Distributed Downloading System

B18CSE051 - Shah Parshv Jigneshkumar
B18CSE061 - Vinay Detani
B18CSE062 - Vivek Khokar

Abstract

Working online with a slow internet speed is frustrating. The frustration reaches its peak especially when there is an upper limit on maximum download speed and you need to download a big software. Even though we have additional resources like computers or additional internet connections, they still remain unused. The objective of our project is to come up with a platform where users can download large files in less time. The approach behind this idea is to divide the file into several partitions and download each partition individually. Once all the partitions are downloaded, they are merged back to make a whole new complete file. The project comprises two files, server and client. A server is carried out on Node that accepts a request from clients. This request contains the name of the file to be accessed by the user and the amount of partitions to be downloaded into the file. Client allows the user to make an initial download request. As mentioned earlier, this request has the name and a number in which the file will be downloaded. Hereby, our software reduces the downloading time of a file.

Keywords :- Download, Partitioning, Merging, Client, Server.

1. Introduction

In Computer Networks, downloading means to receive data from a remote system, which is usually a server, such as database server, FTP server, E-Mail server or any other similar servers. A download is basically a file that has been downloaded or has been offered to be downloaded or the method of receiving such a file. Each and every file will be stored on different servers on the internet, where each server has a particular IP address. These IP addresses are represented by a URL-called domain name. For example, the URL is www.google.com. It is possible to find each file using a special URL called the download URL. In our project, instead of URL, we will use a file name which is stored in a shared folder at the client side. So, we need to enter the file name in order to download a file. We use a partitioning and merging concept in our project. Consider the analogy of a single work that needs to be completed as

soon as possible. If only one person works on that project, it will take more time in contrast to if we divide the work further into N task and assign these tasks to N number of people. At the end, when all the N tasks are completed, we will merge them together and our work will be completed in less time. Our project also partitions a particular file and downloads them individually and then merge operation is carried out once all the partitions have been downloaded. The time needed to complete the process depends on the number of connections involved in the download process. To define a single connection on a network, an IP address is used. So the IP address and the port number have been provided for the communication between the users. The aspect of this software is to utilize the idle resources which save time during the process. To apply the distributing concept to reduce the time being taken in the normal downloading software and to utilize the idle computers and idle internet connections. Files (that is needed to be downloaded) must be downloaded concurrently on more than one device via more than one Internet link and must be returned to the computer where the download was started. Finally, in the initiated machine, the merge process has to be done to create the actual file.

2. Methodology

Transmission Control Protocol (TCP) : The Transmission Control Protocol (TCP) is the Internet standard that ensures that data packets are efficiently transmitted over a network between users. TCP is the underlying communication protocol for a wide variety of applications, including web servers and websites, email applications, FTP and peer-to-peer apps. TCP operates with the internet protocol (IP) to specify how data is exchanged online.IP is responsible for transmitting each packet to its destination, while TCP ensures the bytes are sent without errors or omissions in the order in which they have been received. The two protocols, combined, are referred to as TCP/IP.

Establishing TCP handshaking connection:

Establishing a TCP connection involves involvement in what is known as a three-way handshake from both the client and the server.The process can be broken down as follows:

1. A client sends the server a SYN packet—a connection request from its source port to a server's destination port.

2. The server replies with a SYN/ACK packet, confirming that the link request has been received.
3. The SYN/ACK packet is received by the client and responds with an ACK packet of its own.

TCP operates by breaking down transmitted data into segments after the connection is formed, each of which is packed into a datagram and sent to its destination.

Partitioning Process: This software partitions the files at the time of downloading into parts of equal size. The client can also give a custom number specifying the number of parts he/she wants. The total file size received from the internet. The beginning and finishing value of the part size for every part has to be determined by the machine.

Downloading Process: It connects to the server by TCP request with the aid of local server and the client information and gets the answer using TCP Response. Two streams are created for the file to read and write. The read operation happens on the server, while the write operation happens on the local storage at client side. Buffer is used to transport the bytes of values from the server to local computer through the stream.

