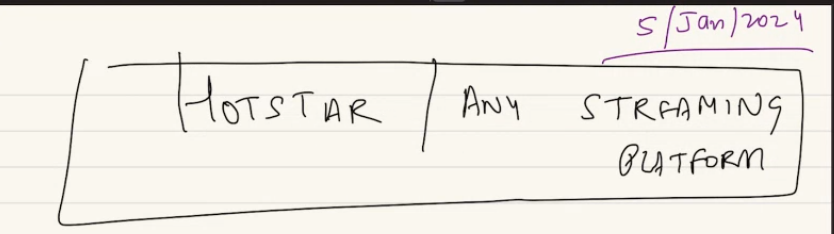




Designing Hotstar: online streaming platform.. will talk about any feature which is important for any streaming platform in general.

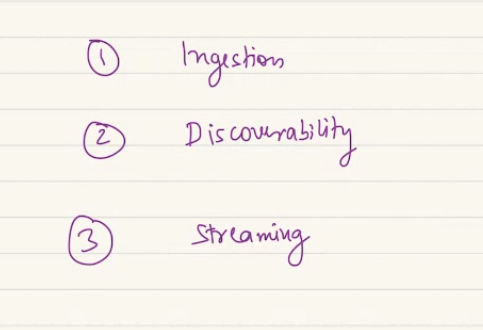


Most streaming platform like Netflix, prime they serve recorded content. But hotstar main feature has been live content. Will look at both features, similarity and problems here.

Static content – CDN will be used, for live content do we use CDN or not? Will look.

Hotstar or any streaming platform has 3 main component:

1. Actually the 3rd component: STREAMING: there is some video data which you want to stream. Streaming = deliverability…
2. Actually the first thing: as a platform you have to do ingection of data. How do you ingest, how do you store so that you can stream. In scaler ingestion is dynamically covered. Ingestion means: inputing data. In hotstar we want to see a movie batman. First thing is I will ingest the movie and save it in my platform. So when user play, I can server it. Ingestion: get to source, get it in my playform. How do you ingest- store the live cricket match and stream. So ingestion is very big component in hotstar.
3. Second thing: DISCOVERABILITY: covered across various class in several things. Say I have 100 of episode in my platform, as a user how can you search the content in my platform. User logins. In landing page user see suggestions.. saw these.. wanna continue or want to watch these that movie.. so discoverability: watch content, related content, search(full text, free text search) all these component comes in discover.



Ingest and streaming is related as I waana ingest in a manner so it is easier for me to stream to user. We will hop in between ingestion and streaming.

Firstly lest look at ingestion point of view: a video has different codex, diff ways of storing a video. Codec = Encoding + decoding

We know extensions like .mp4, .avi etc. a video is some sort of extension we store it. Also a resolution of a video. Each video has a resolution. On youtube you have a option to see resolution. Same we have for IMAGE. Resolution 480p, 720p, 1080p, 4k, 8k etc. a video at very assence is nothing but a bunch of pictures played one after another. Played in frame one after another very quickly. How many frame the video has the smoother video looks. More frames in a sec means smother vieo. Less frame / sec then video seems very fast to me. Person in car. Now walking on road if frame not there.

Generally we have 20, 30 60 FPS. Lower FPS less quality of video look like.

Resolution: diff quality of images.

When we go on YouTube we can see diff resolution we can select on.

Lets look a single image. An image is matrix of pixels. 2D.

Dot are pixels. Each dot has a pixel value. For black-white then every pixel will have value between 0-255 for color image, the picture/ frame is a combo of 3 frames. Means every pixel in image will have 3 values. One value will represent Red = R, G= green, B = Blue.

Computer vision me yeh sab hai.

Color image me every pixel has 3 values. RGB.

Resolution means: means what is the no of pixels on your screen. How densly populated on your screen. If larger pixel = better quality.

Less pixel = less quality.

More resolution = better quality.

