully.