6650 A1 Skier System Design Document and Test Results

1. Major classes and packages

Package Part1:

Class SkierClient:

This is the major client class that generate the 3 thread pools for phase 1 and 2, and start the three phases in a step-controlled manner with use of countdown latches. All threads are added to a BlockingQueue so that the statistic information can be retrieved later for analysis. As the main method create and start the thread pools, it also time the wall time for all thread to finish and after that calculate the statistics based on information stored in each thread. The wall time and overall success count of threads are printed to console.

One thing to note is that swagger skiers api is called synchronously to send over the requests the servlet.

Class SkierUpdateThread:

This is the thread class that generated to send out designated number of http requests with randomly generated skier id, lift id, time etc. Those numbers are generated within a designated range.
Each SkierUpdateThread object will also time the http request and store the start time, end time, and http response code of each request it sent into a ResponseStat object. Each ResponseStat object is added to an instance variable list.

Package Part2:

Class MetricsReporter:

It reads data store in all threads, and calculate the mean, median, p99, max latencies as well as the through put. The stats are printed to console.
Besides, it also extracts all http requests data from each thread and write them into disk file in csv format.

Class ResponseStat:

POJO that stores start time, end time and http response code of a request. It is instantiated as each thread sending a new request, stored in an instance variable list of that thread, and accessed by MetricsReporter later for statistical analysis and report.

2. Relationships

In summary, SkierClient creates SkierUpdateThread. SkierUpdateThread send http requests and store request data in POJOs of ResponseStat. MetricsReporter read those data and output to screen and files.

Test Result: 32 Thread, 20000 skiers, 20 runs/skier

-----------------Part 1 Report------------------

number of successful requests sent: 400000

number of unsuccessful requests sent: 0

Wall time: 1394291 (23.2min)

