Design UBER/OLA App which is a ride sharing app

Solution :- ① Find out the Functional Requirements

(a) User :-
(i) Book a cab (ii) ETA of the cab arrival
(iii) Cancel the cab (iv) Details of the ongoing trip

(b) Driver :-
(i) Notification for the new booking
(ii) Accept / reject a new Booking
(iii) Update any location very frequently
(iv) Details of the ongoing trip.

② Find out the Non-Functional Requirements

(i) It must be real-time, no stale data
(ii) It should be scalable, in peak hours like evening hour, noon hours your app should work smoothly.
(iii) It should be Reliable.
(iv) You keep your system highly Available making your system eventually consistent.

③ Get Estimation / Back of the Envelop Calculation

① How much compute and storage and what will be network bandwidth, you must have numbers to make the infrastructure ready.

Let's Assume :-

Total Customers / riders = 500 Million

Total number of drivers = 5 million

Daily active riders = 20 million

Daily active drivers = 3 Million (since most of the drivers are ready to give service)

Total storage needed for all the drivers if Rider's data (i.e id, name, vehicle registration number) is 1KB

= 500 Million × 1KB

= $500 \times 10^6 \times 1$ KB

= 500 GB

Every day, users / riders are increasing at the rate of 50,000 / days

↳ 50MB/day is the storage requirement

for the next 1 year

Current → 500 GB (storage cap.)

Next 365 days = 365 × 100/2 MB

= 182.5 × 100 MB

= 18.2 GB

In 1 year, 9 require $518.2 GB \approx 520 GB$

so as to make this application work smoothly for an year,

Everyday, let drivers (new) come in the app as 5000

In an year total new drivers = 365 × 5000

$= 365 \times \frac{10000}{2}$

$= 182.5 \times 10000$

$= 1.825 \times 10^6$

Now, Each drivers meta data let say take 1KB of space,

Then Extra space required in next 365 days = $1.825 \times 10^6 \times 10^3$

$= 18.25 MB \times 100$
$= 1.825 \times 10^9$ Bytes
$= 1.825$ GB

Total storage for drivers $= 3GB + 1.825GB$
$= 4.825$ GB

Trips data storage:-
since, we have daily active 20 million drivers, so there will be 20 million trips per day
Let, 1 Trip data = 100 Bytes
1 day storage $= 20 \times 10^6 \times 100$ Bytes
$= 2 \times 10^9$ Bytes
$= 2$ GB

On 365 days,
Storage required $= 365 \times 2$ GB (for an year)
$= 730$ GB

Noted Points:-
① Everytime a trip happens, it results in queries/API calls
At what rate all these things are happening, how many transactions are happening, how many queries are happening, in order to understand the scale at which it is operating?

On 24hrs = 20 Million
$= 20 \times 10^6$

1 second $= \frac{20 \times 10^6}{24 \times 60 \times 60} = 230$ trips/seconds]

Traffic Bandwidth of Trip = x Trip/seconds x size of 1 Trip
/ Throughput
$= 230 \times 100$ Byte/second
$= 23$ KB/seconds

② Driver needs to update its location very frequently:-
Let, driver location data take 36 Bytes
Now, Daily active drivers = 20 million
So, Data to storage location $= 36 \times 20$ million Bytes
$= 36 \times 20 \times 10^6$ Bytes
$= 720$ MB
Throughput $= 720/4$ location/second $= 180$ MB/second

How many servers are required?
Let, server can take 8000 requests in a day.
Then, Number of servers Required = $\frac{20 \times 10^6}{8000}$

$= \frac{20 \times 10^3}{8}$

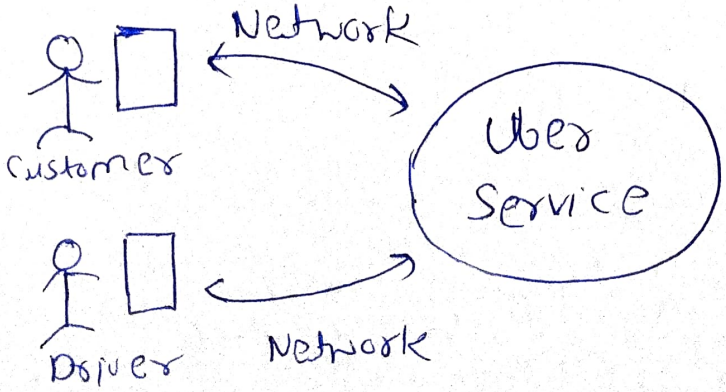
$= \frac{1 \times 10^4}{4}$

$= 2500$ servers

Why are companies trying to decrease the carbon footprint? What does it mean?
When you book a cab, it takes 5ml of water, so whenever you are booking a cab, you are depleting the natural environment resources. This is because water is used in machines in order to ensure machines run at the ambient temperature so that it can run all the time.
Resources used are Electricity and water, sometime for electricity petrol/diesel are used.
Because natural resources take some $ so big companies Microsoft/Google/Linked/Amazon etc. are focussing to minimize carbon footprint.

