

Lab 3

Tanay P Joshi
Megha Singh

Scenario 1: Leader doesn't fail

Client:

```
tanayjoshi@Tanays-MacBook-Air client % python3 client.py & python client.py
[1] 43791
zsh: command not found: python
tanayjoshi@Tanays-MacBook-Air client % GET => 200 {'name': 'fox', 'qty': 100, 'cost': 9.99}
GET => 200 {'name': 'fox', 'qty': 100}
GET => 200 {'name': 'lego', 'qty': 100, 'cost': 29.99}
GET => 200 {'name': 'lego', 'qty': 100}
GET => 200 {'name': 'whale', 'qty': 100, 'cost': 19.99}
POST => 200 {"data": {"order_number": 0}}
Order Number: 0
GET => 200 {'name': 'bicycle', 'qty': 100, 'cost': 99.99}
GET => 200 {'name': 'bicycle', 'qty': 100}
POST => 200 {"data": {"order_number": 1}}
Order Number: 1
GET => 200 {'name': 'bicycle', 'qty': 95, 'cost': 99.99}
POST => 200 {"data": {"order_number": 2}}
Order Number: 2
GET => 200 {'name': 'marbles', 'qty': 100, 'cost': 4.99}
```

Frontend:

Here, we also see how the cache is used in real-time. When a request for bicycle comes in, it goes into a cache miss the value is not previously seen by the service. It processes the order and adds bicycle to the final cache status. But thereafter, since there is a buy request on bicycle, which reduces its count, it also invalidates bicycle from the cache as the quantity has now changed.

Therefore, it deletes bicycle from the cache. It should also be noted that because it is an LRU cache, the most-recent item, ie, bicycle was added to the top of the cache. (Here, the rightmost side of the cache is the top of the cache, and as we go left, we go into the least recently used items)

```
Cache Miss OrderedDict([('fox', 100), ('lego', 100)])
RESPONSE <Response [200]>
New Cache Populated OrderedDict([('fox', 100), ('lego', 100), ('bicycle', 95)])
127.0.0.1 - - [05/May/2024 22:18:46] "GET /products/bicycle HTTP/1.1" 200 -

Final cache status OrderedDict([('fox', 100), ('lego', 100), ('bicycle', 95)])
Order Data {'name': 'bicycle', 'qty': 4}

Invalidating bicycle from cache...

Cache after deletion OrderedDict([('fox', 100), ('lego', 100)])

127.0.0.1 - - [05/May/2024 22:18:46] "DELETE /delete/bicycle HTTP/1.1" 200 -
Response Data {'status': 200, 'data': {'data': {'order_number': 2}}}}
127.0.0.1 - - [05/May/2024 22:18:46] "POST /orders HTTP/1.1" 200 -
```

Scenario 2 Leader election (on frontend):

```
tanayjoshi@Tanays-MacBook-Air frontend % python3 frontend.py

Starting election

Checking heartbeat for port: 8012

response from check heartbeatstatus for port 8012
Heartbeat received from LEADER node. 8012

Priority and order port in conduct election 3 8012
Notifying other replicas about new leader
Response from notify followers: <Response [200]>
Response from notify followers: <Response [200]>
Response from notify followers: <Response [200]>
LEADER_ORDER_PORT 8012
Checking leader heartbeat... 8012
```

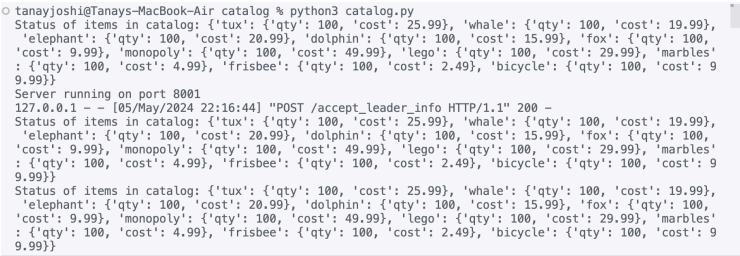
Order:

```
tanayjoshi@Tanays-MacBook-Air src % ./start_orders.sh
tanayjoshi@Tanays-MacBook-Air src % Order Service running on port 8012
Order Service running on port 8010
127.0.0.1 - - [05/May/2024 22:16:44] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:16:44] "POST /accept_leader_info HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:16:44] "POST /accept_leader_info HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:16:44] "POST /accept_leader_info HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:16:44] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:16:47] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:16:50] "GET /check_heartbeat HTTP/1.1" 200 -
tanayjoshi@Tanays-MacBook-Air src % 127.0.0.1 - - [05/May/2024 22:16:53] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:16:56] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:16:59] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:17:02] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:17:05] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 - - [05/May/2024 22:17:08] "GET /check_heartbeat HTTP/1.1" 200 -
```

