1st Week • (001-364)

JANUARY

TH FR SA SU MO TU WE TH FR SA SU MO TU WE TH FR
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15

TIMING

01 January • Thursday

# Netflix Architecture :-

1) OC → open ~~client~~ connect
2) Backend
3) Client

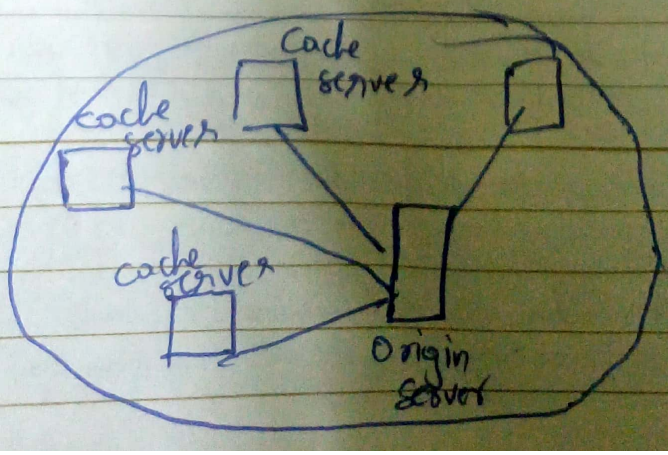
Client :- Any device from which we play the video
Ex:- Desktop, android, iphone etc

Open Connect :- (Netflix own CDN ~~CDNs~~)

Anything that is not involved in video streaming is handled by AWS

Anything that involved in video streaming is involved in Open Connect

Open Connect is Netfix own ~~CDNs~~ CDN

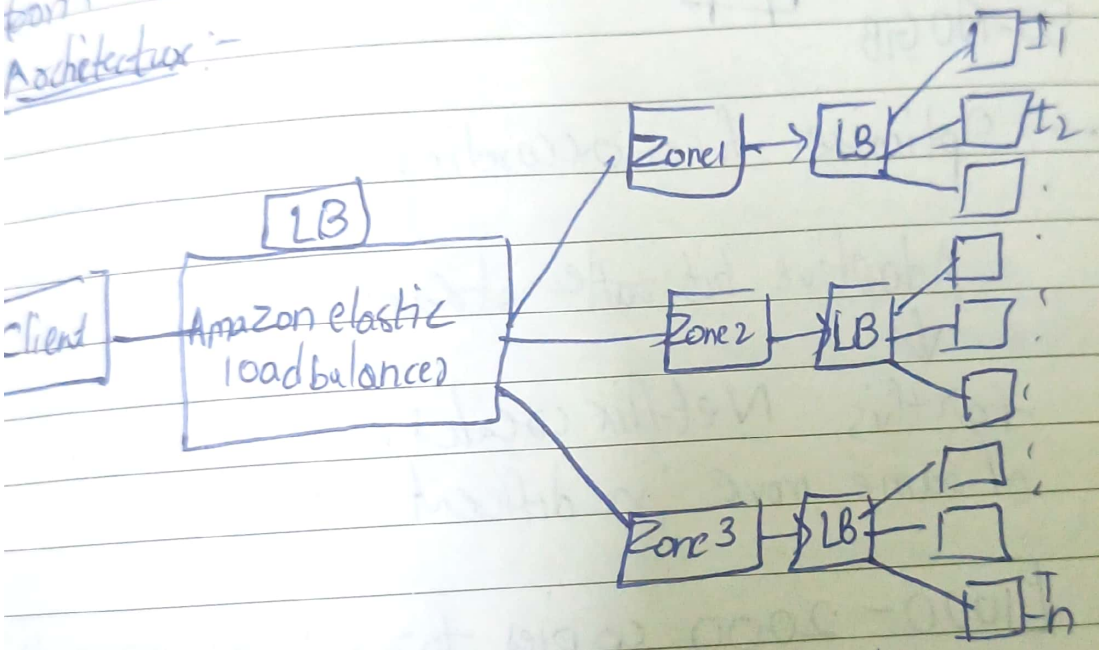


2026

Friday • January 02

If user for fam origin server requests for a video fam fam. to send that video, there will be probkms with latency. So we maintai cache servers and when video is requested we will send the data fam nearest cache server.

Architecture :-




It uses Round Robin fomm of load balancing distributing the load on the instances.

