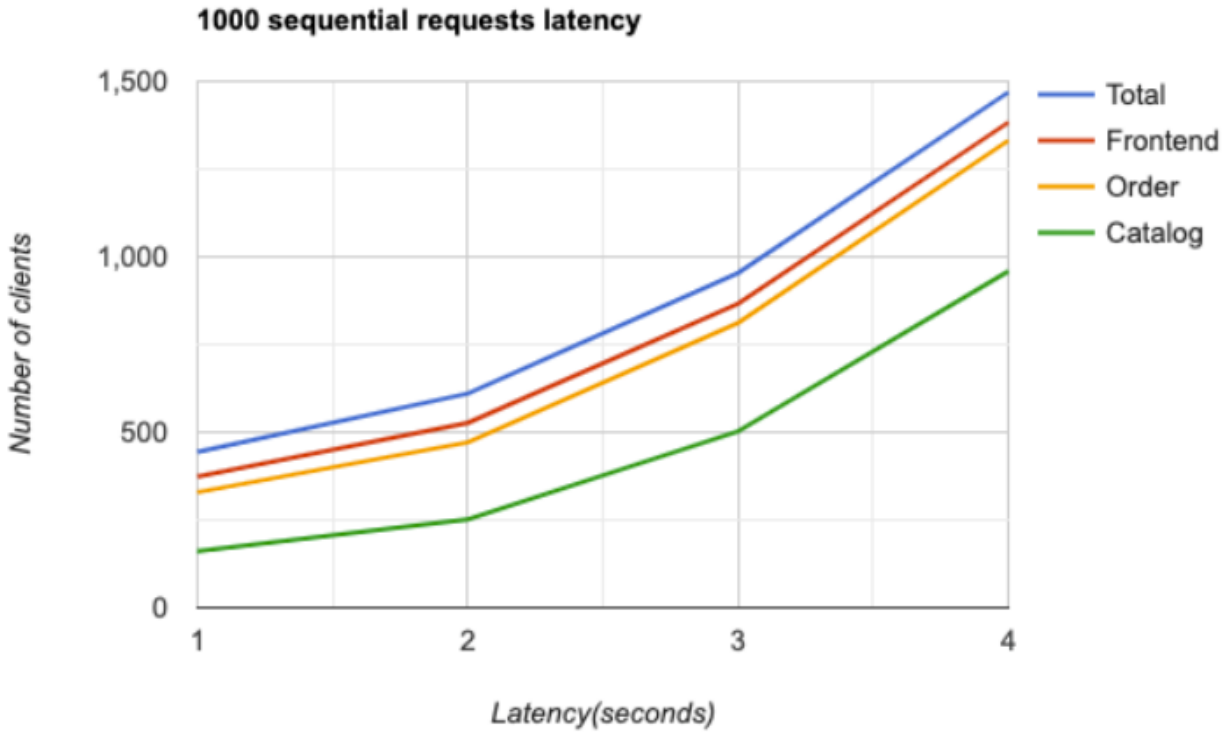
EVALUATION: AVERAGE RESPONSE TIME MEASUREMENTS

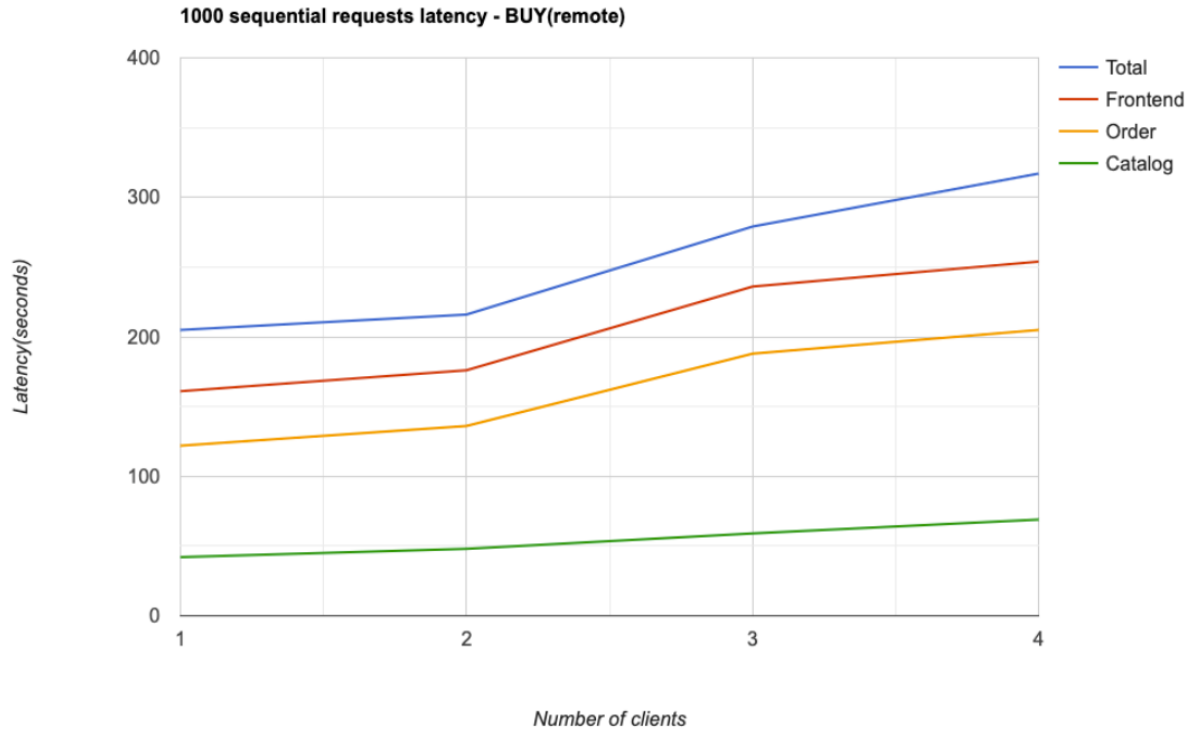
We measured average response time per client search request by measuring the end-to-end response time seen by a client for 1000 sequential requests. We also measured response times when multiple clients are concurrently making requests to the system. We tested this for varying number of clients i.e. N=1,2,3,4 and