

## 1 Output of Latency Tests:

1,2,3,4,5,6 client test. Demonstrates that the front end service runs as a thread per session approach

```
elinux2 lab2) > ./run.sh
```

6 Clients

Average Latency: 0.06642975409825642

Average Latency: 0.0655673370239841

Average Latency: 0.06710173083889868

Average Latency: 0.06719034231161769

Average Latency: 0.06815959551395515

Average Latency: 0.12863602046093908

5 Clients

Average Latency: 0.06557115198860705

Average Latency: 0.06556636417234266

Average Latency: 0.06744622704166695

Average Latency: 0.06754021455120567

Average Latency: 0.06453649597878781

4 Clients

Average Latency: 0.06528948909706539

Average Latency: 0.06650034197560557

Average Latency: 0.0646963519538009

Average Latency: 0.0660366082341416

3 Clients

Average Latency: 0.06735441152998012

Average Latency: 0.06579912925253109

Average Latency: 0.06582739546492293

2 Clients

Average Latency: 0.06510136855973138

Average Latency: 0.06597395420074463

1 Clients

Average Latency: 0.06502459407631868

```
elinux2 lab2) >
```

1,2,3,4,5 clients lookup only. hosted on docker VM:

```
elinux2 lab2) > ./run.sh
```

5 Clients

Average Latency: 0.06417081594467162

Average Latency: 0.06623378992080689

Average Latency: 0.06627068519592286

Average Latency: 0.06688476085662842

Average Latency: 0.0675485920906067

4 Clients

Average Latency: 0.06365367174148559

Average Latency: 0.06385019779205323

Average Latency: 0.06425481081008912

Average Latency: 0.06535226583480835

3 Clients

Average Latency: 0.0642075490951538  
Average Latency: 0.06439903497695923  
Average Latency: 0.06477782011032104

#### 2 Clients

Average Latency: 0.06292051792144776  
Average Latency: 0.06352087497711181

#### 1 Clients

Average Latency: 0.06307222604751588  
elinux2 lab2) >

1,2,3,4,5 clients trade only. hosted on docker VM:  
elinux2 lab2) > ./run.sh

#### 5 Clients

Average Latency: 0.06706074953079223  
Average Latency: 0.0675226902961731  
Average Latency: 0.06854172468185425  
Average Latency: 0.06932191610336304  
Average Latency: 0.07051957607269287

#### 4 Clients

Average Latency: 0.06947309017181397  
Average Latency: 0.07088325500488281  
Average Latency: 0.07140974044799804  
Average Latency: 0.07186569929122925

#### 3 Clients

Average Latency: 0.06828480958938599  
Average Latency: 0.06946743726730346  
Average Latency: 0.07202903032302857

#### 2 Clients

Average Latency: 0.06728380680084228  
Average Latency: 0.0683591890335083

#### 1 Clients

Average Latency: 0.07101038932800292  
elinux2 lab2) >

1,2,3,4,5 clients lookup and trade. hosted on docker VM. (p=0.5):  
elinux2 lab2) > ./run.sh

#### 5 Clients

Average Latency: 0.06674920133992929  
Average Latency: 0.06644911607106527  
Average Latency: 0.06634038647279045  
Average Latency: 0.06627201724362064  
Average Latency: 0.06847892780052989

#### 4 Clients

Average Latency: 0.06712995138433245  
Average Latency: 0.06614967445274453  
Average Latency: 0.0668612594728346  
Average Latency: 0.06853837626320976