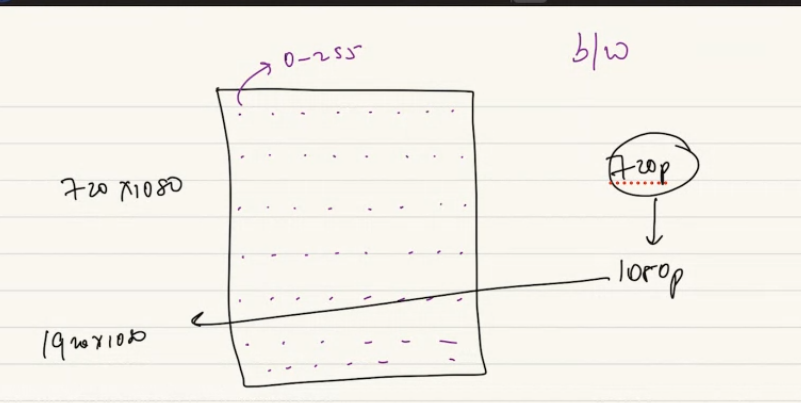
For interview always follow 4 step process… sometimes he had skipped some steps in some problem.

We will skip the steps today but look at how to approach a problem step by step.

A 720p image resolution contains 720 lines of pixels along the vertical axis, and 1,280 lines of pixels along the horizontal axis. Thus, the image is 720 pixels tall and 1,280 pixels wide. This results in a total pixel count of 921,600. 720 horizontal lines/1280 columns

720 pixel means 720p x 1020p..

1080 p = 1920 x 1080



Higher resolution = more no of data entry. Higher resolutiuon means more storage. More data you store per image. When you store high when you transfer to user more data needs to transfer.

High quality = high store= high transfer to user.

**Lets look at calculation:**

A movie is avg 2 hours movie. We live in color movie era. 2 hrs = 2x60x60 seconds.

Lets say we have 30 FPS is frame rate = 30 frames in a seconds.

2x60x60x30 = no of frame.

If a movie is 720P resolution… means

2x60x60x30x720x1280

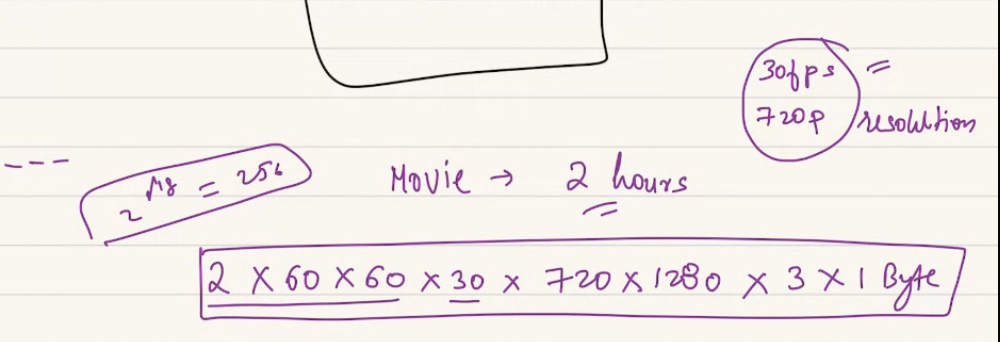
For every pixel, I sgonan have 3 values, RGB.

2x60x60x30x720x1280 x 3

Every pixel will have a value 0-255… how many bits required to store 256.. 8 bits.

1 bits me store till 2. So 2^8 = 256. So 8 bits required = 1 byte

2x60x60x30x720x1280 x 3x 1 byte



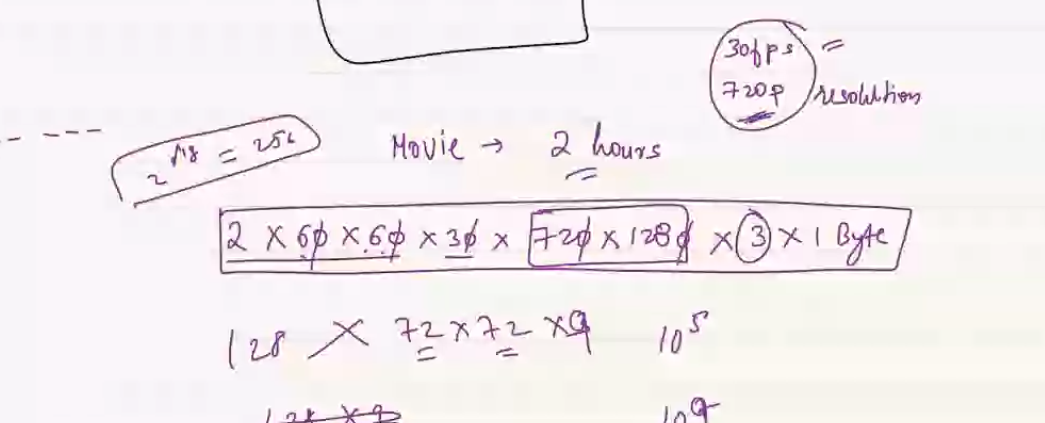
256 = for every

3= for every pixel will have RGB 3 values..

