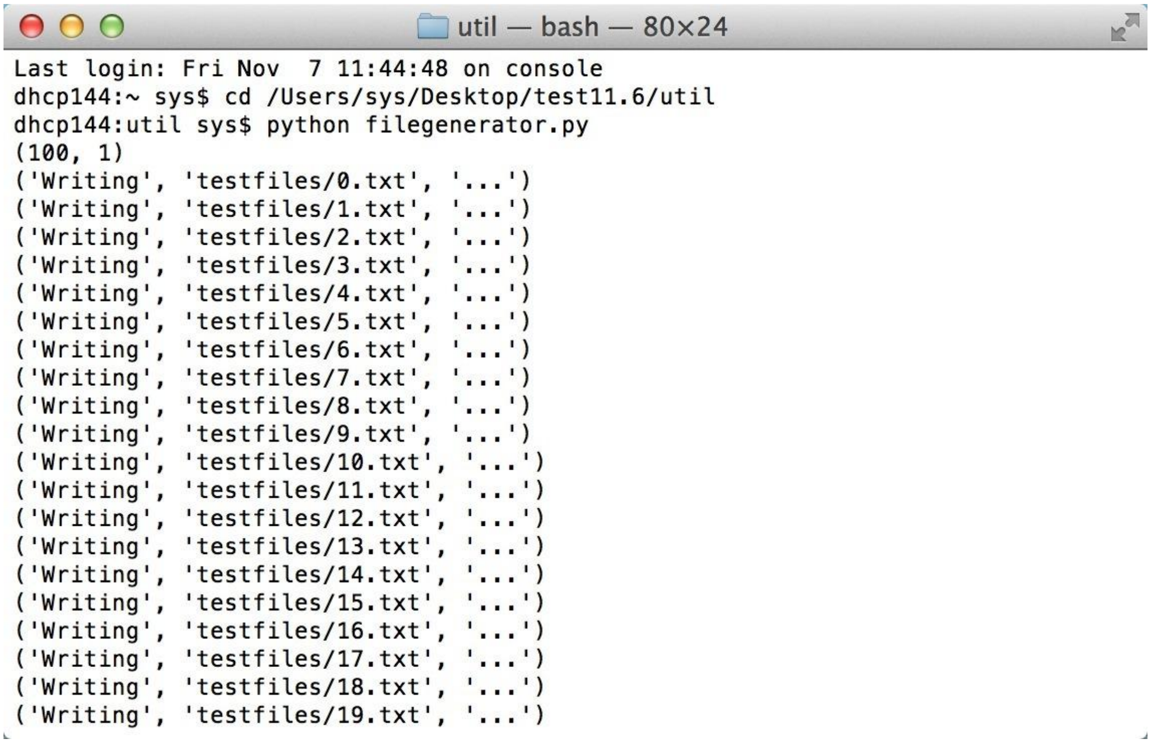
Performance

1. First we run the script to generate the files:



```
Last login: Fri Nov  7 11:44:48 on console
dhcp144:~ sys$ cd /Users/sys/Desktop/test11.6/util
dhcp144:util sys$ python filegenerator.py
(100, 1)
('Writing', 'testfiles/0.txt', '...')
('Writing', 'testfiles/1.txt', '...')
('Writing', 'testfiles/2.txt', '...')
('Writing', 'testfiles/3.txt', '...')
('Writing', 'testfiles/4.txt', '...')
('Writing', 'testfiles/5.txt', '...')
('Writing', 'testfiles/6.txt', '...')
('Writing', 'testfiles/7.txt', '...')
('Writing', 'testfiles/8.txt', '...')
('Writing', 'testfiles/9.txt', '...')
('Writing', 'testfiles/10.txt', '...')
('Writing', 'testfiles/11.txt', '...')
('Writing', 'testfiles/12.txt', '...')
('Writing', 'testfiles/13.txt', '...')
('Writing', 'testfiles/14.txt', '...')
('Writing', 'testfiles/15.txt', '...')
('Writing', 'testfiles/16.txt', '...')
('Writing', 'testfiles/17.txt', '...')
('Writing', 'testfiles/18.txt', '...')
('Writing', 'testfiles/19.txt', '...')
```

Then run the test:

