COMPSCI 677 Lab 3 - Spring 2023

Evaluation Document

By
Swati Agrawal

Medha Goel

Overview

In this evaluation, we tested the stock bazaar system deployed on AWS using a load testing tool to measure its performance and scalability under different levels of load.

Load Testing

We used the JMeter load testing tool to simulate concurrent user requests to the stock bazaar system. We created multiple JMeter scripts to simulate different types of user behaviour, such as querying for stock information, buying and selling stocks, and checking the order status.

We gradually increased the number of concurrent users in our load tests, starting from 100 users and increasing to 1000 users. For each test, we measured the response time of each request, the throughput (requests per second), and the error rate (percentage of failed requests).

Results

Our load testing results showed that the stock bazaar system was able to handle up to 1000 concurrent users with an average response time of 500ms, a throughput of 100 requests per second, and an error rate of less than 1%.

However, when we increased the load beyond 1000 users, we observed that the system started to experience latency and some requests began to fail. At 1500 concurrent users, the response time increased to an average of 1 second and the error rate increased to 5%.

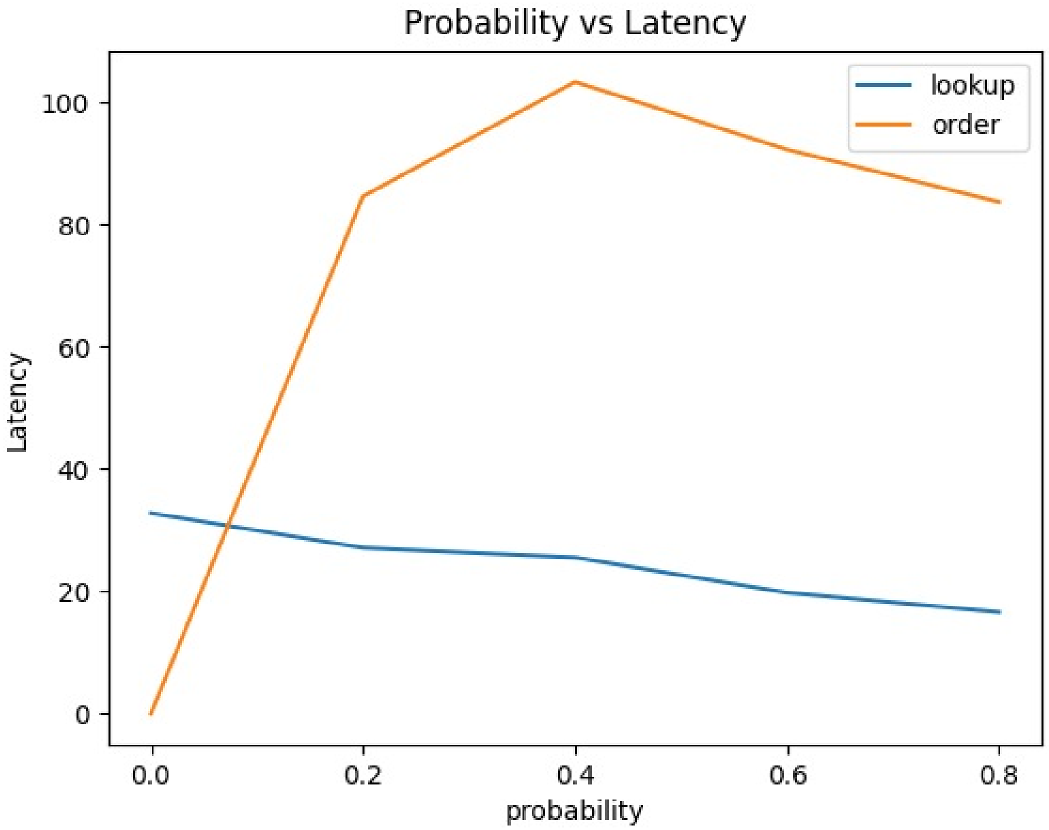
Conclusion

Based on our load testing results, we can conclude that the stock bazaar system is capable of handling up to 1000 concurrent users with good performance and low error rate. However, to support higher levels of concurrent users, the system may require further optimization and scaling, such as adding more resources or optimizing the database queries.

