Video On Demand - High Performance Cost-Effective Model

Balaji Balasubramani
Department of Computer & Information Science & Engineering
University of Florida
Gainesville, FL, US
UF ID: 9876-3981

Divyareddy Surapa Reddy
Department of Computer & Information Science & Engineering
University of Florida
Gainesville, FL, US
UF ID: 1372-7408

Mohit Kalra
Department of Computer & Information Science & Engineering
University of Florida
Gainesville, FL, US
UF ID: 1390-6151

Priya Ramchandra Prabhu
Department of Computer & Information Science & Engineering
University of Florida
Gainesville, FL, US
UF ID: 9556-1044

Rohan Sanjay Shahane
Department of Computer & Information Science & Engineering
University of Florida
Gainesville, FL, US
UF ID: 6859-1943

Abstract **- Video streaming service is a substantial portion in the present day's world of the internet and is exploding. Studies have predicted that videos will account to 81% of the internet traffic by 2021 [32]. The most challenging research for the past decade is performed on how to build a cost effective distributed VoD architecture while maximizing the quality of experience for the end users. After doing a study on how to improve the performance of VoD systems, we have identified transcoding, storage, caching, load balancing, and CDN integrations as the major contributing areas for providing users a better quality of experience. Video streams usually require conversion, also known as transcoding depending on the device the client uses to stream live or on-demand videos. However, this is an expensive option to transcode all the videos because of the high computation, storage, and retrieval involved in the management of these videos. Due to the constant increase of network traffic in the network backbone, caching plays a significant role in helping the overall system by storing the multimedia content nearer to the end user. To balance the load across the network while transmitting the video it is important to maintain constant metrics like bit rate, movements of frames. By combining VoD with CDN, high quality of service can be provided to the users with less bandwidth cost and maximum throughput. Various methods from the recent studies are identified and explained in this paper for each of these sections, and based on that we have proposed a hybrid model that combines all of these approaches to provide a high performance and cost-effective design for VoD servers.**

Keywords: VoD (Video-on-demand), Transcoding, Storage, Caching, Load balancing CDN, cost-effective, performance

I. INTRODUCTION

In the earlier times, people had to sit in front of the television to watch their favorite show or movie at a given time. However, since the Internet has become very popular which allows the user to watch the video at their convenience on their smartphones or laptops. This is a major reason for the ever-increasing popularity of Video on Demand Services.

Irrespective of what kind of device videos are being streamed on, they need to be transcoded to make sure that the resolution, frame rate and the bandwidth of the device is compatible with the transcoded video.

Video transcoding is a procedure which is quite heavy in the terms of computing and hence it is practically not possible to transcode the video

for every device. Hence, one alternative was to transcode the videos in different ways and store them. But again, this would incur a huge cost for storage and upgrading the infrastructure to adjust with the ever-growing requirements of video transcoding. To overcome this, the paper [2] has proposed a research to transcode the video streams in a lazy manner by making use of the computing services provided by cloud storages. The paper proposes to make sure that the Quality of Services is not compromised by retrieving the videos from the cloud in a lazy manner.

The main concerns taken care of in the paper are:

- Improving clients' contentment by making sure the video streams startup delay and presentation deadline miss rate is delayed.
- Taking care of clients' QoS requirements and service providers' cost while developing dynamic cloud resource policy.

This paper makes the following contributions:

- Cloud-based Video Streaming Service to permit service providers to make use of cloud services with low cost and maximum user satisfaction.
- Having minimum deadline miss rate and minimum startup delay while developing a method to link the tasks on the cloud resources.

Though different algorithms and approaches to determine and reduce the number of pre-transcoded videos are developed, video streaming providers still need large storage for the video files. Many methods are proposed to pre-transcode and store the videos based on various factors like the popularity of the video. But the exceptional growth of the Internet and video contents makes this subset of videos to sum up to a huge size. The next biggest challenge that streaming servers face is the ability to retrieve video files efficiently. Therefore, for economic and efficient storage of data, VoD servers should consider what type of cloud storage and storage devices are to be used for different types of video content.

With rising demand and having huge file sizes of multimedia videos, the network traffic in the network resulting from VoD requests is very large. This added traffic causes load to the network to increase in the delay at the point of end user. In order to reduce the traffic in the backbone network due to rising popularity of VoD services and to reduce the delay at the users end, various caching mechanisms are used and the multimedia content is stored in a cache near the user to enhance the service provided to them.

Video on Demand along with Content Delivery Network holds an enormous impact in providing the high-quality video. Thus video streaming content providers change to the VoD CDN approach [29]. In the present day, users are in search of high broadcast quality along with faster response time without any delay. The considered design of combining CDN with VoD can meet the above demands with less cost and better efficient results.

The contents of the video are cached at multiple CDN servers, which are distributed globally. Once when a user requests for the video, the server which is nearby handling the request accelerates the response with the requested video. This need for high performance in delivering the content has driven the development of new industries based on CDN development like Akamai, Limelight, etc. Netflix, Hulu uses these CDN services, pay them for delivering content with flat latency rates, and maximizing the throughput.

The next section of the paper, II explains the various algorithms and approaches developed and proposed for each of the mentioned areas in the past decade to design a highly performing and cost-efficient system. Section

III discusses suggesting how various approaches in each of the components can be combined to make the overall system scalable and finally section IV concludes it.

II. APPROACHES

A. Transcoding:

Adjusting the features of the video by streaming servers to match the spatial resolution, frame rate, bit rate and the client's network bandwidth is known as transcoding. Following are the types of transcoding operations:

1. Bit Rate Adjustment:

The bit rate of the videos being streamed must be high to make sure that the quality of the video is high. High bit rate comes with a requirement of a huge network bandwidth for the video transmission. This factor must be considered by the service providers to ensure trouble-free streaming.

2. Spatial Resolution:

The encoded size of a video in terms of dimension is known as spatial resolution. Spatial resolution is required because the dimensions of the original video does not necessarily be the same as the dimensions of the users' screens. Spatial resolution usually requires to be reduced without compromising the video quality.

3. Temporal Resolution Reduction:

In cases where the client's device supports low frame rate, temporal resolution reduction happens, and the streaming servers have to drop a few frames. Methods to achieve temporal resolution reduction have to make sure that the motion vectors do not become invalid for the incoming frames.

Cloud Based Video Streaming Services Architecture:

The Cloud Based Video Streaming Service Architecture is shown in the figure 1, from [2]. The architecture displays the operations performed when a video is requested by a client from a streaming service provider. The main components of the architecture include Video splitter, task scheduler, transcoding virtual machines, elasticity manager and caching policy. These operations provide low cost on-demand video transcoding with a good QoS.

