



---

## COL 334 : COMPUTER NETWORKS

---

### ASSIGNMENT 2 REPORT

**Naman Nirwan**

**Entry No: 2021CS50593**

**Rishit Singla**

**Entry No: 2021CS10547**

**Kushal Srova**

**Entry No: 2021CS10114**

**Achyut Salunkhe**

**Entry No: 2021CS10121**

**Indian Institute of Technology**

**Delhi**

## Contents

<b>1</b>	<b>Description</b>	<b>1</b>
1.1	Algorithm Used . . . . .	1
<b>2</b>	<b>Single Client</b>	<b>2</b>
2.1	Graphs of performance . . . . .	2
2.2	Deductions . . . . .	2
<b>3</b>	<b>Two Clients at a Time</b>	<b>3</b>
3.1	Graphs of performance . . . . .	3
3.2	Deductions . . . . .	3
<b>4</b>	<b>Three Clients at a Time</b>	<b>4</b>
4.1	Graphs of performance . . . . .	4
4.2	Deductions . . . . .	4
<b>5</b>	<b>Fours Clients at a Time</b>	<b>5</b>
5.1	Graphs of performance . . . . .	5
5.2	Deductions . . . . .	5
<b>6</b>	<b>Choosing Main Client</b>	<b>6</b>
<b>7</b>	<b>Exceptions Handled</b>	<b>8</b>

## 1 Description

### 1.1 Algorithm Used

#### **Brief Description**

In our Algorithm we assign the title "main client" to one of the client. Rest of the clients are called "dummy clients". Each of the client will be connected to all other clients as well as server and information will be retrieved from server as well as shared to all other clients if required.

#### **Detailed Description**

First the program for main client is executed in which total number of dummy clients is given as an argument. This is to ensure that main client do not connect to more than that given number of dummy clients for obvious security reason. Main client will print its IP which needs to be given as argument to all the dummy clients during their execution. The program on all dummy clients will be executed after receiving the IP from main client. Each of the dummy client will send connection request to main client followed by sharing its IP. Main client will store these IPs and send them to all of the dummy clients afterwards. The main client then orders all the dummy clients to connect to each other. Once a dummy client is connected to all other dummy client it will send its confirmation to the main client. On receiving all the confirmation, main client will ask all of the dummy clients to connect to web server and then it will connect itself to the web server as well. Thus the role of main client is to coordinate all of the connections among the peers. After this all the dummy clients and main client are on same footing and have equal role. (Due to this reason, We have avoided master slave nomenclature, as after all of the P2P connection are established main client and dummy clients are same and hence there is no master or slave). Each of the client requests a line from web server repeatedly and stores it. If the line was already present with the client then it will ignore it otherwise it will send the line to all other dummy clients. Sending lines to dummy clients, Receiving lines from dummy clients and receiving lines from web server, all of this is done in different threads.

## 2 Single Client

### 2.1 Graphs of performance

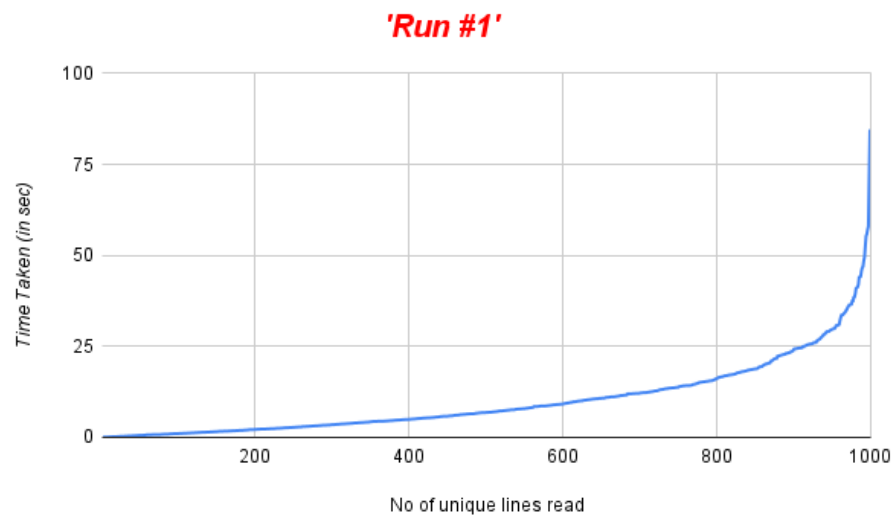


Figure 1: Graph of 1 Client Downloading the file

### 2.2 Deductions

- From the above graph the 1 client downloading of the file takes on an average about 80 sec.
- In the starting of the process the rate of getting a unique line is much higher than at the end of process.
- In the starting of the process the graph tends to be nearly straight line but as the graph pass around 800 lines the time start increasing exponentially.