Merging Process: After the download method, this module is invoked. Download module passes the various file locations to this merge module. It gets all the parts from the computer. After all parts are gathered in the local computer, the merge operation starts. With the support of the stream all parts are merged into a single file.

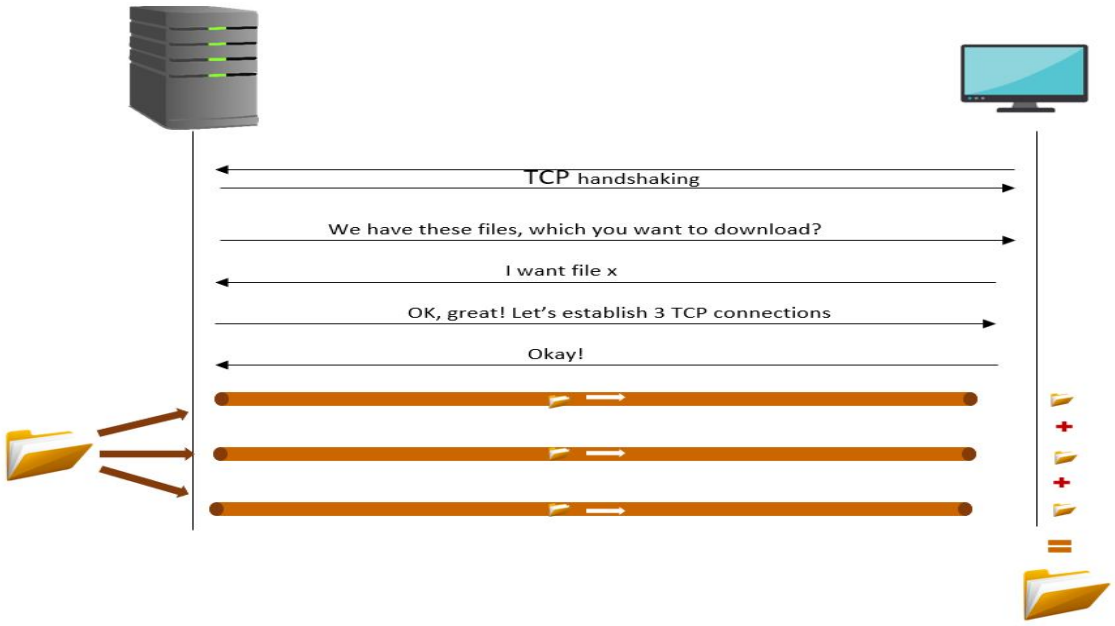

Fig 1 :- Workflow of our implementation.

Instead of having to download only one file, the files are partitioned into numbers of sections and those parts are accessed in parallel on the allotted machine. This decreases the time for downloading the file as it uses the idle resources in the machine. First, the computer which starts the process of downloading establishes a TCP connection via TCP handshake with the server. After that it chooses the download file, storage location and number of partitions and sends them to the server. Then, with its internet connection, it does the downloading process. The requested file gets partitioned into a number of parts and the computer does the task of downloading each part separately and parallely. The computer may end its link to the Internet after this process. After acquiring or downloading all the parts from the main server, the merger process begins. The merge process merges each and every part into a single file, which is the file that we had originally asked for.

```
PS C:\Users\Vinay\Desktop\CN> python c1.py
[+] Connected.
0 . 1.pdf   24911KB
1 . asw.gif   999KB
2 . Limitless.S01E08.720p.Hindi.WEB-DL._(_Themoviesflix.co_).mkv.mp4   301857KB
3 . Social-Emotional Learning by Slidesgo.pptx   46013KB
choose the file you want to download
```

Fig 2 :- Working, Selecting the file to download.