1. Video Splitter:

In this component of the architecture, as the name suggests, the video stream is split in multiple GOPs which are further transcoded without having the need to depend on other GOPs. In [4], the authors have constructed a transcoding segment using multiple GOPs. In

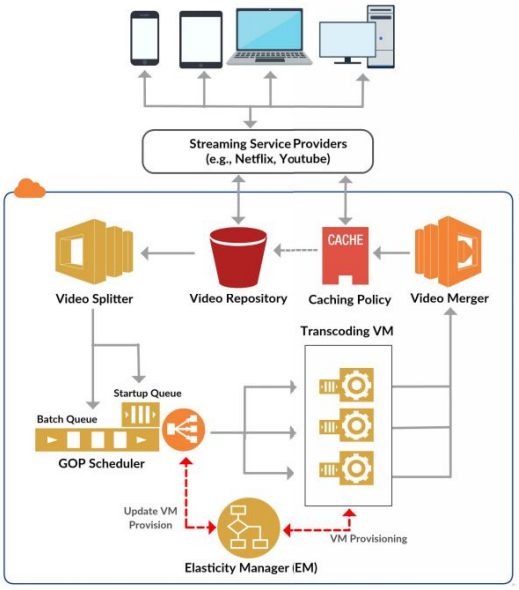

Fig. 1 An overview of the Cloud-based Video Streaming Service Architecture[]

With this approach, we have considered that transcoding segments using a single GOP works better for scheduling and hence every GOP is considered to be an individual deadline.

Earlier studies showed that if the deadline is missed by a GOP, the transcoding still has to be completed. In this survey, we consider

close-GOP type where every GOP processing can be done independently.

2. Transcoding GOP Task Scheduler:

The mapping of GOPs to transcoding servers is done by the transcoding task scheduler, also known as transcoding scheduler. The customers' QoS demands require minimum startup delay as well as minimum deadline miss rate. These are taken care of by the transcoding scheduler.

3. Transcoding Virtual Machine (VM):

GOPs are loaded in a local queue of VM before the execution starts. The GOPs are mapped to the VM by the scheduler till the time the local queue becomes full. As completely dispensed VMs that execute transcoding tasks are homogeneous, size of their neighborhood lines is the equivalent over all the allocated VMs. When any position in the VM local queue becomes empty, the scheduler is notified of the same and it again maps the GOP to the VM. In this paper, we consider that the tasks in the queue follow First Come First Server manner.

4. Elasticity Manager (EM):

Elasticity Manager looks after the transcoding VM functionality in the CVSS architecture. Further it resizes the cluster to satisfy clients' requirements of QoS and also minimize the cost incurred by the video stream service provider. To achieve this, the EM consists of elastic resources servicing the protocols that take care of allocation and deallocation of the VMs depending on the clients' requirements.

EM makes sure to allocate VM and add those to the cluster in case where the QoS of the video streams gets violated at a higher rate or the local queue sizes of the scheduler increases. EM functions periodically and it monitors whether the allocated VMs are enough to satisfy the QoS requirements. When the set of allocated VM clusters is updated, the VM notifies the scheduler about the ordering of the VM cluster.

5. Video Merger:

The video merger as the name suggests is responsible to put every transcoded GOPs in the correct place to deliver the required output video stream. Video merger transmits the transcoded video to the cloud or the repository to be available to the clients.

6. Caching Policy:

The studies in paper [5] show that access rate of the videos obey the long tail distribution method. This means that there are some videos that have high demand and are requested by the users at high frequency whereas there are quite many videos that are very rarely requested by the users. The videos that are frequently requested may come under the trending category and transcoding the trending videos again and again leads to high cost to the video streaming service providers. In order to avoid this unnecessary transcoding of high frequency videos, this proposed architecture implements a caching policy wherein it makes a decision of whether or not a transcoded video should be cached. In cases where the videos are rarely requested by the clients, there is no requirement to cache the videos. Such videos can be transcoded on demand in a lazy manner.

B. Storage

Video on demand (VoD) popularity gain has been substantial in recent times. Several platforms have been launched, providing a variety of streaming content to the end-users. With the constant improvements in the range of bandwidth from the internet service providers, the quality of the video content also increased. As the quality of the videos improved, the size of the video files also increased. However, to adapt to the client's devices, the streaming providers compress the videos. The process of this conversion is called Video Transcoding. Video stream providers follow two approaches, pre-transcoding, and re-transcoding. In the pre-transcoding, a video

stream is converted and stored in the servers in many versions and resolutions before-hand, which requires large, powerful storage services.

There are other types of multimedia servers like textual data servers, but they significantly differ from the VoD servers. A VoD server has to deliver the data continually because of the video streams that are a continuous sequence of video frames. If they fail to deliver the streams on time, hiccups and jitters will result in the video play. Large storage and high bandwidth are the other must-have characteristics of a VoD server [8]. The length and bit rate of a video is used to calculate the size of that video. With the growing media content, these VoD servers should be able to store all this data and should serve multiple concurrent stream requests. Many users request different video streams at the same time, and these video streams may belong to the same file or different files. Even though it belongs to the same file, the requested streams might be at various parts of the file. Hence the storage devices should handle numerous read operations simultaneously and should support large read bandwidth [8]. Thus, overall, storage devices and VoD servers should support continuous data retrieval, large storage, and high bandwidth capacity to be scalable.

The two popular storage devices are Hard Disk Drive (HDD) and Solid State Drive (SSD). Each of them has a few pros and cons.

- *Hard Disk Drive* is a storage device that is made up of spinning disks. It magnetically stores the data, and using an arm and multiple heads, they read and write the data. To access the data in a particular position, the arm moves the head to the specific location of the disk [11].

- *Solid State Drive* does not have any spinning or moving disks. It stores the data in ICs (Integrated circuits) that provide a vital improvement in terms of size and performance.

Comparison between HDD and SSD: HDDs provide high capacity as it has been developing regularly. But the delay and access latency of the disk is high because of the mechanical operations inside the spinning disk in accessing the data. A large number of input-output operations when performed results in the delay because of the disk platter movements. However, since SSDs do not contain any moving parts and have integrated chips, they do not possess any access latency. Also, the data transfer rate in SSD is much higher, and random read-write operations perform better when compared to HDD [12]. In terms of cost, HDDs are cheaper than SSDs. The cost of a regular 1TB hard drive ranges between $50 and $60, whereas the same SSD would cost $100. When we compare the cost per GB, HDD lies in the range of 5 to 6 cents, and SSD around 10 cents [13]. VoD servers require the storage capacity to be hundreds of Terabytes, and the difference in the cost would then drastically increase.

When multiple simultaneous stream requests are sent to the server, the delay and the latency of the HDDs decrease the performance of the VoD servers. With the growing popularity of streaming services, big storage and high-performance disks are required for the VoD servers to be scalable. Particularly for streaming the data in real-time, SSD would be the best choice. However, the cost factors of the SSD makes it less preferable, and HDD suits more in such scenarios. Thus the tradeoff between the cost and performance remains one of the biggest challenges in designing scalable VoD storage servers.

The response time of I/O operations was explained in [9]. The four main factors that drive the response of I/O time in the disk drives are rotation, seek latency, command overhead, and transfer time of the data. Rotation and seek latency are the time taken by

the head to move to the particular cylinder and then to the required sector containing the needed data. The time taken to act upon a request by the hard disk processor is the overhead time of the command. The data transfer time is the total time to transfer the data to the client, which mainly depends on the size of the data and the transfer rate. Unlike HDD, the average response in SSD time only varies on the transfer time of data. SSD does not lose any time in locating data due to the absence of any movement [9].

