Name: Nachiket Dani

Assignment-2 Report

CS6650 : Distributed Systems

The following describes the final (Server) modifications for the assignment (building upon Assignment1) that simulates purchase post requests through multiple threads comparing a single server to a scaled (4) server setup with a load balancer within AWS environment. Git links provided in Canvas submission.

A database was introduced in this assignment to store purchase data from the client made POST requests. We use an Amazon RDS storage via a local mySQL Workbench instance. We create a new POJO: PurchaseModel that encases the Purchase class which in turn holds PurchaseItems. This allows us to use a DAO layer (PurchaseDao) and a database connector class (DBConnector) to pass the parsed request body information to the database. The PurchaseModel created reflects the database schema. It consists of the fields: storeID, customerID, purchaseDate and items. We do not use a primary key for the table as we want to prioritize data insertion. Additionally, we use a JSON data type for items. Refer "PurchasesDB_Schema.sql" for the database schema. This file is embedded in the CS6650_a2Servlet project files.

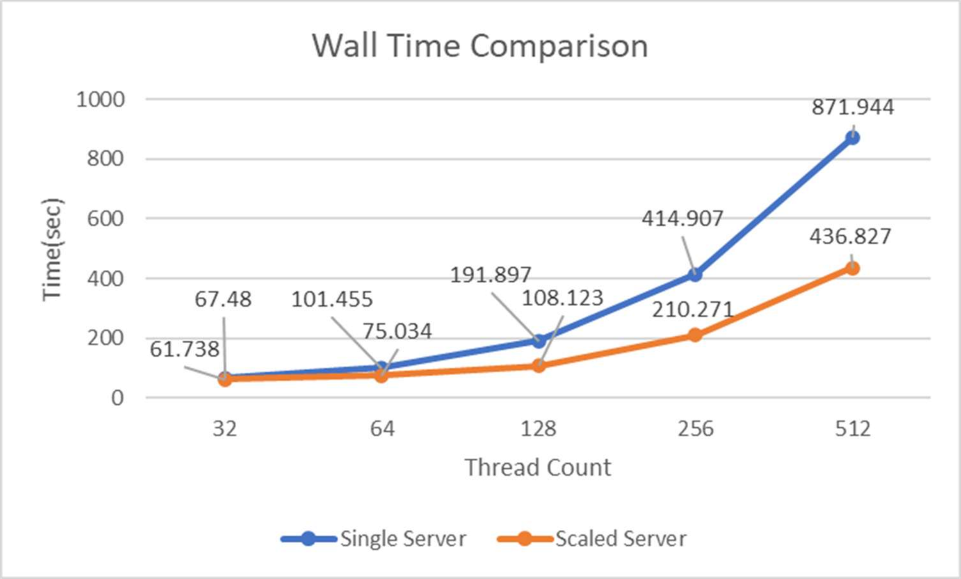
The DBConnector uses HikariCP which is a lightweight JDBC connection pool to handle the volume of write requests effectively. This decision was made after consulting the TA since the Apache Commons DBCP was not able to handle the data flow.

I was able to run the Single Server as well as the Scaled system successfully and write records into the database for 32, 64, 128, 256 and 512 threads (maxStores). The client largely remains the same, however I incorporated a large enough apiClient timeout value to achieve results. Another additional change made to the client includes modifying the default numPurchases from 60 to 300 based on the new requirement.

Comparison Results

WALL TIME

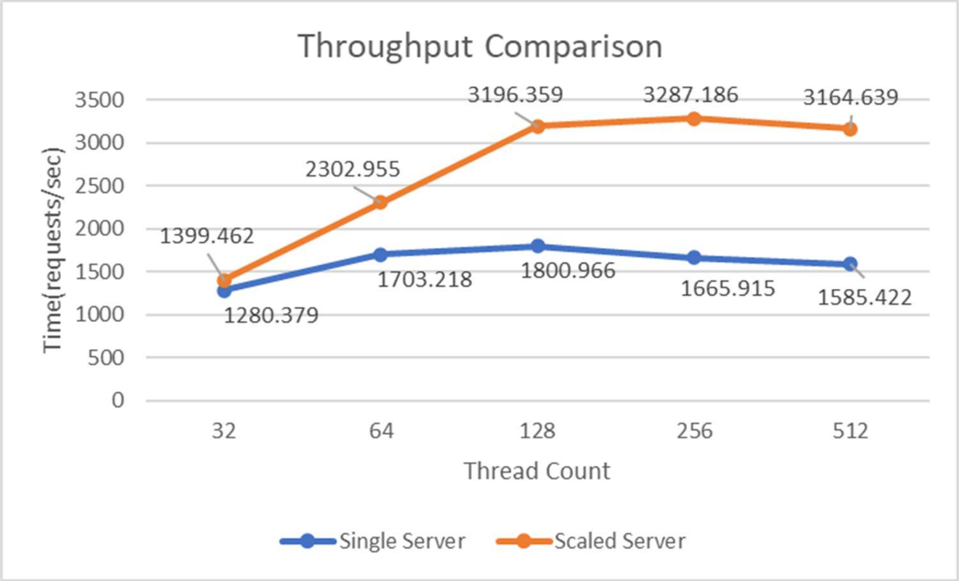
	32	64	128	256	512
Single Server	67.48	101.455	191.897	414.907	871.944
Scaled Server	61.738	75.034	108.123	210.271	436.827



