distributed-systems-work

Assignment1 Repo for CS6650 Building Scalable Distributed Systems 23fall@neu

https://gortonator.github.io/bsds-6650/assignments-2022/Assignment-1

Github link: https://github.com/98may/distributed-systems-work/

1: info

my aws ec2 ip address: `3.80.33.155:8080`

shortcuts:

Go get : http://3.80.33.155:8081/IGORTON/AlbumStore/1.0.0/albums/1

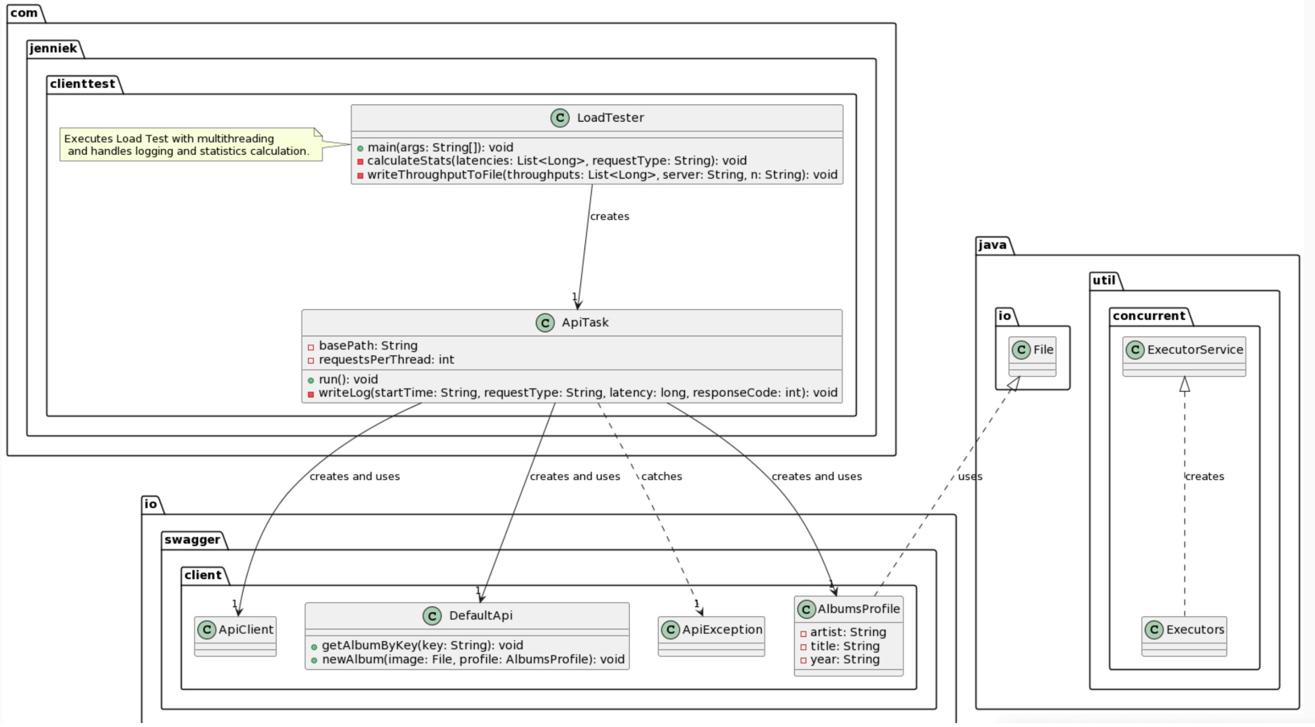
Java get: http://3.80.33.155:8080/AlbumApp/albums/2