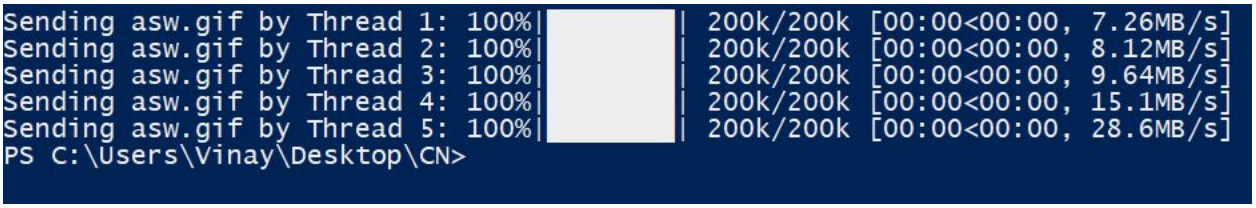

Fig 3 :-Working, Parts getting downloaded.

3. Results and Analysis

We tried to send different types of files like .mp4 file, .txt file, .png file, .pdf file etc. We were able to transfer all these files successfully. Here are some results that we analysed by choosing different number of connections:
The results are not satisfactory since we are using only one device for the server and client as well. Ideally the downloading speed should be directly proportional to number of connections. But the results we got are not exactly proportional to the number of connections. The reason might be the propagation delay, which is absent due to the same machine for client and server. Here, for the .mp4 file, we chose a different number of connections and got the following results. Processing delay is increasing by the number of connections. Ideally for 5 connections, time taken to download should be 5 times lesser than the single connection. For larger connections processing delay is very significant that the overall download time is even more than for a single connection. So these connections can be useful for larger files where this delay will be neglected.

Data with different number of connections:

File Name	File Size(KB)	Download Time(Sec)	Average Downloading Speed(MB/sec)	Connections
Limitless.mp4	301857	3.675803	82.12	1
Limitless.mp4	301857	3.646225	82.78	2
Limitless.mp4	301857	4.041485	74.68	5
1.pdf	24911	0.254160	98.01	1
1.pdf	24911	0.318865	78.12	2

1.pdf	24911	0.327964	75.96	5

The graph shown below is basically the mapping of time and number of connections for the particular file. Linear graph with decreasing slope in ideal. But due to reasons stated above we got this graph.

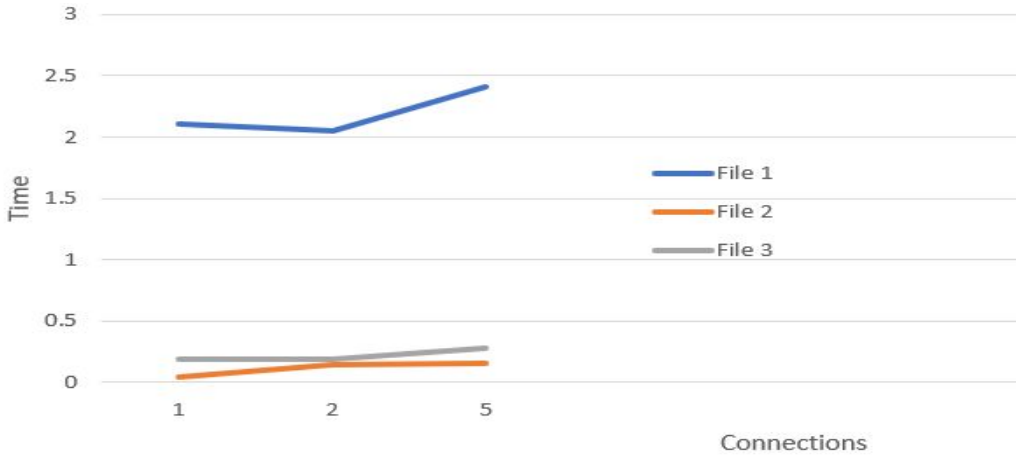

Fig 4:- Mapping of Connections and Time for files

4. Conclusion

Thus our project enables users to involve in the process of downloading using idle resources which ideally reduces the time taken for downloading in other software. However, in practice we did not find this to be agreeable. One possible reason can be because the download speed was not proportional to the number of cases, which ideally should be directly proportional. Absence of propagation delay might be the reason, as we are using client and server on the same machine.

5. References

We have referred to the following websites whenever there are struggles while proceeding.

[1] www.stackoverflow.com - Stack OverFlow
[2] www.geeksforgeeks.org - GeeksforGeeks
[3] www.wikipedia.org - Wikipedia