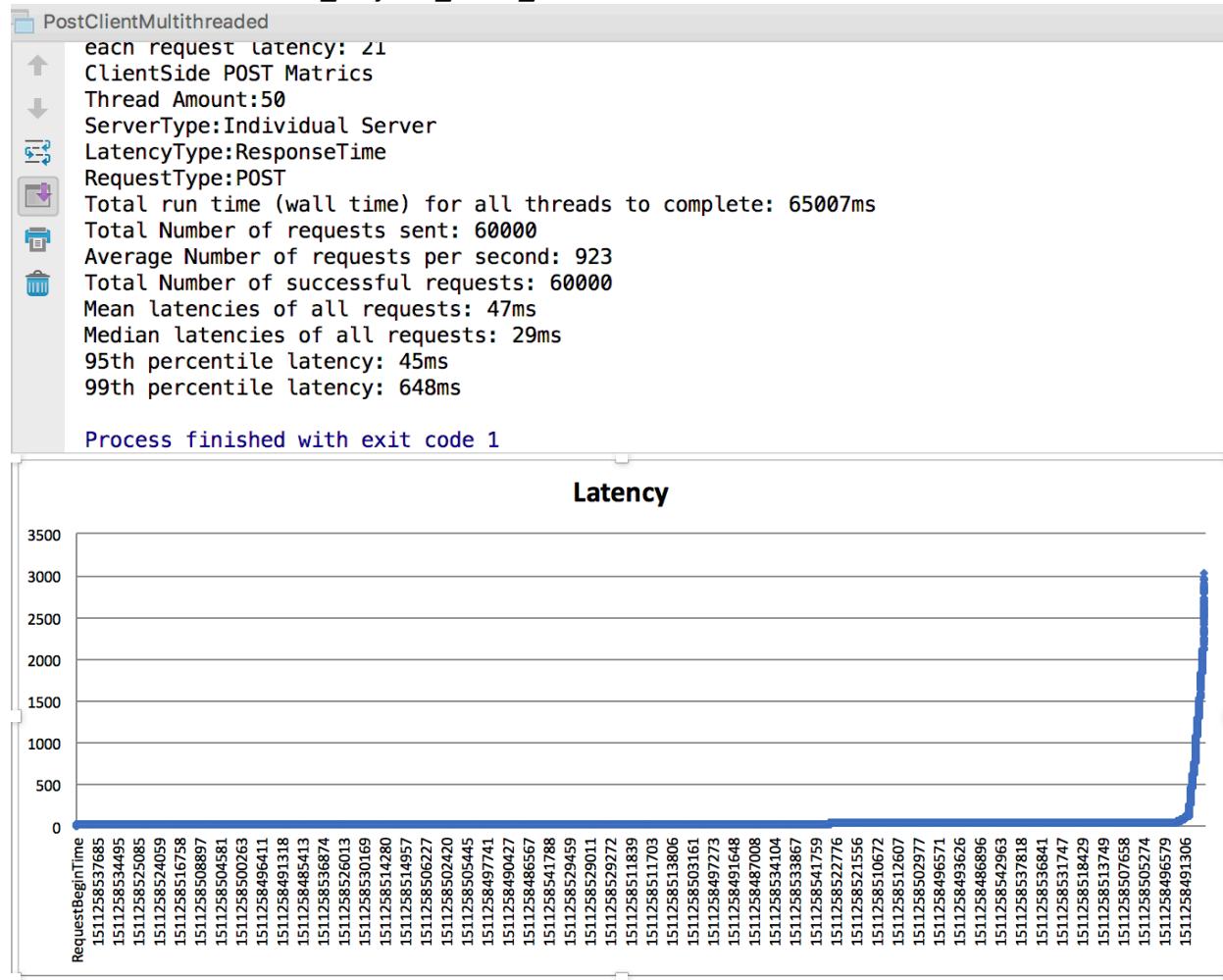


HW3-Report

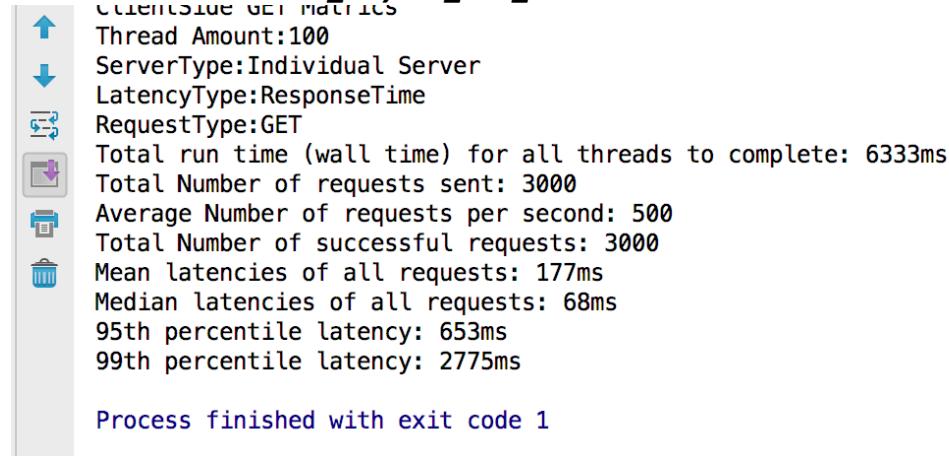
Github:<https://github.com/Kanuo007/DistributedSystem/tree/master/HW3/src>

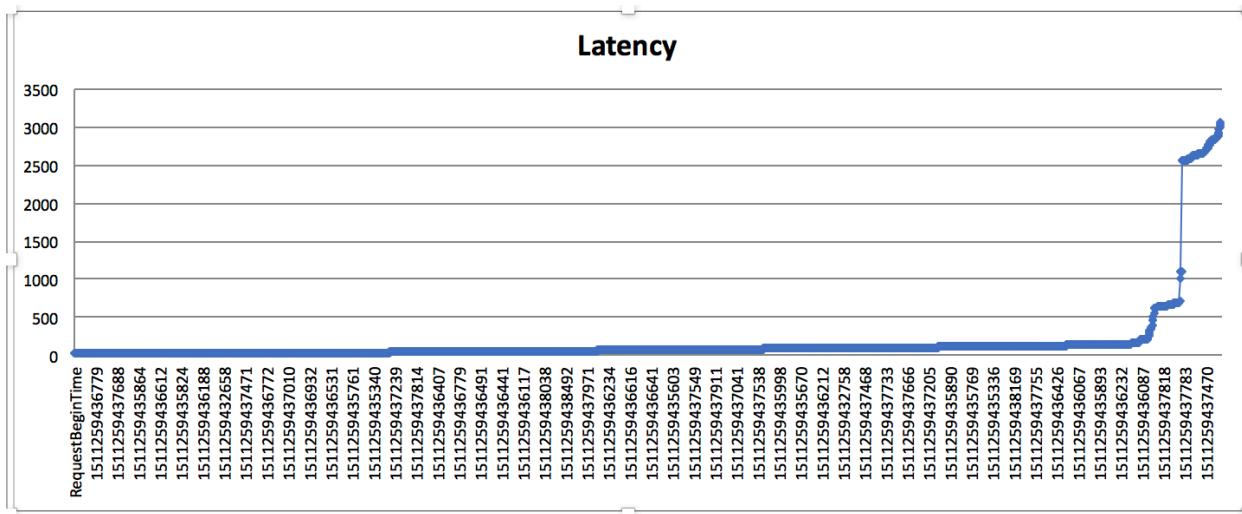
Step1

- *Individual Server_Day999_POST_Client*



- *Individual Server_Day999_GET_Client*



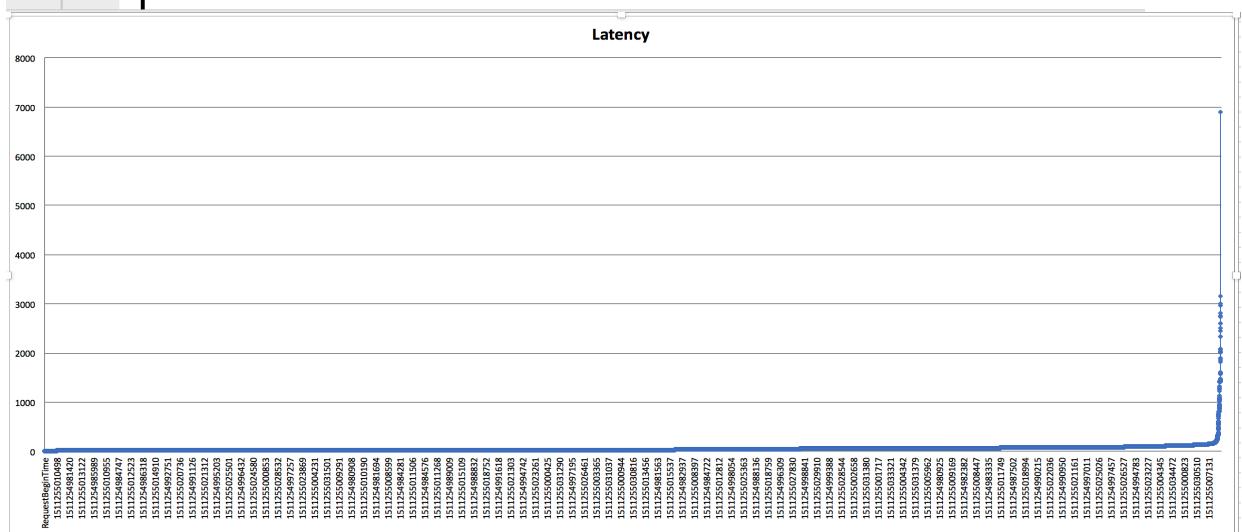


- *Three Servers_Day999_POST_Client*

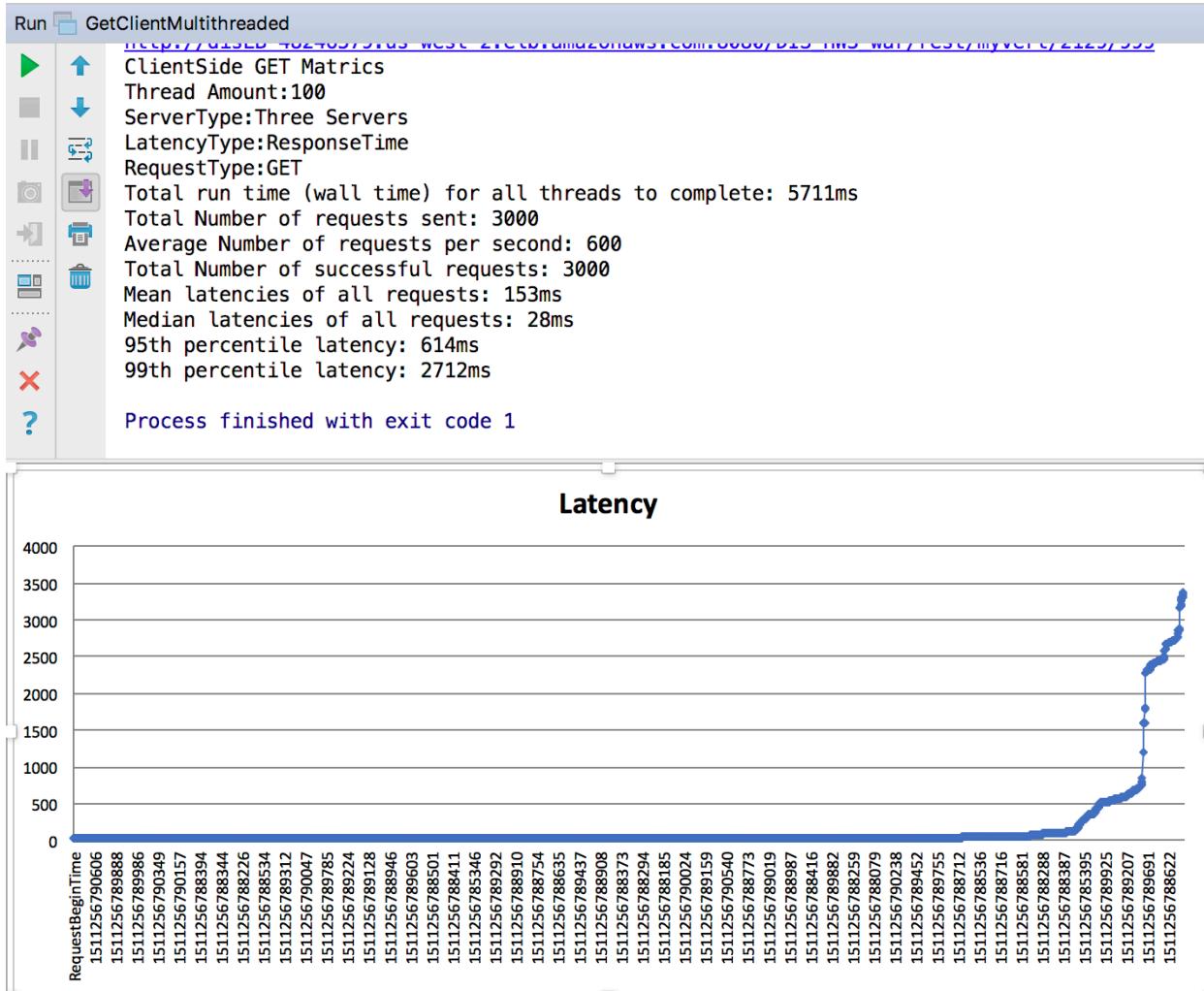
Run PostClientMultithreaded

```
ClientSide POST Metrics
Thread Amount:50
ServerType:Three Servers
LatencyType:ResponseTime
RequestType:POST
Total run time (wall time) for all threads to complete: 60718ms
Total Number of requests sent: 60000
Average Number of requests per second: 1000
Total Number of successful requests: 60000
Mean latencies of all requests: 46ms
Median latencies of all requests: 29ms
95th percentile latency: 106ms
99th percentile latency: 149ms

Process finished with exit code 1
```



- **Three Servers_Day999_GET_Client**



Comment: Post request works same under one server and three servers. The reason is that I use batch to insert. Get request has remarkable improvement.

Step2

Logic:

I monitor the server side latency and store into queue cache. I have a thread keep monitoring the queue. Once there is data, the thread will be responsible for inserting the data into DB. I also create a table to store these latencies. Same logic as error detection.

Before I implement this way, I also tried to put the latencies into SQS. If I continue using SQS, I would create another server keeping polling the data from SQS and then store into DB. However, I found time cost and performance does not have big difference for this assignment. Though database connection might cost longer time and insert DB is a little heavy, each time I could only pull 10 data into SQS, which also has cost. In addition, by using SQS, in order to reduce the impact for server, I might create a new server to monitor SQS and insert data to DB. It takes implementation cost. The most important reason is that I check CPU, there is quite enough capacity to monitor cache and do latency DB interstation. After the condition above, I choose to implement in this way.

If the database is much larger, I might consider to use SQS.

