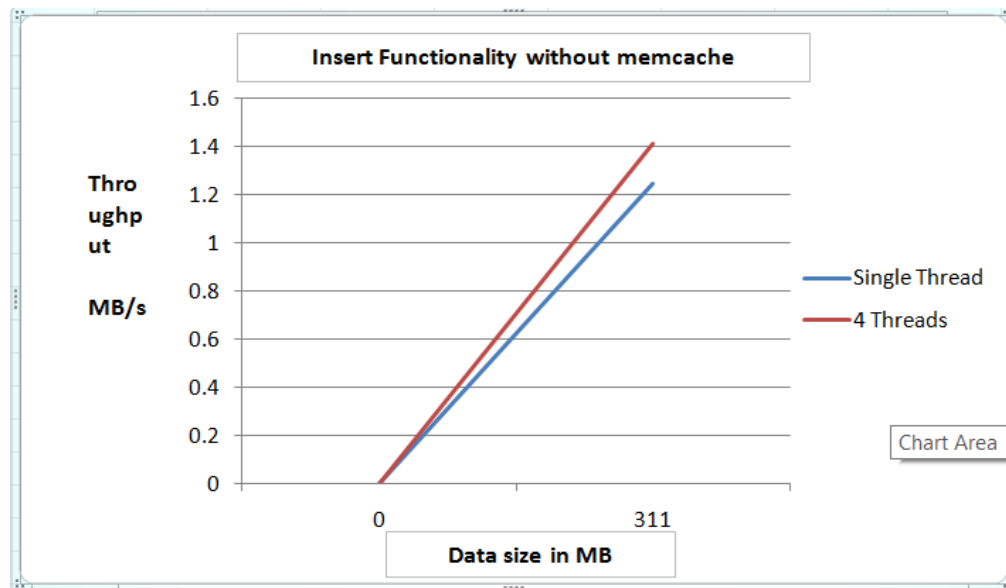


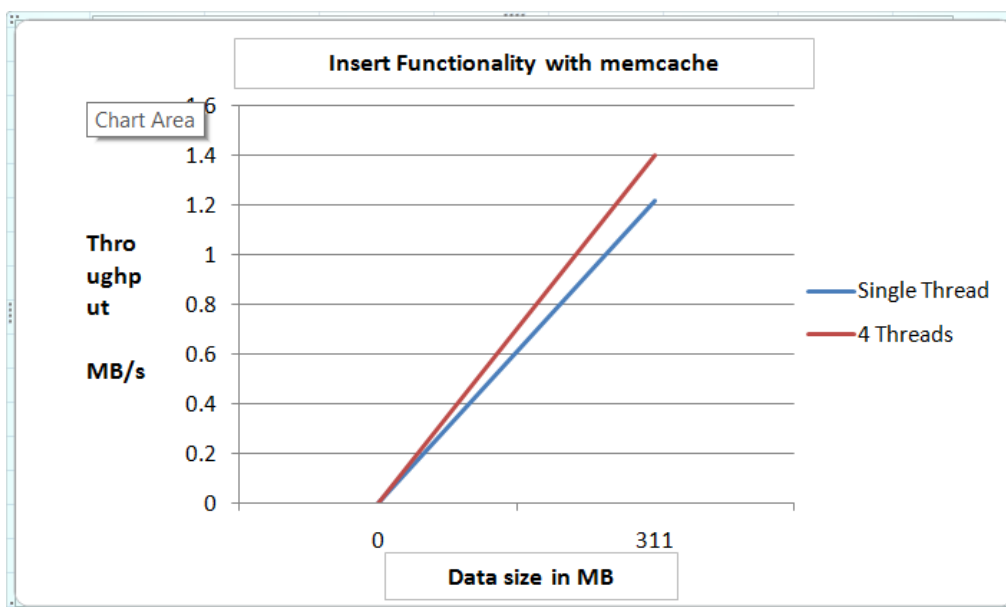
Performance Report



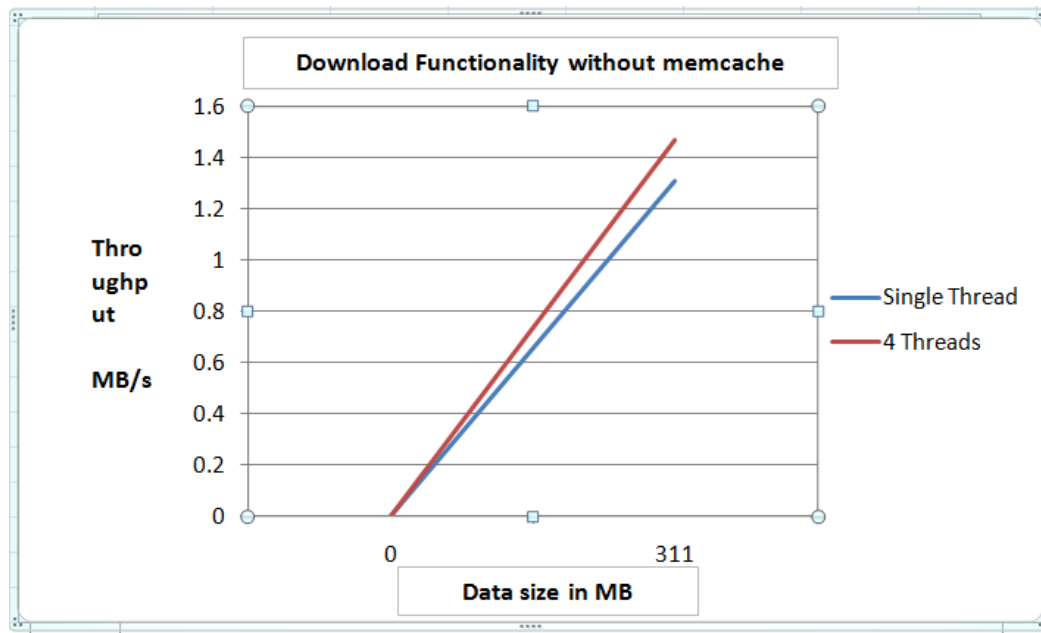
The figure above shows the overall throughput for inserting 411 files of the size 311 MB without memcache. Here we plot the 0-0 axis to obtain a linear line graph, to depict the results of the divergence between the thread variations. **The Y axis denotes the *throughput*.**

Throughput = Total Data transferred/Time taken in minutes

