ISSN 2278-7763

PVP: Peer-to-Peer Video on-Demand Streaming with Peer Cache Adaption*

Anjana P. Nair[1]/ Research Scholar, & Minu Lalitha Madhav[2]/ Asst. Professor

[1]Department of Computer Science & Engineering, Sree Buddha College of Engineering, Alappuzha, India; [2]Department of Computer Science & Engineering, Sree Buddha College of Engineering, Alappuzha, India
Email: anjana.p.nair@gmail.com

ABSTRACT

Interactive Video on-Demand Systems based on the internet has long been the target of new internet business on realizing the large scale user base. Researches have shown that using Peer-to-Peer (P2P) to provide Video on-Demand (VoD) service can support large scale users as well as reduce server's bandwidth cost. P2P VoD streaming using Peer Cache Adaption (PCA): PVP presents a novel P2P VoD streaming system that determines each peer's cache capacity adaptively according to the server's upload bandwidth constraint. The packet transfer rate will be a constant when number of nodes increases, since each peer acts as a temporary server for data transfer. As a result of high packet transfer rate downloading speed can be improved. Since the number of cache for each video is proportional to its popularity, upload load will be the same for each peer and fairness between download and upload amounts can be achieved. PVP is evaluated and the Server Bandwidth Rate is found to be a constant after a particular packet transfer rate when compared to µTorrent.

Keywords : Video Streaming, Video On-Demand, Peer-To-Peer, Cache Adaption, Caching.

1 INTRODUCTION

As the popularity of broadband internet access increased, more users become interested in media streaming services. Video streaming is the process of providing video data or content via a web page. Video Streaming services are of two types: Downloading and streaming. In downloading, the entire file which is delivered from the source is saved on some devices or disk. But here the user has to wait for the whole file to download before any of it can be viewed. In streaming the file is sent to the user in a constant stream, and the user watches it as it arrives, no waiting is involved. This true streaming can be used to broadcast live events. But true streaming supports one-time play only. There is also a hybrid method known as progressive download. In this method the video clip is downloaded but begins playing as soon as a portion of the file has been received. This progressive download or progressive streaming is the basis for Video-on-Demand Streaming. Video-on-Demand (VoD) is an interactive multimedia service which should enable users to start watching the video of their choice at the time of their choice, after waiting for a small start-up delay, and perform VCR-like control operations such as play, pause, fast forward, fast search, reverse search and rewind while downloading the video in parallel.

Initially to implement VoD client server approach was used. But the server upload bandwidth became a constraint as the number of users increased. Then Content Distribution Network (CDN) was introduced, which is still used in YouTube. But for CDN scalability is an issue as the deployment cost increases with increase in number of users. Then peer-to-peer (P2P) approach is there. In tree-based P2P approach an overlay tree is generated in advance among the peers and packets are pushed from parent towards children. In mesh-based P2P approach peers are self organized into an overlay mesh and packets are pulled from neighbouring peers. Many P2P VoD streaming services uses mesh-based approach.

Even though many P2P VoD streaming systems exist, they do not consider the server's upload bandwidth constraint. In this paper a novel peer cache adaption method: PVP is proposed to meet the upload bandwidth constraint of server. Each peer's cache capacity is determined adaptively so that each peer can act as a temporary server. So the packet transfer rate becomes a constant as the number of nodes increases. Upload load for each peer will be the same regardless of the popularity of the video. Also fairness can be achieved between upload and download amounts at each peer.

2 RELATED WORK

2.1 Review Stage

PROMISE: P2P Media Streaming Using CollectCast [1], works on the basis of a novel P2P service called CollectCast, which operates entirely at the application level. ``One receiver collecting data from multiple senders'' is the working principle of CollectCast. To perform the streaming task traditionally performed by a dedicated entity (a mediaserver), the limited capacity of peers are aggregated in CollectCast.
The features of PROMISE include

- It accounts for peer heterogeneity, reliability, and limited capacity.
- It matches a requesting peer with a set of supplying peers that are likely to achieve the best streaming

ISSN 2278-7763

quality.
- It dynamically adapts to network fluctuations and peer failure.