We observe that the wall times significantly improve for our scaled-out system. The times halved across the thread counts compared.

THROUGHPUT

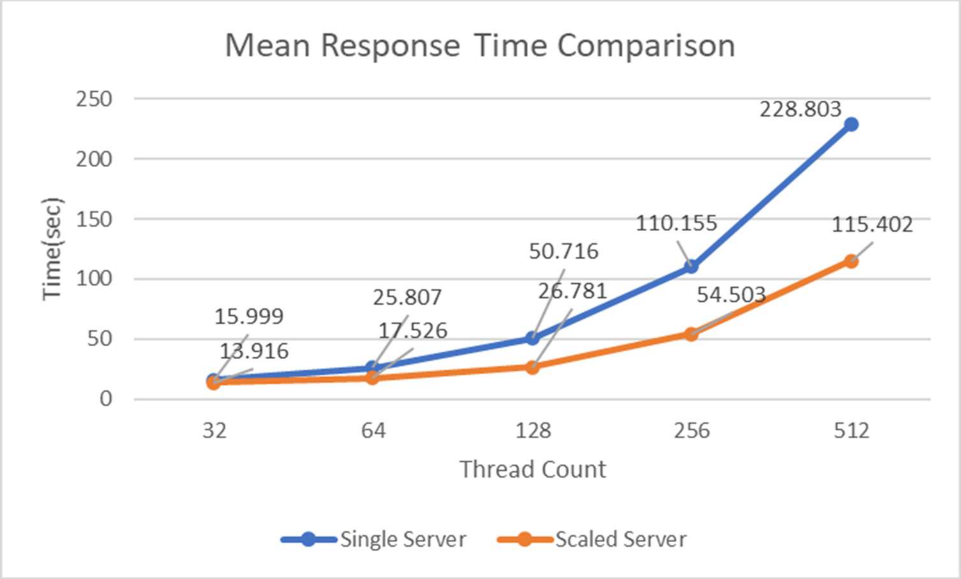
	32	64	128	256	512
Single Server	1280.379	1703.218	1800.966	1665.915	1585.422
Scaled Server	1399.462	2302.955	3196.359	3287.186	3164.639



Throughput for the scaled system is also significantly better. We do see that however for both systems as thread counts increase, throughput stalls and falls off.

Mean Response Time

	32	64	128	256	512
Single Server	15.999	25.807	50.716	110.155	228.803
Scaled Server	13.916	17.526	26.781	54.503	115.402



Mean response times for threads in the scaled system are also better as compared to the single server instance. See the raw data section at the end of this report for the raw data images for the response times and database records tabulated.

System Design Considerations

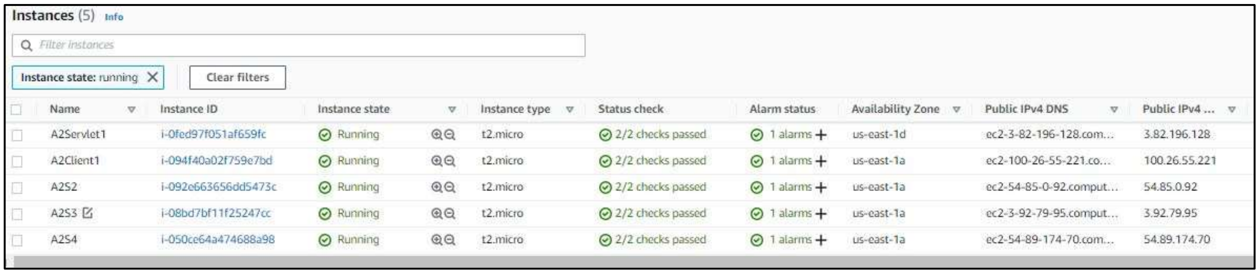
The Server is implemented by examining the requirement from the POST Purchase servlet described in the existing Swagger client authored by Prof. Gorton.