DataBase schema

RawData Table

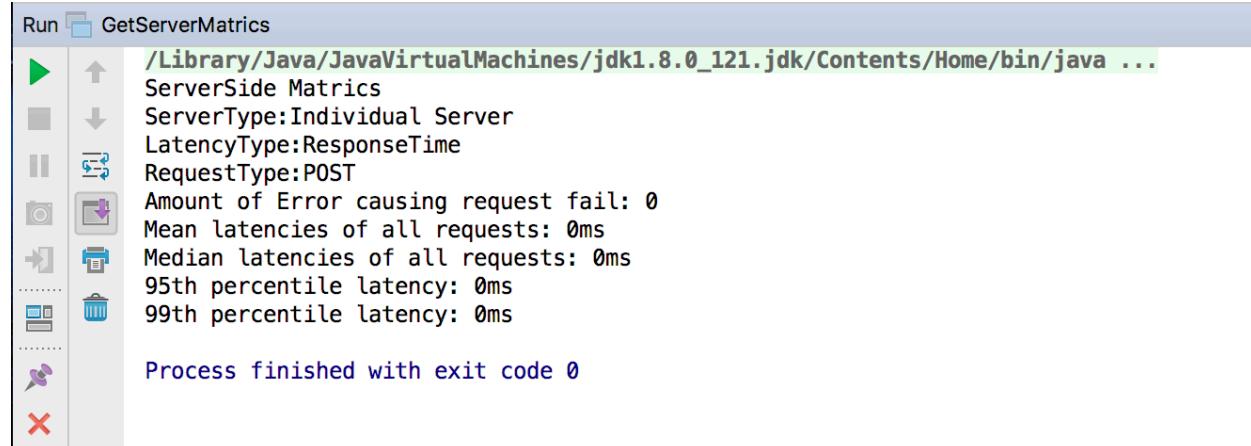
Latency Table: hostName, dayNum, RequestType(POST/GET),

LatencyType(QueryDbTime/ResponseTime, latency)

Error Table: error

Note: Each server run post request with 50 threads and get request 100 threads.

- *Individual Server_Day999_POST_Server*

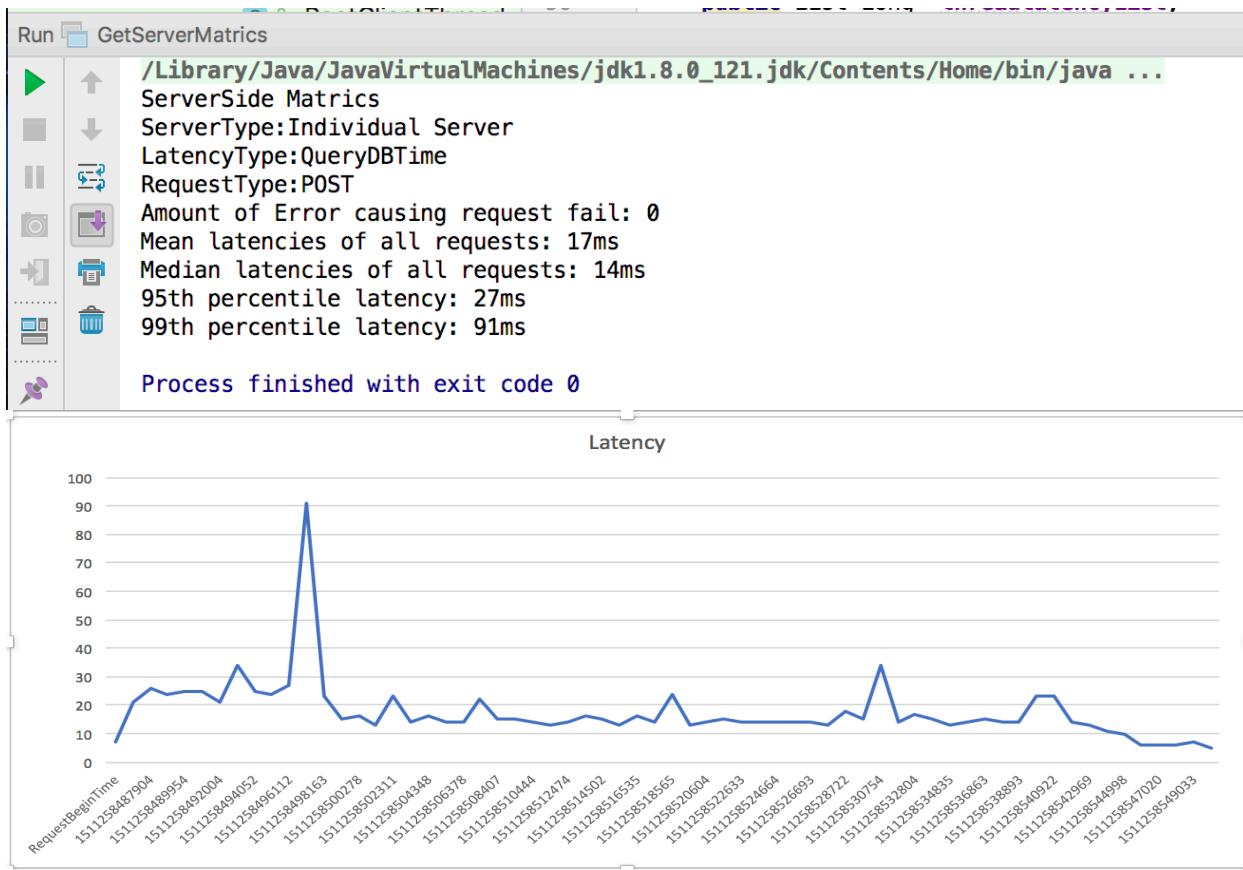


```
Run GetServerMetrics
▶ /Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
ServerSide Metrics
ServerType:Individual Server
LatencyType:ResponseTime
RequestType:POST
Amount of Error causing request fail: 0
Mean latencies of all requests: 0ms
Median latencies of all requests: 0ms
95th percentile latency: 0ms
99th percentile latency: 0ms

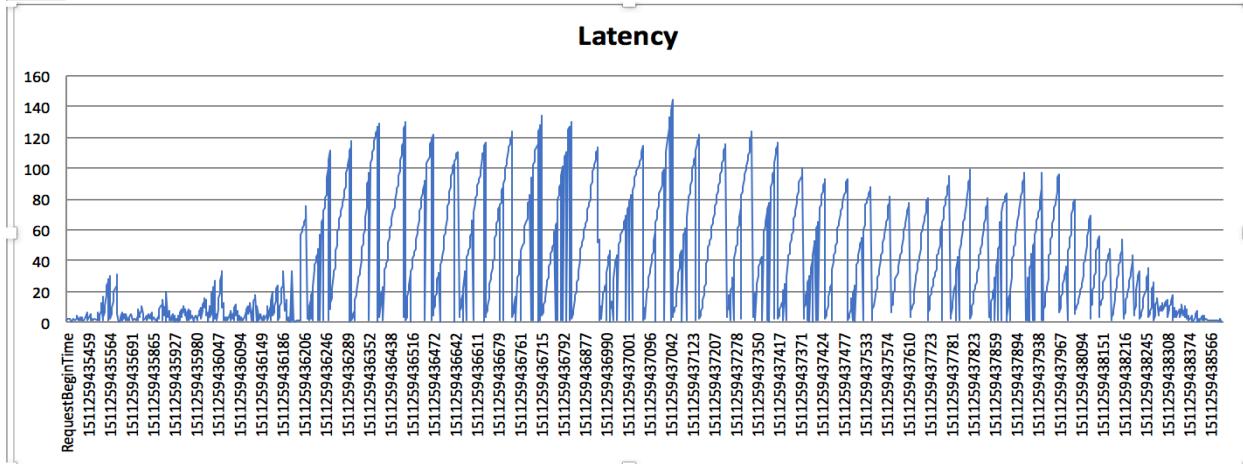
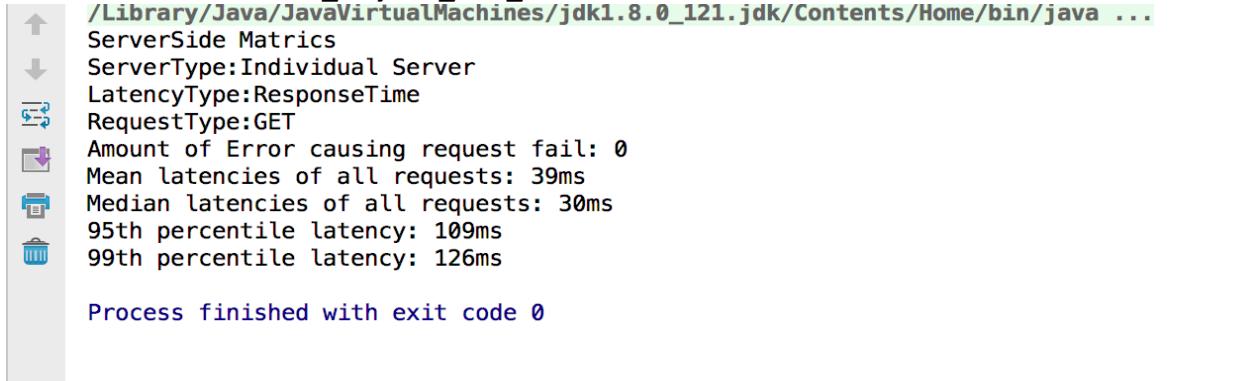
Process finished with exit code 0
```

The reason QueryDB time is longer than the server response time is because I use a queue to store the data to be inserted. There is a listener monitoring the queue and be responsible to insert into DB.

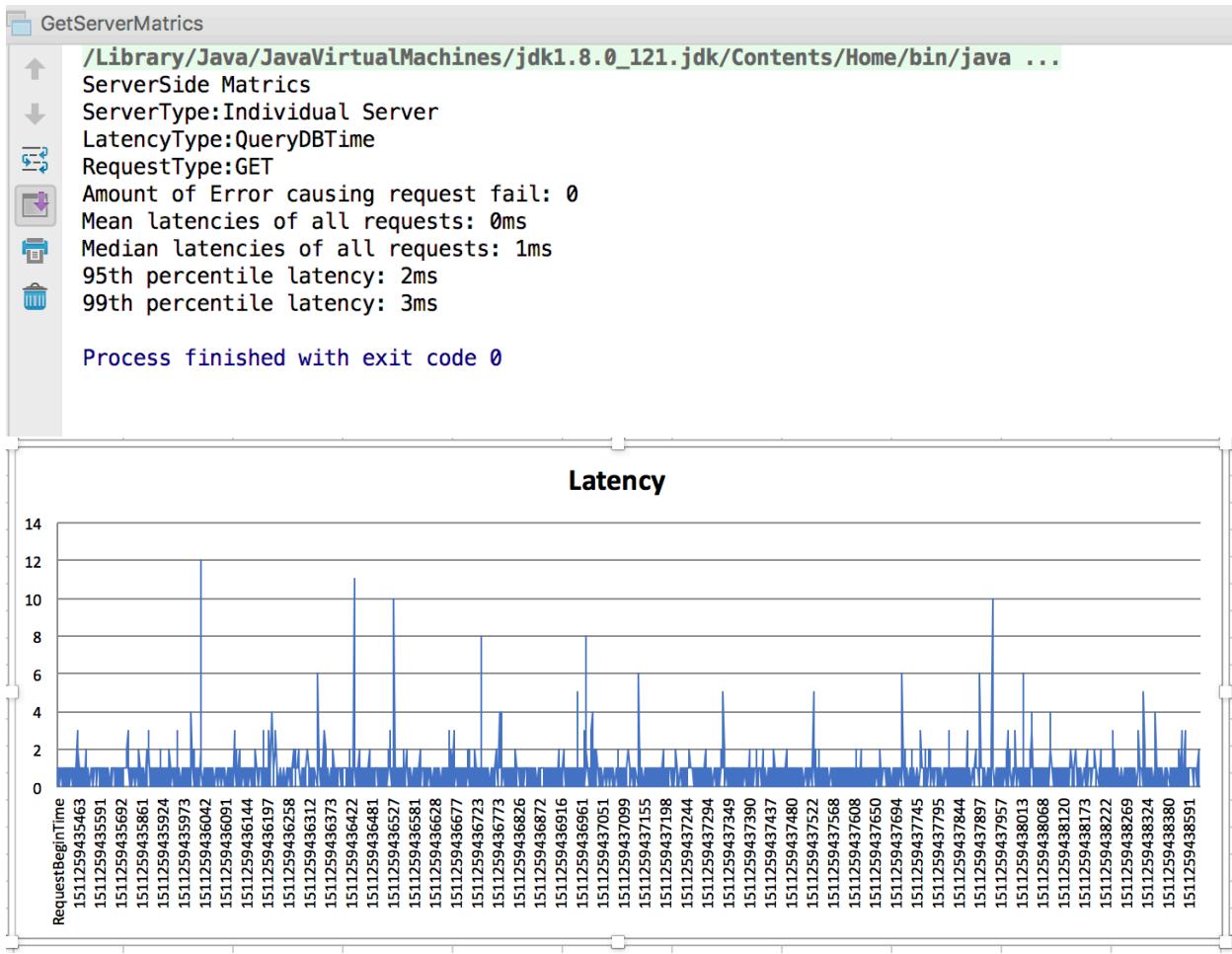
- *Individual Server_Day999_POST_QueryDB*



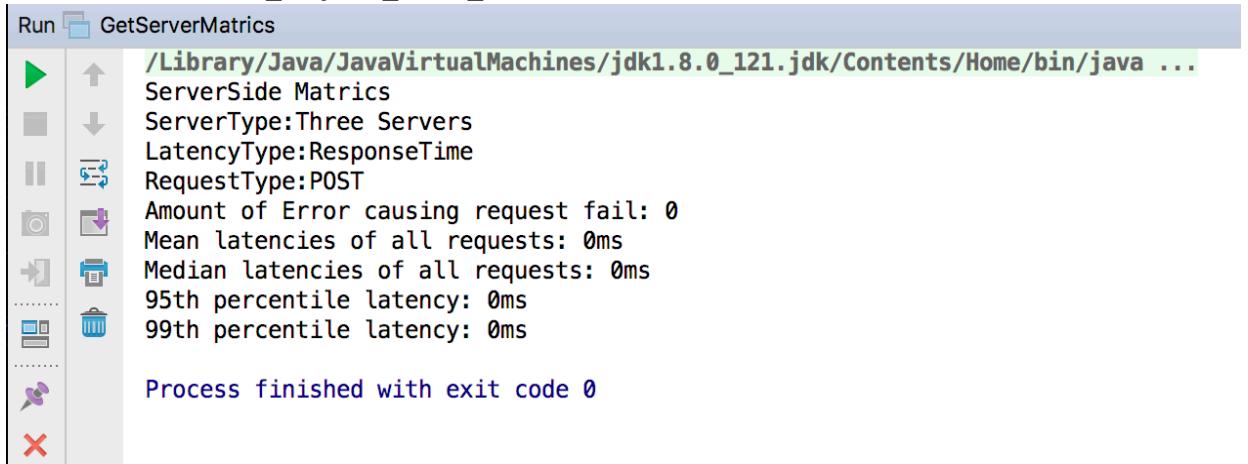
- ***Individual Server_Day999_GET_Server***