BiToS is a single video approach that basically uses BitTorrent [2]. BitTorrent enables to reproduce a file by exchanging segments with neighbours. In BitTorrent, as the segments for a video may be downloaded in out-of-order, the video cannot be played until the download is complete. So BiToS modifies its segment selection mechanism to choose segments close to the current played one with high priority, which enables a video to be played while downloading.

BASS is also based on BitTorrent where peers exchange segments with each other using BitTorrent [3]. In BASS an external media server is introduced from which, peers download segments in order skipping the segments that have been already downloaded by BitTorrent.

In Bullet-Media, all the segments of a video are replicated by overlaying in advance [4]. Distributed hash table DHT is used to locate and download the segments. Digital versatile disk (DVD)-like fast forward and random search functionalities are supported. But in the single video approach the server upload bandwidth should be proportional to the number of concurrently requested videos. This will causes scalability problem.

Multiple video approaches have been proposed to overcome this issue. In multiple video approaches the upload burden of the server is shared with peers. The peer cache scheme is first used by pcVoD [5]. When a peer wants to download a video, it first contacts a tracker. This Tracker provides the address of a peer who caches the video. Then the video is downloaded from that peer.

Push-to-peer is a content placement algorithm that improves content availability and uses each peer's uplink bandwidth more in a controlled environment [6]. The relationship between the number of peers, the number of videos, and the cache capacity at each peer assuming stationary popularity of the videos in the system was formulated in statistical modelling [7]. But if peers are under the control of a content provider then only this model can be applied.

A content placement strategy to maximize peers' uplink bandwidth in P2P VoD systems was proposed in optimal content placement [8]. But this method does not consider the fact that peers tend to provide their resource as small as possible.

VMesh which is a scheme close to each peer uses a part of its local storage to cache videos [9]. But in VMesh each peer's cache size is fixed and the same for all the peers.

3 PROPOSED SYSTEM

3.1 Structure of the System

The video server contains V distinct videos, each of which is encoded at a bit rate R. Only the video server can upload a video. When a video is uploaded its IP address is set to that of severIP & status is set to 0. Uploaded videos are stored in a folder named pecanVideos at video server. The length of each video is L(seconds). Each video is divided into several segments each of which has the same length (bytes). Each segment is identified by

- VideoID
- SegNum

N denotes the number of peers. Only registered users can access the system. Videos uploaded by the video server are listed at the users. When a peer wants to watch a video it makes a request to the system and the participating peers are searched for the requested video. The video search is implemented by using Distributed Hash Table. Each peer caches the video it watched at its local storage to send when requested by some other peers [10]. While streaming a video a peer can serve a request for that video from other peers. If a requested video is not cached at any peer, that request is served directly by the video server. A video cached at local storage is deleted manually.

3.2 Distributed Hash Table Based Search

The lookup function is implemented using a hash table. System searches for peers that caches desired video using a DHT. Videos are uploaded by video server. When a video is added its videoID and serverIP is added to the DHT. First request for a video is served directly by the server. While a peer streams a video V, ip address corresponds to V is changed to ip address of that peer. When a new request comes for V, video server returns the ip address corresponds to V in DHT. Now a peer, whose ip matches the ip address returned by video server, will serve the request.

When Peer A requests for video V, it first finds a Peer B that caches V through DHT in video server. Now Peer A streams V from Peer B's cache and in DHT ip address corresponds to V becomes the ip of Peer B. When a new request comes from Peer C for V it is served by Peer B and in DHT ip became that of Peer C and so on. When streaming is completed ip address corresponds to video V is reset to serverIP. Hence the next request for V will be directly served by the video server.

