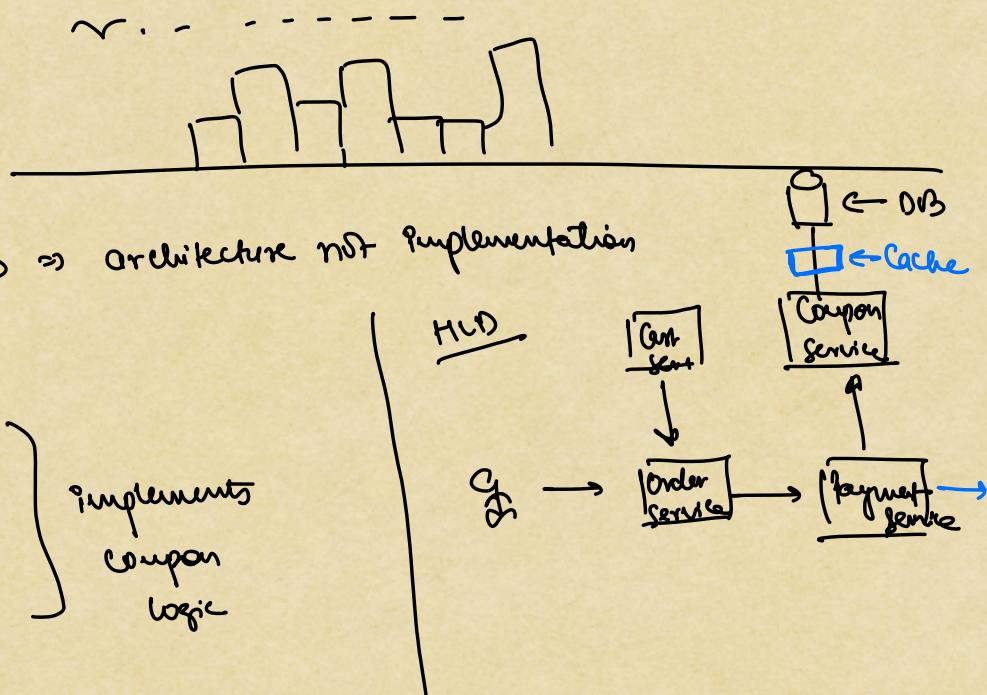


Agenda

- ✓ i) What is HLD?
- ✓ ii) Why HLD is important?
- ✓ iii) How to approach HLD?
- ✓ iv) HLD of OTT platforms (Netflix, Prime Video etc)
 - + Video ingestion
 - + Video streaming
 - + Recommendations
- v) Live streaming of sports event

⇒ HLD ⇒ High level Design

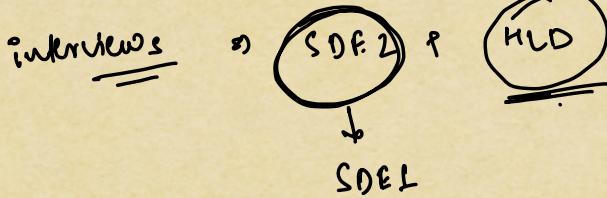
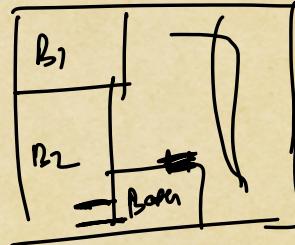
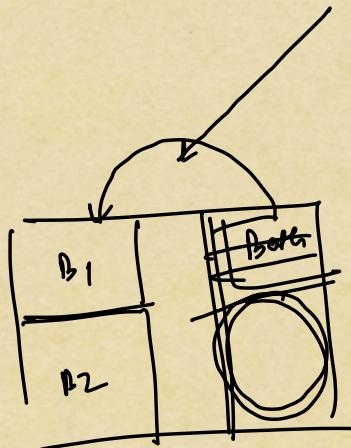
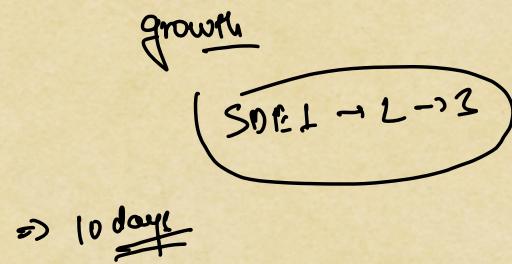
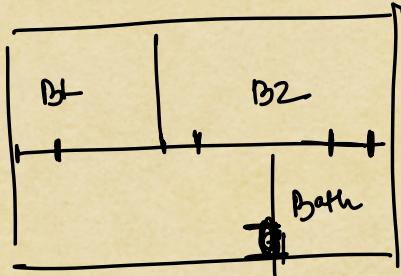
* Birds Eye View