- ***Individual Server_Day999_GET_QueryDB***

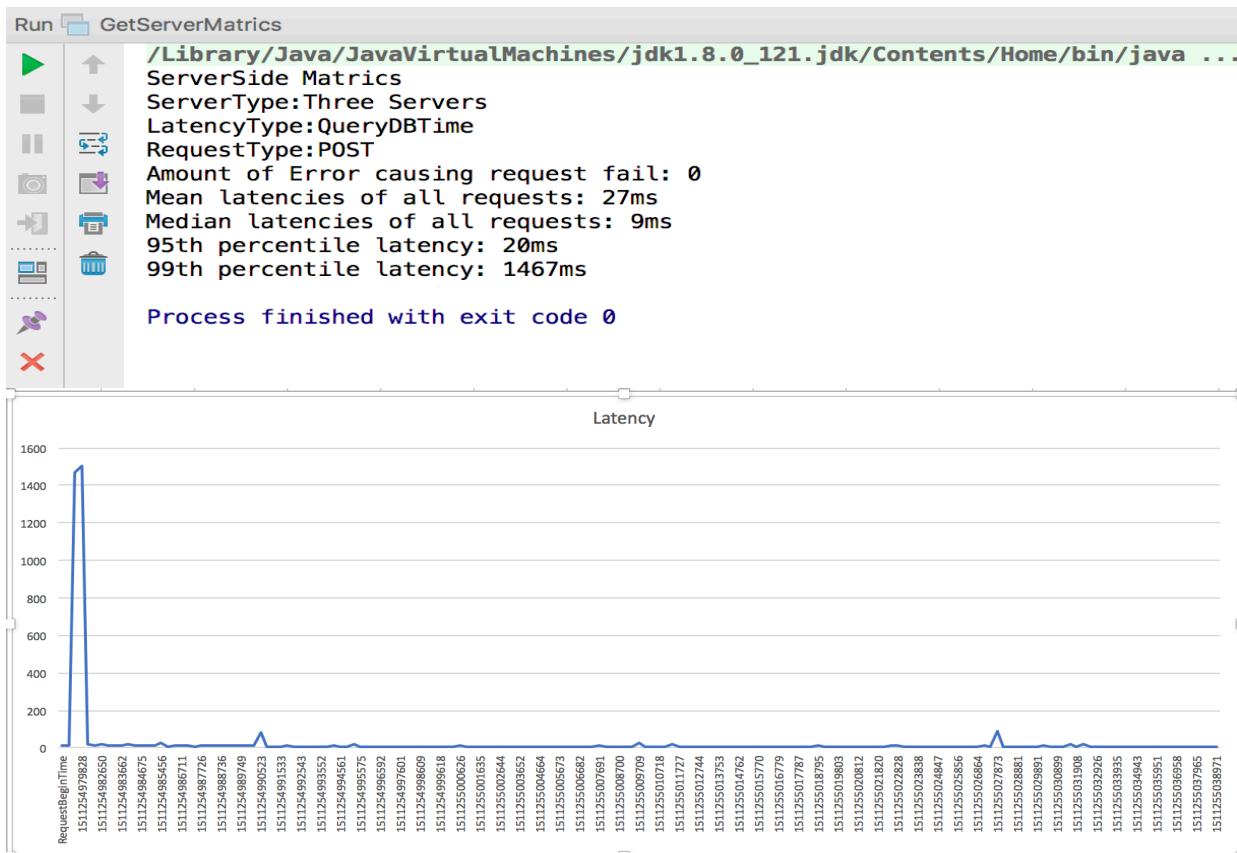


- ***Three Server_Day999_POST_Server***



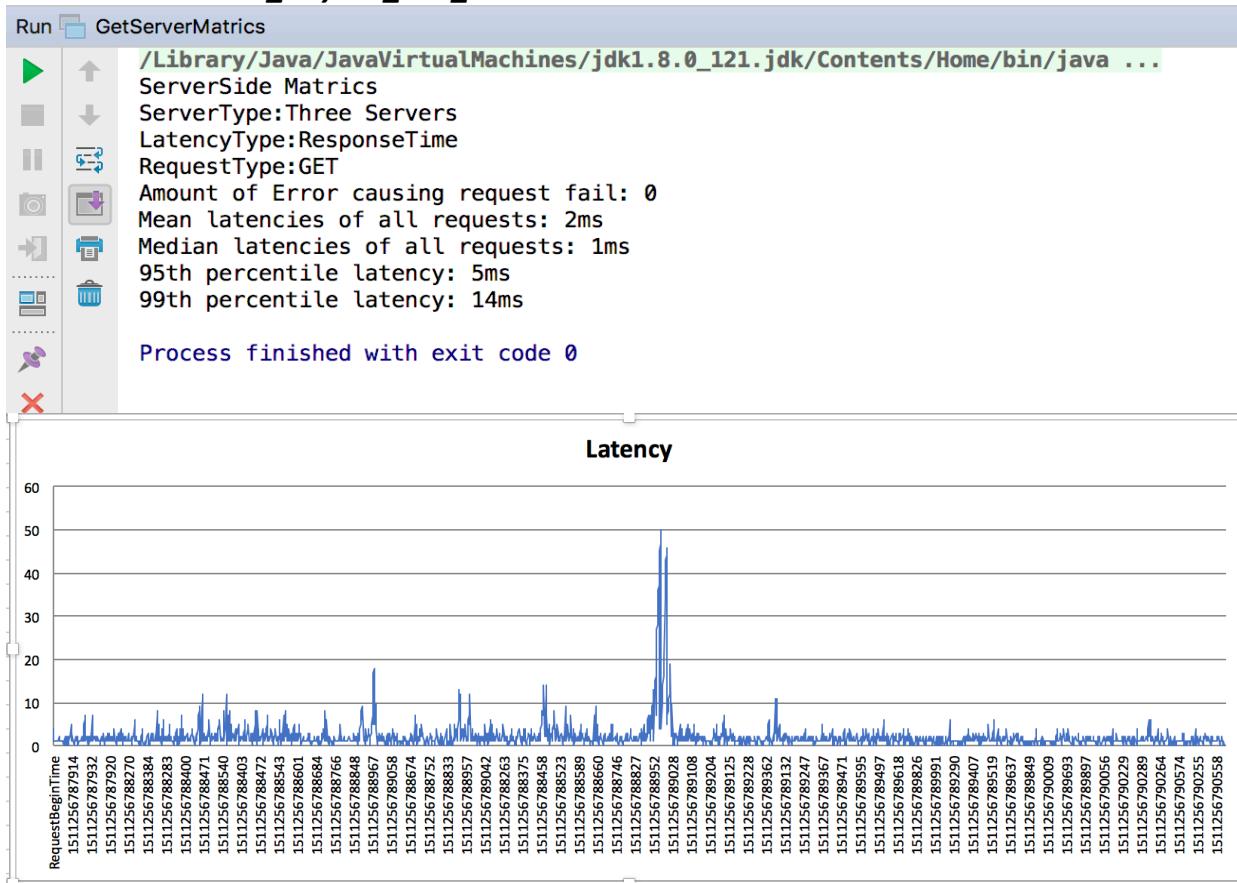
- ***Three Servers_Day999_POST_QueryDB***

Note: for I use batch insert, I do not have much insert DB query



The DB query time is extremely high is because the initialized database connection cost some time or the network latency.

- Three Server_Day999_GET_Server**



- *Three Servers_Day999_GET_QueryDB*

Run GetServerMetrics

ServerSide Metrics
ServerType:Three Servers
LatencyType:QueryDBTime
RequestType:GET

Amount of Error causing request fail: 0
Mean latencies of all requests: 0ms
Median latencies of all requests: 1ms
95th percentile latency: 1ms
99th percentile latency: 3ms

Process finished with exit code 0

Latency

The chart displays the distribution of request latencies. The x-axis represents the RequestBeginTime, and the y-axis represents the latency in milliseconds. The data shows a high frequency of low-latency requests (mostly below 5ms) and a single outlier at approximately 18ms.

RequestBeginTime	Latency (ms)
1511256787518	0.5
1511256787563	6.5
1511256788151	1.5
1511256788149	1.5
1511256788291	1.5
1511256788333	1.5
1511256788309	1.5
1511256788422	1.5
1511256788462	1.5
1511256788496	1.5
1511256788564	1.5
1511256788633	1.5
1511256788708	1.5
1511256788755	1.5
1511256788653	1.5
1511256788734	1.5
1511256788717	1.5
1511256788953	1.5
1511256789035	1.5
1511256788653	1.5
1511256788735	1.5
1511256788462	1.5
1511256788755	1.5
1511256788757	1.5
1511256788759	1.5
1511256788905	1.5
1511256788943	1.5
1511256788679	1.5
1511256788764	1.5
1511256788448	1.5
1511256789055	1.5
1511256789063	1.5
1511256789140	1.5
1511256789078	1.5
1511256789166	1.5
1511256789292	1.5
1511256789103	1.5
1511256789196	1.5
1511256789255	1.5
1511256789441	1.5
1511256789554	1.5
1511256789472	1.5
1511256789597	1.5
1511256789797	1.5
1511256789669	1.5
1511256790028	1.5
1511256789731	1.5
1511256789523	1.5
1511256790099	1.5
1511256790142	1.5
151125679022	1.5
1511256790391	1.5
1511256790503	1.5
1511256790394	1.5

Step3

Day3

- *Individual Server_Day3_POST_Client*

Run PostClientMultithreaded

Clientside POST Metrics

Thread Amount:50

ServerType:Individual Server

LatencyType:ResponseTime

RequestType:POST

Total run time (wall time) for all threads to complete: 207231ms

Total Number of requests sent: 200000

Average Number of requests per second: 966

Total Number of successful requests: 200000

Mean latencies of all requests: 47ms

Median latencies of all requests: 32ms

95th percentile latency: 50ms

99th percentile latency: 465ms

Process finished with exit code 1

Latency

The graph displays the distribution of response times for 200,000 requests. The x-axis represents the request index, and the y-axis represents the latency in milliseconds. The distribution is highly skewed, with the vast majority of requests (around 95%) taking less than 50ms. A single outlier request at index 199,942 takes approximately 465ms, which is the 99th percentile latency mentioned in the metrics. The y-axis has major ticks at 0, 1000, 2000, 3000, 4000, 5000, and 6000.

RequestIndex	Latency (ms)
0	~10
1	~15
2	~20
3	~25
4	~30
5	~35
6	~40
7	~45
8	~50
9	~55
10	~60
11	~65
12	~70
13	~75
14	~80
15	~85
16	~90
17	~95
18	~100
19	~105
20	~110
21	~115
22	~120
23	~125
24	~130
25	~135
26	~140
27	~145
28	~150
29	~155
30	~160
31	~165
32	~170
33	~175
34	~180
35	~185
36	~190
37	~195
38	~200
39	~205
40	~210
41	~215
42	~220
43	~225
44	~230
45	~235
46	~240
47	~245
48	~250
49	~255
50	~260
51	~265
52	~270
53	~275
54	~280
55	~285
56	~290
57	~295
58	~300
59	~305
60	~310
61	~315
62	~320
63	~325
64	~330
65	~335
66	~340
67	~345
68	~350
69	~355
70	~360
71	~365
72	~370
73	~375
74	~380
75	~385
76	~390
77	~395
78	~400
79	~405
80	~410
81	~415
82	~420
83	~425
84	~430
85	~435
86	~440
87	~445
88	~450
89	~455
90	~460
91	~465
92	~470
93	~475
94	~480
95	~485
96	~490
97	~495
98	~500
99	~505
100	~510
101	~515
102	~520
103	~525
104	~530
105	~535
106	~540
107	~545
108	~550
109	~555
110	~560
111	~565
112	~570
113	~575
114	~580
115	~585
116	~590
117	~595
118	~600
119	~605
120	~610
121	~615
122	~620
123	~625
124	~630
125	~635
126	~640
127	~645
128	~650
129	~655
130	~660
131	~665
132	~670
133	~675
134	~680
135	~685
136	~690
137	~695
138	~700
139	~705
140	~710
141	~715
142	~720
143	~725
144	~730
145	~735
146	~740
147	~745
148	~750
149	~755
150	~760
151	~765
152	~770
153	~775
154	~780
155	~785
156	~790
157	~795
158	~800
159	~805
160	~810
161	~815
162	~820
163	~825
164	~830
165	~835
166	~840
167	~845
168	~850
169	~855
170	~860
171	~865
172	~870
173	~875
174	~880
175	~885
176	~890
177	~895
178	~900
179	~905
180	~910
181	~915
182	~920
183	~925
184	~930
185	~935
186	~940
187	~945
188	~950
189	~955
190	~960
191	~965
192	~970
193	~975
194	~980
195	~985
196	~990
197	~995
198	~999
199	~999
200	~999
201	~999
202	~999
203	~999
204	~999
205	~999
206	~999
207	~999
208	~999
209	~999
210	~999
211	~999
212	~999
213	~999
214	~999
215	~999
216	~999
217	~999
218	~999
219	~999
220	~999
221	~999
222	~999
223	~999
224	~999
225	~999
226	~999
227	~999
228	~999
229	~999
230	~999
231	~999
232	~999
233	~999
234	~999
235	~999
236	~999
237	~999
238	~999
239	~999
240	~999
241	~999
242	~999
243	~999
244	~999
245	~999
246	~999
247	~999
248	~999
249	~999
250	~999

- *Individual Server_ Day3_POST_Server*

```
GetServerMetrics
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
ServerSide Metrics
ServerType:Individual Server
LatencyType:ResponseTime
RequestType:POST
Amount of Error causing request fail: 0
Mean latencies of all requests: 0ms
Median latencies of all requests: 0ms
95th percentile latency: 0ms
99th percentile latency: 0ms

Process finished with exit code 0
```