Logs replicated across order nodes:



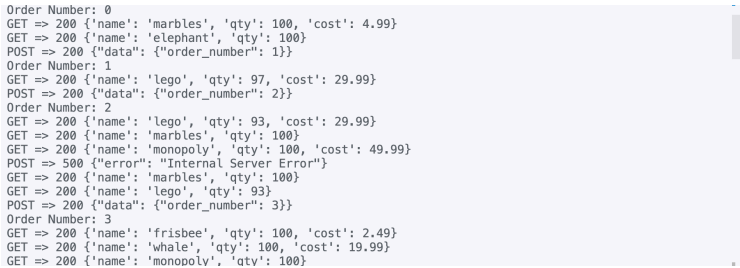
Catalog:



Scenario 3: Leader fails

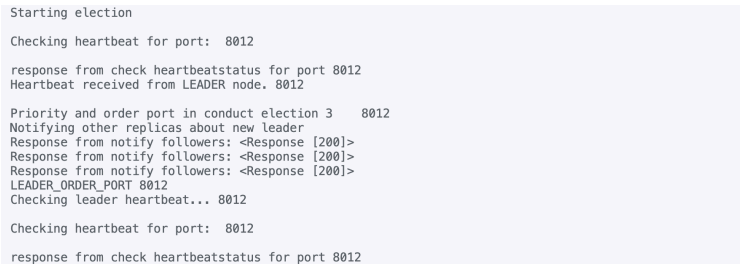
Client:

Internal Server Error indicates leader crashed.



Frontend:

Initially, all order replicas are up. Order process (port 8012) is the leader



Leader is killed. Frontend restarts the election.



Order process (port 8011) is the **new** leader and continues serving requests.

```
Checking heartbeat for port: 8011

response from check heartbeatstatus for port 8011
Heartbeat received from LEADER node. 8011

Priority and order port in conduct election 2 8011
Notifying other replicas about new leader
Response from notify followers: <Response [200]>
Response from notify followers: <Response [200]>
Port is not available 8012
elected order port from except part of periodic leader check 8011
New leader elected on port 8011

Serving marbles from cache... OrderedDict([('lego', 93), ('marbles', 100), ('monopoly', 100)])
127.0.0.1 -- [05/May/2024 22:41:20] "GET /products/marbles HTTP/1.1" 200 -

Final cache status OrderedDict([('lego', 93), ('monopoly', 100), ('marbles', 100)])
Order Data {'name': 'marbles', 'qty': 4}
```

Order:

Initially all order nodes are up:

```
tanayjoshi@Tanays-MacBook-Air src % Order Service running on port 8011
Order Service running on port 8010
Order Service running on port 8012
tanayjoshi@Tanays-MacBook-Air src % 127.0.0.1 -- [05/May/2024 22:40:47] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:40:47] "POST /accept_leader_info HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:40:47] "POST /accept_leader_info HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:40:47] "POST /accept_leader_info HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:40:47] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:40:50] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:40:53] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:40:56] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:40:59] "GET /check_heartbeat HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:41:02] "GET /check_heartbeat HTTP/1.1" 200 -
```

Logs are transferred once the dead node is up again. We killed the order process (port 8012) and brought it up after a few seconds.

```
tanayjoshi@Tanays-MacBook-Air src % ./kill_leader.sh
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
Python 45913 tanayjoshi 6u IPv4 0x5dc3f3817bb99879 0t0 TCP *:8012 (LISTEN)
Killing process with PID 45913...
tanayjoshi@Tanays-MacBook-Air src % ./start_leader.sh
tanayjoshi@Tanays-MacBook-Air src % Fetched missed orders for port : 8012
Order Service running on port 8012
127.0.0.1 -- [05/May/2024 22:41:32] "POST /replicate_order HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:41:32] "POST /replicate_order HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:41:35] "POST /replicate_order HTTP/1.1" 200 -
127.0.0.1 -- [05/May/2024 22:41:35] "POST /replicate_order HTTP/1.1" 200 -
```