#### 3 Clients

Average Latency: 0.06542890270551045  
Average Latency: 0.06382932598958879  
Average Latency: 0.06654680239689814

```
2 Clients
Average Latency: 0.0655783832847298
Average Latency: 0.06499463931108132
```

```
1 Clients
Average Latency: 0.0658747771891152
elinux2 lab2) >
```

```
1,2,3,4,5 clients lookup only. hosted on naive computer:
elinux2 lab2) > ./run.sh
```

```
5 Clients
Average Latency: 0.04435750246047974
Average Latency: 0.04439772605895996
Average Latency: 0.044757723808288574
Average Latency: 0.0460951566696167
Average Latency: 0.046717052459716794
```

```
4 Clients
Average Latency: 0.044660289287567136
Average Latency: 0.045778648853302004
Average Latency: 0.04627708911895752
Average Latency: 0.046452808380126956
```

```
3 Clients
Average Latency: 0.04330036163330078
Average Latency: 0.043649990558624265
Average Latency: 0.044895312786102294
```

```
2 Clients
Average Latency: 0.04294870138168335
Average Latency: 0.044201779365539554
```

```
1 Clients
Average Latency: 0.04136239051818848
```

```
1,2,3,4,5 clients trade only. hosted on naive computer:
elinux2 lab2) > ./run.sh
```

```
5 Clients
Average Latency: 0.05693872928619385
Average Latency: 0.05730640649795532
Average Latency: 0.057508246898651125
Average Latency: 0.058334386348724364
Average Latency: 0.06005401849746704
```

```
4 Clients
Average Latency: 0.0473453426361084
Average Latency: 0.051038458347320556
Average Latency: 0.05205904722213745
Average Latency: 0.05213435173034668
```

```
3 Clients
Average Latency: 0.047065963745117186
Average Latency: 0.04709573984146118
Average Latency: 0.04826335668563843
```

```
2 Clients
Average Latency: 0.04532717704772949
Average Latency: 0.047648122310638426
```

```
1 Clients
Average Latency: 0.044635841846466066
elinux2 lab2) >
```

```
1,2,3,4,5 clients lookup and trade. hosted on naive computer. (p=0.5):
elinux2 lab2) > ./run.sh
```

```
5 Clients
Average Latency: 0.045236916769118535
Average Latency: 0.04724589439287578
Average Latency: 0.04727411106841205
Average Latency: 0.04685761439089744
Average Latency: 0.04726266553325038
```

```
4 Clients
Average Latency: 0.046906200244272354
Average Latency: 0.04515150981613352
Average Latency: 0.04841959005907962
Average Latency: 0.050258150287702974
```

```
3 Clients
Average Latency: 0.04841995392090235
Average Latency: 0.048825728587615184
Average Latency: 0.05061060071780982
```

```
2 Clients
Average Latency: 0.0450793263074514
Average Latency: 0.04512343956873967
```

```
1 Clients
Average Latency: 0.04427638730487308
elinux2 lab2) >
```

## 2 Output of Test Cases:

```
C:\Users\thewi\Desktop\CS677\lab2\src\test-cases>py testcases.py 10.0.0.246 56893
#####
##Test Case 1 - Lookup Functionality: ##
#####
Input Request: GET /stocks/nvidia
JSON Reply: {'data': {'name': 'nvidia', 'price': 240.63, 'quantity': 880}}

#####
##Test Case 2 - Lookup Error Handling (Stock not found): ##
#####
Input Request: GET /stocks/imaginarycompany
JSON Reply: {'error': {'code': 404, 'message': 'stock not found'}}

#####
##Test Case 3 - Sell Functionality: ##
#####
Input Request: POST /orders
Input JSON: {"name": "intel", "type": "sell", "quantity": 10}
Stock Information before Request: {'data': {'name': 'intel', 'price': 28.01, 'quantity': 1399}}
JSON Reply: {'data': {'transaction_number': 278}}
Stock Information after Request: {'data': {'name': 'intel', 'price': 28.01, 'quantity': 1409}}

#####
##Test Case 4 - Buy Functionality: ##
```

```
#####
Input Request: POST /orders
Input JSON: {"name": "apple", "type": "buy", "quantity": 10}
Stock Information before Request: {'data': {'name': 'apple', 'price': 152.59, 'quantity': 959}}
JSON Reply: {'data': {'transaction_number': 279}}
Stock Information after Request: {'data': {'name': 'apple', 'price': 152.59, 'quantity': 949}}

#####
##Test Case 5 - Buy Error Handling (Amount to buy > num of stocks avail.): ##
#####
Input Request: POST /orders
Input JSON: {"name": "ford", "type": "buy", "quantity": 10000000}
JSON Reply: {'error': {'code': 404, 'message': 'not enough stocks available to buy'}}

#####
##Test Case 6 - Buy/Sell Error Handling (Invalid quantity of stocks): ##
#####
Note: PROTO already defines quantity needs to be an int. As such, the only invalid value will be negat
Input Request: POST /orders
Input JSON: {"name": "amazon", "type": "buy", "quantity": -1000}
JSON Reply: {'error': {'code': 404, 'message': 'invalid number of stocks'}}

Input Request: POST /orders
Input JSON: {"name": "amazon", "type": "sell", "quantity": -1000}
JSON Reply: {'error': {'code': 404, 'message': 'invalid number of stocks'}}

#####
##Test Case 7 - Buy/Sell Error Handling (Invalid request type): ##
#####
Input Request: POST /orders
Input JSON: {"name": "meta", "type": "trade", "quantity": 100}
JSON Reply: {'error': {'code': 400, 'message': 'invalid request type'}}

#####
##Test Case 8 - Buy/Sell Error Handling (Stock not found): ##
#####
Input Request: POST /orders
Input JSON: {"name": "imaginarycompany", "type": "buy", "quantity": 100}
JSON Reply: {'error': {'code': 404, 'message': 'stock not found'}}

C:\Users\thewi\Desktop\CS677\lab2\src\test-cases>
```