10^6 = MB,

10^9 = GB

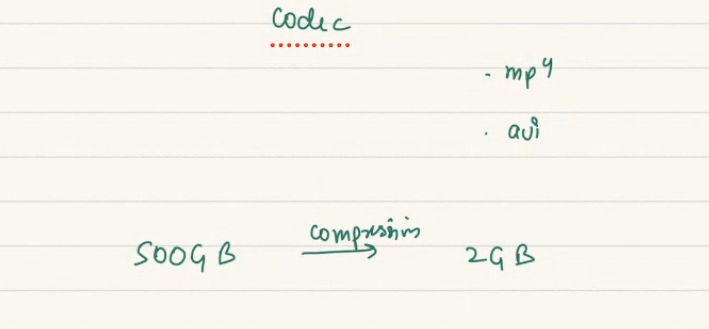
10^11B = 200-500 GB.. ans = 597GB



In reality it will be higher FPS,a dn righer resolution. Here its huge as it’s a RAW video= raw video me uncompressed video.

When you apply CODEC = encoding+ decoding = compression algo.

We get a codec/ compressed video in MP4, AVI format. Then we get a 500GB video will convert to some GB of video.



because each frame in a second has lot of similarities?

30FPS me most pixel/ frame will ve same. Codec will store the mimilar photo once.

But in a RAW it will be GB-TB of size.

As hostar we wil server a CODEC not RAW video.

If I pay crs of money to production the client will give me RAW video not a 2GB video in pendrive.

So I get a raw video, So ingestion happen on RAW video but play need to happen on codec video. When we compress, they are lossy compression. We loose some of it. If I am hotstar I will have ability of showing a movie in 4k, 8K, 256P, 480P, 140P. I have the ability to give a low or better quality video..

As hotsar I wont get one typ of video.. I will get a raw. My engg team will do and sell itto user. So in ingestion I will get TB of data for a particular upload.

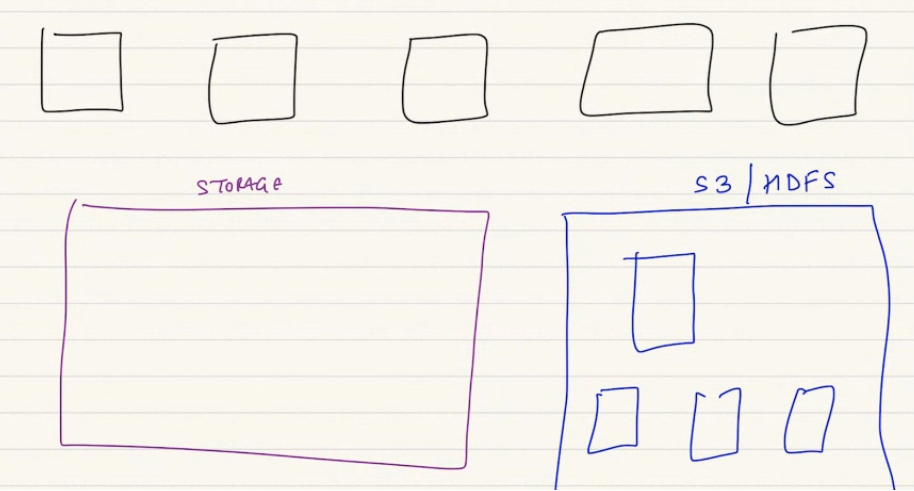
**Lets build hotstar step by step:**

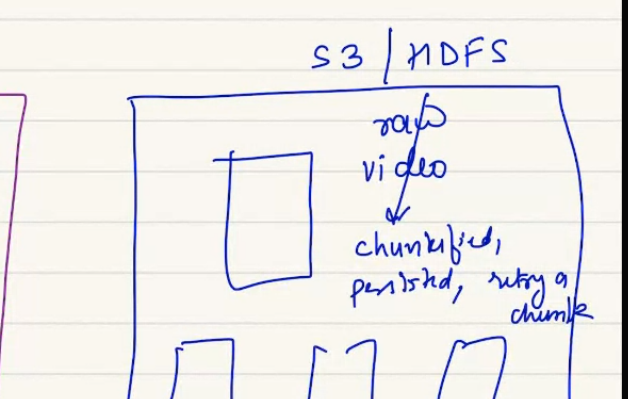
Here client is through which we will upload video to Hotstar.

