International Journal of Recent Engineering Science (IJRES),
ISSN: 2349-7157, *volume1 issue2 March to April 2014*

DYNAMIC SCHEDULING BASED-DIST RIBUTED SEGMENT STORAGE FOR P2P INTERACTIVE VIDEO STREAMING

V.Uma, Dr.P.Palanisamy
Research scholar, Assistant Professor,
Department of Computer Science and Engineering,
Swami Vivekananda Institute of Engineering & Technology, India

Abstract - Random access functions in peer-to-peer video-on-demand streaming is challenging,due to not only the asynchronous user interactivity but also the unpredictability of group dynamics. Video-on-demand is one of media streaming services where movies are delivered to desktops of distributed users with low delay and free interactivity. P2P technologies have been proven as a scalable solution to multicasting and file sharing among distributed users. In P2P, user interactivity is an issue because user may jump forward, backward and resume its playback any time.

Videoes in the video server are split into video segment and it is transmitted to the node in P2P network and send it to client. The P2P network is having a track sever which contains the information about each and every node in the peer to peer network. The node sends the collected video segment to the client and plays the video at the client appropriately. This project uses static local storage instead of sliding window buffering to help handle user interactivity efficiently and to reduce the complexity.The scheduling issues in video streaming will be addressed by using factored queue length. This scheme schedules the video with maximum factored queue length is obtained by each video queue length with factor which is biased against the more popular videos.

1 INTRODUCTION

A.VIDEO STREAMING

On-demand media streaming has recently gained intensive consideration due to its promising usage in a rich set of Internet-based services such as video on demand, distance learning, media distribution, etc. However, there are still many challenges towards building efficient, scalable on-demand streaming systems due to the high bandwidth and delay requirements for media streaming.

Scalability — the system is simple and scalable to large number of users with low server bandwidth requirement. It is completely decentralized, without the need of a server to organize overlay nodes.
Efficiency — users can start playing the media with low delay as it is based on streaming without file downloading.
Failure-resilience — the system is r obust to peer dynamics, node and lin k failu res to o ffer continuous streaming.
Interactiv ity support — users can in teract with the media at any time.

A majority of existing on-demand media streaming systems follows the Client-Server design, in which videos are stored in aset of central servers. Video-on-demand (VoD) is one of such services where mov ies are delivered to desktops of distributed users with low delay and free interactiv ity(in terms of pause, jump forward/backward,etc.). providing VoD with traditional client- server architecture where each client is not scalable to large number of clients.

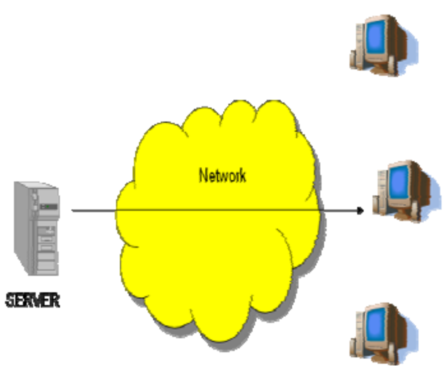

Figure1 Video Streaming

This is mainly due to heavy server load and limited network bandwidth at the serverside. Though IP multicast may be used as a scalable solution for media streaming, providing such services worldwide is still challenging due to the lack of widely deployed multicast capable networks and dedicated proxy servers [10], [8]. Recently, peer-to-peer (P2P) technologies have been proven as scalable solution to many applications, e.g., multicasting and file sharing among distribu ted u sers [9],.

The process of providing video data or content via a web page. Video delivery by video streaming attempts to overcome the problems associated with file download, and also provides a significant amount of additional capabilities. The basic idea of video streaming is to split the video into parts, transmit these parts in succession, and enable the receiver to decode and playback the video as these parts are received, without having to wait for the entire video to be delivered. Video streaming can conceptually be thought to consist of the follow steps:

1. Partition the compressed video into packets
2. Start delivery of these packets
3. Begin decoding and playback at the receiver while the video is still being delivered.

A media stream can be on demand or live. On demand streams are stored on a server for a long period of time, and are available to be transmitted at a user's request. Live streams are only available at one particular time, as in a video stream of a live sporting event.

B.VIDEO-ON-DEMAND

VOD systems either stream content through a set-top box, allowing viewing in real time, or download it to a device such as a computer, digital video recorder, personal video recorder or portable media player for viewing at any time. The majority of cable and telephone company based television providers offer both VOD streaming, such as pay-per-view, whereby a user buys or selects a movie or television program and it begins to play on the television set almost instantaneously, or downloading to a DVR rented from the provider, for viewing in the future.

