Evaluation Document

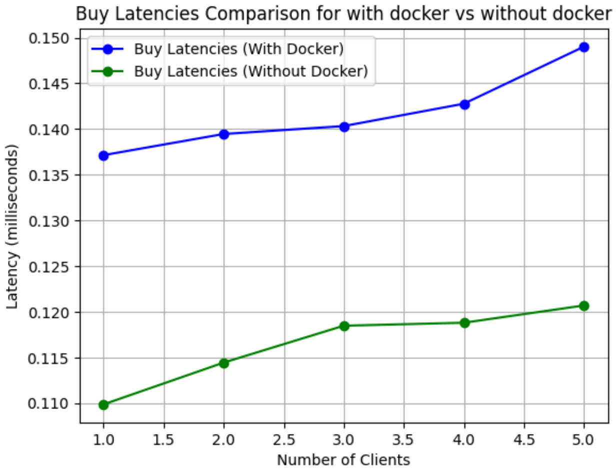
Does the latency of the application change with and without Docker containers? Did virtualization add any overheads?

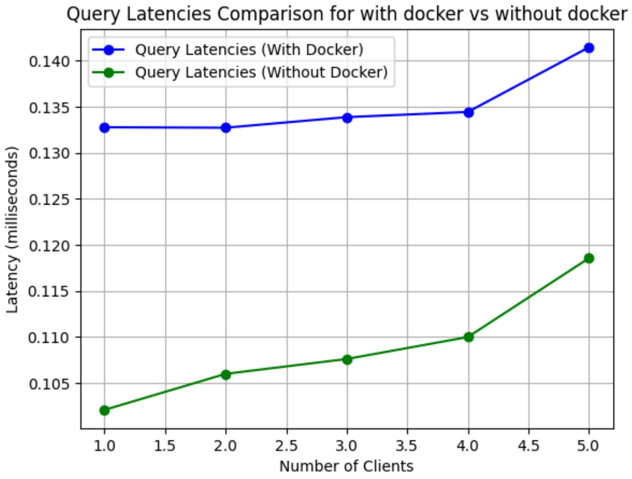
Number of Clients	Average Buy latency(with Docker)	Average Buy Latency (without Docker)
1	0.1371427023	0.1098585988
2	0.1394561463	0.1144334255
3	0.14032426	0.1184692183
4	0.1427720517	0.1187968088
5	0.1490238152	0.1206864783

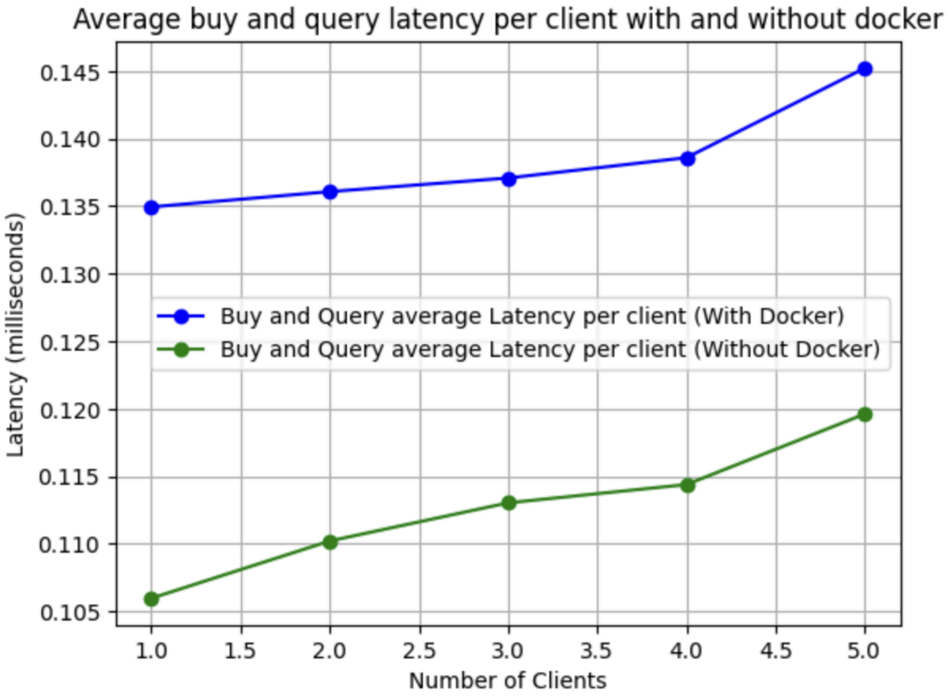
Buy Latencies Comparison for with docker vs without docker

