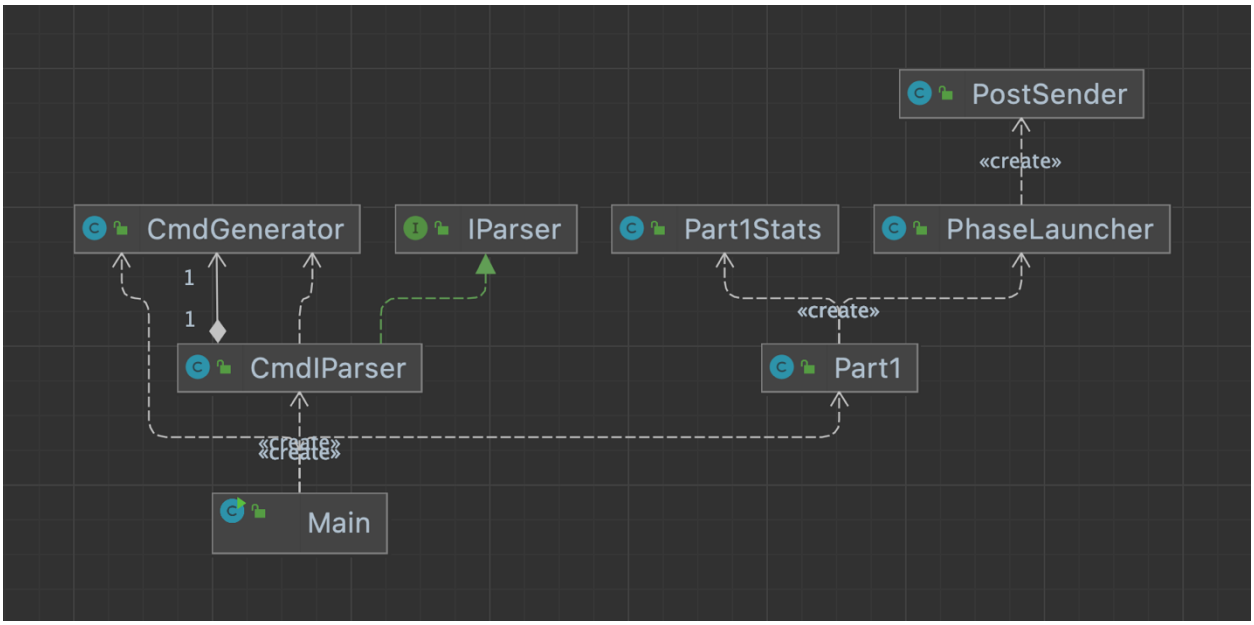
CS6650 Assignment 1

URL of git:

https://github.com/SeanChenXH/CS6650

Client Design Description:

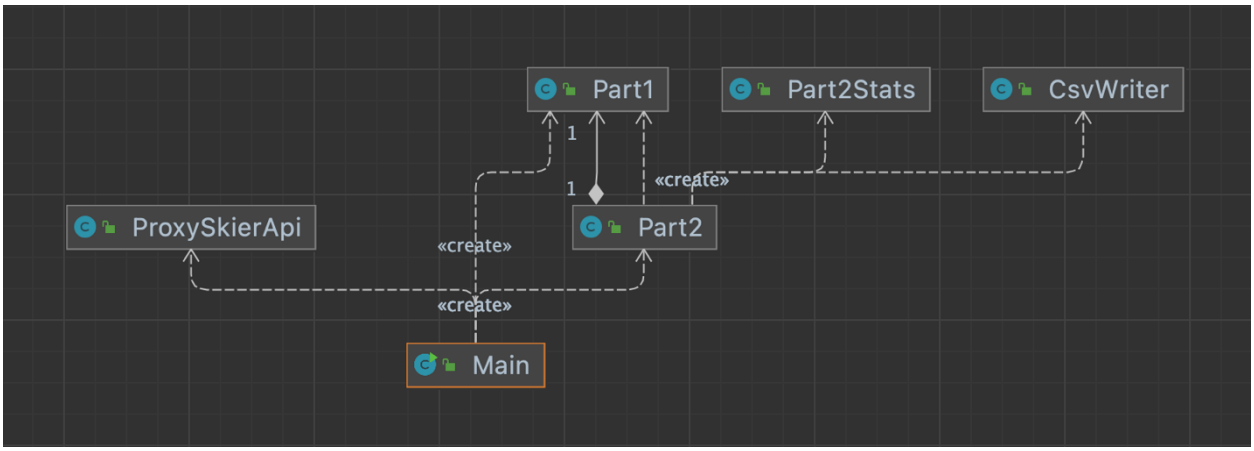
Client 1:



1. The **Main** class is the entry point to run **Part1** by specifying the **configuration** of **Parser** and **SkiersApi**
2. The interface **IParser** is to parse command line String[] into Parameters object
3. The **CmdGenerator class** is to generate system command-line options, some of which are required and some of which are optional.
4. The **CmdParser** class that implements **IParser** interface is to parse the external command line parameters of numTreads, numSkiers, numLifts, numRuns, and IP/port as **Parameter** Object. It also acts as a validator to check the correctness of the input command line arguments. For example, the maximum number of threads should not exceed 1024, and the IPv4 address should be referred to as dotted-decimal format or "localhost".

5. The **Config** class acts as the system's internal configuration center, coordinating with **IParser**, **Parameters**, **ISkierAPi**, and **Counter**. By decoupling classes, we can choose different mechanisms more flexibly. For example, we can send API requests through Swagger API or Apache Java HTTP API, or other means.
6. The **Part1** class will launch three phases by **PhaseLauncher**. After three phases have been completed, the statistics will be printed by **Part1Stats**.
7. The **PhaseLauncher** class is to execute 3 phases, each of which will send many lifts ride requests to the server APIs by **PostSender**. Phase 1 will launch numTreads / 4 threads. **Phase 1 CountDownLatch** is used to monitor the completion of Phase1 threads. Once 20% of the threads in Phase 1 have been completed, Phase 2 begins. Phase 2 will launch numTreads threads to send requests. As same as before, **Phase 2 CountDownLatch** is used to monitor the completion of Phase 2 threads. Once 20 % of the threads in Phase 2 have been completed, Phase 3 begins. **Complete CountDownLatch** is designed as a global CountDownLatch which is to monitor the completion of the whole 3 phases.

Client 2:



1. The **Main** class is the entry point to run **Part2** by specifying the configuration of the **Parser** and **SkiersApi**. In Part2, we used Proxy Design Pattern to provide an Object **ProxySkierApi** that acts as a substitute for **SkiersApi**, adding some additional behaviors, such as calculating the latency of each post request.
2. The **Part2** class is to run **Part1** again. In addition, it will use **CsvWriter** to write out a **Record** to a CSV file containing start time, request type, latency, and response code for

each request. In the end, it will use **Part2Stats** to print out the mean response time, median response time, throughput, p99, minimum response time, and maximum response time.

3. **ProxySkierApi** class acts as a substitute for **SkiersApi**, providing more additional behaviors of calculating the latency of each post and recording the latency to the Record Object that will be written to CSV files after all phases of post requests are completed
4. The **Part2Stats** class is to calculate the data from **Part2** and print out the statistics.
5. The **CsvWriter** class is to write record list into csv files

Little's Law throughput predictions:

Threads	Latency(milliseconds)	Little's Law throughput predictions(requests/s)
32	50	640
64	50	1280
128	50	2560
256	50	5120