Scalability ← { → What DB to use?
 → Cache required or not?
 → micro-services

⇒ Why HLD is important?

incremental change



→ How to approach HDS?

* Requirement gathering :-

→ SQL vs NoSQL

→ How many users?

→ How much data?

→ CAP \Rightarrow availability or consistency
(A) (P)

CAP theorem



Consistency
Availability
Partition Tolerance]
any 2 only,
can't have all 3

* Guesstimations [back of the envelope calculations]

→ How much data do we

need to store?

→ # of users expected?

→ request scale?

\Rightarrow Estimations for OTT

Youtube \Rightarrow 100 hr of content / min
1 hr video \Rightarrow 2GB

$$1 \text{ min} \Rightarrow 100 \text{ hr} \Rightarrow 100 \times 2 \text{ GB} \Rightarrow 200 \text{ GB / min}$$

$$\begin{aligned}1 \text{ year} &\Rightarrow 200 \times 60 \times 24 \times 365 \\&\quad \downarrow \quad \downarrow \quad \downarrow \\&\Rightarrow 200 \times 100 \times 75 \times 400 \\&\Rightarrow 200 \times 10^6 \text{ GB} \\&\Rightarrow 200 \times 10^3 \text{ TB} \\&\Rightarrow \underline{\underline{200 \text{ PB (year)}}}\end{aligned}$$

$$5 \text{ yrs} \Rightarrow 10^3 \text{ PB} \Rightarrow \cancel{\text{in one machine}}$$

Netflix

1 hour \Rightarrow 2GB \Rightarrow Youtube
 \downarrow
1 movie \Rightarrow 200GB

100000 video / year \Rightarrow $10^5 \times 200 \text{ GB}$
 $\Rightarrow \underline{\underline{20 \text{ PB}}}$

\rightarrow 5 yrs \Rightarrow 100PB

1 machine \Rightarrow 10TB

$$100PB \Rightarrow \frac{100PB}{10TB} \Rightarrow \frac{100 \times 10^3}{10} \Rightarrow 10^4 = 10000$$

machines



data is
distributed
among 10K
machines.

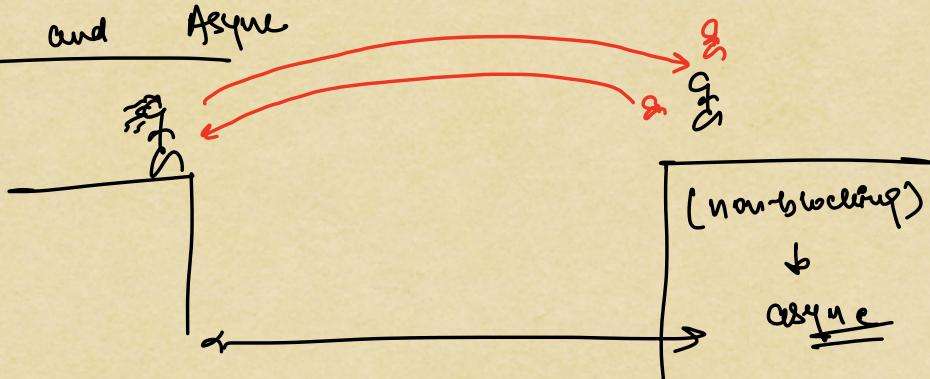
* Communications:

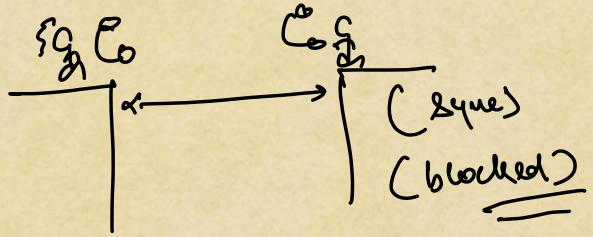
i) Sync or ii) Async

ii) Apps

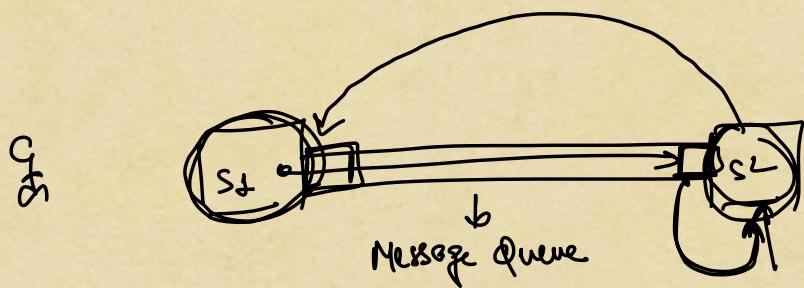
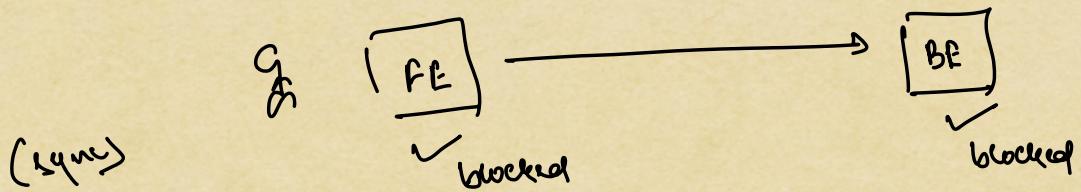
- upload the video
- stream the content
- ads
- recommendations