Gateway… app server 🡪 DB/ Storage..

For a system like hotstar I will have S3 or HDFS kind of system which will store my video for me. There we have name node. Data node etc. I store these GB of data as a component in my S3/ HDFS. Inside these class we have namenode, data node. If a user is uploading and something goes bad. One chunk has to be reuploaded everything else will be uploaded.

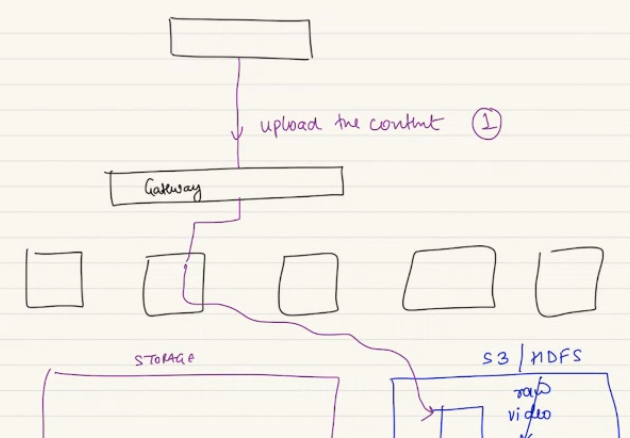
It should follow all principle: video is chunkified, persisted, retry in chunk.





As of now I have just uploaded the video.. first step of ingestion. There will be a lot of other things..

I have to do a lot of steps so my content is viewed by users.



What experience for end user we can imagine?

Based on that I can do things to my video,,

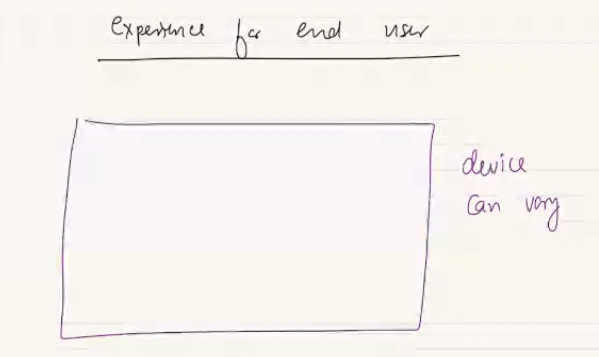
First thing: I might be using a mobile phone, laptop, tv screen, home theater. There can be various kind of device.

Second: the internet connection can be very bad, can be 5g, some plays in bad or high quality.

No matter if I am using am using a good or bad, I expect hotstar to server me good. Hotstar need to be prepared for every scene. So the experinace is very demanding.

When you wanna watch a movie in hotstar.. do you download first.. a 3 hr long movie of 2GB size user don’t wait for hours. Also user don’t have space to store a 2 GB movie. So no time to let it download or to store space.

So it is gonna stream to clinet and client will simultinuously watch it.



Most heavy-lifting will happen on ingestion end..

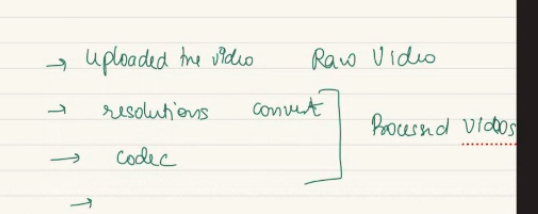
First thing we done is uploaded a raw format with S3 or file storage system of hotstar. Once that happen there after entire ingestion pipeline followes.

In Kafka class we saw, if we have a bunch of step to perform we can create a ASYNC..

