Chunk Based Approach for Video Streaming in Peer to Peer Network

Mrs. Nilam S. Patil , Mrs. D. A. Phalke

***Abstract*— Peer-to-Peer system has emerged as a promising technology to provide video-on-demand service. Several P2P-VoD systems have been deployed and attracted a large population of viewers. However, it is challenging to design P2P media streaming networks because of the stringent time constraints on the delivered media streams, which require more efficient and resilient overlay architectures. Compared with existing client server system model, where the single server must have enough resources to support all simultaneous clients. Proposed system focuses on design of P2P media streaming networks, which requires the fast distribution of multimedia content to clients. Proposed algorithm is described as 1)serial and parallel scheduling of chunks correctly switched to get required quality of service 2)dynamic nature of peers is considered 3)fault tolerance is provided in case of peer failure.**

Keywords*— Peer to Peer System, Video on Demand (VoD), Content Delivery Network, Overlay Network, Bittorrent, File sharing network.

I. Introduction

The early model for content distribution is a centralized one, in which the service provider simply sets up a server and every user downloads files from it. In this type of network architecture (server-client), many users have to compete for limited resources in terms of bottleneck bandwidth or processing power of a single server. As a result, each user may receive very poor performance. From a single user's perspective, the duration of a download session, or the download time for that individual user is the most often used performance metric. Server storage and bandwidth is expensive. Content uploading depends on policy used for uploading. YouTube employs CDN to stream video to end users" [8]. "YouTube videos today are typically less than 10 minutes in length and have a bit rate under 200 kbps**"**. As video files are large in size and require continuous delivery.

II. Literature Review

P2P VoD is a P2P architecture for VoD streaming proposed in [6]. Peers are organized in a tree-based overlay, grouped by generations where a generation is a group of peers having always the same smallest numbered block in their cache. Children receive data using a push-based scheme from a single parent,which uploads data from its own playing cache. Parents are proposed to be chosen using a round robin, smallest delay, or smallest distance selection algorithm, from a list obtained from the server.

BiToS (BitTorrent Streaming) is P2P treaming protocol for VoD proposed in [7]. Peers are organized in a mesh-based overlay and data distribution follows a pull-based scheme. Parents are chosen using a tracker program running on a server. Pieces requests and exchanges among peers follow both rarest-piece-first and tit-for-tat policies. Pieces from a particular video file are contained in three components called Received Pieces, High Priority set and Remaining Pieces Set. Pieces from the Received Pieces set can be shared with peers.

HON (Hybrid Overlay Network) is a P2P protocol for VoD streaming [5] where the system constructs both a tree overlay and a mesh overlay which collectively deliver the video data to the clients; much of the data is delivered through the mesh overlay while a node will only resort to the tree overlay if it fails to fetch some segment after a certain deadline. Both overlays use a pull-based scheme for data delivery. Nodes are assigned their parent in the tree overlay by a managing node , responsible for constructing and maintaining the overlay. Multiple parent peers in the mesh overlay are chosen using a gossip selection algorithm.

III. Proposed System

Proposed chunk based approach uses file sharing nature of Peer to Peer system to design distributed cluster to store entire information peers participating in video delivery and using this information delivery time of large video files is improved than traditional client server approach. The performance of system is improved as the number of peers participating in peer to peer file sharing network increased.

Due to the distributed nature of the P2P network, searching and locating data of interest in the network has been an important issue in the project. In reality, data searching time only contributes a very small portion of a download session while the most delay is caused by actually transferring the file from source peers. Thus, to minimize the download time for each user, reducing the actual file transfer time would make more noticeable difference. Proposed system focused on reducing the total download duration of Video file.

Chunk based approach solves issue of scalability problem by making system distributed. Every peer acts as server as well as client. Video files are delivered by partitioning them into

Mrs. Nilam S. Patil, Mrs. D. A. Phalke
Department of Computer Engineering,
D. Y. Patil College of Engineering,Akurdi,
University of Pune.
India

Mrs. Nilam S. Patil
e-mail: neelam_v_patil@rediff.com

Mrs. D. A. Phalke
email: a_dhanashree@rediff.com

chunks. Parallel downloading chunks to improve download rate. During downloading if system is stucked because of some bad peer means failure in delivery of chunk, there is need of failure recovery.

