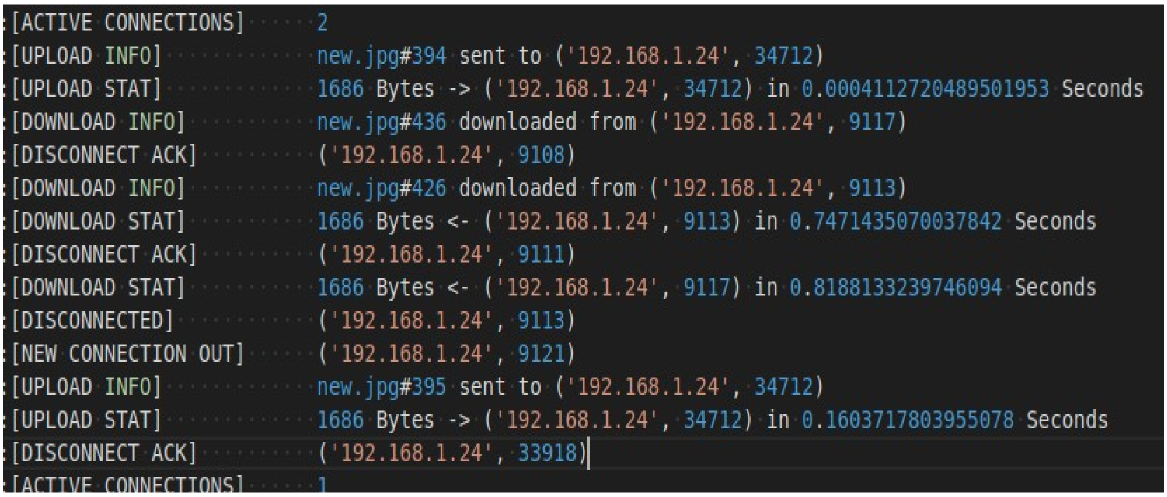
Programming Assignment 4
CS550 - Advanced OS
Fall 2020
Amit Nikam
A20470263
anikam@hawk.iit.edu

EVALUATION REPORT

For this evaluation I used an image of size 724,953 bytes. Each chunk of data file was of the size 1536 bytes. In total there were 472 chunks of data that made up the file.

Since the size of log files(output) was exceeding 190MB size, the files have been stored in the following drive location: https://drive.google.com/drive/folders/1oZJyyOGdrvQbmoh2uI3DUImJr0h-0E8I?usp=sharing

While scaled at 128 nodes, looking at the node activity as follows:



```
:[ACTIVE CONNECTIONS]     2
:[UPLOAD INFO]            new.jpg#394 sent to ('192.168.1.24', 34712)
:[UPLOAD STAT]            1686 Bytes -> ('192.168.1.24', 34712) in 0.00041127204895O1953 Seconds
:[DOWNLOAD INFO]          new.jpg#436 downloaded from ('192.168.1.24', 9117)
:[DISCONNECT ACK]         ('192.168.1.24', 9108)
:[DOWNLOAD INFO]          new.jpg#426 downloaded from ('192.168.1.24', 9113)
:[DOWNLOAD STAT]          1686 Bytes <- ('192.168.1.24', 9113) in 0.7471435070037842 Seconds
:[DISCONNECT ACK]         ('192.168.1.24', 9111)
:[DOWNLOAD STAT]          1686 Bytes <- ('192.168.1.24', 9117) in 0.8188133239746094 Seconds
:[DISCONNECTED]           ('192.168.1.24', 9113)
:[NEW CONNECTION OUT]     ('192.168.1.24', 9121)
:[UPLOAD INFO]            new.jpg#395 sent to ('192.168.1.24', 34712)
:[UPLOAD STAT]            1686 Bytes -> ('192.168.1.24', 34712) in 0.1603717803955078 Seconds
:[DISCONNECT ACK]         ('192.168.1.24', 33918)
:[ACTIVE CONNECTIONS]     1
```

It can be observed that the byte size with the chunks wrapped in the payload along with the meta data is actually 1686 bytes as per ‘UPLOAD STAT’ and ‘DOWNLOAD STAT’.

Furthermore it is observed that the general download time for a chunk is ~0.780seconds. This is due to the way the node is programmed, the buffer tends to go to sleep for 0.5seconds after requesting the chunk from another node, shortly after checking again, if the data is available to be read then it continues, else it sleeps for same period again. Thus download times are always going to be higher than upload times. Upload times tend to vary a lot depending on the load on the node. Usually the time to upload is ~0.001 seconds.

In general, compared to the previous assignment, although the speed of downloading a file has increased significantly due to parallel download, the bandwidth usage has also gone up. Thus is due to the nodes exchanging meta data, and also since the additional over head cost of each chunk is also responsible for this increased use of bandwidth.

As seen in log image above, the node downloads image file chunks from different sources, at the same time, it is uploading some of the data it holds. Node 1 usually has the highest uploads due to the window rounding around in the round robin implementation. This is quite observable as all other nodes have similar bandwidths than Node 1.

It was observed that the download times were bad for node scales 80 and higher. This was due to my system's CPU(i5-5200U) working at maximum ~ limited computing capacity of my system.

Conclusion: In the end, the desired goal of implemented a P2P architecture based file service was implemented successfully. Parallel access of files at the chunk level significantly improved the download times, although it also increased the bandwidth due to the overhead involved.

Due to the breaking down of a file at chunk level and the overabundance of the log data due to parallel downloads and threaded nodes, it became difficult to keep track of sequence of chunks to plot a gantt chart.

Although the nodes functioned as expected, and reported the bandwidth statistics during upload and download as well as at the end of the download completion. This report is based on those very readings.