Many hybrid approaches are proposed, in which both HDD and SSD are integrated into the VoD storage architecture to gain the full advantage of both HDD and SSD. Having the HDD enables the server to have a large capacity of storage at a lower cost, and the random read-write operations of SSD helps the server to eliminate the high latency delays and increases performance. Many of these approaches follow the method of categorizing the video contents into two, popular videos which are trending or more frequently requested ones, and the other type of less frequent videos. Some of the approaches are explained in the next sections.

A standard VoD server is distributed and contains a central server, proxy, and a unit cluster [9]. The unit cluster serves the client requests. If it holds the requested data, it directly streams the data. Otherwise, it fetches the data from the central server. [9] suggested a media management server and a hybrid storage server inside the unit cluster to collect the metadata about the location of video files in the storage disks. The hybrid storage server is a combination of multiple HDDs and SSDs. The popular videos are saved in the SSD as they are more frequently requested, and having them in SSD helps the server to access and stream it quickly. HDD holds all other videos and is fetched on each request. RAM compliments HDD by acting as a buffer cache for the less frequent videos. The access pattern of the videos is collected and is used to update the contents in SSD regularly. The experimental results of this approach showed that the HDD+SSD storage Vod server offers better performance and is more preferred for large scale VoD systems than the servers that were built only on HDD.

[10] extended this approach to make it more cost-effective, minimize the latency between the start of the request and the stream of the video content, and further maximize the simultaneous client requests at the server. HDD remains the primary storage in the server, and SSD is used to store the popular videos. But, it differs slightly from the above approach in having the portion of SSD as a cache buffer instead of RAM. This buffer cache is employed to store the prefetched data from the disk, which are typically less frequent videos. Thus SSD replaced RAM to utilize its traits of high input-outputs per second and is also used as a cache for simultaneous writing safely. Since the buffer cache is limited, Least Frequently Used (LFU) caching mechanism is used, which replaces the video files with less frequency when there is no space available [10].

In another approach [14], SSDs of small capacities were integrated with HDD to form a hybrid storage subsystem to provide high performance. In addition to that, based on the studies conducted on users behaviors in using VoD systems, a dynamic replication strategy has been implemented. According to study, most of the users end the requested video after a few minutes from the beginning [15]. Therefore, the first several minutes of the frequently accessed videos are stored in the SSD and the remaining portions of the videos are stored in the HDD. So, when the user requests for the trending video, the first few minutes are streamed quickly from the SSD with its high speed accessibility avoiding any startup latency.

Video streaming platforms have widely become popular in recent years because of the number of videos streamed on various devices. At the same time, cloud computing and cloud services have also gained popularity, and many companies are moving towards the cloud. Cloud offers the two main required services for video streaming.

- *Computation* - Virtual machines are used to accomplish the transcoding process of the videos. These services are charged on an hourly basis.

- *Storage* - Service providers offer various types of storages based on the bandwidth and the capacity requirements of the users. These servicers are charged mostly on a monthly basis.

Because of the rapidly increasing streaming content, video streaming companies are using cloud services to process and store the data. Since the streaming providers have to match the quality of video with the end-user devices based on their bandwidth and resolution, the video contents have to be compressed and stored in various supported formats. This transcoding process is performed offline to offer quick streaming services, and when users request the video, the suitable format is selected and sent to the user. As an example, Netflix does pre transcoding for all the videos and stores the same video in 70 formats [16].

The exhaustive and high computations involved in the transcoding process makes the video streaming companies choose cloud services to reduce the costs incurred. The cloud service charges its users only for what they have used. The cost of cloud storage is much cheaper compared to the cost of virtual machines doing computations. So, the videos are pre-transcoded and stored in the cloud. Again, to efficiently use cloud storage cost-effectively, many approaches and algorithms were proposed because most of the transcoded videos in storage are frequently not accessed. An algorithm is designed in [17] to decide whether to transcode the video and store in storage or to transcode it on demand, based on how frequently the video is streamed. It is called the hotness degree [17]. In this method, the popular videos are stored in the cloud, and the other videos are pre-transcoded partially based on the value of the hotness degree. It is vital to understand how this value is determined. The number of times a video is requested within a period is the access rate for the video. Based on past studies [18], the Group Of Pictures (GOPs) access distribution can be represented by Power law. Using this law, the access rate to the jth GOP in a video stream with access rate v_i can be determined as below. Here, alpha is the power coefficient with a value of 0.1 [17].

$$\varepsilon_{ij} = v_i * j^{-\alpha}$$

To measure the hotness of the video stream, first, the hotness of GOPs are calculated. A GOP is hot if the re-transcoding is costlier than the pre-transcoding. Now, if all the GOPs in a video stream are hot, they all have to be pre-transcoded, and so the video stream is also considered hot. Thus by measuring the hotness and deciding to pre or partially pre transcode the video, incurred cost is greatly reduced.

In another research [19], an algorithm is proposed using clustering of videos or GOPs based on the popularity and hierarchically storing them in the cloud. The main objective of the approach is to minimize the video streaming cost on the cloud. Though we develop mechanisms that store only the most accessed videos in the cloud, the cost of storing all those videos in the same storage space costs more. These popular videos can still be sub-categorized based on the view count and stored in a hierarchical structure within the cloud storage. For instance, Amazon web services offer multiple storage types - S3 standard, S3 standard - Infrequent Access (S3- standard-IA), S3 One Zone - IA, and S3

Glacier storage that depend on access bandwidth rates [19].

Since a video stream is segmented into multiple sequences that consist of many GOPs, the algorithm is developed at the GOP level. Each GOP consists of I, P, and B frames, and the transcoding is carried out independently at this level. When the GOPs in the video, its transcoding time, size, number of requests to the video, and the cost of the cloud storage are passed as input to the algorithm, it clusters the GOPs based on the frequency and saves them in storage. K-means clustering mechanism is applied to the GOPs that are pre-transcoded and uses the number of requests as criteria to decide the storage of each GOP. Thus the popular videos are further clustered into multiple groups based on the request count and are grouped into different storage buckets. For example, S3 standard storage which has higher access bandwidth and higher cost can be mapped to the cluster with the highest requested count. Subsequently, the other clusters of similar count videos can be mapped to other storage types, with the last cluster mapped to the storage type with lower bandwidth and lower cost. With this clustering approach, instead of storing all the videos in a single cloud storage type, they are distributed into multiple storage with lower charges, thus making this approach more cost-effective.

C. Caching

Thanks to the high growth in demand of over the top (OTT) video streaming services like YouTube, Netflix, Hulu, etc, the total bandwidth usage for these services has skyrocketed. The service providers not only have to cater to the huge amount of audience, but they also have to deliver the content to them on-demand and with high quality of service. Providing high quality of service in the terms of video on demand application means enabling the users to play the videos on-demand with minimal delay and jitter.

