Hi Ashim,

Class details:

Batch Name: Academy Nov22 Intermediate 1

Topic to be covered: HLD: Microservices 1.

Class Time: 09:00 pm (IST)

Link to join: Classroom link

Following are the topics we shall be covering in the upcoming class.

Topic 1: Agenda

Topic 2: Services

Topic 3: Monolithic Architecture

Topic 4: Service Oriented Architecture Vs Microservices

Topic 5: Flipkart example

Topic 6: Client to Microservice interaction

Topic 7: API Gateway

Topic 8: Properties of Microservices

Topic 9: Key Design choices with Microservices

Topic 10: Observability

Topic 11: Observability - Distributed Tracing

Topic 12: Graphs and TimeSeries DB

Topic 13: Doubt Resolution

We also recommend that you solve the remaining problems from your past class/classes to have a better understanding of concepts.

We hope to see you in class.

Remember, consistency is the compass that guides student success - We urge you to commit to steady effort, an effort that starts with attending your classes.

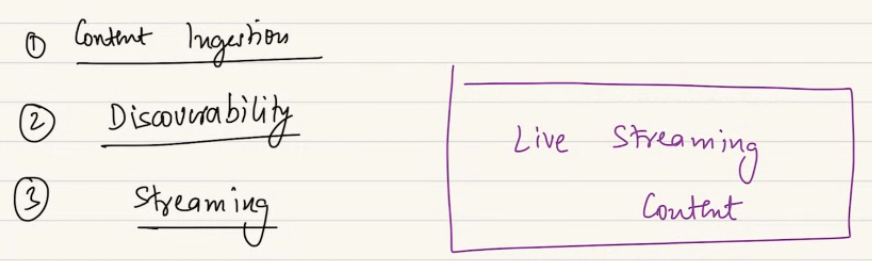
Thank You,

Team Scaler

How a system like HST need do handle data/ content ingestion.

Second point = discoverability. How user can search , how u will show landing page.

Stream the content once he selects a piece of content.



Live content streaming. Not pre-ingested.. happing right ow. Within short latency you will show to end users.

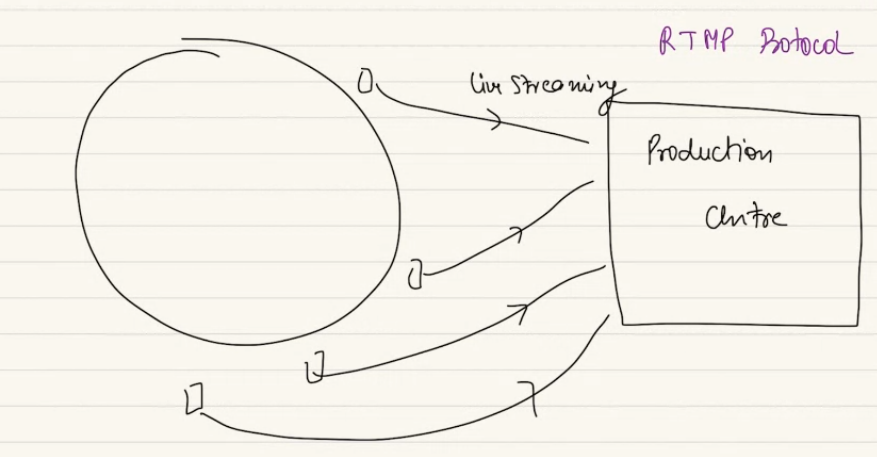
How will live stream content be different and canbe handled by hotstar..

Whats the structural difference:

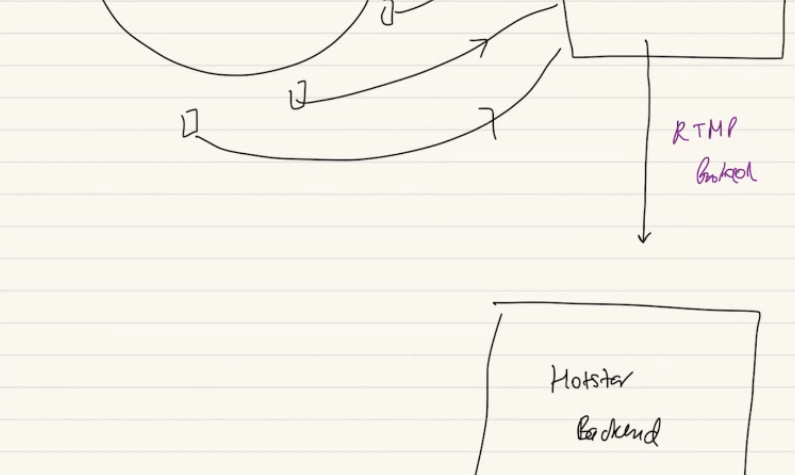
Content is being created right now. And you cant do ingestion pipeline in best possible manner and serve it. So the ingestion is gonan happen quickly and at same time yu will stream the content.