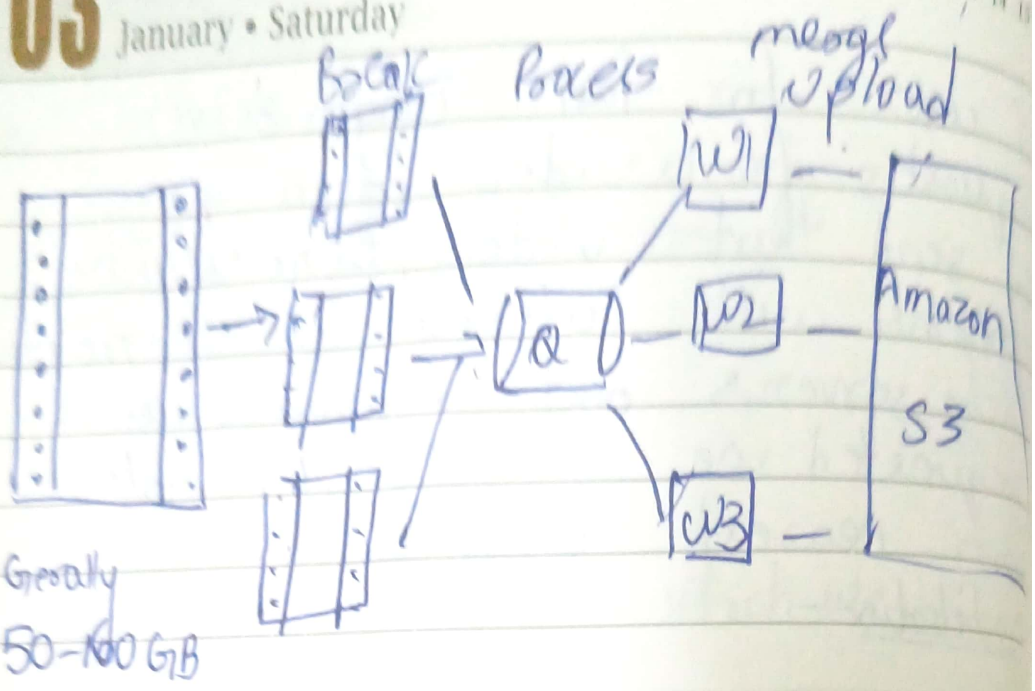
How Netflix add video to the app?

Before uploading the video, netflix does a lot of pre processing

finding errors, converting to diff reso-lutions

[Trans coding]

# 03 January • Saturday




Greatly
50-100 GB

→ optimize file according to the network

Adaptive bit rate streaming

↓

for this Netflix creates multiple copies of same movie in different resolutions

[1000-2000 copies for a single movie !!]

Amazon S3 ⇒ Amazon S3 is a

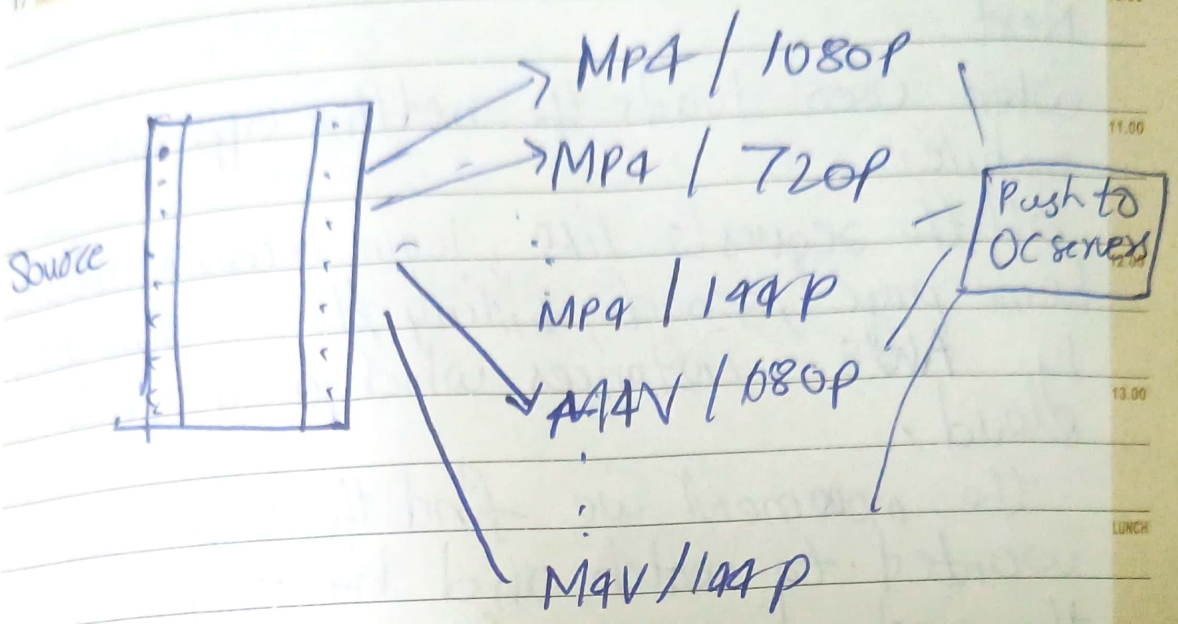
Aws cloud object storage service offering massive scalability, durability and security for storing and retrieving any amount of data, from anywhere

