An Enhanced VoD Streaming Model for P2P Cloud Computing Systems

Mohammad H. Alshayeji[1], Yousef Sharaf[1], Sa'ed Abed[1] and M.D. Samrajesh[2]
[1]Computer Engineering Department, Kuwait University, Kuwait.
[2]Different Media, Kuwait.
m.alshayeji@ku.edu.kw

***Abstract*—Availability of ample digital multimedia content and advances in network bandwidth has fueled the growth of Video on Demand (VoD). Client-server based centralized system have limitations in the number of users who can get connected at a time. The growth of Peer-to-Peer (P2P) based VoD streaming systems, together with cloud computing offers better availability of the video content, improved scalability and load balancing among peers. Providing the end user a smooth VoD playback has become an essential quality assurance component of VoD based system. However, higher quality of service and better user experience during video playback is still an open challenge. In this paper, we propose an Enhanced VoD Streaming Model (EVSM) for P2P cloud based system. The proposed model has four layers where each layer considers the various quality assurance factors of the P2P VoD and ensures better video streaming. Our evaluation results show that the proposed model decreases the video seek time and lowers the startup delay when compared to other methods.**

***Index Terms*—Cloud Storage; Peer-to-Peer (P2P); Multimedia; Video on Demand (VoD).**

I. INTRODUCTION

The popularity of streaming videos through the Internet has significantly increased during the last decade. Traditionally, Video-on-Demand (VoD) systems were client-server based. Later, Content Delivery Networks (CDN) came into existence and it reduced the overall server load by distributing content and the load to various servers. Currently, cloud based Peer-to-Peer (P2P) VoD systems are getting popularity due to its scalability, availability and much more. Cloud-based multimedia system as in [1], P2P scheme as in [2], and a combined scheme of using the above as in [3-5] are some examples of cloud based P2P systems. Existing system such as in [6-8] have proved that cloud based P2P systems are effective solution for video streaming services. Several VoD based multimedia providers (such as Netflix) apply cloud storage for their services. However, providing better quality of service to end users is still a challenge [9].

A typical P2P-VoD system that consists of several nodes, servers, and a tracker. An efficient and scalable P2P VoD streaming system should support large number of users with minimal resources. However, designing such an effective system is a challenge [10]. One of the major issues in such systems is longer video playback startup time and large playback delays among the peers. Moreover, yet another issue in P2P-VoD system is poor and unstable video streaming quality when long-tailed unpopular videos are requested. Furthermore, some videos could vanish from the P2P system over time, hence maintaining them is an issue [11].

Several technical challenges and issues of video streaming over P2P networks are discussed in [12]. There are different types of P2P network, such as: tree-based, mesh-based, structured network, and unstructured network, or hybrid. Choosing the right type of the P2P networks is one of the challenge. The second issue is the heterogeneity of peers i.e. managing different capacity resources based on the content.

This paper presents a new video streaming model that can provide an efficient and scalable VoD streaming, using P2P and cloud server storage. The proposed Enhanced VoD Streaming Model (EVSM) for P2P cloud system consists of four hierarchical layers. The first layer is the server layer that assures the availability of the requested video. The second layer is the category-management layer used to arrange and manage the connection between the peers and server. The third layer is called the supervision layer used to manage the peers in order to serve the requesting nodes, and finally, the fourth layer is the streaming layer were the peer's streams videos.

Our main objective is to offer, end-user better quality of service in P2P-VoD streaming. The video startup delay and large playback lags among the peers is solved by grouping peers with similar interests into one category. Our evaluation results demonstrate the proposed model offers a lower startup delay and a fast video seek time.

The remaining of this paper is organized as follows: Section 2 presents the related work; Section 3 presents the proposed enhanced VoD streaming model for P2P cloud computing systems; Section 4 presents our evaluation and discussion and finally Section 5 has our conclusion and further research directions

II. RELATED WORK

P2P-VoD is an active area of research [13]; existing papers have proposed various techniques for enhancing P2P-VoD services. The current advances in networking has popularized interactive networked based applications. However, integrating different platforms such as P2P and cloud computing environments is currently a challenge [14].

Novasky is a VoD system that uses coding-aware and replacement strategy using Reed-Solomon codes [15]. It generates coded video segments instead of keeping original video segments. Novasky uses adaptive servers that depend on push-to-peer strategy instead of passive servers for enhancing video availability. Although Novasky system enhances the video availability, it occupies an additional storage for the coded segment.