-----------------Part 2 Report------------------

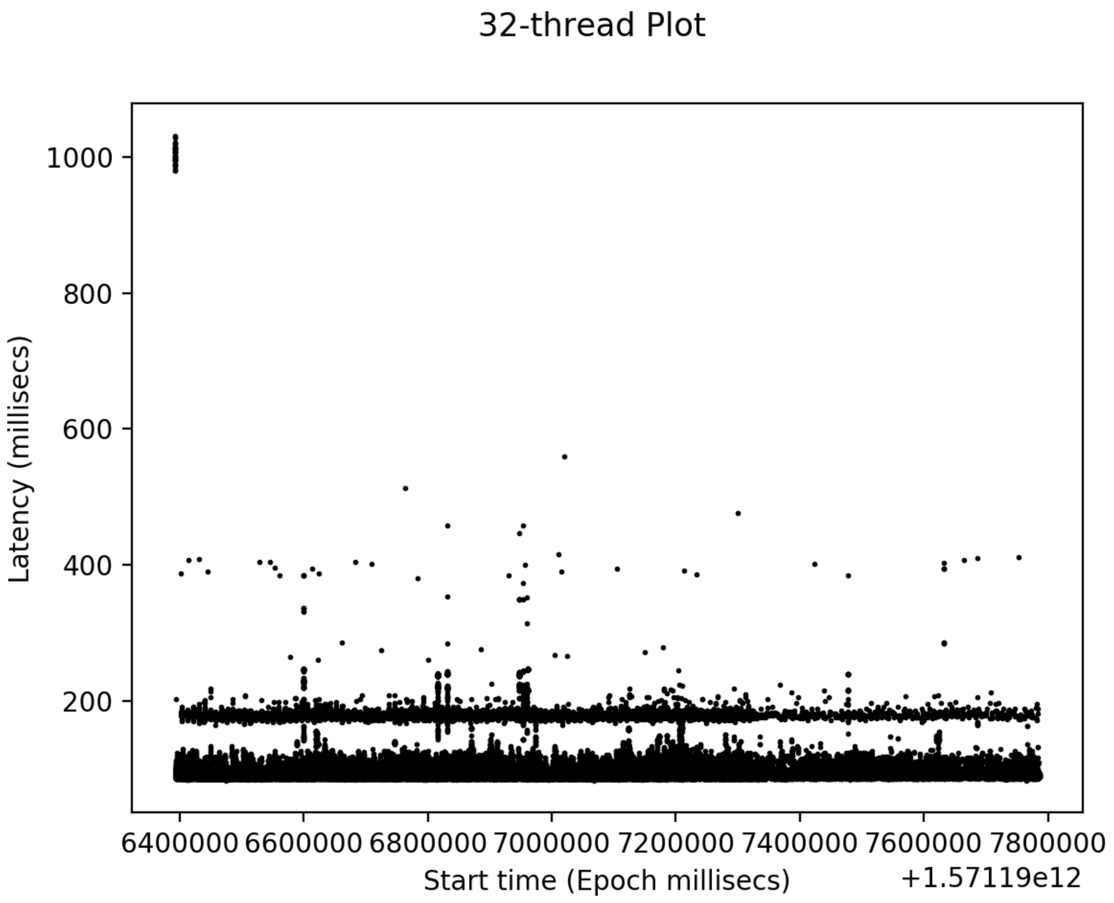
mean response time (millisecs): 93.1106

median response time (millisecs): 91.0

throughput (total number of requests/wall time): 3.4857275

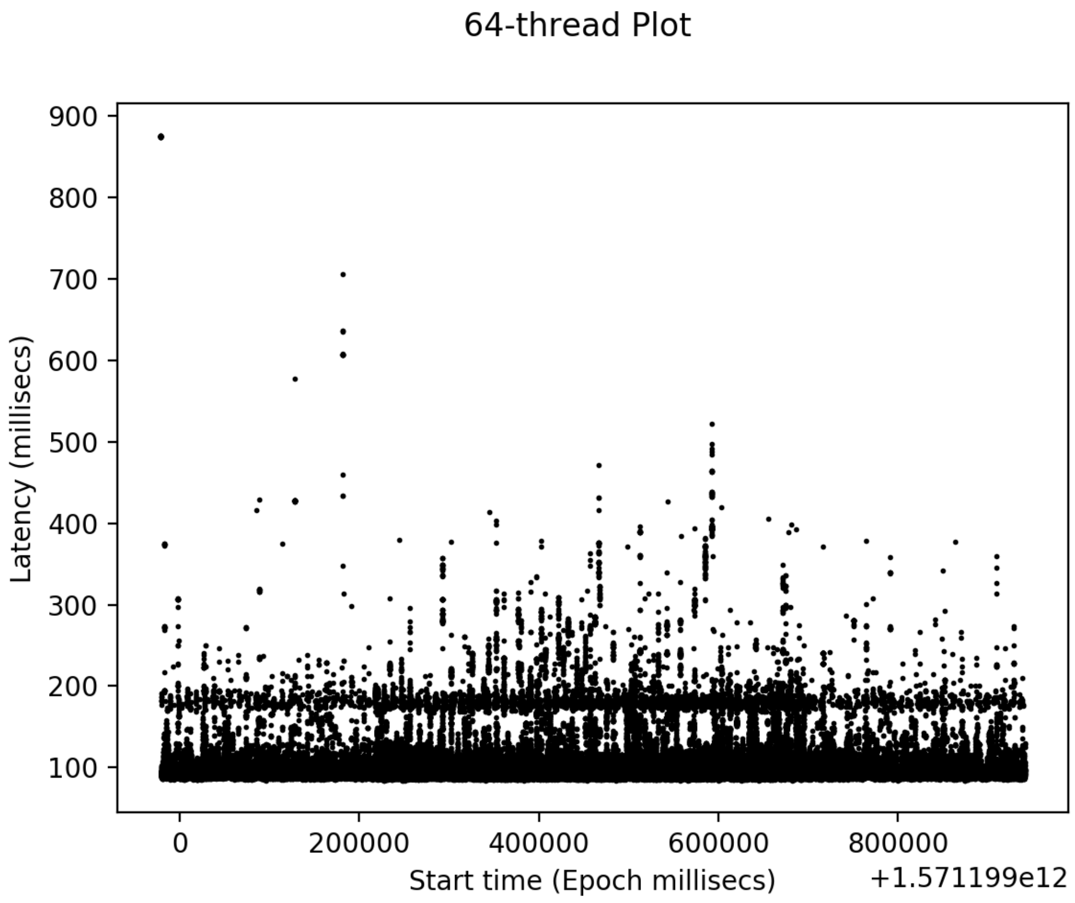
p99 response time (99th percentile, millisecs): 182.0