- *Individual Servers_ Day3_POST_QueryDB*

GetServerMetrics

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
```

↑ ServerSide Metrics
 ↓ ServerType:Individual Server
 LatencyType:QueryDBTime
 RequestType:POST
 Amount of Error causing request fail: 0
 Mean latencies of all requests: 16ms
 Median latencies of all requests: 15ms
 95th percentile latency: 26ms
 99th percentile latency: 60ms

Process finished with exit code 0

- Individual Server_Day3_GET_Client**

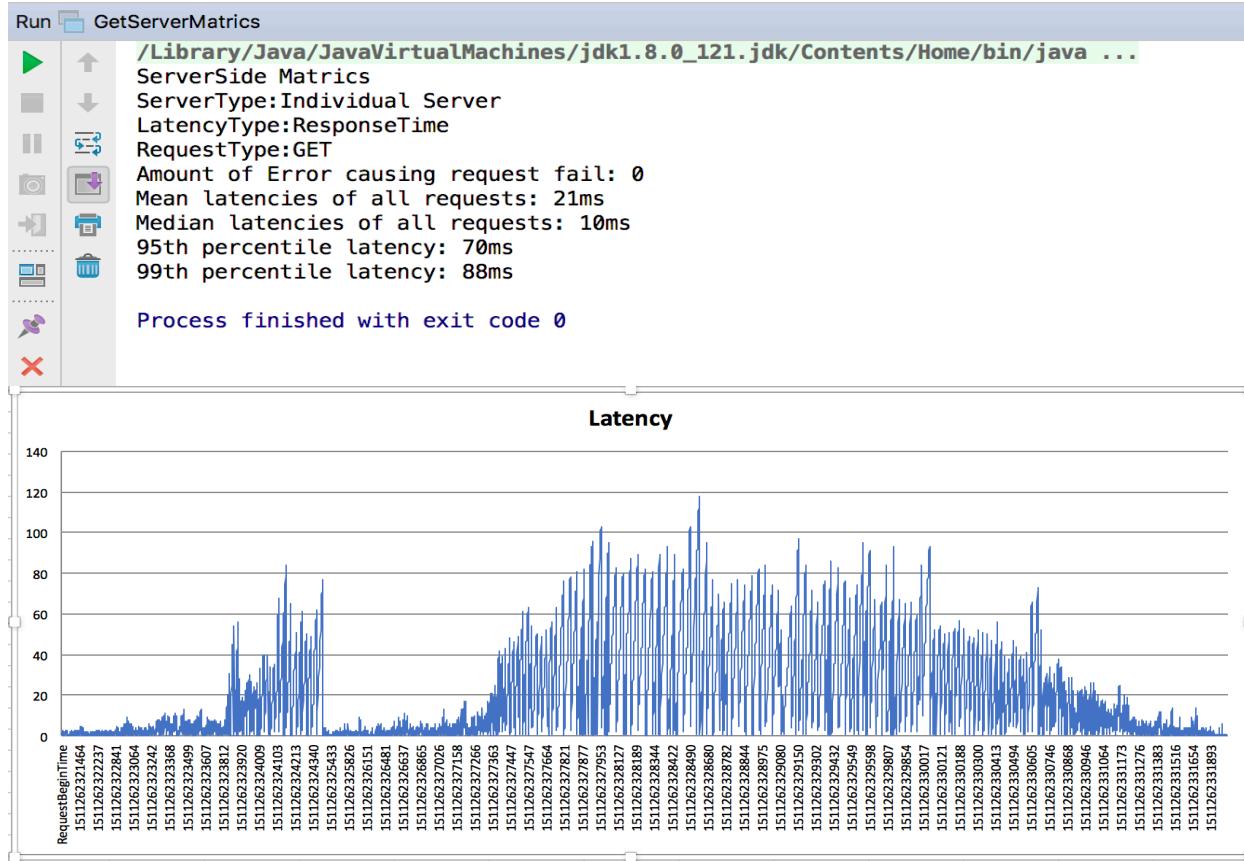
GetClientMultithreaded

```
http://ec2-54-186-64-113.us-west-2.compute.amazonaws.com:8080/D15_HW3_war/rest/myvert/5899/3
```

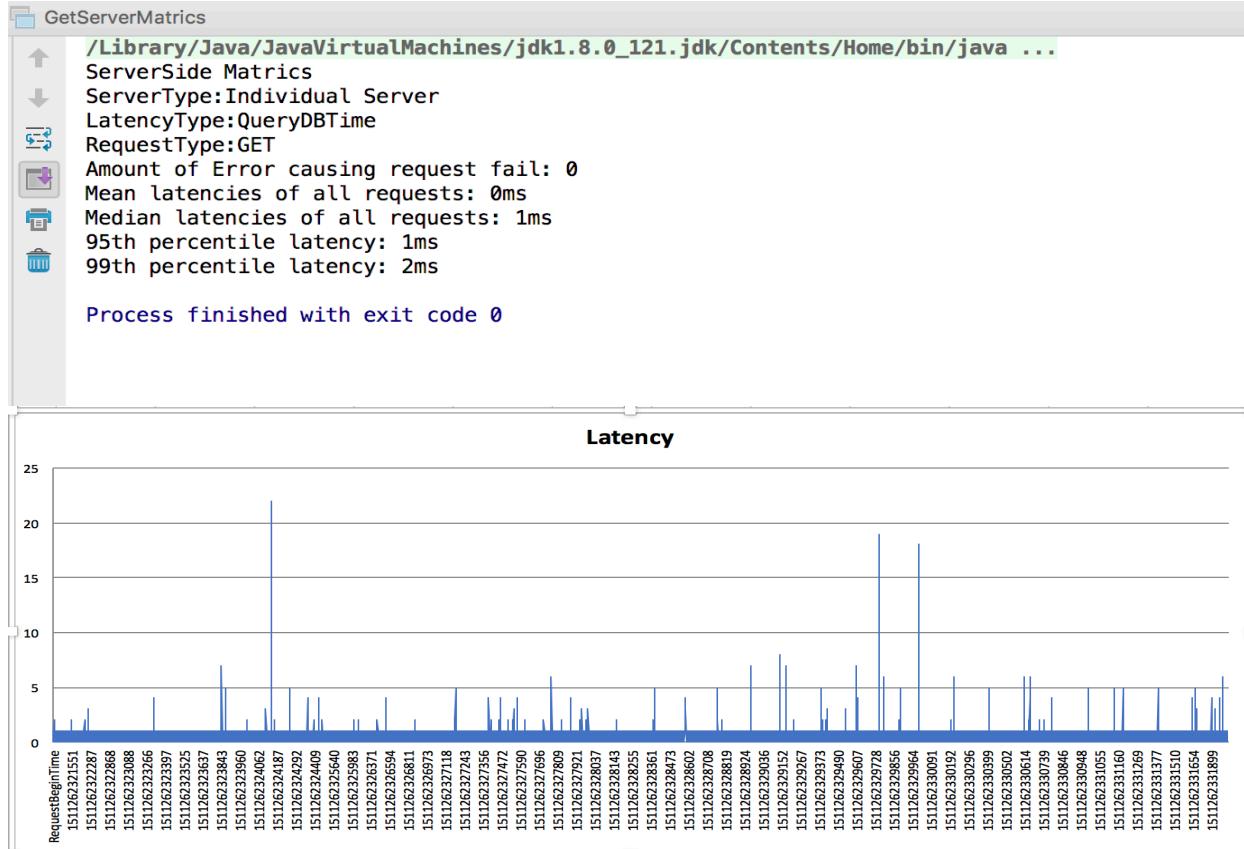
↑ ClientSide GET Metrics
 ↓ Thread Amount:100
 ServerType:Individual Server
 RequestType:GET
 Total run time (wall time) for all threads to complete: 18698ms
 Total Number of requests sent: 10000
 Average Number of requests per second: 555
 Total Number of successful requests: 10000
 Mean latencies of all requests: 139ms
 Median latencies of all requests: 40ms
 95th percentile latency: 334ms
 99th percentile latency: 3827ms

Process finished with exit code 1

- *Individual Server_Day3_GET_Server*



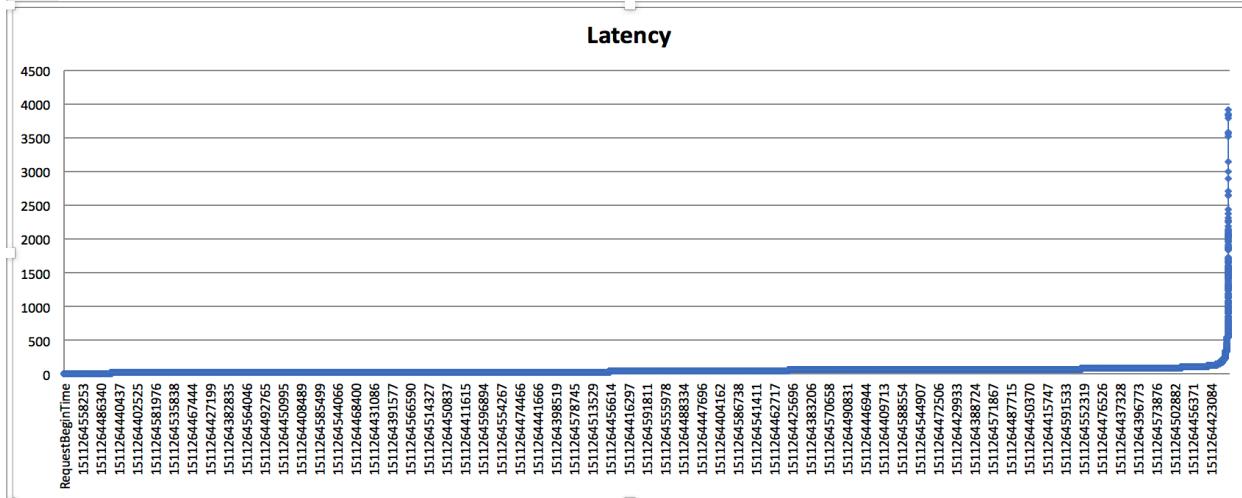
- *Individual Servers_Day3_GET_QueryDB*



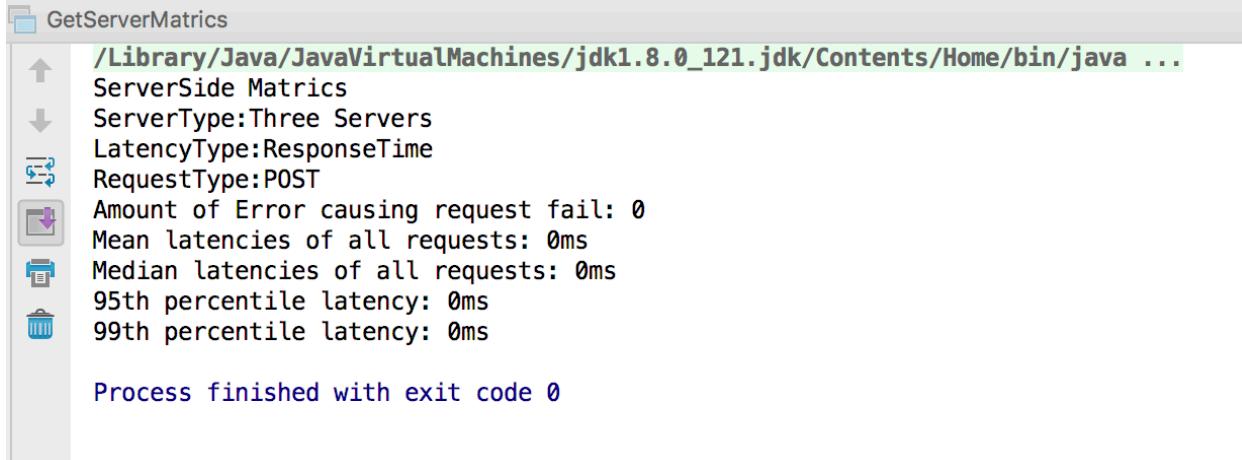
- ***Three Server Day3 POST Client***

each request latency: 28
 ClientSide POST Metrics
 Thread Amount:50
 ServerType:Three Servers
 RequestType:POST
 Total run time (wall time) for all threads to complete: 201691ms
 Total Number of requests sent: 200000
 Average Number of requests per second: 995
 Total Number of successful requests: 200000
 Mean latencies of all requests: 49ms
 Median latencies of all requests: 45ms
 95th percentile latency: 94ms
 99th percentile latency: 158ms

Process finished with exit code 1



- ***Three Server Day3 POST Server***



- ***Three Servers Day3 POST QueryDB***

GetServerMetrics

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
↑ ServerSide Metrics
↓ ServerType:Three Servers
←→ LatencyType:QueryDBTime
RequestType:POST
Amount of Error causing request fail: 0
Mean latencies of all requests: 11ms
Median latencies of all requests: 9ms
95th percentile latency: 26ms
99th percentile latency: 61ms
```

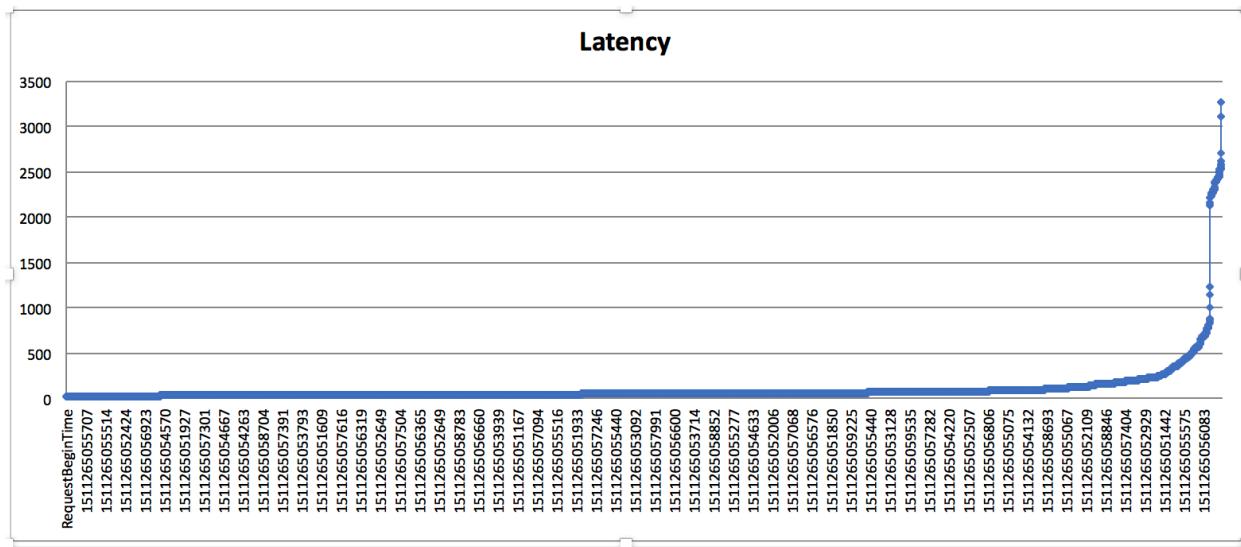
Process finished with exit code 0

- ***Three Server_ Day3_ GET_ Client***

GetClientMultithreaded

```
http://d1slb-4824b579.us-west-2.elb.amazonaws.com:8080/D1S_HW3_war/rest/myvert/1/99/3
↑ ClientSide GET Metrics
↓ Thread Amount:100
ServerType:Three Servers
RequestType:GET
Total run time (wall time) for all threads to complete: 11359ms
Total Number of requests sent: 10000
Average Number of requests per second: 909
Total Number of successful requests: 10000
Mean latencies of all requests: 102ms
Median latencies of all requests: 50ms
95th percentile latency: 272ms
99th percentile latency: 2132ms
```

