Survey on Multimedia Systems in Social Media

Shubham Agiwal, Ekampreet Singh, Chaitanya Maddala, Aroushi Sharma, Sandeep Basvaraju

Department Computer and Information Science Engineering

University of Florida

Abstract: Multimedia is growing exponentially over the internet and it has attracted a lot of users to share their audio visual content. An uncompressed audio visual content consumes a lot of bandwidth in the network. This degrades the user experience. To overcome this issue, social networking websites such as Facebook, Netflix, Hulu and Instagram have designed their own architectures. This survey mainly focuses on architectures used by Facebook, Netflix, Instagram and Hulu. It also suggests usage of helper proxies to help Hulu and Netflix in their quest to perform live streaming. Most of the mentioned websites above use recommendation algorithms to serve the right content to their users and this survey covers the recommendation algorithms used by Hulu, Netflix and Spotify. Although there are several papers and blogs related to this topic, there is no single survey which collaborates the architectures and recommendation algorithms used by the above mentioned social media websites.

I. Introduction

As the internet is continuously growing, it had attracted a lot of attention towards multimedia streaming on web and videos in the recent years. There is a lot of research being conducted on the above which has attracted both the academia and industries. In our project, we focus on the multimedia aspects of the Social Networking sites. Multimedia adds value to Social Networking by creating an emotional connection between the content and the person receiving the content. It increases the range and scope of visibility of information. It also cultivates relationships between the mode of delivering information and the person receiving information. Social Media has brought the people together from various part of the world. The rise of social network sites geared toward specific content types – video, audio, text – has enabled a diverse ecosystem of user-created content. As individuals and organizations more readily engage with this ecosystem, social networks themselves will evolve their capabilities to support all of these numerous types of media.

Not only social media, development and advancement in various fields of communication has made a lot of progress. This progress has made the use of audio visual systems effectively in various fields. For example:-

- Business,
- Medical and Health care,
- Education,
- Industries,
- Entertainment,
- Hospitality,
- And finally as Media applications.

The use of such an audio-visual systems has various benefits. It is able to make much more sense of information, by organizing it and enabling a collective view of the same. The progress of technology is taking place at a global scale, and translation requires audio-visual systems, hence making its need more prominent. With increasing use of multimedia in various domains, social media has been constantly on the push, to promote multimedia as a source of communication. Besides using images for various purposes, Facebook now supports video sharing and live video streaming. Netflix is based on video-on-demand, and continuous research is going on to support live streaming. Instagram, recently acquired by Facebook, supports both image and video based multimedia content. Snapchat, one of the most popular social media applications, support image, video based content and live-streaming of videos as well.

Although, social networking sites are constantly promoting multimedia, in the form of audio and visual content, however this move is difficult. First issue faced is the enormous amounts of data in the form of audio and video. An uncompressed 10 second audio clip, with a low sample rate, has the size range, 0.11MB to 1.76MB. An uncompressed video of the same duration has a size range of 5.76MB to 276.5MB[1]. However, in the same size one could store a lot of text data. Due to such an enormous amount of data, companies often face the issue of, how manage and store such huge amount of data. Also, how to distribute this data to various users, since there could be packet loss or packet out of order delivery, hence there should be an architecture managing this as well. Also, another issue most social networking sites face now days is providing the useful content out of all this data, i.e. having recommendation system assistance while providing data to the user.

Through this paper we intend to survey various techniques employed by companies now days to deal with multimedia in their organizations. Section II describes various techniques being carried out in the past to handle the various issues related to audio, image and video based data. Section III, explains various techniques and technologies being employed by famous social networking sites such as Hulu, Netflix etc to handle such huge amounts of data (audiovisual). Section IV gives a brief improvement strategy that Netflix and Hulu can incorporate to provide live streaming in the near future.

Section V talks about the implementation details of recommendation systems for Hulu, Spotify and Netflix. The conclusion and the references for this survey is given in Section VI and Section VII respectively.

II. Related Work

With audio and visual media gaining more and more importance, there is an urgent need for developing architecture which could support such a data. As pointed earlier size is one of the biggest problems faced while handling such data. In order to solve this problem, compression was considered as an alternative. JPEG compression and MPEG compression are one of the most famous schemes being used to compress audio, images and videos.

A. JPEG Compression

Joint Photographic Experts Group(JPEG)[2] is an image encoding algorithm designed to compress photographs and similar images effectively, often 5 to 15 times over a raw bitmap format. It's a lossy compression scheme that exploits properties of human vision to eliminate information that is difficult to distinguish. This encoding combines three different completely independent methods of image compression: down sampling the chrominance channels, quantization of the discrete cosine transformation, and entropy coding. The figure below explains the procedure for JPEG compression.

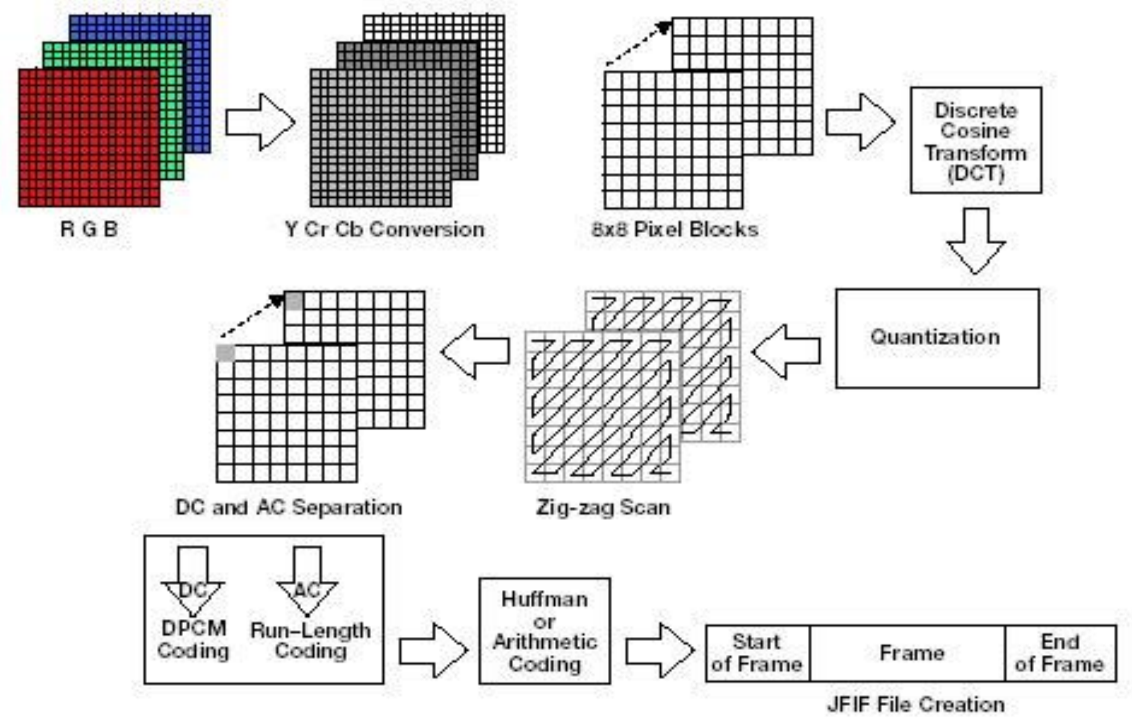


Fig 1: JPEG Compression Scheme

B. MPEG Compression

The **Moving Picture Experts Group** (**MPEG**)[3] is a working group of authorities that was formed by ISO and IEC to set standards for audio and video compression and transmission. It was established in 1988 by the initiative of Hiroshi Yasuda (Nippon Telegraph and Telephone) and Leonardo Chiariglione, group Chair since its inception.

MPEG compression basically employees temporal differences together with spatial differences to encode data. As shown in figure, video stream is broken down to I, B and P frame, where B and P frame capture the motion vector, i.e. temporal differences. Every I frame is encoded using JPEG scheme[2] and is sent to the receiver along with motion vectors. The encoding and display order of this scheme is different.

The above scheme describe basic model of MPEG compression, other MPEG schemes, like MPEG2[4](Packetized Elementary Stream), MPEG4[5](Content based compression) and MPEG7[6](Description Definition Language) use this and other sophisticated techniques to perform more efficient compression.

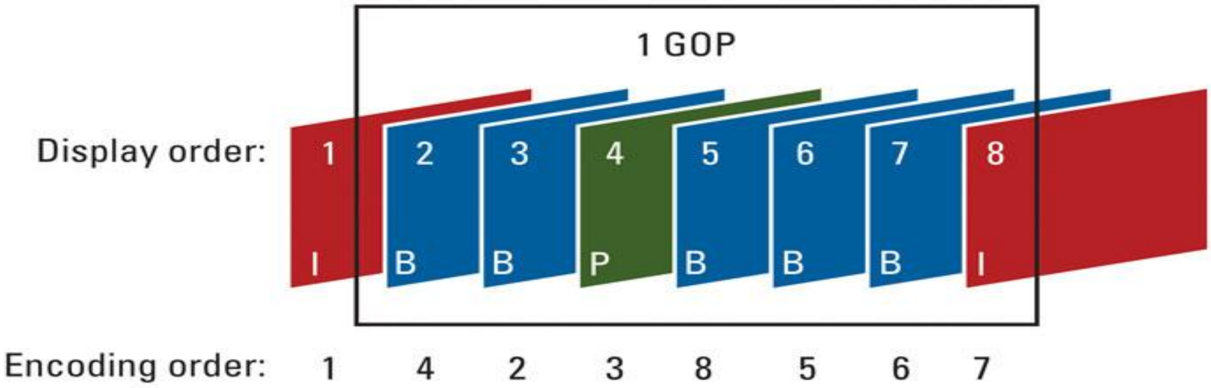


Fig 2: Encoding and Display order of frames in MPEG

Use of data compression schemes make it possible for large videos to be transmitted across a network. Once the video has been transferred, next problem arises in streaming it. Video streaming has been an extensive research problem for which research is still going on. We tackle some of its details below.

C. Real Time Multimedia

Real Time Multimedia implies that there is a timing constraint. For example, a video contains both video and audio data. If both the data do not arrive on time, then the playout process has to pause and this will annoy the user. There are two modes for transmission of videos namely the download mode and the streaming mode. In the download mode the video is downloaded by the user before playing it. The video cannot be played by the user until the entire video is downloaded. On the other hand, streaming video is sending compressed content at real time to user, so the user does not have to download the file and then play it. Here the videos that are streamed are stored and forwarded to the respective clients[7].