Sync and Async



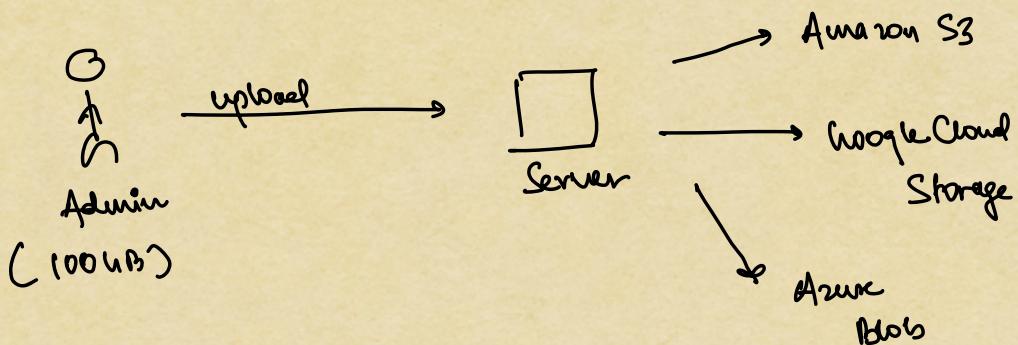


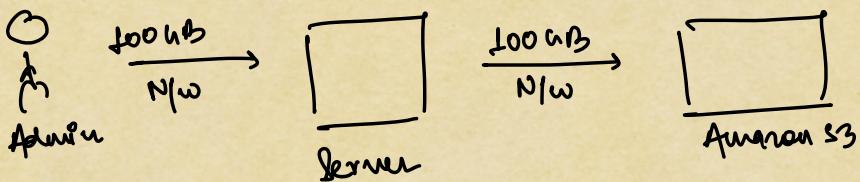
REST API



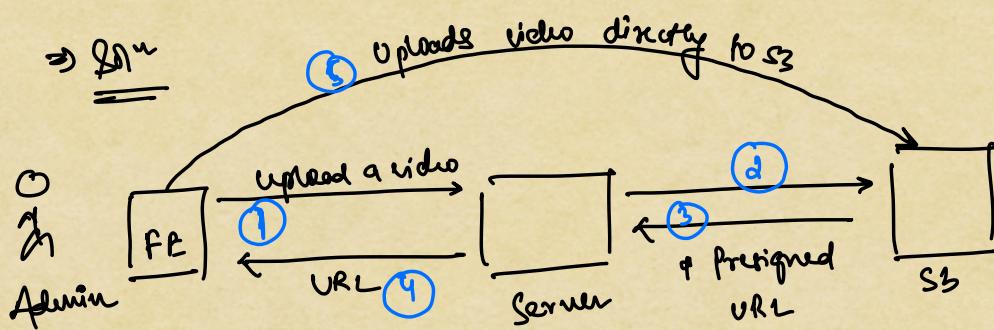
⇒ Design of OIT

⇒ Video Ingestion ⇒ uploading of content





- * Content is uploaded 2 times
- * High latency



Presigned URL \Rightarrow will not require any authentication

\Rightarrow highly reduces the latency

* What if client side uploaded 50GB and suddenly network failure happened \Rightarrow

\Rightarrow Should not retry again for already uploaded 50GB.

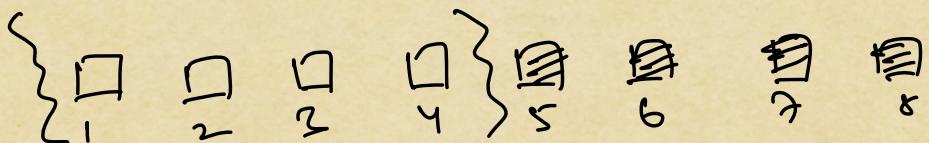


Chunkification \Rightarrow Chunkify the content

⇒ we divide the video into multiple chunks.

⇒ Chunks ⇒ duration or size

$$\begin{array}{ccc} \downarrow & & \downarrow \\ 1\text{ min / chunk} & & 10\text{ MB / chunk} \end{array}$$



↳ upload retry
should happen

* storage and retrieval mechanism should be streamlined

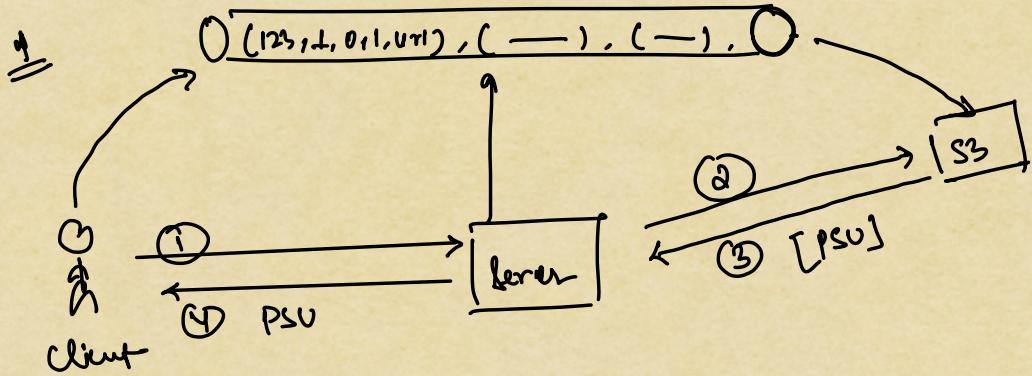
8 mins ⇒ 1 min / chunk

8 chunks ⇒ whatever order

* chunk should be identifiable

⇒ Amazon S3 built in support for chunk upload.

