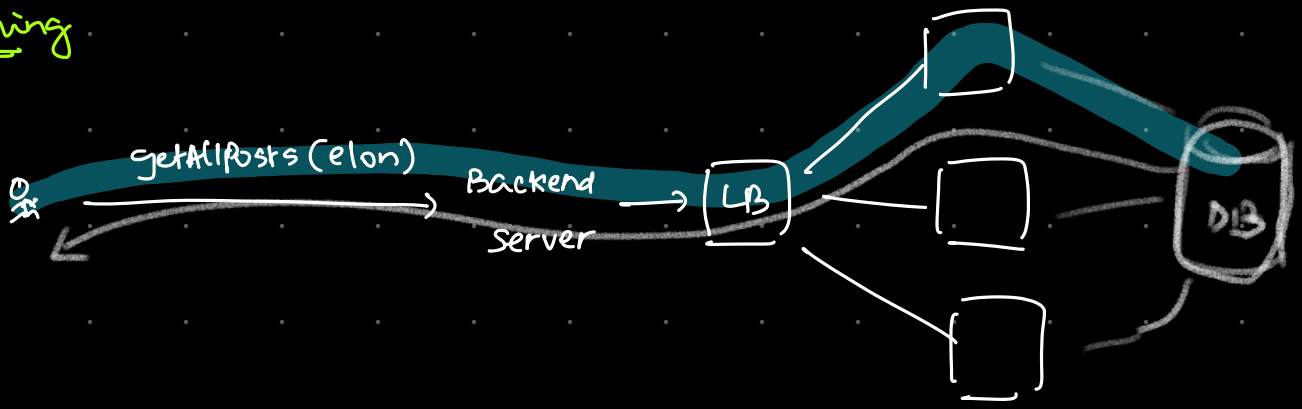
Caching

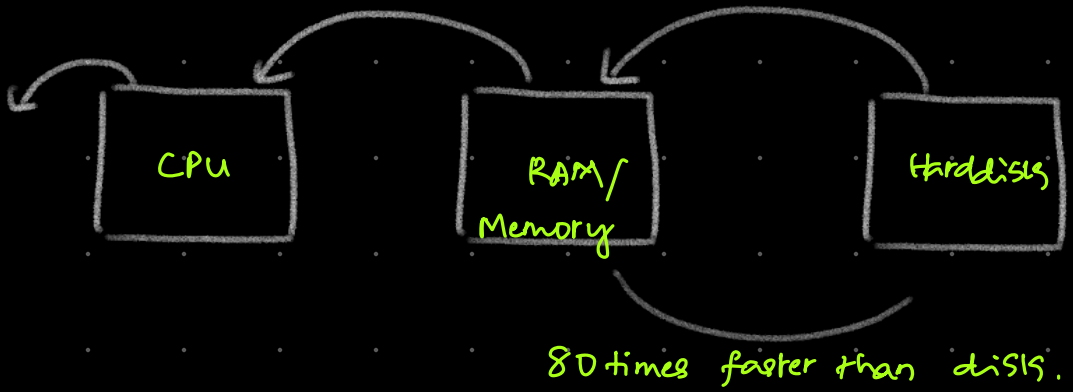


Where's the data on the database is stored?..

Ans: disk.