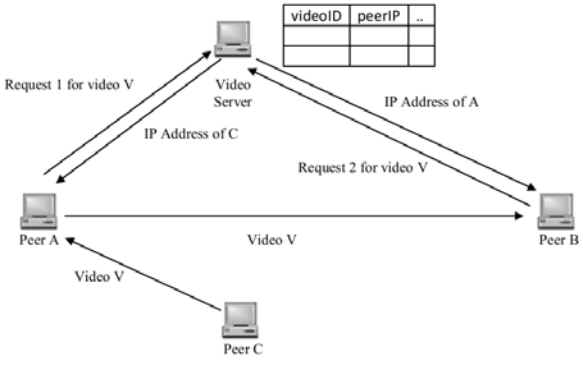

Fig. 1. Steps for Video Search & Cache Replacement

Conventional routing cache systems store destination IP addresses in their cache directory. In this paper a routing cache technique is proposed that stores the most recently used

ISSN 2278-7763

route prefixes, instead of IP addresses, to achieve a significantly smaller cache size. A nesting prefix is partially represented in this cache by its minimal expansions. Such expanded prefixes are obtained using an incremental technique without any modifications to the routing table. Consequently, cache works with most of the common route lookup algorithms and efficiently maintains coherency with the routing table.

3.3 Upload Load and Fairness

As more peers request a video its request rate increases as a result its popularity increases. If M videos are there, each video will have different popularity. A peer that caches more popular segments receives more requests which results in the use of larger upload bandwidth. So upload load for each peer may differ.

In PVP whenever a video is requested, a new cache is created for that video at the requesting peer. So the number of cache for a video is proportional to its popularity. Each peer acts as a temporary server and serves the next request to the video which is being downloaded by it, at that moment. Whenever a new peer streams a video, the ip address corresponds to that video is changed to that of the new peer in DHT. So a new request must be served by the last served peer or last client becomes the new server. So the upload load will be the same for each peer regardless of the popularity of the video.

If the cache capacity for each peer is the same, the same amount of segments is uploaded by each peer onto each other. Peers generating low segment request rates upload more segments than they download and vice versa. The peers with high request rates should upload more segments to achieve the fairness between upload and download amounts, hence they should be with more cache capacity.

Here a peer's cache capacity is proportional to the rate of video request it makes. While Peer A downloads video V, a request for V by Peer B must be served by Peer A. Now the next request for V by Peer C must be served by Peer B and so on. Each peer can upload a video V to some other peer -which requests for V- while downloading V in parallel. So upload and download amounts at each peer will be the same and hence fairness can be achieved.

3.4 Server Upload Bandwidth Constraint

When a peer requests for a video, the video size is calculated and an equal amount of local storage is taken as cache from that peer. Hence while streaming a video the cache capacity of a peer is adjusted adaptively. As a result the server's upload bandwidth constraint can be met. Each peer that caches a video can act as a server. So the server needs to handle only one peer when several peers request the same video at the same time. This will result in a constant packet transfer rate. In existing systems each peer acts as a seed and starts data transfer. As the number of peers increases each node including the server behaves as seed for new coming load thus increasing the packet transfer rate of the server.

3 Performance Analysis

3.1 Server Bandwidth Rate

PVP is evaluated in real time and a performance analysis is done. PVP is compared with µTorrent in real time. For streaming a video of size 699Mb, µTorrent takes 1 hr (actually it is 8 hrs, since number of seed is low set it as 1 hour) while PVP takes only 15mins (with 3 peers).

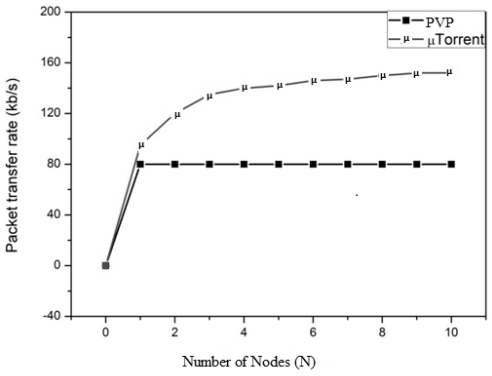

Fig.2. Server Bandwidth Usage of the Proposed System Compared to µTorrent.

The server bandwidth usage of PVP compared to µTorrent is evaluated. The server bandwidth usage based on the number of nodes and packet transfer rate (kb/s) is depicted. In PVP, when number of nodes increase the packet transfer rate is constant because each node will behave as a temporary server for data transfer so the server need to handle only one node at time. But in other system, the each node will act as a seed and start the data transfer. If the number of nodes increases, all nodes including the server behave the seed for new coming load. So the packet transfer rate of server will increase while the number of nodes increases.