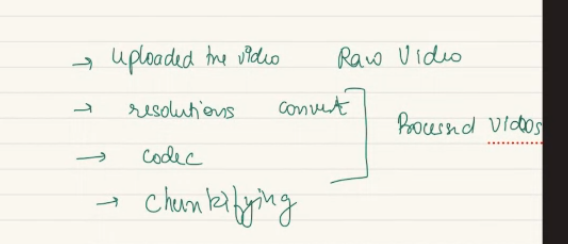
After I uploaded a video, I will say I will have to convert the video into diff resolutions.. 1080p, 144p, etc..diff user will need diff things. So I need copies. For each resolution I created I cant have in raw, so I need to do codec on top of that so size can reduce.

Once you do codec, then you have to do FPS, 30FPS, 60FPS. You will handle different FPS also.

RAW video ingested already stored in chunks in S3. We created the chunks first.



The processed video in S3. But that processed video I will myself have to devide in chunks. As I cant expect user download a 2GB video before watching. End user will download small small mp4 video. Watch it. Simultaneously request for next chunks..

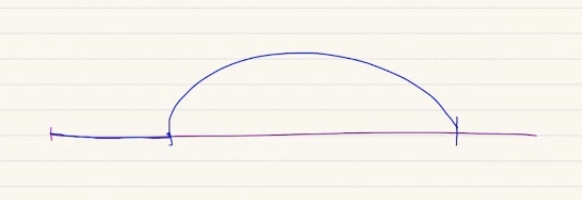


So again I need a logic for creating chunks.

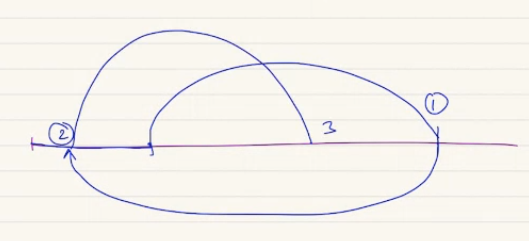
YouTube is similar platform. Ingestion happens in scale. For HT ingestion happens by hotstar team but YT me ingestion can happen by any content creatyor.. so in YT if we watch in smooth/ properly, it will stream properly. But if I fast forward it might get stuck..

As smooth me. While you watch 2nd chunk it loads next chunks priemptively next chunk

But when you jump a video. The chunk needs to loaded.



I am demanding a chunk which is not in my client. After a while if I go back , I may or may not have to buffer. As that part is cached in my device. But if device has thrown it I will have to buffer again.



Again if I skip to future at point 3, I will again have to buffer…

This chunkifying is different than the chunkifying we did on storing.

Me as the ingestion pipeline also have to create chunks , I give diff names. When user/ client watch it seeking takes time, going back don’t take time..

When I watch a recorded video, At scaler when ingestion pipeline, convert into chunks. After sometime recording will be available to consume. Not immediately. When we watch recoding, scaler send one one chunk I will watch it smoothly.

Codec Role: raw video me size is 500GB. Cant be served. CODEC does compression and convert a raw big video into smaller manageable video. All these is lossy compression. Audio is anyhow part of video.. audio is different stream of top video.

***Ingestion Pipeline:***

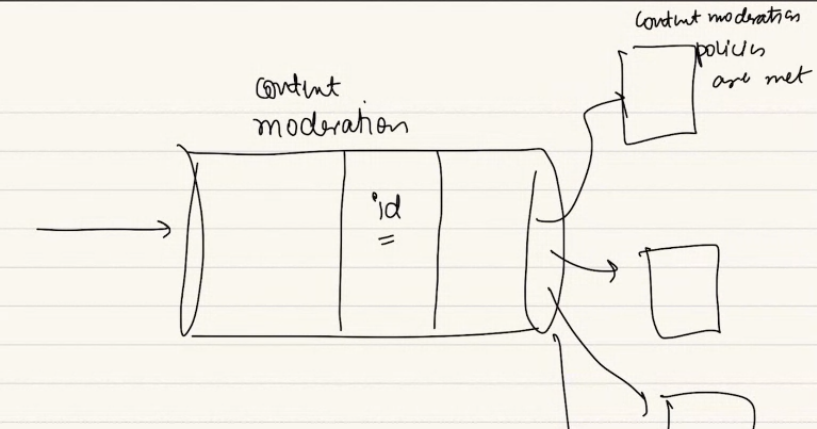
The first step is the client uploads the raw video.. and the raw video gets stored in file storage. In my normal database iw ill keep some metadata table where you keep a track of whatever happens on my content.