Ref: https://app.swaggerhub.com/apis/gortonator/GianTigle/1.0.0#/purchase/newPurchase

The Swagger codegen was used to create the Swagger client API interface which the client is built on.

Server implements a simple URL validation methodology. To verify the request body, I created a POJO PurchaseItem and Purchase. Then used GSON to decode the incoming JSON string and test casting into the POJO. Failure to do so was considered a POST failure.

Both Server and Client were deployed on separate instances of Amazon EC2 (free tier, Virginia). I used the image created of the single server to create additional instances (3) and used a application load balancer to coordinate usage between the replicas. *(Images now disabled)*


Instances (5) Info

Filter instances

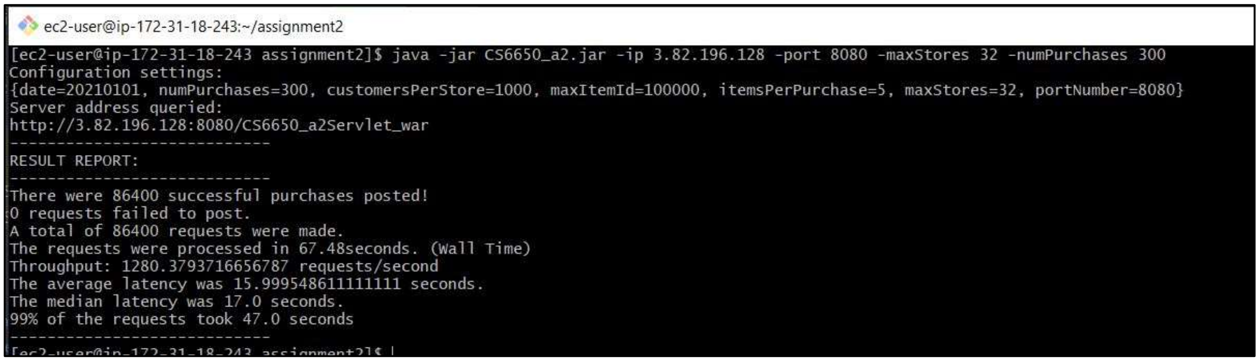
Instance state: running ✕ | Clear filters

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
□	A2Servlet1	i-0fed97f051af659fc	✓ Running	t2.micro	✓ 2/2 checks passed	✓ 1 alarms +	us-east-1d	ec2-3-82-196-128.com...	3.82.196.128
□	A2Client1	i-094f40a02f759e7bd	✓ Running	t2.micro	✓ 2/2 checks passed	✓ 1 alarms +	us-east-1a	ec2-100-26-55-221.co...	100.26.55.221
□	A2S2	i-092e663656dd5473c	✓ Running	t2.micro	✓ 2/2 checks passed	✓ 1 alarms +	us-east-1a	ec2-54-85-0-92.comput...	54.85.0.92
□	A2S3	i-08bd7bf11f25247cc	✓ Running	t2.micro	✓ 2/2 checks passed	✓ 1 alarms +	us-east-1a	ec2-3-92-79-95.comput...	3.92.79.95
□	A2S4	i-050ce64a474688a98	✓ Running	t2.micro	✓ 2/2 checks passed	✓ 1 alarms +	us-east-1a	ec2-54-89-174-70.com...	54.89.174.70

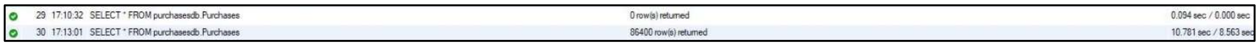
Raw Data Screengrabs

Single Server Data:

32 Threads:

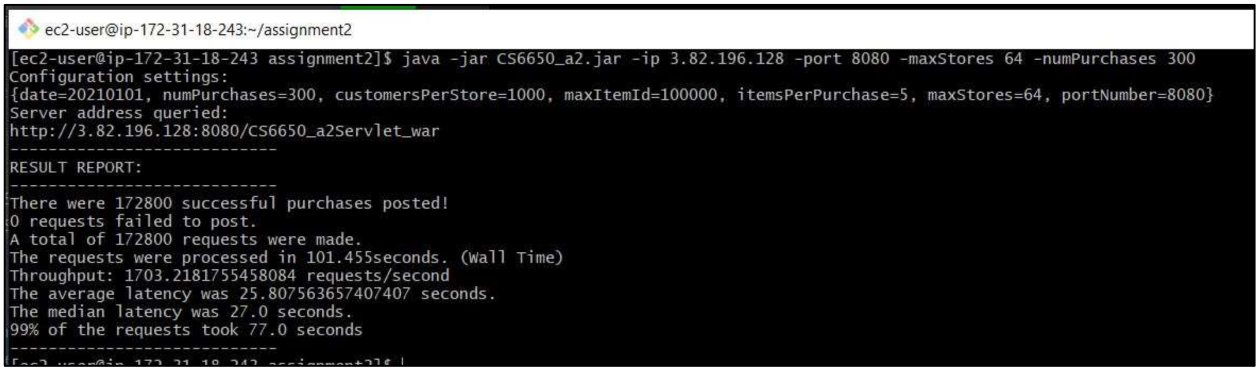

ec2-user@ip-172-31-18-243:~/assignment2

```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip 3.82.196.128 -port 8080 -maxStores 32 -numPurchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=32, portNumber=8080}
Server address queried:
http://3.82.196.128:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 86400 successful purchases posted!
0 requests failed to post.
A total of 86400 requests were made.
The requests were processed in 67.48seconds. (Wall Time)
Throughput: 1280.3793716656787 requests/second
The average latency was 15.999548611111111 seconds.
The median latency was 17.0 seconds.
99% of the requests took 47.0 seconds
----------------------------
[ec2-user@ip-172-31-18-243 assignment2]$ |
```

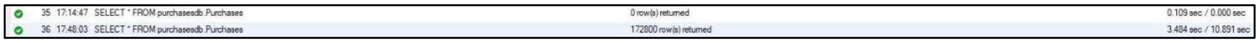


```
29  17:10:32  SELECT * FROM purchasesdb.Purchases    0 row(s) returned        0.094 sec / 0.000 sec
30  17:13:01  SELECT * FROM purchasesdb.Purchases    86400 row(s) returned    10.781 sec / 8.563 sec
```

64 Threads:

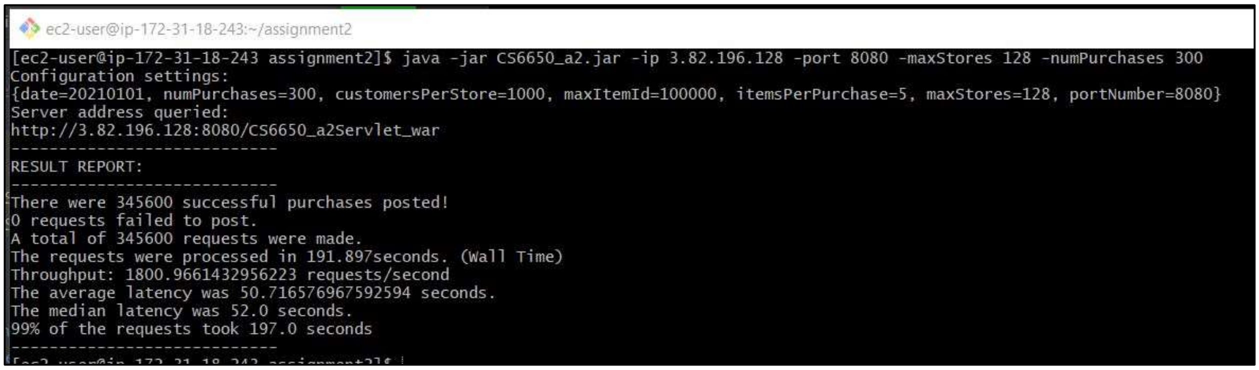

ec2-user@ip-172-31-18-243:~/assignment2

```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip 3.82.196.128 -port 8080 -maxStores 64 -numPurchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=64, portNumber=8080}
Server address queried:
http://3.82.196.128:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 172800 successful purchases posted!
0 requests failed to post.
A total of 172800 requests were made.
The requests were processed in 101.455seconds. (Wall Time)
Throughput: 1703.2181755458084 requests/second
The average latency was 25.807563657407407 seconds.
The median latency was 27.0 seconds.
99% of the requests took 77.0 seconds
----------------------------
[ec2-user@ip-172-31-18-243 assignment2]$ |
```

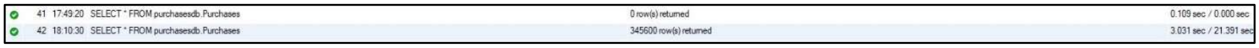


```
35  17:14:47  SELECT * FROM purchasesdb.Purchases    0 row(s) returned         0.109 sec / 0.000 sec
36  17:48:03  SELECT * FROM purchasesdb.Purchases    172800 row(s) returned    3.484 sec / 10.891 sec
```