2026

SA SU MO TU WE TH FR SA SU MO TU WE TH FR SA SU
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

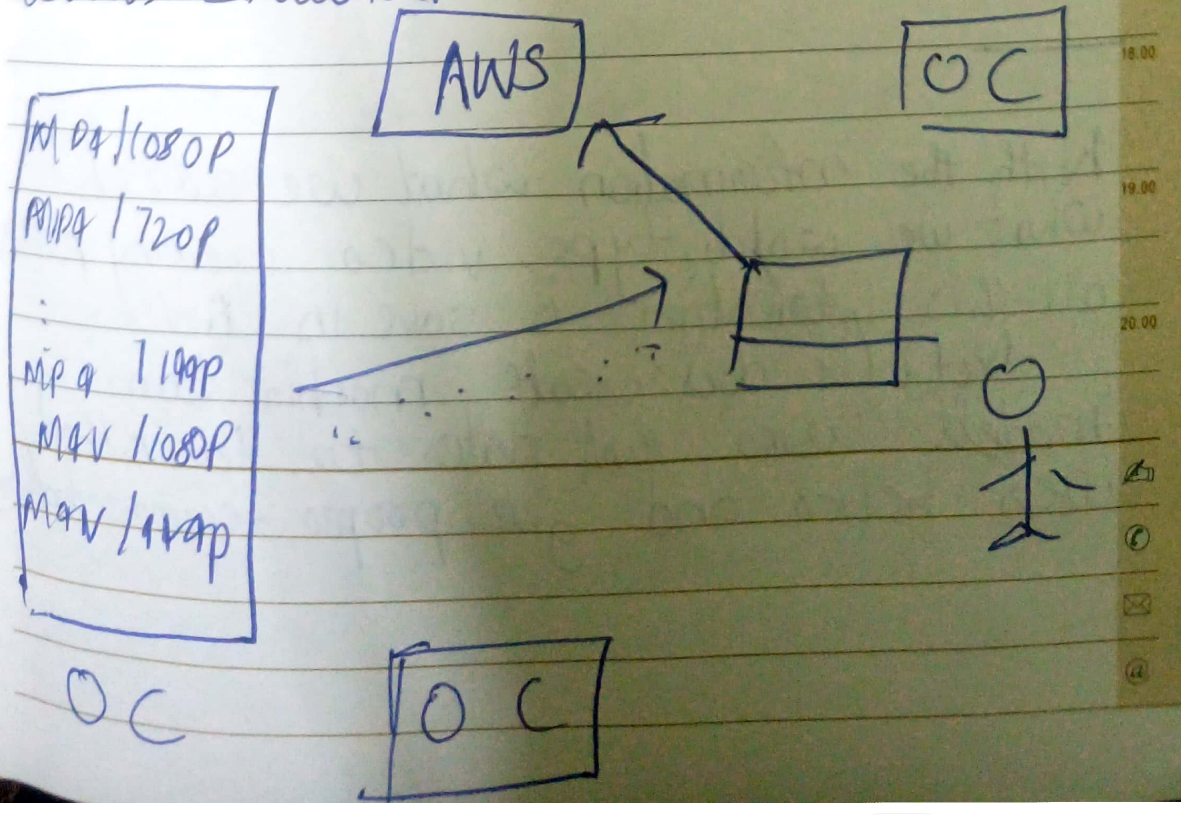
Sunday • January 04

TIMING




Amazon S3 converted main source file to different copies of movie into different resolutions and formats