compared the performances of deployment on local and on AWS. We have created graphs and also furnished Tables for reference below. In the end, we make certain observations from our end.

Graph for buy operation on Local:



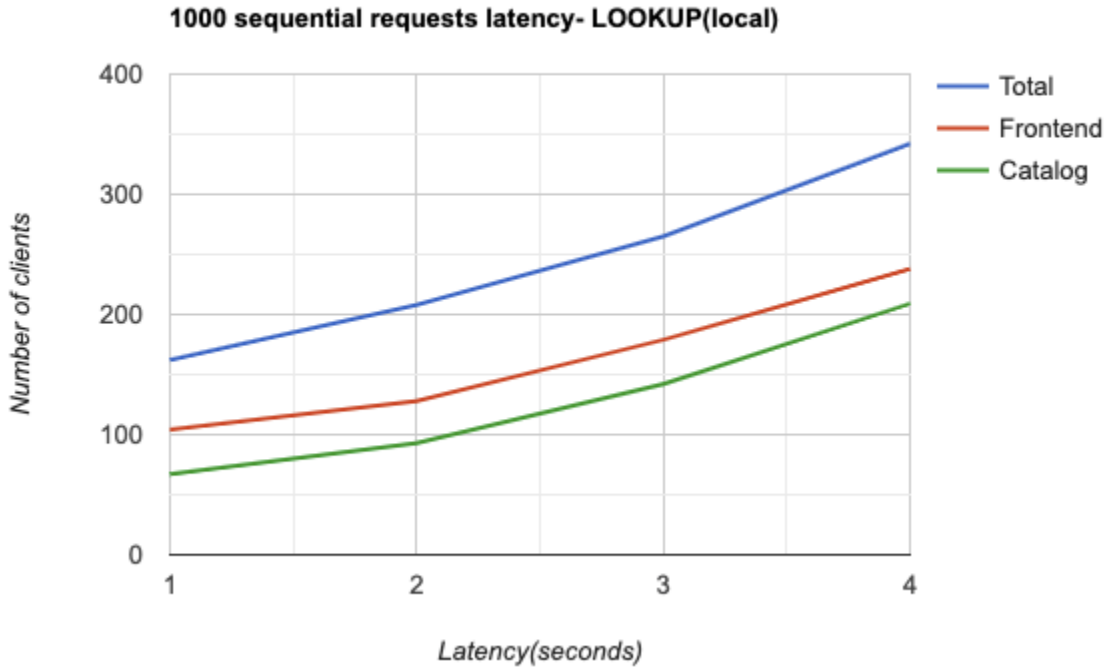
No. of clients	Total time	Frontend Time	Order Time	Catalog Time
1	444	374	329	161
2	610	527	471	251
3	954	866	811	502
4	1469	1383	1331	959

Graph for buy operation on remote (AWS):



