An Efficient Design and Analysis of Large-Scale Video-on-Demand Advantage from User's Support

P.Vaisagan[1] L.Vinothini[2] S.Sebaatini[3]
[1]Assistant Professor
[1,2,3]Department of Information Technology
[1,2,3]Jeppiaar Engineering College, Chennai

Abstract— *For the perfect service in peer-assisted Video-on-Demand systems, we propose an systematic structure to define the scaling laws for the extra bandwidth that servers should contribute, by considering the significant features such as bandwidth heterogeneity, peer churn, zipf-like videos. This leads to the vast effect in system performance due to the distribution of content popularity and size of catalog. We provide an attractive solution to control the cost induced by content providers and to limit the burden on servers for vast range of system performance by the users' cooperation.*
Keywords— Cooperative networking, Video-on-Demand

I. INTRODUCTION

The traffic module is predicted to be increase in 2016 because of Video-on-Demand. The video providers are forced to frequently improve the content delivery network (CDN) due to traffic volumes, which provides contents to ISPs. A recent drift of VoD providers to utilize cloud services has relatively reduced this burden by resource allocation [1]. Although, in terms of scalability any solution depends on CDNs has critical restraint. The peer-to-peer model is the barely scalable solution preferred until now, in which the resources are provided by the users [2], [3]. Although the peer-to-peer model is the best solution for scalability issues, it leads to several problems, which has been tested in many applications[4] - [7].Because of these reasons we defend that cloud services must be supported by peer assisted models, that interfere to support the current demand, when the user's resources are not enough. In theory, we particularly focus on specifying the bandwidth in addition for the perfect services to users that should guarantee by the servers. Our study allows to firmly specify the system performance as the number of available videos and user's increase, the large-scale VoD scalability utilizing users' cooperation is evaluated. We experimented using single video in previous work [8], by providing asymptotic characterization to the bandwidth of servers, as the number of user's increases. In this, multivideo system is used in which users can search the videos in catalog and give request simultaneously and can watch the videos... We assess the cases in which users can only serve the dispensation of the previously selected videos. This is called passive system because it depends on video popularity and it is not under the control of system. In the active system, we also frame the allocation of resources that allows attain the theoretical performance. When users leave the system after watching certain videos, we come to the conclusion that users can able to upload the video they downloaded previously. The unpopular videos are rarely duplicated among the peers for the same reason. So, the additional contribution of the work to reduce the joint impact of peer churn and video popularity.

II. PRESUMPTIONS OF SYSTEM

A. Users Cooperation and Service Identification

The users run application by modeling a VoD system which permit the users to select the videos from the catalog. We suppose that request is satisfied as soon as the video is choose and watched without any issues till the end. i.e., the user is provide with greater data flow which provides steady playback of video. We consider that the many different videos are available in the system catalog and it is characterized based on its selection probability, size and its bytes. We assume that at least one video is watched at a time videos. Users provide their available bandwidth for the distribution of video: The part of the video or full from the requested video can be recovered from the another users by saving resources of servers. The storage capacity of the system provided by the users are limited to a particular level. The size of the buffer is not considered in our analysis. We consider that at most one video is stored in addition to the playing video at present.

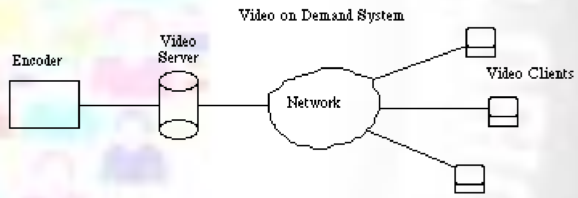

Fig. 1: Video On demand system

B. User Dynamics

We represent arrival rate of users when they issue request for the first video and joins the system. The users can be in two states: providing phase is defined as in which user contribute their upload bandwidth to the system. In this state user can watch the videos by downloading and notice that upload bandwidth is contributed even though user does not watch any videos because it keeps its VoD application open and in the running state. In the sleeping state, it is neither downloading nor uploading videos because user's application is idle. Assume that the users download the complete required videos. The users stay in the providing state for specific time after the downloading is completed and then move to sleeping state where they remain for any amount of time. The VoD application can be stopped after watching a particular video. The average time and the total time spent by the user in the system is calculated both in contributing and sleeping state which changes from user to user.