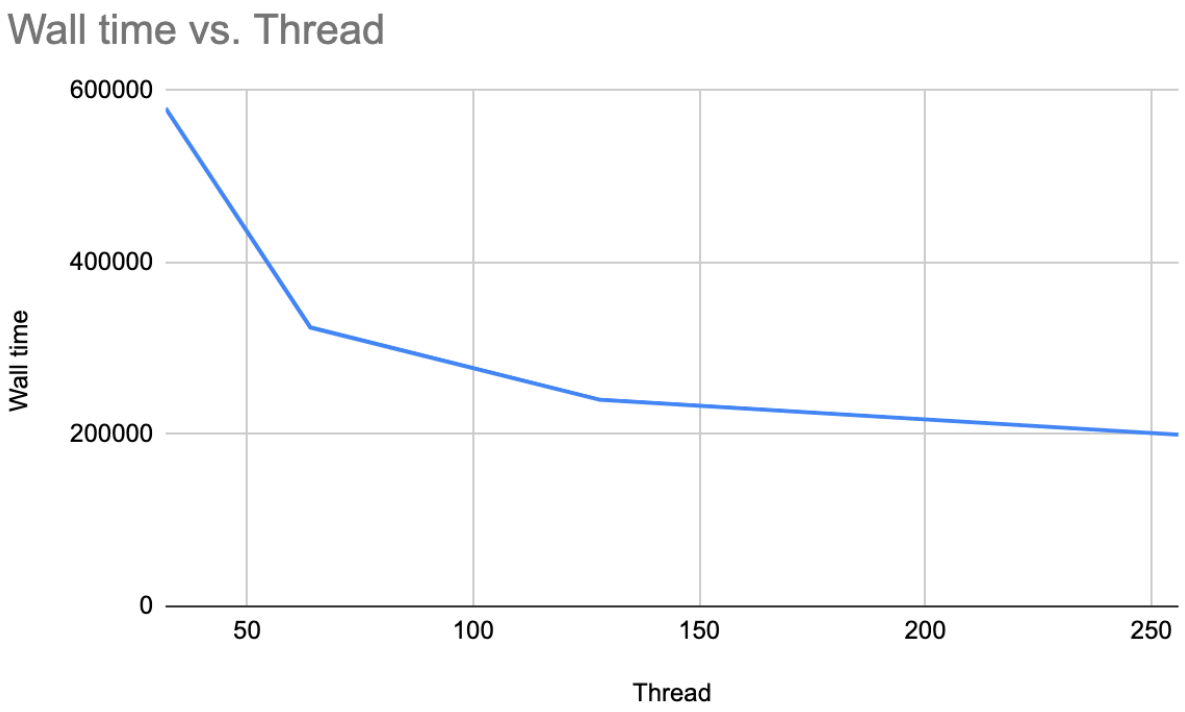
Client Part 1:

```
Thread: 32
-----------------------------------------------------
Number of successful requests sent:   320006
Number of unsuccessful requests sent: 0
Wall time: 579272 milliseconds
Throughput: 552.43 requests/seconds

Process finished with exit code 0
```

```
Thread: 64
-----------------------------------------------------
Number of successful requests sent:   319628
Number of unsuccessful requests sent: 0
Wall time: 324321 milliseconds
Throughput: 985.53 requests/seconds

Process finished with exit code 0
```

```
Thread: 128
-----------------------------------------------------
Number of successful requests sent:   319640
Number of unsuccessful requests sent: 0
Wall time: 240325 milliseconds
Throughput: 1330.03 requests/seconds

Process finished with exit code 0
```

```
Thread: 256
-----------------------------------------------------
Number of successful requests sent:   319538
Number of unsuccessful requests sent: 0
Wall time: 199700 milliseconds
Throughput: 1600.09 requests/seconds

Process finished with exit code 0
```



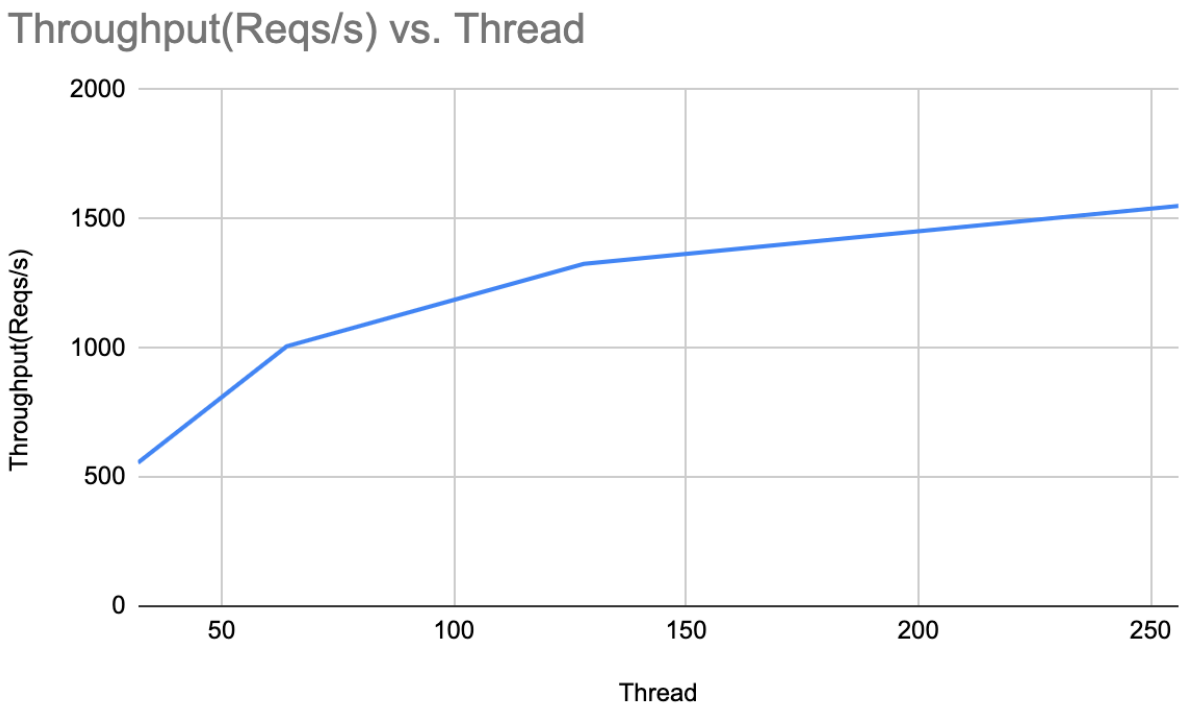
Client Part 2:

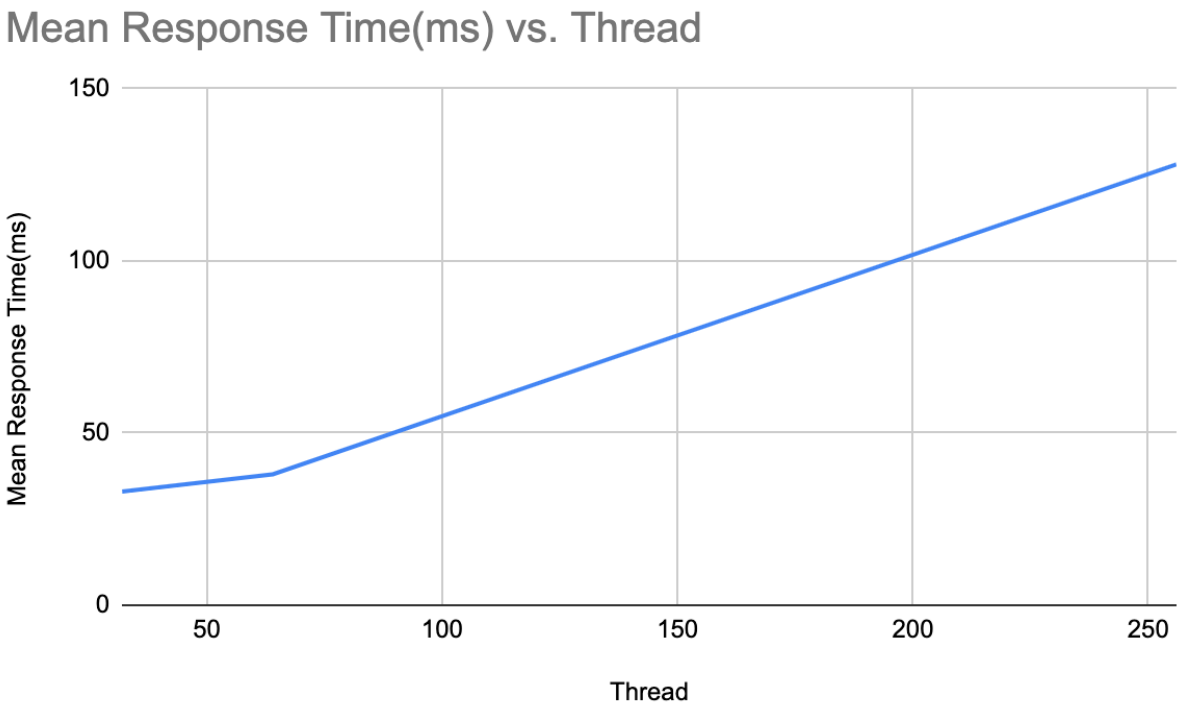
```
Thread: 32
-----------------------------------------------------
Number of successful requests sent:   320006
Number of unsuccessful requests sent: 0
Wall time: 609352 milliseconds
Throughput: 525.16 requests/seconds
-----------------------------------------------------
Mean response time: 35 milliseconds
Median response time: 32 milliseconds
Throughput: 525.16 requests/seconds
99th percentile response time: 86 milliseconds
Minimum Response time: 16 milliseconds
Maximum Response time: 500 milliseconds
-----------------------------------------------------
Wall time (part1): 609352 milliseconds
Wall time (part2): 1385 milliseconds

Process finished with exit code 0
```

```
Thread: 64
-----------------------------------------------------
Number of successful requests sent:   319628
Number of unsuccessful requests sent: 0
Wall time: 317795 milliseconds
Throughput: 1005.77 requests/seconds
-----------------------------------------------------
Mean response time: 38 milliseconds
Median response time: 36 milliseconds
Throughput: 1005.77 requests/seconds
99th percentile response time: 91 milliseconds
Minimum Response time: 15 milliseconds
Maximum Response time: 745 milliseconds
-----------------------------------------------------
Wall time (part1): 317795 milliseconds
Wall time (part2): 1514 milliseconds

Process finished with exit code 0
```

```
Thread: 128
-----------------------------------------------------
Number of successful requests sent:   319640
Number of unsuccessful requests sent: 0
Wall time: 241131 milliseconds
Throughput: 1325.59 requests/seconds
-----------------------------------------------------
Mean response time: 68 milliseconds
Median response time: 69 milliseconds
Throughput: 1325.59 requests/seconds
99th percentile response time: 258 milliseconds
Minimum Response time: 15 milliseconds
Maximum Response time: 999 milliseconds
-----------------------------------------------------
Wall time (part1): 241131 milliseconds
Wall time (part2): 1412 milliseconds

Process finished with exit code 0
```