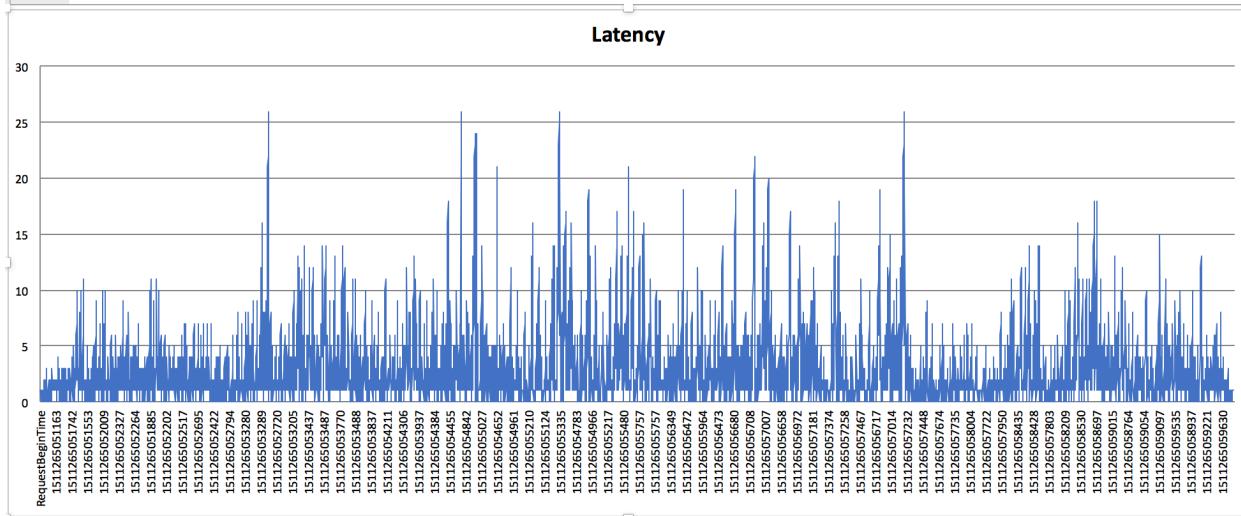
Process finished with exit code 1



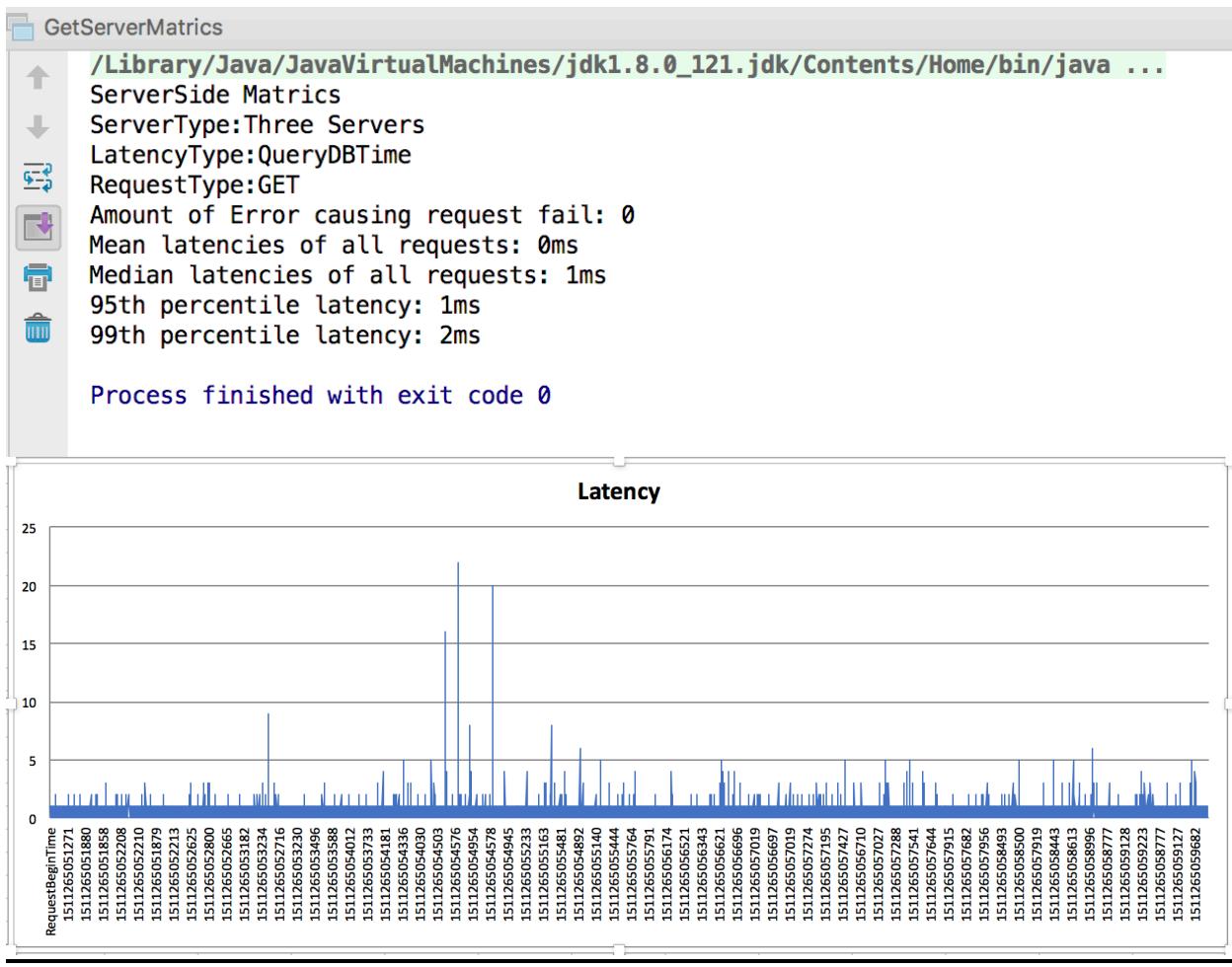
- ### **• *Three Server_ Day3_ GET_Server***

```
GetServerMetrics
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
ServerSide Metrics
ServerType:Three Servers
LatencyType:ResponseTime
RequestType:GET
Amount of Error causing request fail: 0
Mean latencies of all requests: 2ms
Median latencies of all requests: 2ms
95th percentile latency: 10ms
99th percentile latency: 15ms

Process finished with exit code 0
```

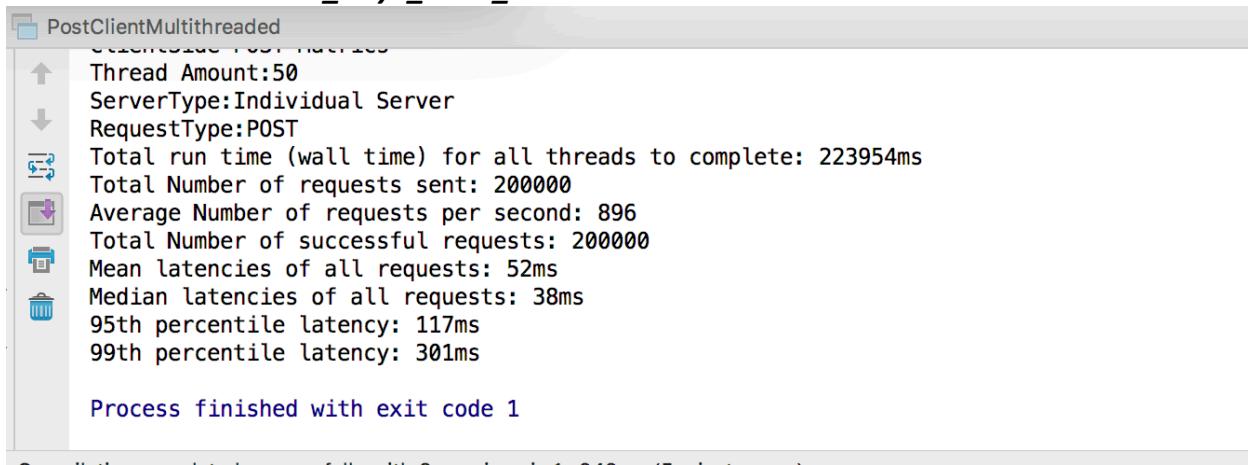


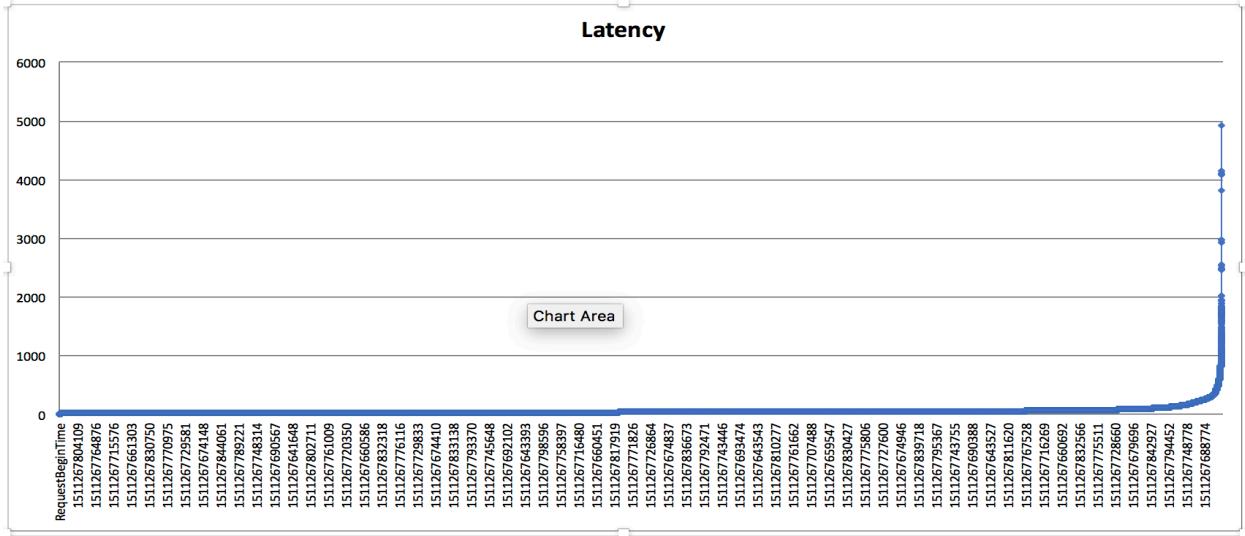
- *Three Servers_ Day3_ GET_QueryDB*



Day4

- *Individual Server_Day4_POST_Client*





- ***Individual Server_Day4_POST_Server***

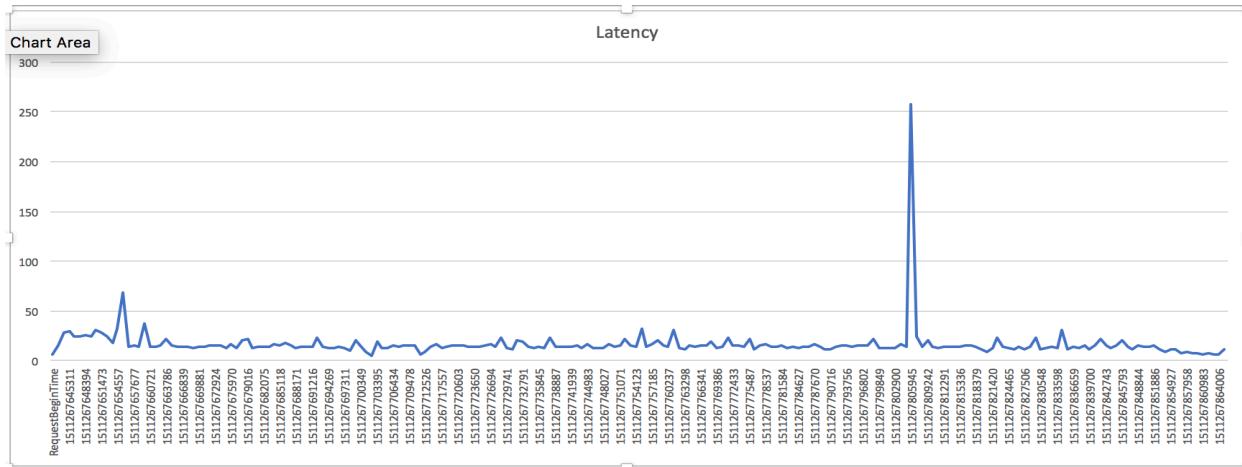
```
GetServerMetrics
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
ServerSide Metrics
ServerType:Individual Server
LatencyType:ResponseTime
RequestType:POST
Amount of Error causing request fail: 0
Mean latencies of all requests: 0ms
Median latencies of all requests: 0ms
95th percentile latency: 0ms
99th percentile latency: 0ms

Process finished with exit code 0
```

- ***Individual Servers_Day4_POST_QueryDB***

```
GetServerMetrics
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
ServerSide Metrics
ServerType:Individual Server
LatencyType:QueryDBTime
RequestType:POST
Amount of Error causing request fail: 0
Mean latencies of all requests: 16ms
Median latencies of all requests: 14ms
95th percentile latency: 28ms
99th percentile latency: 37ms

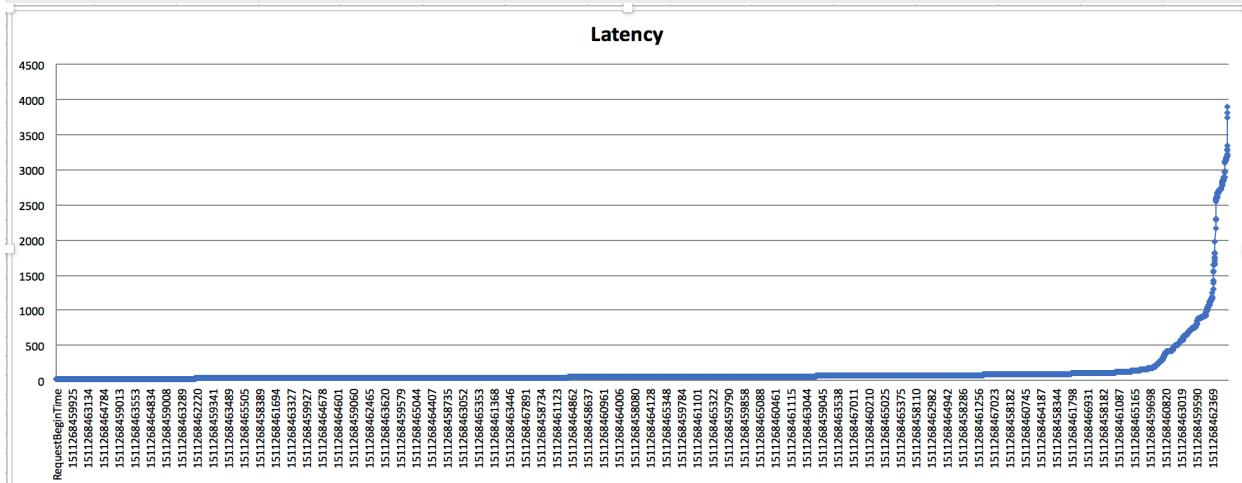
Process finished with exit code 0
```



- *Individual Server_ Day4_GET_Client*

```
GetClientMultithreaded
http://127.0.0.1:22555/test/2/comparegameandaws.com/6000/B12-TMS-WAR/TEST/myapp-0.0.1/
↑ ClientSide GET Metrics
Thread Amount:100
↓ ServerType:Individual Server
RequestType:GET
Total run time (wall time) for all threads to complete: 18493ms
Total Number of requests sent: 10000
Average Number of requests per second: 555
Total Number of successful requests: 10000
Mean latencies of all requests: 118ms
Median latencies of all requests: 52ms
95th percentile latency: 422ms
99th percentile latency: 2585ms

Process finished with exit code 1
```



- *Individual Server_ Day4_GET_Server*

GetServerMetrics

/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...