C. System Scaling

As the system grows the servers should guarantee for the complete service for all the users by asymptotically characterizing average bandwidth. Since the size of the catalog is supposed to increase, like the users, we assume that each video is tied to the number of users. The new videos are made available to the users when the system grows. The characteristics of the new inserted videos in the catalog based on the download rate and the size of the file.

D. Video Popularity

The selection probability of particular videos is to be specified, so in the catalog frame the relative popularity of the videos. For this purpose zipf's law is used to observe the traffic and how widely the performance is adopted [9], [10]. In this work, the development of the content popularity has been ignored above time timescale principle. i.e., dynamics of popularity at a timescale is slower than contents' download time, which is hardly acceptable in case practical work.

E. System Load

The load prompt by all the videos in the given system catalog can be computed. Consider a large interval time, a video will be requested for maximum amount of times, during this large interval. The requested video has effect on the system. It needs an amount of bytes to download and upload data. The ratio between the amount of data downloaded and uploaded considered in both passive and active system. Although, in the active systems, the extra data downloaded by the system when the user is instructed does not taken into account. (Data bundling) and the effect of bundling is considered later by the system load.

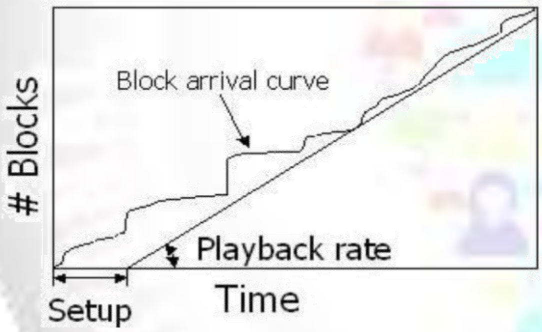

Fig. 2: playing rate of video

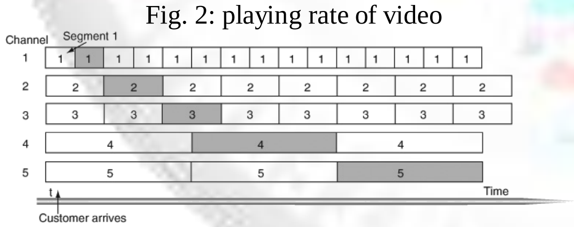

Fig. 3: Videos in segment

III. PASSIVE SYSTEM

In the passive system, users are able to distribute only the last selected video. This denotes when requesting a video, they download/ upload data only belonging to that selected video. This system is easy to manage and implement, it can control the videos independent of each other, since large number of different videos are decoupled.

A. Single-video Systems' Asymptotic Results:

Here we analyze the system in an easy way because there is just one video, we want to support the distribution of all the videos in the catalog before the servers request for the bandwidth. (i.e.,) the request rate of single video tends to be infinite. When the request rate remain finite or becomes invisible when number of users grows large, the pessimistic assumption provide the rough bound that has the entire bandwidth needed to sustain the downloads given by the servers.

B. Multi-video systems' Asymptotic Results:

The results of the single video system used to calculate the aggregate bandwidth needed for the servers in the multi-video systems. We obtain the upper bound on the overall bandwidth requested from the servers by adding the contributions of the individual video. In multi-video system we divide the video catalog into two parts: 1) the request rate of the videos tends to infinite was comprised as hottest portion of the catalog 2)the request rate remains unchanged or vanishes was comprised as coldest portion of the catalog. In former portion we provide tight bound to the bandwidth requested from the servers. In latter portion we limit ourselves to apply the bound to the bandwidth. The server requested for the aggregate bandwidth of the cold videos was captured correctly by the bound. Because, peer assistance was rather in effective for every cold videos (in passive system). In a given time either the seeds are sleeping or it does not support the distribution of the video, when average number of available seeds for any cold videos keep bounded. The server requires the minimum amount of bandwidth to maintain the distribution of the video. Finally, when the number of simultaneous downloaders is finite then the bandwidth fraction needed by the peers also finite. Since the users are limited to provide their complete upload bandwidth for the last requested video, hence this scheme was easy to implement and manage and it was a rigid scheme, by providing their bandwidth the resources of the users are not fully utilized. In section IV, by the guidance and analysis gained from the passive systems, we will discuss the active systems performance.