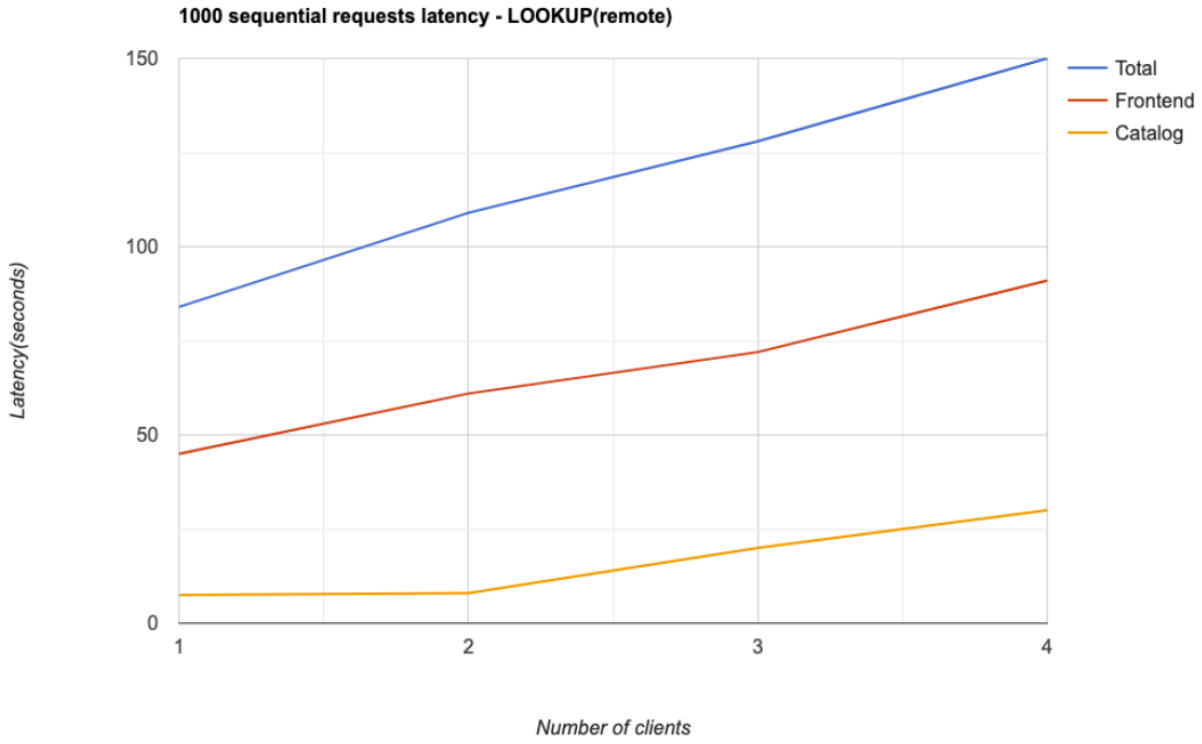
No. of clients	Total time	Frontend Time	Order Time	Catalog Time
1	205	161	122	42
2	216	176	136	48
3	279	236	188	59
4	317	254	205	69

Graph for Lookup operation on Local:



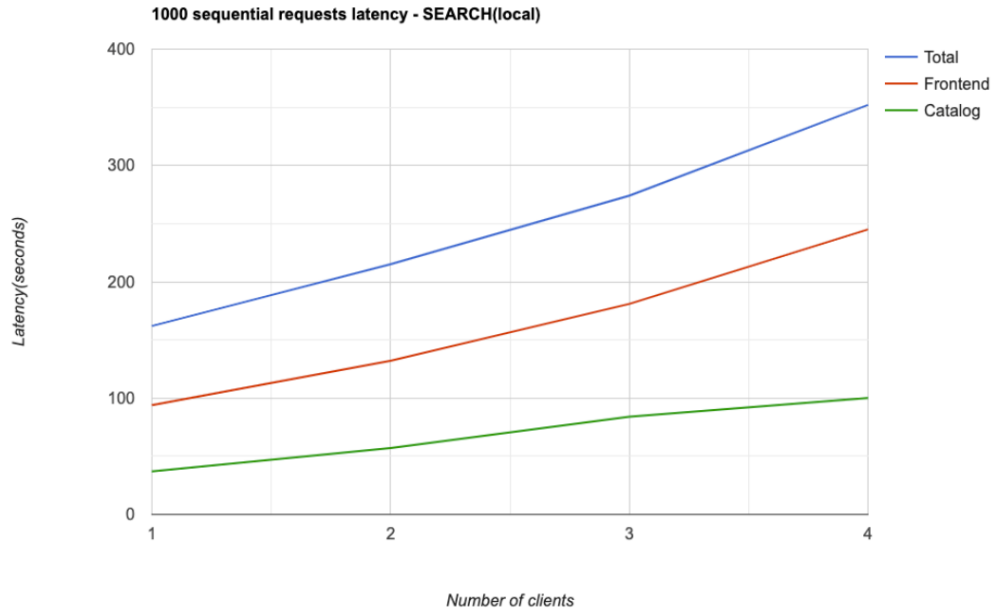
No. of clients	Total time	Frontend Time	Catalog Time
1	162	104	67
2	208	128	93
3	265	179	142
4	342	238	209