As we inference from the figure, usage of threading has increased the throughput by a considerable extent. This experiment has been recorded *without memcache*.



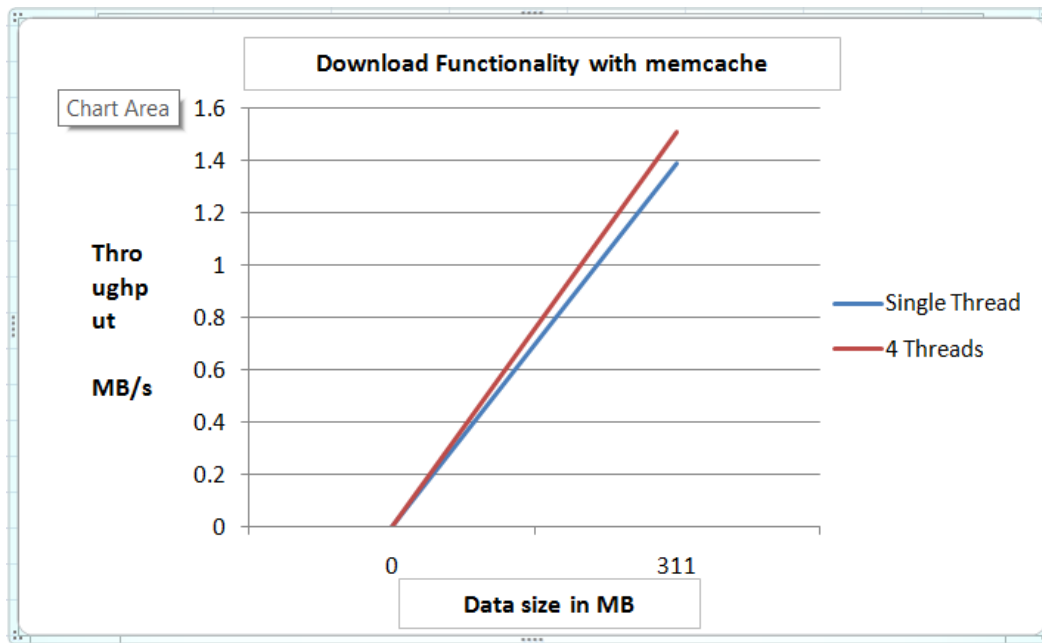
The figure above shows the overall throughput for inserting 411 files of the size 311 MB with memcache. The throughput for inserting files is reduced by a very small extent when we use memcache, because about a non negligible percent of the total data files will be stored in cache as well as in the storage. Again as expected, when we scale the threads, throughput is increasing marginally.