Lets look at pipeline. How to achieve. What twicks need to do.

Lets say we have a cricket match happning in Mumbai and live is getting recorded by camera. This live feed is getting transmitted to a **production center**. Office nearby the place of event. Live stream is coming to me. Gonna happen by physical cable or setelite. By a protocol RTMP protocol. Responsible for having the data getting transferred from place.



The same critic is shows in hotstar and star sports. Out of 20 camera angle. Production manager will decide what feed to be selected. Someone will select the feed to se selected what time.. multiple stream will be selected and combined. This PC wil transfer the produced content using RTMP protocol to backend server of hotstar. Real-Time Messaging Protocol (RTMP). Like a tcp, but optimized to transferring content in serilized.



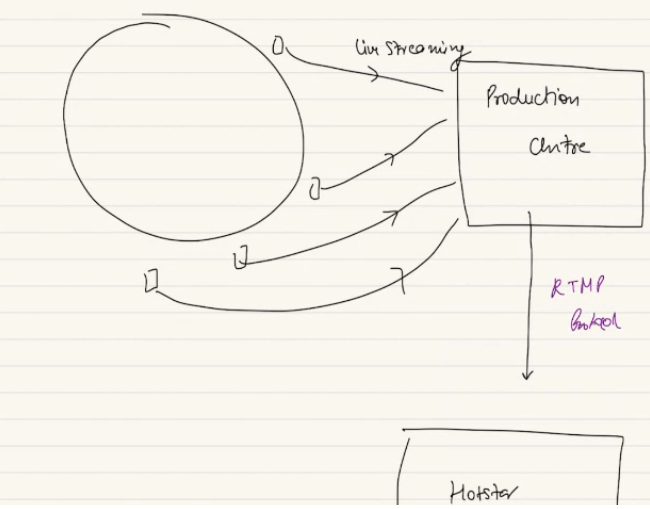
Crux of problem how will u server the data to million of user simultaneously. One to many.

1-2 is RTMP protocol..

1 to millon problem…. Concurrently streaming to any users.

This production data is transferred to hotstar backend. Any data transfer have some latency. Probably of few seconds of latency. Production 🡪 hotstar backend. prod centre can serve this to a diff platform as well,

Donot related hotstar use case to our live class or a live video call on whatsapp. There are some difference. Will come later.



Commentary is just a audio added on video. Which stream to use and which sound source to use..

Some segment me commentary and some me ground noise.

In case of jio cinema they gave select what feed at some point of time. At this camera angle. Jio does it. From prod center send multiple stream to backend rather one stream. Then you overwhelem even more by allowing user to select.

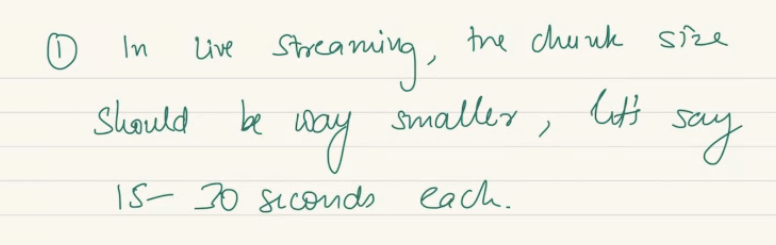
Hotstar/ jio cinema allow to change the view screen, a view where you can see a shorter view of ground. Another stream from backend to user. Max view is how hotstar try to save data transfer from server to user.

For content which is pre recorded, we do many thing.. converting to codec, chunkfying to smaller chunks. Same things is required in live stream.

Viewo codec si reqd as raw is bigger size, data consume. So we cant serve raw video. So this video which is getting transferred from PC to hotsatr server will be codec. To save latency between Prod serv🡪 hotsar server.

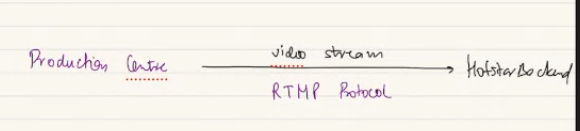
We have chunks of 2-3 min and we serve those content to user. This alow to ensure even chunk gets lost just retry few mb not everything. Also allow to make normal http call to request chunk from backend. So we will use this same chunify idea here. A client request a chunk id we server.