Download and streaming video on demand systems provide the user with a large subset of VCR functionality including pause, fast forward, fast rewind, slow forward, slow rewind, jump to previous/future frame etc. These functions are called trick modes. For disk-based streaming systems which store and stream programs from hard disk drive, trick modes require additional processing and storage on the part of the server, because separate files for fast forward and rewind must be stored. Memory-based VOD streaming systems have the advantage of being able to perform trick modes directly from RAM, which requires no additional storage or CPU cycles on the part of the processor.

It is possible to put video servers on LANs, in which case they can provide very rapid response to users. Streaming video servers can also serve a wider community via a WAN, in which case the responsiveness may be reduced. Download VOD services are practical to homes equipped with cable modems or DSL connections. Servers for traditional cable and Telco VOD services are usually placed at the cable head-end serving a particular market as well as cable hubs in larger markets. Some airlines offer audio VOD as in-flight entertainment to passengers through individually-controlled video screens embedded in seatbacks or armrests or offered via portable media players. Airline AVOD systems offer passengers the opportunity to select specific stored video or audio content and play it on demand including pause, fast forward, and rewind.

Push video on demand is a technique used by a number of broadcasters on systems that lack the interactivity to provide true video on demand, to simulate a true video on demand system. A push VOD system uses a Personal Video Recorder (PVR) to automatically record a selection of programming, often transmitted in spare capacity overnight. Users can then watch the downloaded programming at times of their choosing.

C.PEER TO PEER NETWORK

A distributed network architecture may be called a Peer-to-Peer (P-to-P, P2P,.) network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers,.). These shared resources are necessary to provide the Service and content offered by the network (e.g. file sharing or shared workspaces for collaboration). They are accessible by other peers directly, without passing intermediary entities. The participants of such a network are thus resource Service and content) providers as well as resource (Service and content) requestors (Servent-concept).

Type of P2P Networks

1. Centralized P2P network such as Napster
2. Decentralized P2P network such as KaZaA
3. Structured P2P network such as CAN
4. Unstructured P2P network such as Gnutella
5. Hybrid P2P network (Centralized and Decentralized) such as JXTA

A distributed network architecture has to be classified as a Pure Peer-to-Peer network, if it is firstly a Peer-to-Peer network according to secondly if any single, arbitrary chosen .Terminal Entity can be removed from the network without having the network suffering any loss of network service. A pure P2P network does not have the notion of clients or servers but only equal peer nodes that simultaneously function as both "clients" and "servers" to the other nodes on the network. This model of network arrangement differs from the client-server model where communication is usually to and from a central server. A typical example of a file transfer that is not P2P is an FTP server where the client and server programs are quite distinct, the clients initiate the download/uploads, and the servers react to and satisfy these requests.

Advantages of P2P Networks

An important goal in P2P networks is that all clients provide resources, including bandwidth, storage space, and computing power. Thus, as nodes arrive and demand on the system increases, the total capacity of the system also increases. This is not true of a client-server architecture with a fixed set of servers, in which adding more clients could mean slower data transfer for all users.

The distributed nature of P2P networks also increases robustness in case of failures by replicating data over multiple peers, and -- in pure P2P systems -- by enabling peers to find the data without relying on a centralized index server. In the latter case, there is no single point of failure in the system.

II. RELATED W ORK

The peer-to-peer concept has been applied to more general video-on-demand services. Peer-to-peer networks for streaming video on the Internet have generated a lot of interest recently. P2P-based streaming systems completely rely on peer connections, which make the system vulnerable to peer or connection failures. Then combine P2P techniques with the current server-client streaming model to build a hybrid system that is both scalable and robust. First, propose a streaming system, BitTorrent Assisted Streaming System (BASS) [5] for VoD services, where we add the use of an external streaming server to a slightly modified BitTorrent. Clients can simultaneously stream from the media server as well as each other via BitTorrent P2P connections. By maintaining these connections, we can reduce the aggregate bandwidth used by the media server and decrease client waiting times.

In current VoD systems, whenever a new client starts a video session, a dedicated stream is allocated to serve the user till the end of the viewing session. Video data are transmitted to the client in a point-to-point manner, known as unicast streaming. In unicast streaming, clients can control playback of the video at will, such as performing pause/resume and seeking [2] Unlike unicast transmission, a multicast video stream can be shared by more than one receiver.

Scheduling policies for batching are first-come-first-served (FCFS) and maximum queue length (MQL).[3] Neither of these policies lead to entirely satisfactory results. MQL tends to be too aggressive in scheduling popular videos by only considering the queue length to maximize batch size**,** while FCFS has the opposite effect.[7] The factored queue length is obtained by weighting each video queue length with a factor which is biased against the more popular videos.