All copies are pushed into open connect servers situated at different places



Next
when user loads the netflix app on mobile,
all the requests like login, recommendations, home page, search, billing etc is handled by AWS instances which are in AWS cloud,
the moment we find the video we wanted to watch and hit the play button, the application will figure out which open connect server is avilable and connects us to that server playing video with our desired resolution

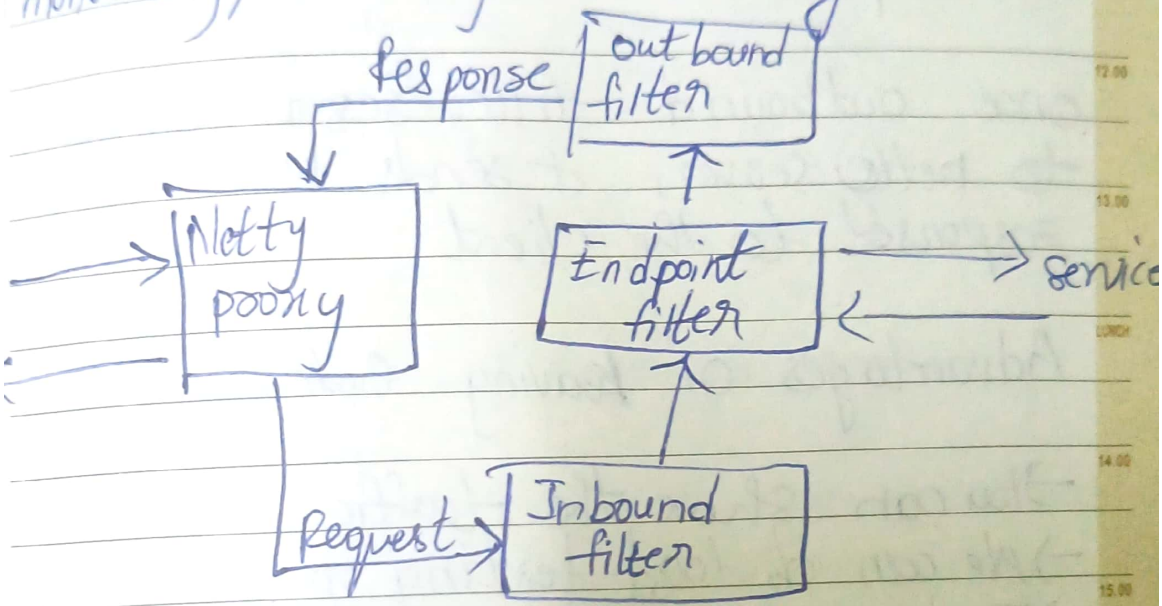
Open connect servers are dynamic and can shift between one another dynamically to give user best watching experince

---

With the information what we search, what we watch, type, video viewing patter all this information is save in AWS.
Netflix does create machine learning models using that data to learn the user better and give pooper recommendatio

# ZUUL

Gateway service that provide dynamic routing monetoring, resiliency and security.




Request first hit the Netty proxy and it will proxy the request

the request moves to inbound filter run before proxying the request and can be used for authentication, routing or decorating the requests

Request next goes to endpoint filter it can be used to return the static response or to forward request to backend services

Outbound filter gets response from Backend services which is passed from endpoint filter.

outbound filters can be used for collecting metrics, add or remove the headers from the response

once outbound filter sends response to netflix server, it sends back the response to the client.