Algorithm: Chunk Based Approach

```
Symbols used:   N : Total no of peers
            SP: Service peer list (Neighbour list)
            $V_i$ : Video file
            $P_r$ : Requesting peer
           $BS_i$ : Bit sequence sent by service peer Spi
          $|BS_i|$ : Total length of $BS_i$            $\delta$=0..2MB

Input  : Video file name
Output : Video File

Serial Scheduling:
$P_r$ searches  SP  for $V_i$
$SP=SP_n$                 where  $1<=n<=N$
IF FS($V_i$) =$\delta$
            Pr  chooses  a service peer SPi
            BuildConnect
            Receive Streaming Data
Else
            Chunk Scheduling

Chunk Scheduling:
For each $SP_i$
            Send Split Request
             $S_n$=Build Parallel Sessions   \\ n no.of neighbors with fil
            FailureRecovery()
            Receive streaming data
            FailureRecovery()
             Arrange chunks in Ascending order
            Disconnect Sessions
End

FailureRecovery()
Finding next  best Neighbour
```

IV. Proposed System Implementation

A. *Peer to Peer System Architecture*

Peer to peer system architecture used for proposed system design is shown below.

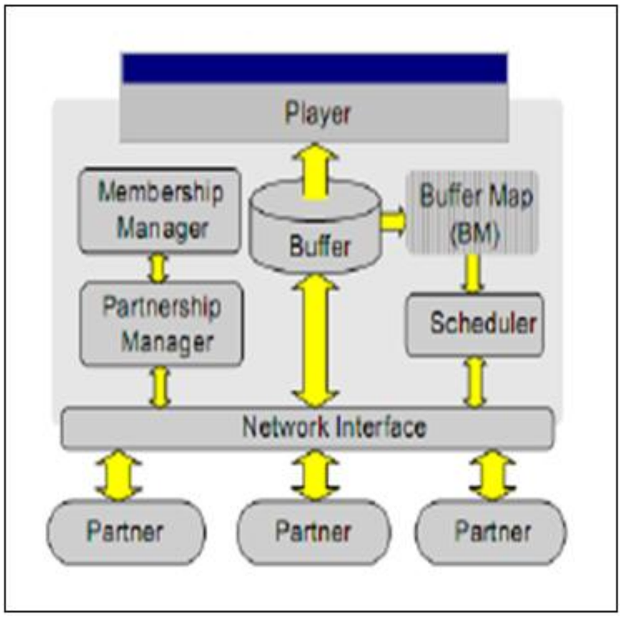

Figure1. Peer to Peer System Architecture

.

Functions of different components in above Figure are described as

- Membership manager : Maintains the partial view of the Network structure.
- Partnership manager : Maintains partnership with other nodes.
- Scheduler : Schedule data transmission across streams.
- Buffer : Store video data before playbacks.
- Buffer Map : Represents the current status of the buffer and data request

B. *Implementation*

The proposed system is divided into three modules.

- **NetworkClusteringDaemon:**

This is the main module responsible for the controlling the different activities and for communicating among the clients. Basically main functions of this module are:

Task-id generation:
Processes communicating using MPI are collected into groups, often to allow the subdivision of work into independent chunks. Processes can only communicate with other processes that are in the same group, although processes can be members

of multiple groups. Processes are identified by their positions (rank) within the groups that they belong to.
NetworkClusteringDaemon creates a task-id of the client in the form of computer name and virtual port number, on which it is going to receive the requests.

Creating Buffer and Packing and Unpacking of buffer:
As in case of Clustering communication among the multiple client to be done by sending and receiving the messages among them, NetworkClusteringDaemon creates buffer which is used for sending a data or message to another client and receiving a data or message from the another clients, It is done by packing the buffer means the data to be sent has to be packed into the buffer on sending side, and it should be unpacked on the receiver side it has to be collected by unpacking the buffer.

Generation of parallel environment:

To achieve parallel processing, parallel environment should be created. NetworkClusteringEnviornment is responsible for this, which have the information of all the clients and their tasks. Once this is done, all clients will have their job for execution. And clients run simultaneously to achieve parallelism.

Receive Process task and Execute task:

To increase the efficiency, our program is divided into subtasks and these tasks are distributed among the multiple clients, who on receiving the tasks execute them and transfer it to the server.

- **Messaging:**

Message passing provides a means for complex tasks and computations to be broken down into discrete pieces, each of which can be performed in parallel on different processors. Messages containing processing requests and data can be passed from one processor to another in a controlled, synchronized manner.The main function of this module is to generate a message tag, passing the message and receiving the message.

- **Process Execution**

Once the necessary tasks are distributed among the multiple clients, this module runs these processes.

V. Results

In the proposed system, algorithm focuses on the average download time of each video requested by user in a P2P network. With the devastating usage of network resources by P2P applications in the current Internet, it is highly desirable to improve the network efficiency by reducing each user's download time. In contrast to the commonly-held client server architecture focusing on centralized content store, proposed system considers peer to peer file sharing system.

Experimental setup is prepared for video files of size ranging from bytes to MB and number of peers ranging form 1 to 4.