A prefix caching technique [6] whereby a proxy stores the initial frames of popular clips. Upon receiving a request for the stream, the proxy initiates transmission to the client and simultaneously requests the remaining frames from the server. By transmitting large frames in advance of each burst, work ahead smoothing substantially reduces the peak and variability of the network resource requirements along the path from the proxy to the client. The proxy prefix caching technique for peer-to-peer video streaming require upstream bandwidth of a peer to be larger than video playback rate.

III PROPOSED SYSTEM ARCHITECTURE

A) INTRODUCTION

The section describes the modules of the system to be implemented and it includes both high level and detailed design through pseudo code.

B) SYSTEM ARCHITECTURE

This section presents a brief description of the main modules present in the system.

Video sever stores the videos for user access. Each video is divided into N segments like frames, each of them identify by its video ID and segment ID. The existing work uses “cache and relay” paradigm in which a peer caches what it has recently played out for a period of time before discarding it and uses the cached content to serve others.

This project uses static local storage instead of sliding window buffering to help handle user interactivity efficiently and to reduce the complexity. In this scheme the peer can start caching the next segment when it nearly finishes playing out the current segment. Depending on the capacity of its local storage, each peer stores a number of segments randomly chosen from the N segments of the video. These peers are referred as storage peers. There are multiple copies of each video segment in the network. When a client want to play a segment it first looks for the supplying peers of that segment in the P2P network, then sends requests to those peers for the service.

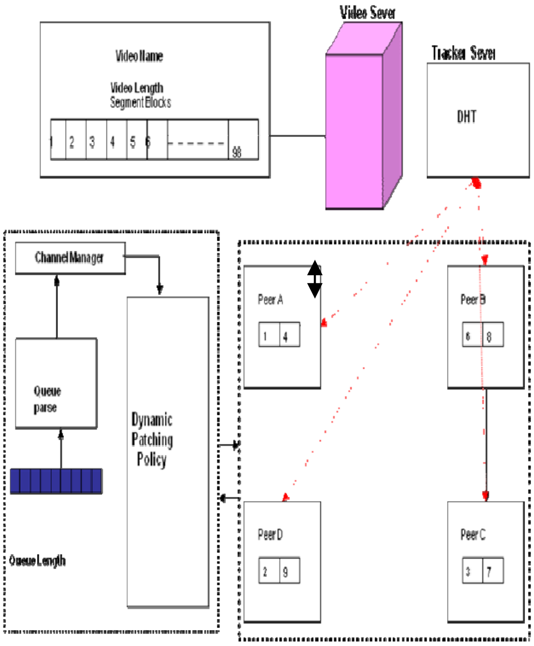

Figure 2 Architecture

If there is no supplying peer, the requesting peer requests the media server. video segments are distributed among peers, utilizes Distributed Hash Table(DHT) to locate these segments. DHT is a structured overlay constructed among peers., where images are periodically captur A storage peer holds a few video segments and keeps a list of peers who have the next, previous, or current segments for seek and load balancing purposes The pointers are used to redirect users to the appropriate peers storing the next requested segments. In case of joining or jumping, a VoD client in our system searches for its parents using DHT and gets the stream from those parents.

1) Video Sever- splitting and transmitting video

Video server stores the videos for user access. Each video is divided into N segments each of them are identifiable by its video Name, ID and segment ID. In full splitting scheme, each video is divided into smaller number of data blocks that are proactively transmit to peers inP2P during low network utilization time. That segmented video stored on nodes.

2)Segments distributing and searching

Video requests from client to any nodes in P2P. When the required video content is not available in particular node searches the segment by using Distributed Hash Table. Each stored segment should be registered in the DHT network by the storage peer, so as to allow other peers to locate the segment.DHT maintains all node information in P2P . DHT contains video name, video id, segment id and its nodes.

Input:
candidateList: candidate peers sorted by GSt in descending order;
num_candidates: number of candidates;
num_segs: number of segments;
num_replicas: number of replicas for the first segment;
Assigning:
j = 1;
***for** (i = 0; i < num_segs + num_replicas; i++) {*
select jth node in candidateList, suppose the selected node is Nk;
***if** (i < num_replicas + 1)*
assign segment 0 to node Nk;
else
assign segment (i - num_replicas) to node Nk;
***if** (j == num_candidates)*
j = 1;
else
j++;
}

Segments distributing algorithm

3) Scheduling for multiple clients using bandwidth

Multiple clients asking same videos mean we go for scheduling. Now introduce client side scheduling that is dynamic batching policy. In this scheduling want different suppliers to send different portion of a segment to the receiver at the same time. Further divide each segment into equal-sized blocks. Thus the receiver can parallel download different blocks from different supplying peers in real-time model.