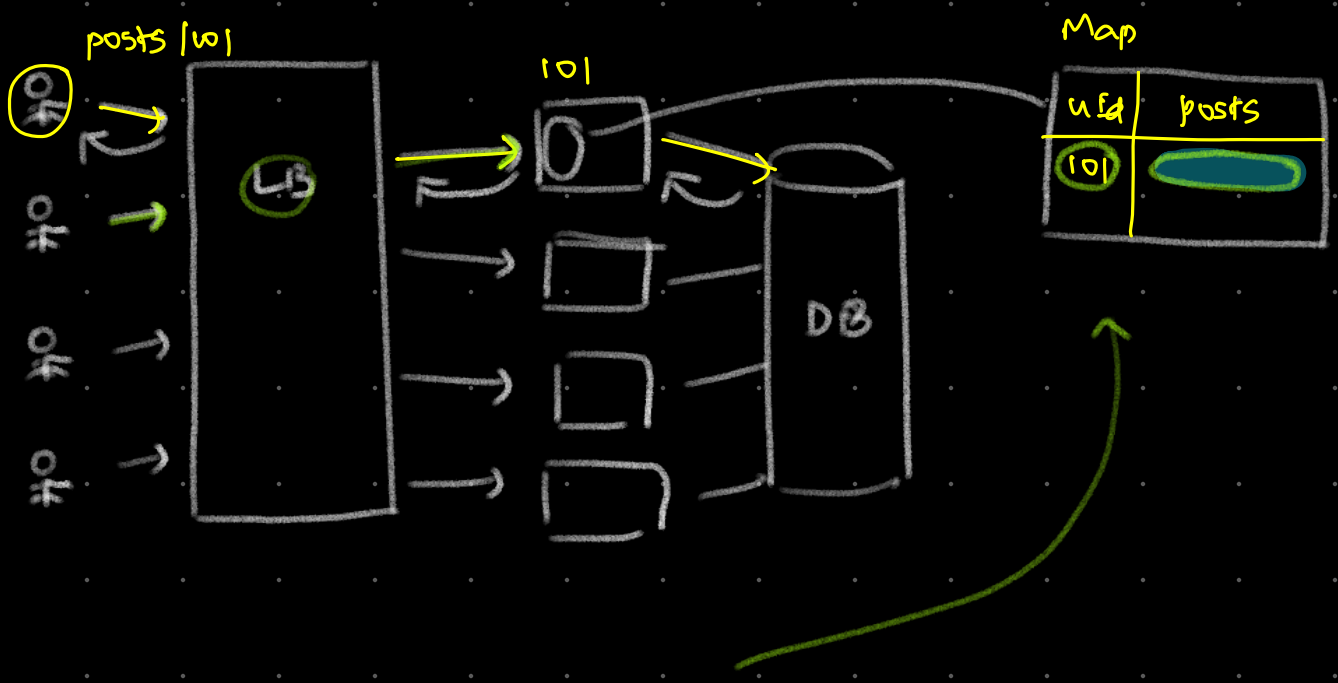
Getting posts of elon is a very common request

this would put pressure on DB & also slow down the api response time.



Can we do something better?..

$171 \times 3 + 171 \times 3 = 513 + 513 = 1026.$



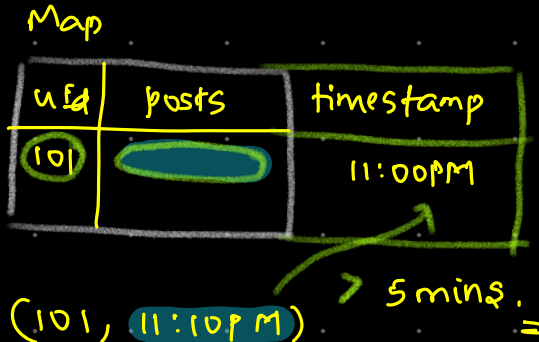
The data is stored on the server itself (RAM)

=) This will speed up the API response time.

Advantages

1) DB load will reduce
2) API response is faster.

How to update the cache. —



→ (101, 11:10PM) ← 5 mins.

=) You hit the db and update the posts.

(duplicate / processed)

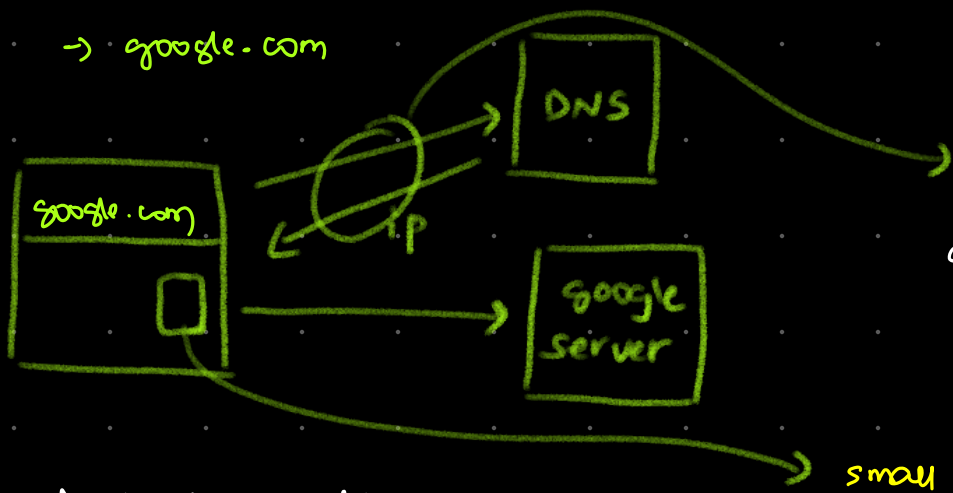
Caching: Storing the data in some place so that we don't have to recalculate it

