

Test Output Document

By: Aishwarya Malgonde, Sahil Jindal

Course: CS 677 | Professor Prashant Shenoy

Lab 3: Asterix and Double Trouble -- Replication, Caching, and Fault Tolerance

April 29, 2023

Following are the images attached for test cases -

Frontend Service

```
(venv) aishwarya@Aishwaryas-MacBook-Air-2 tests % python3 test_frontend_service.py

Testing GET requests for CACHE INVALIDATION

GET - http://0.0.0.0:8000/cache_invalid/FishC - 200 - {"data": "cache invalidation successful"}
GET - http://0.0.0.0:8000/cache_invald/FishCo - 404 - {"error": {"code": 404, "message": "Invalid request url"}}

All GET requests ran successfully which expected responses for CACHE INVALIDATION
.
Testing GET requests for STOCK LOOKUP

GET - http://0.0.0.0:8000/stocks/BoarCo - 200 - {"data": {"name": "BoarCo", "price": 28.59, "quantity": 1102}}
GET - http://0.0.0.0:8000/stocks/Ash - 400 - {"error": {"code": 400, "message": "Bad request. Ash stock not found in db"}}
GET - http://0.0.0.0:8000/stcks/BoarCo - 404 - {"error": {"code": 404, "message": "Invalid request url"}}

All GET requests ran successfully which expected responses for STOCK LOOKUP
.
Testing GET requests for ORDER LEADER

GET - http://0.0.0.0:8000/get_order_leader - 200 - {"data": {"order_leader": 9092}}
GET - http://0.0.0.0:8000/get_order_lader - 404 - {"error": {"code": 404, "message": "Invalid request url"}}

All GET requests ran successfully which expected responses for ORDER LEADER
.
Testing GET requests for ORDER QUERY

GET - http://0.0.0.0:8000/orders/1 - 200 - {"data": {"number": 1, "name": "MenhirCo", "type": "buy", "quantity": 18}}
GET - http://0.0.0.0:8000/orders/1ac - 404 - {"error": {"code": 404, "message": "Invalid request url"}}
GET - http://0.0.0.0:8000/ordrs/2 - 404 - {"error": {"code": 404, "message": "Invalid request url"}}

All GET requests ran successfully which expected responses for ORDER QUERY
.
Testing POST requests

POST - 200 - {"data": {"transaction_number": 27}}
POST - 400 - {"error": {"code": 400, "message": "Bad request. Hey stock not found in db"}}
POST - 400 - {"error": {"code": 400, "message": "Invalid quantity -10"}}
POST - 400 - {"error": {"code": 400, "message": "Unknow trade type - negotiate"}}
POST - 404 - {"error": {"code": 404, "message": "Invalid request url"}}
POST - 500 - {"error": {"code": 500, "message": "Insufficient volume for GameStart stock"}}

All POST requests ran successfully with expected responses
.
-----
Ran 5 tests in 0.129s

OK
```

Catalog Service

```

• (venv) aishwarya@aishwaryas-MacBook-Air-2 tests % python3 test_catalog_service.py

Testing Catalog Lookup requests

Lookup - FishCo - 200 - {'data': {'name': 'FishCo', 'price': 17.99, 'quantity': 600}}
Lookup - Ash - 400 - Bad request. Ash stock not found in db

All Catalog Lookup requests ran successfully which expected responses
.

Testing Catalog Update requests

Update - FishCo, 10 - 200 - success
Update - FishCo, LOL - 500 - ufunc 'add' did not contain a loop with signature matching types (dtype('int64'), dtype('<U3')) -> None
Update - Meh, 10 - 400 - Stock name not found

All Catalog Update requests ran successfully with expected responses
.
=====
Ran 2 tests in 0.043s

OK

```

Order Service

[illegible]

Part 1: Caching

Below is the screenshot of frontend logs. We can see that the frontend service has started.

The first block of logs show that the cache response was none for a lookup request at /stocks/FishCo. This lookup response is added to cache.