max response time(millisecs): 1031.0



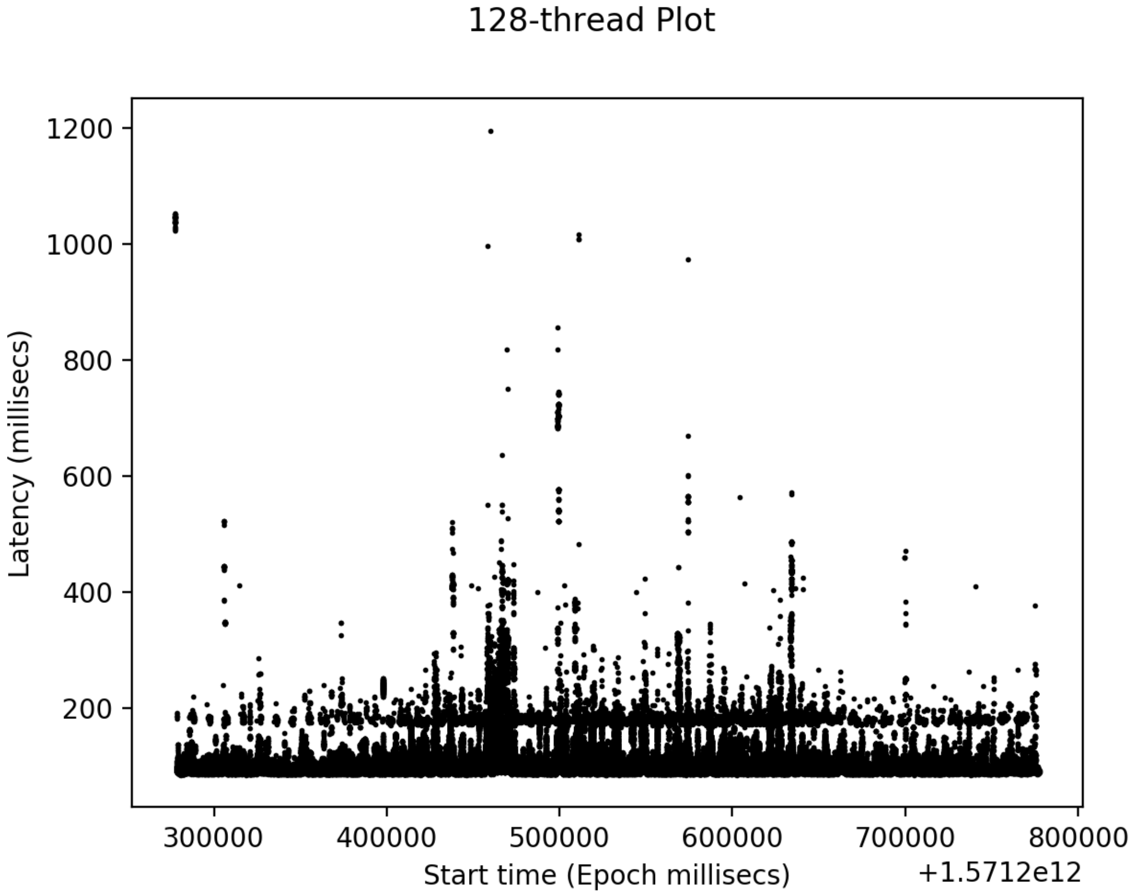
Test Result: 64 Thread, 20000 skiers, 20 runs/skier

-----------------Part 1 Report------------------
number of successful requests sent: 400000
number of unsuccessful requests sent: 0
Wall time: 963551 (16.1min)
-----------------Part 2 Report------------------
mean response time (millisecs): 96.5650925
median response time (millisecs): 92.0
throughput (total number of requests/wall time): 2.4088775
p99 response time (99th percentile, millisecs): 237.0
max response time(millisecs): 876.0