One of the mechanisms employed in order to reduce the delay faced by the end-user is caching. The multimedia content can be cached on strategically placed caches that are closer to the user. Thanks to these caches, the user does not need to request the top-level server for the multimedia content as the intermediate cache takes care of the client's requests. Although cache reduced the delay faced by the end-user significantly, the usefulness of the cache depends upon the caching mechanism and the decision of what to cache. As the multimedia Video on Demand providers like Netflix, Hulu, YouTube store the same multimedia content in different resolutions and quality at the same time, the cache replacement strategy becomes even more important. Below we discuss, compare, and contrast different approaches to caching the multimedia data and cache eviction policies.

An Announcement-based Caching Approach for Video-on-Demand Streaming

The paper [1] talks about using caches in intermediate locations nearer to the end-users to reduce the delay faced by the end-user to serve the multimedia content faster. The caching mechanisms used in the multimedia streaming systems is different from the conventional web caches, as the data caches in the conventional web systems is that of the images and web pages, which are used to load web pages. On the other hand, the multimedia data is made up of large file sizes. But an important point to note is that not all of the file needs to be present at the client to start playing the video stream. So the huge continuous file is segmented based on the timestamps so that they can be accessed, encoded, and decoded independently. This caching mechanism takes the advantage of the fact that when the user requests for the nth segment of the video, due to the nature of the multimedia video playback, we know that the

user will then request for the (n+1)th segment. Caching strategies can be further improved at a higher level. It is observed in the cinemablend survey[33] that average users of the streaming services like Netflix, Hulu stream 2.3, or more episodes of the TV show they are watching. This behavior is called binging and can be used to cache the data more efficiently.

Since the cache storage is limited, the data stored in the cache must be indeed improving the user experience and providing them a better quality of service. An important design consideration while designing a cache is the replacement policy of the cache. The authors use the terminology 'deadline' for the start time of the video playback. In the cache designed by the authors, each caching server decides which segments to keep cached and which segments should be evicted. Since the knowledge of deadlines is important in this caching mechanism, the caching servers get these deadlines either from the client based on the segment requests or these are determined by the server itself based on the client's previous requests. These two deadlines are kept in two different sets, namely Announced Deadline D_A and Perceived Deadlines D_P. When a client requests a segment of a multimedia video stream, the caching node then calculates the corresponding deadline for that segment to be cached and it is added into the announced deadlines set D_A. In an event when the node does not explicitly get the deadline from the client, it will store the perceived deadline in the perceived deadlines set D_P. The deadline gets expired when the streaming session ends. At this stage, the deadline is removed from the cache. Based on the values of the announced and perceived deadline information in the respective sets, each caching server makes a decision on whether to keep a particular segment or not. When a video segment is requested from the caching server, and if that segment is not cached the server has to decide if it wants to store it in the cache based on the deadline. If the available space is insufficient, a candidate segment is chosen for eviction based on the earliest announced reuse time. This candidate segment is evicted and the new segment is cached.

Popularity-aware Caching Algorithm for Video-on-Demand Delivery over Broadband Access Networks

Although caching is a very popular technique to improve the performance of Video on Demand services, and caching the most frequently viewed popular content near to the end-users will end up improving the overall quality of service, the authors of paper [3] point out that the popularity of any multimedia videos is dynamic and what is popular today might not be just as popular after a while[6]. The authors make a point that the caching algorithm needs to adapt to the changing popularity of the multimedia content over time[7]. The proposed algorithm in this paper is called the 'Last-k' algorithm and it determines the popularity of the movies using the time interval between the request for that movie.

The authors point out that the algorithms which decide the cache eviction policies based on the recency of use, like Least Recently Used(LRU) or Least Frequently Used(LFU) are not suitable for the distributed video on demand systems, because firstly, the multimedia files are humongous in size, so streaming one of these files to the cache takes longer compared to other web documents and images this makes the recency based algorithms unsuitable. And secondly, if the frequency-based caching algorithms are used, then it will take longer to evict the old and previously popular multimedia content from the cache which means the newly released viral multimedia content will take longer to be cached.

In the proposed 'Last-algorithm, the popularity of the movie is taken into account. The

author's way of finding out popularity is based on the time difference between 2 consecutive requests for that movie. So, the time difference between the requests of two popular movies should be less than compared to the time difference between the requests of two less popular movies. The figure below shows the pattern of requests for the movies.

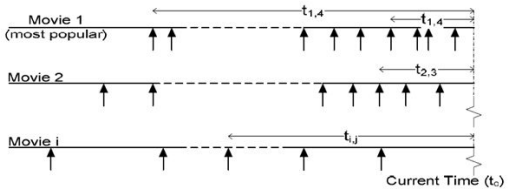

Fig. 2 Movie request arrival at VoD server [3]

In the above figure, $t_{i,j}$ is the time between the j^{th} request for the movie I, called the backward distance in this discussion. Let $T_{i,k}$ be the summation of last k backward distances such that:

$$T_{i,k} = \sum_{j=1}^{k} t_{i,j}$$

This $T_{i,k}$ denotes the popularity, such that when popularity decreases, $T_{i,k}$ increases. The k in this equation stands for the number of past requests taken into account for the calculation. As k is increased, the accuracy is also increased along with the processing overhead and response time. The value of k is selected so we can have the needed set of top movies, like top 10, top 50, and so on. The larger values of k result in unnecessary processing time and overhead. Finding the optimum value of k remains an open problem for future research.

The performance of the last-k and LFU algorithms were compared. In the time period when the popularity of movies was unchanged, both algorithms reported a similar hit ratio. However, after the introduction of new movies, the performance of both algorithms degraded. But between these two algorithms, the last-k algorithm was able to quickly adapt to the new movie's popularity compared to LFU.

D. Load Balancing

In this section of the paper, we present some of the works done in the area of load balancing algorithms for video on demand servers. In the previous sections we saw in detail as in how the video on demand uses most of the downstream bandwidth on the world wide web. So it has become increasingly important to develop techniques on the server side of things, to ensure Quality of Service(Qos) despite the increase in traffic. Not only this, but the advent of smart cellular phones has enforced a different requirement serving the video to a mobile device. This includes having a mobile compatible encoding to be delivered by the servers.

The content on the video on demand servers can be easily categorized into popular versus not so popular content. In a loose zipfian setting, 80% of traffic is for the 20% of the popular content, while on the other hand, the rest of the 20% traffic is towards the 80% of not so popular content. This overloads the servers holding popular content, thus, declining the quality of service. We surveyed some of the important ways of solving this disparity problem and we present the solutions discussed across some of them. We will then use our learning from these papers to help us evolve a hybrid load balancing technique in the end. Solving the above problem was just one part of solving the load balancing problems. If we see it another way, if we have our content available on all servers, how do we choose the server which will help us serve our request. We present the performance and analysis of some of the most popular algorithms for solving this problem namely randomized, genetic, fcfs and other heuristic algorithms.