### 3 Native Host Example:

```
Command Prompt - py catalogserver.py
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\thewi>cd desktop/cs677/lab2/src/catalog-service

C:\Users\thewi\Desktop\CS677\lab2\src\catalog-service>py catalogserver.py
Server started on: 10.0.0.246:56892

Command Prompt - py orderserver.py
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\thewi>cd desktop/cs677/lab2/src/order-service

C:\Users\thewi\Desktop\CS677\lab2\src\order-service>py orderserver.py
Server started on: 10.0.0.246:56891

Command Prompt - py frontend.py
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\thewi>cd desktop/cs677/lab2/src/front-end

C:\Users\thewi\Desktop\CS677\lab2\src\front-end>py frontend.py
C:\Users\thewi\AppData\Local\Programs\Python\Python310\python.exe: can't open file 'C:\\Users\\thewi\\Desktop\\CS677\\lab2\\src\\front-end\\frontend.py': [Errno 2] No such file or directory

C:\Users\thewi\Desktop\CS677\lab2\src\front-end>py frontend.py
http server is starting...
Server started on: 10.0.0.246:56893
http server is running...
```

Figure 1: Output of the initialized microservices.

```
Command Prompt - py frontend.py
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\thewi>cd desktop/cs677/lab2/src/front-end

C:\Users\thewi\Desktop\CS677\lab2\src\front-end>py frontend.py
C:\Users\thewi\AppData\Local\Programs\Python\Python310\python.exe: can't open file 'C:\\Users\\thewi\\Desktop\\CS677\\lab2\\src\\front-end\\frontend.py': [Errno 2] No such file or directory

http server is starting...
Server started on: 10.0.0.246:56893
http server is running...

128.119.243.168 - [24/Mar/2023 18:35:41] "GET /stocks/amazon HTTP/1.1" 200 -
128.119.243.168 - [24/Mar/2023 18:35:41] "GET /stocks/imaginarycompany HTTP/1.1" 404 -
128.119.243.168 - [24/Mar/2023 18:35:41] "GET /stocks/apple HTTP/1.1" 200 -
128.119.243.168 - [24/Mar/2023 18:35:41] "POST /orders HTTP/1.1" 400 -
128.119.243.168 - [24/Mar/2023 18:35:41] "GET /stocks/apple HTTP/1.1" 200 -
128.119.243.168 - [24/Mar/2023 18:35:41] "POST /orders HTTP/1.1" 200 -
128.119.243.168 - [24/Mar/2023 18:35:41] "GET /stocks/intel HTTP/1.1" 200 -
128.119.243.168 - [24/Mar/2023 18:35:41] "POST /orders HTTP/1.1" 200 -
128.119.243.168 - [24/Mar/2023 18:35:41] "GET /stocks/imaginarycompany HTTP/1.1" 404 -
128.119.243.168 - [24/Mar/2023 18:35:41] "POST /orders HTTP/1.1" 400 -
128.119.243.168 - [24/Mar/2023 18:35:41] "GET /stocks/amazon HTTP/1.1" 200 -
128.119.243.168 - [24/Mar/2023 18:35:41] "POST /orders HTTP/1.1" 400 -
128.119.243.168 - [24/Mar/2023 18:35:41] "GET /stocks/ford HTTP/1.1" 200 -
128.119.243.168 - [24/Mar/2023 18:35:42] "GET /stocks/amazon HTTP/1.1" 200 -

OpenSSH SSH client
eLinux client) > python3 HTTPClient.py 73.186.87.78 56893 0.5 0
Input: amazon
Output: {'data': {'name': 'amazon', 'price': 94.88, 'quantity': 1154}}

----- Lookup -----
Input: imaginarycompany
Output: {'error': {'code': 404, 'message': 'stock not found'}}

----- Lookup -----
Input: apple
Output: {'data': {'name': 'apple', 'price': 152.59, 'quantity': 692}}
Trade -----
Input: {'name': 'apple', 'type': 'trade', 'quantity': 41}
Output: {'error': {'code': 400, 'message': 'invalid request type'}}

----- Lookup -----
Input: apple
Output: {'data': {'name': 'apple', 'price': 152.59, 'quantity': 692}}
Trade -----
Input: {'name': 'apple', 'type': 'sell', 'quantity': 13}
Output: {'data': {'transaction_number': 334}}

----- Lookup -----
Input: intel
Output: {'data': {'name': 'intel', 'price': 28.01, 'quantity': 1396}}
Trade -----
Input: {'name': 'intel', 'type': 'buy', 'quantity': 87}
Output: {'data': {'transaction_number': 335}}

----- Lookup -----
Input: apple
Output: {'data': {'name': 'apple', 'price': 152.59, 'quantity': 705}}
Trade -----
Input: {'name': 'apple', 'type': 'trade', 'quantity': -9}
Output: {'error': {'code': 400, 'message': 'invalid request type'}}

----- Lookup -----
Input: imaginarycompany
Output: {'error': {'code': 404, 'message': 'stock not found'}}
Trade -----
Input: {'name': 'imaginarycompany', 'type': 'buy', 'quantity': 62}
Output: {'error': {'code': 404, 'message': 'stock not found'}}

----- Lookup -----
Input: amazon
Output: {'data': {'name': 'amazon', 'price': 94.88, 'quantity': 1154}}
Trade -----
Input: {'name': 'amazon', 'type': 'trade', 'quantity': 20}
Output: {'error': {'code': 400, 'message': 'invalid request type'}}

----- Lookup -----
Input: ford
Output: {'data': {'name': 'ford', 'price': 11.93, 'quantity': 938}}
```