2: client design

key1: Use ThreadPool to group threads

key2: Use count down latch to graceful wait for threads to finish

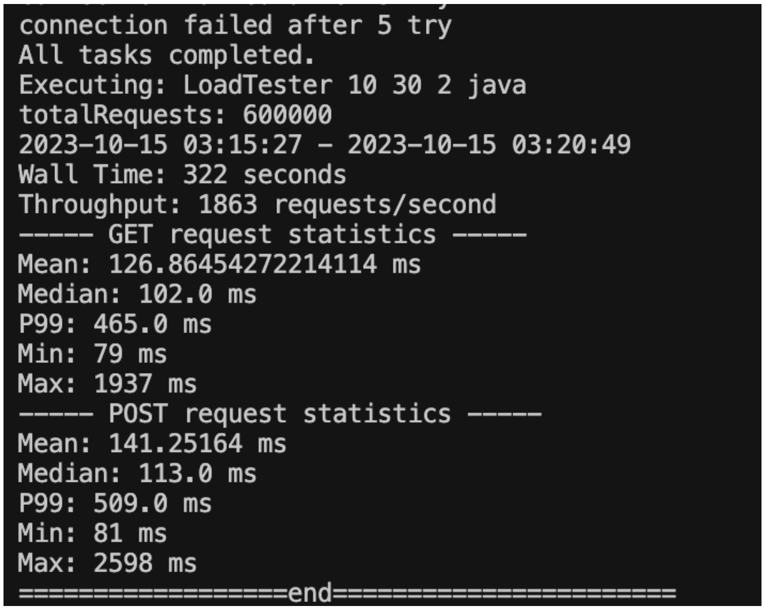
![diagram_loadTester](/Users/may/Desktop/neu/cs6650_distributed/distributed-systems-work/hw1/diagram_loadTester.png)


com
jenniek
clienttest
Executes Load Test with multithreading and handles logging and statistics calculation.
LoadTester
main(args: String[]): void
calculateStats(latencies: List<Long>, requestType: String): void
writeThroughputToFile(throughputs: List<Long>, server: String, n: String): void
creates
1
ApiTask
basePath: String
requestsPerThread: int
run(): void
writeLog(startTime: String, requestType: String, latency: long, responseCode: int): void
creates and uses
creates and uses
catches
creates and uses
uses
creates
io
swagger
client
1
ApiClient
1
DefaultApi
getAlbumByKey(key: String): void
newAlbum(image: File, profile: AlbumsProfile): void
1
ApiException
1
AlbumsProfile
artist: String
title: String
year: String
java
io
File
util
concurrent
ExecutorService
Executors