3.4 Query Accessibility Rate

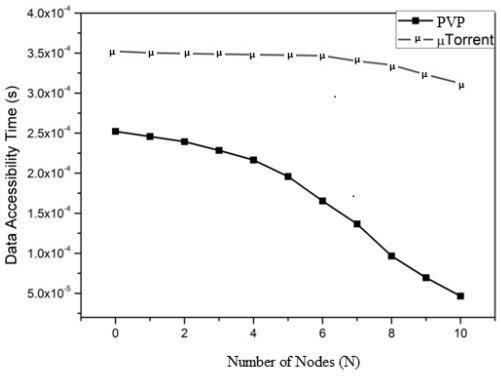

Fig.3. Query Accessibility Rate of the Proposed System Compared to µTorrent.

ISSN 2278-7763

Here the query accessibility rate of PVP and µTorrent is evaluated on the based on the number of nodes and data accessibility time(s). Here in PVP initial value for data accessibility is an constant value because the time required to identity the local server. But in µTorrent this initial value is higher than PVP because it required to collect all the available peers in their network. When the number of nodes increases the data accessibility time is suddenly decrease for PVP due to the easily availability of server. But in µTorrent the data accessibility time almost maintain the same rate, because all the time it requires initialing all peers for accessing data.

From this performance evaluation it is proved that PVP provides more downloading speed when compared to µTorrent. Start-up delay to start watching a video is also reduced in PVP.

4 Conclusion

PVP is a novel cache adaption method proposed for P2P VoD streaming systems to meet server's upload bandwidth constraint. Server bandwidth usage is a constant at some Packet transfer rate in the proposed method. Fairness between upload and download amounts at each peer can also be achieved.

As a future work the proposed method can be integrated with BitTorrent architecture for better performance in P2P VoD Streaming.

Acknowledgment

I express my sincere thanks to my guide Ms. Minu Lalitha Madhav, Asst. Professor, Dept of Computer Science, Sree Buddha College of Engg., for her support throughout this work.

I would like to extend my gratitude to the anonymous referees for their extremely useful comments on an earlier draft of this article.

References

[1] A. Vlavianos, M. Iliofotou, and M. Faloutsos, "BiToS: Enhancing BitTorrent for supporting streaming applications," in Proc. IEEE Global Internet Symp., Apr. 2006.

[2] C. Dana, D. Li, D. Harrison, and C.-N. Chuah, "BASS: BitTorrent assisted streaming system for video-on-demand," in Proc. IEEE MMSP, Oct. 2005.

[3] N. Vratonjic, P. Gupta, N. Knezevic, D. Kostic, and A. Rowstron, "En-abling DVD-like features in P2P video-on-demand systems," in Proc.

[4] L. Ying and A Basu, "pcVOD: Internet peer-to-peer video-on-demand with storage caching on peers," in Proc. DMS, Sept. 2005.

[5] K. Suh, C. Diot, J. Kurose, L. Massoulie, C. Neumann, D. Towsley, and M. Valleo, "Push-to-peer video-on-demand system: Design and evaluation," IEEE J. Sel. Areas Commun. , Dec. 2007.

[6] Y. Zhou, T. Z. J. Fu, and D. M. Chiu, "Statistical modeling and analysis of P2P replication to support VoD service," in Proc. IEEE INFOCOM, Apr.

[7] B. Tan and L. Massoulie, "Optimal content placement for peer-to-peer video-on-demand systems," inProc. IEEE INFOCOM, Apr. 2011.

[8] W.-P. Yiu, X. Jin and S.H. Chan, "VMesh: Distributed segment storage for peer-to-peer interactive video streaming," IEEE J. Sel. Areas Commun., Dec. 2007.

[9] Jongtack Kim and Saewoong Bahk, "PECAN: Peer Cache Adaptation for Peer-to-Peer Video on-Demand Streaming",IEEE journal of communications and networks, vol. 14, no. 3, june 2012.