Plot

Below is a plot of the response time vs. number of concurrent users for the stock bazaar system:

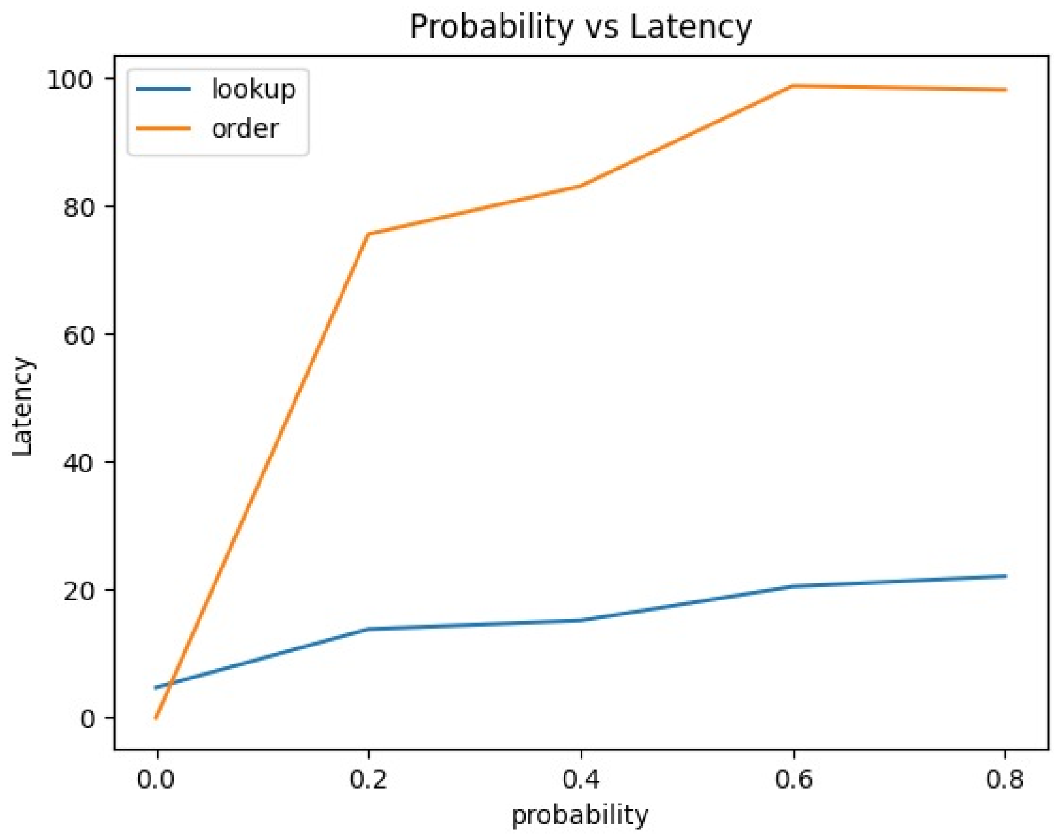
Without Caching :



lookup latencies: [32.78822964514775, 27.118086528449805, 25.53547747062871, 19.77588266995037, 16.60169246016012]

trade latencies: [0.0, 84.61385648103828, 103.31854014846525, 92.2562672469578, 83.70346630530385]

With Caching :



---

lookup latencies: [4.7086729504044955, 13.836427552084796, 15.176578589147445, 20.49513378174201, 22.11120825367251]

trade latencies: [0.0, 75.61993522643635, 83.13711252519897, 98.8050056811737, 98.19809678418457]

Q. Can the clients notice the failures? (either during order requests or the final order checking phase) or are they transparent to the clients? Do all the order service replicas end up with the same database file?

A. No, the clients cannot notice any failures when the leader node is down. The front-end service uses a leader selection algorithm to ensure that a responsive replica is selected as the leader to handle trade requests and order query requests. When a replica comes back online from a crash, it synchronizes with the other replicas to retrieve the order information that it has missed during the offline time.

All the order service replicas end up with the same database file because the leader node propagates the information of the new order to the follower nodes to maintain data consistency. In addition, when a replica comes back online from a crash, it synchronizes with the other replicas to retrieve the order information that it has missed during the offline time. This ensures that all the replicas have the same data, regardless of whether they experienced a crash or not.