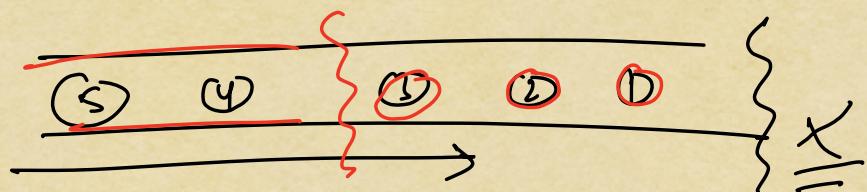
⇒ uploading ⇒ Async ⇒ Messaging queues ⇒ Kafka

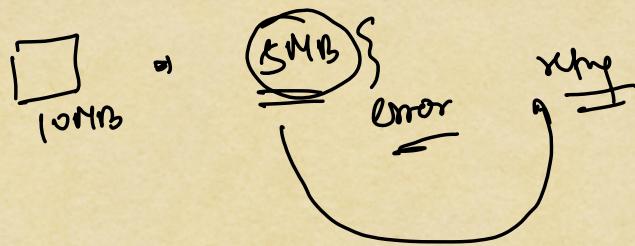


1) metadata transferred to server

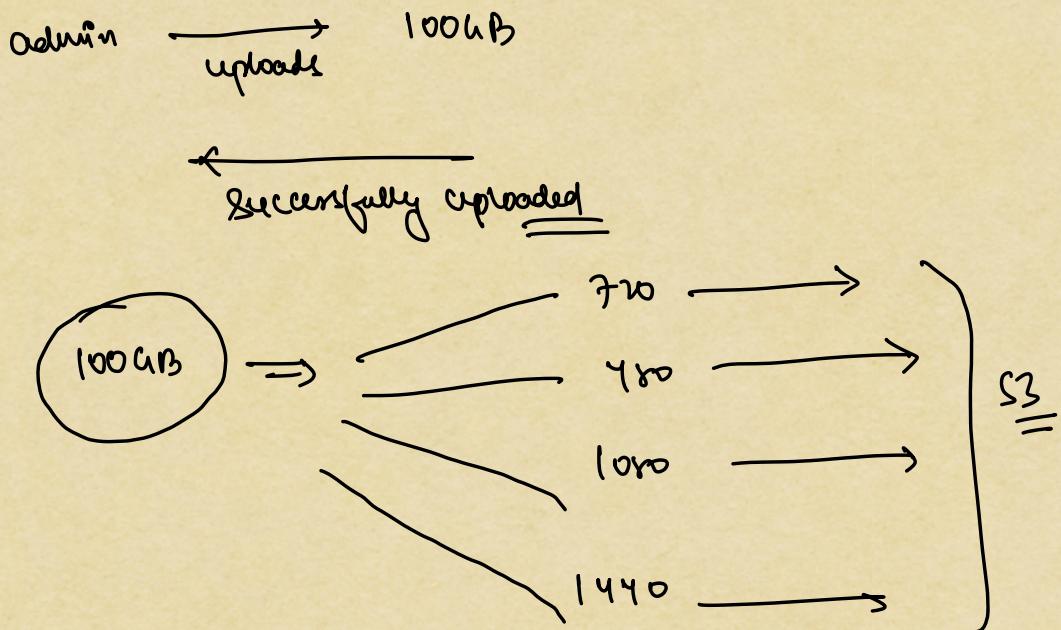
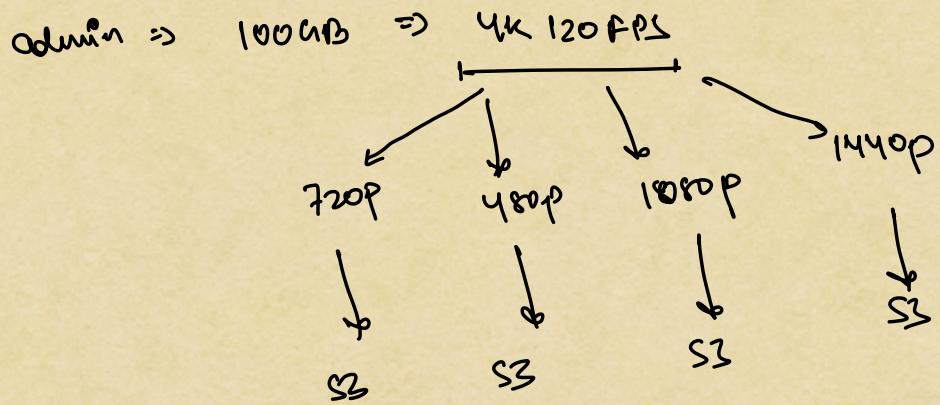
↓
video \Rightarrow chunks

videoid	chunkid	starttime	endtime	url
123	1	0	+	
123	2	+	2	
123	3	2	3	
.	.	3	4	.
!	!	!	!	
!	!	!	!	