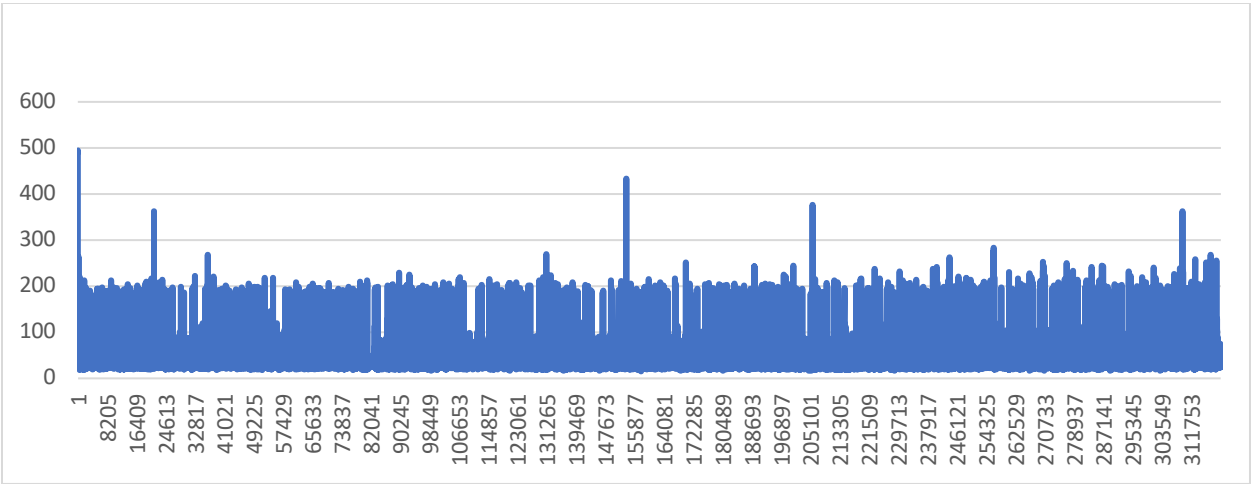
```
Thread: 256
-----------------------------------------------------
Number of successful requests sent:   319538
Number of unsuccessful requests sent: 0
Wall time: 206186 milliseconds
Throughput: 1549.76 requests/seconds
-----------------------------------------------------
Mean response time: 128 milliseconds
Median response time: 117 milliseconds
Throughput: 1549.76 requests/seconds
99th percentile response time: 541 milliseconds
Minimum Response time: 15 milliseconds
Maximum Response time: 6089 milliseconds
-----------------------------------------------------
Wall time (part1): 206186 milliseconds
Wall time (part2): 1401 milliseconds

Process finished with exit code 0
```