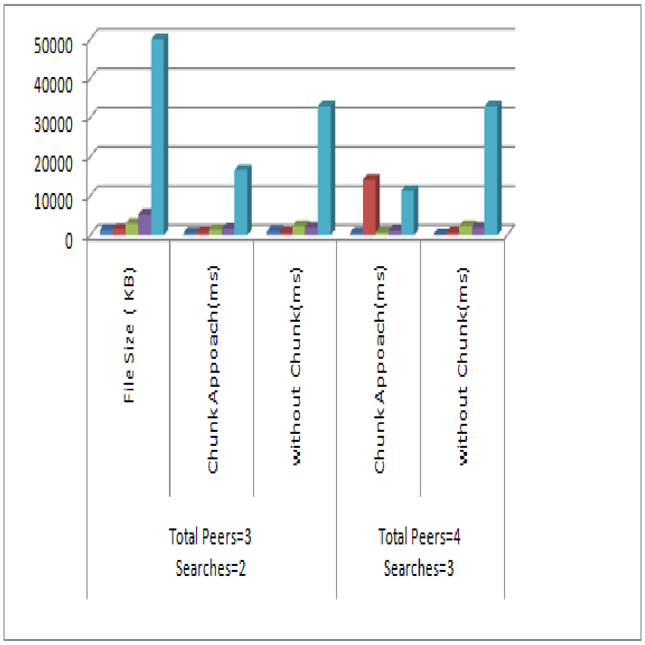

Figure2. Number of peers vs Different approach used for download time

Figure 2 shows that with increase in number of peers download time decreases for larger videos with size greater than 2MB.Therefore performance of the system increases with increase in number of peers and the system is scalable.

After downloading using chunk approach is started, if a peer departs or failed, then next neighbor is searched for required chunk. Dynamic departure of peers, will not terminate the downloading process. Thus, system is fault tolerant .

The experimental results indicate that proposed mechanism for P2P-based Video Delivery System could reduce the server's workload, thus improving the system scalability greatly, at the cost of the neighborhood communication among the peers. However, the gossiping cost is trivial for the following reasons: (1) a peer only exchanges gossip messages with its neighboring peers, which is much fewer than the total online peers; (2) a gossip message is much smaller in size compared with a media segment, and for the secondary buffers the delay is tolerable

VI. Conclusion

Centralized video-on-demand systems such as CNN Pipeline, YouTube both have their drawbacks. Services are paid for by the audience or advertisement, the video quality is low or the number of concurrent users is limited. All these drawbacks are due the limited scalability of these systems. Peer-to-peer video-on-demand systems have an improved scalability over centralized video-on demand systems. These systems are far less centralized or do not contain centralized components at all. However proposed distributed approach is more complex, it accelerate time of video delivery, system resources are effectively used, as scalability increases performance is improved or stable. Fault tolerance is also provided. Chunk based approach reduces cycle times. It can be useful for arranging training session, Next generation P2P TV. System is designed to consider wired network. It can be extended for wireless or wired-wireless hybrid network

References

[1] G. Sakaryan, M. Wulff, and H. Unger, "Search Methods in P2P Networks: A Survey," Innovative Internet Community Systems, pp. 59-68, 2006.

[2] J. Risson, and T. Moors, “Survey of research toward peer-to-peer networks: Search methods,” Computer Networks,vol. 50, no. 17, pp. 3485-3521, 2006.

[3] D. Kundur, L. Zhu, M. Merabti et al., "Advances in Peer-to-Peer Content Search", pp. 404-407.

[4] Danqi Wang, Chai Kiat Yeo, "Super chunk based Fast Search in P2P-VoD System", in IEEE GLOBECOM Global Telecommunications Conference, pp.1-6,2009

[5] Ming Zhou, Jiangchuan Liu, “A Hybrid Overlay Network for Video-on-Demand” in Proceedings of IEEE International Conference on Communications (ICC), Seoul, Korea, pp. 1309-1311,2005.

[6] Do T.T., Hua K.A., and Tantaoui M.A., “P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment”, in Proceedings of IEEE International Conference on Communications, 2004, Volume 3, pages 1467- 1472, 20-24 June 2004

[7] Vlavianos A., Iliofotou M., and Faloutsos M., “BiToS: Enhancing BitTorrent for supporting streaming applications”, in Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006), Barcelona, Spain, pages 1–6, April 2006

[8] Yong Liu, Yang Guo, Chao Liang,” A survey on peer-to-peer video streaming systems”,Peer to Peer Networking and Applications , Volume 1, Issue 1, pp 18-28, March 2008

[9] Peer-to- peer interactive video streaming,” S Areas in Communications, IEEE Journal on, vol. 25, p. 1717–173 1,December 2007.

About Author (s):

Nilam Patil, student of ME Computer in University of Pune. Her research interests include Peer to Peer network, Distributed System.

D. A. Phalke , Received ME from University of Pune. She is Assistant Professor. Her research interests are Distributed System, Database System.