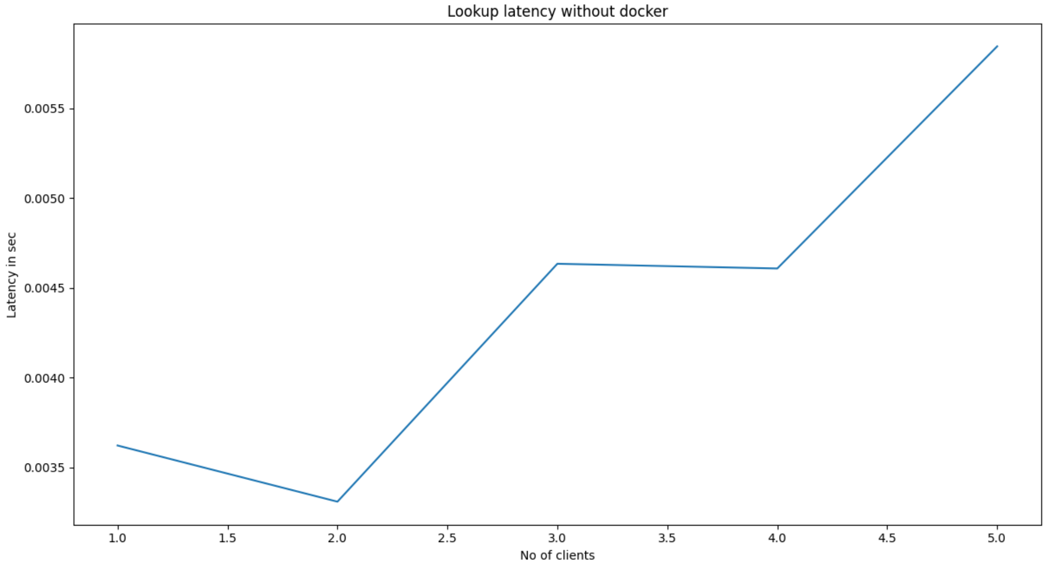
Evaluation Doc

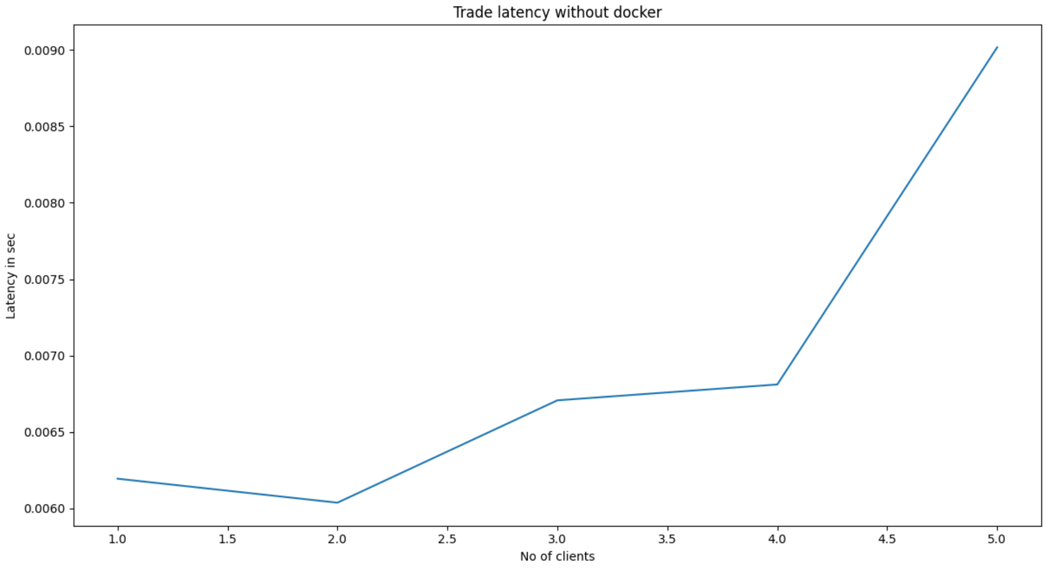
Steps we followed to load test our system:

Inside the client, we executed the program with a for loop for 100 iterations and initialized our probability with p=1. In every iteration, we are calling lookup, and based on the probability trade request is also executed (almost every time) we calculated the latency for each request and took the average latency for 100 iterations. We followed this process by varying the number of clients from 1 to 5. We achieved this using a bash script to run the clients concurrently. And we plotted the following plots.