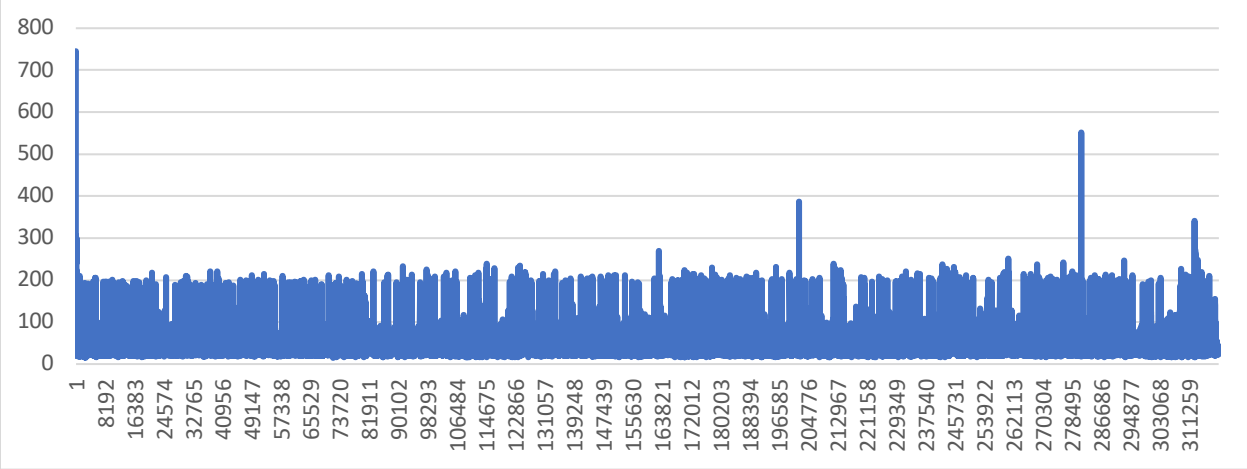


Bonus Points

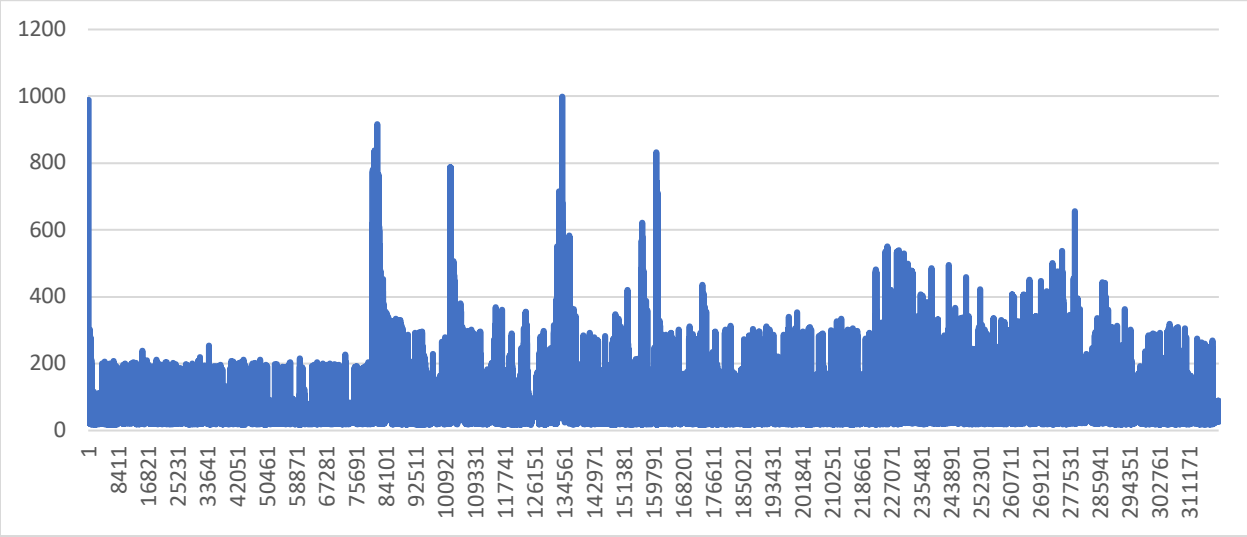
32 threads



64 threads



128 threads



256 threads