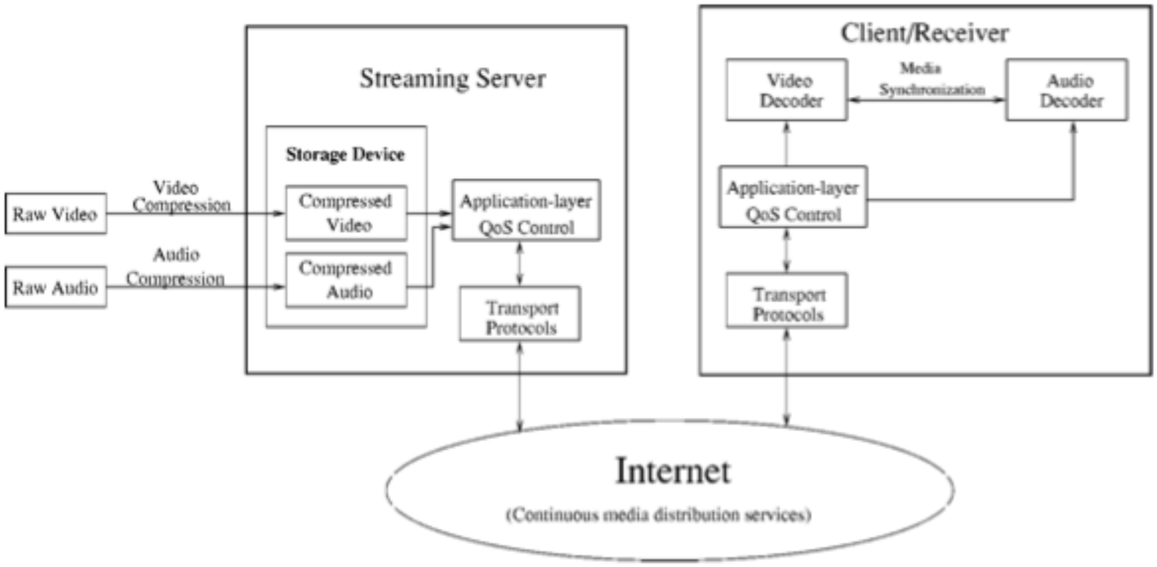


Fig 3: Architecture of Real Time Multimedia

The above figure represents the architecture for video streaming[7]. It has six components, they are as follows.

- **Video and audio compression***:* Raw video and audio cannot be transmitted across the internet as it uses a large amount of bandwidth which will congest the network. JPEG and MPEG techniques are used to compress video contents.
- **Application-layer QoS control:** Various application-layer QoS control techniques have been proposed. The application-layer techniques include congestion control and error control.
- **Continuous media distribution service:** Network filtering, application-level multicast, and content replication are included to ensure good quality media.
- **Streaming servers:** The streaming services are provided by the streaming servers which ensures that the multimedia is delivered to the recipient within a given timing constraint. These servers should support operations such as pause/resume, fast forward and fast backward.
- **Media synchronization mechanisms:** Media Synchronization must be present at the receiver end to ensure that the media streams are in the same order as that of the sender.
- **Protocols for streaming media:** The Internet today use the following protocols for streaming multimedia which are as follows network addressing(IP), transport(UDP or TCP), and session control(real-time streaming protocol (RTSP)).

The above architecture suggest the use of a central server scheme, which is vulnerable to a DOS attack. Hence, to minimize service delays[8-11] and reduce the loads placed on the streaming server, one of the feasible solutions is to use a VCN (video content delivery)[13] which uses cache space of the distributed system along the delivery path between server and users for caching popular videos.

VCN has a number of caching elements along the delivery routes between the clients and the video server. VCN follows a dynamic approach, in the sense that, it can detect changes in network topology. The approach is completely decentralized. VCN based approach could result in a cache hit or miss. In case of a miss, team formation is done, which executes cache reservation algorithm, ensuring adjacent cache elements have a portion of the video which was not present[12].

When no free space is left, the caching element will purge the least popular video to make way for the new video[12]. In case of a cache hit, CE element sends a cache hit message to the team leader, which in response transfers the video blocks to respective clients.

An advantage of this approach against the conventional proxy caching setting was that latency of the network was close to zero. However, this approach also has its cons. First, it makes an assumption on the popularity profile of the video. Second, resource reservation algorithms have extra overhead. And finally, it is dependent on the cache size and caching elements.

An important aspect to be considered is the storage scheme incorporated for streaming of real time multimedia . Video on Demand(VOD) server is the heart of the VOD systems. Disks contain compressed form of video, while RAM consists segments of videos currently being transmitted for ongoing playback[13]. The most important task of the VOD server is the efficient storage and retrieval of the video from the memory and the RAM respectively.

According to the authors of this paper[13], many models have been designed to improve the performance of the multimedia video on demand streaming. One of the methods involved increasing the buffer size(RAM), however, this brought about extra overhead as well. Hence new methods were instead proposed. The entire video is divided into chunks, now instead of the user waiting for the entire video to be transmitted, a randomized based approach, wherein few chunks of the next video stream were also encoded here. Following this, user did not have to wait for the entire buffer time, but was replaced by new video slot as soon as a random slot was done transmitting. However, this approach required extra buffer size to function.

This led to a novel storage scheme, phase based with RAM replication[13] which consists of immediate playback RAM and sustainable playback RAM. Optimal utilization of the RAM was ensured and user experience was not degraded. Using this scheme a 5%

reduction of storage was observed compared to earlier storage schemes[13]. The limitation of this scheme is that its effect was not observed for multiple videos.

The above VCN and novel storage schemes did not emphasize on larger geographical range, security of data and the application, and high traffic load management. These shortcomings led to the use of Cloud Distributed Networking(CDN)[14]. CDNs were first established in the late 1990s, and were far too expensive. However, now it is available at lower cost and forms an integral part in most of the video streaming sites such as Hulu and Netflix. The Building blocks of CDN are Point of Presence(POP) architecture, Caching server and SSD/ RAM[14]. Since CDN needs to be accessible across a large geographical area in the world we use POP architecture, reducing the round trip delay for getting content from the server. Caching Server are responsible for storage and delivery of cached files. Each CDN generally has multiple SSDs and RAMs for faster processing of frequently accessed items[14].

III. Social Media Sites

In the paper, we compare the architectures of famous companies which serve multimedia content on a global front. These are Facebook , LinkedIn, Instagram, Hulu and Netflix.

A. Facebook

History

Facebook[15], like many social networking sites was found by university students, who initially vended their product to other fellow students. It was launched in 2004 as a Harvard-only exercise and remained a campus-oriented site for two full years. Even by that time, Facebook was considered big business, which let to investments by bigwigs such as Paypal co-founder and billionaire Peter Thiel who invested tens of millions of dollars just to see it flourish.

The features that account to success of facebook are its ease of use, its multitude of easily-accessed features, and a wide variety of features and capabilities to interest the user. Recently, Facebook has implemented a highly targeted advertising model. Regardless of these implementations, facebook universal agreement is on one thing that it promotes both honesty and openness. People really enjoy being themselves, and throwing that openness out there for all to see. It connects people on day to day basis.

A series of smart moves and innovative features have set the platform apart for Facebook from the rest of the social media pack. First and foremost, the 2007 launch of the Facebook Platform was key to site's success. The open API made it possible for third-party developers to create applications that work within Facebook itself. Almost immediately after being released, the platform gained a massive amount of attention.

Farmville had gained a huge amount of popularity during the initial days of facebook. Meanwhile Twitter, created its own API and enjoyed similar success as a result of creation of an Open Platform for users to interact[16].

The much talked about feature which was a huge success was the Facebook's omnipresent 'Like' button. People can now 'like' or "tweet' just about everything even when you're on Facebook or Twitter. Analyzing the power of social networking, Google also decided to launch their own social network (Google+) in 2007[16]. It was different from Facebook and Twitter because it wasn't necessarily a full-featured networking site, but more of a social "layer" of the overall Google experience. Initially, Google generated a lot of buzz with the service's Hangouts feature, which allowed users to enter live video chats with other online friends but later the same feature was introduced by Facebook.

In a span of just four weeks, Google+ had garnered 25 million unique visitors, with as much as 540 million active monthly users as of June 2014[16]. Regardless, Facebook still stood resilient and was still much more preferred interface. It has showed the world that there is more scope for innovation and competition in the realm of social networking.

Architecture

Facebook is a wonderful example of the network effect, in which the value of a network to a user is exponentially proportional to the number of other network users.

Facebook's power derives the "social graph" which is the sum of the wildly various connections between the site's users and their friends; between people and events; between events and photos; between photos and people; and between a huge number of discrete objects linked by metadata describing them and their connections[17].

Facebook maintains data centers in Santa Clara, CA; San Francisco; and Northern Virginia. The centers are built on the backs of three tiers of x86 servers loaded up with open-source software, some that Facebook has created itself[17].

The main facility, in Santa Clara, has the top, middle and the bottom tier of Facebook from which the rest of the sibling units of Facebook draw their functionality.

- The top tier of the Facebook network is made up of the Web servers that create the Web pages that users see, most with eight cores running 64-bit Linux and Apache[17]. Many of the social network's pages and features are created using PHP[17]. But Facebook also develops complex core applications using a variety of full-featured computer languages, including C++, Java, Python, and Ruby. In order to manage the complexity of this approach, the company created Thrift, an application framework that lets programs compiled from different languages work together.

- The bottom tier consists of eight-core Linux servers running MySQL, an open-source database server application[17]. This tier stores all the metadata about every object in the database, such as a person, photo, or event.
- The middle tier consists of caching servers. Highly efficient cache servers, running Linux and the open-source Memcache software, help in providing immediate results.
- Photos, videos, and other objects that populate the Web tier are stored in separate filers within the data center.

The San Francisco facility replicates the Web and cache tiers[17], as well as the filers with the database objects, but it uses the Santa Clara MySQL database tier.

The Virginia data center completely duplicates the Santa Clara facility, using MySQL replication to keep the database tiers in sync.