128 Threads:

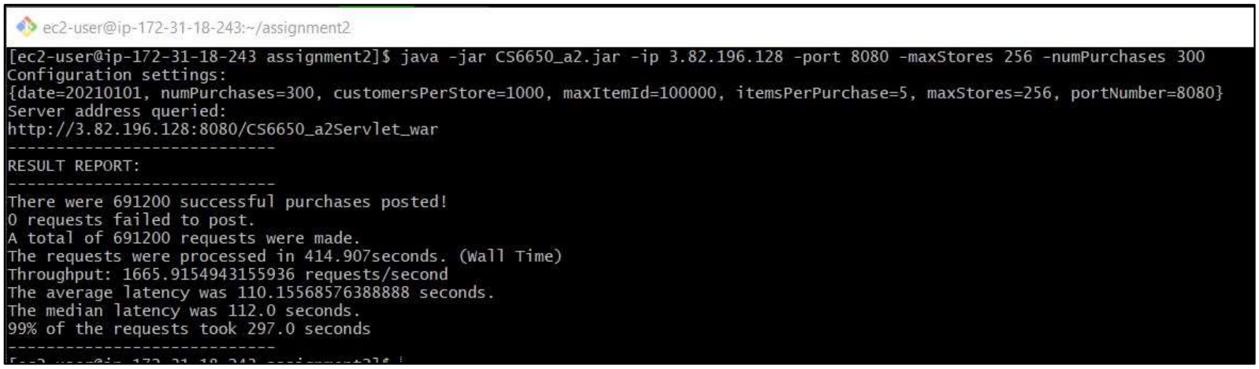


```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip 3.82.196.128 -port 8080 -maxStores 128 -numPurchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=128, portNumber=8080}
Server address queried:
http://3.82.196.128:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 345600 successful purchases posted!
0 requests failed to post.
A total of 345600 requests were made.
The requests were processed in 191.897seconds. (Wall Time)
Throughput: 1800.9661432956223 requests/second
The average latency was 50.716576967592594 seconds.
The median latency was 52.0 seconds.
99% of the requests took 197.0 seconds
----------------------------
[ec2-user@ip-172-31-18-243 assignment2]$
```

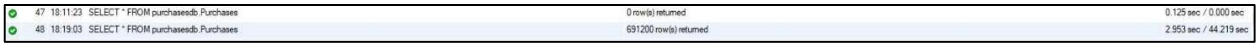


```
41  17:49:20  SELECT * FROM purchasesdb.Purchases    0 row(s) returned         0.109 sec / 0.000 sec
42  18:10:30  SELECT * FROM purchasesdb.Purchases    345600 row(s) returned    3.031 sec / 21.391 sec
```

256 Threads:

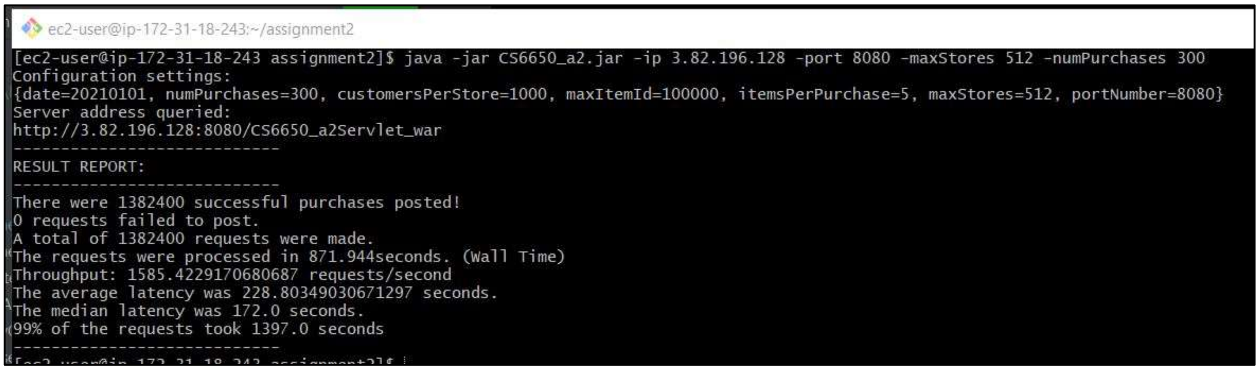


```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip 3.82.196.128 -port 8080 -maxStores 256 -numPurchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=256, portNumber=8080}
Server address queried:
http://3.82.196.128:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 691200 successful purchases posted!
0 requests failed to post.
A total of 691200 requests were made.
The requests were processed in 414.907seconds. (Wall Time)
Throughput: 1665.9154943155936 requests/second
The average latency was 110.15568576388888 seconds.
The median latency was 112.0 seconds.
99% of the requests took 297.0 seconds
----------------------------
[ec2-user@ip-172-31-18-243 assignment2]$
```

