

Design

Youtube

Youtube (functional)

- upload ✓
- Search
- watch ✓
- Comment
- Recommendation.
- Analytics.

Youtube (non-functional)

→ Reliability → video should not get corrupted.

→ Scalability

↳ 1B active users / day.

↳ A user watch 5 videos / day.

↳ 100 watch : 1 upload video.

→ Availability or Consistency?

↳ Ans :- Availability. (Since we want my feed to always have video).

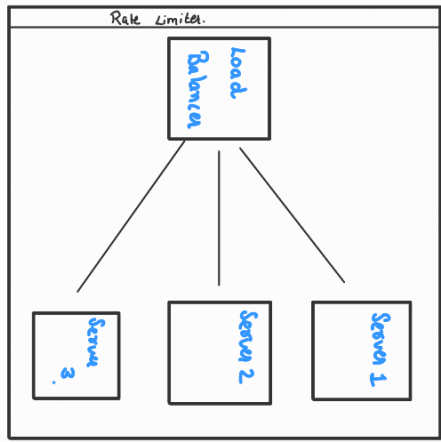
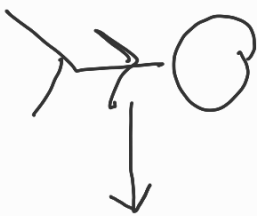
Consistency might have issue when :-

I am reading server 1 video.

but my content creator uploaded video in server 2.

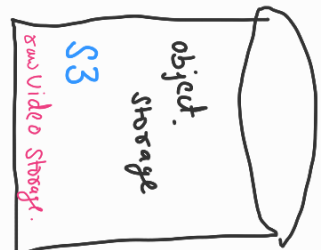
So, now it will take time to copy from

server 2 to 1. Therefore I might have to wait.



Cache (LRU)
for faster video access

* popular videos.



S3 will automatically replicate video for me.
So, no tension to loose data.



(For video compression.)

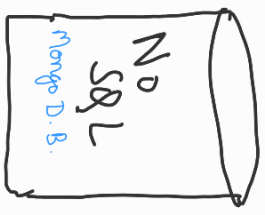
Encoding

Suppose 50m daily video upload.
⇒ ≈ 500 videos / sec.
⇒ Suppose, on an avg., it takes one min to encode the video.
⇒ So, we will require about 500 x 60 sec ≈ 30000 workers hrs.



Best because videos are immutable

CDN
(edge based)



(metadata & user info.)

upload (title, desc, video.mp4, UserID,
thumbnail)

To - watch

↳ we will divide the videos into chunk and load only that chunk which is req. at that time. This will reduce the latency.

s Secondly... we will use **TCP** because we do not want to miss frames in the video. while using UDP might lead to better latency, but TCP works best here.

fact

Is youtube originally used My SQL database. Later Sharded it And made an engine called **viteess** which made the Sharded D. B talk to the Application layer.

