International Journal of Computer Applications in Engineering Sciences
[VOL I, ISSUE II, JUNE 2011] [ISSN: 2231-4946]

A Novel Load Balancing Strategy for Cluster Based Hybrid Video-on-Demand Systems

Vinay. A[1], Prateek Saxena[1], T N Anitha[2], M Bharathi[2]

[1] *Department of Information Science and Engineering, P E S Institute of Technology*
Bangalore, Karnataka, INDIA

[2] *Department of Computer Science and Engineering, S J C Institute of Technology*
Chikkaballapura, Karnataka, INDIA

[1]`a.vinay@pes.edu`

***Abstract*— Video-on-Demand (VoD) has been an active area of research for the past few years in the multimedia research community. Significant research efforts have been directed towards reduction of network bandwidth requirements, improvement of server utilization, and minimization of start-up latency. Server load and network bandwidth are the major performance issues in streaming video over the Internet. Thus, load balancing among servers is one of the major challenges. This paper presents a novel load balancing strategy for cluster based hybrid Video-on-Demand (VoD) systems. The architecture used for load balancing contains peer servers in different levels of hierarchy. The simulation results of the proposed technique illustrates that the load on the servers in a group is balanced up to 90% and the rejection rate is reduced.**

***Keywords*— Video-on-Demand (VoD) systems, load balancing, clustering, replication, intra-cluster load balancing, inter-cluster load balancing**.

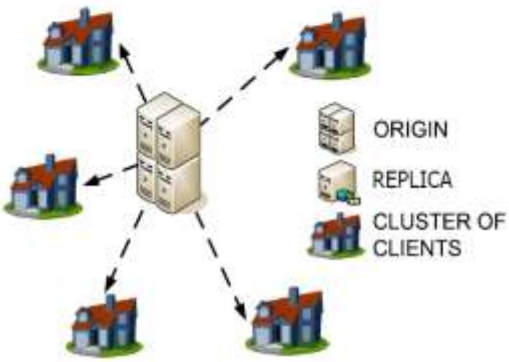
I. INTRODUCTION

Video-on-demand (VoD) systems have evolved as one of the most popular types of multimedia service on the Internet over the past decade. Video news clips, full-length movies, TV shows, video lectures, videos made and shared by common people are watched by millions of users every day over Internet. VoD systems exert enormous pressure on both the video servers and the network because of their high bandwidth and stringent delay requirements. The video servers' throughput and the network traffic volume grow linearly with the number of clients and further lead to imbalance in the load. The net effect is on the performance of the overall system in terms of long delays or even request rejections. Hence appropriate techniques for load balancing among servers is utmost important in the current scenario mainly for achieving high throughput, availability, and scalability. In this direction, this paper presents a novel load balancing strategy for cluster based hybrid Video-on-Demand (VoD) systems.

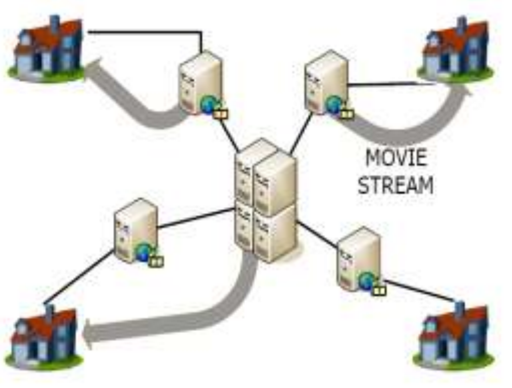
The rest of the paper is organized as follows; section II describes the VoD system architectures; section III deals with the problem formulation; section IV describes related work; section V deals with proposed system; section VI illustrates simulation results and section VII concludes the work carried out.

II. VOD SYSTEM ARCHITECTURES

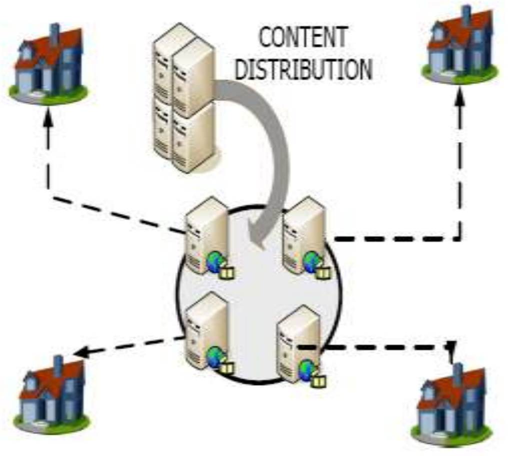
Video-on-Demand (VoD) is an interactive multimedia service which enables users to start watching the video of their choice at the time of their choice, after waiting for a small startup delay, and perform VCR-like control operations such as play, pause, fast forward, fast search, reverse search and rewind. A wide number of architectures have been developed and deployed by the researchers working in this domain. Four important architectures include centralized architecture, proxy-based architecture; content delivery networks approach and peer-to-peer overlays. Figure 1 illustrates the said architectures. Each of the architecture has its own pros and cons.



