Building Scalable Distributed Systems

Assignment 2

Git Repository: https://github.com/akksshah/building-scalable-distributed-systems

Server Design Description

The server receives the request from the client in the form of Json. It converts the Json string to the PurchaseOrder object that, the client requests the server to persist in the MySQL database. The conversion of json string to PurchaseOrder object is done using Jackson library. The server then tries to save the object by calling on the data access object (DAO)'s method: savePurchaseOrder(order), in our case it is the instance of the class PurchaseTransaction.
The DAO, requests connection from the ConnectionUtility class. The ConnectionUtility class maintains a pool of database connections. It provides the DAO with Connection objects as and when it requests for the same. For the single server tests, the connection pool size for the Database was set to 60. For the load balanced system, each server was granted 15 connections to the database access pool. The DAO on getting a connection would initiate a transaction, it would then try to insert the purchaseOrder into the database. On successful insertion, it would then commit to the database. If an exception would occur, it would roll back the transaction. In such case, where the insert failed, the server would send the client a SC_NOT_OK status code. The client can then retry the request again.

Database design.
Initially I went with a design which had two tables, the first stored the storeId, customerId and date (with an auto generated primary key called id). The second table stored the itemId, numberOfItems purchased of that Item and the foreign key, the "id" from the first table. Having a normalized form was good for querying, however, it compromised on write speeds.

I then switched to a single table where each row stored storeId, customerId, date, itemId, numberOfitems purchased for that time. Moving to this database schema gave a very high throughput compared to the first schema. This schema although is not efficient, it yields high write throughputs.

The first database schema started to use burst credits for the same to keep up with the requests, on the other hand, the second schema scaled pretty fast and easily and did not use any burst credits for the same.

Connection Pools:
Hibernate connection pools were way slow compared to Hikari Connection Pools. Hikari Connection Pool provided almost 10 time the throughput compared to the Hibernate connection pools.

Single Server Test:

32 Threads

```
Starting Execution at: 2021-03-9 12:42:07.763
Launching stores for eastern time
Launching stores for central time
Launching stores for pacific time
Total request sent: 86400
Total request successful: 86400
Total unsuccessful request: 0
Total wall time: 119789ms
Throughput: 726.0504201680673
Mean response time: 43.36069444444445
Median response time: 38.5
p99 (99 percentile): 211.0
Max response time: OptionalDouble[2643.0]
Ending execution at: 2021-03-9 12:44:07.828

Process finished with exit code 0
```

64 Threads

```
Starting Execution at: 2021-03-9 12:45:00.089
Launching stores for eastern time
Launching stores for central time
Launching stores for pacific time
Total request sent: 172800
Total request successful: 172800
Total unsuccessful request: 0
Total wall time: 186100ms
Throughput: 929.0322580645161
Mean response time: 68.16756365740741
Median response time: 45.0
p99 (99 percentile): 219.0
Max response time: OptionalDouble[992.0]
Ending execution at: 2021-03-9 12:48:06.536

Process finished with exit code 0
```