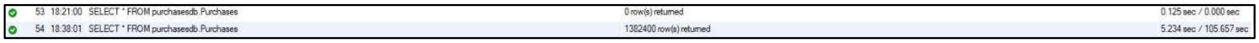


```
47  18:11:23  SELECT * FROM purchasesdb.Purchases    0 row(s) returned         0.125 sec / 0.000 sec
48  18:19:03  SELECT * FROM purchasesdb.Purchases    691200 row(s) returned    2.953 sec / 44.219 sec
```

512 Threads:



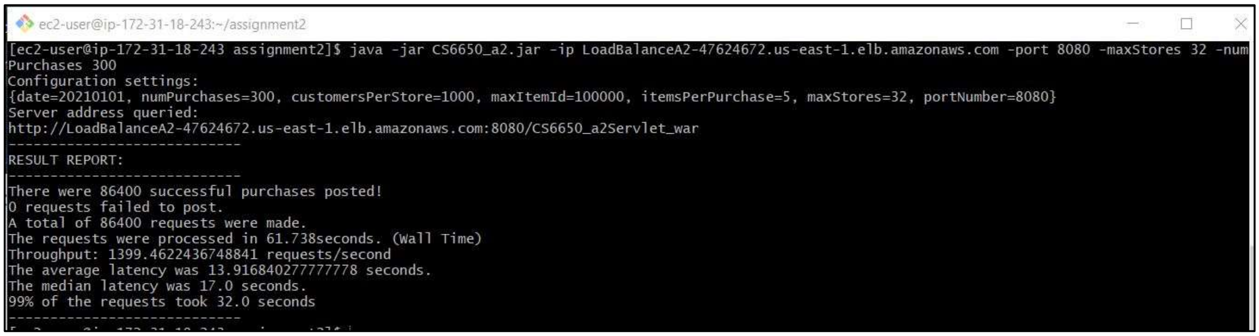
```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip 3.82.196.128 -port 8080 -maxStores 512 -numPurchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=512, portNumber=8080}
Server address queried:
http://3.82.196.128:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 1382400 successful purchases posted!
0 requests failed to post.
A total of 1382400 requests were made.
The requests were processed in 871.944seconds. (Wall Time)
Throughput: 1585.4229170680687 requests/second
The average latency was 228.80349030671297 seconds.
The median latency was 172.0 seconds.
99% of the requests took 1397.0 seconds
----------------------------
[ec2-user@ip-172-31-18-243 assignment2]$
```



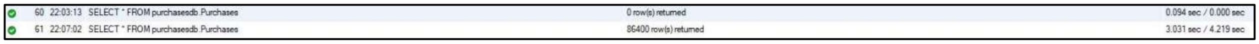
```
53  18:21:00  SELECT * FROM purchasesdb.Purchases    0 row(s) returned          0.125 sec / 0.000 sec
54  18:38:01  SELECT * FROM purchasesdb.Purchases    1382400 row(s) returned    5.234 sec / 105.657 sec
```

Scaled System Data:

32 Threads

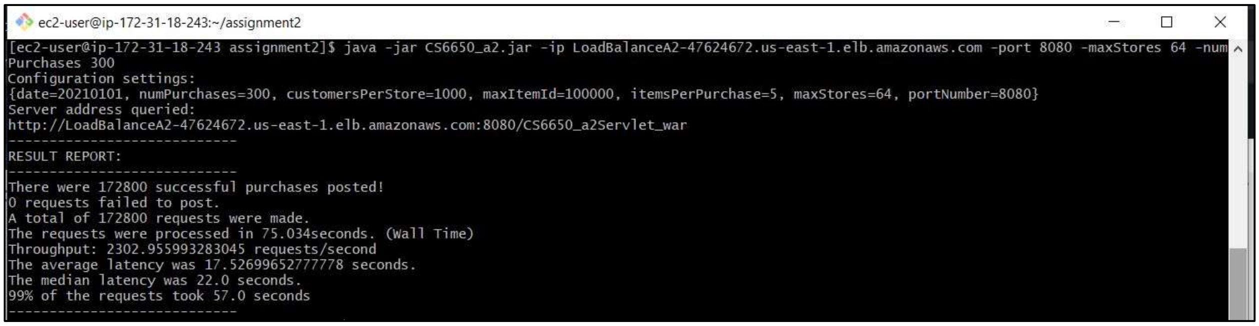


```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com -port 8080 -maxStores 32 -num
Purchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=32, portNumber=8080}
Server address queried:
http://LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 86400 successful purchases posted!
0 requests failed to post.
A total of 86400 requests were made.
The requests were processed in 61.738seconds. (Wall Time)
Throughput: 1399.4622436748841 requests/second
The average latency was 13.916840277777778 seconds.
The median latency was 17.0 seconds.
99% of the requests took 32.0 seconds
----------------------------
```

