Jon George
CS360
Dr. Zappala

Download Accelerator

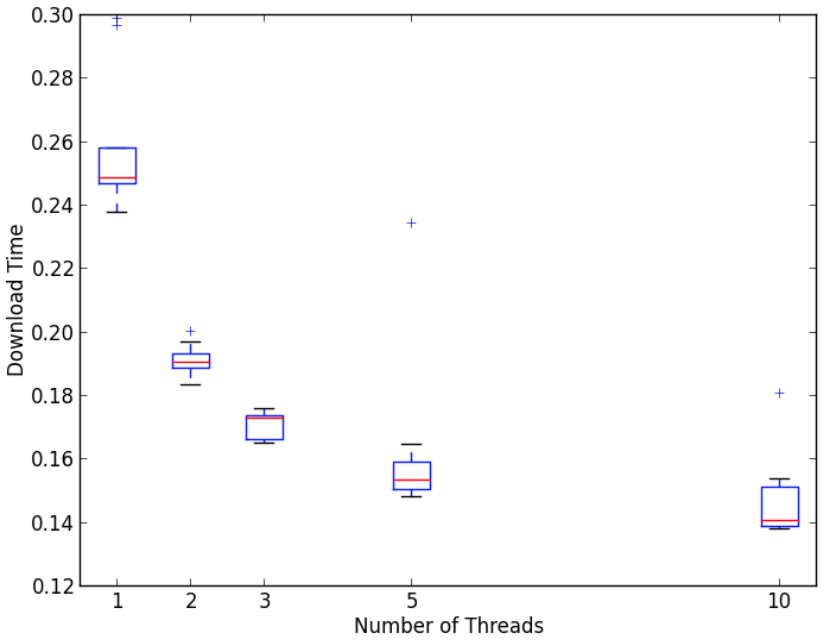
Methodology:

For this experiment I used the experiment.py file provided on the CS360 website. This provided 10 samples of 5 different thread executions for small, medium and large file sizes for a total of 150 samples. I downloaded the "http://www2.census.gov/" group of urls for my tests provided in the experiement.py file. I gathered my data at home on wifi using the google fiber paid plan.

Results:

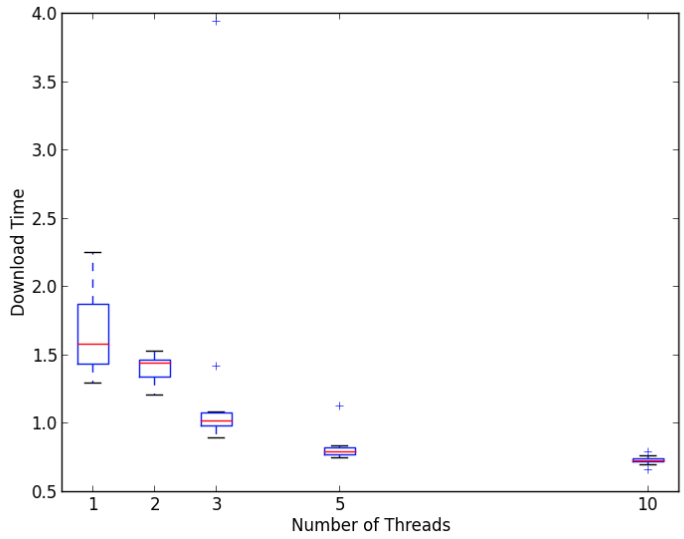
Overall I saw pretty expected results. As the number of threads increased, the download speed also increased. However, once the number of threads surpassed the number of cores on my processor the advantage of increasing threads diminished. This pattern is clear in each of my graphs below.

Small data - 1 MB file - tl_2013_10_tract.zip



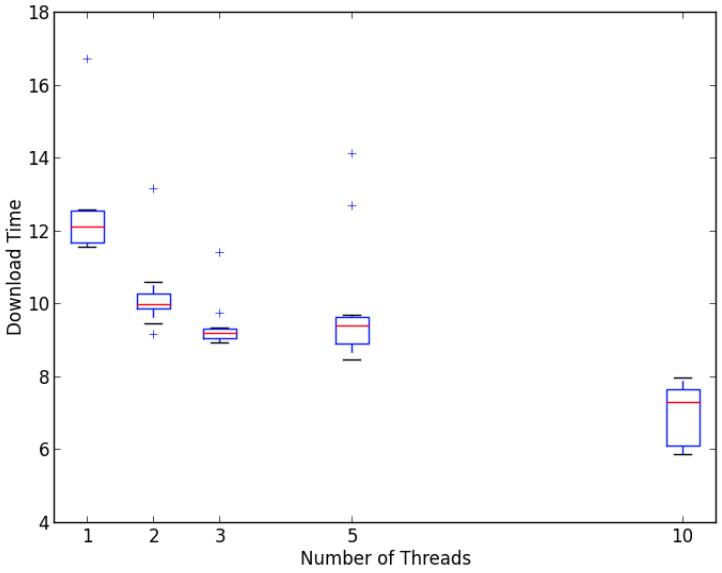
The data above shows that with a single thread, it took approximately 0.25 seconds to download 1 MB file or 4 MB/s. With ten threads, however, it only takes 0.14 seconds to download or a decrease of 44%.

Medium Data - 10 MB - tl_2013_35005_edges.zip



The medium set is similar to the small set, however it is interesting to note that the biggest decrease in download time from 2 threads to 3 threads. Also the overall difference between 5 and 10 treads is minimal.

Large Data - 100 MB - tlgdb_2013_a_39_oh.gdb.zip



The large data had interesting results because it did not decrease between 3 and 5 threads and had a significant drop between 5 and 10 threads.

Conclusion

Overall, I got the expected pattern of increasing threads causing decreased download time. However, it was interesting that in all data set I got several outlier with much high download times than average. As well, it seemed inconsistent to which number of threads would drop the download speed by the most percentage. More experiments would be necessary to show whether these differences are caused by the size of the file downloaded or by connection variables.