ex:

→ scaler.com

→ youtube.com

→ google.com



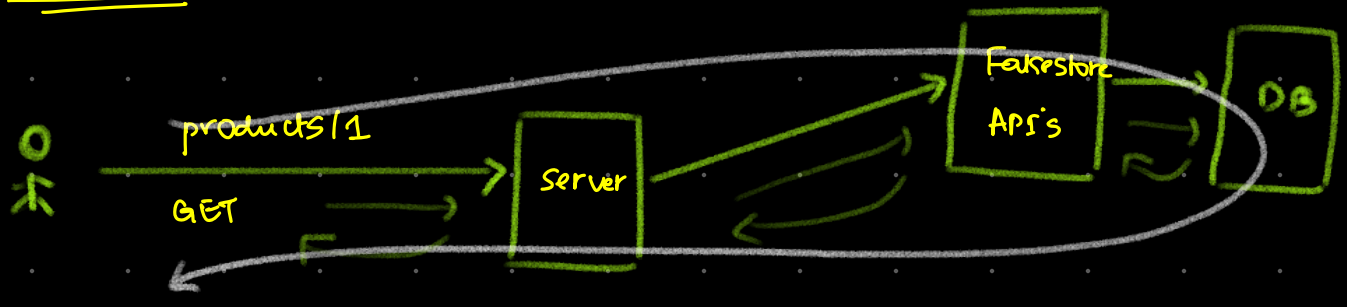
It's enough if you do this once a day.

small memory in the browser itself.

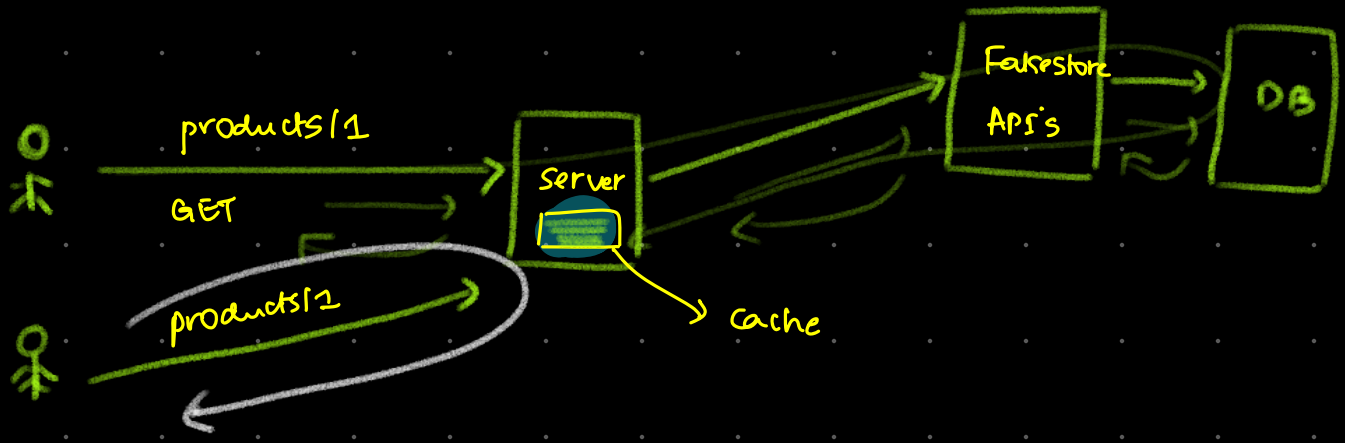
This is Client side caching.