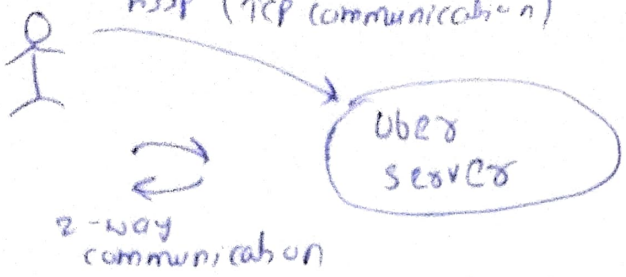
Note:- A prompt to Chatgpt also take 15-20ml of water to keep server machine temperature stable

(3) How an overall system looks like:-



Now, Refining the design of system:-

(a)

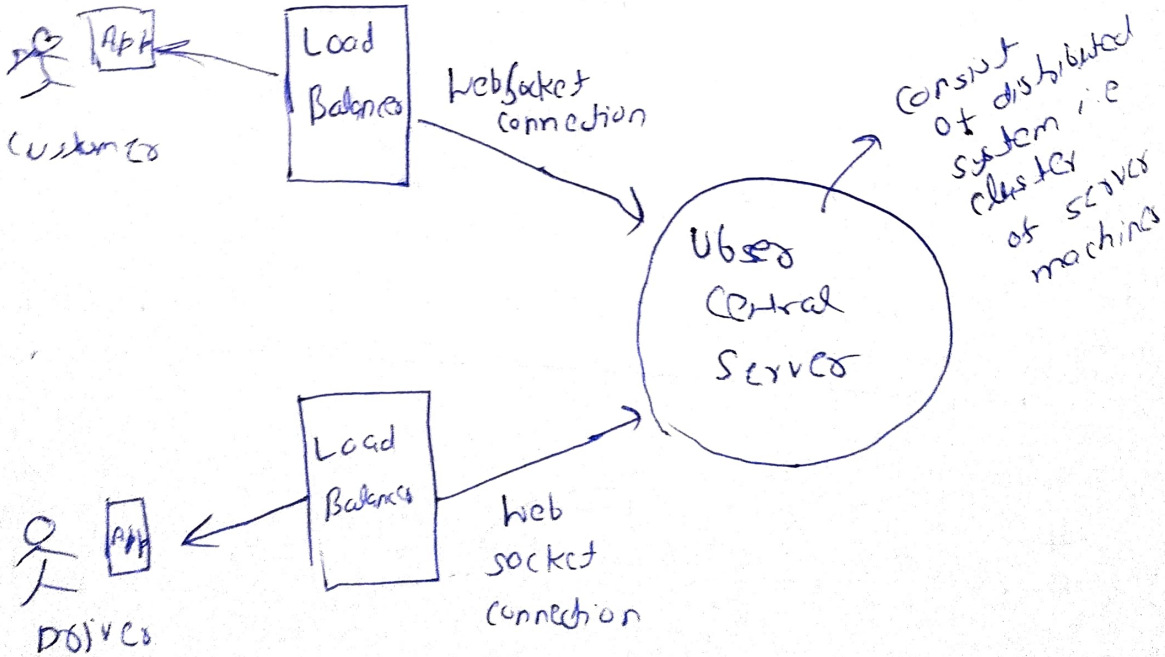


Approach (a) is resource intensive as http is short lived.

(b) So, to simplify approach (a), we can use socket i/o as it provides bidirectional communication but it is long-lived
So, n/w for customer - central Uber system as well as for Driver - central uber system will be the web-socket connection.

Note:- 2500 Central users system are there as we need 2500 servers.

(c) Since, there are so many servers and I have to balance the load, hence we need Load Balancer between customer & distributed system of User services, as well as with between driver and distributed system.