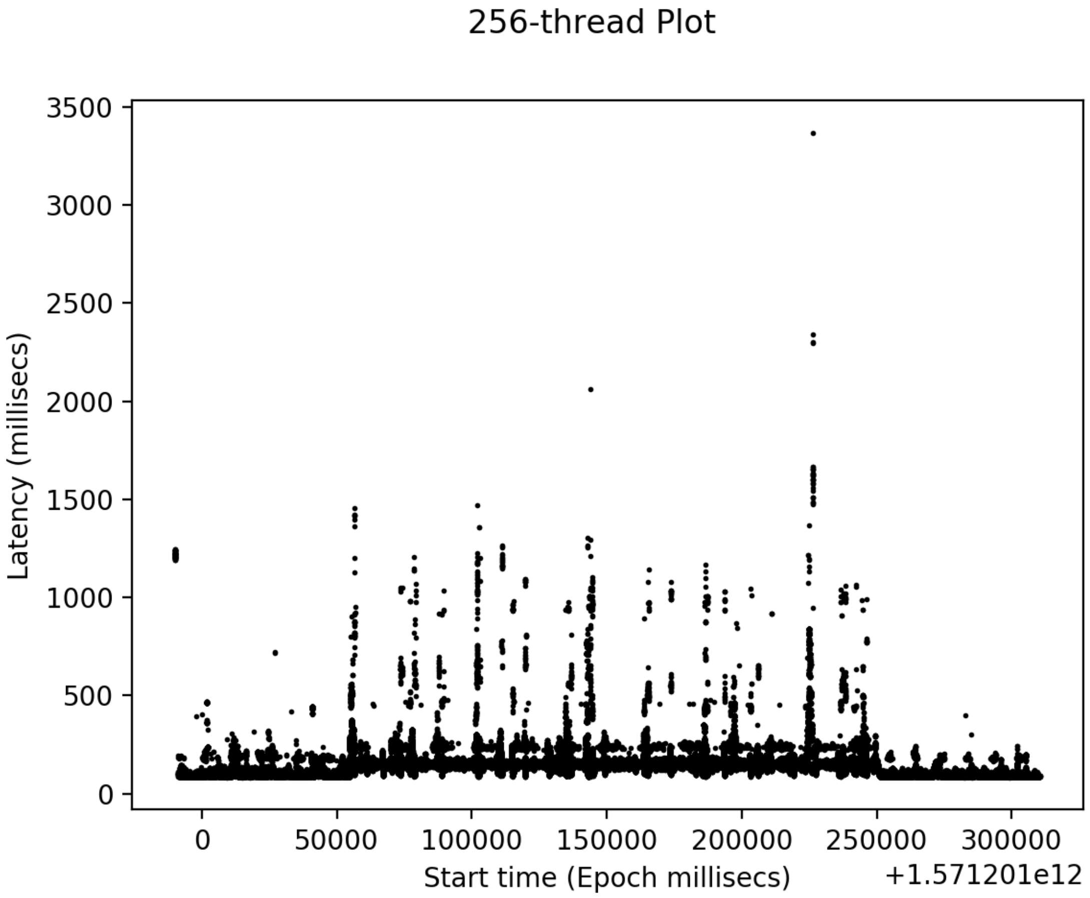
Test Result: 128 Thread, 20000 skiers, 20 runs/skier

```
-----------------Part 1 Report------------------
number of successful requests sent: 400000
number of unsuccessful requests sent: 0
Wall time: 500576 (8.3min)
-----------------Part 2 Report------------------
mean response time (millisecs): 102.0627975
median response time (millisecs): 95.0
throughput (total number of requests/wall time): 1.25144
p99 response time (99th percentile, millisecs): 329.0
max response time(millisecs): 1196.0
```