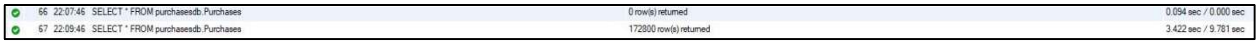


```
60  22:03:13  SELECT * FROM purchasesdb.Purchases   0 row(s) returned        0.094 sec / 0.000 sec
61  22:07:02  SELECT * FROM purchasesdb.Purchases   86400 row(s) returned    3.031 sec / 4.219 sec
```

64 Threads

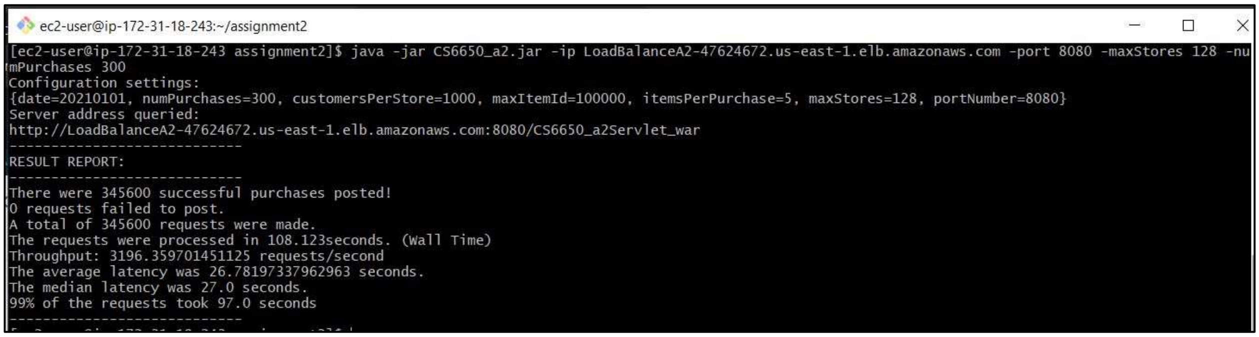


```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com -port 8080 -maxStores 64 -num
Purchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=64, portNumber=8080}
Server address queried:
http://LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 172800 successful purchases posted!
0 requests failed to post.
A total of 172800 requests were made.
The requests were processed in 75.034seconds. (Wall Time)
Throughput: 2302.955993283045 requests/second
The average latency was 17.52699652777778 seconds.
The median latency was 22.0 seconds.
99% of the requests took 57.0 seconds
----------------------------
```

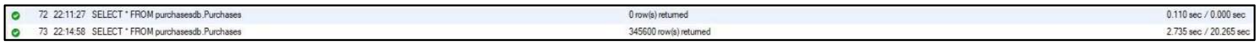


```
66  22:07:46  SELECT * FROM purchasesdb.Purchases   0 row(s) returned         0.094 sec / 0.000 sec
67  22:09:46  SELECT * FROM purchasesdb.Purchases   172800 row(s) returned    3.422 sec / 9.781 sec
```

128 Threads

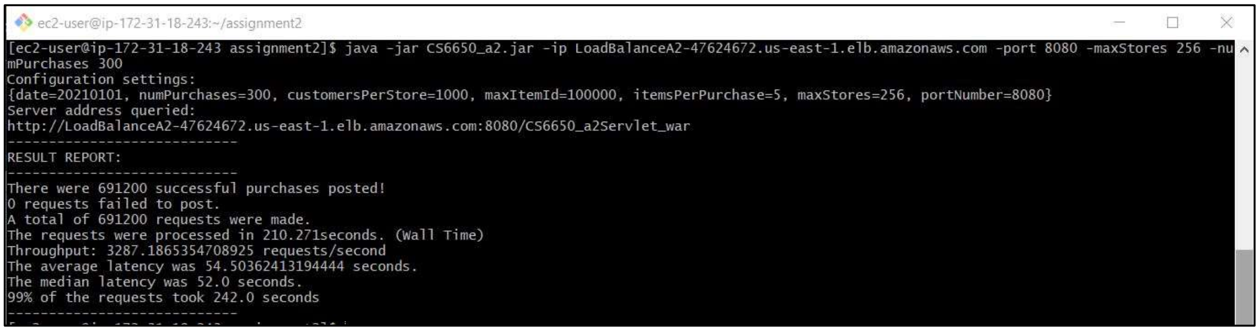


```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com -port 8080 -maxStores 128 -nu
mPurchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=128, portNumber=8080}
Server address queried:
http://LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 345600 successful purchases posted!
0 requests failed to post.
A total of 345600 requests were made.
The requests were processed in 108.123seconds. (Wall Time)
Throughput: 3196.359701451125 requests/second
The average latency was 26.78197337962963 seconds.
The median latency was 27.0 seconds.
99% of the requests took 97.0 seconds
----------------------------
```

