Evaluation doc (For Parts 4 of Grading Rubric)

1. How did you deploy your application on AWS?

1. Generate a pair of public keys with "ssh-keygen".
2. Copy "id_rsa.pub" content to GitHub SSH keys setting (https://github.com/settings/keys)
3. git pull lab3 source code.
4. sudo apt install docker-compose
5. sudo docker-compose up

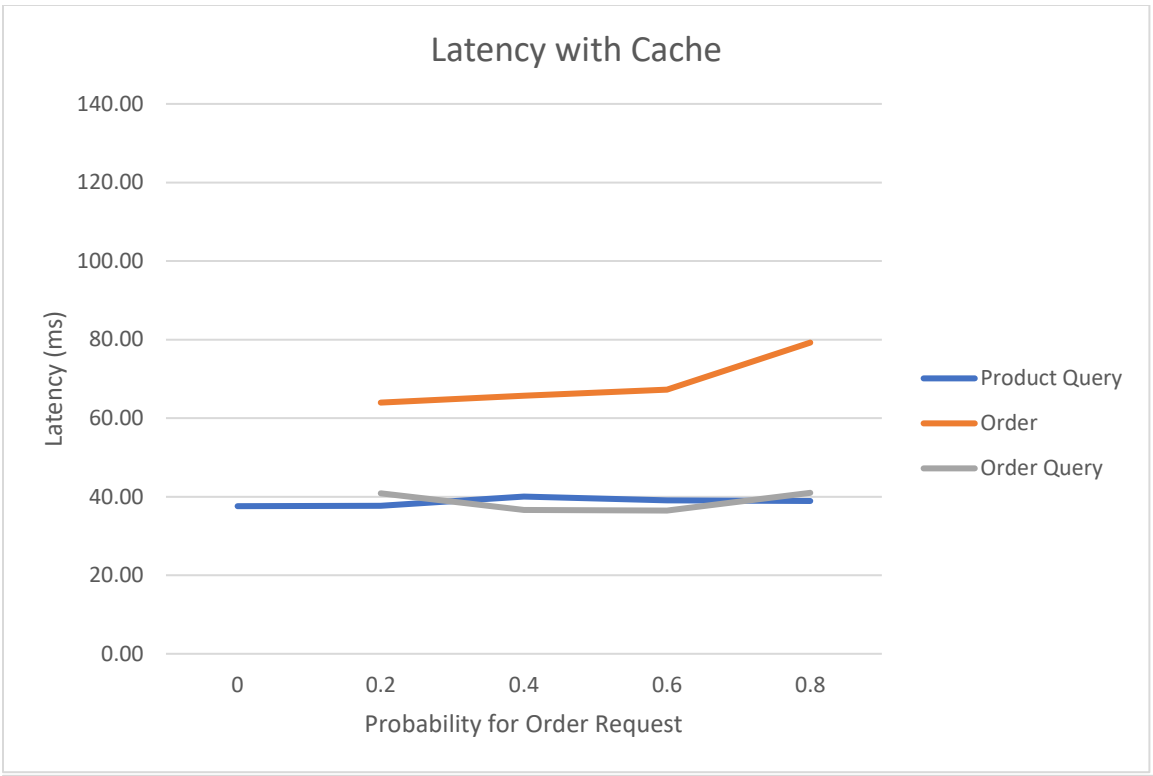
2. Measurement results and plots

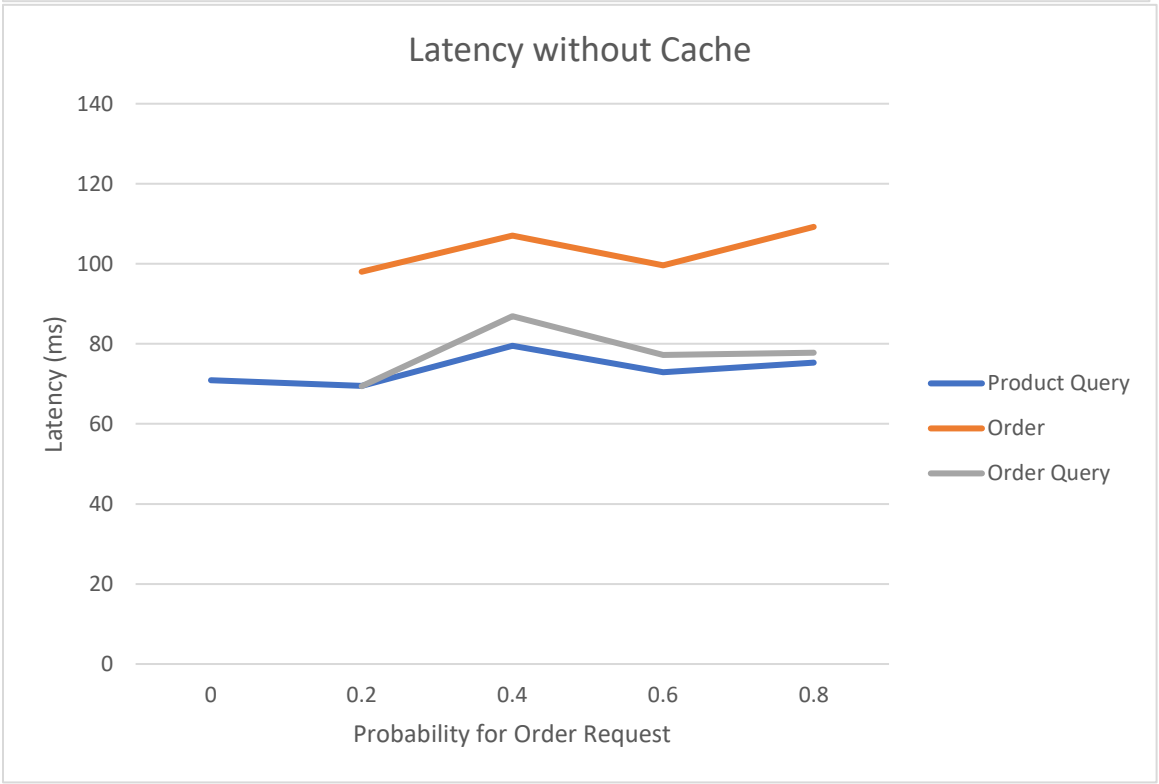
(How to run test script? Check out README at ./docs)

- Measurement results:
  Note: We tested by starting 5 clients concurrently.

Probability for Order Request	Product Query Latency With Cache (ms)	Product Query Latency Without Cache (ms)
0	37.58	70.92
0.2	37.68	69.5
0.4	40.03	79.52
0.6	39.06	72.92
0.8	38.90	75.3
Probability for Order Request	Product Order Latency With Cache (ms)	Product Order Latency Without Cache (ms)
0		
0.2	63.95	98.02
0.4	65.68	107.02
0.6	67.22	99.62
0.8	79.23	109.23
Probability for Order Request	Order Query Latency With Cache (ms)	Order Query Latency Without Cache (ms)
0		
0.2	40.82	69.42
0.4	36.59	86.91
0.6	36.46	77.2
0.8	40.98	77.81

- Plots:


Latency with Cache
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
Latency (ms)
0
0.2
0.4
0.6
0.8
Probability for Order Request
Product Query
Order
Order Query


Latency without Cache
140
120
100
80
60
40
20
0
Latency (ms)
0
0.2
0.4
0.6
0.8
Probability for Order Request
Product Query
Order
Order Query

- c. **Observations**:

From Request Latency with cache and Request Latency without cache plots, we can observe that the latency with cache is lower than latency without cache, it makes sense, that for cache hits, it will return from cache instead of requesting the real service, which causes latency to be lower.

Without cache, it must retrieve information by calling real service, which causes latency to be higher.

3. Analysis of the results and answers to the questions in part 3

- Can the clients notice the failures (either during order requests or the final order checking phase) or are they transparent to the clients?

Answer: The client cannot notice the failure of order service. They are transparent to the clients. That is to say, when any replica crashes (including the leader), toy purchase requests and order query requests can still be handled and return the correct result.

- Do all the order service replicas end up with the same database file?

Answer: Yes, all the order service replicas end up with the same database file.