Figure 2: Example output of the stock service. Front end hosted natively on left. SSH Edlab client connecting to the server on the right.

## 4 Docker Host Example:

```
cshi@DESKTOP-00162FU: /mnt/c/users/thewi/desktop/cs677/lab2/src
connection refused
cschi@DESKTOP-00162FU:/mnt/c/users/thewi/desktop/cs677/lab2/src$ ./build.sh
Sending build context to Docker daemon 150kB
Step 1/5 : FROM python:3.8-alpine
--> 0e47cc44f093
Step 2/5 : RUN pip install flask redis grpcio grpcio-tools
--> Using cache
--> 910180275657
Step 3/5 : WORKDIR /app/catalog-service
--> Using cache
--> ee0f4ec30e70
Step 4/5 : COPY ./catalog-service /app/catalog-service
--> 24a612411de4
Step 5/5 : ENTRYPOINT ["python", "-u", "catalogServer.py"]
--> Running in a7382eb437d3
Removing intermediate container a7382eb437d3
--> 1287aa88c62f
Successfully built 1287aa88c62f
Sending build context to Docker daemon 150kB
Step 1/5 : FROM python:3.8-alpine
--> 0e47cc44f093
Step 2/5 : RUN pip install flask redis grpcio grpcio-tools
--> Using cache
--> 910180275657
Step 3/5 : WORKDIR /app/order-service
--> Using cache
--> bd1985e0d108
Step 4/5 : COPY ./order-service /app/order-service
--> 6ba3580f980f
Step 5/5 : ENTRYPOINT ["python", "-u", "orderServer.py"]
--> Running in 38d2c95733cf
Removing intermediate container 38d2c95733cf
--> 978891d54a61
Successfully built 978891d54a61
Sending build context to Docker daemon 150kB
Step 1/5 : FROM python:3.8-alpine
--> 0e47cc44f093
Step 2/5 : RUN pip install flask redis grpcio grpcio-tools
--> Using cache
--> 910180275657
Step 3/5 : WORKDIR /app/front-end
--> Using cache
--> 1e87345cdb9b
Step 4/5 : COPY ./front-end /app/front-end
--> 578d849a0090
Step 5/5 : ENTRYPOINT ["python", "-u", "FrontEnd.py"]
--> Running in 6ab8ba201f02
Removing intermediate container 6ab8ba201f02
--> f4cc577b9ced
Successfully built f4cc577b9ced
cschi@DESKTOP-00162FU:/mnt/c/users/thewi/desktop/cs677/lab2/src$
```

Figure 3: Output of the build.sh file. build.sh builds the images for the dockerfiles corresponding to each of the microservices.

```
cshi@DESKTOP-00162FU: /mnt/c/users/thewi/desktop/cs677/lab2/src
cschi@DESKTOP-00162FU:/mnt/c/users/thewi/desktop/cs677/lab2/src$ docker-compose up
[+] Running 3/3
  Container src-frontend-1 Created                                0.4s
  Container src-order-1    Created                                0.4s
  Container src-catalog-1  Created                                0.4s
Attaching to src-catalog-1, src-frontend-1, src-order-1
src-catalog-1 | Server started on: 10.0.0.242:56892
src-frontend-1 | http server is starting...
src-order-1    | Server started on: 10.0.0.241:56891
src-frontend-1 | Server started on: 10.0.0.243:56893
src-frontend-1 | http server is running...
```