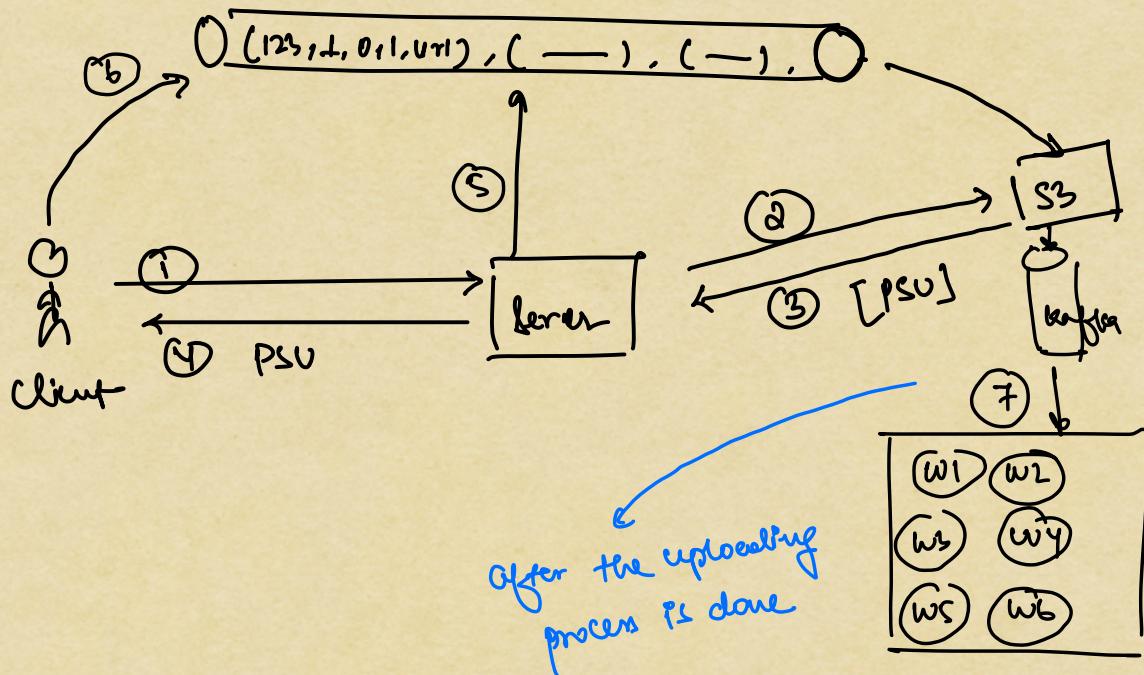


different resolutions

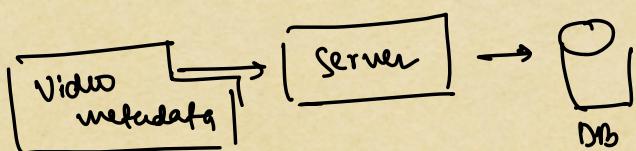


Worker Node

- i) Get the raw uploaded video from S3
- ii) Converts the chunks into multiple resolutions
- iii) Compresses the chunks.
- iv) uploads the converted and compressed back to storage [S3, Blob etc.]