A mesh-based P2P-VoD system architecture is proposed in [2]. The design is based on a dual spatially organized P2P network, where there are two overlays, one overlay for neighborhood observation, and the other one is for data distribution. The paper attempted to make the system an incentive by providing a plethora of service levels to users with several advantages for them in the system. However, system suffers from one major issue the least viewed videos will be removed by time, this affects future video availability.

A social relationships based approach for P2P-VoD is proposed in [16], the system structure has a new hierarchical overlay by exploring the existing social relations of users and the similarities of video channels on two different levels. The stress on the server is increased as new jobs are assigned to it. In addition, the user has to subscribe to channels to get the services.

A unifying model consisting of replication and scheduling of content in P2P VoD is presented in [17]. The proposed model combines request scheduling, such that maximum number of peers can be served in a single request. The scheduling is called Fair Sharing with Bounded Out-Degree (FSBD). The proposed system aims to balance load among peers and minimize the discrepancy of load between peers. It uses both centralized content assignment algorithms, as well as adaptive content replication algorithms. However, its optimal solution does not balance the load totally.

Our proposed Enhanced VoD Streaming Model for P2P cloud system (EVSM) requires no additional amount of storage for the coded segment in the cache. It reduces the number of servers by transferring few of jobs directly to the participated peers. Furthermore, our approach attempts to minimize the bottlenecks at the server.

III. THE PROPOSED MODEL

The proposed EVSM for P2P cloud system consists of four layers as shown in Figure 1. The first layer has the servers of the P2P VoD system; the second layer is the category management layer. The third layer consists of the supervision layer, and finally the fourth layer is the streaming layer, which has a P2P cloud-based storage on a virtual space [18]. The peers are grouped together in each layer based on their function.

A. The layers of EVSM

Figure 1 shows the proposed model's diagram and the interactions between its layers and components. Arrow labeled 1 shows the interaction between the servers and the Master Management Node (MMN) for managing the requested videos. Arrow labeled 2 represents the interaction between peers and MMN for managing the P2P cloud storage. The arrows labeled 3 and 4, the video chunks are being pushed through the layers to its cluster (category). Arrow labeled 5 represents the communication and video chunks distribution among the nodes in the same category.

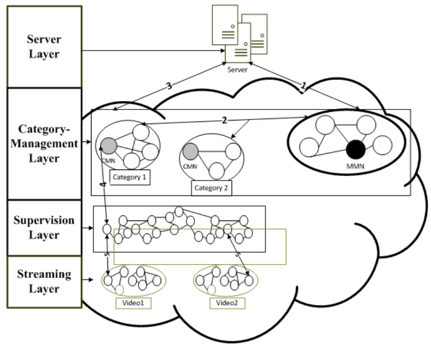

Figure 1: The layers of our proposed model

a. The server layer

The media servers remain in this layer. The servers cooperate with the P2P cloud storage in the second layer to provide videos to large number of peers, depending on the video category and requests. The goal of these servers is to insure content availability of the rarely requested videos and maintain service quality.

b. The category-management layer

A cloud-based storage consists of the Management Nodes (MNs) that have different functions:

- Maintain video metadata and tables in all nodes, duplication more than one nodes such as to prevent single point of failure.
- Group MNs with highest performance, and choose one node among them to be the Master Management Node (MMN) for making decisions.
- Group other nodes near each other for managing specified category of videos having their own master node Cluster Management Node (CMN). Their roles will push new videos in this category to the lower layer, and also receive requests from lower layer's nodes, and assist the MN if required.
- Re-organize the MNs in the groups and update the tables once a critical MN departs the system.

c. The supervision layer

The nodes in this layer will contain completed videos (or sub parts for some videos). This layer's role is to send the requested chunks (from the nodes in this layer or the above layer) to the nodes on the streaming layer. The nodes in this layer will be waiting for new videos in the area (cluster) that the user prefers to watch. The nodes are grouped and connected to each other depending on different classification (such as: Age / Language / Video Quality (3D/HD/LD)). These classifications is used for categories that have high popularity to reduce the user's startup delay by inserting its node near to its regular demands. Each node will have its own cache memory and the node will save the last viewed chunks by the user.

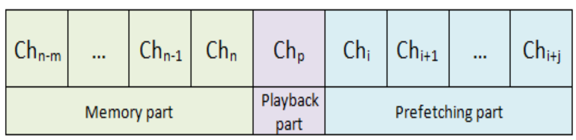

Figure 2: The node cache

d. The streaming layer

Every node's cache will have multiple chunks with fixed size. The cache will be used for prefetching, playback, and storing the played parts. A sample of node cache is shown in Figure 2 where Ch_n is the last seen chunk, Ch_{n-m} is the first available chunk in the cache, Ch_p is the playback chunk, Ch_i is the next chunk to be played, and Ch_{i+j} is the last prefetched chunk in the cache. The nodes cache will be associated with three specific roles: (1) Serve data to the neighborhood's nodes in this layer (2) Request data from a peer in the supervision layer (or category-management layer in some cases) (3) Observe other neighborhoods.