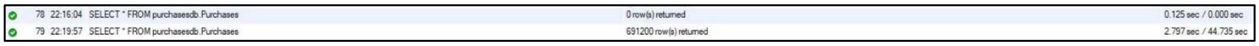


```
72  22:11:27  SELECT * FROM purchasesdb.Purchases   0 row(s) returned         0.110 sec / 0.000 sec
73  22:14:58  SELECT * FROM purchasesdb.Purchases   345600 row(s) returned    2.735 sec / 20.265 sec
```

256 Threads

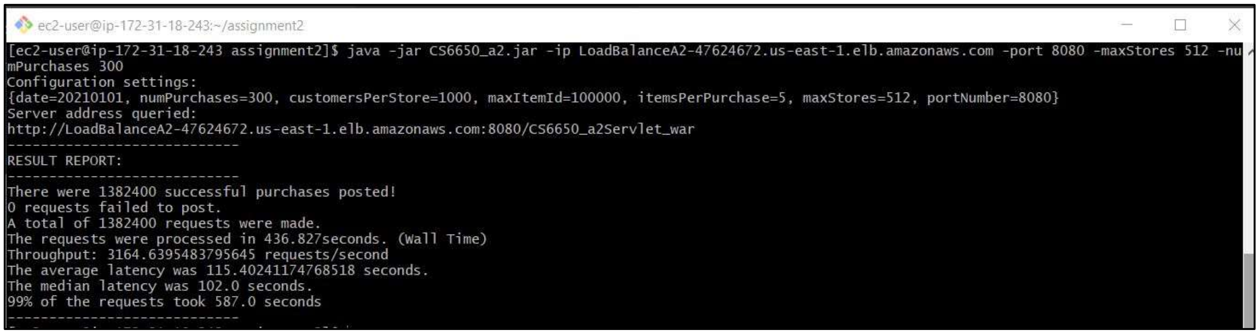


```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com -port 8080 -maxStores 256 -nu
mPurchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=256, portNumber=8080}
Server address queried:
http://LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 691200 successful purchases posted!
0 requests failed to post.
A total of 691200 requests were made.
The requests were processed in 210.271seconds. (Wall Time)
Throughput: 3287.1865354708925 requests/second
The average latency was 54.50362413194444 seconds.
The median latency was 52.0 seconds.
99% of the requests took 242.0 seconds
----------------------------
```

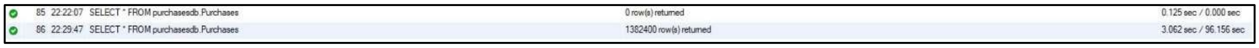


```
78 22:16:04 SELECT * FROM purchasesdb.Purchases    0 row(s) returned         0.125 sec / 0.000 sec
79 22:19:57 SELECT * FROM purchasesdb.Purchases    691200 row(s) returned    2.797 sec / 44.735 sec
```

512 Threads



```
[ec2-user@ip-172-31-18-243 assignment2]$ java -jar CS6650_a2.jar -ip LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com -port 8080 -maxStores 512 -nu
mPurchases 300
Configuration settings:
{date=20210101, numPurchases=300, customersPerStore=1000, maxItemId=100000, itemsPerPurchase=5, maxStores=512, portNumber=8080}
Server address queried:
http://LoadBalanceA2-47624672.us-east-1.elb.amazonaws.com:8080/CS6650_a2Servlet_war
----------------------------
RESULT REPORT:
----------------------------
There were 1382400 successful purchases posted!
0 requests failed to post.
A total of 1382400 requests were made.
The requests were processed in 436.827seconds. (Wall Time)
Throughput: 3164.6395483795645 requests/second
The average latency was 115.40241174768518 seconds.
The median latency was 102.0 seconds.
99% of the requests took 587.0 seconds
----------------------------
```



```
85 22:22:07 SELECT * FROM purchasesdb.Purchases    0 row(s) returned          0.125 sec / 0.000 sec
86 22:29:47 SELECT * FROM purchasesdb.Purchases    1382400 row(s) returned    3.062 sec / 96.156 sec
```