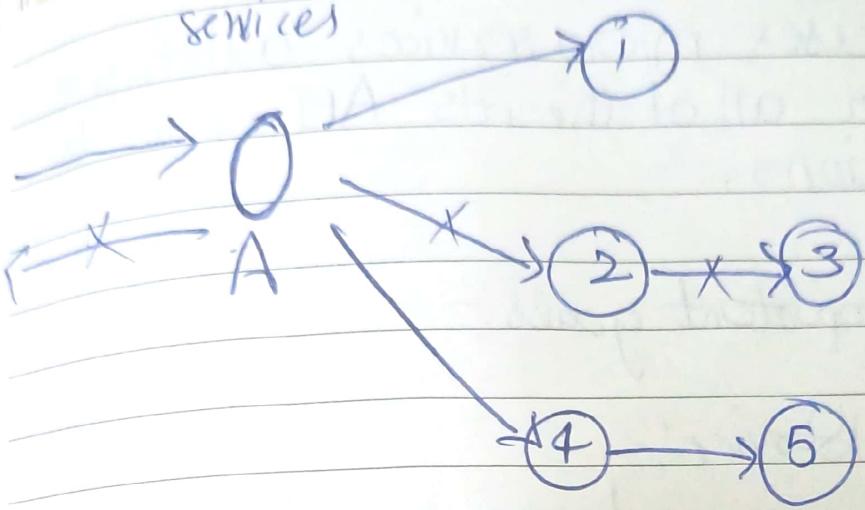
Advantages of having Gateway services

→ You can shape the traffic
→ We can do load testing on it
→ Test new services [Beta testing]
→ filter bad requests

## Hystrix

Hystrix is latency and fault tolerance library design to isolate the points of access to remote systems, services and thirdparty libraries

Microservices




service A, might be requesting many microservices,

these micro services can be internal or 3rd party service request.

But even if any one microservices goes down, that can cause problem and latency issue to main Client. So to solve this we use "histrix"

histrix helps to stop cascading failures

Applications of "Histrix":
→ Timing out call > time
[cancel if time out]
→ Reject request when thread pool is full
→ Disconnect the service when errors
→ fall back to default response
→ metrices,

Netflix uses micro services architecture to power all of the it's API needs to applications.

Two important goals:-

1) Critical Service:-
During an outage we should atleast make basic things on website [search, home-page, login etc) to work without failing
We need to seperate them as critical endpoints and make them availab at anytime

2) Make it stateless:-

If for an example our ~~server~~ api is talking to a server and for some reason that endpoint is throwing back some error we should be able to switch user to different server and get response. So state should not be preserved in server, cache.

## Caching :-

caching in the cloud involves using temporary, high-speed storage (like in-memory or edge servers) to hold copies of frequently accessed data.

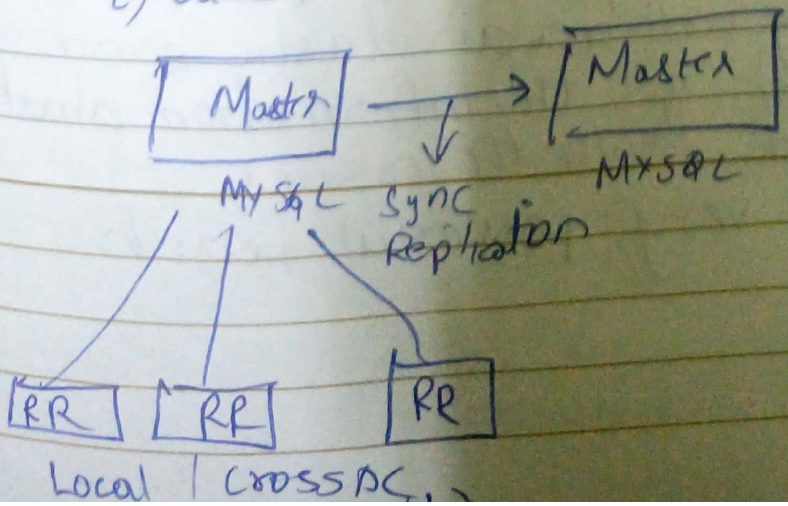
This speeds up the applications by reducing the latency and load on primary databases and helps in CDN.

## Advantages of caching :-

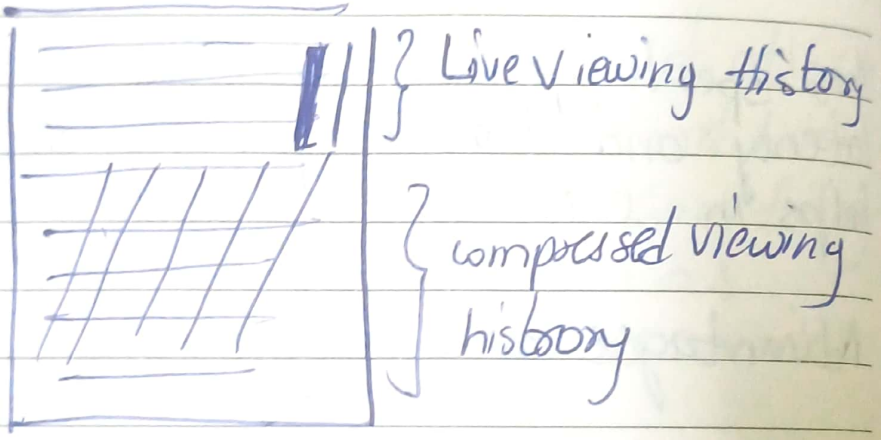
- ↳ better throughput
- ↳ Reduced latency
- ↳ Save some cost