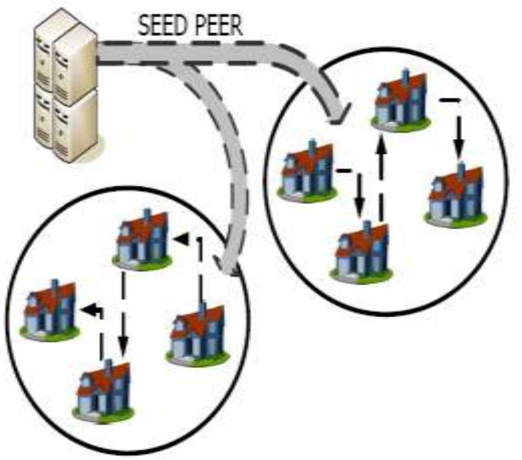
(a) Centralized Architecture



(b) Proxy-based Architecture



(c) CDN Approach



(d) Peer-to-peer Overlay Architecture

Figure 1: Typical Video-on-Demand (VoD) System architectures

Figure 1 (a) illustrates, the centralized architecture. Here all the clients are directly connected to the server. The server is responsible for data access and / or storage and data transfer. The client on the other hand is responsible for decoding and playing the video. The rest of the application logic may reside at the client, server or be divided over the two. The advantage of this architecture is it is simple and easy to design. But it suffers from scalability, reliability, availability, consumes higher bandwidth and difficult to maintain. An alternative solution or an enhanced version of the centralized architecture is the proxy based architecture as depicted in figure 1(b). A proxy server can be considered as an intermediate node along the server-to-client path. This in effect will partition the server-to-client path into a server-to-proxy path and a proxy-to-client path. When a video is accessed by a client, the whole video or a portion of the video may have already been cached in the proxy server. If the video is not stored in the proxy server, it needs to be accessed from the central server and delivered to the client. It is possible at the same time the video will be cached in the proxy server to improve the future accesses. However, the proxy based architecture too suffers from many drawbacks as that of the centralized architecture.

The first approach to deal with the some of the issue of centralized and proxy based Video-on-Demand systems is the use of Content Delivery Network (CDN). Figure 1(c) illustrates the architecture of CDN. A CDN is a network infrastructure with several network elements for scaling and enhancing the delivery of content from providers to many end-users over the Internet. CDNs provide services that improve network performance by maximizing bandwidth, improving accessibility and maintaining correctness through content replication. They offer fast and reliable applications and services by distributing content to cache or edge servers located close to users. A CDN has some combination of content-delivery, request-routing, distribution and accounting infrastructure. CDN balances the load through multiple servers, but cost involved in purchasing and maintaining the server is very high. Thus CDN approach is performance-effective but not cost-effective. CDNs are currently able to provide video streaming services with a satisfying video quality. Nevertheless, they may still raise a scalability issue in case of increasing number of users and very high quality videos encoded at higher bit rates.

The second approach to deal with the some issue of centralized and proxy based Video-on-Demand systems is use of Peer-to-Peer overlays as depicted in figure 1(d). P2P architectures enable a variety of new applications to take advantage of the distributed storage and increased computing resources offered by such networks. P2P systems represent a scalable and cost effective alternative to classic media delivery services, which allows for extended network coverage in the absence of IP multicast or expensive CDNs. Their advantage resides in their ability for self organization, bandwidth scalability, and network path redundancy, which are all very attractive features for effective delivery of media streams over networks. However, specificities of media applications in terms of bandwidth, delay, and reliability are not completely addressed by the characteristics of P2P systems. The lack of coordination of such systems, the limited peer capabilities, and the low system stability over time represents a great challenge for the deployment of high quality P2P streaming applications. The replacement or extension of conventional media delivery infrastructures with P2P systems clearly necessitates adaptation of existing coding, routing, and scheduling algorithms to unreliable network environments.