Twitter example is server side caching.

In product service



In a minute, 100 requests to get the product details with id=1

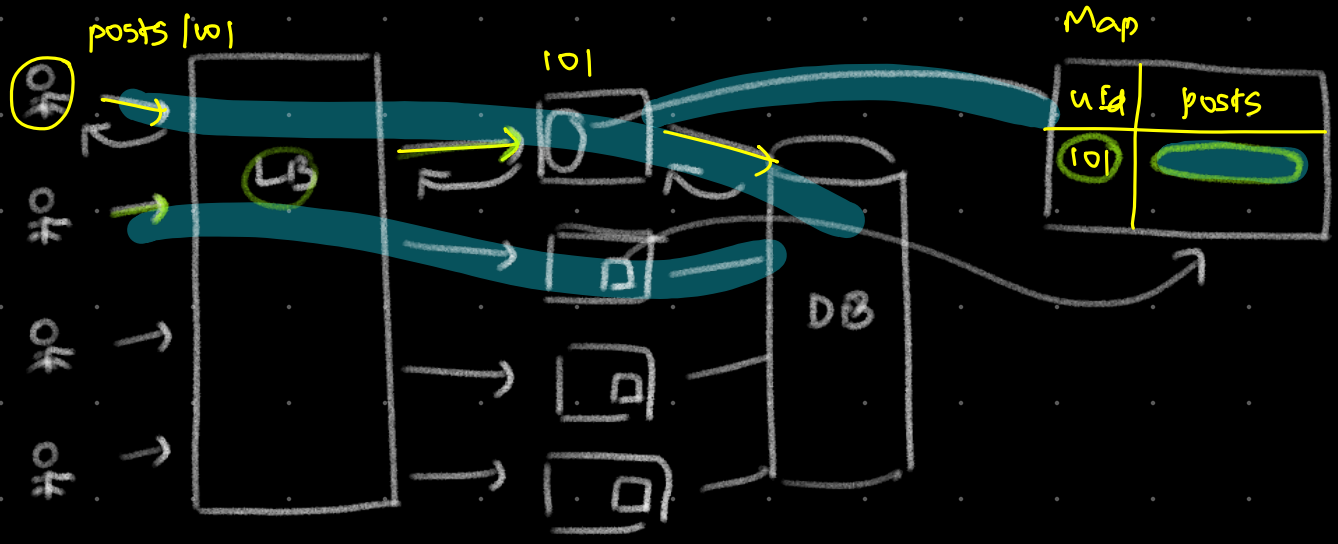


We hit fakestore api only once in a day.

1000 requests, 1 req will hit fakestore (slow)

999 req, they'll get response faster.

Issue; There's no guarantee that all the requests will go to same server.



Let's assume we've 50 servers.

1000 req are coming, req's are sent to all the servers.

1000 reqs → 50 are slow (to hit db)

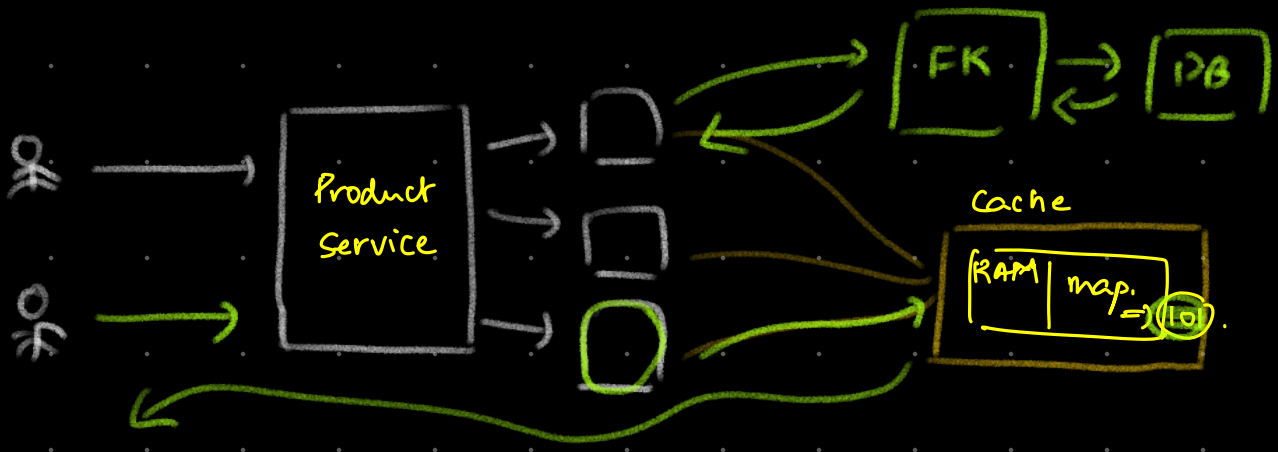
→ 950 are faster.