↑
↓
↶
↷
↶↷
↶↶↷↷
↶↶↷↷

ServerSide Metrics
ServerType:Individual Server
LatencyType:ResponseTime
RequestType:GET
Amount of Error causing request fail: 0
Mean latencies of all requests: 16ms
Median latencies of all requests: 11ms
95th percentile latency: 50ms
99th percentile latency: 66ms

Process finished with exit code 0

Latency

RequestBeginTime
1511268456201
1511268456435
1511268456667
1511268456898
1511268456918
1511268457018
1511268457287
1511268457563
1511268457809
1511268458032
1511268458183
1511268458291
1511268458393
1511268458494
1511268458628
1511268458715
1511268458728
1511268458808
1511268458866
1511268458979
1511268459079
1511268459201
1511268459297
1511268459405
1511268459526
1511268459626
1511268459729
1511268459833
1511268459973
1511268460095
1511268460258
1511268460349
1511268460542
1511268460636
1511268460779
1511268460796
1511268461081
1511268461177
1511268461369
1511268461717
1511268461958
1511268462098
1511268462118
1511268462311
1511268462911
1511268463104
1511268463208
1511268463319
1511268464190
1511268463354
1511268463375
1511268464460
1511268464593
1511268464745
1511268464891
1511268465047
1511268465207
1511268465375

- *Individual Servers_ Day4_GET_QueryDB*

GetServerMetrics

Screen Shot 2017-11-21 at 4.54.22 AM es/jdk1.8.0_121.jdk/Contents/Home/bin/java ...

Serverside Metrics

ServerType:Individual Server

LatencyType:QueryDBTime

RequestType:GET

Amount of Error causing request fail: 0

Mean latencies of all requests: 0ms

Median latencies of all requests: 1ms

95th percentile latency: 1ms

99th percentile latency: 2ms

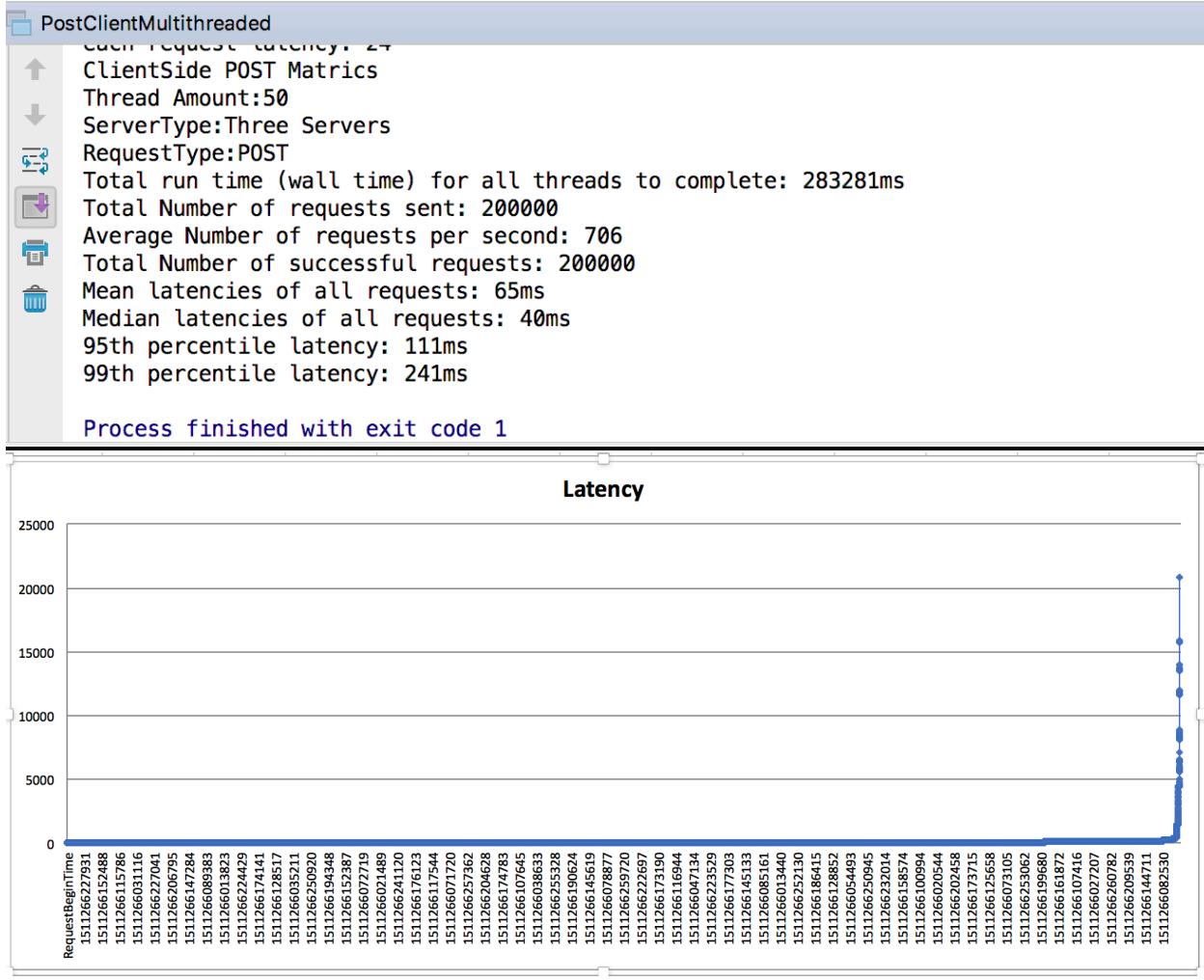
Process finished with exit code 0

Latency

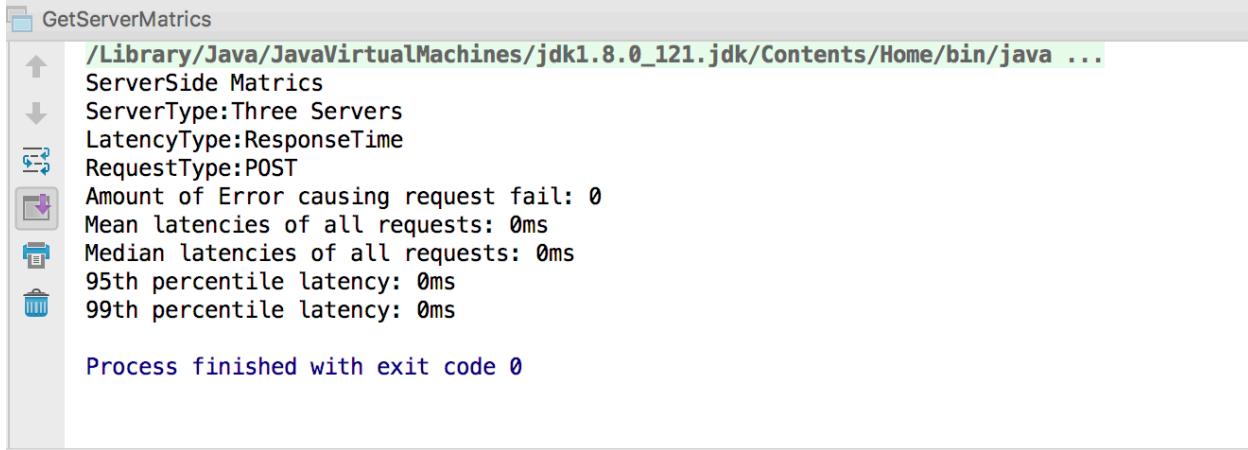
The histogram displays the following approximate data points:

Latency (ms)	Count
0.5	10
1.0	100
2.0	10
3.0	10
4.0	10
5.0	10
6.0	10
7.0	10
8.0	10
9.0	10
10.0	10
11.0	10
12.0	10
13.0	10
14.0	10

- ***Three Server_Day4_POST_Client***



- ***Three Server_Day4_POST_Server***



- ***Three Servers_Day4_POST_QueryDB***

```

GetServerMetrics
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
↑ ServerSide Metrics
↓ ServerType:Three Servers
⠄⠄ LatencyType:QueryDBTime
⠄⠄ RequestType:POST
Amount of Error causing request fail: 0
Mean latencies of all requests: 8ms
Median latencies of all requests: 8ms
95th percentile latency: 16ms
99th percentile latency: 26ms

Process finished with exit code 0

```

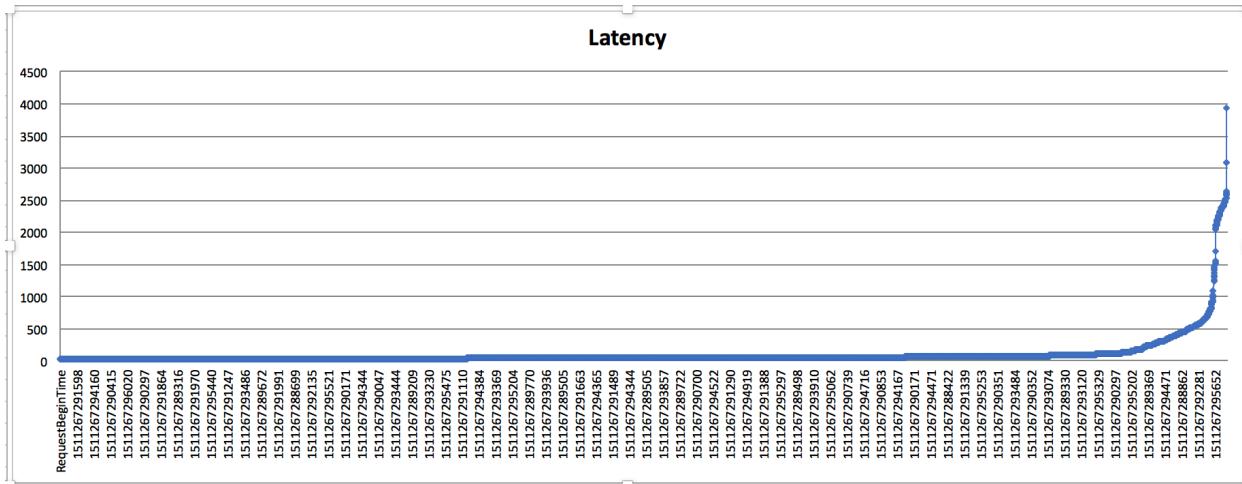
- ***Three Server_ Day4_GET_Client***

```

GetClientMultithreaded
HTTP://127.0.0.1:8080/test/latency?count=10000&type=GET
↑ ClientSide GET Metrics
↓ Thread Amount:100
ServerType:Three Servers
RequestType:GET
Total run time (wall time) for all threads to complete: 13094ms
Total Number of requests sent: 10000
Average Number of requests per second: 769
Total Number of successful requests: 10000
Mean latencies of all requests: 97ms
Median latencies of all requests: 43ms
95th percentile latency: 351ms
99th percentile latency: 2051ms

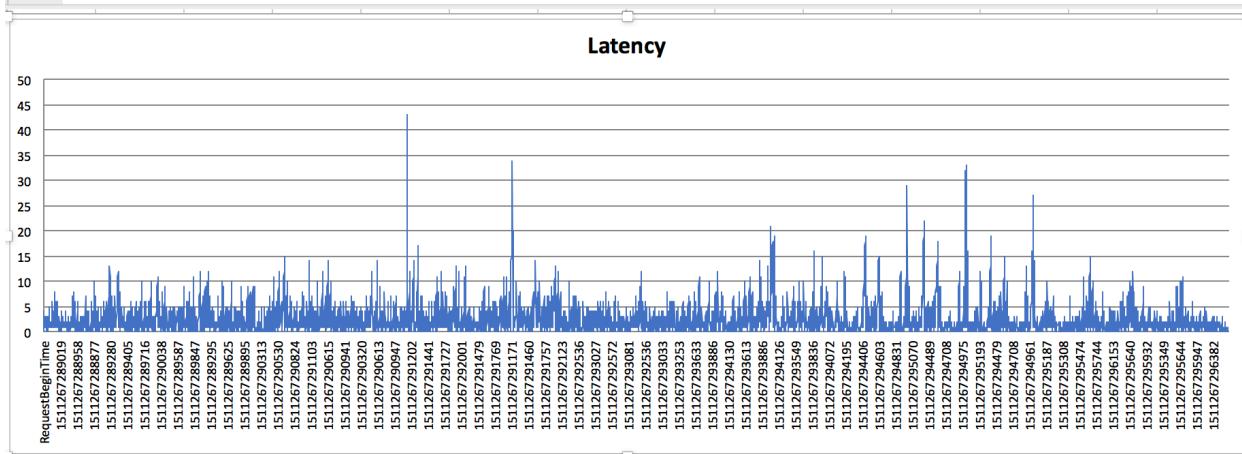
Process finished with exit code 1

```



• *Three Server_Day4_GET_Server*

```
GetServerMetrics  
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...  
ServerSide Metrics  
ServerType:Three Servers  
LatencyType:ResponseTime  
RequestType:GET  
Amount of Error causing request fail: 0  
Mean latencies of all requests: 2ms  
Median latencies of all requests: 1ms  
95th percentile latency: 8ms  
99th percentile latency: 14ms  
  
Process finished with exit code 0
```



- *Three Servers_Day4_GET_QueryDB*

GetServerMetrics

```

↑ /Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
↓ ServerSide Metrics
ServerType:Three Servers
LatencyType:QueryDBTime
RequestType:GET
Amount of Error causing request fail: 0
Mean latencies of all requests: 0ms
Median latencies of all requests: 1ms
95th percentile latency: 1ms
99th percentile latency: 2ms

Process finished with exit code 0

```

Latency

The histogram displays the latency distribution for the 'GetServerMetrics' test. The x-axis represents the RequestBeginTime, with ticks every 10 units. The y-axis represents the latency in milliseconds, ranging from 0 to 16. The distribution is characterized by a large number of low-latency requests (mostly below 5ms) and several high-latency outliers, notably around 10ms, 12ms, and 14ms.