The second block of logs shows that for the same stock lookup request, the response is gathered from cache.

```
MainThread : Calling health_check service for 9094 order
MainThread : Health check failed for 9094 order
MainThread : Calling health_check service for 9092 order
MainThread : Order service leader found - 9092 order
MainThread : Notifying 9090 replica of order leader
MainThread : Notifying 9092 replica of order leader
MainThread : Notifying 9094 replica of order leader
MainThread : Cannot reach 9094 replica - nameserver down
MainThread : Starting threaded httpd server on 0.0.0.0:8000

Thread-1 (process_request_thread) : Received GET request on url /stocks/FishCo
Thread-1 (process_request_thread) : Checking Cache
Thread-1 (process_request_thread) : Cache response - None
Thread-1 (process_request_thread) : Calling lookup service
Thread-1 (process_request_thread) : Adding lookup response to cache
Thread-1 (process_request_thread) : Received lookup response - 200 {'data': {'name': 'FishCo', 'price': 17.99, 'quantity': 518}}
127.0.0.1 - - [01/May/2023 15:15:06] "GET /stocks/FishCo HTTP/1.1" 200 -
Thread-1 (process_request_thread) : Finished GET request

Thread-1 (process_request_thread) : Received GET request on url /stocks/FishCo
Thread-1 (process_request_thread) : Checking Cache
Thread-1 (process_request_thread) : Cache response - b'{"data": {"name": "FishCo", "price": 17.99, "quantity": 518}}'
Thread-1 (process_request_thread) : Getting cache response
Thread-1 (process_request_thread) : Got lookup response from cache - {'data': {'name': 'FishCo', 'price': 17.99, 'quantity': 518}}
Thread-1 (process_request_thread) : Received lookup response - 200 {'data': {'name': 'FishCo', 'price': 17.99, 'quantity': 518}}
127.0.0.1 - - [01/May/2023 15:15:06] "GET /stocks/FishCo HTTP/1.1" 200 -
Thread-1 (process_request_thread) : Finished GET request
```

Below is the screenshot of frontend logs. We can see in the first block of logs that, while the POST request for order trade of FishCo stock is being processed, the frontend receives a GET request for cache invalidation. And the POST request completes successfully after that.

Then we get a lookup request at /stocks/FishCo, but the information is not extracted from the cache. This lookup response is added to cache.

Lastly, another lookup request is sent at /stocks/FishCo, and here we can see that the response is gathered from cache.

```
Thread-1 (process_request_thread) : Received POST request on url /orders
Thread-1 (process_request_thread) : data - {'name': 'FishCo', 'quantity': 10, 'type': 'sell'} <class 'dict'>
Thread-1 (process_request_thread) : Calling trade service
Thread-2 (process_request_thread) : Received GET request on url /cache_invalid/FishCo
Thread-2 (process_request_thread) : Making Cache invalid
Thread-2 (process_request_thread) : Cache Invalidated in memory
127.0.0.1 - - [01/May/2023 15:15:06] "GET /cache_invalid/FishCo HTTP/1.1" 200 -
Thread-2 (process_request_thread) : Finished GET request

Thread-1 (process_request_thread) : Received trade response - 200 {'data': {'transaction_number': 8}}
127.0.0.1 - - [01/May/2023 15:15:06] "POST /orders HTTP/1.1" 200 -
Thread-1 (process_request_thread) : Finished POST request

Thread-1 (process_request_thread) : Received GET request on url /stocks/FishCo
Thread-1 (process_request_thread) : Checking Cache
Thread-1 (process_request_thread) : Cache response - None
Thread-1 (process_request_thread) : Calling lookup service
Thread-1 (process_request_thread) : Adding lookup response to cache
Thread-1 (process_request_thread) : Received lookup response - 200 {'data': {'name': 'FishCo', 'price': 17.99, 'quantity': 528}}
127.0.0.1 - - [01/May/2023 15:15:06] "GET /stocks/FishCo HTTP/1.1" 200 -
Thread-1 (process_request_thread) : Finished GET request

Thread-1 (process_request_thread) : Received GET request on url /stocks/FishCo
Thread-1 (process_request_thread) : Checking Cache
Thread-1 (process_request_thread) : Cache response - b'{"data": {"name": "FishCo", "price": 17.99, "quantity": 528}}'
Thread-1 (process_request_thread) : Getting cache response
Thread-1 (process_request_thread) : Got lookup response from cache - {'data': {'name': 'FishCo', 'price': 17.99, 'quantity': 528}}
Thread-1 (process_request_thread) : Received lookup response - 200 {'data': {'name': 'FishCo', 'price': 17.99, 'quantity': 528}}
127.0.0.1 - - [01/May/2023 15:15:06] "GET /stocks/FishCo HTTP/1.1" 200 -
Thread-1 (process_request_thread) : Finished GET request
```