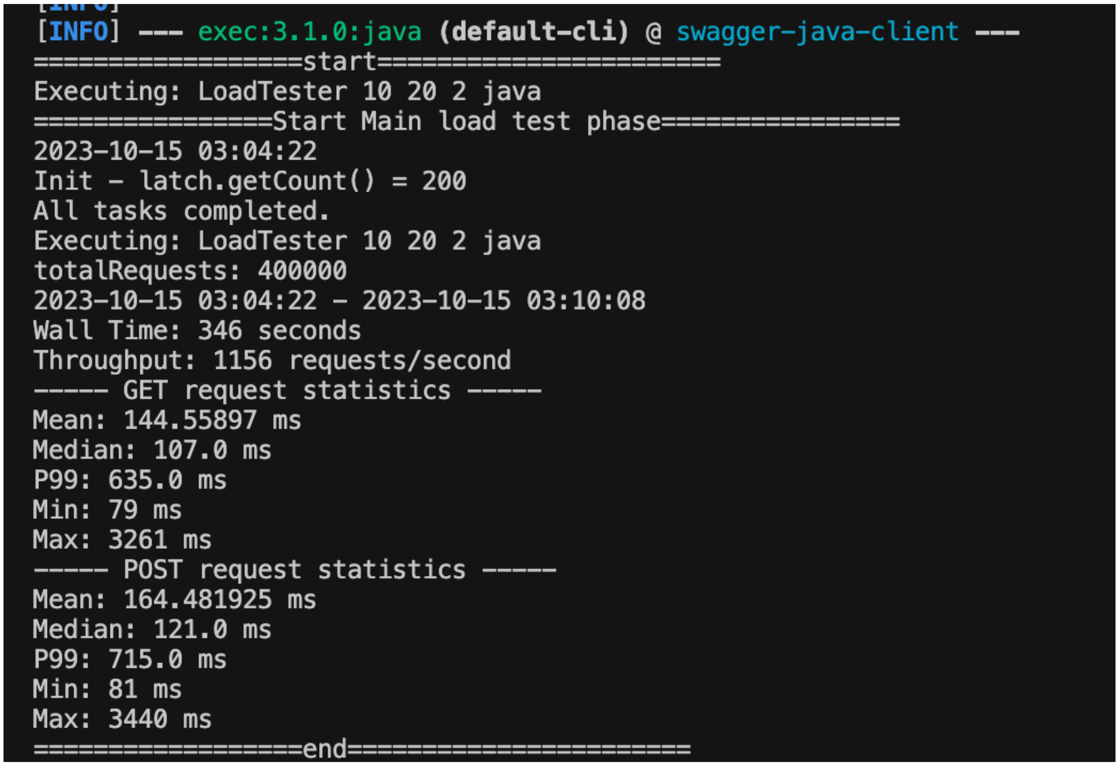
3: test results

Complete test results under /test_results folder
Here are some highlights:

Java Servlet

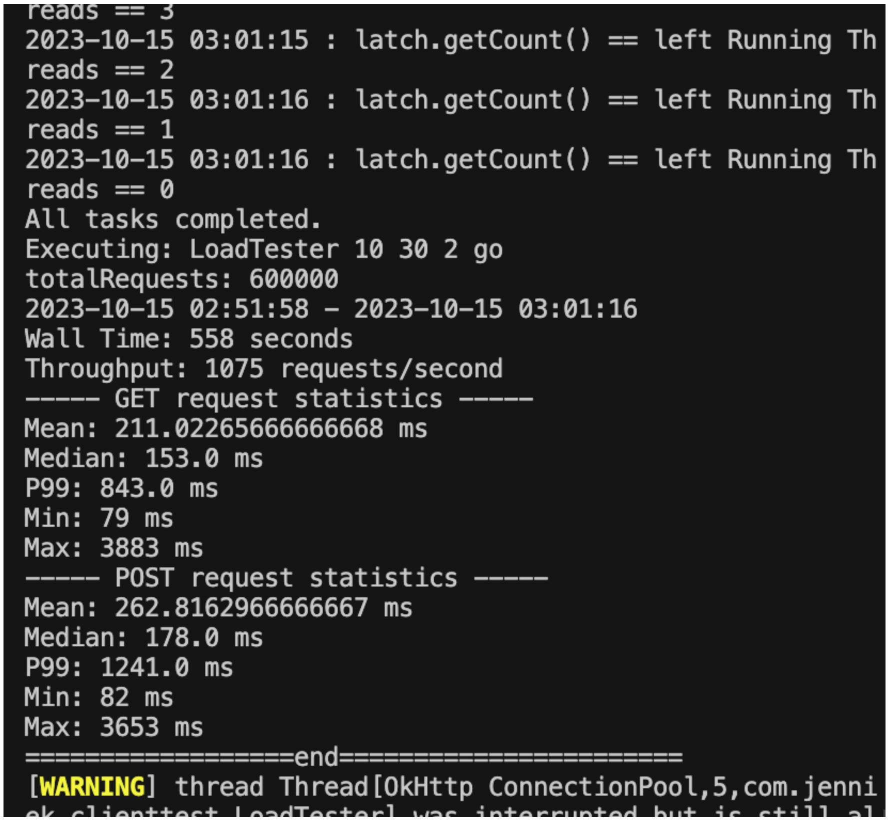


```
connection failed after 5 try
All tasks completed.
Executing: LoadTester 10 30 2 java
totalRequests: 600000
2023-10-15 03:15:27 - 2023-10-15 03:20:49
Wall Time: 322 seconds
Throughput: 1863 requests/second
----- GET request statistics -----
Mean: 126.86454272214114 ms
Median: 102.0 ms
P99: 465.0 ms
Min: 79 ms
Max: 1937 ms
----- POST request statistics -----
Mean: 141.25164 ms
Median: 113.0 ms
P99: 509.0 ms
Min: 81 ms
Max: 2598 ms
==================end=======================
```