IV. ACTIVE SYSTEMS

We start considering the active systems, users can be able to download/upload data belonging to any videos, with the following constraints: 1) the videos which they want to watch must be downloaded;2)the data which was previously downloaded can be updated. In some cases, chunks or stripes which are not belonging to the requested videos can be upload or download by the users (data bundling). However we are not discuss about the extreme cases, that stripes/chunks can be very small, the size of chunk/stripes can go to zero, because it is not practically implemented. We use that with restriction (size of chunks/stripes cannot go to zero).The limitations of passive system was overcome by the skillful strategies. For this, we want to solve two problems: 1) the scaling of passive system was prevented by the presence of videos in deficit mode (load balancing). 2) The presence of cold videos, that are detrimental to the system (catalog warming). For both problems, here we present many solutions that gives basic idea about how they work and the main results. We expect that, when we solve above two problems, it will be an easy task to compute the system performance. Before us explaining the proposed techniques, we need to emphasize that, for the active system, the open network of infinite server queues are used to describe the dynamics of users that become a BCMP. The above discussed queue network is more complex than the passive case and the number of possible states in the queue is larger than the passive case.

A. Seed Reallocated Load Balancing:

The aim of the video equalization is to make the loads incurred by global system corresponds to the individual video load (4). The easiest way to reach this goal is to take away the condition that peers act as a seed for last downloaded videos only. Here, we denote that seeds are not needed for the videos with small video rate, rather the videos which had long duration and larger in size are very difficult. The technique which want to show their good performance by concentrate on the seeds among the videos which are more difficult. This could be achieved when peers offering the previously downloaded video content to the other servers. To execute this technique, the peers need to store the exact content for which information they act as a seed. The passive system videos are allocated with extra seeds, but the approach is simple. The performance of this technique was constrained clearly hence finite number of videos only downloaded by the users before they leave the system, so they cannot act as seeds for any videos. Similarly videos may cause wastage of bandwidth yet that may be reduced from the larger bandwidth resources of other videos. Notice that when we subtract the video load with the seeds it becomes too small. It has only one solution the natural download rate was increased artificially, decreasing the timing of download. Because of that, we had an effect that average time of the uploading was increased after downloading the video, thereby the download speed was increased. This technique works as: 1) videos are downloaded at the same speed; so we can decrease the time 2) peers are divided into two categories: seeds allocated to particular video. Instead of act as the seed for last downloaded video, seeds allocated to videos for their residual lifetime in the system.

B. Stripe Bundled Load Balancing:

Stripe bundling technique is based on the idea: Every video was fragmented into stripes (called as sub-streams), when the user requesting the video that was downloaded in parallel by a user and the sub-streams that was transmitted is reassembled by the decoder. When user downloading a certain video which are also forced to download a stripe of another video in a fraction and a lot the remaining bandwidth to upload the video with respect to the bundled stripe. In the intense state, when the peers download a single video it leave the system, after that it cannot support any other video distribution, nevertheless we force them to download at least one piece of the another video when they enter into the system. We minimize the bundled data by dividing them into number of stripes that are downloading at the stable state. The stripes are considered as the full video and are stored in the video catalog as being comprised by independent contents to the set of all the stripes. Users gather the stripes of the requested video in parallel and a lot a same fraction of upload bandwidth for every stripe, because of that load of each stripe is equal to the load associated with the entire video. When the downloaded video had some excess bandwidth, immediately peer request the video and start downloading the stripe of the video later that it permit the fraction of the upload bandwidth for the each stripe of the requested video, then the remaining was used for distribution. However, the cost of the bundling was considered as the important factor, we describe that increasing cost was become very small by increasing the number of stripes by using the changed version of the bundled scheme. This technique was executed by: 1) the videos which have the maximum video rate was downloaded; 2) make the seeding video in a predetermined/coordinated fashion. The implementation of this mechanism quite complex because it need both video striping and bundling techniques. On the other side, stripe bundling becomes more effective before vanishes from the system.