B. Video Chunk Scheduling Algorithms

The main idea behind our algorithm is to place video in peers in the best position for sending and receiving the requested videos. Important notations used include. *T:*Time step, *S*:Number of servers, *Nij*: Node 'j' for user 'i ', MN_i: Management node for user 'i', C_k :Category 'k'. *CMNk*: Cluster Management Node for category 'k',V_{kl}: Video number l in category 'k', N_k Nodes in category *Ck.*

The proposed dynamic scheduling algorithm is given in Algorithm 1. For each time step '*t',* the algorithm iterates on '*t'* to reallocate the nodes in the system, to adopt with the change in the system. The highest performing peer (cache size, bandwidth, and participated time) among MNs is selected as the Master Management Node (MMN) in the system. After that, the system will organize the categories into clusters, and each cluster will assign its own master node (Cluster Management Node (CMN)). Algorithm 2 adds the node into a category.

	Algorithm 1: Scheduling and positioning new peers Entering the system
1:	**for** t = 1, 2, … **do**
2:	assuming we have running servers on the system
3:	**Input:** N_{ij} and MN_i entering into the system
4:	**if** (new $user_i$) **then**
5:	add 1 MN_i
6:	**end if**
7:	**if** (t = 1 or idle status) **then**
8:	choose highest performance MN to be MMN
9:	organize C groups and assign CMN_k for each group
10:	**end if**
11:	assign N_{ij} in the category by calling Algorithm 2
12:	**end for**

Algorithm 1: Scheduling and positioning

The algorithm checks if it is the first node in the category. If so, the system will go through lines 4 to 10 and perform the following: 1) The first node in the category will be inserted in the third layer (supervision layer). 2) Communication will be established between the node in the third layer and the servers. 3) The node will inquire its category nodes in the above layer (second layer). 4) The CMN for this category will inquire the MMN, 5). The MMN will check the availability of the video; if it finds it will secure a connection between the servers and the CMN. 6) The CMN (or one of its neighbors) will send the requested video to the requested node in the third layer, and now, the peer can watch the requested video.

Algorithm 2 is invoked for assigning user's nodes in the selected category. The else part represents cases where there are other nodes in the category. At line 12, the node with the requested video will inquire the nodes in the supervision layer (third layer) for the video. In lines 13 and 14, if the video is found, the node will be inserted in the streaming layer (fourth layer), and the video will be sent to the node from its neighbors (if available), or from the node in the above layer. In lines 15 and 16, if the video is not found in the category of the streaming layer, steps (4) to (10) in the algorithm will be repeated. Once the peer finishes watching the video, it will be removed from the streaming layer to the supervision layer. Therefore, the peer waiting for another video request in this category. It keeps serving other peers in the streaming layer when request arrive, if none it terminates.

IV. EVALUATION AND DISCUSSION

A. Simulation setup

We evaluate our proposed model with existing system using two parameters under varied number of videos. The proposed model and comparative algorithms are coded and implemented using Java programing language using Eclipse SDK. The simulation setup values are shown in Table I. Video content size is 2-10 chunks, the initial number of videos is set to 10, numbers of video categories are 5. The initial number of peers in the system is 100. The cache size for peers is set between 2-10 chunks. The upload capacity of each peer is in the range of [0-3] chunks.

Table 1
Simulation setup

Video content size	2-10 chunks
Initial number of Videos	10
Number of Categories	5
Initial number of Peers	100
Cache size for peers	2-10 chunks
Upload capacity of each peer	[0-3] chunks
Download capacity of each peer	[1-3] chunks
Videos added in each iteration	5

B. Evaluation Parameters

a. Video seek time

The time taken to locate the peer requested video and all its chunks for download is considered as the video seek time. Higher time represents the non-availability of the videos chunks in the neighboring peers.

b. Startup delay

Startup delay is the time delay between the moment the peer requests the video (e.g. user clicks the on "play" button) and moment until the buffer is sufficiently filled for video playback. This delay is also influenced by the network bandwidth and the video quality etc.