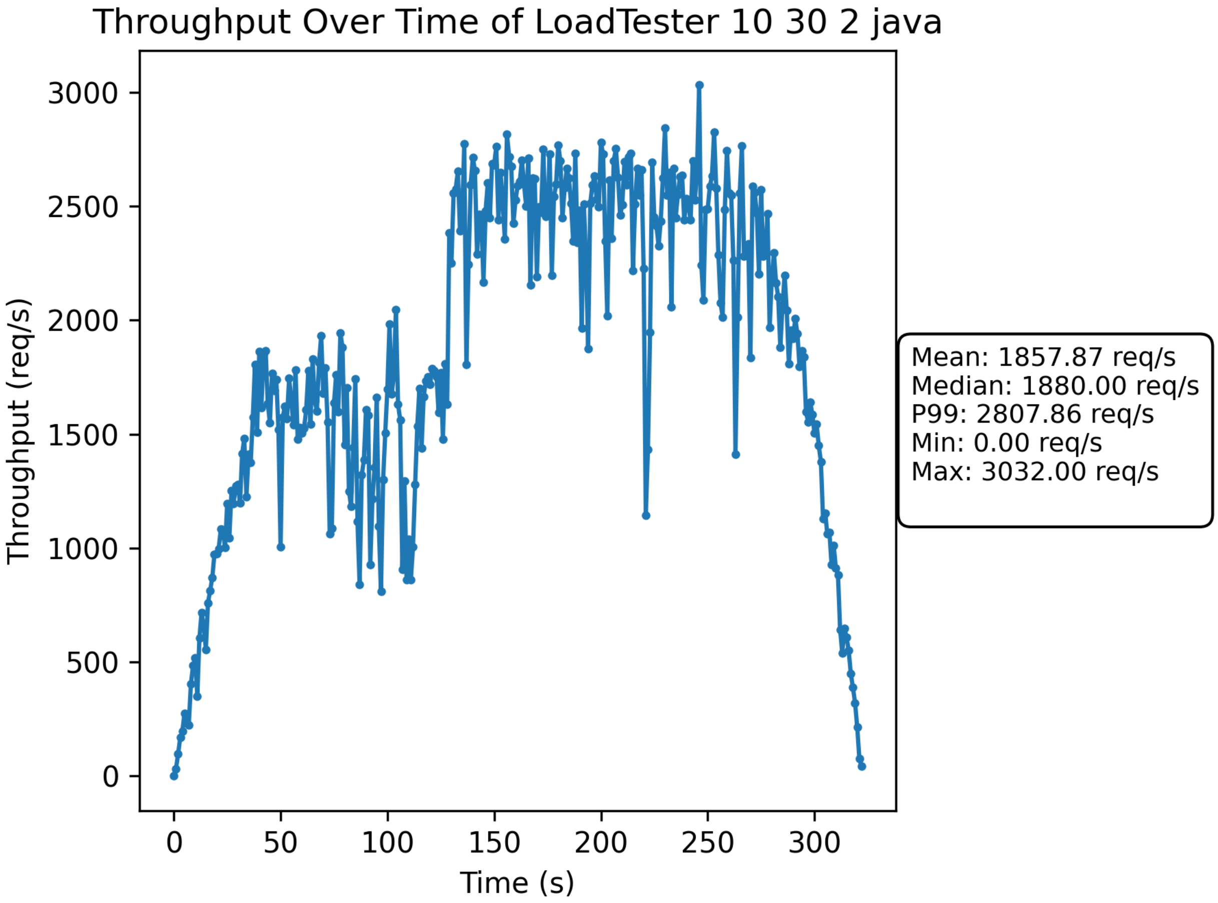
```
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ swagger-java-client ---
==================start=======================
Executing: LoadTester 10 20 2 java
================Start Main load test phase================
2023-10-15 03:04:22
Init - latch.getCount() = 200
All tasks completed.
Executing: LoadTester 10 20 2 java
totalRequests: 400000
2023-10-15 03:04:22 - 2023-10-15 03:10:08
Wall Time: 346 seconds
Throughput: 1156 requests/second
----- GET request statistics -----
Mean: 144.55897 ms
Median: 107.0 ms
P99: 635.0 ms
Min: 79 ms
Max: 3261 ms
----- POST request statistics -----
Mean: 164.481925 ms
Median: 121.0 ms
P99: 715.0 ms
Min: 81 ms
Max: 3440 ms
==================end=======================
```