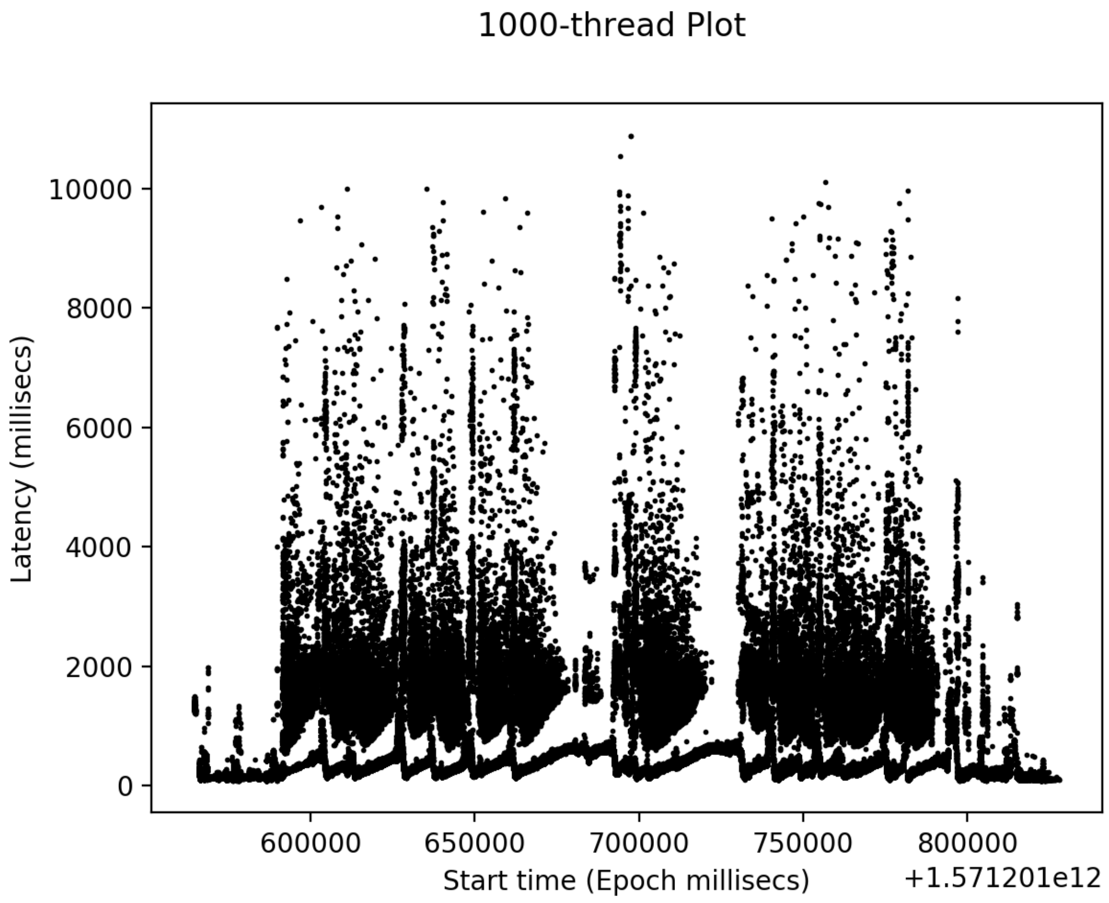
Test Result: 256 Thread, 20000 skiers, 20 runs/skier

```
-----------------Part 1 Report------------------
number of successful requests sent: 399744
number of unsuccessful requests sent: 0
Wall time: 320859 (5.4min)
-----------------Part 2 Report------------------
mean response time (millisecs): 145.71423210854948
median response time (millisecs): 151.0
throughput (total number of requests/wall time): 0.8021475
p99 response time (99th percentile, millisecs): 767.0
max response time(millisecs): 3367.0
```



Stress Test Result: 1000 Thread, 20000 skiers, 20 runs/skier

```
-----------------Part 1 Report------------------
number of successful requests sent: 399721
number of unsuccessful requests sent: 279 (A lot of request timeout)
Wall time: 263850 (4.4min)
-----------------Part 2 Report------------------
mean response time (millisecs): 536.7079588012639
median response time (millisecs): 312.0
throughput (total number of requests/wall time): 0.659625
p99 response time (99th percentile, millisecs): 7424.0
max response time(millisecs): 10887.0 (not including timeout)
```



Observation on the Test Results:

1. Overall (average, median and p99) latency increases as number of threads increases, eventually lead to a portion of them time out when the number reach 1000 under stress test.

2. The standard deviation of the latency also increases as the result of higher load.

3. The latency has pattern of waves as it increases from base line drastically at certain time. This might be because of how server balance the load it receives.