C. Discussions

The proposed model is compared against two existing models 1) Novasky 19 uses an adaptive scheme that pushes content to peers in the P2P storage cloud and 2) Random Peer Selection (RPS) as in [20] where a random selection of peers is done based on content availability, this algorithm is used by most P2P systems. Figure 3 shows the evaluation of video seek time EVSM reduced the searching time between 9.5% (10 videos) initially to 61.7% (35 videos) when compared to Novasky.

Algorithm 2: inserting users node into categories

1: **for** *each* $N_{ij} \in user_i$ **do**
2: assuming we have k categories
3: **if** (N_{ij} is the 1st N in C_k) **then**
4: N_{ij} asks MNs_k for V_{kl}
5: CMN_k asks MMN
6: **if** ($V_{kl} \in S$) **then**
7: send V_{kl} to N_{ij}
8: **else**
9: send message "Video is not available"
10: **end if**
11: **else**
12: N_{ij} ask Ns_k in C_k for V_{kl}
13: **if** ($V_{kl} \in Ns_k$) **then**
14: send V_{kl} to N_{ij}
15: **else**
16: repeat steps (4) to (10)
17: **end if**
18 **end if**
19: **end for**

Algorithm 2: Inserting node

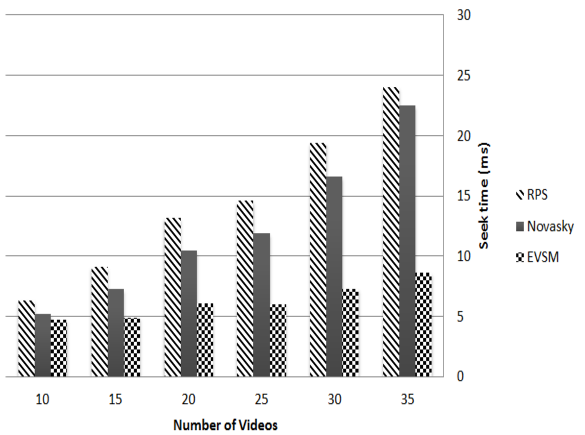

Figure 3: Evaluation of video seek time

In EVSM in order to find a specific video, the process is done in the Management Nodes (MNs) instead of the video servers. Hence, the video seek time is considerably less. Moreover, the supervisory layer ensures the availability of video in least possible time. EVSM when compared to Novasky at 10, 35 number of servers, the video seek time is less by 9% and 61%, respectively. Figure 4. Shows the evaluation of startup delay under different number of videos. EVSM when compared to RPS, Novasky at 25 number of serves the startup delay is less by 67% and 64%, respectively.

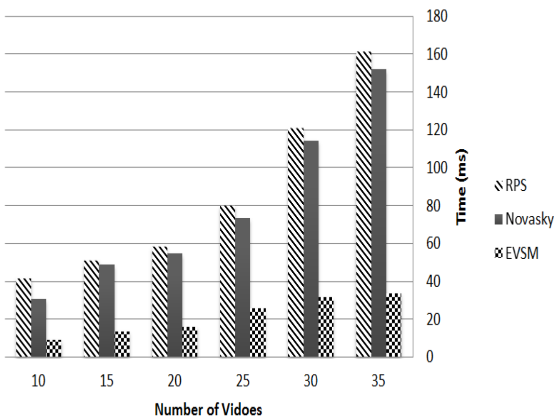

Figure 4: Evaluation of startup delay

V. Conclusion and Future Work

In this paper, we presented a multi-layered Enhanced scalable VoD streaming model (EVSM). The participating nodes are categorized based on the services the peer contributes. The proposed Model consists of four hierarchical layers. The layers include the server layer which assures the availability of the requested videos. The category-management layer used to arrange and manage the connection between the peers and server. The supervision layer used to manage the peers in order to serve the requesting nodes, and finally, the streaming layer is where the peers streams videos.

EVSM requires no additional amount of storage for the coded segment in the cache and better resource utilization in the peers and the servers. It reduces the number of servers by transferring some of the server jobs directly to the participated peers. Our simulation results show that the proposed model has a reduced startup delay and video seek time. One known limitation is management of multiple layers, however, some tasks are performed in the background. In future, we plan to perform more tests under different scenarios, and use different video bitrates.

References

[1] Lin, C.; Chin, H.; Deng D., "Dynamic multiservice load balancing in cloud-based multimedia system," IEEE on Systems Journal, no. 1, pp. 225-234, 2014.

[2] Maha Abdallah; Romain Cavagna; Damien Laval, "Incentive-based on-demand video streaming using a dual spatially-organized peer-to-peer network," In 12th Annual IEEE on Consumer Communications and Networking Conference (CCNC), pp. 689-695, 2015.