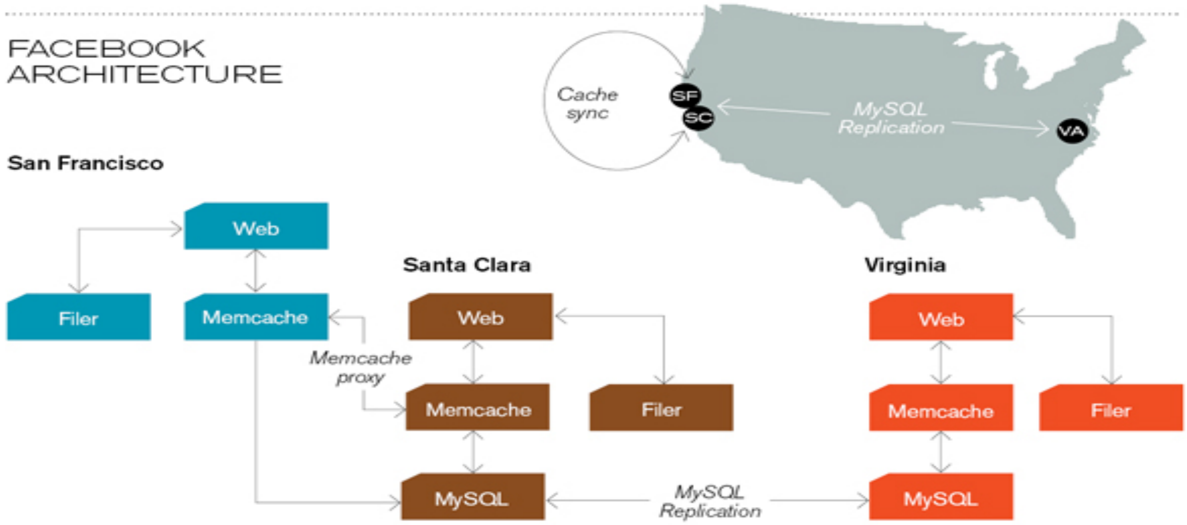


Fig 4 -Facebook Architecture

Technology Behind Preview Photos

Facebook is nothing without pictures. They form an imperative part of a person's profile, but sometimes they can be slow to download and display. This can be due to low connectivity which leads to one staring at a large number of grey boxes. To make a smooth user's experience, they decided to make changes to their existing architecture.

Facebook initially focused on the cover photos which are the top header pictures in a user's profile and pages. The cover photo is one of the most visible parts of these surfaces, yet it's also one of the slowest to load. The reasons were as follows: Firstly, the cover photos often reach 100 KB even after JPEG compression technique[18]. Secondly, before downloading a picture, the application makes network request for the picture's URL from the GraphQL server[18]. To actually get the image, an another request to CDN is done to get the image bytes. The time taken for this request is longer than the first network request.

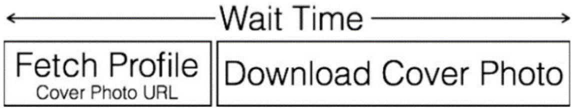


Fig 5 -Wait Time

To solve this problem, facebook decided to create an impression of the image using only 200 bytes[]. In order to remove the second network request, they needed to include a copy of the image itself in the initial network request. It was determined[18] that if they could shrink a cover photo down to 200 bytes, it could efficiently be delivered via the GraphQL response. The interesting aspect of this solution, if successful, was that it successfully addressed the above problems. Also it was estimated that this would enable facebook to display the preview photo in their first drawing pass, thereby, reducing the total time to display profiles and page headers significantly. However, they still have to download the full-size image from CDN which could be done in the background. The challenging part of this solution is to compress the cover photo to a size of 200 bytes.

To achieve this compression to display the preview image facebook used a Gaussian Blur Filter. It is a widely used effect in graphics software, to reduce image noise and details. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen. The required blur radius for the Gaussian filter was determined. From this blur radius, facebook was then able to compute the lowest-resolution image that would still give them the desired final image. The resolution of the cover photos was 42 pixels[18]. Above a 42x42-pixel image, no additional fidelity is present in the image. However assuming 3 bytes per pixel (for RGB components), we would get 42x42x3, or 5,292 bytes which is higher than 200 byte desired target.

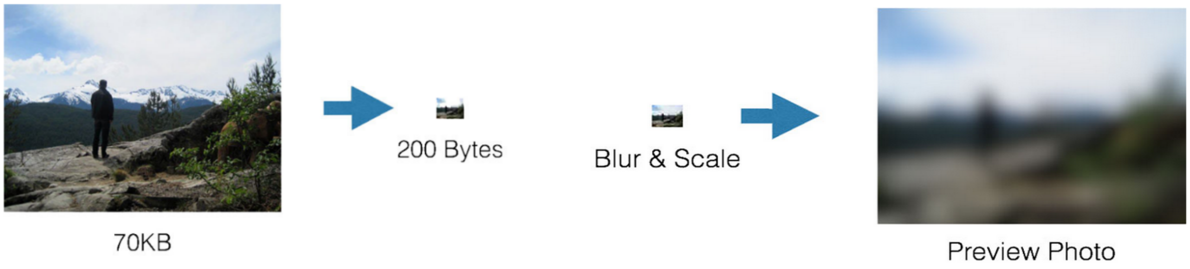


Fig 6 - Preview of Image after scaling down and applying gaussian blur filter.

JPEG compression in Facebook

There are a few tables within the JPEG header, which accounts for its size. The question then was to generate a fixed header that could be stored on client and therefore is not needed to be transmitted, only the payload would need to be sent. It is known that for a given Q value, the quantization table is fixed; through experimentation and measurement, Q=20 produced an image that would meet the visual needs[]. Images were not a fixed size but restricted at 42x42. This amounted to 2 bytes that could be prepend to the payload and that could be placed by the client in the correct spot to make the header valid. As they looked through the rest of the standard JPEG header, the only other table that could change with different images and options was the Huffman table[18].

A version number was added to the beginning of the Huffman Table so that the format would be future-proof. If there are any extreme cases or better tables in the future, they can update the version number for those images and ship new tables on the clients. Finally the format became one byte for version number, one byte each for width and height, and approximately 200 byte payload[18].

In summary, the server would just send the above format as part of the GraphQL response, and the client could simply append the JPEG body to the predefined JPEG header, patch the width and height, and treat it as a regular JPEG image. After the standard JPEG decoding, the client ran the predetermined Gaussian blur and scaled it to fit the window size. This resulted in generating an efficient algorithm to generate these preview images. Images can now be loaded on slower connections and the user experience remains swift.

B. Instagram

History

Instagram is an online mobile photo-sharing, video-sharing, and social networking service that enables its users to take pictures and videos, and share them either publicly or privately on the app, as well as through a variety of other social networking platforms, such as Facebook, Twitter, Tumblr, and Flickr. It was founded in the year 2010 by Mike Krieger and Kevin Systrom. Instagram had one million users after 2 months of the launch and it kept on growing ever since.

Currently it has 16 million photos shared, 1.2 billion likes every day, 55 million photos uploaded everyday. Instagram was acquired by Facebook in the year 2012.

Core principles followed by them when choosing a system are:

- Keep it very simple
- Don’t re-invent the wheel

- Go with proven and solid technologies when you can

Architecture

OS/Hosting

Instagram uses Ubuntu Linux 11.04 (“Natty Narwhal”) on Amazon EC2[19]. Rest of the versions had unpredictable behavior with EC2, while Natty performed well, hence making it the final choice. Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. It eliminates the need to invest in hardware up front, so one could develop and deploy applications faster. Amazon EC2 can be used to launch as many or as few virtual servers required, configure security and networking, and manage storage. Amazon EC2 enables to scale up or down to handle changes in requirements or spikes in popularity, reducing the need to forecast traffic.

Load Balancing

Load balancing is used to improve the concurrent user capacity and overall reliability of applications. It works on the principle of reverse proxy, by distributing network and application traffic across different servers. Every request to Instagram servers goes through load balancing machines; we used to run 2 nginx machines and DNS Round-Robin between them. The above figure describes the nginx architecture. It basically provides web-server for open source applications and helps in load balancing. The downside of this[20] approach is the time it takes for DNS to update in case one of the machines doesn’t work. Recently, Instagram moved to using Amazon’s Elastic Load Balancer, with 3 NGINX instances behind it that can be swapped in and out (and are automatically taken out of rotation if they fail a health check). The elastic load balancer basically works by distributing incoming traffic to healthy EC2 instances, hence reducing the stress on a single server. Amazon’s Route53 is used for DNS.

Application Servers

Next up comes the application servers that handle our requests. Instagram runs Django on Amazon High-CPU Extra-Large machines[20]. They use http://gunicorn.org/ as WSGI server. To run commands on many instances at once (like deploying code), they use Fabric, which recently added a useful parallel mode so that deploys take a matter of seconds.

Data Storage

Instagram uses PostgreSQL to store all its data. The data is stored using a technique called sharding[21], which works on the intuition of dividing data into separate fragments and storing on different platforms. Before storing however, for each user data was molded as a structure which could be useful while retrieval as well. Each user gets a separate ID, with all it data, and this ID contains a timeframe as well, which could help in sorting data. Once the data is sorted these time buckets then makeup various stores and various stores makeup a shard. A hierarchy is established, hence, occupying the minimal possible space.

Instagram makes use of an open source technology, vmtouch to manage the data content in the memory for faster access. The PostgreSQL instances run in a master-replica setup using Streaming Replication, and Elastic book store, for managing the data with EC2, snapshotting is used to take frequent backups of our systems. XFS file system is employed, which allows freezing and unfreezing the RAID arrays when snapshotting, in order to guarantee a consistent snapshot (our original inspiration came from ec2-consistent-snapshot).