Part 2: Replication

Case 1:

Replicas running - 9094, 9092, 9090

On frontend startup, we can see that it chooses 9094 as the leader.

```
umass-cs677-current > lab-3-asterix-and-double-trouble-sahil-n-ash > src > logs > ≡ frontend.log
1 MainThread : Redis config : {'maxmemory': '10485760'} - {'maxmemory-policy': 'volatile-ttl'}
2 MainThread : Calling ping (health check) service for 9094 order
3 MainThread : Order service leader found - 9094 order
4 MainThread : Notifying 9090 replica of order leader
5 MainThread : Notifying 9092 replica of order leader
6 MainThread : Notifying 9094 replica of order leader
7 MainThread : Starting threaded httpd server on 0.0.0.0:8000
8
```

The frontend has chosen 9094 as the leader, and the below image shows leader processing requests. And the replicas saving the order details forwarded by the leader.

Leader - 9094

```
umass-cs677-current > lab-3-asterix-and-double-trouble-sahil-n-ash > src > logs > ≡ order_9094.log
1 MainThread : Existing Order Log File Found.
2 MainThread : TXN_NUM is initialized as 10
3 MainThread : Loading order log
4 MainThread : Order replica just started. Trying to sync
5 MainThread : Checking for leader at startup time
6 MainThread : Starting new HTTP connection (1): 0.0.0.0:8000
7 MainThread : NewConnectionError - Aborting - Frontend is not up!
8 MainThread : No leader found at startup time, no sync required
9 MainThread : Order service ready
10 Pyro-Worker-4684656784 : This process is the leader of the replicated order pool - Flag is True
11 Pyro-Worker-4684648784 : Received trade request - {'name': 'FishCo', 'quantity': 43, 'type': 'sell'} <class 'dict'>
12 Pyro-Worker-4684648784 : Calling lookup service - FishCo
13 Pyro-Worker-4684648784 : Received lookup response - 200 {'data': {'name': 'FishCo', 'price': 17.99, 'quantity': 506}}
14 Pyro-Worker-4684648784 : FishCo present in db
15 Pyro-Worker-4684648784 : Calling update service - FishCo
16 Pyro-Worker-4684648784 : Received update response - FishCo
17 Pyro-Worker-4684648784 : Write lock - acquired
18 Pyro-Worker-4684648784 : Write lock - released
19 Pyro-Worker-4684648784 : Notifying 9092 replica of order leader - txn_num 11
20 Pyro-Worker-4684648784 : Response from replica - OK
21 Pyro-Worker-4684648784 : Notifying 9090 replica of order leader - txn_num 11
22 Pyro-Worker-4684648784 : Response from replica - OK
23 Pyro-Worker-4684648784 : Executing - trade_code 200 <class 'int'> - response {'data': {'transaction_number': 11}} <class 'dict'>
24 Pyro-Worker-4684648784 : Executing sell trade for FishCo with 43 quantity
25 Pyro-Worker-4684655440 : Received trade request - {'name': 'BoarCo', 'quantity': 6, 'type': 'sell'} <class 'dict'>
26 Pyro-Worker-4684655440 : Calling lookup service - BoarCo
27 Pyro-Worker-4684655440 : Received lookup response - 200 {'data': {'name': 'BoarCo', 'price': 28.59, 'quantity': 992}}
28 Pyro-Worker-4684655440 : BoarCo present in db
29 Pyro-Worker-4684655440 : Calling update service - BoarCo
30 Pyro-Worker-4684655440 : Received update response - BoarCo
31 Pyro-Worker-4684655440 : Write lock - acquired
32 Pyro-Worker-4684655440 : Write lock - released
33 Pyro-Worker-4684655440 : Notifying 9092 replica of order leader - txn_num 12
34 Pyro-Worker-4684655440 : Response from replica - OK
35 Pyro-Worker-4684655440 : Notifying 9090 replica of order leader - txn_num 12
```