C. Video Bundled Catalog Warming:

When analyzing the passive system, we gain knowledge about the cold videos, which are the important component of the bandwidth requested from the servers. Hence we need to get an important reduction of, when we increase the request rate of cold videos artificially. To make some peers in the contributing state when they are not downloading any other content, download the entire (unrequested) videos was the easiest approach to reach that goal. The mechanism that described above would not need chunking/stripping video and can be given by load balancing strategy explained in IV-1. When the peer goes into the sleeping state, the downloading of the bundled video is interrupted, if the peer restarts the contribution to the system the downloading was promptly resumed, without downloading any other videos. This strategy has the negative effect when the load induced by the hot-videos (the videos whose request rate was not increased) becomes negligible for number of users becomes infinite. In our strategy the request rate of the videos are artificially increased and has the potential effect on that load of videos. The bandwidth needed to download the videos was less than the average of upload bandwidth of the seeds. When this condition was reached with the probability 1, it is used in the sufficiently large system, without making any video in the deficit mode.

D. Chunk Bundled Catalog Warming:

The above technique urge again a limitation on the system parameters. When it is neglected, the similar approach was applied to the every pieces of cold videos (chunks) instead of entire videos with restriction. The peers, which are made to contribute to a video which was not requested by the users neither want to retrieve it completely nor to sequentially download it. We ask some peers to download a randomly chosen chunks contributing to the distribution of the cold videos, which was cut into chunks. Bandwidth consumes by every artificial downloader reduced by the factor called Chunkization , its potential contribution on the upload was kept constant

E. Stripe Bundled Catalog Warming:

A same idea was applied to stripes instead of chunks and it was coincide to the load balancing technique explained in Section IV-2. Primarily, hot-videos requested by the peers with certain probability are also forced to download the constant size cold videos. We need to compute the impact of the load incurred by individual videos of the proposed technique with the limitation of the load less than 1.finally we can see that when size of the system increases, the load increased for the hot-videos disappeared. This scheme also applied in sufficiently larger system.

F. Implementations Issues:

We highlight that existing technique incur very different in terms of the difficulty and overhead to achieve load balancing and catalog warming. In load balancing by seed relocation technique peers want to decide that which content acts as the seed (remaining that it follows passive system). The peers receive control messages periodically and overview the downloading process by one or more servers that information was comprised in a centralized manner. Load balancing by stripe bundling needs the system to support video stripping but it was a difficult technique. But this technique had some unavoidable problem because it forced to download unwanted video stripes as described in IV-2. Likewise, the peers download the possible unnecessary data in catalog warming. This problem was made more efficient by using chunking/striping as explained in the active system. We can see that chunk bundling was same as the stripe bundling and which needs the peers to implement the high level operations. On the other side striping/reconstruction of video was more difficult, while all peer assisted system support chunking/unchunking of videos that was very simple operations. Also that, load balancing utilize the video stripping. Finally, we comment on our discussion that we neglect signaling bandwidth and protocol overhead effects. Our proposed system also eliminates the: 1) limitation of number of peers from which data was downloaded by the peer; 2) the network had possible congestion effects. Although, we would not give importance to system performance issues for the simplicity and tractability reasons.

G. Servers Requested For Bandwidth:

Finally, we can compute the performance by applying the active techniques explained in previous sections. 1) The loads incurred by the individual videos must be balanced 2) the download rate of cold videos was increased sufficiently.

V. RELATED WORK

In [12], the limitations and characteristics of peer to peer streaming system shown by stochastic fluid model. The minimum tree depth and server load, maximum streaming rate is derived in [13]. Anyway these two papers deal with single channel streaming system. In [9], authors develop network queuing model to explain about multichannel live streaming systems. We mention that live streaming is different from VoD systems. To predict the minimum server bandwidth for each requested channel is proposed in [14], and this work focused only on live streaming systems. Based on the sequential delivery appeared in [4], the mathematical formulation needed by VoD system is obtained. In [11], same formulation has been considered. The server load derived for upper and lower bounds in [15]. The dynamic demand of the user in peer to peer VoD system is predicted by proposing queuing model in [16]. [17] Server load savings for VoD system is presented by trace driven evaluation and it shows by using hybrid CDN-P2P for two main CDNs Akamai and Limelight. In [18], an analysis of peer-assisted VoD systems developed by author with scalable video streams. In case of flash crowds, number of peer admitted into the system is estimated using analytical model. In respect to all previous work, we analytically estimate the benefits in terms of scalability by proposing a framework in this paper. [19] Xunlei is an interesting implementation considered in our work, which is popular in china. Xunlei combines both peer-assisted and server-assisted techniques.