Can my chunk size can remain 2-3 minute? Hotsar backend will have to wait 2 min and convert and send to me. Cant effort to wait 2-3 min.. so in loive stream the chunk size should be WAY smaller. 15-30 seconds each..



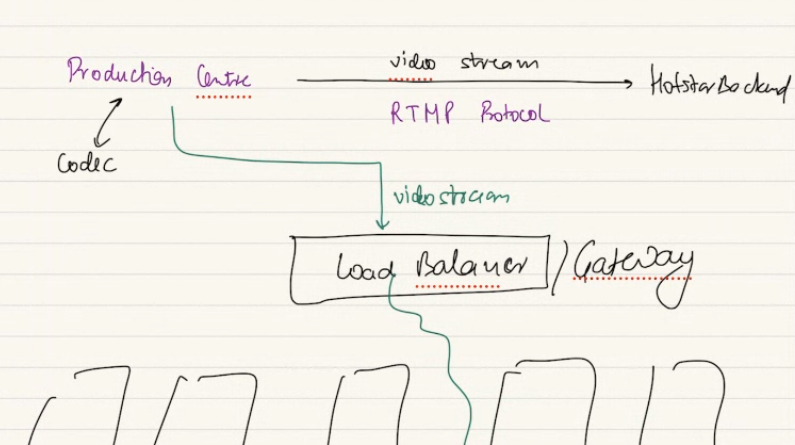
***Flow diagram:***

A prod center 🡪 send a video stream, over persistant connection between sender and receiver.. usinga protocol RTMP protocol.. it enables the transfer of live stream from PROD center to hotstar backend.



S3 me video file/ HDFS kind of system..

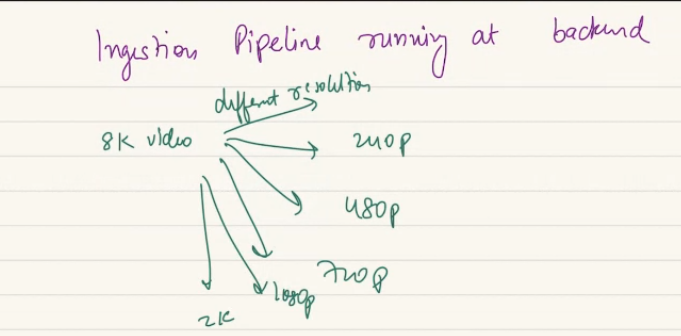
Prod center 🡺 video stream go tohotsrat backend. Stored in S3/ HDFS.. this should be codec. Ensure low latency. We wont ingest for a recored video.. as soon as I ingest I wanna server. Save latency. Bandwidth is not right word here.



A raw video is 1000 times more heavy than a codec video. A codec of 8K is very less. When you transfer a 1000 times big video takes latency. We ransfer 8K but the codec video. A raw will be 100-1000 times. Latencyof transfer will be higher. In live stream we will reduce latency as much as pssible. Ultimate latency user face is the sum of all latency. But converting to codec at start will be time consume. So we cut shot the data we transfer. We do codec earlier. We wont do many resolution here. Just the highest resolution. Rest resolution we can do later on.

Prod center: converts raw to codec.

Once we send the video to backend. Backend will do ingestion pipeline running at backend. Which will do a bunch of things. If I had received a 8K video. Will convert the 8k into multiple diff resolution video. Not every have bandwidth of highest resol.



For every resolution, we will create chunk. As I cant server the whole thing. We will get the stream from production server. We make chunk.

Will save chunk in S3.

For each chunk will create metadata. Will be stored in different database. This metadata will be needed by clint in real time. Will store the metadata so can serve.