Replica 9090

```

umass-cs677-current > lab-3-asterix-and-double-trouble-sahil-n-ash > src > logs > ≡ order_9090.log
1  MainThread : Existing Order Log File Found.
2  MainThread : TXN_NUM is initialized as 10
3  MainThread : Loading order log
4  MainThread : Order replica just started. Trying to sync
5  MainThread : Checking for leader at startup time
6  MainThread : Starting new HTTP connection (1): 0.0.0.0:8000
7  MainThread : NewConnectionError - Aborting - Frontend is not up!
8  MainThread : No leader found at startup time, no sync required
9  MainThread : Order service ready
10 Pyro-Worker-4841943952 : This process is a replica - Flag is False
11 Pyro-Worker-4841944656 : Writing trade data from leader - {'transaction_number': 11, 'stock_name': 'FishCo
12 Pyro-Worker-4841944656 : Write lock - acquired
13 Pyro-Worker-4841944656 : Write lock - released
14 Pyro-Worker-4841944656 : TXN_NUM is - 11
15 Pyro-Worker-4692063312 : Writing trade data from leader - {'transaction_number': 12, 'stock_name': 'Boar
16 Pyro-Worker-4692063312 : Write lock - acquired
17 Pyro-Worker-4692063312 : Write lock - released
18 Pyro-Worker-4692063312 : TXN_NUM is - 12
19

```

Replica 9092

```

umass-cs677-current > lab-3-asterix-and-double-trouble-sahil-n-ash > src > logs > ≡ order_9092.log
1  MainThread : Existing Order Log File Found.
2  MainThread : TXN_NUM is initialized as 10
3  MainThread : Loading order log
4  MainThread : Order replica just started. Trying to sync
5  MainThread : Checking for leader at startup time
6  MainThread : Starting new HTTP connection (1): 0.0.0.0:8000
7  MainThread : NewConnectionError - Aborting - Frontend is not up!
8  MainThread : No leader found at startup time, no sync required
9  MainThread : Order service ready
10 Pyro-Worker-4561260176 : This process is a replica - Flag is False
11 Pyro-Worker-4561268432 : Writing trade data from leader - {'transaction_number': 11, 'stock_name': 'FishCo
12 Pyro-Worker-4561268432 : Write lock - acquired
13 Pyro-Worker-4561268432 : Write lock - released
14 Pyro-Worker-4561268432 : TXN_NUM is - 11
15 Pyro-Worker-4561267728 : Writing trade data from leader - {'transaction_number': 12, 'stock_name': 'BoarCo
16 Pyro-Worker-4561267728 : Write lock - acquired
17 Pyro-Worker-4561267728 : Write lock - released
18 Pyro-Worker-4561267728 : TXN_NUM is - 12
19

```

Case 2

Replicas running - 9092, 9090; 9094 is not running

On frontend startup, we can see that the health check to replica 9094, fails. Next the frontend pings replica 9092 and the health check is successful, and chooses it as the leader. And then notifies all the replicas about it.

```
umass-cs677-current > lab-3-asterix-and-double-trouble-sahil-n-ash > src > logs > ≡ frontend.log
1   MainThread : Redis config : {'maxmemory': '10485760'} - {'maxmemory-policy': 'volatile-ttl'}
2   MainThread : Calling ping (health check) service for 9094 order
3   MainThread : Health check failed for 9094 order
4   MainThread : Calling ping (health check) service for 9092 order
5   MainThread : Order service leader found - 9092 order
6   MainThread : Notifying 9090 replica of order leader
7   MainThread : Notifying 9092 replica of order leader
8   MainThread : Notifying 9094 replica of order leader
9   MainThread : Cannot reach 9094 replica - nameserver down
10  MainThread : Starting threaded httpd server on 0.0.0.0:8000
11
```

