

I ran my experiments on my own laptop using a wired connection on the BYU campus. I am using VirtualBox to run a Linux VM. I ran the experiments.py script to collect the data for the small, medium, and large files. I then ran the plot.py script to create the 3 plots shown below. I used the files located on the Dropbox server.

Figure 1: Small file data

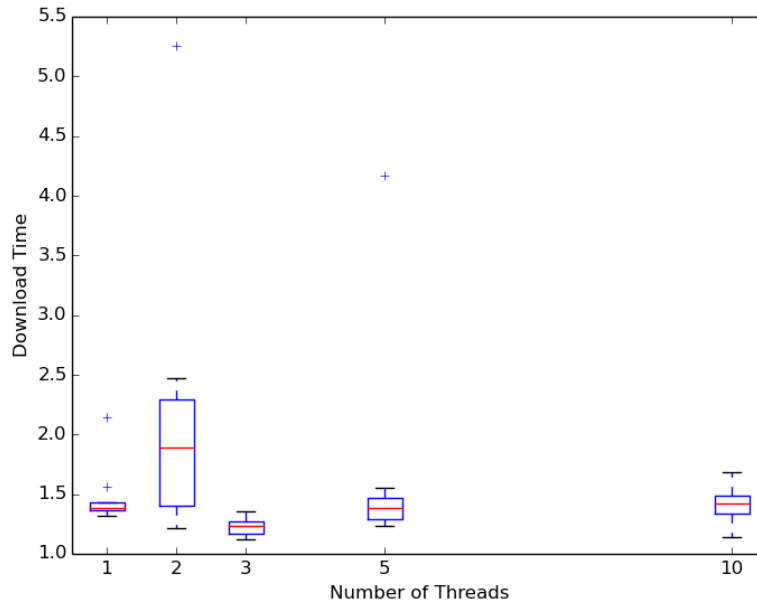


Figure 2: Medium file data

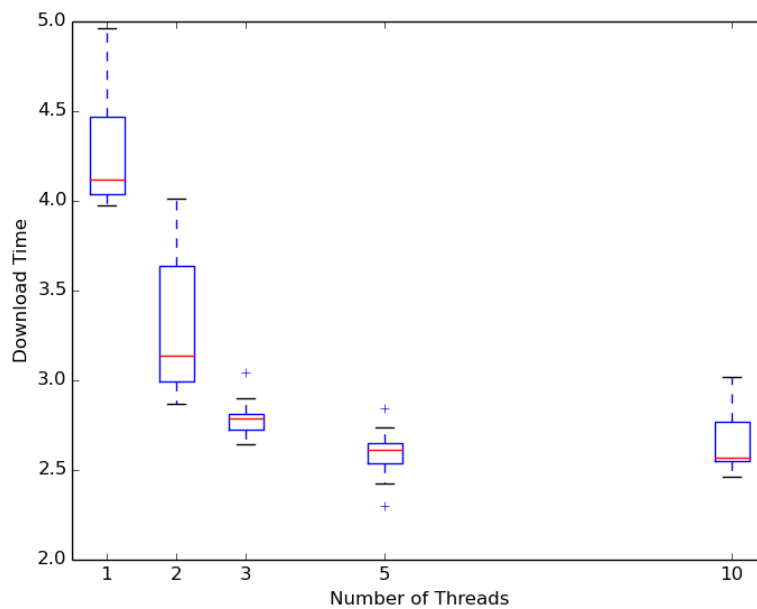
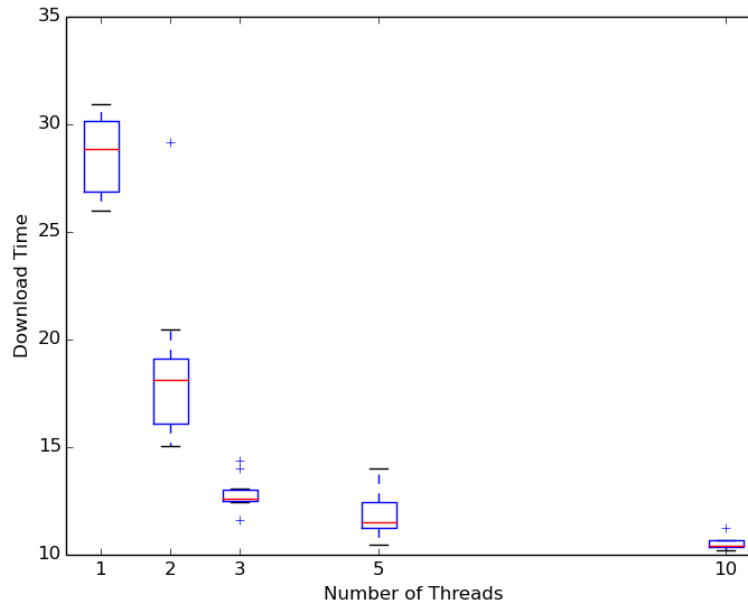


Figure 3: Large file data



For the small file we don't see too much difference when we increase the number of threads. In figure 1 we can see that it says pretty constant except for 2 threads where we see some spikes. This could have been a result of the server getting more users at that time. The lack of change could be because the file is so small that when you break up a small file you aren't really reducing the size of each segment because you are adding overhead in breaking it apart sending each part separately and reassembling it after. When the files get bigger we notice a difference with the number of threads because each chunk is bigger so the overhead isn't as noticeable.

In figure 2 we see that the medium file took less time as we increased the threads. The only anomaly was at 10 threads where it took a longer than 5 threads on average. The server could have had increased traffic at this time but overall it was still pretty short a time.

In figure 3 we see the steepest curve which means as we increased the threads the performance was increased significantly. It is especially interesting to see how fast the file downloaded using 10 threads compared to downloading the file using only 1 thread. This is because each thread is downloading a portion of the file at the same time so it is the amount of time to download a small file and we just do that for each thread simultaneously.

Overall it seems that for small files there really isn't much benefit but as the files get larger the benefits do become noticeable. Some of the tradeoffs could be that the server doesn't like getting so many connections for the same file. It could prematurely cut some of the threads off. This could cause problems when you are reassembling the data. Also running a large amount of threads could use a lot of local computer resources compared to just one thread downloading the file.