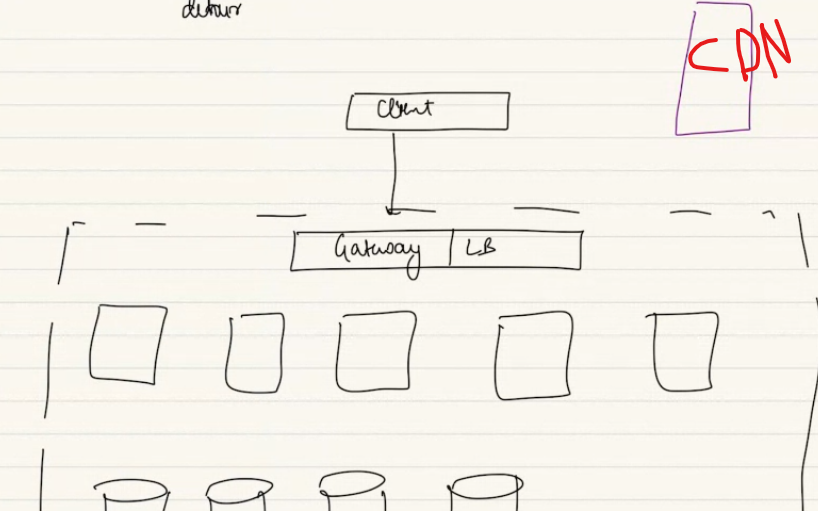
**why clients would need metadata ?**

to get chunks in order. as a client first thing I do ask the server can you give me metadata of the chunk. Once I get metadata of all chunk then as a client I request video for 1st, 2nd and 5-6 chunk. We ask for diff diff chunk. First call was to get all metadata. Metadata is very important. We ask chunk metadata.

So that’s why metadata stores in DB but given this is live streaming, the MD I create is gonan be required by million is user in next sec, will populate simultaneous to global cache. So end user when they ask for metadata can be serverd from cache not need to go to DB.

**I would want.. to store in global cache. Not in CDN?**

Cdn are there to sue static data, like video, photo. I wanan cache the chunk in CDN but not the metadata. As metadata is small. Rather returning cdn url I can send



Cdn is not backend.

In Netflix user req metadata then demand video content. If every user come to me for the movie. I will be overwhelmed to transfer throuout the world that too GB of data. Server at one plavce data somewhere. As a server I will piggyback on CDN. Here I store the big video file. I will give CDN link. So big gb of static data they can get from there..

I don’t need to transfer big file from server to user. They goto nearby CDN. Big static files used by many users will fetch from CDN.

Metadata is small textual data they come to me.. I will server directly. If I give cdn link for small data. So bad design. Metadata will be cached in global cache.

**is the entire movie in the cdn? 0.49 min**

S3 link in metadata. CDN link…

We will use CDN for big static file. Not for metadata..

CDN is for static data, how will we store live stream… how can live stream called static data. Static data is what next gonan change.. 1 wicket falls on 2nd ball, video recored. Remains same. So in live stream each chunks are static. Once created not gonna change.

Non static: whats changes, keep changes. A class has 100 people. 10 min later count of user changes. Keep changing..

Facebook post.. I will edit. More comment, likes. Keep changes the data.

But a photo which I upload.. I wont change the frame of video. Its static.

If u wanna server big file that rarely change, CDN is the way..

cdn link will vary for different region..

I am storing chunk in s3, metadata is stored in cache. And in DB. Cunks are small 15-30sec big.

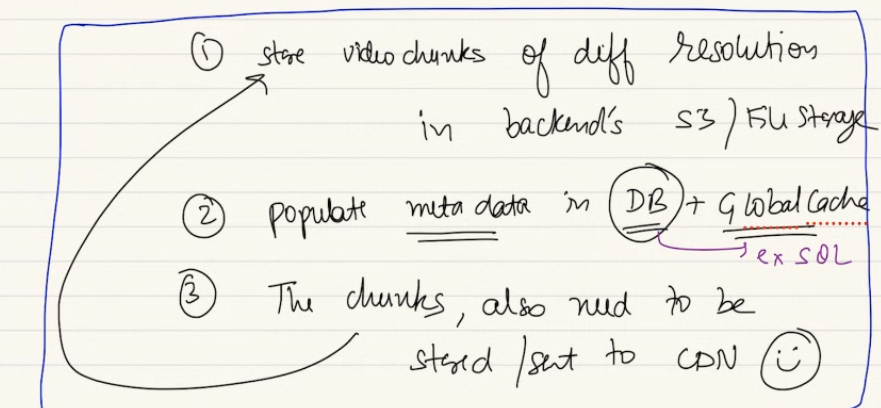
**Write through cache update… read it,.. u skipped it. 55.20**

My metadata remains in backend(hdfs/ s3). But video chunk not only stay in backend. But I have to populate to cdn.

1. Store video chunks of different resolutions in backends S3/ file storeage
2. Populate metadata in DB. And global cache.
3. Different pipeline component, diff consumer group. These chunks also stored in CDN. The chunks also need to be stored or sent to CDN.