Day5

Individual Server_Day5_POST_Client

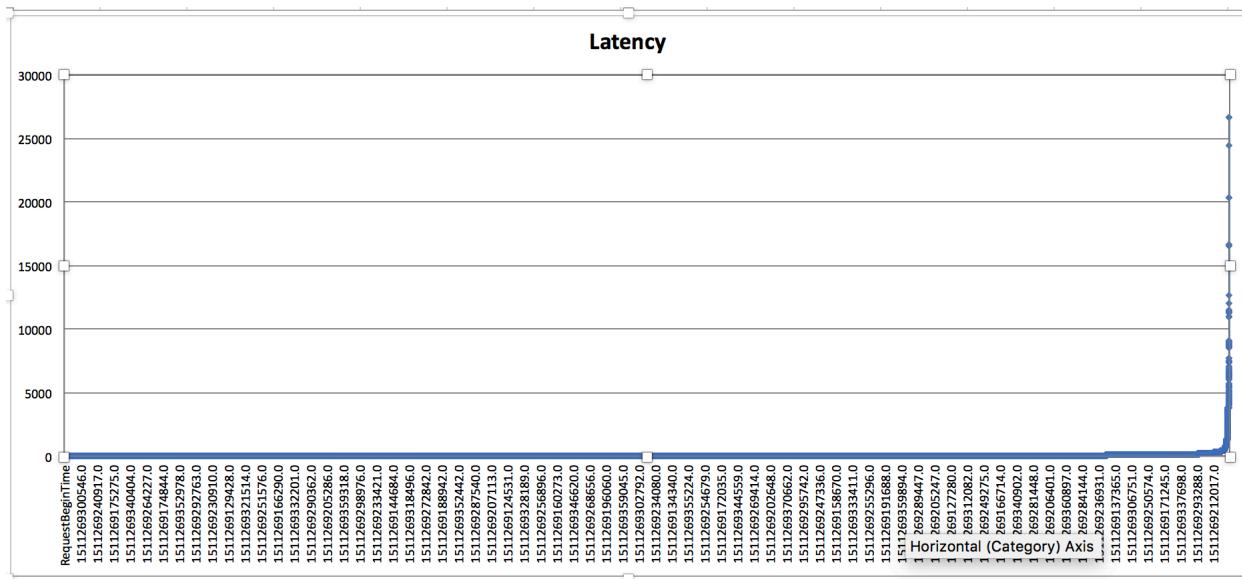
PostClientMultithreaded

```

↑ Thread Amount:50
↓ ServerType:Individual Server
RequestType:POST
Total run time (wall time) for all threads to complete: 275549ms
Total Number of requests sent: 200000
Average Number of requests per second: 727
Total Number of successful requests: 199998
Mean latencies of all requests: 61ms
Median latencies of all requests: 38ms
95th percentile latency: 110ms
99th percentile latency: 329ms

Process finished with exit code 1

```



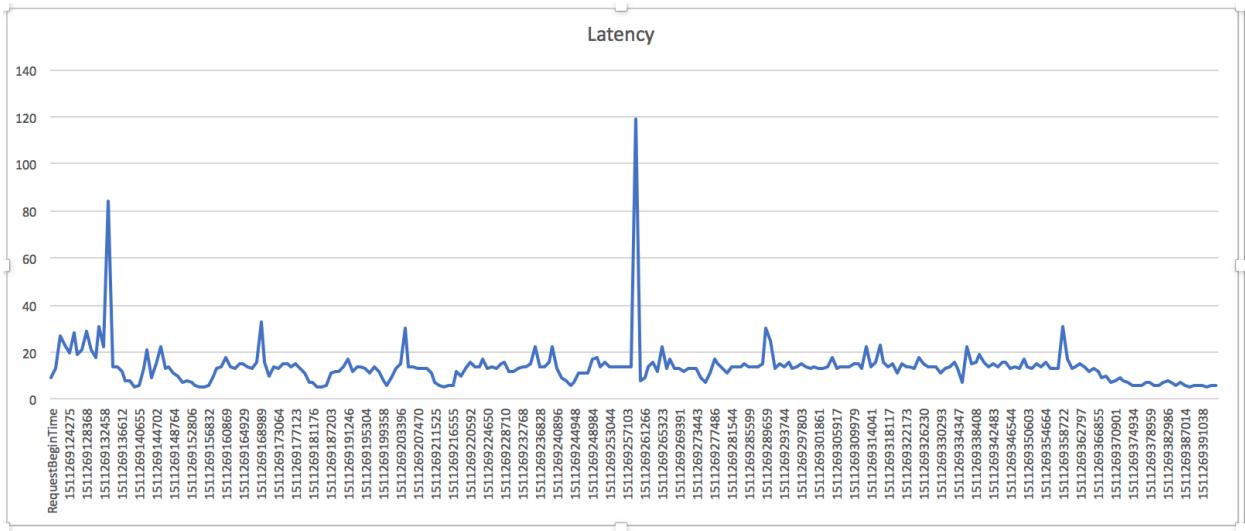
- ### **• *Individual Server_Day5_POST_Server***

```
GetServerMetrics
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
ServerSide Metrics
ServerType:Individual Server
LatencyType:ResponseTime
RequestType:POST
Amount of Error causing request fail: 0
Mean latencies of all requests: 0ms
Median latencies of all requests: 0ms
95th percentile latency: 0ms
99th percentile latency: 0ms

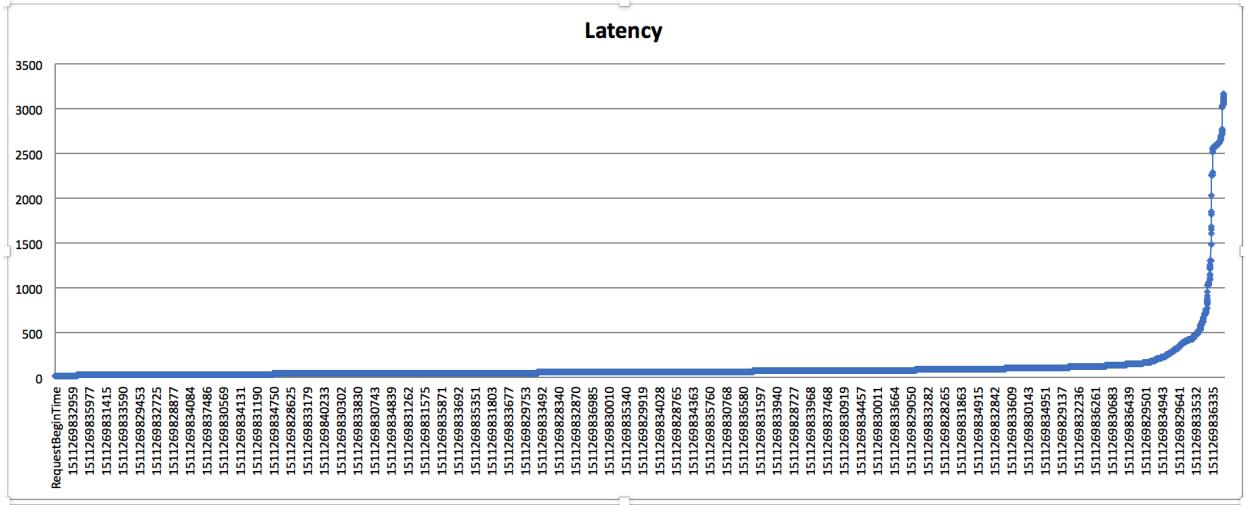
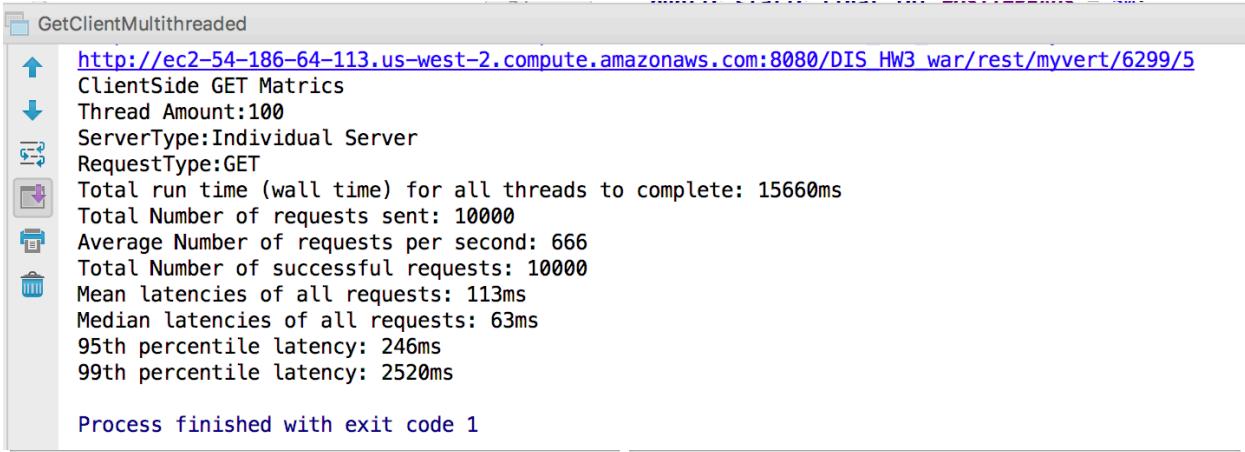
Process finished with exit code 0
```

- *Individual Servers_ Day5_POST_QueryDB*

```
GetServerMetrics  
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...  
ServerSide Metrics  
ServerType:Individual Server  
LatencyType:QueryDBTime  
RequestType:POST  
Amount of Error causing request fail: 0  
Mean latencies of all requests: 13ms  
Median latencies of all requests: 14ms  
95th percentile latency: 22ms  
99th percentile latency: 33ms  
  
Process finished with exit code 0
```



- ***Individual Server_Day5_GET_Client***



- ***Individual Server_Day5_GET_Server***

GetServerMetrics

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
↑
ServerSide Metrics
↓
ServerType:Individual Server
LatencyType:ResponseTime
RequestType:GET
Amount of Error causing request fail: 0
Mean latencies of all requests: 23ms
Median latencies of all requests: 18ms
95th percentile latency: 65ms
99th percentile latency: 95ms

Process finished with exit code 0
```

Latency

RequestBeginTime	Latency (ms)
1511269828343	10
1511269828356	15
1511269828615	20
1511269828748	30
1511269828880	40
1511269829058	50
1511269829181	60
1511269829288	70
1511269829398	80
1511269829484	90
1511269829599	100
1511269829734	110
1511269829734	120
1511269829843	130
1511269829948	140
1511269830086	150
1511269830194	160
1511269830309	170
1511269830553	180
1511269830731	190
1511269830903	200
1511269831175	210
1511269831282	220
1511269831499	230
1511269831605	240
1511269831806	250
1511269831973	260
1511269832143	270
1511269832747	280
1511269833138	290
1511269833238	300
1511269833367	310
1511269833467	320
1511269833571	330
1511269833677	340
1511269833782	350
1511269833911	360
1511269834082	370
1511269834189	380
1511269834230	390
1511269834452	400
1511269834593	410
1511269834711	420
1511269834838	430
1511269834979	440
1511269835101	450
1511269835230	460
1511269835372	470
1511269835429	480
1511269835526	490
1511269835691	500
1511269835814	510
1511269835981	520
1511269836103	530
1511269836216	540
1511269836344	550
1511269836473	560
1511269836604	570
1511269836730	580
1511269836887	590
1511269837045	600
1511269837234	610
1511269837322	620
1511269837487	630
1511269837659	640
1511269837713	650
1511269837893	660
1511269837987	670
1511269838087	680
1511269838187	690
1511269838287	700
1511269838387	710
1511269838487	720
1511269838587	730
1511269838687	740
1511269838787	750
1511269838887	760
1511269838987	770
1511269839087	780
1511269839187	790
1511269839287	800
1511269839387	810
1511269839487	820
1511269839587	830
1511269839687	840
1511269839787	850
1511269839887	860
1511269839987	870
1511269839987	880
1511269839987	890
1511269839987	900
1511269839987	910
1511269839987	920
1511269839987	930
1511269839987	940
1511269839987	950
1511269839987	960
1511269839987	970
1511269839987	980
1511269839987	990
1511269839987	1000
1511269839987	1010
1511269839987	1020
1511269839987	1030
1511269839987	1040
1511269839987	1050
1511269839987	1060
1511269839987	1070
1511269839987	1080
1511269839987	1090
1511269839987	1100
1511269839987	1110
1511269839987	1120
1511269839987	1130
1511269839987	1140
1511269839987	1150
1511269839987	1160
1511269839987	1170
1511269839987	1180
1511269839987	1190
1511269839987	1200
1511269839987	1210
1511269839987	1220
1511269839987	1230
1511269839987	1240
1511269839987	1250
1511269839987	1260
1511269839987	1270
1511269839987	1280
1511269839987	1290
1511269839987	1300
1511269839987	1310
1511269839987	1320
1511269839987	1330
1511269839987	1340
1511269839987	1350
1511269839987	1360
1511269839987	1370
1511269839987	1380
1511269839987	1390
1511269839987	1400
1511269839987	1410
1511269839987	1420
1511269839987	1430
1511269839987	1440
1511269839987	1450
1511269839987	1460
1511269839987	1470
1511269839987	1480
1511269839987	1490
1511269839987	1500
1511269839987	1510
1511269839987	1520
1511269839987	1530
1511269839987	1540
1511269839987	1550
1511269839987	1560
1511269839987	1570
1511269839987	1580
1511269839987	1590
1511269839987	1600
1511269839987	1610
1511269839987	1620
1511269839987	1630
1511269839987	1640
1511269839987	1650
1511269839987	1660
1511269839987	1670
1511269839987	1680
1511269839987	1690
1511269839987	1700
1511269839987	1710
1511269839987	1720
1511269839987	1730
1511269839987	1740
1511269839987	1750
1511269839987	1760
1511269839987	1770
1511269839987	1780
1511269839987	1790
1511269839987	1800

- Individual Servers_ Day5_ GET_ QueryDB

GetServerMetrics

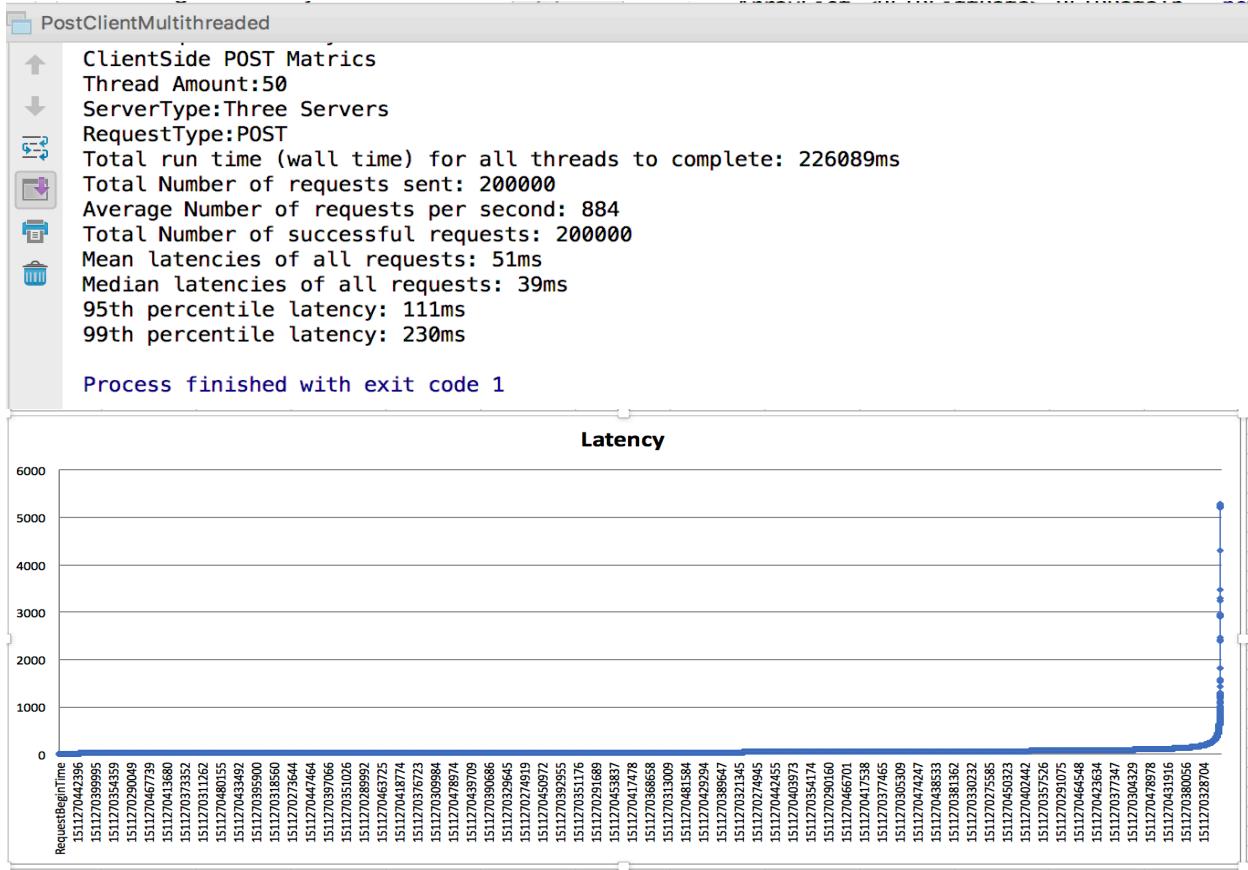
```
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
↑
ServerSide Metrics
↓
ServerType:Individual Server
LatencyType:QueryDBTime
RequestType:GET
Amount of Error causing request fail: 0
Mean latencies of all requests: 0ms
Median latencies of all requests: 1ms
95th percentile latency: 1ms
99th percentile latency: 2ms