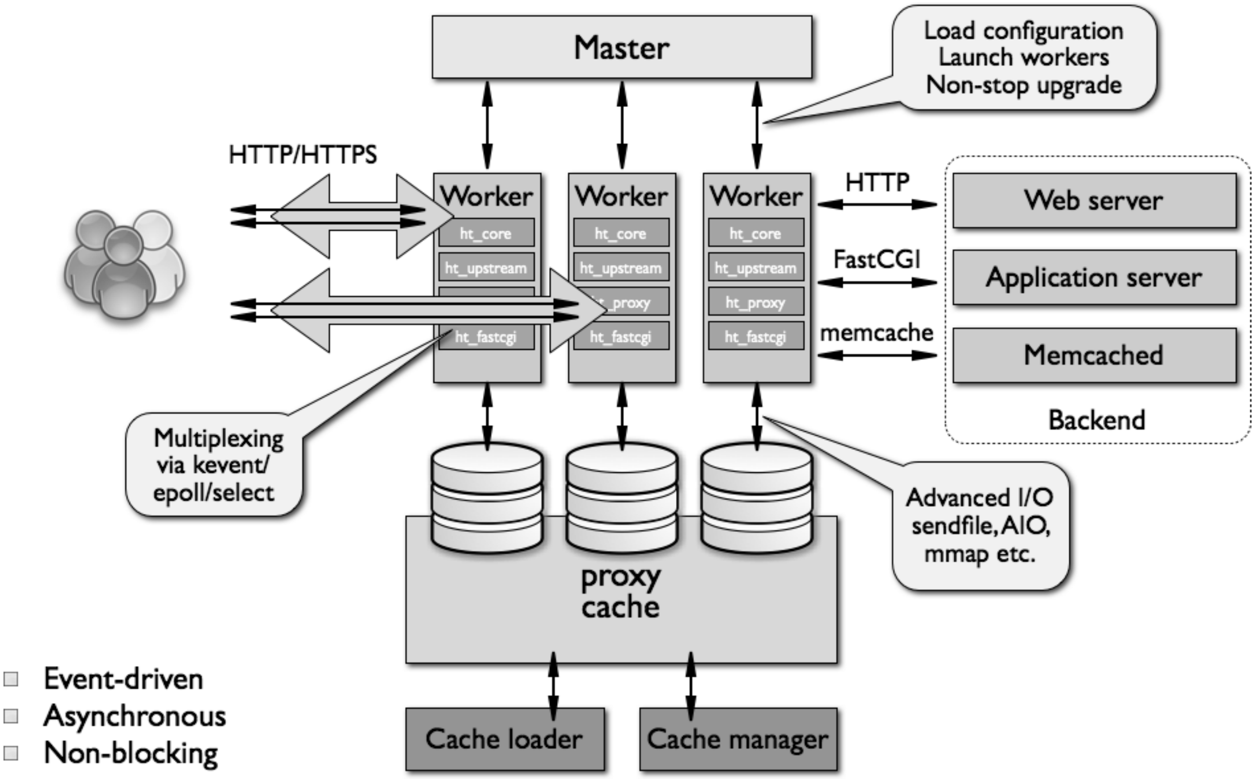


Fig 7- Instagram Architecture

In order to enhance the performance of retrieval calls, Pgbouncer was used to pool connections to the PostgreSQL databases. It is a lightweight connection pooler to PostgreSQL, and incorporates, session, transaction and statement pooling.

The photos themselves go straight to Amazon S3, which currently stores several terabytes of photo data for us. Amazon CloudFront is used as CDN, which helps with image load times from users around the world (like in Japan, our second most-popular country).

Instagram employs Redis extensively; it powers the main feed, activity feed, sessions system, and other related systems. All of Redis' data needs to fit in memory, so they end up running several Quadruple Extra-Large Memory instances for Redis, too, and occasionally shard across a few Redis instances for any given subsystem. Redis runs in a master-replica setup, and have the replicas constantly saving the DB out to disk, and finally use EBS snapshots to backup those DB dumps (we found that dumping the DB on the master was too taxing). Since Redis allows writes to its replicas, it makes for very easy online failover to a new Redis machine, without requiring any downtime.

Like any modern Web service, Instagram uses Memcached for caching[21], and currently have 6 Memcached instances, which are connected to using pylibmc & libmemcached. Amazon has an Elastic Cache service which they've launched recently, but it's not any cheaper than running their instances, so they haven't pushed themselves to switch quite yet.

Finally, as we all know Facebook took over Instagram in 2012[22], and after that move engineers at Instagram thought migrating their databases and other production was the most viable move. Hence, for this purpose, engineers at Instagram had two options, first migrating from AWS to Amazon web cloud, and then operating with Facebook. Instead, Instagram developed Neti which could sync in the Facebook infrastructure with Instagram operations hence making it much easier.

C. Hulu

History

Large scale video content delivery is becoming one of the most important contributor to internet traffic. According to [23], 82 percent of US audience views online videos and hulu is one of the largest online video service providers. In addition to this, hulu also offers subscription based service called Hulu Plus which runs on mobile phones, set top boxes etc. Initially, hulu was a personal blog site owned by Amy Hung, which he used, to upload and share family photos. Later NBC contacted him and bought the domain for an undisclosed amount. Hulu does not maintain its own content delivery infrastructure due to its high cost and vast expertise needed to maintain it. Hulu Plus is the most popular service that hulu offers to its desktop users.

Architecture

According to [24], the high level architecture of Hulu is as follows

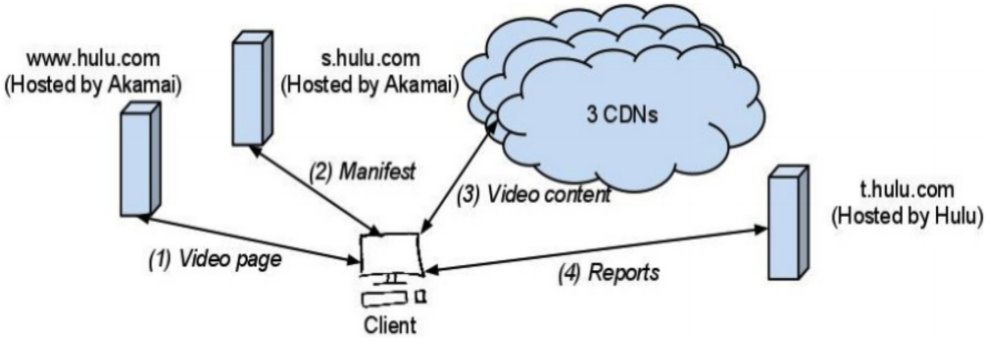


Fig 8 - Hulu Architecture

To study hulu service, [24] multiple videos were run on different web browsers in different locations with different ISP's both with and without firewalls and proxy servers on different devices such as Nexus, Iphone, Ipad etc. The devices were connected to a Wi Fi Network and tcpdump was run on a laptop. HTML and its corresponding objects were obtained from the hulu server to different clients. The clients used the instructions from the manifest files from s.hulu.com and sent a periodic status report to the t.hulu.com[24]. The videos were streamed at a 480 to 700 kbps. Also, the clients have the ability to change to multiple bit rates during the playback. Hulu uses three CDN's to deliver its video content which are Akamai, Limelight and Level3[24]. The protocol used in Hulu to stream videos is encrypted RTMP. However on mobile devices Hulu Plus uses adaptive HTTP streaming[24].

Real Time Messaging Protocol(RTMP) is a TCP based protocol which has persistent connections between the clients and the server. The major principle behind RTMP is that it divides the video streams into fragments and the size of the fragments is dynamically changed based on the client and server. The default size of the fragment for video data is 128 bytes. It also support multiplexing of different streams on a single connection.

The advantages for RTMP is as follows

- RTMP can do live streaming, so people can watch your video while it is being recorded.
- RTMP can do dynamic streaming, where the video quality automatically adjusts to changes in bandwidth.
- Players can seek to later parts in a video, which is particularly useful for files > 10 minutes.
- Players maintains a tiny buffer, instead of downloading a video during playback, saving bandwidth.

However there disadvantages also which are as follows

- RTMP uses different protocols and ports than HTTP, which makes it vulnerable to getting blocked by (corporate) firewalls. This issue can be prevented by streaming in RTMPT (tunneling over HTTP), which comes at a server performance cost.
- RTMP data is streamed to the player, which means bandwidth of the connection must be larger than data-rate of the video. If the connection drops for a couple of seconds, the stream will stutter. This issue can largely be prevented by using dynamic streams that include a low-quality file.

The effect of network condition on the CDN strategy was experimented[24] using dummynet. The servers of the CDN was initially set to send data at 1500 Kbps and was lowered every minute by 100 Kbps until it reaches 1 Kbps. Lowering the sending rate did not change the CDN selected[24]. The above observation was a conclusive proof that adjusting the bit rates did not change the CDN preference[24]. The user will remain with the same CDN for the entire duration of the video.

Many factors contributing towards the selection of the CDN are bandwidth between the client and servers of different CDN, the previous history of different CDNs and non technical reasons such as pricing and business contracts[24]. A packet trace which was periodically sent to the server contained the detailed information of the status of the client machine at that time and other problems etc. The status reports are been sent to the t.hulu.com server which maps to a single IP address in the US. Using the status report for a given client, hulu is able to select the respective CDN for the client[24].

To better understand the CDN selection strategy employed by Hulu, a manifest was requested every second for the same video from the same computer for 100 seconds. From the fig 9, it was inferred that the network condition did not depend on the network condition[24].

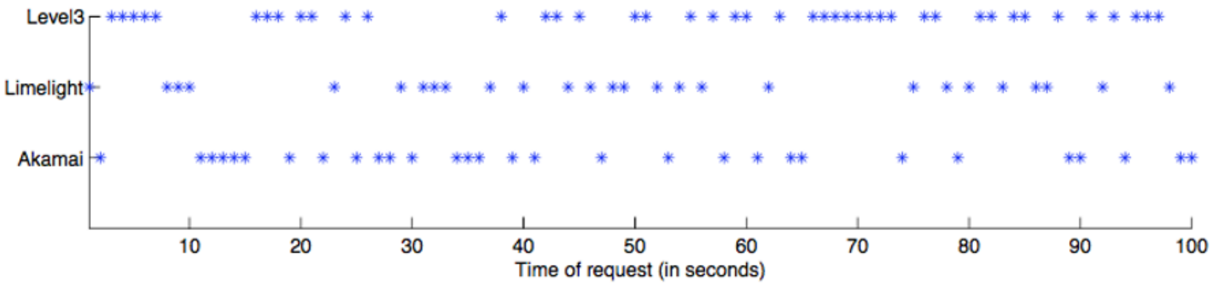


Fig 9 – CDN preference change in a short interval

A get-flash-videos tool[24] was used to collect the manifest files from 61 different videos of the different genres length, popularity and ratings available on Hulu for 13 different locations for multiple amount of days(24 days) on different networks on different ISP's including Comcast, AT&T, Verizon and CenturyLink[24]. In fig 10, level3 CDN is the preferred CDN for hulu with a CDN preference percentage of 47% over limelight(47%) and Akamai(28%). Similar trend in fig 11 was captured for clients from different geographical regions. The experiment was carried out at various locations and at different times for 24 days and the CDN preference of level3 did not change.This can be observed from fig 12 and fig 13 which are CDN preference over videos and time respectively.