[3] Hung-Chang Hsiao; Hsueh-Yi Chung; Haiying Shen; Yu-Chang Chao, "Load rebalancing for distributed file systems in clouds." IEEE Transactions on Parallel and Distributed Systems, vol. 24 pp. 951-962, 2013.

[4] Chunfeng Yang; Yipeng Zhou; Liang Chen; Tom ZJ Fu; Dah Ming Chiu, "Turbocharged video distribution via P2P," IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, pp. 287-299, 2015.

[5] Chourouk Hammami; Achraf Gazdar; Imen Jemili; Abdelfettah Belghith, "Study of VOD streaming on BitTorrent," In International Symposium on Networks, Computers and Communications (ISNCC), pp. 1-6, 2015

[6] Shadi Ibrahim; Bingsheng He; and Hai Jin, "Towards pay-as-you-consume cloud computing," In IEEE International Conference on Services Computing (SCC), pp. 370-377, 2011.

[7] 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," In Proceedings IEEE on INFOCOM, pp. 1620-1628, 2012.

[8] Ruben Torres; Alessandro Finamore; Jin Ryong Kim; Marco Mellia; Maurizio M. Munafo; and Sanjay Rao, "Dissecting video server selection strategies in the youtube cdn," In Distributed Computing Systems (ICDCS), pp. 248-257, 2011.

[9] Barba-Jimenez, Carlos, Raul Ramirez-Velarde, Andrei Tchernykh, Ramón Rodríguez-Dagnino, Juan Nolazco-Flores, and Raul Perez-Cazares. "Cloud Based Video-on-Demand Service Model Ensuring Quality of Service and Scalability." Journal of Network and Computer Applications, pp. 102–113, 2016.

[10] Xu, Ke, Xin Liu, Zhen Ma, Yifeng Zhong, and Wenlong Chen. "Exploring the policy selection of the P2P VoD system: A simulation-based research." Peer-to-Peer Networking and Applications 8, no. 3, pp. 459-473, 2015

[11] Wang, Zhi, Jiangchuan Liu, and Wenwu Zhu. "Propagation-Based Social Video Content Replication." In Social Video Content Delivery, Springer International Publishing, pp. 35-47, 2016.

[12] Naeem Ramzan; Hyunggon Park; Ebroul Izquierdo. "Video streaming over P2P networks: Challenges and opportunities," on

Signal Processing: Image Communication 27, no. 5, pp. 401-411, 2012.

[13] Brienza, Simone, Sena Efsun Cebeci, Seyed Saeid Masoumzadeh, Helmut Hlavacs, Öznur Özkasap, and Giuseppe Anastasi. "A survey on energy efficiency in P2P systems: File distribution, content streaming, and epidemics." ACM Computing Surveys (CSUR) 48, no. 3 , pp. 1-37, 2016

[14] Díaz, Manuel, Cristian Martín, and Bartolomé Rubio. "State-of-the-art, challenges, and open issues in the integration of internet of things and cloud computing." Journal of Network and Computer Applications 67, pp. 99-117, 2016.

[15] Liu, F.; Shen, S.; Li, Bo; Li, Baochun; Jin, H., "Cinematic-quality VoD in a p2p storage cloud: design, implementation and measurements," IEEE Journal on Selected Areas in Communications, no. 9 pp. 214-226, 2013.

[16] Wei Chang; Jie Wu, "Social VoD: A Social Feature-Based P2P System," In 44th International Conference on IEEE on Parallel Processing (ICPP), pp. 570-579, 2015.

[17] Zhou, Yipeng, Tom ZJ Fu, and Dah Ming Chiu. "A unifying model and analysis of P2P VoD replication and scheduling." IEEE/ACM Transactions on Networking (TON) 23, no. 4, pp. 1163-1175, 2015

[18] Eliya B.; Abdallah M., “Efficient triangulation for p2p networked virtual environments,” In Multimedia Tools and Applications Journal - Special Issue on Massively Multiplayer Online Games (MTAP-MMO), vol. 45, no. 1-3, pp. 291–312, 2009.

[19] Liu, F., Shen, S., Li, B., Li, B., Yin, H. and Li, S., 2011, April. Novasky: Cinematic-quality VoD in a P2P storage cloud. In Proceedings IEEE INFOCOM, pp. 936-944, 2011

[20] Yang, Xiaoyuan, Minas Gjoka, Parminder Chhabra, Athina Markopoulou, and Pablo Rodriguez. "Kangaroo: video seeking in P2P systems." In 8th International Workshop on Peer-to-Peer Systems (IPTPS), pp. 260-275, 2009.