RAW video metadata will contain , upload ID , title, size, who is the owner. S3 link.. Etc..



That’s not all… the moment raw is uploaded by user, I will need to do some step. Codec, chunkfy, resolution conversion.

A platform like hotstar, YT, Netflix there may be they check community guideline check.. this much nudity. Copyright law, violence, check for public hatred etc… some moderation policy internally. YT will go and ensure its not a copy of someone else.. content against community guideline.. for that we might assume there is a queue.. a kafka kind of queue which is content moderation queue, whenever a raw is pusher. Insert the id of the video.. so a consumer machine who can parelley consue the video. Ensure the content mnodetaon policy are met.



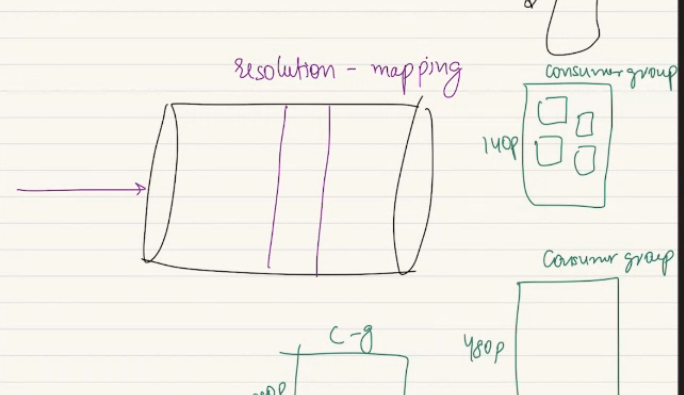
Its a combo of manual + automated process. Cant be completely manual..

One quee which is part of pipeline.

Once video is checked for content moderation. I will look at resolution. I insert if a Queue where I do resolution mapping.. a raw video is inserted, there will be diff consumer groups. Every messgage is read by every consumer groups..

Many consumer groups..

One can convert to 144p, another ocnverts to 1080P resolution.



So paralley we will do diff resolution creatyoion.,.

CG are reading and converting to resolution of choice.

After every steps whatever I am creating. Will do 2 things.. video will stored in S3, metadata will be mapped in db.

When RAW video is processed..

In live video these happens a lot faster.. we will cut corner.. for live me we cant do content moderation… in youtube in live me if you upload a video if it’s a copy YT will badn.. but if Live YT will catch later.. a live YT is few second-minute delayed.

Also a real cricket match inside the stadium vs TV there is a gap of 30-40 seconds.. that’s for satellite tv. For HOtstar like system its even a bigger gap. In same office multiple device has diff stage.. hence we check crickbuzzz..

It seems live, not live. It has lag as ingestion pipeline happens.

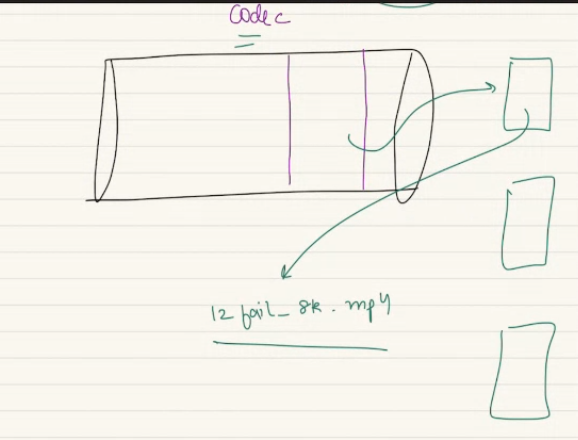
Live stream will come later. Lets finish content.

Normal upload rom client- stire in S3, do metadata. Put entry in moderation queue. Then insert in another queue for esolution creation. Diff consumer groups will consume same raw and convert in their choice. Then the consumer machine will store in S3 as a new files. S3 will have multipell copies of same. Raw copy video. Raw copy for another resolution and another resolutions.

S3 always stores in Chunks. Automatically creates 128 mb of chunks now phele 64mb chunks.