Before we present the techniques, it's important to understand the nature of data

transfer for a video content over the network. Videos are composed of different types of frames which are sent over the network and played on the user side frame by frame. To maintain a constant movement of frames on the user side(refresh rate) we have to make sure that the video transmission happens with a constant bitrate. To serve the ever increasing requests for video on demand, a principle of replication of content can be used. This notion of replication can be picked up from distributed systems which employ replication for high availability of resources. Replication involves an added network overhead which will be talked about later.

Hot and Cold serves

Hot and cold as the names suggest, tell us about the amount of requests the server gets. Hot servers are loaded with tons of requests, as they host popular content. This means that they need more bandwidth to be able to serve their requests to the client. Due to lack of bandwidth, quality of service from these servers suffer. On the other hand, there are servers which are not utilized properly. These servers are cold which have their bandwidth under-utilized. There are various ways of solving this problem. One of the ways is horizontally scaling the servers with load but that poses to be a costly solution. We can replicate the popular content across the servers and based on previous traffic analysis which hit the servers, we can have a way of load balancing the incoming load. But this method is somewhat static and fails to work in a real time scenario. There is, thus, a need for load balancers which dynamically load balance traffic on the basis of the current server load and by analyzing the logs generated by them. In this paper, we summarize a dynamic content management system and a load balancing algorithm which works dynamically. These techniques have been discussed in [31] which talk about IShoot Content management system and the iSCON load balancing algorithm.

Makespan

The performance of a server is a sum total of a bunch of factors like the server configuration, the routing technique and the policy for queuing up tasks against a given server. To load balance a set of servers, we need to be able to have an idea of what amount of traffic is currently being served by the cluster, what kind of information we need to collect from servers to analyze it and some previous knowledge about the kind of traffic the server farm has already been processed in the past. Makespan is one of the most important ways to measure the performance of a cluster. It's the time that the load balancer took to complete a bunch of tasks assigned to it.[30] presents an equation to measure server load terms of 4 different variables which is a multiplication of average daily visitors, average page views, average page size, fudge factor and a constant 31.Average daily visitors as the name suggests is an expected number of visitors requesting video content from the servers. The average page size, average daily downloads and average file size are self explanatory metrics. Fudge factor helps us cushion our bandwidth estimation and is a real number greater than 1.

Modeling Server Performance

As there could be a number of ways of being able to model the performance of a server, we want to be able to formalize the server performance through the introduction of a couple of variables one of which has been described in detail above. Talking about a general scenario, for playing any video content, an html page holding the resource url of video content is asked for first. The video is then played, which further requests the server for the next frames of the video and streaming continues thereafter.

1. Makespan

As discussed above, makespan helps us understand the total time a server takes for serving a bunch of requests. Makespan helps us

understand the responsiveness of the server as the more the number of requests are served, the more the responsiveness of the server is.

2. Resource Utilization

For a single server, resource utilization is the ratio of idle time to the total time for a single server. For a server farm, it can be seen as just the addition of resource utilization for each of the servers. To be able to have the most resource utilization, we discuss a popular set of algorithms.

ISHOOT CMS and iSCON Algorithm

[31] presents a novel algorithm based on content replication policy which helps handle huge loads during high traffic. The architecture which the algorithm requires certain components which have been described below:

1. Server Selector

This the server where the requests first arrive at. It has the responsibility of taking a look at the system and ensuring that the request is routed to a server which has the most available bandwidth. This is the server which acts as a load balancer for the incoming requests.

2. Information Collector

For the server selector to be able to make any decision it needs information around the current state of the system. For this, we have a kernel level thread whose job is to fetch the current level of load on each of the servers. A record is maintained corresponding to each of the servers which are then referred by Server Selector. The server selector checks the load of the server farm every fixed unit of time, which is usually around 5 minutes or so.

3. Life cycle manager

This server helps the load balancer copy contents from one server to another. When the load balancer detects that some hot servers are unable to manage load, it then uses content replication policies and life cycle manager to replicate the content to some cold servers.

4. iSCON Algorithm

This load balancing algorithm mainly consists of 3 major steps, following which it is able to ensure quality of service of a highly loaded server. The server selector calls a function called reallocate, every chosen unit of time, which is largely set to five minutes. This function helps the server selector to see which servers are loaded beyond a certain threshold. The contents which are responsible for causing traffic are then replicated to another server with less load. These steps are repeatedly performed. Replication is performed through a function called replicate. After a period of time, when the server selector checks if replication is finished, post which it starts recalibrating the traffic to the new servers, to which the popular content was copied.

Popular Server Selection Algorithms

Server selection algorithms are important to Quality of service of video on demand servers due to many reasons. The way the load balancer decides to select a server from the server farm and schedule a task, decides how fast the server farm will respond to the requests, how scalable the architecture is and how well the load is actually balanced to help prevent hot servers from breaking down.

1. First Come First Serve

First come first serve algorithms serve requests in the order they arrive. This is the most naive way of load balancing a set of requests coming in because the decision does not involve any analysis regarding the nature of the request. So if there's a request which ends up taking a lot of time to be served, the requests lined up after that end up starving.

2. Random

Another way of handling requests could be randomly assigning tasks to available servers in the server farm. This proves to be better in handling the server load as we still have some way of distribution of requests across the farm.

Another challenge in this algorithm is to have a random number generator which produces numbers in a real random manner. If the numbers end up being really random, this algorithm ends up performing better than some of the best algorithms for server selection.

3. **Genetic**

Natural selection as seen in nature is one of the best ways of eliminating the weak entities and finally making sure that the fittest survive. Similarly, based on the history of the serving requests, the load balancer can learn to select a server which fits to serve a particular kind of request.

4. **Min Min**

The ideology behind the min min algorithm is that if we execute tasks in ascending order of their estimated time of execution it will be the most efficient. So this algorithm first estimates the execution time of incoming tasks. It then allocates these tasks to respective servers.

5. **Max Min**

The idea of this algorithm is to have resources which prioritize larger tasks over smaller tasks. When a bigger task arrives, it is served through these resources, while the smaller ones can be worked on parallely while a larger task executes. This helps to increase the throughput of the systems and prevents starvation of smaller requests.

E. Content Delivery Network (CDN)

Video streaming services serve several hundreds of customers simultaneously. These services can be delivered using numerous options like OTT (Netflix, HULU, Amazon, and YoutTube), live streaming content providers (CCTV, UUSee, etc.) [20], and so on with the deployment of robust content servers around the world. This is all done using the Internet.

As there is a thriving population seen in both videos and users, the most crucial problem faced by the content providers is how efficiently they can distribute the videos by meeting the streaming requirements at a low and affordable cost. And also, they must be served in time with no delays and utilize less bandwidth.

To handle this exploding nature of users and with the restriction to serve them with minimum cost, video streaming providers exercise more pressure on the delivery networks which are underlying and switch to use a large number of cache servers.

Replication of videos is done and is stored in these helper servers. When these servers are the customers themselves, then they frame a peer-to-peer mesh system. In this scenario, there is no need for video replication, but there are many challenges and questions regarding the availability of peers.