VI. CONCLUSION

Finally, our result show that problems in the servers can be reduced by the users' cooperation in large scale VOD systems. Because of the need to give impulse mechanism to the users and protect the system against the attacks and misbehavior however we believe they take into consideration in future, that was the only solution to make the VOD system completely scalable. Though we show that user cooperation reduce the complexity when distributing the user generated contents hence in this case the amount of users inherently scales linearly with the amount of users.

REFERENCES

[1] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-assured cloud bandwidth auto-scaling for video-on-demand applications," in Proc. IEEE IN- FOCOM, 2012, pp. 460–468.
[2] C. Huang, J. Li, and K. Ross, "Can Internet video-on-demand be profitable?" in Proc. ACM SIGCOMM, 2007, pp. 133–144.
[3] Y. Huang, T. Z. J. Fu, D. M. Chiu, J. C. S. Lui, and C. Huang, "Challenges, design and analysis of a large-scale P2P VoD system," in Proc. ACM SIGCOMM, 2008, pp. 375–388.
[4] Softonic, Barcelona, Spain, "PPLive," [Online]. Available: http://pplive.en.softonic.com
[5] "GridCast," [Online]. Available: http://www.gridcast.cn/
[6] "PPStream," 2006 [Online]. Available: http://www.ppstream.com/
[7] TVU Networks Corporation, Mountain View, CA, USA, "TVU," [Online]. Available: http://www.tvunetworks.com/
[8] D.Ciullo, V.Martina, M.Garetto, E.Leonardi, and G.L.Torrisi, "Stochastic analysis of self-sustainability in peer-assisted VoD systems" in Proc. IEEE INFOCOM, 2012, pp. 1539–1547.
[9] D. Wu, Y. Liu, and K. Ross, "Modeling and analysis of multichannel P2P live video systems," IEEE/ACM Trans. Netw., vol. 18, no. 4, pp. 1248–1260, Aug. 2010.
[10] M. Cha, H.Kwak, P. Rodriguez, Y.Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in Proc. IMC, 2007, pp. 1–14.
[11] W. Wu and J. C. Lui, "Exploring the optimal replication strategy in P2P-VoD systems:

Characterization and evaluation, "IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 8, pp. 1492–1503, Aug. 2012.
[12] R. Kumar, Y. Liu, and K. Ross, "Stochastic fluid theory for P2P streaming systems," in Proc. IEEE INFOCOM, 2007, pp. 919–927.
[13] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, "Performance bounds for peer-assisted live streaming," in Proc. SIGMET- RICS, 2008, pp. 313–324.
[14] C. Wu, B. Li, and S. Zhao, "On dynamic server provisioning in multi- channel P2P live streaming," IEEE/ACM Trans. Netw, vol. 19, no. 5, pp. 1317–1330, Oct. 2011.
[15] Y.Zhou, .Z.J.Fu, and D.M.Chiu, "Statistical modeling analysis of P2P replication to support VoD service," in Proc. IEEE INFOCOM, 2011, pp. 945–953.
[16] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "Cloud Media: When cloud on demand meets video on demand," in Proc. ICDCS, 2011, pp. 268–277.
[17] C. Huang, A. Wang, J. Li, and K. W. Ross, "Understanding hybrid CDN-P2P: Why Limelight needs its own red swoosh," in Proc. NOSSDAV, 2008, pp. 75–80.
[18] K. Mokhtarian and M. Hefeeda, "Analysis of peer-assisted video-on- demand systems with scalable video streams," in Proc. MMSys, 2010, pp. 133–144.
[19] P.Dhungel, K.Ross, M.Steiner, Y.Tian, ndX.Hei, "Xunlei: Peer-assisted download acceleration on a massive scale," in Proc. PAM, 2012, pp. 231–241.