Each video we store need to go trowugh the queue which will create a codec version of it. If a have a raw video in 8k, 4k,144p.. each of these video will now processed inside codec . where one of machine will process, which will stroe a new copy..

Batman-8k.mp4 = sixze is into 100gb to few gb.. again will syore but lot smaller as its after codec.



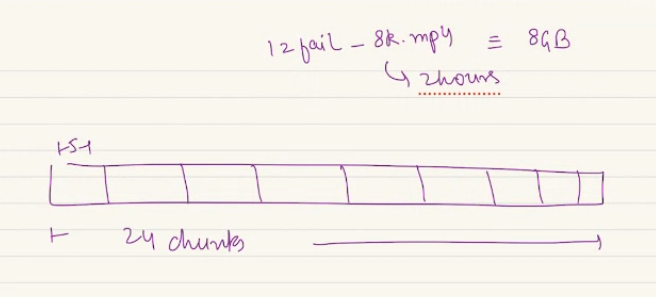
Hotsatr me multiple codec. Mp4 to avi etc. multiple consumer group for multiple codecs.

Here we say kafka consumer grup..

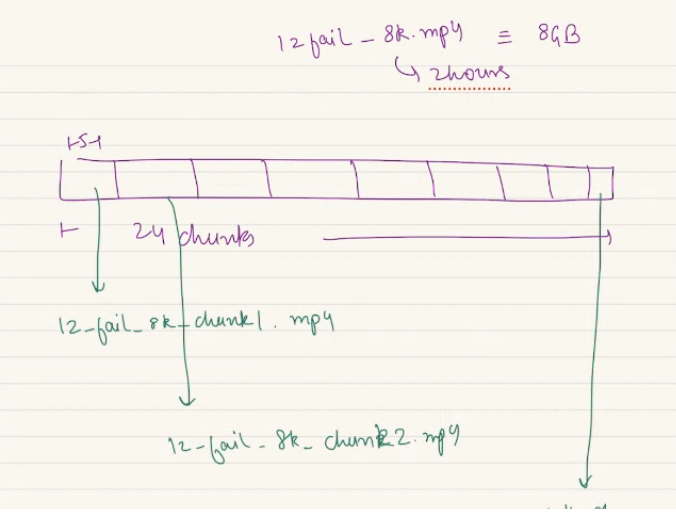
After codec ahs been stored. There wil be queue which will finally will do chunkify for me..

A movie picked up then chunk of 2 min, 5 min each will be crated. And intermideate videos will be stored in S3 or file stored.

Batman-8k.mp4 is 8gb.. 2 hours long. I wil create chunks of 2 min or 5 min. I get 24 chunks.



Depending on length of movie diff chunks will be created.



All these chunks wil stored in S3.

**Why S3 stores 2 diff type of chunks??**

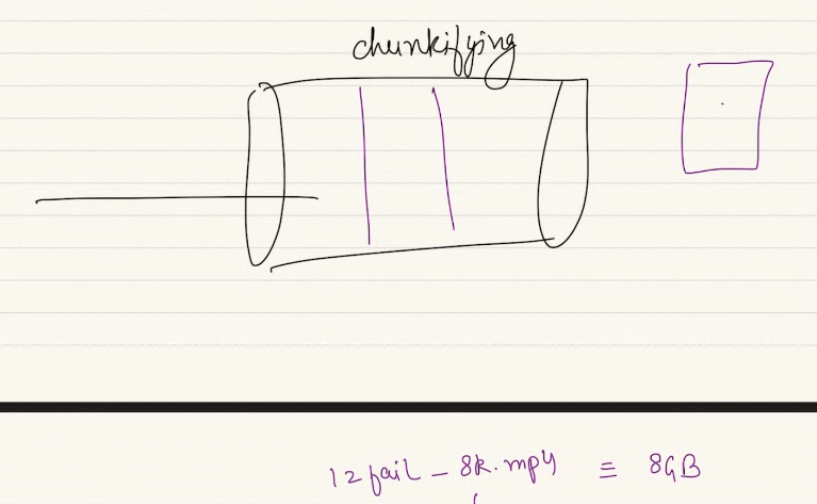
100tb or 256 kb file it will be stored in chunks. Naturally s3 stores in chunks. Chunks make it powerful. Read purely, retry in chunks as we store in chunks. Also between utilize storage. The chunk s3 interanally created, that need not be the exactly the chuck I willgive it to user. As the chunk I show to user will be different property.