So, Content Delivery Network (CDN) based video-on-demand (VoD) can assist the customers to distribute the video stream data in a fast and reliable manner to their users and also provide these streaming services with QoE - quality of experience guaranteed [20].

Many more advantages are present if CDN is used, it offers better bandwidth and storage support and additionally provides security by defending them against ddos attacks.

Content Distribution Network is a network of geographically spread of several proxy servers and data centers [21]. They are connected using multiple fiber optic high speed internet lines delivering content at a faster rate along with the support of security. This network of machines run advanced algorithms so the routing of packets is done in an intelligent manner and thereby reducing network congestion and speed is increased.

The advantages of using CDN is the below [21]:

Global content delivery

In 2018, around 2.38 billion users use video streaming services and more usage of cellular devices was observed. When streaming using small servers there are high chances for latency and bufferings observed in videos with decreased quality. But with the help of CDN having a large distribution of servers in important regions, the content will reach the destination at rapid rates.

Technical outsourcing and cost savings

The technicalities to handle such load is taken care of CDNs and it relatably increases the savings of the clients. The economic cost and advanced smooth functioning methodology is a major plus.

Security and redundancy

Privacy of data is ensured and even hacking, digital attacks, DDoS attacks, MiTM attacks, ransomware and a lot more cyber crimes are taken care of by CDNs. It is estimated that by 2021, the cybercrime costs could be around $6 trillion.

Scalability with availability

Their huge availability of bandwidth is a major plus as it routes all the requests intelligently and manages abundant requests. The contents are also all time available as they use swappable devices which are hot and built in redundancy support is always present. If the content is not always available there could be loss of customers because of the startup delays.

Fast content loading and reduced buffering techniques

One of the most beneficial benefits of using CDN is fast content loading, In 2017 it is told that 63% of users said buffering issues were observed. By using CDN bottlenecks upstream can be avoided and also provides higher quality of the videos.

Detailed analysis of a few situations where usage of CDN plays a major role in VoD. Issues faced and how it is overcome with advanced techniques.

1. Internet Service Providers bill the VoD content providers for their bandwidth usage by calculating the 95 percentile rule [22]. This value is measured by calculating the average bandwidth used by the server for every 5 minutes in a month and putting them in a set. The smallest number from the set which is greater than 95% of the values in the same is the 95 percentile value.

As the user's demands vary dynamically based on time, the bandwidth utilization varies drastically and the utilization ratio would be low leading to less 95 percentile value than estimated. Even flash crowd provision would lead to an increase in cost.

This can be resolved by switching to content cloud platforms which use the “pay-as-you-go” format, and also reduces the management effort as well. This idea of hybrid cloud-assisted development is a better solution and is made of 4 parts: clients, servers, cloud storage and cloud CDN.

Cloud storage is used to store chunks of videos and then forward them to the CDN. Then these parts are delivered to customers and results in best performance. Payment is done only based on the number of bytes used. Even bursty traffic is handled by the CDNs at low cost by redirecting the requests which are additional to the cloud.

Since CDNs can store large contents and uploading them to the clouds is an increase in cost as more bandwidth is utilized, there is a need to control on what is to be uploaded and careful design of such migrations have to be considered. So to save money only the videos that are watched more frequently and that is predicted to be watched more are uploaded.

The simulation results show around 30% of the bandwidth expense is saved and also flash crowd is handled with little expense. The future work is to concentrate on building

Efficient migration strategies for both client/server VoD and P2P based VODs.

2. CDN approach, QoE for the user is guaranteed by the server which provides content. But the bandwidth cost is huge and there is a need to save it [20]. Earlier in P2P based VoD, coding schemes used was pure coding which is also called as pure chunk scheduling.

Now in a CDN-based approach the cost is reduced by combining coding and scheduling. This mixture can be satisfying various requirements on reducing the overhead which is caused by coding with more blocks and also reducing the additional complexity in the systems by modifying the size of the blocks. The network coding [23] has various benefits as this scheme enables content distribution in a hybrid environment of CDN and P2P systems. This type of coding can also reduce the redundancy used for storage in CDNs and save more bandwidth in P2P. Also they introduced a TCP based protocol for streaming by using this technique of network coding.

3. In addition to distributing the streams of video contents using CDN from content vendors to clients, CDN can also provide additional benefit of transcoding of the video in their own platforms. They are working on deploying elastic and cloud based transcoding platforms which are optimized [24]. Since there is a need to support the increasing demands of various types of users' platforms the video content is required to be transcoded in multiple formats for the ease of streaming.

Content delivery networks are highly pressurized because video streaming services need to provide the data in a short time interval without any delay and also require a high bandwidth to do the same. Streaming quality of the content is also required to be satisfied.

Transcoding of any video involves complex coding of the algorithms and they are highly computational. Since there is huge traffic of requests to transcode the data, CDN providing companies require cloud based platforms which are both elastic and optimized.

Cloud-ready transcoding with CDNs was built and it had the following efficient components: Ingesting cloud which is responsible for providing less response time and more capacity for the users who upload content to the system and Transcoding cloud which actually allocates and deallocates dynamically the transcoding participating nodes based on the burstiness of the traffic.

Videos are transcoded to various multiple bit-rates which can accommodate the dynamic network's nature. Operation cost is reduced and by coupling the unique properties of streaming videos. CDN based delivery to the end users saves a lot more transit cost. The prioritized task is Optimized enterprise-level transcoding and streaming using the models based on cloud computing and CDN.

4. The content distribution cost can be further reduced by combining P2P streaming with cloud based CDN. Locality-aware VoD streaming solutions in this hybrid environment gives the best trade-off between huge bandwidth consumption and not desirable internetwork ISP traffic present in the whole system [25].

A set of stochastic and optimization models are considered for calculating the demanded upload bandwidth. At first a loss network is considered and total bandwidth is derived for such huge networks under huge chunk distribution patterns amongst the peers.

Then the inter-ISP traffic is also calculated by doing some investigation to find the minimum cloud bandwidth required. When the permitted inter-ISP traffic is below the actual volume then the cloud bandwidth consumption increases linearly and there is also noted to be a decrease in the allowed inter-ISP traffic. Simulations are also performed and it states

realistic results and at the same time the performance of the designed protocol is observed.

5. Decentralization of delivery of the content is one of the famous solutions for saving bandwidth cost in CDN and is quite inexpensive. Clients are allowed to request the content of the video from the other clients present in the network and they are called the seeds. But there might be a seed scarcity problem [26]. This might be caused when there is a low count of seeds present which satisfies the request for a video resource.

So in order to reduce this problem many content providers have used a mechanism called video-push. This mechanism sends the latest video resources directly to some randomly selected seed so that could handle more requests. But currently, there is no mechanism to detect which video will be updated in near future and which video will become scarce or it also fails to identify the difference in uploading the seed capacity.

