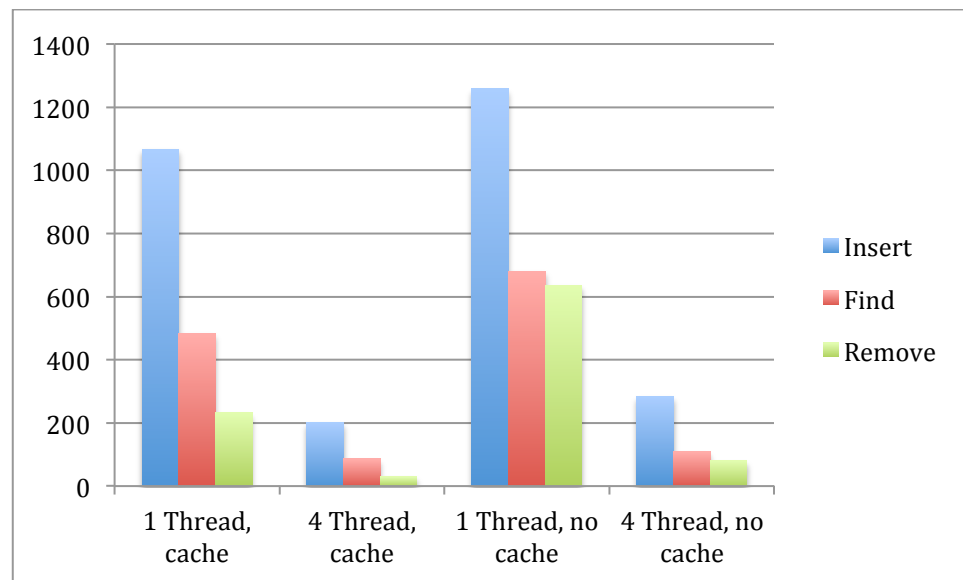## PERFORMANCE

These are the metrics obtained for the different operations required in this assignment. Note that we are varying the number of threads, the operation and the fact of being using or not memcache.

	Insert	Find	Remove
1 Thread, cache	1066,42921	482,75428	231,558848
4 Thread, cache	202,554376	85,865238	30,16058
1 Thread, no cache	1258,689849	680,853981	635,722395
4 Thread, no cache	284,070785	109,202287	80,581697

As we can see the insert operation is the one that requires more time to execute because we are writing the files on Google App Engine. Find and remove do not take as long as insert, because we are only looking for a file or deleting.

As expected, the number of threads is the most relevant fact. What we can observe on this results that the time seems to be divided by four when we use 4 threads instead of just one and it is because the process is splitted and we can run it concurrently.

Also we can notice that memcache has an important role on finding or removing methods because when we try to access to the correspondent file it is previously checked if it is in memcache and returns it in a very faster way than if it was in Cloud Storage.

To calculate the latency we insert a file through the client to get the upload time and then we find the same file to get downloading time. The latency will be the addition of this both times.

Ping to our GAE application: 20 ms

> File less than 100 KB, stored in Memcache

Latency for a memcache file: 48+25 = 73ms

```
⊕ 2014-11-11 21:35:00.668  /find  200  25ms  0kb  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2  module=
⊕ 2014-11-11 21:34:54.625  /  200  6ms  0kb  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2  module=defa
⊕ 2014-11-11 21:34:51.919  /insert  200  48ms  0kb  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2  modul
```

Logs in GAE for the last two operations on a file in memcache.

> File more than 100 KB, stored in Google Cloud Storage (2MB)

Latency for a non-memcache file: 1029+713 = 1742 ms

```
Response Size  21:33:49.321  /find  200  713ms  1205kb  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2  mo
⊕ 2014-11-11 21:33:43.626  /  200  6ms  0kb  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2  module=defa
⊕ 2014-11-11 21:33:42.024  /insert  200  1029ms  0kb  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2  mo
```

Logs in GAE for the last two operations on a file in Google Cloud Storage.

As we can see, the latency for a memcache file is slightly bigger than the ping. But if we are working with a file that is bigger than 100 KB, the latency is very much bigger than the other cases.

The througput is defined as the inverse of the latency.

For a memcache file: 109.58 Kb/s

For a non-memcache: 4.59 Kb/s