The order_data_3.csv contains the missing logs:

order_data_1.csv	order_data_2.csv	order_data_3.csv
1 number: 0, name: lego, qty: 3	1 number: 0, name: lego, qty: 3	1 number: 0, name: lego, qty: 3
2 number: 1, name: elephant, qty: 2	2 number: 1, name: elephant, qty: 2	2 number: 1, name: elephant, qty: 2
3 number: 2, name: lego, qty: 4	3 number: 2, name: lego, qty: 4	3 number: 2, name: lego, qty: 4
4 number: 3, name: marbles, qty: 4	4 number: 3, name: marbles, qty: 4	4 number: 3, name: marbles, qty: 4
5 number: 4, name: frisbee, qty: 2	5 number: 4, name: frisbee, qty: 2	5 number: 4, name: frisbee, qty: 2
6 number: 5, name: fox, qty: 1	6 number: 5, name: fox, qty: 1	6 number: 5, name: fox, qty: 1
7 number: 6, name: marbles, qty: 1	7 number: 6, name: marbles, qty: 1	7 number: 6, name: marbles, qty: 1
8 number: 7, name: dolphin, qty: 2	8 number: 7, name: dolphin, qty: 2	8 number: 7, name: dolphin, qty: 2
9 number: 8, name: marbles, qty: 5	9 number: 8, name: marbles, qty: 5	9 number: 8, name: marbles, qty: 5

Catalog:

Catalog keeps running displaying item status

```
tanayjoshi@Tanays-MacBook-Air catalog % python3 catalog.py
Status of items in catalog: {'tux': {'qty': 100, 'cost': 25.99}, 'whale': {'qty': 100, 'cost': 19.99}, 'elephant': {'qty': 100, 'cost': 20.99}, 'dolphin': {'qty': 100, 'cost': 15.99}, 'fox': {'qty': 100, 'cost': 9.99}, 'monopoly': {'qty': 100, 'cost': 49.99}, 'lego': {'qty': 100, 'cost': 29.99}, 'marbles': {'qty': 100, 'cost': 4.99}, 'frisbee': {'qty': 100, 'cost': 2.49}, 'bicycle': {'qty': 100, 'cost': 9.99}}
Server running on port 8001
127.0.0.1 -- [05/May/2024 22:40:47] "POST /accept_leader_info HTTP/1.1" 200 -
Status of items in catalog: {'tux': {'qty': 100, 'cost': 25.99}, 'whale': {'qty': 100, 'cost': 19.99}, 'elephant': {'qty': 100, 'cost': 20.99}, 'dolphin': {'qty': 100, 'cost': 15.99}, 'fox': {'qty': 100, 'cost': 9.99}, 'monopoly': {'qty': 100, 'cost': 49.99}, 'lego': {'qty': 100, 'cost': 29.99}, 'marbles': {'qty': 100, 'cost': 4.99}, 'frisbee': {'qty': 100, 'cost': 2.49}, 'bicycle': {'qty': 100, 'cost': 9.99}}
Status of items in catalog: {'tux': {'qty': 100, 'cost': 25.99}, 'whale': {'qty': 100, 'cost': 19.99}, 'elephant': {'qty': 100, 'cost': 20.99}, 'dolphin': {'qty': 100, 'cost': 15.99}, 'fox': {'qty': 100, 'cost': 9.99}, 'monopoly': {'qty': 100, 'cost': 49.99}, 'lego': {'qty': 100, 'cost': 29.99}, 'marbles': {'qty': 100, 'cost': 4.99}, 'frisbee': {'qty': 100, 'cost': 2.49}, 'bicycle': {'qty': 100, 'cost': 9.99}}
127.0.0.1 -- [05/May/2024 22:41:08] "GET /products/elephant HTTP/1.1" 200 -
```