Totally run 12 experiment

Experiment 1:

Operations: Insert

Number of client threads: 1

Memcache: Yes

Experiment 2:

Operations: Insert

Number of client threads: 1

Memcache: No

Experiment 3:

Operations: Insert

Number of client threads: 4

Memcache: Yes

Experiment 4:

Operations: Insert

Number of client threads: 4

Memcache: NO

Experiment 5:

Operations: Find

Number of client threads: 1

Memcache: Yes

Experiment 6:

Operations: Find

Number of client threads: 4

Memcache: Yes

Experiment 7:

Operations: Find

Number of client threads: 1

Memcache: NO

Experiment 8:

Operations: Find

Number of client threads: 4

Memcache: NO

Experiment 9:

Operations: Remove

Number of client threads: 1

Memcache: Yes

Experiment 10:

Operations: Remove

Number of client threads: 1

Memcache: No

Experiment 11:

Operations: Remove

Number of client threads: 4

Memcache: Yes

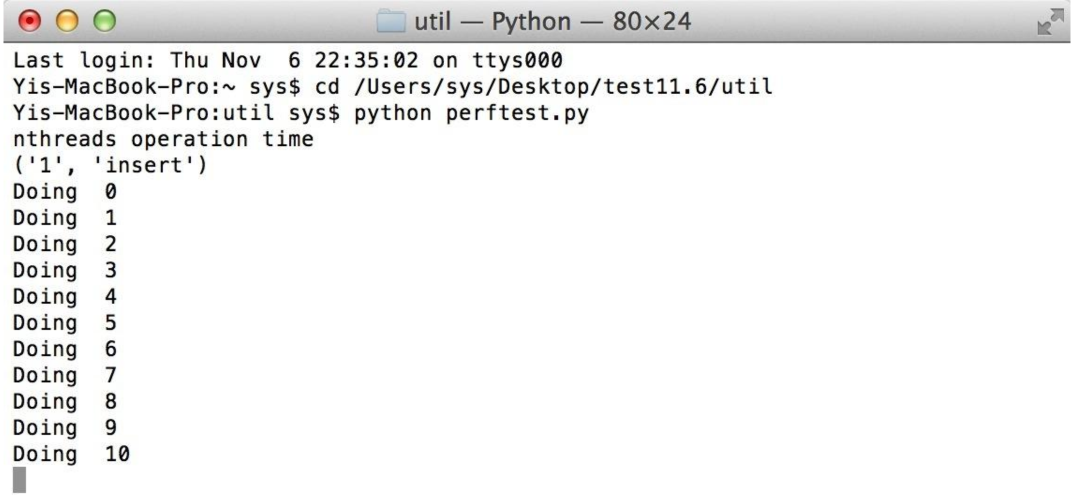
Experiment 12:

Operations: Remove

Number of client threads: 4

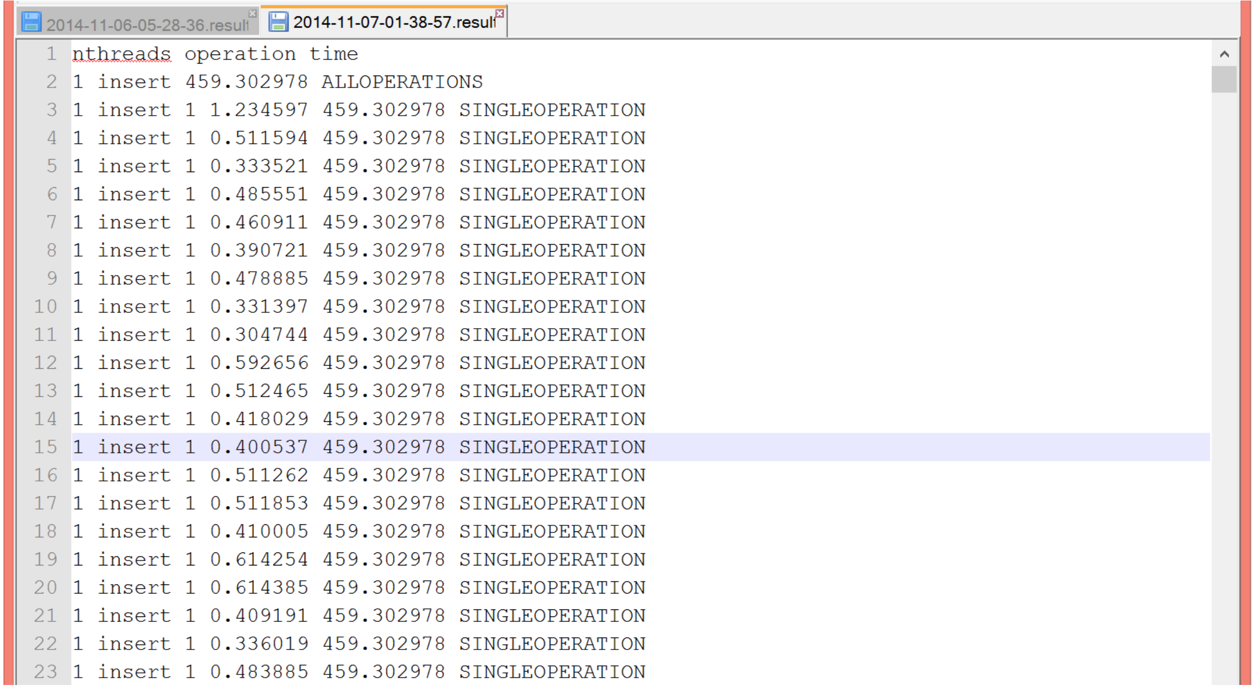
Memcache: No

When doing script testing, it shows the process with the number of threads, operation and time.



```
Last login: Thu Nov  6 22:35:02 on ttys000
Yis-MacBook-Pro:~ sys$ cd /Users/sys/Desktop/test11.6/util
Yis-MacBook-Pro:util sys$ python perftest.py
nthreads operation time
('1', 'insert')
Doing  0
Doing  1
Doing  2
Doing  3
Doing  4
Doing  5
Doing  6
Doing  7
Doing  8
Doing  9
Doing  10
```

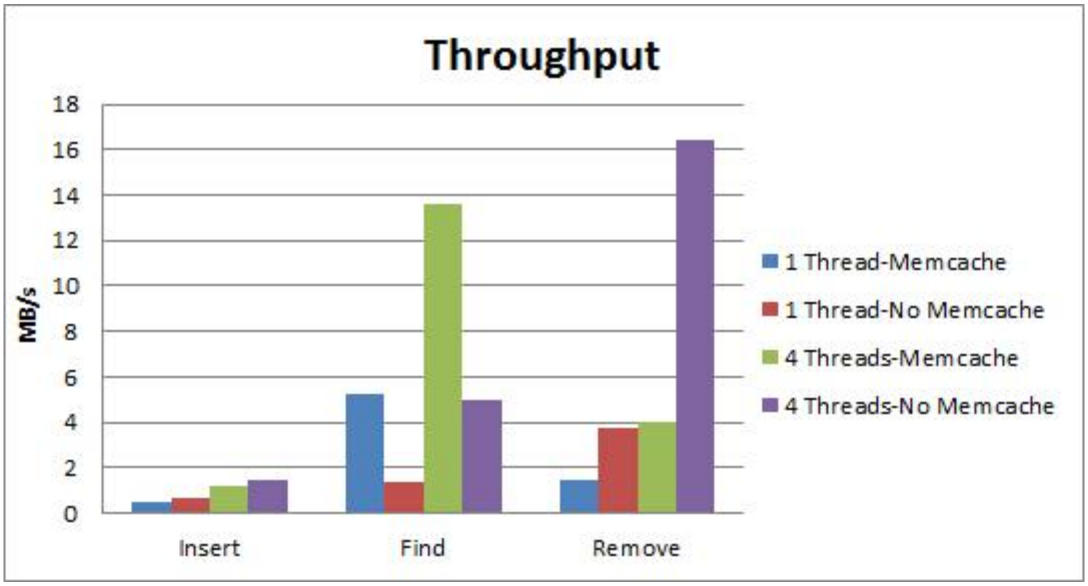
The Output is shown in the log as follows.



```
nthreads operation time
1 insert 459.302978 ALLOPERATIONS
1 insert 1 1.234597 459.302978 SINGLEOPERATION
1 insert 1 0.511594 459.302978 SINGLEOPERATION
1 insert 1 0.333521 459.302978 SINGLEOPERATION
1 insert 1 0.485551 459.302978 SINGLEOPERATION
1 insert 1 0.460911 459.302978 SINGLEOPERATION
1 insert 1 0.390721 459.302978 SINGLEOPERATION
1 insert 1 0.478885 459.302978 SINGLEOPERATION
1 insert 1 0.331397 459.302978 SINGLEOPERATION
1 insert 1 0.304744 459.302978 SINGLEOPERATION
1 insert 1 0.592656 459.302978 SINGLEOPERATION
1 insert 1 0.512465 459.302978 SINGLEOPERATION
1 insert 1 0.418029 459.302978 SINGLEOPERATION
1 insert 1 0.400537 459.302978 SINGLEOPERATION
1 insert 1 0.511262 459.302978 SINGLEOPERATION
1 insert 1 0.511853 459.302978 SINGLEOPERATION
1 insert 1 0.410005 459.302978 SINGLEOPERATION
1 insert 1 0.614254 459.302978 SINGLEOPERATION
1 insert 1 0.614385 459.302978 SINGLEOPERATION
1 insert 1 0.409191 459.302978 SINGLEOPERATION
1 insert 1 0.336019 459.302978 SINGLEOPERATION
1 insert 1 0.483885 459.302978 SINGLEOPERATION
```