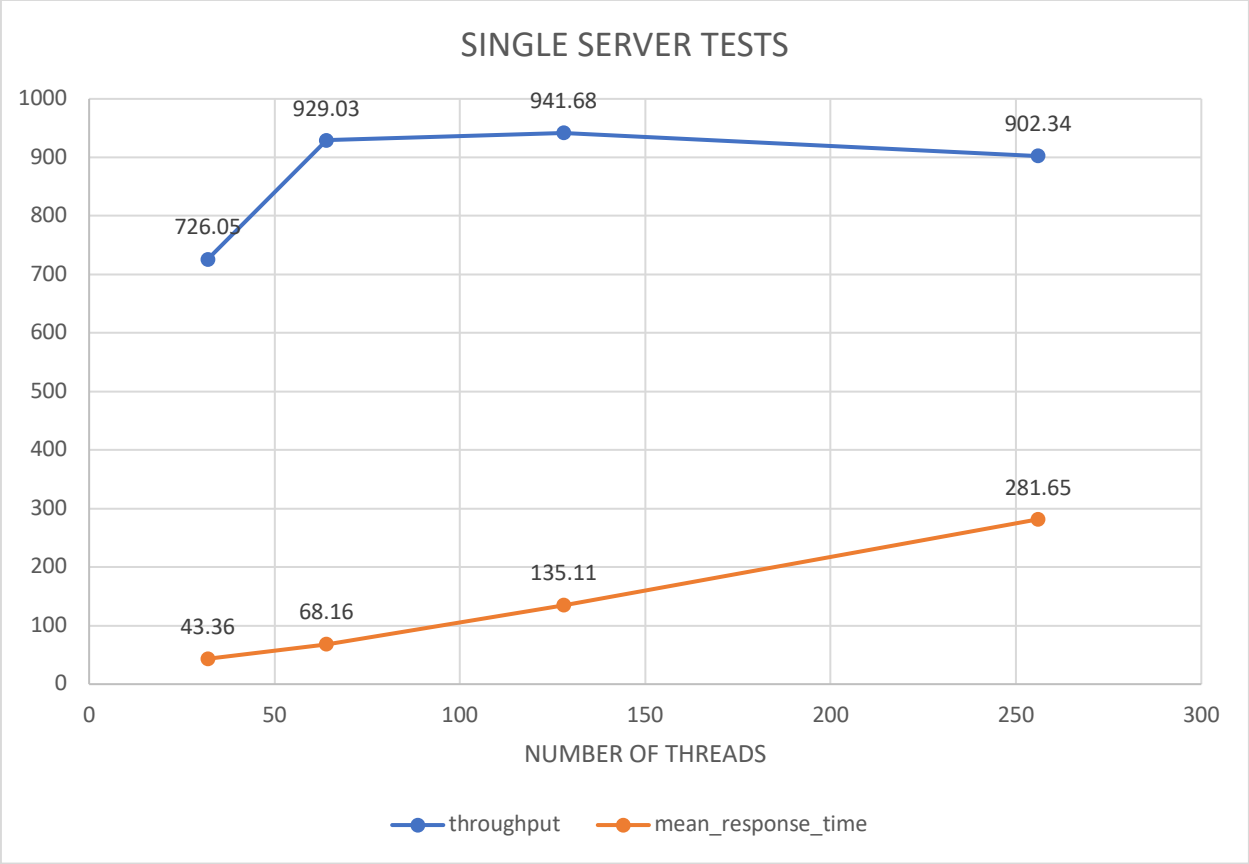
128 Threads

```
Starting Execution at: 2021-03-9 12:48:18.810
Launching stores for eastern time
Launching stores for central time
Launching stores for pacific time
Total request sent: 345600
Total request successful: 345600
Total unsuccessful request: 0
Total wall time: 367419ms
Throughput: 941.6893732970027
Mean response time: 135.11162905092593
Median response time: 242.0
p99 (99 percentile): 290.0
Max response time: OptionalDouble[1046.0]
Ending execution at: 2021-03-9 12:54:26.831

Process finished with exit code 0
```

256 Threads

```
Total request sent: 691200
Total request successful: 691199
Total unsuccessful request: 1
Total wall time: 766405ms
Throughput: 902.3498694516971
Mean response time: 281.6542867476852
Median response time: 265.5
p99 (99 percentile): 978.0
Max response time: OptionalDouble[10005.0]
Ending execution at: 2021-03-9 13:07:31.852

Process finished with exit code 0
```

Performance graph



Load Balanced Server Test

32 Threads

```
Total request sent: 86400
Total request successful: 86400
Total unsuccessful request: 0
Total wall time: 115869ms
Throughput: 751.304347826087
Mean response time: 42.387685185185184
Median response time: 48.0
p99 (99 percentile): 86.0
Max response time: OptionalDouble[866.0]
Ending execution at: 2021-03-8 18:09:23.776
```

64 Threads

```
Total request sent: 172800
Total request successful: 172800
Total unsuccessful request: 0
Total wall time: 130206ms
Throughput: 1329.2307692307693
Mean response time: 47.56027777777778
Median response time: 45.0
p99 (99 percentile): 144.0
Max response time: OptionalDouble[1095.0]
Ending execution at: 2021-03-8 18:07:16.247
```