Metadata DB and global cache. Best place is usually a SQL db. Hotstar use acamai CDN.  
Are cdn urls regular public URL or do they ensure some authentication ?  
If they required authentication who will take care of it

for example : I retried one cdn chunk from hotstar I can play it from hotstar but not from regular browser.  
link: <https://hses6-vod-cf.cdn.hotstar.com/videos/hotstar/echotrailer/1260157856/1260158297/v3/1701840995870/c811af9e895e40605b055c0c2c4227f1/audio_aac_1701840995870/hi/mp4a/3/seg-14.m4s>



CDN is help to scale the streaming. AWS Cloudfront or cloudflare.. cloudflare…Company changes deal with diff CDN provider..

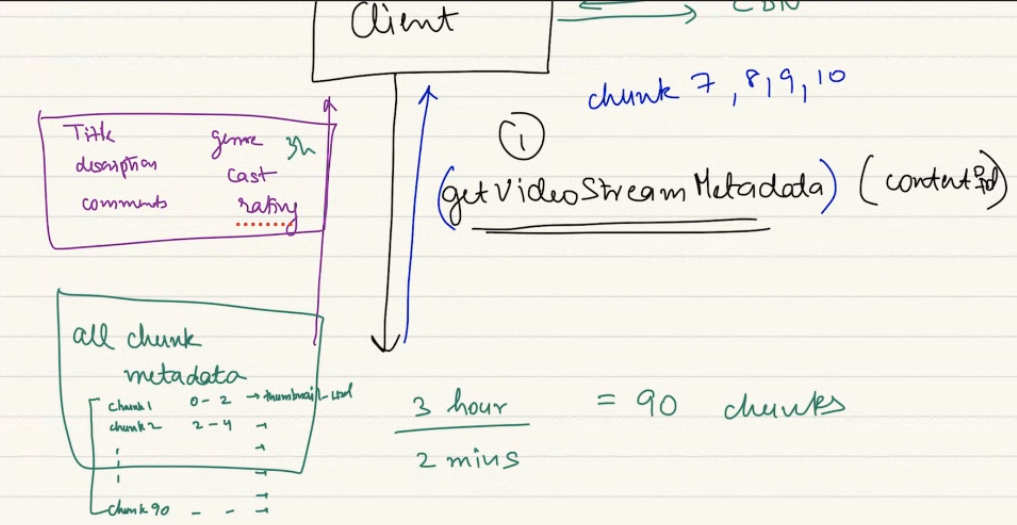
**What is gonna happen in streaming side?**

Discoverability remains same.. need not change.

**Streaming for a live feed.**

Millions of other clients they will be with me streaming at same time. Client is make a request. Request is gonna goto gateway. Then the gateway is gonna have a backend app server. A DB which contains metadata.. another db which is file store. Where chunks are stored. A global cache.. CDN where we sent our video chunks..

A client sends a request to backend. For recorded content the first request I sent



Server gives metadata of all chunks in the video.. a movie has 3 min chunks 120minutes all 40 chunks ka metadata I get in beginning..

This cant happen in a live match. We cant get all chunk metadata in one call. As I don’t have the next chunks pre created…

So it means only a single getMetadata call in beginning is not gonna enough. As a client I keep sending get metadata request again and again.

In movie I send getMetadata request once.

When a client send a getMetadata request. Server should do the latest metadata of latest chunk..

Answer is NO…

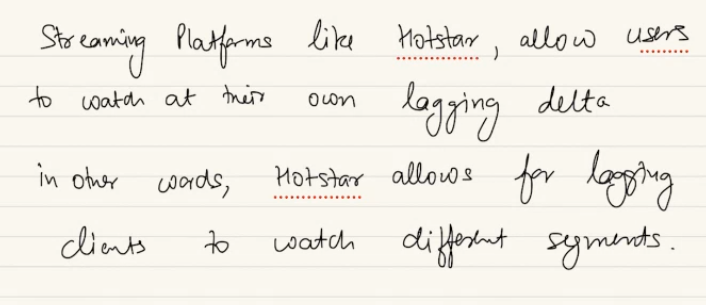
In last class.. when in office 100 people same office watching at same time. Not all watch at same frame. Some client lagg by 4 min, 3 min.. so every client goes by different pace. If my network was weak in between and took slightly longer to watch and it froze.. hotstar wont bring to latest ball… allows all the balls to watch. Server doesn’t bring me to lastet ball. Allow to watch at that ball.