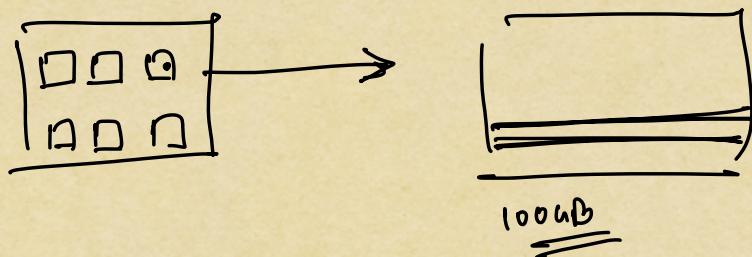


homescreen page



- 1) admins add metadata about movie
[properties]
- 2) admins choose the content file to upload
- 3) chunkification [automated]
- 4) connect to server
- 5) PLU
- 6) connection to Kafka
- 7) pushing chunks to S3
- 8) worker nodes \Rightarrow [convert & compress]

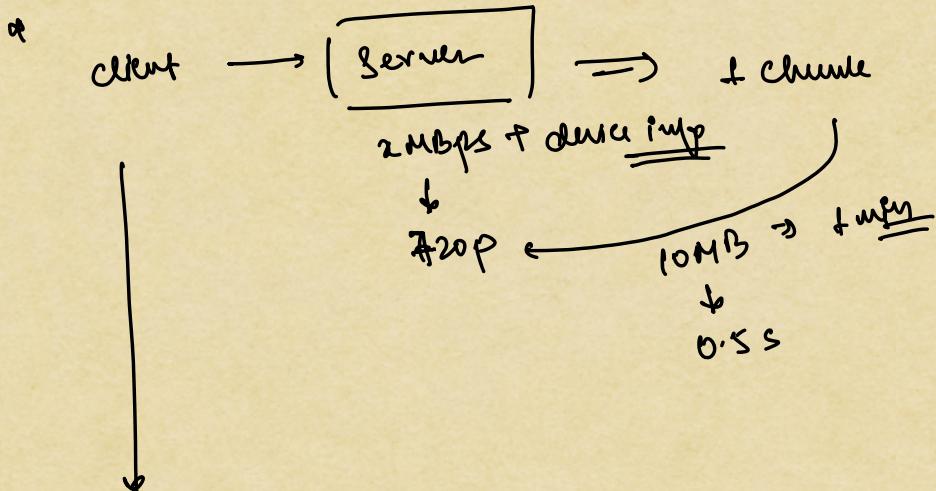
* Video Consumption



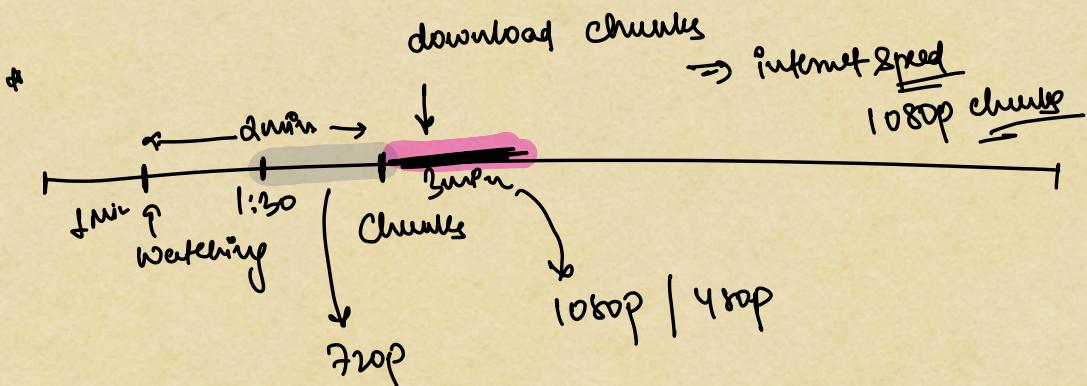
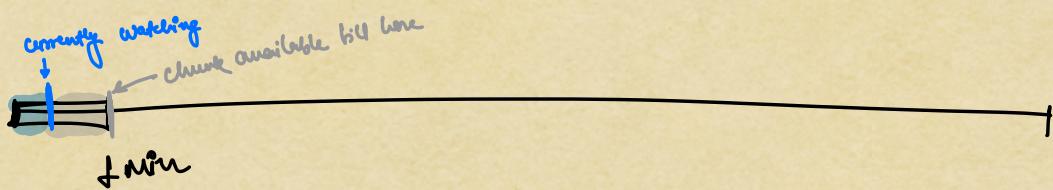
metadata for chunks