#

Input:
num_suppliers: number of suppliers;
supplier[i]: the suppliers sorted by Bwr in descending order;
Bwr[i]: reserved bandwidth at supplier[i];
num_blks: number of blocks;
blocks[i]: blocks;
blk_size: block size;
deadline: deadline for finishing downloading the last block;
Scheduling:
***for** (i = 1; i <= num_suppliers; i++) {*
time_left[i] = deadline; // time_left[i]: time left for supplier
// to send blocks;
}
curr_blk = num_blks-1;

```
while (curr_blk ≥ 0) {
max_t = max {time_left[1], … ,
time_left[num_suppliers]};
for (i = 1; i <= num_suppliers; i++) {
if (time_left[i] == max_t) {
assign blocks[curr_blk] to supplier[i];
time_left[i] = time_left[i] – blk_size / Bwr[i];
curr_blk --;
}
if (curr_blk < 0)
break;
}
}
```

IV RESULTS AND DISCUSSIONS

This project is implemented using java media framework and oracle database. Each video will be split into equal sized segments, and these segments will be distributed to peers according to our segments distributing algorithm.

The Peer-to-Peer approach is well-suited to cooperative distribution of stored video among nodes in a P2P network, where the peers themselves operate under provider control. Segmented video is transmit to peers in P2P and stored at peers local storage in distributed manner.

The factored queue length is obtained by weighting each video queue length with a factor which is biased against the more popu lar videos. An optimization problem is formulated to solve for the best weighting factors for the various videos.

Figure4 Video split into video segment

Figure3 Video display in client system

VCONCLUSION AND FUTURE DIRECTION

#

Providing interactive VoD service in P2P network is challenging, not only due to the asynchronous user access pattern but also the unpredictability of group dynamics and user interactivity.videos in the video sever are split into video segment and it is transmitted to the nodes in peer to peer network in a distributed manner. The P2P network is having a track sever which contains the information about each and every node in the peer to peer network. When a client requests for a video to any node in the peer to peer network, it seeks the network for the starting video segment then next segment and so on. The node sends the collected video segment to the client and plays the video at the client appropriately. A video mesh is built upon peers to support playback and jumping. A peer, who has a video segment stored in its local storage,
connects to the peers who have the same, the previous and the next video segments.

A simple but powerful concept called the factored queue length as a basis for scheduling batched videos in a VOD environment. The factored queue length is derived from an analytical model which optimizes the average latency time of a viewer in the absence of defections.

VI REFERENCES

[1] W.P. Ken Yiu, Xing Jin, S.-H. Gary Chan"VMesh: Distributed Segment Storage forPeer-to-Peer Interactive Video Streaming" IEEE Journal on Selected Areas in Communications,vol.25,no.9,December2007, Page No 1717-1731.

[2] Ying Wai Wong, Jack Y. B. Lee, Victor O. K. Li, and Gary S. H. Chan, "Supporting Interactive Video-on-Demand With Adaptive Multicast Streaming", IEEE Transactions on Circuits and Systems For Video Technology, Vol. 17, No. 2, February 2007, Page No 129-142.

[3] Charu C. Aggarwal Joel L. Wolf and Philip S. Yu "On Optimal Batching Policies for Video-on- Demand Storage Servers "IEEE Proceedings of Multimedia '96 Page No 253-258.

[4] Xin Liu,Jun Wang,Son T. Vuong,"A Peer-to-Peer Framework for Cost-Effectiven-Demand Media Streaming" IEEE Communication Society subject matter experts for publication in the IEEE CCNC 2006 proceeding ,Page No314-318.

[5] C. Dana, D. Li, D. Harrison, and C. Chuah. "BASS: BitTorrent assisted streaming system for video-on-demand". In Proc. IEEE MMSP,pp 1-4, 2005.

[6] S.Sen, J. Rexford, and D. Towsley. "Proxy prefix caching for multimedia streams". In Proc. IEEE INFOCOM, pp-1310-1319, 1999.

[7] Kyung Oh Lee, Jeong Bae Lee, Kee Wook Rim"A Dynamic Scheduling Algorithm for Video-on-Demand Servers".In proc. IEEE Transactions on Consumer Electronics, 1114 Vol. 50, No. 4, November 2004.

[8] K. A. Hua, Y. Cai, and S. Sheu, "Patching: A Multicast Technique for True Video-on-Demand Services," in Proceedings of the 6th ACM International Conference on Multimedia (MM), Bristol, England, Sept. 1998.

[9]Y. Chu, S. Rao, and H. Zhang, "A Case for End System Multicast," in Proceedings of ACM SIGMETRICS, Santa Clara, CA, USA, June 2000.

[10] L. Gao, D. Towsley, and J. Kurose, "Efficient Schemes for Broadcasting Popular Videos," in Proceedings of the 8th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video(NOSSDAV), Cambridge, UK, July 1998.