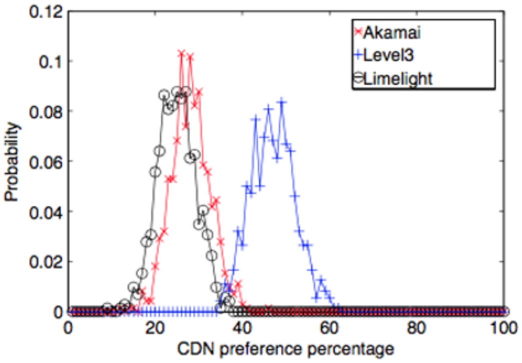


Fig 10 - Overall CDN preference distribution

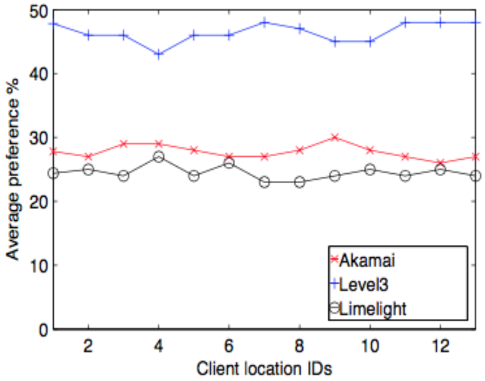


Fig 11- CDN preference observed from different geograhic region

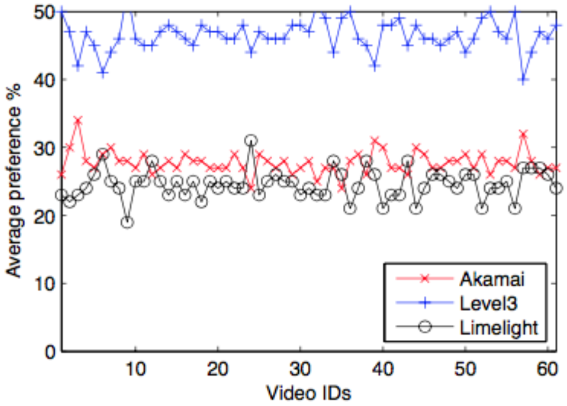


Fig 12 - CDN preference distribution over different videos

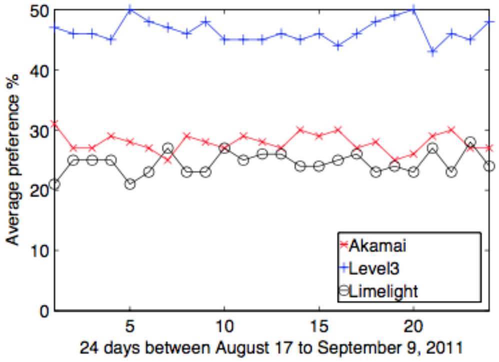
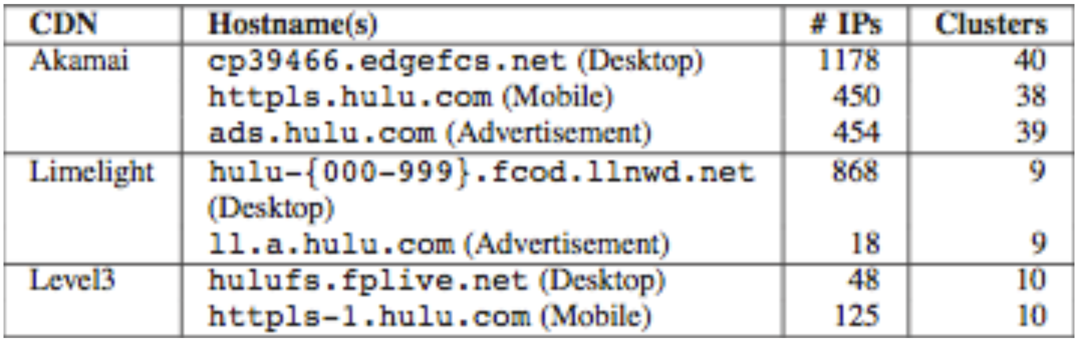


Fig 13 - CDN preference distribution over time

Fig 14 below shows the different CDN servers used by HULU for both the mobile and desktop versions[24].



CDN	Hostname(s)	# IPs	Clusters
Akamai	cp39466.edgefcs.net (Desktop)	1178	40
	httpls.hulu.com (Mobile)	450	38
	ads.hulu.com (Advertisement)	454	39
Limelight	hulu-{000-999}.fcod.llnwd.net (Desktop)	868	9
	ll.a.hulu.com (Advertisement)	18	9
Level3	hulufs.fplive.net (Desktop)	48	10
	httpls-1.hulu.com (Mobile)	125	10

Fig 14 - CDN servers for hulu

Through various experiments[24], it was found that hulu desktop clients Akamai and Level3 has only one hostname while limelight has 1000 hostnames. After the obtaining the IP addresses of each CDN, the ICMP ping mechanism was used[24] to obtain the latencies for the different IP's addresses and map them to different PlanetLab nodes to obtain their locations. Level3 IP addresses serving Hulu content do not respond to ping probes.Therefore to measure the latencies to other “nearby” IP addresses. Using these locations of the different IP addresses, clusters were created. In summary, Akamai uses the largest number of IPs (1178) and clusters (40), while the number of clusters for Limelight and Level3 is much smaller, at 9 and 10 clusters respectively. Also, the number of IPs used by Limelight is close to that of Akamai, and is much larger than the IP addresses used by Level3[24].

For mobile devices, from fig 14 that the only Akamai and Level3 CDN's are been used. Akamai has 450 IP address and 38 clusters for the mobile devices and Level3 has 125 IP addresses and 10 clusters. Also for hulu advertisements has a separate hostname is being provided by Akamai and Limelight. Akamai as 454 unique IP addresses assigned with 39 clusters and Limelight having 18 IP addresses and 9 clusters[24].

D. Netflix

History

Storytelling has always been at the core of human nature. Netflix lies at the intersection of Internet and storytelling. Netflix can be best described as “internet television“. The main strength of Netflix is that it allows members to stream any video from a wide range of collection of movies and tv shows to any Internet connected devices. Internet TV is about choice: what to watch, when to watch, and where to watch, it offers a chance for the viewer to watch their favorite movies or TV shows at a single place, which in a normal cable might be spread across 10 to 20 channels.

Architecture

The Netflix architecture mainly consists of the following four key components : Netflix data center, Amazon cloud, CDNs and players

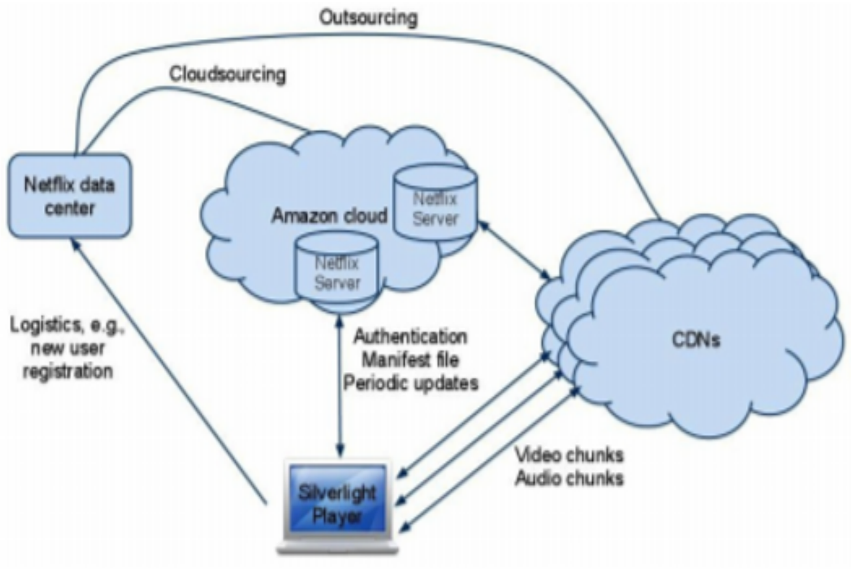
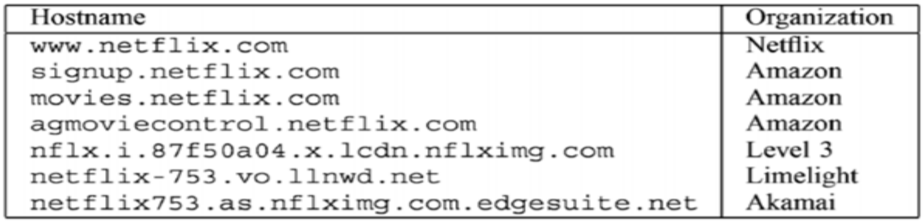


Fig 14 - Architecture of Netflix

- **Netflix data centers:** Netflix has its own IP address space for the hostname www.netflix.com[25]. This primarily has two functions[25]
    - Registration of new user accounts and capture of payment information such as credit card or PayPal .
    - Redirect the users who are signed in to movies.netflix.com and the users who are not signed in are redirected to signup.netflix.com.
- **Amazon cloud** : Netflix uses various Amazon cloud services which do key functions such as content ingestion, log recording/analysis, DRM, CDN routing, user sign-in, and mobile device support. The Netflix host servers like agmoviecontrol.netflix.com and commovies.netflix.com on Amazon cloud.
- **Content Distribution Networks (CDNs):** Netflix employs multiple CDNs to deliver the video content to end users. The encoded and DRM protected videos are sourced in Amazon cloud and copied to CDNs[25]. Netflix employs three CDNs: Akamai, Limelight, and Level-3. For the same video with the same quality level, the same encoded content is delivered from all three CDNs.
- **Players :**Netflix uses Silverlight to download, decode and play Netflix movies on desktop web browsers.. Netflix uses Wii, Roku, etc for the mobile devices players for streaming the videos across the HTTP interface. Netflix uses the DASH (Dynamic Streaming over HTTP) protocol for streaming[25]. In DASH, each video is encoded at several different quality levels, and is divided into small 'chunks' , which are video segments of no more than a few seconds in length.

The client requests one video chunk at a time via HTTP. With each download, it measures the received bandwidth and runs a rate determination algorithm to determine the quality of the next chunk to request. DASH allows the player to freely switch between different quality levels at the chunk boundaries.



Hostname	Organization
www.netflix.com	Netflix
signup.netflix.com	Amazon
movies.netflix.com	Amazon
agmoviecontrol.netflix.com	Amazon
nflx.i.87f50a04.x.lcdn.nflximg.com	Level 3
netflix-753.vo.llnwd.net	Limelight
netflix753.as.nflximg.com.edgesuite.net	Akamai

Fig 14 - Netflix Hostnames

Fig 14 gives the different hostnames used by Netflix. From the table above it is observed that majority of the hostnames of Netflix are owned by other organizations.