Process finished with exit code 0
```

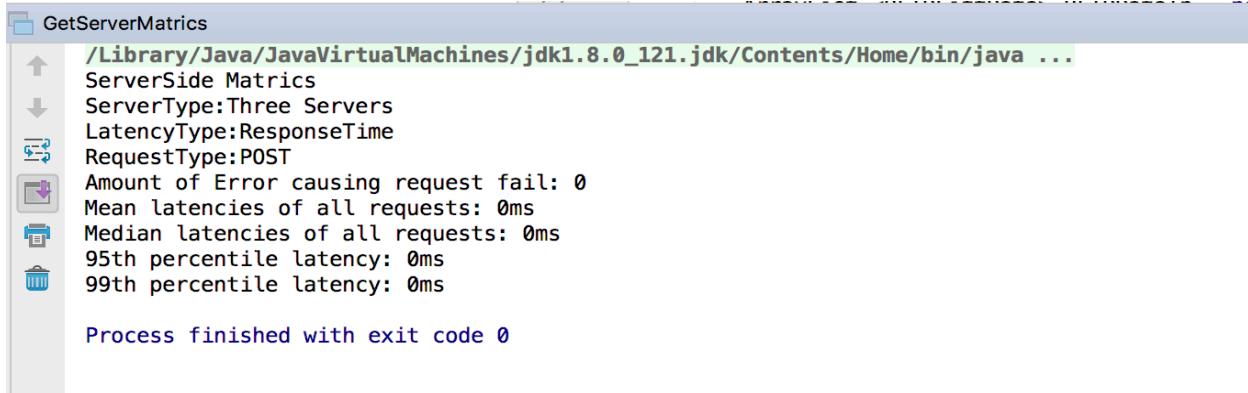
Latency

RequestBeginTime	Latency (ms)
1511269831973	1
1511269832143	2
1511269832747	3
1511269833138	4
1511269833238	5
1511269833367	6
1511269833467	7
1511269833571	8
1511269833677	9
1511269833782	10
1511269833911	11
1511269834082	12
1511269834189	13
1511269834230	14
1511269834350	15
1511269834452	16
1511269834593	17
1511269834711	18
1511269834838	1
1511269834979	2
1511269835101	3
1511269835230	4
1511269835372	5
1511269835429	6
1511269835526	7
1511269835691	8
1511269835814	9
1511269835981	10
1511269836103	11
1511269836216	12
1511269836344	13
1511269836473	14
1511269836604	15
1511269836730	16
1511269836887	17
1511269836993	1
1511269837045	2
1511269837234	3
1511269837322	4
1511269837400	5
1511269837476	6
1511269837548	7
1511269837619	8
1511269837654	9
1511269837713	10
1511269837791	11
1511269837867	12
1511269837937	13
1511269837987	14
1511269838056	15
1511269838138	16
1511269838216	17
1511269838341	18

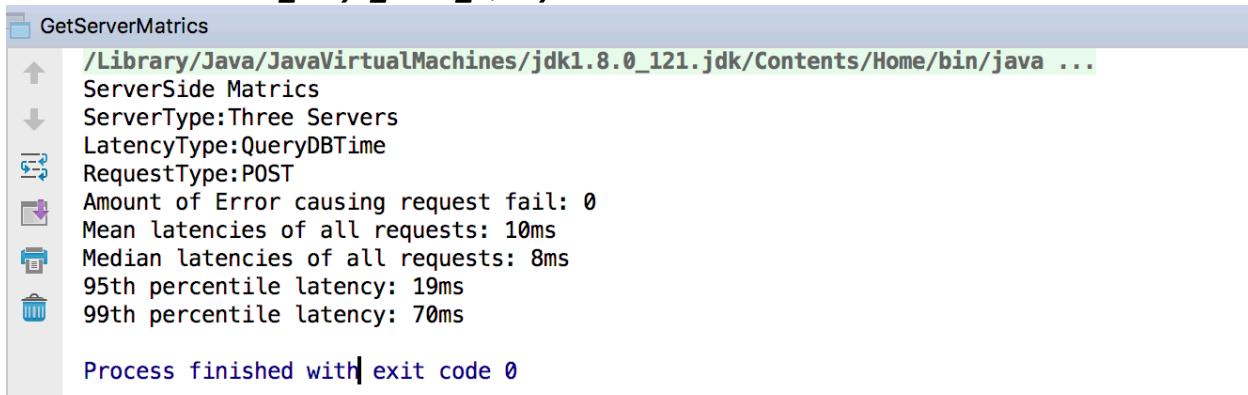
- ***Three Server_Day5_POST_Client***

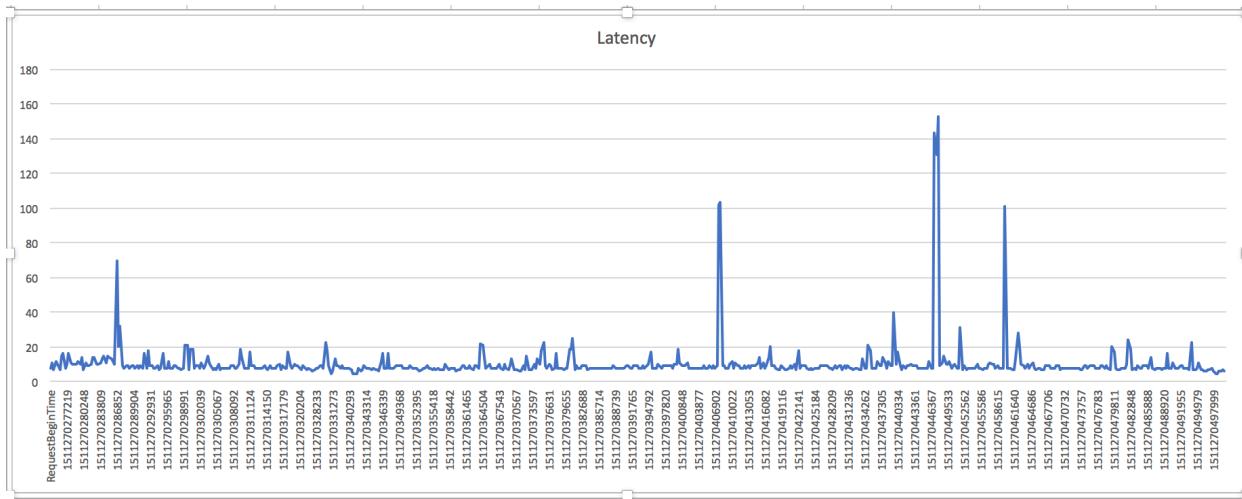


- ***Three Server_Day5_POST_Server***



- ***Three Servers_Day5_POST_QueryDB***

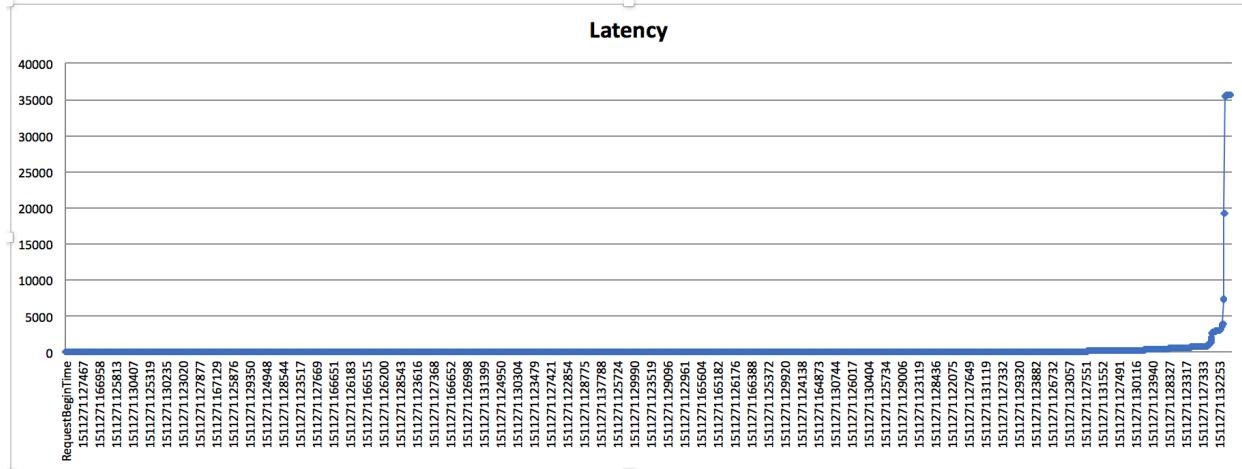




- *Three Server_Day5_GET_Client*

```
GetClientMultithreaded
ClientSide GET Metrics
Thread Amount:100
ServerType:Three Servers
RequestType:GET
Total run time (wall time) for all threads to complete: 49177ms
Total Number of requests sent: 10000
Average Number of requests per second: 204
Total Number of successful requests: 10000
Mean latencies of all requests: 337ms
Median latencies of all requests: 54ms
95th percentile latency: 547ms
99th percentile latency: 3044ms

Process finished with exit code 1
```



- *Three Server_ Day5_ GET_Server*

```

GetServerMetrics
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
↑ ServerSide Metrics
↓ ServerType:Three Servers
LatencyType:ResponseTime
RequestType:GET
Amount of Error causing request fail: 0
Mean latencies of all requests: 2ms
Median latencies of all requests: 2ms
95th percentile latency: 9ms
99th percentile latency: 14ms

Process finished with exit code 0

```

Latency

The chart displays the distribution of response times for individual requests. Most responses are clustered between 0 and 10 ms, with occasional outliers reaching up to 25 ms.

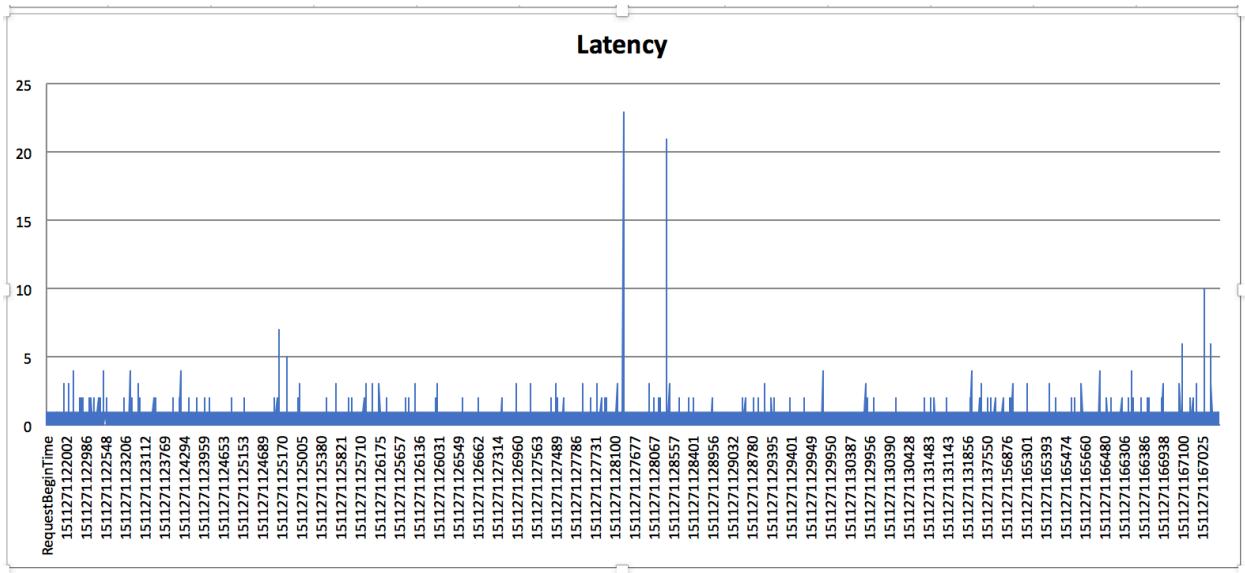
- Three Servers_ Day5_GET_QueryDB**

```

GetServerMetrics
/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Contents/Home/bin/java ...
↑ ServerSide Metrics
↓ ServerType:Three Servers
LatencyType:QueryDBTime
RequestType:GET
Amount of Error causing request fail: 0
Mean latencies of all requests: 0ms
Median latencies of all requests: 1ms
95th percentile latency: 1ms
99th percentile latency: 2ms

Process finished with exit code 0

```



Observation:

1. With three servers, all the get requests for each day have apparent improved. However, the post request keep the same. The reason is that I use batch to insert and dataset is not large, so the improvement is not remarkable.
2. For POST request ClientWallTime > QueryDbTime> ServerResponseTime.

For Get request ClientWallTime > ServerResponseTime > QueryDbTime.

The reason is I use batch to post. Post request will return after put the data into cache, so the latency of the response time in the server side always to be 0ms.

Bottlenecks and ways to improve the server performance

1. If upload data is large, it might be better to create a monitor server and use SQS to communicate.
2. Current latency monitor does not support post two same day data. The latency can be distinguished only by hostName, dayNum, requestType, latencyType. Once we want to re_upload a same day data, we need to reset the database first.