Go Server



```
reads == 3
2023-10-15 03:01:15 : latch.getCount() == left Running Th
reads == 2
2023-10-15 03:01:16 : latch.getCount() == left Running Th
reads == 1
2023-10-15 03:01:16 : latch.getCount() == left Running Th
reads == 0
All tasks completed.
Executing: LoadTester 10 30 2 go
totalRequests: 600000
2023-10-15 02:51:58 - 2023-10-15 03:01:16
Wall Time: 558 seconds
Throughput: 1075 requests/second
----- GET request statistics -----
Mean: 211.02265666666668 ms
Median: 153.0 ms
P99: 843.0 ms
Min: 79 ms
Max: 3883 ms
----- POST request statistics -----
Mean: 262.8162966666667 ms
Median: 178.0 ms
P99: 1241.0 ms
Min: 82 ms
Max: 3653 ms
==================end=======================
[WARNING] thread Thread[OkHttp ConnectionPool,5,com.jenni
ek.clienttest.LoadTester] was interrupted but is still al
```

Plot of throughput by time



Appendix of Command

local

```
// ssh to aws ec2
/Users/may/Desktop/neu/cs6650_distributed/shortcuts/ssh_ec2.sh

  /* java servlet */
// local build java servlet in Maven project to generate the AlbumApp.war file
mvn clean install
// scp java servlet - AlbumApp.war to aws ec2
scp -i /Users/may/Downloads/ec2_2.pem /Users/may/Desktop/neu/cs6650_distributed/distributed-systems-work/hw1/java_servlet/AlbumApp/target/AlbumApp.war ec2-user@ec2-3-80-33-155.compute-1.amazonaws.com:~/tmp
```

```
  /* Go server */
// local run go server(macOS + m1 chip) for local tests
GOOS=darwin GOARCH=arm64 go run ./main.go
// local build go server for aws ec2(linux + arm64)
GOOS=linux GOARCH=arm64 go build -o server main.go
// scp go server - server to ec2 ~/emp to start
scp -i /Users/may/Downloads/ec2_2.pem /Users/may/Desktop/neu/cs6650_distributed/distributed-systems-work/hw1/go_server/go-server-server-generated/server ec2-user@ec2-3-80-33-155.compute-1.amazonaws.com:~/tmp

  /* client - LoadTester */
`Usage: LoadTester <threadGroupSize> <numThreadGroups> <delay> [java|go]`
// e.g.
mvn clean install && mvn exec:java -Dexec.mainClass="com.jenniek.clienttest.LoadTester" -Dexec.args="10 30 2 java"
```

aws ec2

```
// java servlet
sudo mv ~/tmp/AlbumApp.war /usr/share/tomcat/webapps

sudo systemctl restart tomcat
sudo systemctl stop tomcat
systemctl status tomcat

sudo vim /usr/share/tomcat/logs/catalina.out

// go server
~/tmp/server
```