Servicing a Netflix client

Netflix client first downloads the Microsoft Silverlight application and then it authenticates the user[26]. Once authenticated, the player fetches the manifest file from the control server which is a agmoviecontrol.netflix.com. After this, trick play and audio/video chunks are downloaded from different CDNs[26]. Fig 15, indicates the above process for servicing the Netflix client.

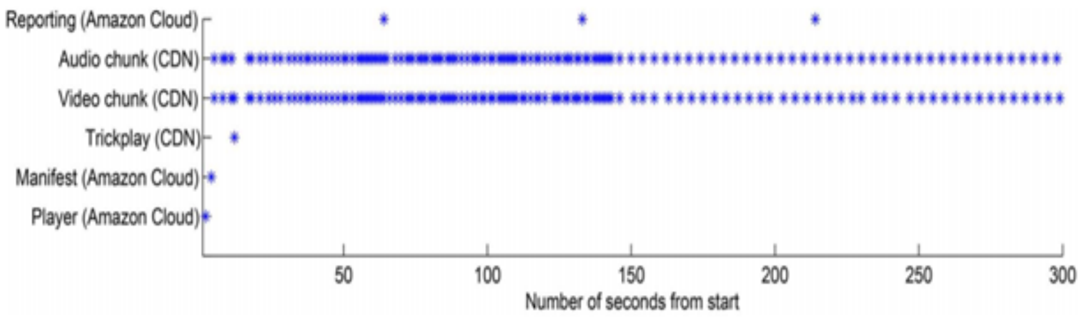
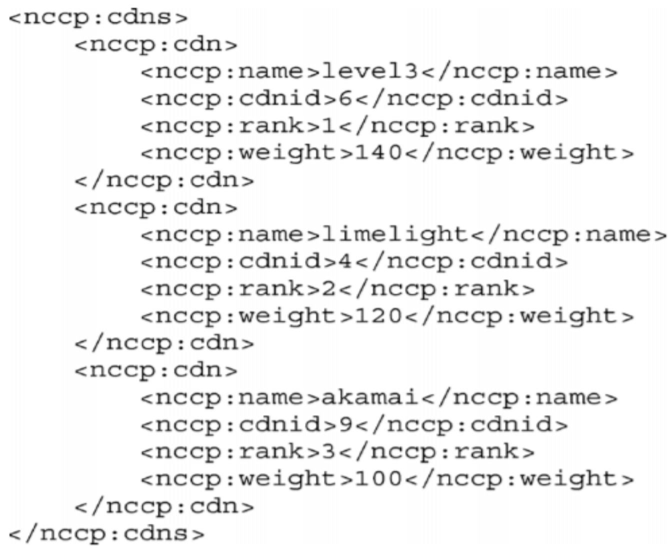


Fig 15 - Timeline in serving a Netflix client

- **Silverlight player**: Video playback on a desktop computer requires the Microsoft Silverlight browser. When the user clicks on “Play Now” button, the browser downloads the Silverlight application and then that application starts downloading and playing the video content[26]. This small Silverlight application is downloaded for every video playback.
- **Netflix manifest file**: Netflix video streaming is dependent on the instructions in a manifest file that is downloaded by the Silverlight client. The Netflix mani-

fest file provides the DASH player metadata to conduct the adaptive video streaming. The manifest files are specific to every client and is generated based on the clients playback capability. Fig 16 provides the manifest file obtained from one of the experiments. The manifest file clearly indicates the ranking of the various CDN's with there respective weights[26].



```
<nccp:cdns>
    <nccp:cdn>
        <nccp:name>level3</nccp:name>
        <nccp:cdnid>6</nccp:cdnid>
        <nccp:rank>1</nccp:rank>
        <nccp:weight>140</nccp:weight>
    </nccp:cdn>
    <nccp:cdn>
        <nccp:name>limelight</nccp:name>
        <nccp:cdnid>4</nccp:cdnid>
        <nccp:rank>2</nccp:rank>
        <nccp:weight>120</nccp:weight>
    </nccp:cdn>
    <nccp:cdn>
        <nccp:name>akamai</nccp:name>
        <nccp:cdnid>9</nccp:cdnid>
        <nccp:rank>3</nccp:rank>
        <nccp:weight>100</nccp:weight>
    </nccp:cdn>
</nccp:cdns>
```

Fig 16 - CDN list in a manifest file

- **Trickplay**: Netflix Silverlight player supports simple trickplay such as pause, rewind, forward and random seek. Trickplay is achieved by downloading a set of thumbnail images for periodic snapshots. The thumbnail resolution, pixel aspect, trickplay interval and CDN from where to download the trickplay file are described in the manifest file. The trickplay interval for the desktop browser is 10 seconds.
- **Audio and Video Chunk Downloading** : Audio and video are downloaded in chunks. In the beginning more contents is downloaded in order to build up the player buffer. The manifest file contains multiple audio and video quality levels. For each quality level, it contains the URLs for individual CDNs.
- **User Experience Reporting**: After the video starts, Netflix player communicates periodically with the control server agmoviecontrol.netflix.com. Log updates and constant alive messages are exchanged periodically.

Netflix and Hulu have some common approaches regarding CDNs they use , in both the cases they factor in user geographic locations, network conditions, and requested video contents[24-26] . Both have the same number of CDNs namely Akamai, Silverlight and Level 3. Although, both of them use different strategies both attempt to distribute and balance the video serving traffic among the CDN in accordance with certain latent distribution[24-26]. The main difference between Netflix and Hulu is that Netflix ties the CDN preference to user accounts[24-26], while Hulu chooses the preferred CDN for each video based on a latent distribution[24-26]. Netflix hosts its servers on Amazon web services, where as Hulu has its own server system. The differences in selection strategies between Netflix and Hulu might be based on the business considerations of the respective organizations.

IV. Improvements

Multimedia can also be extended to live streaming. Continuous research is being done for major sites , like Netflix and Hulu to live stream videos. The biggest complexity involved in live streaming these events is maintaining high streaming rate on a global front. One of the suggestions provided[27] is to use helpers (proxies) which provide partial streams to the various clients around the world. The stepping stone algorithm[27] can help in reducing delay, jitter and improve the streaming rate based on the number of proxies employed in the global front.

In [27], the helper functions ensured that the streams where divided into chunks at the source and passed into the internet. The chunks would then be propagated to various helpers and servers across the network. These chunks would then be propagated from the different servers and helper functions towards the destination where it would be rearranged to get back the original video. Using helper functions [27], we can achieve multi path and multi hop mechanism over an overlay network. In [27], the streaming rate and the delay would increase and decrease respectively with the increase in the number of helper functions.

V. Recommendation Systems

As Internet usage gets more and more popular, information overload poses an important challenge for a lot of online services such as on-demand audio and video streaming. Also with advancement of internet speeds, its easy to access contents irrespective of their size. With all this information pouring out from web, users can be overwhelmed and confused as to what, exactly, they should be paying attention. We have an estimate of more than 65 million members who stream more than 100 million hours of movies and TV shows per day on Netflix[28].

Proposed solution

A solution to the above problem, is to have a recommendation system[28]. This recommendation engine can help users discover information of interest by analyzing users profile, interests, historical behaviors and current trends. Most of the popular social networking and streaming website such as Netflix, YouTube, Facebook, Spotify, Hulu and many others are integrating a recommendation system into their services to help users discover and select information that may be of particular interest to them.

Every company have its own set of attributes to consider while building a recommendation system based on their business needs. We shall examine a few of the implementations in the below section.

A. Recommendation system by Hulu

Hulu's recommendation system is based on the characteristics of the data content it provides. Since a lot of Hulu's content is comprised of episodes or clips within a show, they recommend shows to users instead of individual videos. Its content can be mainly divided into two parts: on-air shows and library shows. On-air shows are highly important since more than half of its streaming comes from them. The traffic for a particular show or genre depends on factors like weekends, summer months.

Architecture:

Collaborative Filtering(CF)[29] is one of the recommendation algorithm that relies on user behaviour data. These are of two types: UserCF and ItemCF. ItemCF assumes that a user will prefer items similar to the assets he or she preferred previously. ItemCF is widely used by many others (for example, Amazon and Netflix). ItemCF[29] is the basic recommendation algorithm in Hulu.

Online-Architecture:

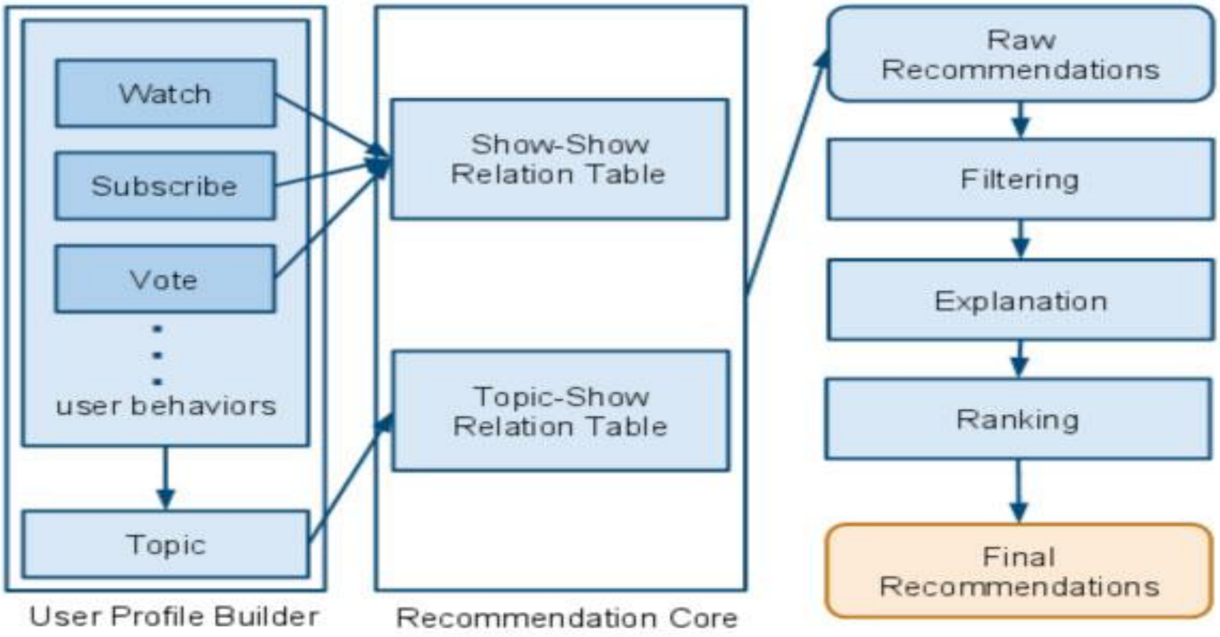


Fig 17 - Online Architecture

User profile builder: A profile is build for every user that comes into the recommendation system. The profile includes the user's historical behaviors and topics.