128 Threads

```
Total request sent: 345600
Total request successful: 345600
Total unsuccessful request: 0
Total wall time: 147415ms
Throughput: 2351.0204081632655
Mean response time: 51.700888310185185
Median response time: 53.0
p99 (99 percentile): 108.0
Max response time: OptionalDouble[7606.0]
Ending execution at: 2021-03-8 18:04:51.776
```

256 Threads

```
Total request sent: 691200
Total request successful: 691200
Total unsuccessful request: 0
Total wall time: 220467ms
Throughput: 3141.818181818182
Mean response time: 79.67632957175925
Median response time: 47.0
p99 (99 percentile): 361.0
Max response time: OptionalDouble[3701.0]
Ending execution at: 2021-03-8 18:02:07.449
```

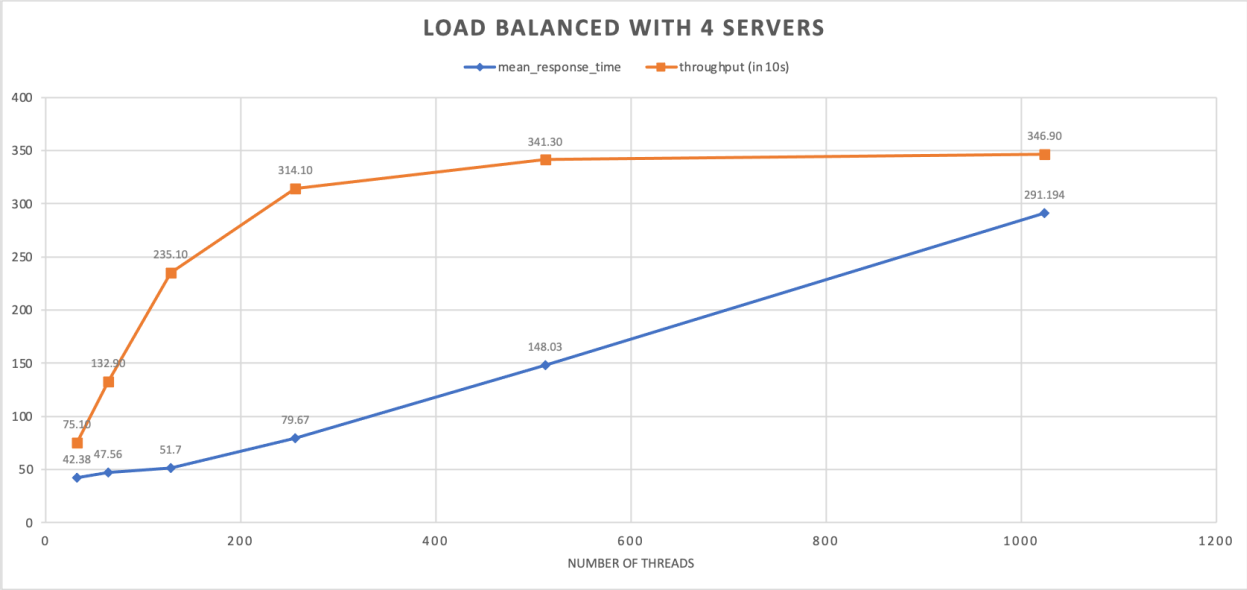
512 threads

```
Total request sent: 1382400
Total request successful: 1382400
Total unsuccessful request: 0
Total wall time: 405707ms
Throughput: 3413.3333333333335
Mean response time: 148.03738642939814
Median response time: 50.5
p99 (99 percentile): 453.0
Max response time: OptionalDouble[4321.0]
Ending execution at: 2021-03-8 18:16:25.177
```

1024 threads

```
Total request sent: 2764800
Total request successful: 2764799
Total unsuccessful request: 1
Total wall time: 797755ms
Throughput: 3469.00878293601
Mean response time: 291.19405454282406
Median response time: 227.0
p99 (99 percentile): 1238.0
Max response time: OptionalDouble[10070.0]
Ending execution at: 2021-03-8 18:31:21.409
```

Performance graph



Note: The throughput has been scaled with 1 point referring to 10 points on the actual scale.