2. Throughput

According to the data statistically by insert, find and remove operation under 1 thread, 4 threads with memcache and without memcache separately, the throughput figure is shown in the following.



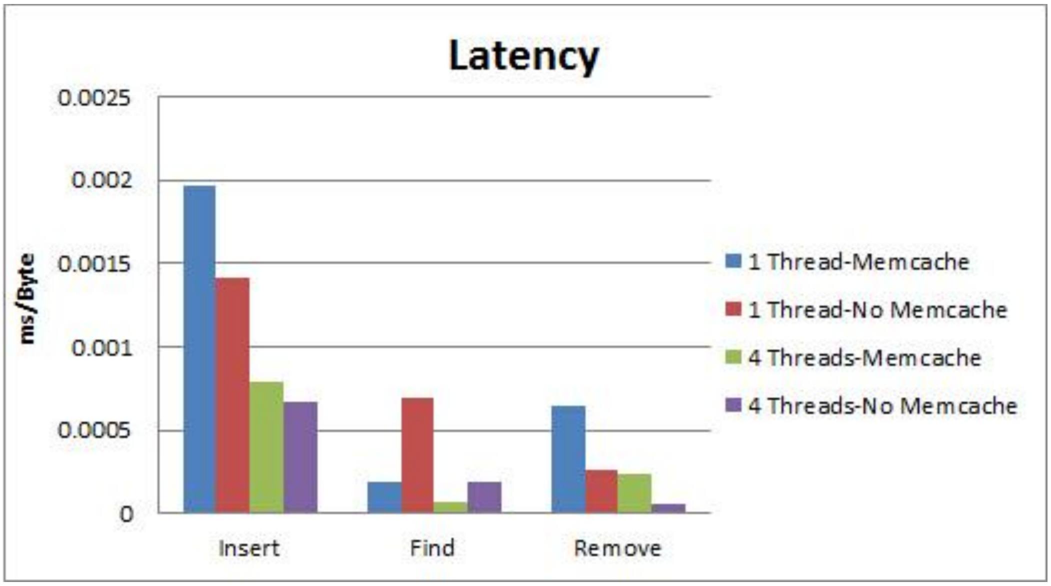
(1). From this figure, we can see that, the throughput of insert operation is much slower than other operations. Actually, insert operation is writing data to Google Cloud Storage or memcache, so it takes more time.

(2). Throughput of 1 thread operations are much slower than multithreads. The more threads we have, the faster speed it will be.

(3). In addition to the influence of different operations and multithreads, whether write data to memcache is another factor, due to the reason that for small size of data, it will store them not only into the Google Cloud Storage, but also into the memcache. The extra operations will increase the time and decrease the throughput.

3. Latency

According to the data statistically by insert, find and remove operation under 1 thread, 4 threads with memcache and without memcache separately, the latency figure is shown in the following.