S3 stores 128 MB of chunks.

I have a 8 gb movie and 24 chunks will be of 330 mb of chunk each. This is a number that’s coming as its 5 min long. In reality I can serve 2 min, 10 min each. Depending on use case the chunk size of every video need not exactly match. S3 wants to store in optimized way. The chunk si need to transfer in optimize way.

So my calculation will be diff. s3 for chunks for storage.

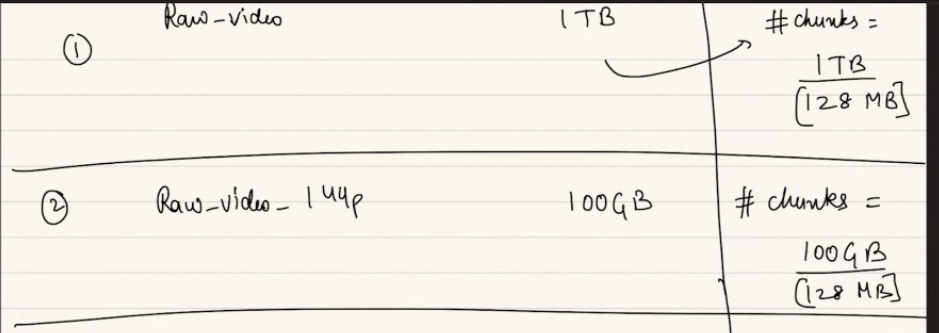
Chunk I want to server to end user.



Recap:

First thing I get a raw video of 1 tb from user. This will be stored in S3. In S3 wew ill store multiple chunks.. 1tb/ 128mb = # of chunks.

The raw video will be converted in diff resolution. Say 144p. is smaller limit. So my overall size of video is 100GB. So this copy of raw video will be stored in S3 in chunks.



This way we will have many copies.

For diff fps of also we will create copies.

Once we do codec to mp4. I am writing 720P by design. The raw of 144p will be convert to codec of 144p. 720 p raw will be created and converted to codec,,

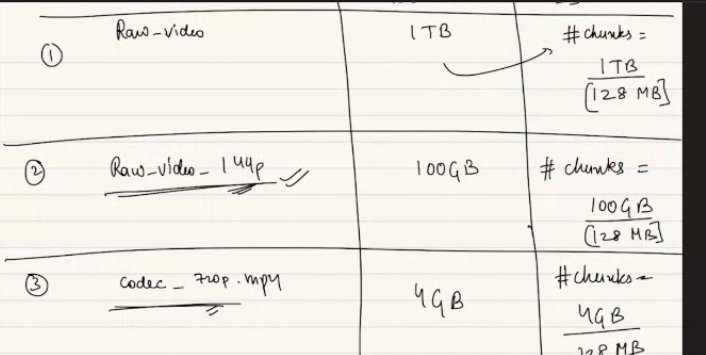
A codec 720 comes from a raw 720.

**can we do codec before resolution?** Probably yes. That deals with exact video procession=ng internals.

In kafka we can rearrange the steps based on requirement. But ultimately becomes a pipeline.

When it’s a encoded file it will be 4 gb. Will be stored in S3. No of chunks = 4gb/ 128 mb.

Raw video I will get should be highest possible resolution..



Mp4 I created is 4 gb which I will server. I will chunkified and serve user, .. usually it's the highest resolution which is available to the customer. If client can watch on 4k then most probably RAW will be on 4K.

Codec of 720 p is gonna have the first chunk, first\_5min\_chunk1.mp4…

Same way will create all these chunks for the video. Each of the chunks of 5 minutes. They will be of some size. Say 500 mb. This small logical chunk will also be stored in chunks of 128 mb.

Chunking at storage is diff, chuning at logical again be different.

**why are storing raw videos ? ultimately we are going to serve codec versions only..**

if I have paid 100 cr rupee to producer. I will but the biggest file. Then I will conbvert to every possible resolution. Still I will keep hold the ultimate source of truth. So I store ultimate copy. If I overburden then I can store in some sort of cold storage. Gets cheaper…Still I will save it.

2.30:

if i have started a video and pause. does it automatically get entire chunks for me as a part of buffering? Will come.