↓
Video is streamed on client side. chunk by chunk

* Internet Speed is Calculated

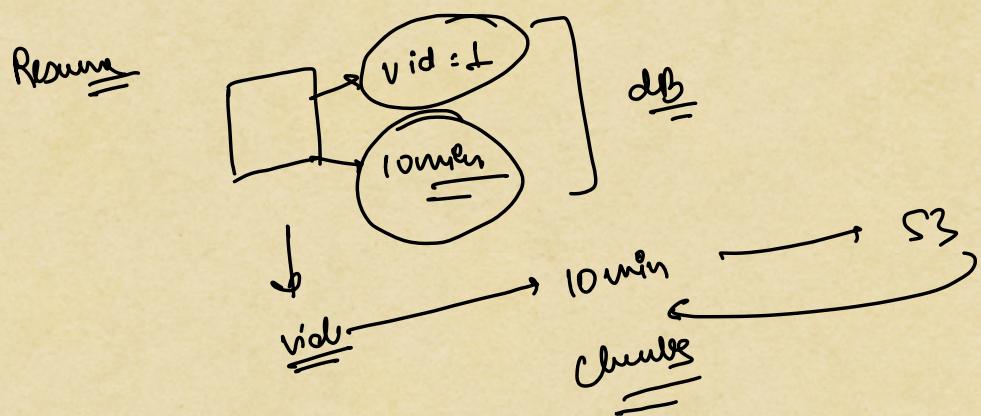
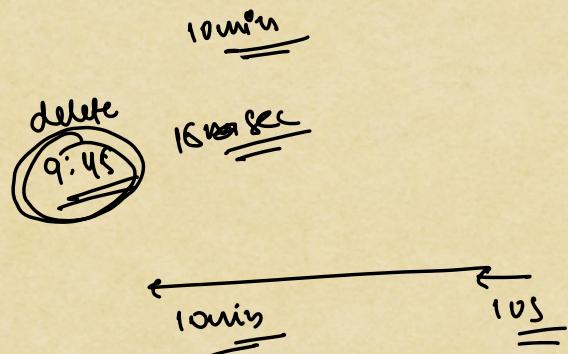
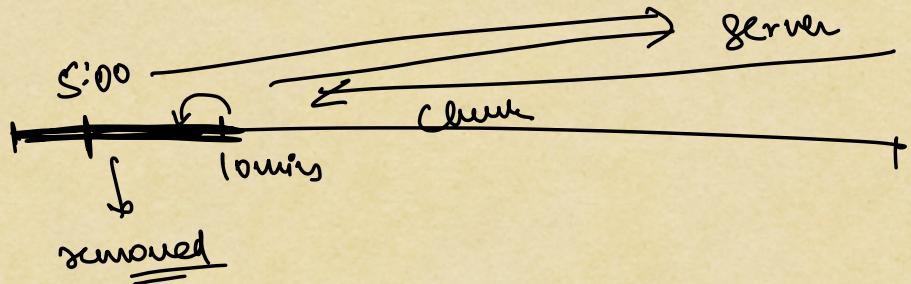


Logic to set the advanced chunk download

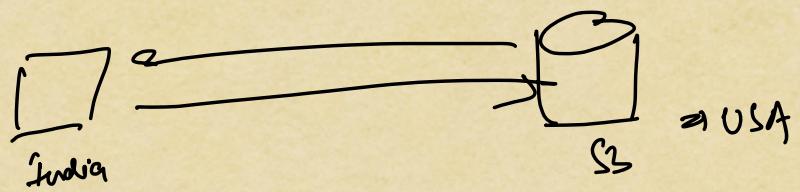


video ⇒ 720P ⇒ ⇒ 1080P

* already downloaded chunks will stay in same resolution



CDN Content Delivery Network





↓
distributed
across all CDN servers in the world

- {⇒ Akamai
- ⇒ AWS CloudFront
- ⇒ Netflix OpenConnect