acceptable?

Maybe Yes :)

Global cache

Instead of caching the data on the app server, we cache the data on a separate machine.



Advantages

1. We're avoiding dep on other service
2. We're fetching data from memory not from DB.
3. Store the data in the format that UI needs.

In general, we keep both app^n servers & the cache server as close as possible on the data center.

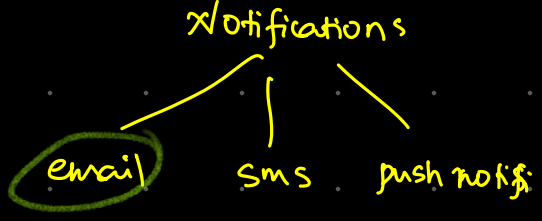
RDBMS → mysql / Mongo

Cache → Redis / Memcache.

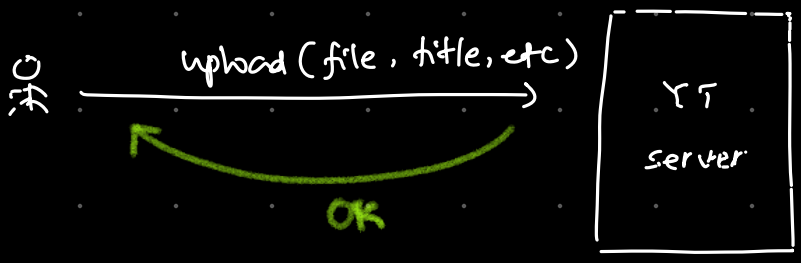
Hands on

Messaging queue

email service using Kafka.
(sending notifications)



Why Queues?..



behind the scenes -

- -> YT store video
- -> transcoding (store in other formats)
- -> subtitle gen
- -> copyright check
- -> breaching any terms.

processing

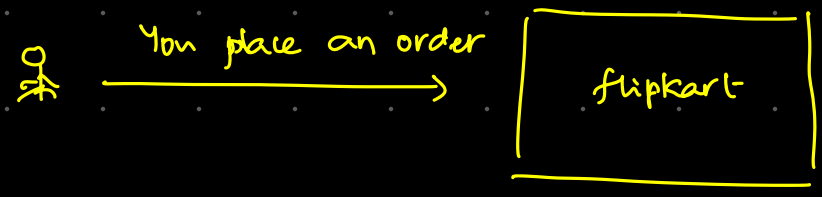
1hr video -> 10-15 mins.

All of the above things will be done asynchronously.

On the server side..

We shouldn't keep the requests alive too long.

Ex:



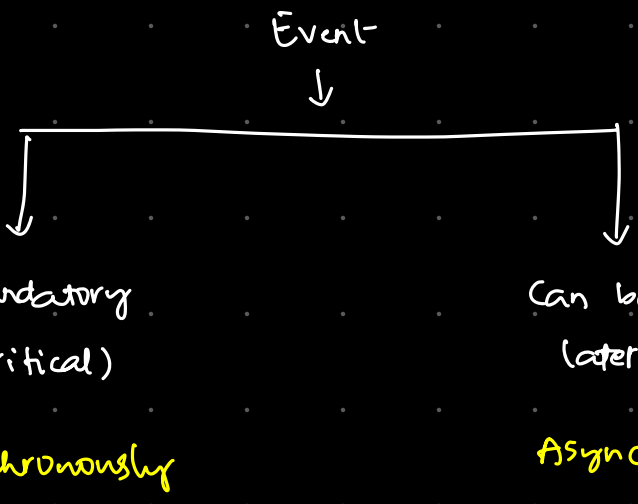
orderPlace()

- saveToDB()
- sendEmail()
- sendSms()
- notifySeller()
- notifyDeliveryPartner()
- SendANotifToGov()

etc.