We can behind 2 minuites. Yes I can click on go live and decide to skip earlier chunk.. still ic an start laging in next second..

But in live scaler class. We will forcefyly bring the live chunk…

So does hotstar gets some benefit by doing this.I think I has to transfer same chunck now to less no of users

**Very critical..**

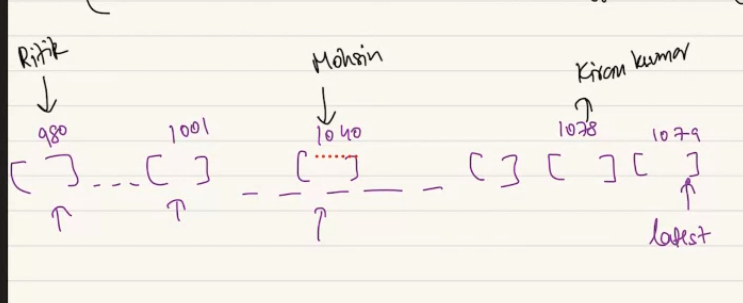


Even in live match if I am funky user I might want to seek back and watch a beautiful shot.. or every user can be at their own lagging delta. Some can be at 3 min.. hotsrta allow 5-10-15 minutes behind.. but not 1 hour back..

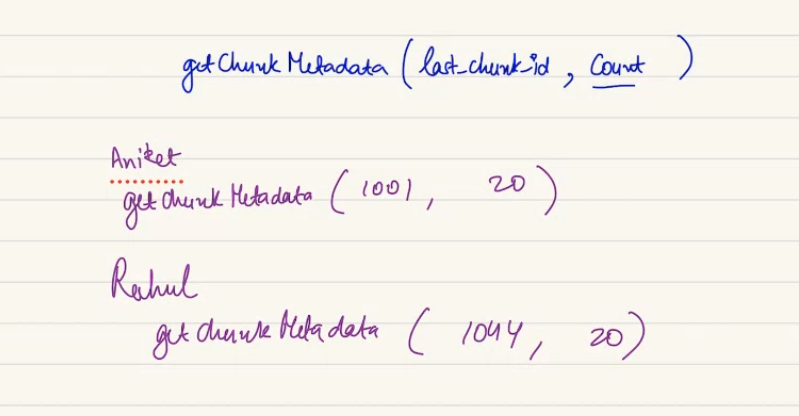
This feature has its own trade off.. some adv and disadv..

At every metadata request I cant send everyone the latest chunk on every call.. I have to send diff metadata ..diff chunk ka metadata.

Hotstar has latest chunk is 1079….A user watching at chunk id 980.. 1001.. diff chunk in stream will have diff ID. A user who is right now consuming 1040 chunk some at 980. Some at 1078.

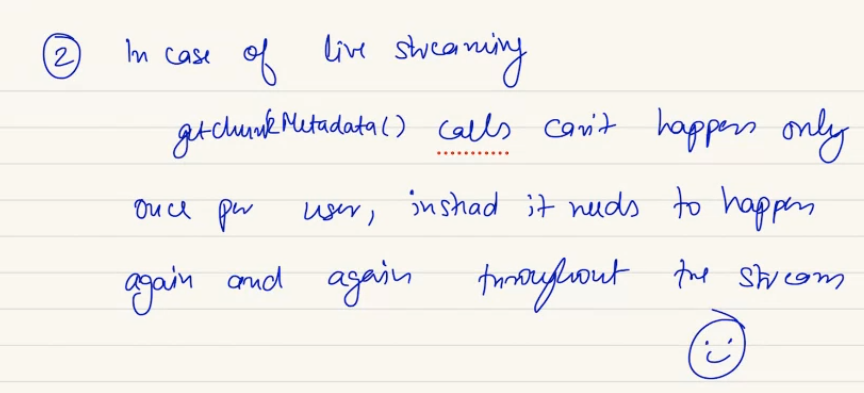


Diff people can be at diff chunk.. when every person send a getmetadata request. I cant return same answer. Everyone will need diff chunks based on where they are. The request that gonna send to backend that will be getChunkMetadata.. this needs to sent the last chunk ID. That they have consumed. Alongside they will send a count.. of chunk they want.



This can happen we don’t have 20 more chunk after this.. but will send whatever we have..

Second big different in case of live streaming: getChunkMetadata call cannt happen only once per user, instead it neds to happen again and again throughout the stream..



For live class we get chunk info of latest chunk. Will discuss later..