Graph for Lookup operation on Remote(AWS):



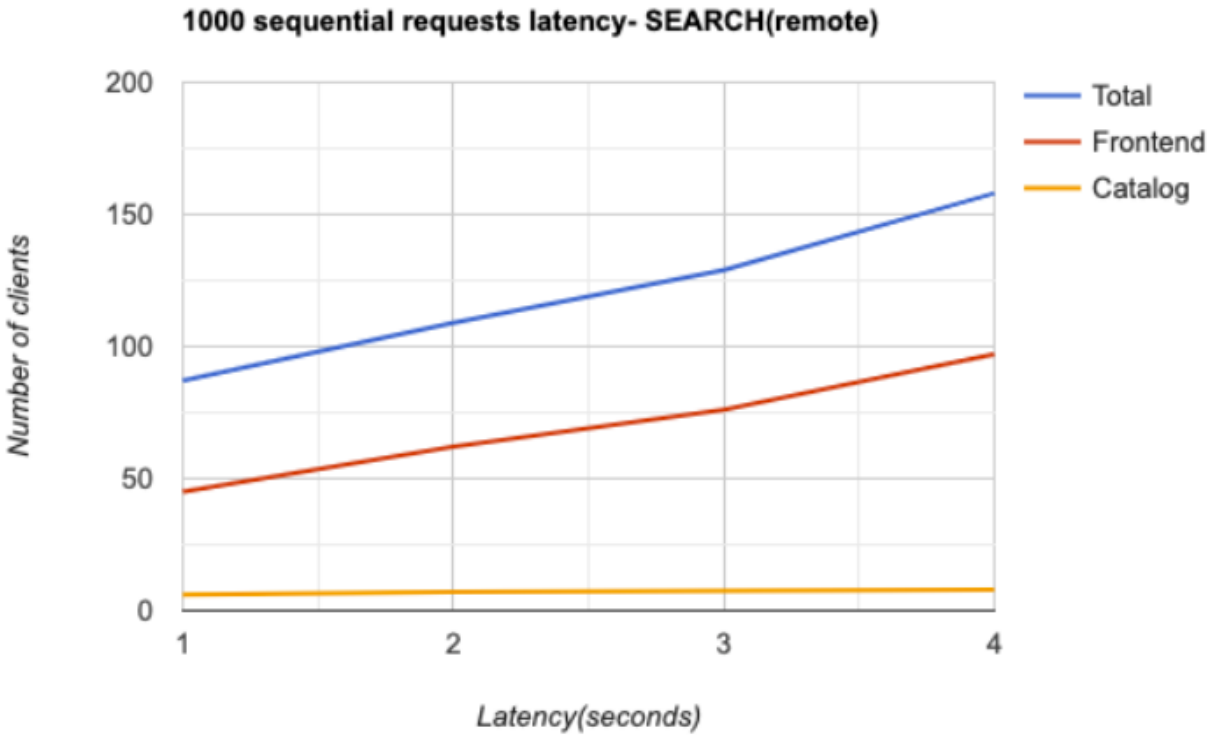
No. of clients	Total time	Frontend Time	Catalog Time
1	84	45	7.5
2	109	61	8
3	128	72	20
4	150	91	30

Graph for Search operation on Local:



No. of clients	Total time	Frontend Time	Catalog Time
1	162	94	37
2	215	132	57
3	274	181	84
4	352	245	100

Graph for Search Operation on Remote (AWS):



No. of clients	Total time	Frontend Time	Catalog Time
1	87	45	6
2	109	62	7
3	129	76	7.5
4	158	97	8

Inferences:

1) We observed that in general the “Buy” operation takes more time compared to search and lookup and this could potentially be attributed to two further reasons:
    a) There are three tiers in buy operation and because of the presence of an additional tier, it takes more time
    b) In buy operation, we are doing writes on the database which takes a lock on the database
2) Performance on remote deployment i.e. AWS is in general better as locally we were getting bottlenecked on network IO and in case of AWS load is getting distributed properly.