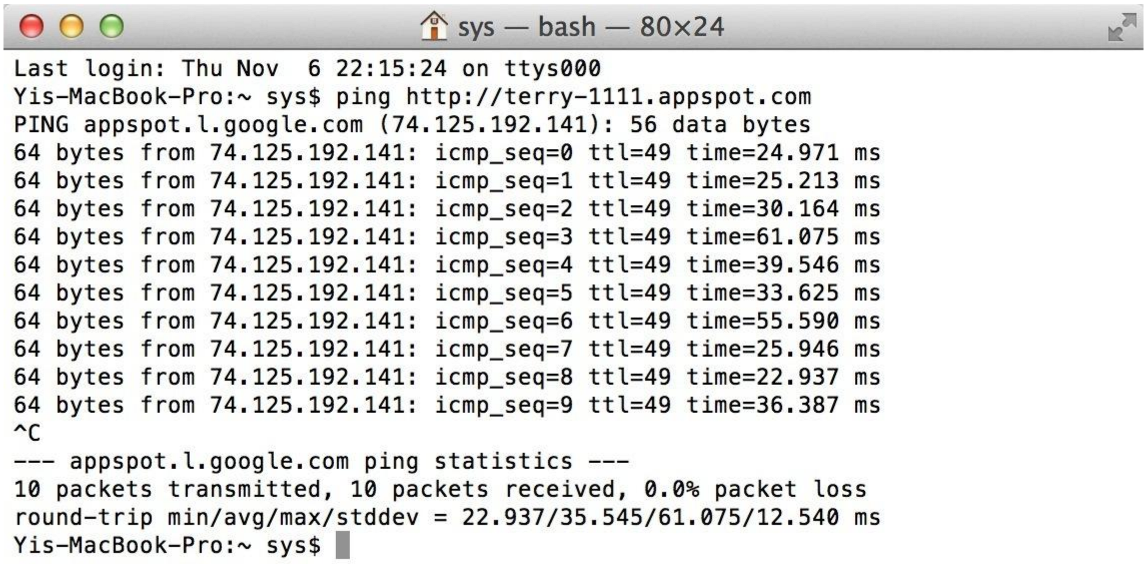
(1). Among the operations of insert, find and remove, the latency of insert operation is the largest. The latency of find operation is similar to that of remove operation.

(2). Latency of 1 thread operations are much higher than multithreads. The more threads we have, the less latency it will be.

(3). In addition to the influence of different operations and multithreads, whether write data to memcache is another factor, due to the reason that for small size of data, it will store them not only into the Google Cloud Storage, but also into the memcache. The extra operations will increase the latency.

4. Ping operation

When use the ping command to ping this url, we can get the minimum round-trip time is 22.937ms for transmitting 64 bytes, so that the latency is 0.3584 ms/byte.



```
Last login: Thu Nov  6 22:15:24 on ttys000
Yis-MacBook-Pro:~ sys$ ping http://terry-1111.appspot.com
PING appspot.l.google.com (74.125.192.141): 56 data bytes
64 bytes from 74.125.192.141: icmp_seq=0 ttl=49 time=24.971 ms
64 bytes from 74.125.192.141: icmp_seq=1 ttl=49 time=25.213 ms
64 bytes from 74.125.192.141: icmp_seq=2 ttl=49 time=30.164 ms
64 bytes from 74.125.192.141: icmp_seq=3 ttl=49 time=61.075 ms
64 bytes from 74.125.192.141: icmp_seq=4 ttl=49 time=39.546 ms
64 bytes from 74.125.192.141: icmp_seq=5 ttl=49 time=33.625 ms
64 bytes from 74.125.192.141: icmp_seq=6 ttl=49 time=55.590 ms
64 bytes from 74.125.192.141: icmp_seq=7 ttl=49 time=25.946 ms
64 bytes from 74.125.192.141: icmp_seq=8 ttl=49 time=22.937 ms
64 bytes from 74.125.192.141: icmp_seq=9 ttl=49 time=36.387 ms
^C
--- appspot.l.google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 22.937/35.545/61.075/12.540 ms
Yis-MacBook-Pro:~ sys$
```