Part 3: Fault Tolerance

Case 1: Order leader is replica 9094, but it has crashed. While processing the POST order trade request, the frontend recognizes this and chooses the next replica, which is 9092, as the leader. And then processes the request successfully.

```
umass-cs677-current > lab-3-asterix-and-double-trouble-sahil-n-ash > src > logs > ≡ frontend.log
26
27 Thread-1 (process_request_thread) : Received POST request on url /orders
28 Thread-1 (process_request_thread) : data - {'name': 'FishCo', 'quantity': 10, 'type': 'sell'} <class 'dict'>
29 Thread-1 (process_request_thread) : Calling trade service
30 Thread-1 (process_request_thread) : 9094 leader of backend order service down
31 Thread-1 (process_request_thread) : Trying request again...
32 Thread-1 (process_request_thread) : Calling ping (health check) service for 9094 order
33 Thread-1 (process_request_thread) : Health check failed for 9094 order
34 Thread-1 (process_request_thread) : Calling ping (health check) service for 9092 order
35 Thread-1 (process_request_thread) : Order service leader found - 9092 order
36 Thread-1 (process_request_thread) : Notifying 9090 replica of order leader
37 Thread-1 (process_request_thread) : Notifying 9092 replica of order leader
38 Thread-1 (process_request_thread) : Notifying 9094 replica of order leader
39 Thread-1 (process_request_thread) : Cannot reach 9094 replica - service down
40 Thread-1 (process_request_thread) : Calling trade service
41 Thread-2 (process_request_thread) : Received GET request on url /cache_invalid/FishCo
42 Thread-2 (process_request_thread) : Making Cache invalid
43 Thread-2 (process_request_thread) : Cache Invalidated in memory
44 127.0.0.1 - - [01/May/2023 17:39:28] "GET /cache_invalid/FishCo HTTP/1.1" 200 -
45 Thread-2 (process_request_thread) : Finished GET request
46
47 Thread-1 (process_request_thread) : Received trade response - 200 {'data': {'transaction_number': 28}}
48 127.0.0.1 - - [01/May/2023 17:39:28] "POST /orders HTTP/1.1" 200 -
49 Thread-1 (process_request_thread) : Finished POST request
50
```


Case 2: Order leader 9094 crashed and is now coming back online. Below image shows that while starting up it syncs with the leader and starts saving the order details from the leader

```
umass-cs677-current > lab-3-asterix-and-double-trouble-sahil-n-ash > src > logs > ≡ order_9094.log
1  MainThread : Existing Order Log File Found.
2  MainThread : TXN_NUM is initialized as 14
3  MainThread : Loading order log
4  MainThread : Order replica just started. Trying to sync
5  MainThread : Checking for leader at startup time
6  MainThread : Starting new HTTP connection (1): 0.0.0.0:8000
7  MainThread : http://0.0.0.0:8000 "GET /get_order_leader HTTP/1.1" 200 32
8  MainThread : Leader found at startup time, syncing... 9092 <class 'int'>
9  MainThread : Calling get_order_log_diff from 9092 order leader
10 MainThread : Sync done
11 MainThread : Updated TXN_NUM is 30
12 MainThread : Order service ready|
13 Pyro-Worker-4919538704 : Writing trade data from leader - {'transaction_number': 30, 'stock_name': 'MenhirCo', 'order_type': 'BUY'}
14 Pyro-Worker-4919538704 : No update required, since data is already captured
15 Pyro-Worker-4919539408 : Writing trade data from leader - {'transaction_number': 31, 'stock_name': 'MenhirCo', 'order_type': 'BUY'}
16 Pyro-Worker-4919539408 : Write lock - acquired
17 Pyro-Worker-4919539408 : Write lock - released
18 Pyro-Worker-4919539408 : TXN_NUM is - 31
19 Pyro-Worker-4854199376 : Writing trade data from leader - {'transaction_number': 32, 'stock_name': 'MenhirCo', 'order_type': 'BUY'}
20 Pyro-Worker-4854199376 : Write lock - acquired
21 Pyro-Worker-4854199376 : Write lock - released
22 Pyro-Worker-4854199376 : TXN_NUM is - 32
23 Pyro-Worker-4394009424 : Writing trade data from leader - {'transaction_number': 33, 'stock_name': 'BoarCo', 'order_type': 'BUY'}
24 Pyro-Worker-4394009424 : Write lock - acquired
25 Pyro-Worker-4394009424 : Write lock - released
26 Pyro-Worker-4394009424 : TXN_NUM is - 33
27 Pyro-Worker-4919538704 : Writing trade data from leader - {'transaction_number': 34, 'stock_name': 'MenhirCo', 'order_type': 'BUY'}
28 Pyro-Worker-4919538704 : Write lock - acquired
29 Pyro-Worker-4919538704 : Write lock - released
30 Pyro-Worker-4919538704 : TXN_NUM is - 34
31 Pyro-Worker-4919539408 : Writing trade data from leader - {'transaction_number': 35, 'stock_name': 'MenhirCo', 'order_type': 'BUY'}
32 Pyro-Worker-4919539408 : Write lock - acquired
```