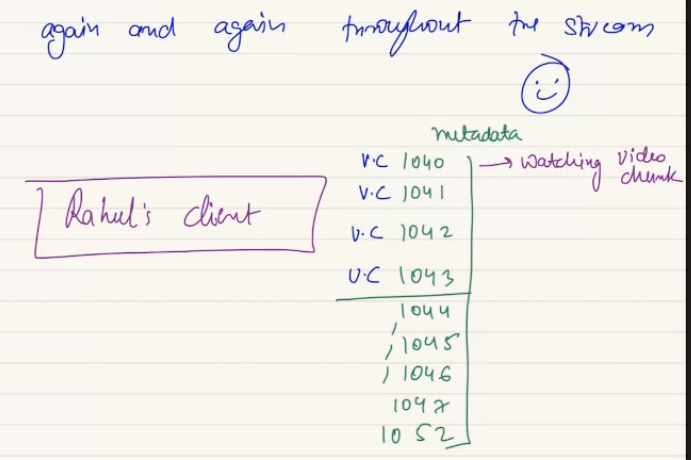
**Why client need to pull… why server cant push??? Send me the chunks I will wait..**

server don't know which chunk to sent to whom .. user won't be online evertime

we have million os user watching at same time. We cant expect hotstar to push. Hotstar cant effort to have a websocket maintained for this many users.. no.

once you have the metadata.. lest say I am talking about Rahul’s client. Watching onlaptop. They have metadata for chun id 1040, 1041,… 1052.. right now he is watching 1040 chunk..

they have downloaded the video chunk for this chunk id..till 1043. Dnt have the chunk for remaining ones.



In client side cache he has metadata.

Two responsibility.. 1. Get metadata beyond this point 1052 till next 20 chunks… preemptively make call to get further chunks metadata.

2. Once you have the metadata with you,, preemptively try to get video chunk for remaining chunks.. I will do. As whatever my user wqill watch in future I will have to preimpty request. Also I have to mind the cache size..

This is required so client can watch withuot any lag… so watch smoothly.

Normal http call. Not websocket.

All these goes to GWY…

Will I give video chunk from filestorage all across world to suer? No. for request I will go to global cache get CDN link. Return CDN link. Based on that return the chunk link based on resolutuion. In BIG tv 4k CDN link if bandwidth support it.. if bandwisth at house reduc badly I wil send 240P. I will give video link of that resolution.

Cdn link wil be in DB and global cache. So client can make request in nearby CDN and get the chunk from CDN.

**i think Anjali is asking that cnd link is not present in the metadata at client?**

You can send the cdn link for some of the package which I will server in future. Tahts possible. But it has own downsize. So we mix and match.

I watch in 4k. why would I request again and again for 4k cdn link. As aprt of response of metadata query I can do. There are some bad effect …

If I ask for all metadata 2 min earlier. I cant say after 2 min users bandwidth stays same. If I give cdn for high. But at that time it might not be the best resolution for me. Its not about client resolution/ bandith. CDN also has traffic load at diff time. In real time I can have next bes CDN which is not the same CDN machine.

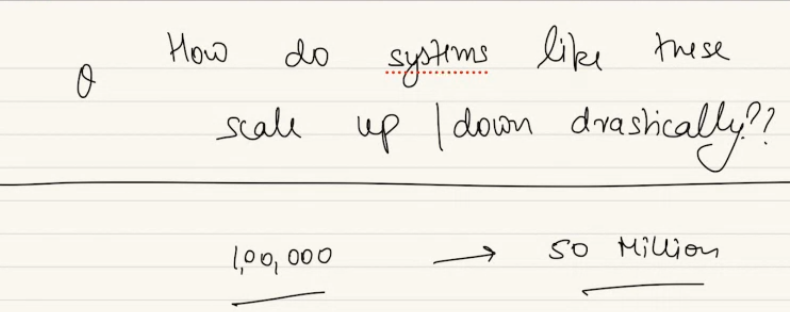
As a user I can get the data from ANY cdn. If a CDN is overwhelmed. But even I am in Mumbai. But hotstar can give the ppune cdn which is less budy… so depend on suer bandwith abd CDN state. So as a part of metadata will not give all CDN links rather wait and give right CDN.

In reality I can use both approach and keep toggling at different points of time. We can toggle based on….

When scale is high.. the badnwith keeps changing.. no of user plusgges in in live stream keeps change when dhoni, virat comes to bat.. lot people joins in.. when someone gets out al ot of people might drop.. users plussed in wil be different. Network will have different dynamic.. The staic video content stores in CDN.