To reduce this problem, a new video push mechanism called proactive-push was introduced. This new mechanism lowers the CDN usage of bandwidth. It can predict the future scarce videos at beforehand and with much competing send to seeds with capabilities of stronger and higher uploading capabilities. Proactive-push mechanism will train the models in the neural network to exactly predict the future scarce video resources by 80% and also it can identify the competent seeds in the network upto 90%.

Real-world pilot-deployment and trace-driven emulation and [26] is evaluated compared to a VoD system which is commercial. Based on the observations from the results, this new mechanism can reduce the download percentage by 21% from CDN and also save upto 18% bandwidth cost at peak time.

6. When the collaboration between content delivery network and ISP has been done, it is shown to have tremendous results for both of them. It's very difficult to achieve state-of-the-art Dynamic Name Service based redirection [27] as the flows to be reassigned during the connected session is not an easy task to achieve as there can be flows reassignment during the active session and it's quite a difficult task to be achieved.

Varying surrogate loads which are caused because of the congestion events and flash crowds in the ISP network will become difficult to be handled. This collaboration between ISP and CDN [27] can be achieved using a deployment which is minimal and is on software-defined networking switches in the network of the ISP.

Huge volume of traffic flows between them at the backend is complemented using standard DNS redirection mechanisms even if it has HTTP state information. Although this approach seems to be a better fit there are oscillations between the previously present approaches and the current one. Migrations of TCP congestion are calculated in before-hand where varying topologies might cause a variation in total migration rates. Even flash crowd scenario ie redirected and handled.

Netflix and Hulu

Leading OTT content service providers are Netflix and Hulu in Canada and the United States [28]. Netflix is causing ~30% of the downstream traffic in the United states before a decade. Both the OTT platforms have a lot of third parties involved and use their infrastructures. Netflix uses Amazon cloud services like Cassandra, S3, Simple DB. Even Microsoft limelight is used by Netflix. Hulu uses Akamai majorly and even limelight and level3 CDNs.

It is identified that Netflix and Hulu delegate the request to the CDN irrespective of the type of request raised. And it is observed that the bandwidth available at the various locations differs and there is a need for

measurement-based [28] and multiple-CDN based video delivery strategy. This proposal can increase the average present bandwidth for each user. The end client's quality of experience should also not be compromised.

DASH protocol is used for serving Netflix's traffic. DASH stands for Dynamic Streaming over HTTP and is the protocol used for content streaming. Every video can be presented in various quality formats and these videos need to be encoded as well. To encode them an entire video stream is divided into smaller chunks and the client actually requests these chunks one by one rather than the entire video file using HTTP.

As and when the download happens the bandwidth requirements are calculated. Alongside a rate determination algorithm is performed for determining what should be the quality of the next requested chunk should be for a better performance. DASH also permits the user to freely and dynamically switch amongst various quality levels at the boundaries of each divided chunk.

RTMP is used by Hulu for delivering the movies to the clients which are the browsers in desktops. RTMP stands for Real Time Messaging Protocol and these movies delivered by Hulu are also encrypted.

Even those movies or videos can be delivered using RTMPT - RTMP tunneled over HTTP. Akamai and Limelight use RTMPT but when the tcp port is blocked all the CDNs perform RTMPT. Desktop video streaming services are provided with the help of similar technologies but iPad, iPhone and HuluPlus is provided with HTTP Live streaming technology.

Netflix selects multiple CDNs present for downloading a video with various qualities but Hulu selects only one CDN which has the content for all variety of quantities. And then it can switch between multiple CDNs for different content. For video playbacks, both OTT providers stick to the same CDN.

Netflix player continuously reports the heartbeat and logs by sending the requests at periodic intervals and this data differs from the actual data. Whereas in Hulu, a packet trace is done by sending the request containing all information regarding the client at that particular time. Query like whois is used to get their own IP addresses.

A small experiment is performed based on the playback of the video and it is observed that by selecting the best performing CDN amongst the three CDNs present improves by 12% the average bandwidth than the static CDN assignment and when all the 3 of them are used simultaneously then there is a spike of more than 50% improvement in the bandwidth usage. This method plays a major-role in the future development schemes when there is huge bandwidth demanded and is required.

A lot of study is performed on the larceny related issues in the video streaming services but none are related to the user interactions with the CDNs. Here the download speed is more prioritized than the HTTP. A framework on centralized control planes is suggested for selecting the CDN for clients and improving more QoE. Rather than the centralized solution, client side interaction initiation is analyzed for relatable betterments.

Also Netflix started its own CDN called Open Connect where ISPs can be directly connected with it. The design keeps changing and improving as OTT evolves and is decided to be more flexible and scalable with continuous changes and enhancements.

III. PROPOSED HYBRID MODEL

More research and studies have been conducted on VoD models in the past decade to improve the efficiency of each component individually. Based on the survey we suggest a hybrid model to combine selected approaches for each component. CVSS architecture can be used to pre-transcode and store the popular videos and

provides low cost on-demand transcoding of other video streams. For storing these videos, multiple HDD and SSD combinations can be deployed at the storage server. The pre-transcoded videos can be stored in SSDs to retrieve quickly and a portion of the SSD can be used as cache for the infrequent videos in HDD. Multiple cloud storages that vary in their bandwidths can also be leveraged to store the videos based on popularity.

Popularity based caching algorithm can be integrated to this model as it quickly adapts to changing popularity of the content and evicts the content from cache and SSD storage if it's no longer being watched as frequently. For balancing the load and to prevent from breaking down the hot servers, the Max Min Server selection algorithm can be used to prioritize the popular video tasks over the others. Here, when the popular tasks arrive, it uses the server resources, while the less popular ones can be parallely worked on as and when the former executes. This idea can help to increase the system's throughput and also prevent starvation of less popular tasks. Further, CDNs can be integrated with VoD by using a Video push mechanism where it sends the popular videos beforehand directly to randomly selected CDN servers which are handling huge loads of requests. These requests are responded with videos with less delay and hence maximum throughput is achieved.

Thus the above suggested model would result in providing high QoS for the clients and minimizes the cost incurred for the stream providers.

IV. CONCLUSION

This paper presents a generalized idea on how to develop a video streaming service which improves satisfaction of the customers with low cost and at the same time achieves high performance. After surveying a decade old research papers on Video on Demand, we have pinned down the following factors as the major contributing areas which have an influence on providing high QoE and QoS to the end-users. The factors include Transcoding, Storage, Caching, Load Balancing and CDN. We have analyzed the problems and solutions of the chosen research papers and tried to propose a hybrid model for achieving better VoD performance for popular videos by selecting optimal approaches from each of the above mentioned factors.

V. REFERENCES:

[1] M. Claeys, N. Bouten, D. De Vleeschauwer, W. Van Leekwijck, S. Latré and F. De Turck, "An announcement-based caching approach for video-on-demand streaming," 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, 2015, pp. 310-317, doi: 10.1109/CNSM.2015.7367376.

[2] X. Li, M. A. Salehi, M. Bayoumi and R. Buyya, "CVSS: A Cost-Efficient and QoS-Aware Video Streaming Using Cloud Services," 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Cartagena, 2016, pp. 106-115, doi: 10.1109/CCGrid.2016.49.