### 3 Two Clients at a Time

#### 3.1 Graphs of performance

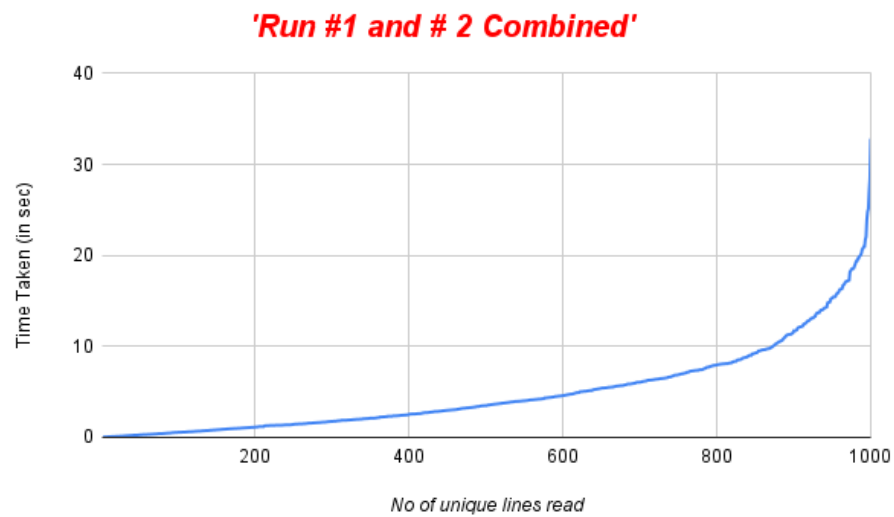


Figure 2: Graph of 2 Clients Downloading the file

#### 3.2 Deductions

- As the No of clients are doubled the average time to download the file has fallen to half of the time taken by 1 client i.e about 35-40 secs.
- The reason behind this is multi threading and cooperation of both the clients to download the file and sharing the data with each other, which nearly caused the rate of downloading the complete file to be doubled.
- Other observations are same as the 1 client download.

## 4 Three Clients at a Time

### 4.1 Graphs of performance

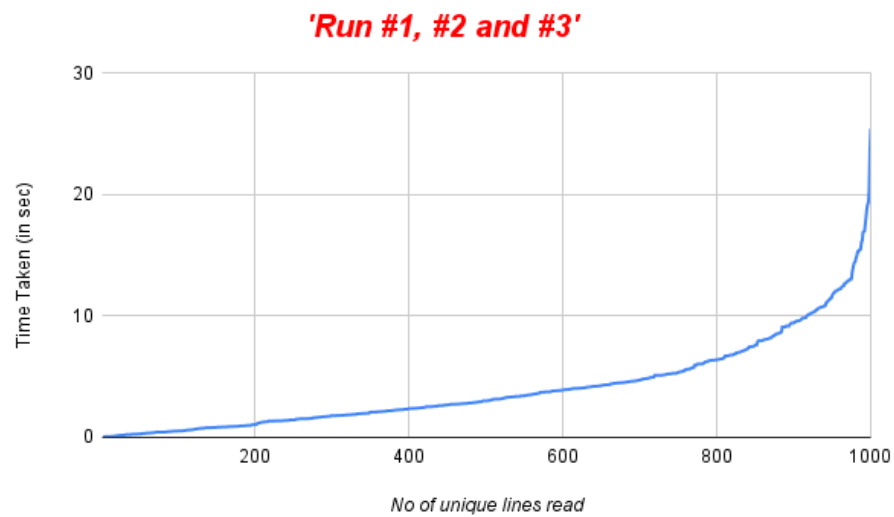


Figure 3: Graph of 3 Clients Downloading the file

### 4.2 Deductions

- As the No of clients are tripled the average time to download the file has fallen to nearly 1/3rd of the time taken by 1 client i.e about 25-30 secs.
- The reason behind this is multi threading and cooperation of all the three clients to download the file and sharing the data with each other, which nearly caused the rate of downloading the complete file to be tripled.
- Other observations are same as the 1 client download.

## 5 Fours Clients at a Time

### 5.1 Graphs of performance



Figure 4: Graph of 4 Clients Downloading the file

### 5.2 Deductions

- As the No of clients are tripled the average time to download the file has fallen to nearly 1/4th of the time taken by 1 client i.e about 15-20 secs.
- The reason behind this is multi threading and cooperation of all the four clients to download the file and sharing the data with each other, which nearly caused the rate of downloading the complete file to be quadrupled.
- Other observations are same as the 1 client download.



## 6 Choosing Main Client

We observed from the data collected that the total time taken to fetch 1000 lines from the Vayu server reduces linearly with the increase in the number of clients. It doesn't matter who is the Main client and who is the Dummy client, the total time remains nearly the same. Thus we can assign main client to any of the given PC without affecting the performance. Main client just has an additional function of connecting other PCs with it. Choosing the main client will not affect the overall performance. We observed this by running our program on all possible combinations of 4 PCs with assigning one of them as main client. The data for this is tabulated below.

Table 1: Data table for running the code on all combinations of 4 PCs to analyze the impact of assigning different PC as main client

Main Client	Child Client	Total Time (s)
1		84.35001135
2		73.54765675
3		78.88581109
4		68.80625463
1	2	32.84338474
1	3	35.50513721
1	4	38.32211137
2	1	36.92387247
2	3	40.70759296
2	4	35.18981957
3	1	36.00186086
3	2	42.88862371
3	4	39.01398277
4	1	34.87714934
4	2	32.93068051
4	3	37.43570971
1	2,3	25.43952107
1	2,4	25.9069531
1	3,4	23.57403994
2	1,3	41.20325756
2	1,4	25.6853261
2	3,4	21.08107948
3	1,2	32.90328193
3	1,4	28.52743578
3	2,4	28.69749928
4	1,2	19.43346977
4	1,3	24.04976416
4	2,3	21.85006022
1	2,3,4	20.73804498
2	1,3,4	22.43796515
3	1,2,4	24.5666523
4	1,2,3	19.06764698

## 7 Exceptions Handled

Following exceptions were handled in the code for client disconnections

- If any of the client disconnects from the vayu server then it will not stop receiving lines from other clients and also it will try to reconnect to server every 1 sec until all of the lines are received or connection is established.
- If any of the client (say client A) disconnects from any other client (say client B) during the time of execution of program the client A will try to reconnect to B when it needs to send some information to it. Similarly B will try to reconnect to A when it needs to send some information to it. In case of connection refusal each of them will try to reconnect after every 1 second until all of the lines are received or connection is established. If any of the client does not try to reconnect then the rest of network will not be affected and will function normally.