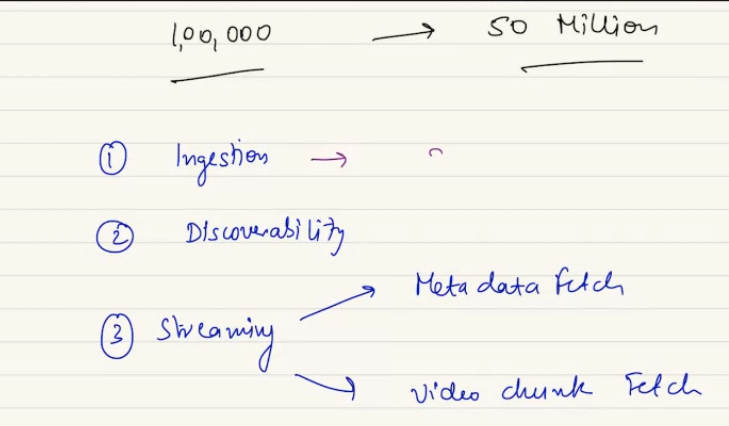
When user move up and down no of users. How do system like these scale up and down Drastically??

1000 people connect and 50 million. People are connected might vary. How can hotstar like system support this..



Autoscaling is the crux of it.

There is a ingestion , discoverability , streaming is part of platform. In streaming we havemetadata fetch and video chunk fetch.. these are diff component in hotstar like system..

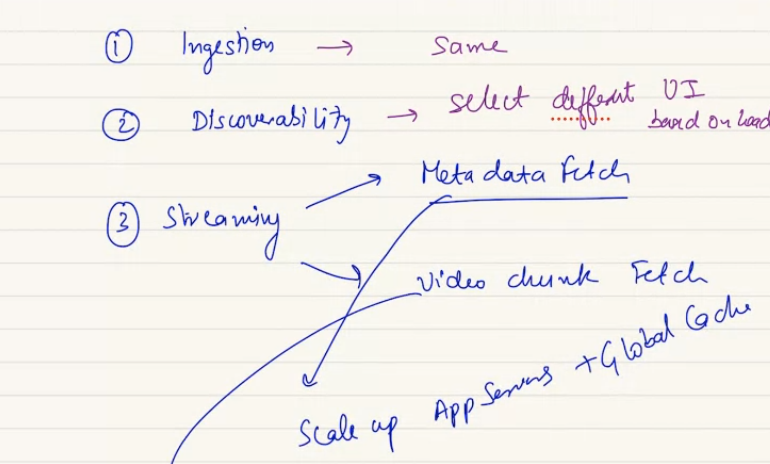


Discoverability? Will it differ.. when hotstar is not under load.. we can see evryting,s croll up and down. When live match going. They will change less bulkyer version of landing page.. big part of screen with match..

Based on load.. gives lighter UI. Less bulky.

Less option. Based on platform they will selelct a different UI. Based on load..

STREAMING: if they have to supporta lot of user, for metadata they will have to scale up their app servers, and their Cache.



For video.. chunk fetch of course scale up cachem appserver, CDN inframmm.

**so for toggling between different resolutions, the client request all the way goes to the app server and come backs to the CDN for different resolutions?**

HOTSTAR VS LIVE CLASS:

Hotstar is lot like process and convert, allow diff client to come to me and give diff metadata an dgive diff video chunk. But inlive class we cnat have 30 million people. So less people. In live class its also a 2 way communication. More of class of lesser no of audience. Its like web socket / http ling pool. Where we can see right now not see whats happening earlier.. I cant see earlier. HTTP longpool/ websocket we can see whatever goes in light now.. a stayic connection. Internet might be bad but when fine we can see whatsever going on right now.

2 way : we reply, speak up. And listen.. more like group calling….

Video call a bit diff: depend on bandwidth, I will give diff quality of video. But goes in right now.. we see latest of whats he is saying not in different point.

But recording watching is like hotstar.

Codec is there so bandiwtidh is lesser..

In live class there wil be some delay. A bit of codec.. to wtch latets frame but not we ask earlier chunk..

For hotstar its never gonan be live. A minute.. that’s why we go to crickbuzz..

In live call like this a delay of few seconds.. this scaler infra less prone to delay. If we as client have bad internet.. then get a lesser quality video, audio.. etc…

But sometime we don’t see face of instrautcor to save bandwidth… to ensure all student at same time and not lagg behind. So latency is lesser..