Distributed Operating System

Part 2

5/17/2022

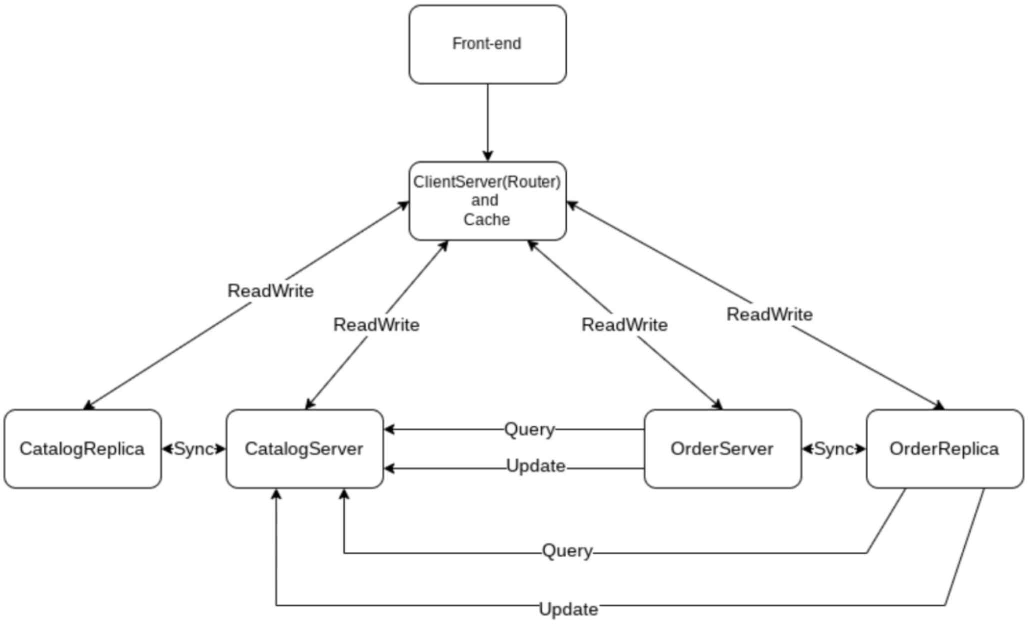
Ameed Omar -11743942

Supervisor: Dr. Samer Arandi

System description:

In this part of the project we add extra features to the system which is the cache server which require extra implementation in the back and the front end , For the back-end part we've asked to make 2 more replicas for our servers(the CatalogAPI server and the OredreAPI server) , so when we say replica first thing we should implement is a system of consistency to keep the data up-to-date between the main server and the replica . For the front-end part, we need to implement an algorithm to achieve load balancing between the main server and the replica and to spread the load on them.

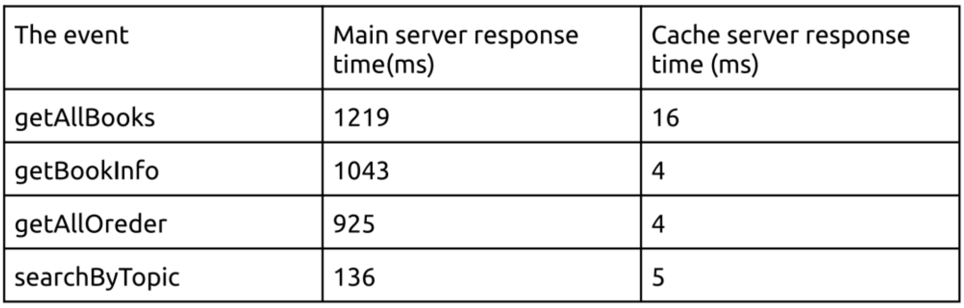
This picture describe the system:



When the client asks for data, he asks the cache server firstly if he finds the data, everything is ok. If not, the cache server will send a request for the server who has the data (catalog or the order server or its replicas) and then store it in the cache memory and give it to the client.

In case if the client wants to add a new book or make an update request for the book cost or stock the client will send the request to the client server and it will forward it to the required server and the server who will get that request will also send it data to the other server to keep the data consistency.

The difference between response time between main server and the cache server:



The event	Main server response time(ms)	Cache server response time (ms)
getAllBooks	1219	16
getBookInfo	1043	4
getAllOreder	925	4
searchByTopic	136	5

We can notice the big difference in response time between the main server and the cache server and the difference is because the cache server stores the data in the memory as a map, however the main server needs to bring the data from the database, which will take a lot of time.

How to run this project:

To run the frontend (clients) you need run this command in the UI folder:

```
flutter run
```

In addition, to run the back end (servers) you need to run the docker compose which is in the DOS_BazarProject folder and you need to run this command:

```
docker-compose up - -build
```