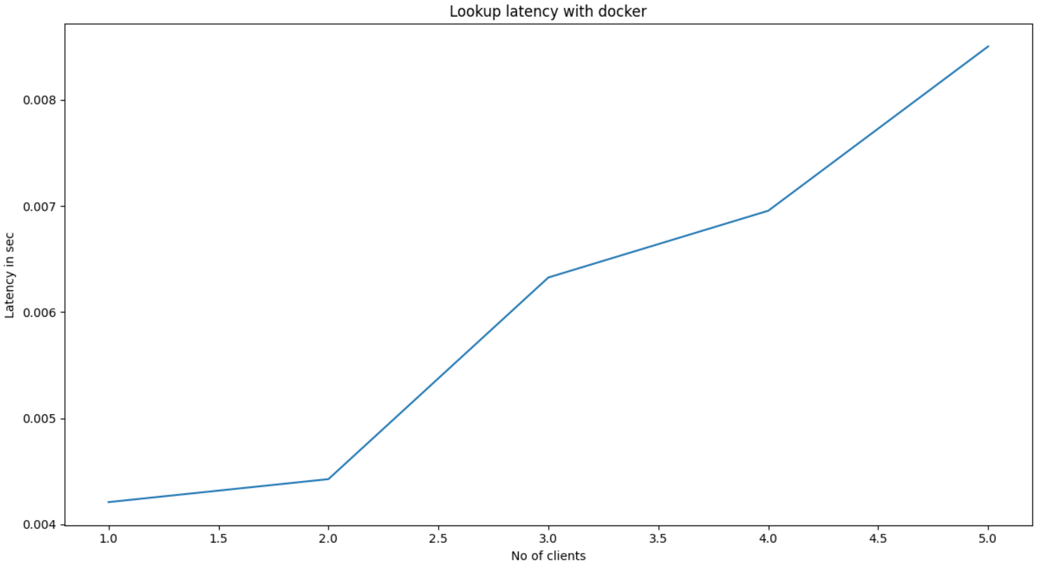
Evaluating Lookup Performance when deployed normally:



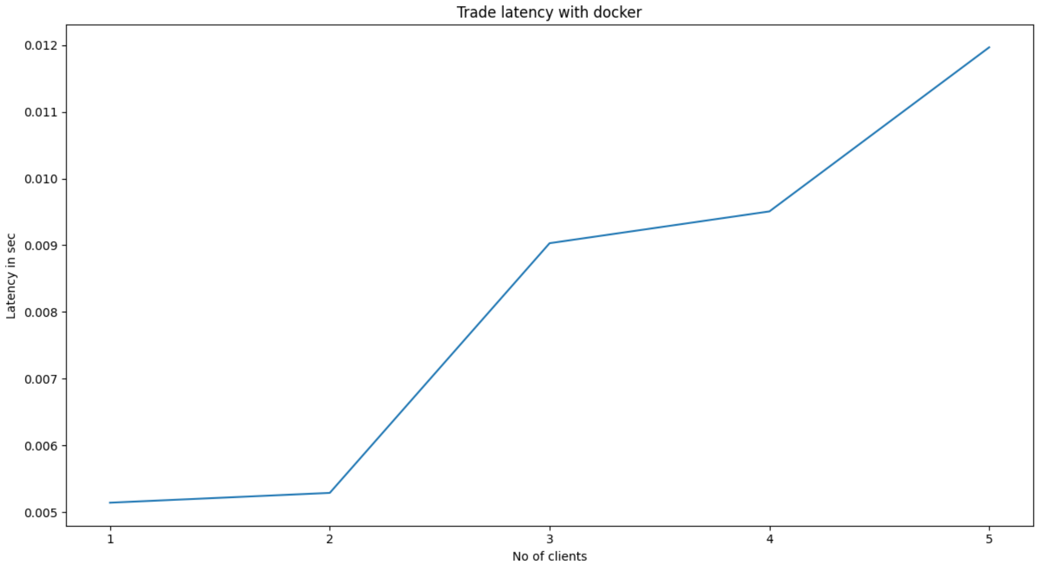
Evaluating Trade performance when deployed normally:



Evaluating Lookup performance when deployed in docker:



Evaluating Trade performance when deployed in docker:



Questions:

1. Does the latency of the application change with and without Docker containers? Did virtualization add any overheads?

   If we observe the above plots by running the application normally and within the dockers, the latency of the application is changed. Both the latencies of Lookup and Trade requests are lower when the application is deployed normally than when the application is deployed within the docker. Yes, in our case the virtualization added additional overheads.

2. How does the latency of the lookup requests compare to trade? Since trade requests involve all these microservices, while lookup requests only involve two microservices, does it impact the observed latency?

   Yes, the Trade requests have more latency than the Lookup requests in both cases, when the application is deployed normally and within the docker. The reason for this is that the Lookup requests involve only communication between 2 microservices and it is just read operations. But, in Trade requests, the communication is happening between 3 microservices and trade requests also involve write operations which are slower operations when compared to read operations.

3. How does the latency change as the number of clients change? Does it change for different types of requests?

Since we used 2 workers in the threadpool in the backend (catalog and order) microservices, the latency is usually similar when no of clients is 1 or 2, 3 or 4, as the requests are distributed equally among the 2 threads and the latency is increasing as the number of clients increases. This trend is observed in every request for both Lookup and Trade in both methods of application deployment i.e. normally and within the dockers.