The figure above shows the overall throughput for downloading 822 files of the size 622 MB without memcache.

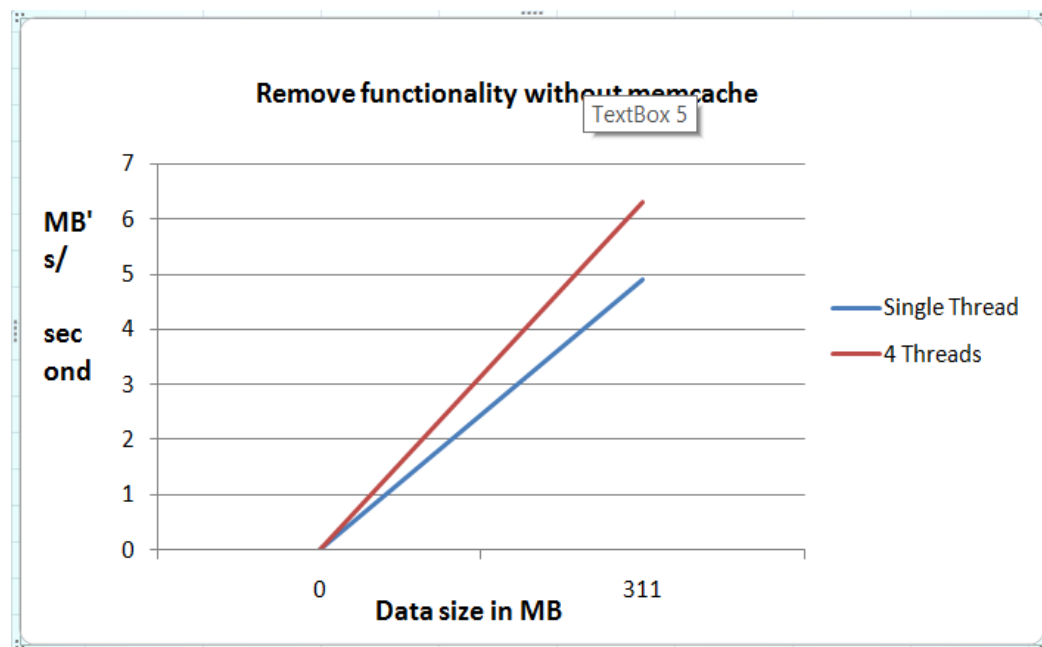
Throughput = Total Data transferred/Time taken in minutes

This experiment is for downloading 411 files (two instances of each) from the cloud storage. As we see in the depiction, threading has increased the throughput. Interestingly when we compare the results of this with the *insert* experiment, the throughput is comparatively high. Reason being that my internet's download bandwidth is higher than the upload provision. We would like to mention that when we tried to download all 822 files in a single go, our server hanged at times, despite that we were able to record the values in some instances.