(d) Let's design Uber Central system

Uber central system consist of :-

(i) Driver locator
(ii) Trip Management
(iii) Location Updation
(iv) Drivers Matching
(v) ETA calculation

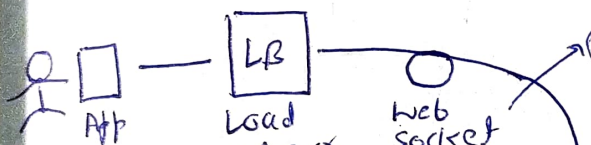
→ If I add all these services in a single machine, then this machine is typically doing multiple tasks

↓

This architecture is called Monolithic architecture

← HOWEVER, all these services are the specialized task, these are the individual task which should be ideally kept separately, so monolithic architecture is not a good approach, so we need to make it more granular.

→ so, better approach can be Service Oriented Architecture



Since this is used, so I will be calling certain APIs which this customer makes a call to this request accepting service

Request Accepting Service

make an API call to the system which can fetch you the drivers

Find Nearby Driver Service

so this system might call to findNearbyDrivers(lat, long)

↳ This service will make an API call, Can you find me the cars as per this lat, long position, so there will be a dedicate service for that

Driver locator Service

Provide you list of drivers based on lat and long

Location Manager

Web Socket i/o

every 4 seconds

every 32 seconds

Redis Cache

QuadTree DS

LB

Drivers

What are the APIs which the customers might be interested in calling? Some of the APIs called by customer:-

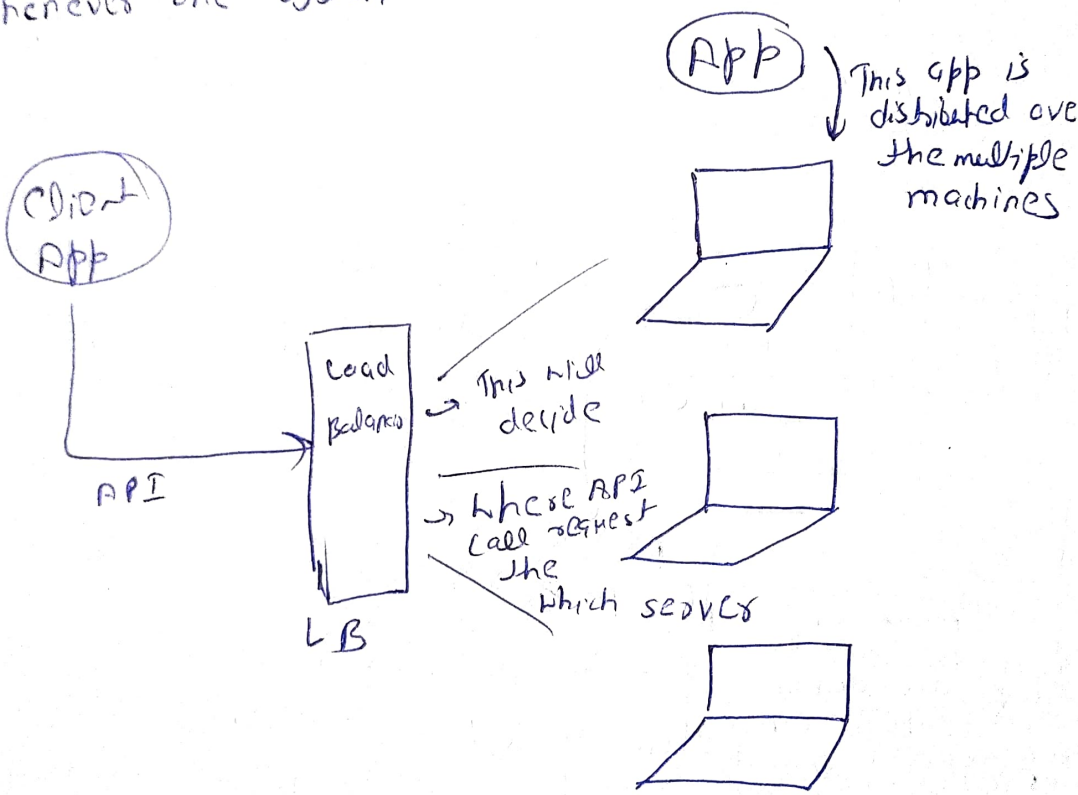
① Request Ride (rideId, lat, long, destlat, destlong, type of Vehicle)

(current latitude and longitude)

Why not chosen Microservices instead of SOA, because currently I am not sure about that Cuz my system is fully independent, so if you are not sure, you must start with SOA not directly Microservices

What is API

Whenever one system talks with another systems over a network, they normally do through the API itself, on the API when you are trying to hit the server / machine, in api you do not hard code the IP Address into it, you only go and provide the end points not IP address, so even if you are making a call to the API, you have to go through Load Balancer. On cloud, you say you have something called as API gateways which is similar to that we done in load balancing



How drivers location service works?