Recommendation Core: After generating the list of user's historical preferences on shows and topics, we put all of those similar shows into raw recommendations.

Filtering: Raw recommendation results cannot be presented to users directly. We need to filter out shows the user has already seen or engaged with the user.

Ranking: This module will re-rank raw recommendations to make them better fit users preferences.

Explanation: The explanation module generates some reasoning for every recommendation result using the user's historical behaviors.

Offline-Architecture:

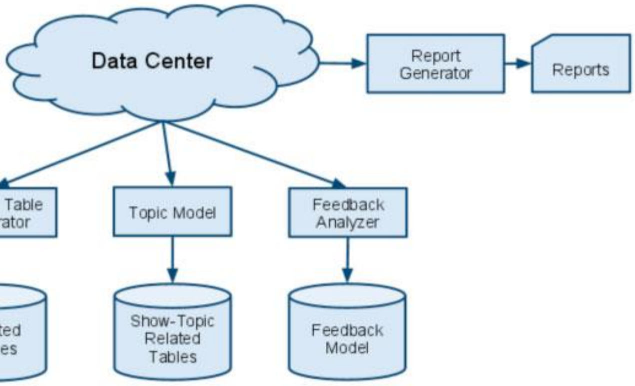


Fig 18 - Offline architecture

The following are the main components of offline architecture[29].

Data Center: The data center contains all user behavior data in Hulu. Some of them are stored in Hadoop clusters and some of them are stored in a relational database.

Related Table Generator: We use two main types of related table: one that's based on collaborative filtering (which we'll call CF), and another based on content.

Topic Model: A topic is represented by a group of shows that have similar content

Feedback Analyzer: Feedback specifically means users' reactions to recommendation results. Using user feedback can improve recommendation quality.

Report Generator: Evaluation is most important part of the recommendation system. The report generator will generate a report including multiple metrics every day to show the quality of recommendations.

B. Recommendation system by Spotify

Music Recommendation Based on Artist Novelty and Similarity

The System proposed here is driven by an emerging and somewhat different need in the music industry-promoting new talents. The system recommends songs based on the novelty of singers (or artists) and their similarity to the user's favorite artists. Novel artists whose popularity is on the rise have a higher priority to be recommended. Specifically, given a user's favorite artists, the system first determines the candidate artists based on their similarity with the favorite artists and then selects those who have a higher novelty score than the favorite artists[30].

SYSTEM OVERVIEW

The proposed music recommendation system is designed to meet the following three requirements:

- The music recommendation system should be able to deal with a relatively small list of favorite singers given by the user.

- The artists of the recommended songs should be new to the user.

- The user acceptance rate should be reasonably high.

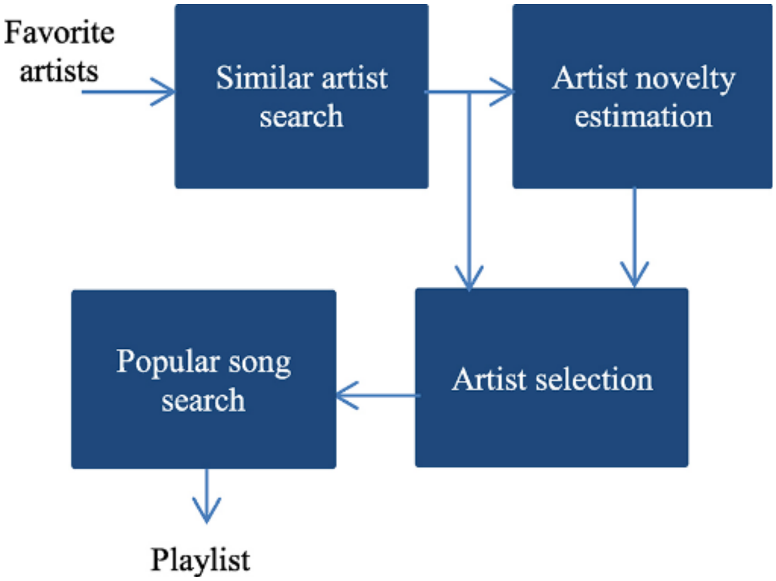


Fig 19 - Recommendation system overview

The following two labels are used for artist classification:

- Like/dislike: This label shows whether an artist is liked/disliked by a user.

- New/known: This label indicates whether an artist is new to a user or not. like/dislike label is activated.

The following two artist attributes[30] are defined:

- Similarity: The more characteristics (such as music genre, timbre, tag, and era) two artists share, the more similar the two artists are.

- Popularity: The degree popularity of an artist.

Similar Artist Search

As mentioned, artists who are similar to the favorite artists of a user are more likely to be liked by the user. Given a favorite artist a of a user and another artist b, the probability that the user would like artist b is related to the similarity between a and b and can be expressed as

$P(like|\, a, b) \propto sim(a, b)$, where sim(·,·) denotes similarity.

Spotify recommendation system[30] uses the list of favorite artists a_1, a_2, ... a_n of a user as input,

$$A \quad = \quad [a_1, a_2, \dots \dots, a_n]\,.$$

For every artist a_i in A, we search for the top N similar artists denoted by S(a_i) according to the similarity score sim(a_i):

$$S(a_i) = \; [b_{i,1}, b_{i,2}, \dots \dots, b_{i,N}\,] \;,$$

$$sim(a_i) = \; [sim(a_i, b_{i.1}), \dots \dots, sim(a_i, b_{i.N})]\,,$$

where b_i,i is the j^{th} similar artist of a_i.

Artist Popularity Estimation

Given a user U and a candidate artist b_i,i' the conditional probability that the artist can be labelled known is proportional to the popularity of the artist in the eyes of the user. That is,

$$P(known|u, b_{i.i}) \propto pop(b_{i.i}|u)\,,$$

where pop($b_{i,j}$|u) denotes the popularity of b_i,i viewed by u.

The conditional probability that the artist can be labelled new can be considered as the degree of novelty of the artist with respect to the user. Therefore, we have

$$\begin{aligned} Novelty(b_{i,j}|u) &= \; P(new|u, b_{i,j}) \\ &= 1 - P(known|u, b_{i,j}) \\ &= 1 - pop(b_{i,j}|u) \end{aligned}$$

where Novelty(-) denotes novelty. In our system, pop($b_{i,j}$lu) is obtained by normalizing the popularity of b_i,i by the popularity of $a_{i'}$ That is,

$$pop(b_{i,j}|u) = \frac{log\,pop(b_{i,j})}{log\,pop(a_i)},$$

where pop('), which denotes popularity, is obtained by the total count of responses to a Google search with the name of artist as the key word.

Artist Selection and Popular Song Search

Similarity of a candidate artist with respect to a favorite artist of a user reflects the preference level of the user for the candidate artist. Thus, the recommendation score of a candidate artist is proportional to its similarity to the favorite artist. Furthermore, as the recommendation is also based on novelty, the final recommendation score[30] for a candidate artist is obtained by multiplying the two factors together. That is,

$$Score(b_{i,j}) = Sim(a_i, b_{i,j}) \times Novelty(b_{i,j}),$$

where Score(·) denotes the recommendation score of the candidate artist given a_i is a favorite artist of the user. Among the candidate artists similar to a_i, the one with the highest recommendation score is selected, and the most popular song of this artist is added to the playlist.

C. Recommendation system by Netflix

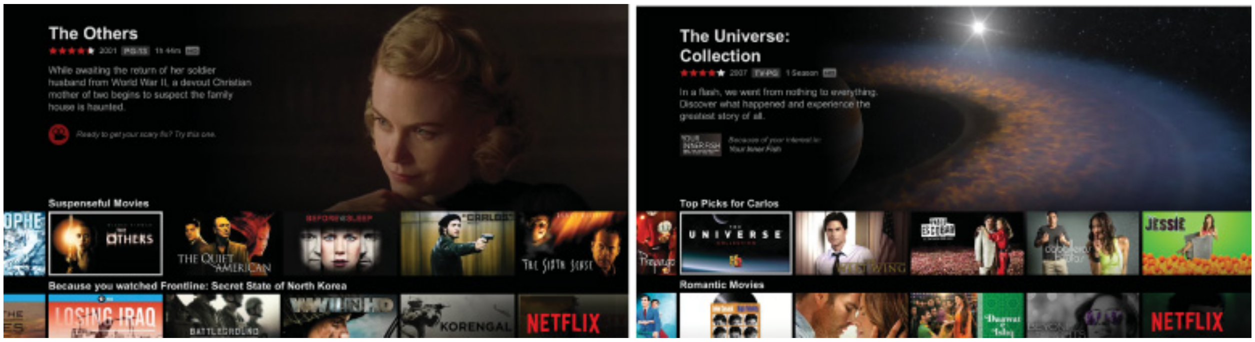
Netflix recommendation system is a collection of different algorithms serving different use cases that make up the efficient netflix recommendations.

Personalized Video Ranker: PVR

Each row of suggestion videos in the homepage typically come from its specific single algorithm. Genre rows such as suspenseful movies are driven by personalized video ranker (PVR) algorithm[31]. This algorithm orders the entire catalog of videos (or subsets selected by genre or other filtering) for each member profile in a personalized way. The resulting ordering is used to select order of videos in a genre and other categories, and this is the reason why same genre row shown to different users may vary .

Top-N Video Ranker

Netflix has a Top N video ranker that produces the recommendations in the Top Picks row shown on the right of Figure 20. The goal of this algorithm is to find the best few personalized recommendations in the entire catalog for each member, that is, focusing only on the head of the ranking, a freedom that PVR does not have because it gets used to rank arbitrary subsets of the catalog.



While awaiting the return of her soldier husband from World War II, a devout Christian mother of two begins to suspect the family house is haunted.

Ready to get your scary fix? Try this one.

In a flash, we went from nothing to everything. Discover what happened and experience the greatest story of all.