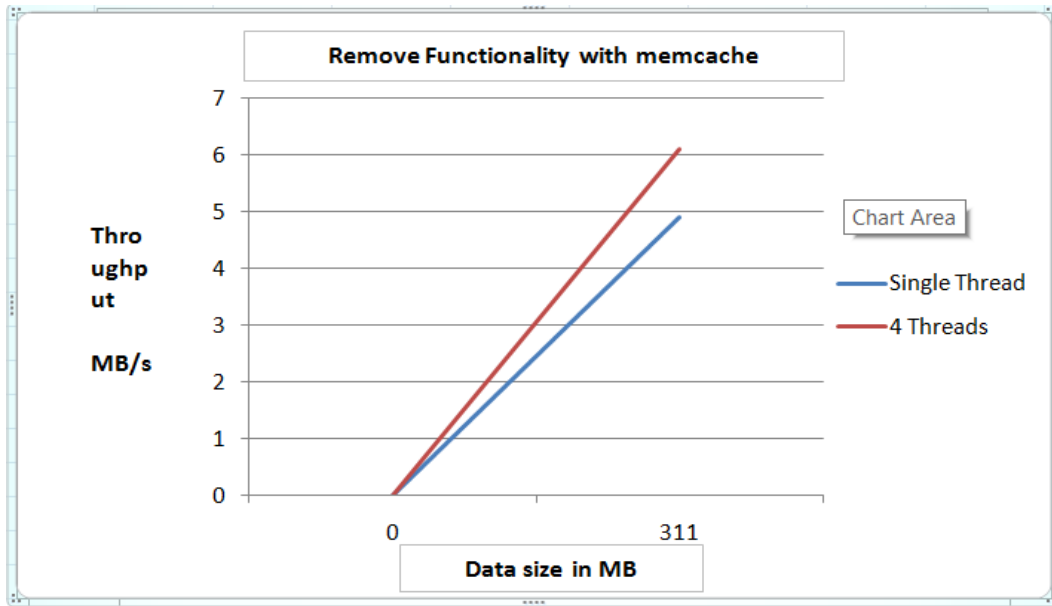
The figure above visualizes the download functionality for 822 files of size 622 MB with caching.

The results interestingly showed that caching has improved download throughput by a small extent. There is some considerable amount of data in cache. The results suggest that downloading data from cache is faster than the static one. Again threading has improved the efficiency of the experiment.

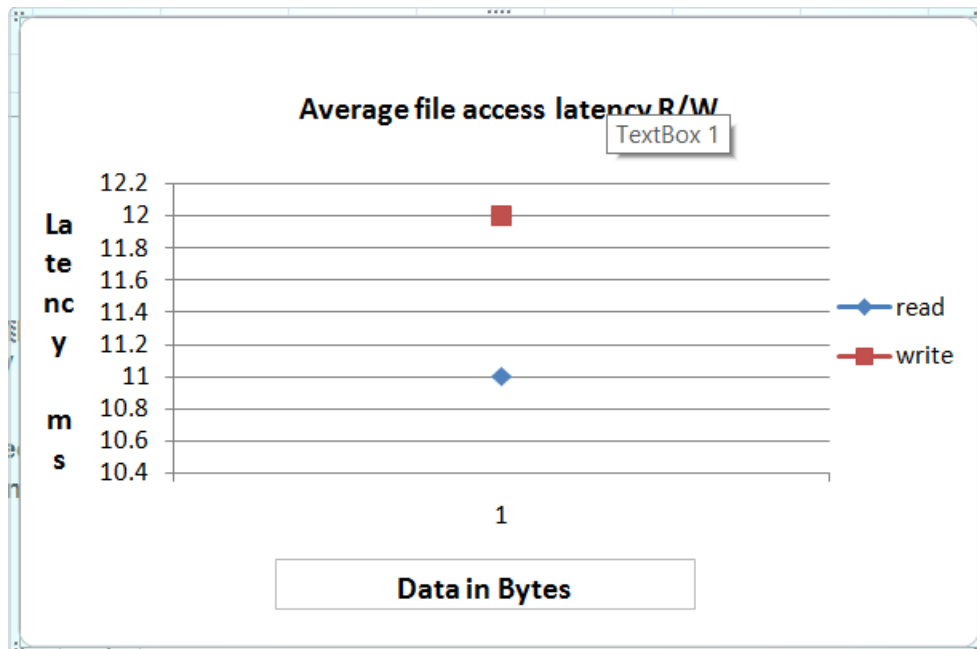


In the above experiment we are removing all the 411 files stored in the cloud without cache. Here throughput is measured in MB/s. Variance in efficiency can be clearly seen by increasing the threads from 1 to 4. Throughput is measured by:

Throughput = Total Data transferred/Time taken in minutes



In the above experiment we are removing all the 411 files stored in the cloud with cache. Here throughput is measured in MB/s. By comparing this experiment with the previous one(removing, without cache) we find that the throughput is decreased by a small amount. The reason should be that the portion of data stored in cache must also be cleared, which must have taken a few more seconds. Threading, again, plays an important role in the efficiency.



The figure above shows the average read/write latency in milli seconds for data of the size 1 Byte. The results showed that the latency is almost same for both the file access access, albiet a marginal difference.