Let, drivers changes its location very fast i.e it updates it's location every 4 seconds.

Now, every driver has some kind of lat & long position which is updating where basically we are mapping the things on the different type of locations, datastructure that we use here is QuadTree Datastructure.

On Zomato, when you will review the different restaurants, each restaurant will have the location → latitude / longitude. And if you have to do it on the global scale, the data structure which very efficiently stores the locations of each of the entities over the hemisphere is Quadtree Datastructure.

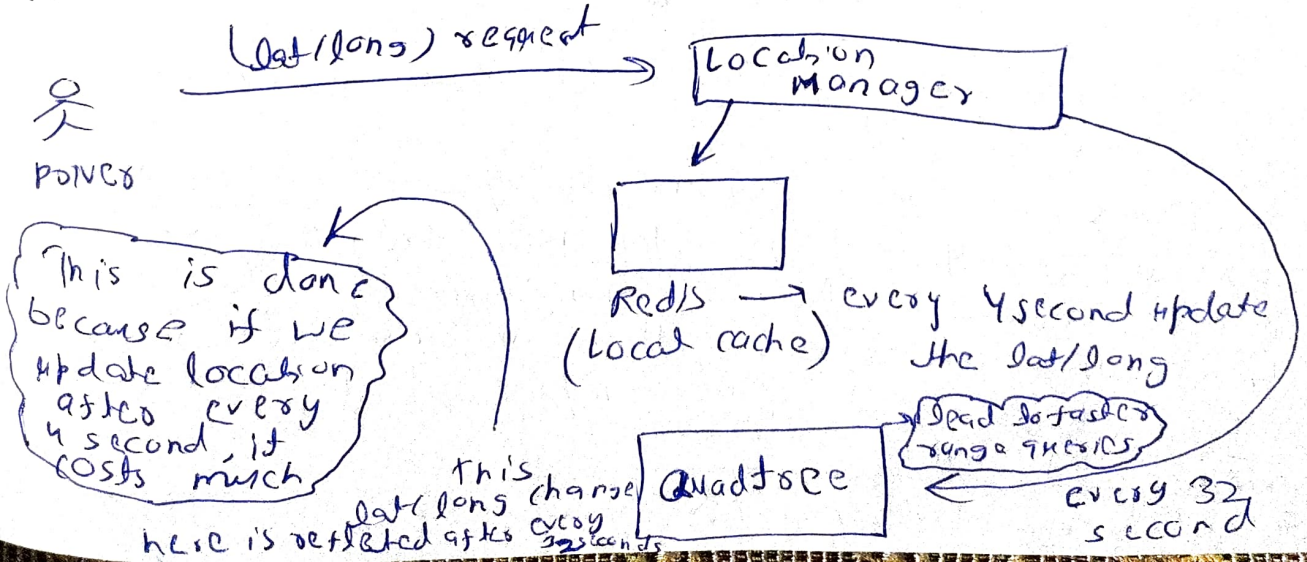
How driver updates their location

Let drivers updates its location after 4 seconds, whenever driver changes its location it makes a call to the Location Manager through socket i/o connection. Location Manager will update the QuadTree Data structure.

Now, whenever if you try to update it is very slow as updation leads to the creation of the additional nodes in the QuadTree.

idea is it helps in
① the fast search of any drivers location

② (lat/long) → driver location near to your location it provides



Note:- Quadtree Data structure is a persistent storage
Quadtree is updated by location manager as after 32 second, Location manager pull data from Redis and put it on QuadTree, location manager updates Redis after every 4 second so sync between Redis and Quadtree is managed by the location manager.

Database used ?

(i) Since I am dealing with huge data and it is scalable
(ii) It provides you high read/write i/o operation
(iii) So, Database can be used here Cassendra
In earlier times, Uber used Cassendra database but they replaced it by **Spanner Database**. Spanner is a google provided database which also provide you high throughput in case of read and write operation.

Note:- If your all database lies in the single machine then you can use RDBMS preferably but if data cannot be stored on single machine RDBMS is not preferred.