Figure 4: Output of docker-compose up.

```
cs@DESKTOP-00162FPU: /mnt/c/users/theu/desktop/cs677/lab2/src$ docker-compose down
[+] Running 3/3
  Container src-order-1   Stopped
  Container src-catalog-1 Stopped
  Container src-frontend-1 Stopped
canceled
cs@DESKTOP-00162FPU: /mnt/c/users/theu/desktop/cs677/lab2/src$ docker-compose down
[+] Running 4/4
  Container src-order-1   Removed
  Container src-catalog-1 Removed
  Container src-frontend-1 Removed
  Network src_vpcbr      Removed
cs@DESKTOP-00162FPU: /mnt/c/users/theu/desktop/cs677/lab2/src$

Attaching to src-catalog-1, src-frontend-1, src-order-1
src-catalog-1 | Server started on: 10.0.0.242:56892
src-frontend-1 | http server is starting...
src-order-1 | Server started on: 10.0.0.241:56891
src-frontend-1 | Server started on: 10.0.0.243:56893
src-frontend-1 | http server is running...
10.0.0.1 - - [24/Mar/2023 22:32:47] "GET /stocks/nvidia HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:47] "POST /orders HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:47] "GET /stocks/apple HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "GET /stocks/intel HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "POST /orders HTTP/1.1" 400 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "GET /stocks/intel HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "POST /orders HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "GET /stocks/imaginarycompany HTTP/1.1" 404 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "POST /orders HTTP/1.1" 404 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "GET /stocks/amazon HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "GET /stocks/nvidia HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "POST /orders HTTP/1.1" 400 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "GET /stocks/meta HTTP/1.1" 200 -
10.0.0.1 - - [24/Mar/2023 22:32:48] "GET /stocks/apple HTTP/1.1" 200 -
^CGracefully stopping... (press Ctrl+C again to force)
Aborting on container exit...
[+] Running 3/3
  Container src-frontend-1 Stopped
  Container src-order-1   Stopped
  Container src-catalog-1 Stopped
canceled
cs@DESKTOP-00162FPU: /mnt/c/users/theu/desktop/cs677/lab2/src$
```

```
OpenSSH SSH client
Output: {'data': {'name': 'nvidia', 'price': 240.6300048828125, 'quantity': 802}}
----- Trade -----
Input: {'name': 'nvidia', 'type': 'sell', 'quantity': 25}
Output: {'data': {'transaction_number': 332}}

----- Lookup -----
Input: apple
Output: {'data': {'name': 'apple', 'price': 152.58999633789062, 'quantity': 692}}

----- Lookup -----
Input: intel
Output: {'data': {'name': 'intel', 'price': 28.010000228881836, 'quantity': 1456}}
----- Trade -----
Input: {'name': 'intel', 'type': 'trade', 'quantity': 50}
Output: {'error': {'code': 400, 'message': 'invalid request type'}}

----- Lookup -----
Input: intel
Output: {'data': {'name': 'intel', 'price': 28.010000228881836, 'quantity': 1456}}
----- Trade -----
Input: {'name': 'intel', 'type': 'buy', 'quantity': 60}
Output: {'data': {'transaction_number': 333}}

----- Lookup -----
Input: imaginarycompany
Output: {'error': {'code': 404, 'message': 'stock not found'}}
----- Trade -----
Input: {'name': 'imaginarycompany', 'type': 'trade', 'quantity': 91}
Output: {'error': {'code': 404, 'message': 'stock not found'}}

----- Lookup -----
Input: amazon
Output: {'data': {'name': 'amazon', 'price': 94.87999725341797, 'quantity': 1154}}

----- Lookup -----
Input: nvidia
Output: {'data': {'name': 'nvidia', 'price': 240.6300048828125, 'quantity': 827}}
----- Trade -----
Input: {'name': 'nvidia', 'type': 'trade', 'quantity': 53}
Output: {'error': {'code': 400, 'message': 'invalid request type'}}

----- Lookup -----
Input: meta
Output: {'data': {'name': 'meta', 'price': 194.02000427246094, 'quantity': 1200}}

----- Lookup -----
Output: {'data': {'name': 'meta', 'price': 194.02000427246094, 'quantity': 1200}}

----- Lookup -----
Input: apple
Output: {'data': {'name': 'apple', 'price': 152.58999633789062, 'quantity': 692}}

Average Latency: 0.06417187854951985
linux3 client>
```

Figure 5: Example output of the stock service. Front end hosted on a Docker container on left. SSH Edlab client connecting to the server on the right.