## Databases

1) MySQL
2) Cassandra [RDBMS]



Database issue by Netflix :-
Netflix has issue with more data to store and the data is so huge to store in Cassandra database. So they optimize the cassandra database completly by following so simpler architecture




## Elastic Search

→ 130 clusters
→ 3300 instances

When a user face issue in playing the video and complaint to Netflix customer care. They will do an elastic search on the user and get the information about user and solve the issues.
This is very powerful search.

2026

3rd Week • (012-353)

Monday • January 12

# Appache Spark & Machine Learning

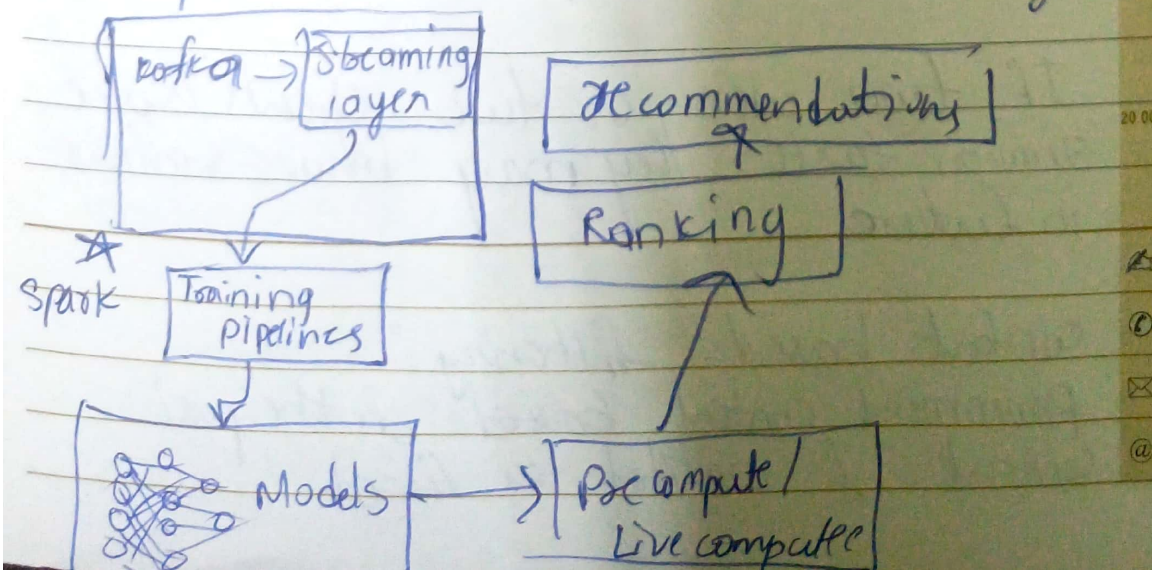
Used for
- ↳ sorting
- ↳ row selection
- ↳ Relevance Rank

} All recommendations

## How Netflix personalize content?

When a user enter Netflix,
Netflix has to show the content and they have proper header image (proper tumbnail)
[They don't know exactly what tumbnail we may like, so they use different Artwork personalizations]
Now first they show random any of already made 9-10 tumbnails and analyze our patterns and show personalized tumbnail

other important is : Movie Recommendation system



Type of data:-

→ Interaction of user data
→ other members taste
→ Metadata of watched movies or shows
  → title
  → Actor
  → Director
  → Genre
→ Device
→ time of the day.

Netflix recommendations work on:

1) collaborative filtering
2) Content based filtering

collaborative filtering:-

If two users or two clients have similar tastes, they may behave similar in future

content based filtering:-
Recommend content based on the past content I watched or liked

# Open Connect

** Only for Netflix

How Netflix cache that large data?

Netflix maintain 2 OC

Small Open Connect

[This has very less data cached for specific region, that is famous only there]

Big Open Connect

[cached with whole Netflix data]

How do they know which content to cache?

↳ Historical Viewing

↳ popular content to that region