Driver might use updateDriverLocation(Driver id, oldLat, oldLong, currLat, currLong) as an API call to location manager

---

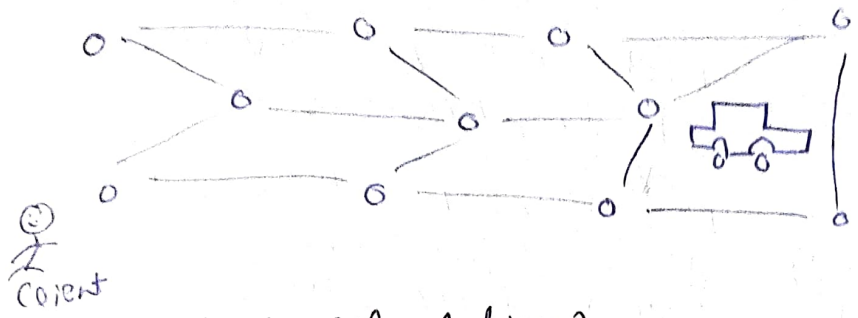
Why can we not directly fetch data from Redis as Redis is fast?
In Redis, we use key, value pairs so, if you query over a key, you will only get one specific value. Now, if you try to do the range queries then you have to scan all the keys, it is complex as it takes a lot of time

---

So, driver is doing write operation

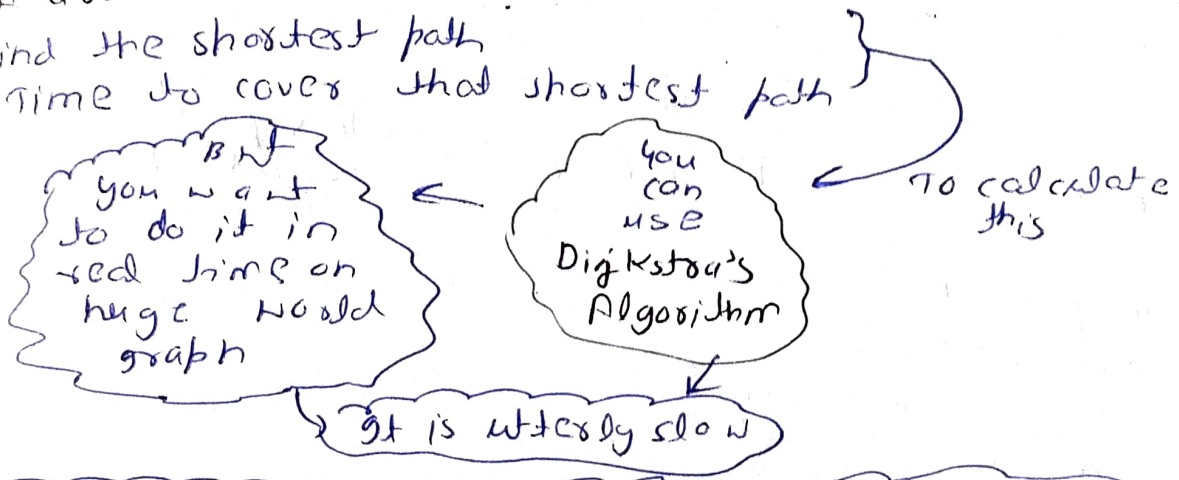
When you Book a cab Uber provide you ETA of where the cab arrives? How does Uber do it?

Any cab drivers make use of Google Maps Google Maps is nothing, it is just a complex graph.



ETA arrival calculation?

① Find the shortest path
② Time to cover that shortest path



Note:- In order to speed up the shortest path stuff, you always do the pre-calculation You always break the entire graph into the smaller partitions and in the background, for each two combination of the source & the destination you always do the pre computation.

But this whatever result you get is always stale data, in the real world, it is not determined by distance only, it is also determined by the traffic / accidents / Adhoc traffic routes, so today's modern ETA does not only require arithmetic / numerical solution / predications, you have to be smart and you have to make use of multiple data informations, so what happens in these days, you are also going to use the

Machine Learning library called as the DeepETA to predict the right ETAs

Noted Points -

Since, Ubes is an open source so they have shared the information about them that how are they doing the model, they used

Traffic Data
Map Data

Both are put into Routing Engines

Based on this data, it find out the shortest path / distance b/w two places and whatever data comes from this routing engine is transferred to

DeepETA → This predicts the right ETAs

So, ETA involves complex AI too. What is source/destination / time of request?