Because of your interest in: Your Inner Fish

Fig 20- An example of the page of recommendations

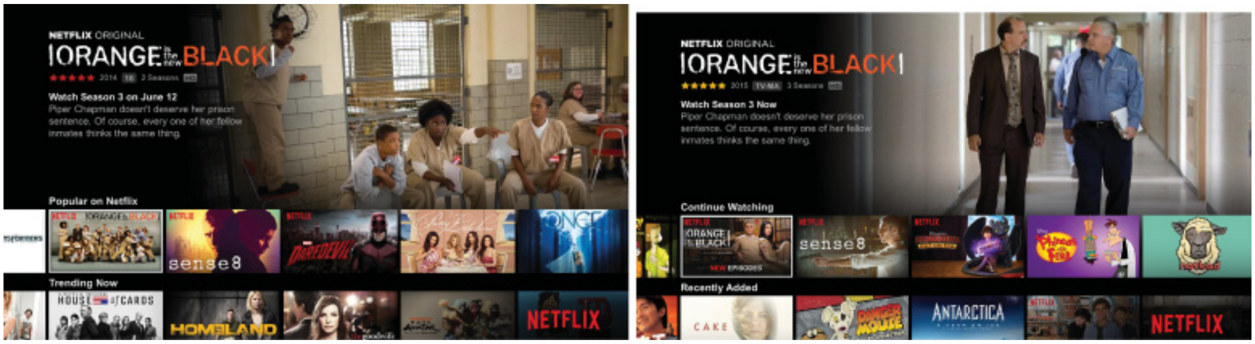
Trending Now

Netflix has also found that shorter-term temporal trends, ranging from a few minutes to perhaps a few days, are powerful predictors of videos that its members will watch, especially when combined with the right dose of personalization, giving them a trending ranker used to drive the Trending Now row.

Continue Watching

Given the importance of episodic content viewed over several sessions, as well as the freedom to view non episodic content in small bites, another important video ranking algorithm is the continue watching ranker[31] that orders the videos in the Continue Watching row

In contrast, the continue watching ranker sorts the subset of recently viewed titles based on our best estimate of whether the member intends to resume watching or re watch, or whether the member has abandoned something not as interesting as anticipated.



Watch Season 3 on June 12

Piper Chapman doesn't deserve her prison sentence. Of course, every one of her fellow inmates thinks the same thing.

Watch Season 3 Now

Piper Chapman doesn't deserve her prison sentence. Of course, every one of her fellow inmates thinks the same thing.

Fig 21- Two rows of recommendations on Netflix home page

Video-Video Similarity

Because You Watched (BYW)[31] rows are another type of categorization. A BYW row anchors its recommendations to a single video watched by the member. The video-video similarity algorithm, which we refer to simply as “sims,” drives the recommendations in these rows.

Page Generation, Row Selection and Ranking

The page generation algorithm[31] uses the output of all the algorithms already described to construct every single page of recommendations, taking into account the relevance of each row to the member as well as the diversity of the page.

VI. CONCLUSION

In this survey, our main focus was to capture the architecture of certain social networking sites such as Facebook, Hulu, Netflix and Instagram. Facebook and Instagram uses the caching servers[18,19] to replicate data and provide the multimedia data to the respective users. However Hulu and Netflix uses CDN based approach to provide content to their users. Also both of them use the same CDN’s providers namely Akamai, LimeLight and Level3[24-26]. The main difference between Netflix and Hulu is that Netflix ties the CDN preference to user accounts, while Hulu chooses the preferred CDN for each video based on a latent distribution. This is due to fact that both Netflix and Hulu incorporate different business models[24-26].

We have also suggested the use of helper based mechanism which uses stepping stone algorithm which can provide live streaming feature for Hulu and Netflix[27]. Helpers are used to achieve low delay and high bit rate, to provide rich path diversity, and improve system throughput and jitters in streaming[27].

Most of the mentioned websites above use recommendation algorithms to serve the right content to their users. Netflix recommendation system comprises of various algorithms that generate the content of home page which is divided into various categories. It is all personalized based on user history and behaviors. Hulu uses collaborative filtering(CF) algorithm which is based on userCF which assumes that a user will prefer items which are liked by other users who have similar preferences to that user. Music recommendation based on artist novelty and similarity[30] recommends artists who are not yet famous- promoting new talents in Spotify.

VII. REFERENCES

[1]Audio and Video File Formats and File Sizes http://www.columbia.edu/itc/visualarts/r4110/f2000/week07/07_06_Audio_Video_File_Size.pdf

[2]Wallace, Gregory K. "The JPEG still picture compression standard." IEEE transactions on consumer electronics 38.1 (1992).

[3]Le Gall, Didier. "MPEG: A video compression standard for multimedia applications." Communications of the ACM 34.4 (1991).

[4]Haskell, Barry G., Atul Puri, and Arun N. Netravali. Digital video: an introduction to MPEG-2. Springer Science & Business Media, 1996.

[5]Pereira, Fernando CN, and Touradj Ebrahimi. The MPEG-4 book. Prentice Hall Professional, 2002.

[6]Chang, Shih-Fu, Thomas Sikora, and Atul Purl. "Overview of the MPEG-7 standard." IEEE Transactions on circuits and systems for video technology 11.6 (2001): 688-695.

[7]Dapeng Wu, Student Member, IEEE, Yiwei Thomas Hou, Member, IEEE, Wenwu Zhu, Member, IEEE, Ya-Qin Zhang, Fellow, IEEE, and Jon M. Peha, Senior Member, IEEE "Streaming Video over the Internet: Approaches and Directions" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 11, NO. 3, MARCH 2001

[8]Divyesh Jadav, Member IEEE, Alok N. Choudhary, Member IEEE, and P. Bruce Berra, Fellow IEEE, "Techniques for Increasing the Stream Capacity of A High-Performance Multimedia Server". High Performance Computing, 1996. Proceedings. 3rd International Conference.

[9] Maxim Claeys, Student Member, IEEE, Niels Bouten, Student Member, IEEE, Danny De Vleeschauwer, Werner Van Leekwijck, Steven Latré, Member, IEEE, and Filip De Turck, Senior Member, IEEE. "Cooperative Announcement-Based Caching for Video-on-Demand Streaming". IEEE Transactions on Network and Service Management (Volume: 13, Issue: 2, June 2016)

[10]Xiangyang Zhang and Hossam Hassanein, Telecommunications Research Laboratory, School of Computing Queen's University, Kingston, Ontario. "Video on-demand streaming on the Internet — A survey".

[11]Xonia Ivonne Olavarrieta, Alberto Leon-Garcia, University of Toronto. "Large Scale Distributed Storage and Search for a Video on Demand Streaming System". Next Generation Internet Networks, 2008. NGI 2008.

[12] Tavanapong W, Tran M, Zhou J, Krishnamohan S. Video caching network for on-demand video streaming. InGlobal Telecommunications Conference, 2002. GLOBECOM'02. IEEE 2002 Nov 17 (Vol. 2, pp. 1723-1727). IEEE.

[13]P.Sumari, M.Merabti and R.Pereira," Video-on-demand server: Strategies for improving performance ".IEEE Proceedings - Software (Volume: 146, Issue: 1, Feb 1999).

[14]“The Essential CDN guide” https://www.incapsula.com/cdn-guide/what-is-cdn-how-it-works.html

[15] “Facebook” https://www.facebook.com

[16]“The history of social networking by Digital Trends staff May 14,2016” http://www.digitaltrends.com/features/the-history-of-social-networking/

[17]“How Facebook works by Alan Zeichick June 23, 2008” https://www.technologyreview.com/s/410312/how-facebook-works/

[18]“The technology behind preview photos Brian K Cabral Edward Kandrot August 6, 2015” https://code.facebook.com/posts/991252547593574/the-technology-behind-preview-photos/

[19]“Search Architecture” https://engineering.instagram.com/search-architecture-eeb34a936d3a

[20]“Instagration Pt. 2: Scaling our infrastructure to multiple data centers.” https://engineering.instagram.com/instagration-pt-2-scaling-our-infrastructure-to-multiple-data-centers-5745cbad7834

[21]“Sharding & IDs at Instagram” https://engineering.instagram.com/sharding-ids-at-instagram-1cf5a71e5a5c

[22]“Scaling the Datagram Team” https://engineering.instagram.com/scaling-the-datagram-team-fc67bcf9b721

[23] Dilip Kumar Krishnappa, Samamon Khemmarat, Lixin Gao, Michael Zink. University of Massachusetts Amherst, USA. "On the Feasibility of Prefetching and Caching for Online TV Services: A Measurement Study on Hulu".

[24] Adhikari, Vijay Kumar, Yang Guo, Fang Hao, Volker Hilt, and Zhi-Li Zhang. "A tale of three cdns: An active measurement study of hulu and its cdns." In Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on, pp. 7-12. IEEE, 2012.

[25] Vijay K. Adhikari, Yang Guo, Fang Hao Member IEEE, Volker Hilt Member IEEE, ZhiLi Zhang Fellow IEEE, Member ACM, Matteo Varvello, and Moritz Steiner. "Measurement Study of Netflix, Hulu, and a Tale of Three CDNs". IEEE/ACM Transactions on Networking (Volume: 23, Issue: 6, Dec. 2015)

[26] Vijay Kumar Adhikari, Yang Guo, Fang Hao, Matteo Varvello, Volker Hilt, Moritz Steiner and Zhi-Li Zhang. "Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery". INFOCOM, 2012 Proceedings IEEE.

[27] Ren, Dongni, Yisheng Xu, and S-H. Gary Chan. "Beyond 1Mbps global overlay live streaming: The case of proxy helpers." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 11, no. 2 (2015): 26.

[28]Seong-Eun Hong, Hwa-Jong Kim. "A Comparative Study of Video Recommender Systems in Big Data Era".Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference.

[29]“Liang Xiang, Hulu's Recommendation System” http://tech.hulu.com/blog/2011/09/19/recommendation-system.html

[30] Ning Lin, Ping-ChiaTsai, Yu-An Chen, Homer H.Chen. "Music Recommendation Based on Artist Novelty and Similarity", Multimedia Signal Processing (MMSP), 2014 IEEE 16th International Workshop.

[31] CARLOS A. GOMEZ-URIBE and NEIL HUNT, Netflix, Inc. "The Netflix Recommender System: Algorithms, Business Value and Innovation". ACM Transactions on Management Information Systems (TMIS), Volume 6 Issue 4, January 2016.