Option 1:

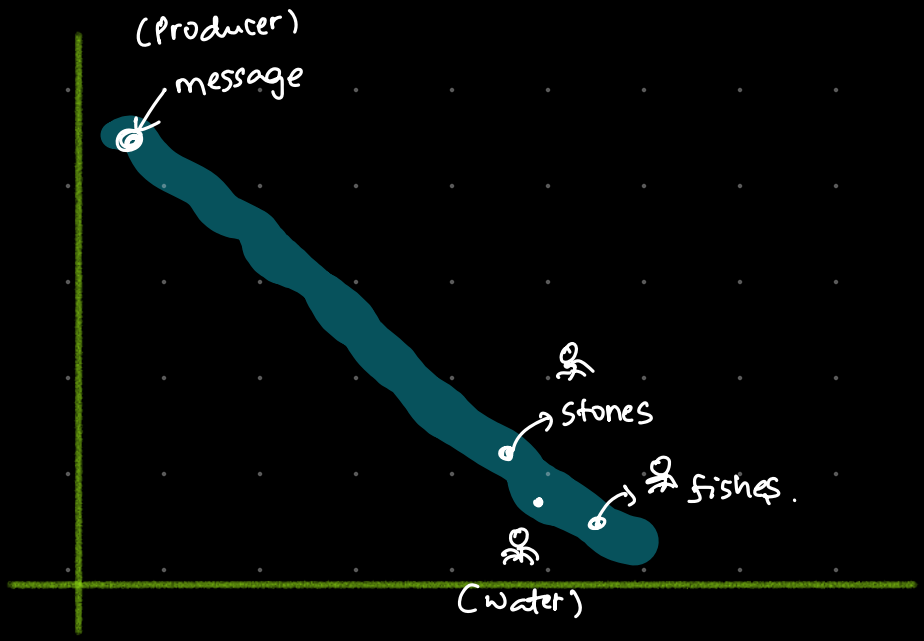
Option 2:

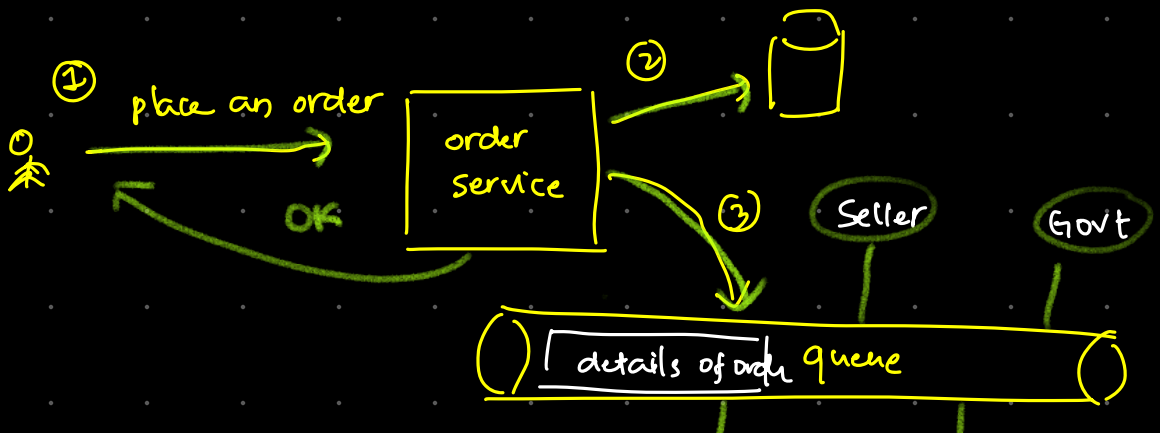


Messaging Queues

Producer : The person/service who creates the message

Consumer : The services which consumes the message.





Order service is done with its responsibility once it puts the message in the queue.