[3] C. Jayasundara, A. Nirmalathas, E. Wong and N. Nadarajah, "Popularity-Aware Caching Algorithm for Video-on-Demand Delivery over Broadband Access Networks," 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, Miami, FL, 2010, pp. 1-5, doi: 10.1109/GLOCOM.2010.5683976.

[4] F. Jokhio, T. Deneke, S. Lafond, and J. Lilius, "Analysis of video segmentation for spatial resolution reduction video transcoding," in Proceedings of IEEE International Symposium on Intelligent Signal Processing

and Communications Systems (ISPACS), pp. 1–6, 2011.

[5] X. Cheng, J. Liu, and C. Dale, “Understanding the characteristics of internet short video sharing: A youtube-based measurement study,” IEEE Transactions on Multimedia, vol. 15, no. 5, pp. 1184–1194, 2013.

[6] Hongliang Yu, Dongdong Zheng, Ben Y. Zhao, and Weimin Zheng. 2006. Understanding user behavior in large-scale video-on-demand systems. In Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006 (EuroSys '06). Association for Computing Machinery, New York, NY, USA, 333–344. DOI:https://doi.org/10.1145/1217935.1217968

[7] D. De Vleeschauwer and K. Laevens, "Performance of Caching Algorithms for IPTV On-Demand Services," in IEEE Transactions on Broadcasting, vol. 55, no. 2, pp. 491-501, June 2009, doi: 10.1109/TBC.2009.2015983.

[8] M. Ryu, H. Kim, and U. Ramachandran, "Impact of flash memory on video-on-demand storage: analysis of tradeoffs", *Proceedings of the second annual ACM Conference on Multimedia systems ACM*, pp. 175-86, Feb. 2011.

[9] O. A. Al-Wesabi and P. Sumari, "Hybrid Storage Architecture for Video on Demand Server", *The International Conference on Software Engineering & Computer Systems IEEE*, pp. 6-10, 2015.

[10] Al-wesabi, O.A., Sumari, P., Al-wesabi, M.A.: Efficient architecture for large-scale video on demand storage server. In: International Conference on Control Systems, Computing and Engineering (ICCSCE), pp. 395–400. IEEE (2015)

[11] SSD vs. HDD: Which Is Best? - https://www.intel.com/content/www/us/en/products/docs/memory-storage/solid-state-drives/ssd-vs-hdd.html

[12] S. Siewert and D. Nelson, "Solid State Drive applications in storage and embedded systems", *Intel Technology Journal*, vol. 13, pp. 29-53, 2009.

[13] SSD vs. HDD: What's the Difference? -https://www.pcmag.com/news/ssd-vs-hdd-whats-the-difference

[14] R. Manjunath and T. Xie, "Dynamic data replication on flash SSD assisted video-on-demand servers", *International Conference on Computing Networking and Communications (ICNC) IEEE*, pp. 502-6, Jan. 2012.

[15] Yu, Hongliang, Dongdong Zheng, Ben Y. Zhao, and Weimin Zheng. 2006. Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems April 18-21, 2006: Understanding User Behavior in Large-Scale Video-on-Demand Systems.

[16] Netflix talks at AWS - http://techblog.netflix.com/2012/12/videos-of-netflix-talks-at-aws-reinvent.html

[17] Mahmoud Darwich, Mohsen Amini Salehi, Ege Beyazit and Magdy Bayoumi, "Cost-Efficient Cloud-Based Video Streaming Through Measuring Hotness", *The Computer Journal*, vol. 62, no. 5, pp. 641-656, May 2019.

[18] Miranda, L.C., Santo, R.L. and Laender, A.H. (2013) Characterizing video access patterns in mainstream media portals. *Proceedings of the 22nd International Conference on World Wide Web*, Rio de Janeiro, Brazil, 13–27 May, pp. 1085–1092. ACM, New York, NY.

[19] M. Darwich, Y. Ismail, T. Darwich and M. Bayoumi, "Cost-Efficient Storage for On-Demand Video Streaming on Cloud," 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2020, pp. 1-4

[20] Y. Zhou, Y. Xu and S. Zhang, "Exploring Coding Benefits in CDN-Based VoD Systems," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 11, pp. 1969-1981, Nov. 2014, doi: 10.1109/TCSVT.2014.2321070.

[21] www.dacast.com - Top 5 benefits of using a live video CDN https://www.dacast.com/blog/top-5-benefits-of-using-a-live-video-cdn/

[22] H. Li, L. Zhong, J. Liu, B. Li and K. Xu, "Cost-Effective Partial Migration of VoD Services to Content Clouds," 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, 2011, pp. 203-210, doi: 10.1109/CLOUD.2011.41.

[23] K. Nguyen, T. Nguyen and S.-C. Cheung, "Video streaming with network coding", *J. Signal Process. Syst.*, vol. 59, no. 3, pp. 319-333, 2010.

[24] Z. Zhuang and C. Guo, "Building cloud-ready video transcoding system for Content Delivery Networks (CDNs)," 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, 2012, pp. 2048-2053, doi: 10.1109/GLOCOM.2012.6503417.

[25] Zhao, J., Wu, C. & Lin, X. Locality-aware streaming in hybrid P2P-cloud CDN systems. *Peer-to-Peer Netw. Appl.* **8,** 320–335 (2015). https://doi.org/10.1007/s12083-013-0233-3

[26] Y. Zhang *et al.*, "Proactive Video Push for Optimizing Bandwidth Consumption in Hybrid CDN-P2P VoD Systems," *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Honolulu, HI, 2018, pp. 2555-2563, doi: 10.1109/INFOCOM.2018.8485962.

[27] M. Wichtlhuber, R. Reinecke and D. Hausheer, "An SDN-Based CDN/ISP Collaboration Architecture for Managing High-Volume Flows," in IEEE Transactions on Network and Service Management, vol. 12, no. 1, pp. 48-60, March 2015, doi: 10.1109/TNSM.2015.2404792.

[28] V. K. Adhikari et al., "Measurement Study of Netflix, Hulu, and a Tale of Three CDNs," in IEEE/ACM Transactions on Networking, vol. 23, no. 6, pp. 1984-1997, Dec. 2015, doi: 10.1109/TNET.2014.2354262.

[29] https://www.akamai.com/us/en/resources/vod-cdn.jsp

[30] Sahoo, Bibhudatta & Prusty, Alok. (2013). Heuristics Load Balancing Algorithms for Video on Demand Servers. International Journal of Artificial Intelligent Systems and Machine Learning. 5.

[31] Kim, Junyeop & Won, Youjip. (2015). Dynamic load balancing for efficient video streaming service. 2015. 216-221. 10.1109/ICOIN.2015.7057885.

[32] https://truefilmproduction.com/study-predicts-84-internet-traffic-will-video-2020/

[33]Cinemablend-http://www.cinemablend.com/television/Unsurprising-Netflix-Survey-Indicates-People-Like-Binge-Watch-TV-61045.html