Number of Clients	Average Query latency(with Docker)	Average Query Latency (without Docker)
1	0.1327486503	0.1020937154
2	0.1326997768	0.1060006298
3	0.1338474696	0.1076135267
4	0.1344088858	0.1099931577
5	0.1414234068	0.1185496795

Query Latencies Comparison for with docker vs without docker



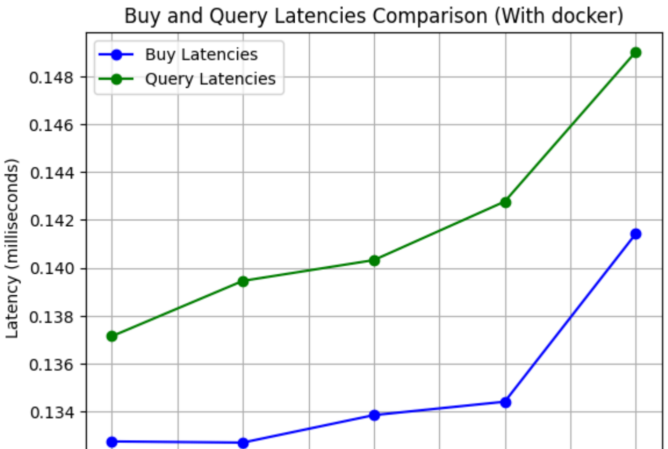


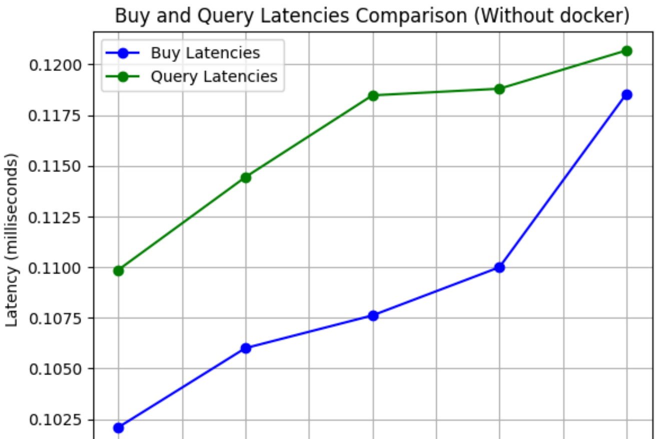


As shown in the above graph, the latency increases in case of docker containerization due to virtualization overhead against no containerization. The conditions under which the above load testing were: all three microservices on the remote Edlab machines and the client a remote PC. p was kept at 0.5 in both the cases and stocks were reset to avoid insufficient quantity error in catalog microservice. The average latency for query endpoint and buy endpoint with no containerization comes out to be **0.10885014183123906** and **0.11644890593032502** respectively. On the other hand, the average latency for query endpoint and buy endpoint with containerization comes out to be **0.13502563788096109** and **0.14174379509186819** respectively.

The average latency for both endpoint with docker and without docker are also very different. With docker the average latency for both endpoint is higher than without docker.

How does the latency of the query compare to buy? Since buy requests involve all three microservices, while query requests only involve two microservices, does it impact the observed laten





As shown in the above graph, the latency increases in case of Buy endpoint due to extra inter-process communication overhead against Query endpoint as we call one extra service as compared to buy endpoint. The conditions under which the above load testing were: all three microservices on the remote Edlab machines and the client a remote PC. p was kept at 0.5 in both the cases and stocks were reset to avoid insufficient quantity error in catalog microservice. The average latency for query endpoints for containerization and without containerization comes out to be **0.10885014183123906** and **0.13502563788096109** respectively. On the other hand, the average latency for buy endpoints without containerization and with containerization comes out to be **0.11644890593032502** and **0.14174379509186819** respectively.

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

Our experiments show that latency increases with an increase in the number of clients as shown in the graphs above. The trend is not very stark in case of containerization with docker, due to virtualization overhead which is more prominent when compared to latency of microservice processing. In an ideal scenario, as the number of clients increases, the latency should also increase due to overload in the servers which causes a resource bottleneck. This matches with the performance evaluation we have done as shown in the above graphs.