In order to address the limitations of all these architectures, a hybrid architecture consisting of combination of different types of architectures has been proposed by many authors. This paper uses one such hybrid architecture consisting of centralized and peer-to-peer concepts and presents a load balancing algorithm for the same.

III. PROBLEM FORMULATION

General formulation of the load balancing problem is as follows: given a large number of requests, find the allocation of requests to servers optimizing a given objective function (e.g. total execution time). The main goal is to design a mechanism that uses the load balancing algorithm. Here it is assumed that each server in the hybrid VoD architecture is characterized by its processing rate and video replica that it contains. Only the servers i know the true value of its processing rate. This is periodically sent by individual servers to the cluster coordinator or load distributor. Video request arrive at the coordinator with a given arrival rate. The load balancing algorithm on the cluster coordinator finds the fraction of load that is allocated to each server in the cluster such that the expected execution time is minimized. The cost incurred by each server is proportional to its utilization.

IV. RELATED WORK

Liang-Teh Lee et al proposed a load balanced PC-Cluster for the VOD server system has been proposed. Two-Tier model is used in the systematic architecture, and PC-Cluster are used as storage system of the VOD server. The load balancing mechanism is based on the Least-Connection-First algorithm. Furthermore, a video placement strategy is also proposed in this paper to share and balance the loads among video servers in the cluster. Accompany with the dynamically adjusted files in each video level, a dynamically cyclical video replacement mechanism has been proposed to replicate and allocate video files for improving load balance of the system [7].

H S Guruprasad et al proposes a load balancing algorithm for a distributed VoD architecture using agents. The proposed approach groups a set of local proxy servers into a Local Proxy Server Group [LPG] for load balancing among the proxy servers. However the proposed algorithm always uses maximum number of channels between the proxy servers in a LPG and also between the CMS and the proxy servers of a LPG by allocating more channels to the more popular videos [8].

Yitzhak Birk work focuses on load-balancing for the purpose of providing throughput that is independent of viewing choices. At the inter-disk level, data striping is the obvious solution, but may lead to a quadratic growth of RAM buffer requirements with system size. At the intra-disk level, multi-zone recording results in variable disk throughput. Deterministic schemes for solving each problem are discussed, as well as their joint operation. Finally, efficient staging of data from tertiary storage devices to disk is shown to be possible [9].

Anant Nimkar et al proposes a greedy video placement and disk load balancing algorithms to minimize the static and dynamic loads of disks respectively. The proposed greedy video placement along with round robin DLB improves the performance of the VoD system by 10% as compared to random video placement along with round robin DLB. The proposed DLB using the variance minimization algorithm along with random video placement evenly distributes the load among disks continuously except at startup time of the VoD server [10].

Pushpendra Kumar Chandra et al propose an algorithm for a wide variety of workload conditions including I/O intensive and memory intensive loads. However, the CPU requirements of the system is minimum as the tasks which arrive are mostly video fetch tasks which require negligible system interaction but a lot of I/O consumption. The proposed load balancing algorithm tries to achieve the effective usage of global disk resources in the VOD cluster [11].

V. PROPOSED METHOD

A. *SYSTEM MODEL*

The proposed system model for VoD system is illustrated in Figure 2. The architecture is a cluster based hybrid architecture consisting of server peer at various levels and client server model in the base level or level 0 in the architecture. The components of the architecture are Multimedia Server (MS), Global Server Group (GSG), Intermediate Server Group (ISG), Local Server Group (LSG), Local Coordinator (LC), Intermediate Coordinator (IC), Global Coordinator (GC) and Clients. The MS consists of high end data storage with all the video files. The GSG and ISG consist of a set of intermediate peer servers and are associated with ISG and LSG through respective coordinators. The peer servers in turn consist of subset of the videos that are stored on MS.

Each cloud in the figure represents service provider regions i.e. cluster. The client sends request to LC, if the requested movie / video are available among the servers of the LSG, then the LC forwards request to the server in the LSG and the server streams video to the client. If the video request cannot be satisfied by the servers of LSG, the LC forwards it to next level in the hierarchy. Similar set of actions are performed at the ISG and GSG levels. The servers in the LSG, ISG and GSG are classified strong cluster server and weak cluster server based on the popularity of the video that they contain. The servers in the LSG contain highly popular videos. The popularity decrease as one moves up in the hierarchy. The MS contains all video files. Either in servers of LSG, ISG or GSG, the popularity of a video is the measured through the hit rate of the video. Table 1 lists the various nomenclatures used in the algorithms to represent various system components.

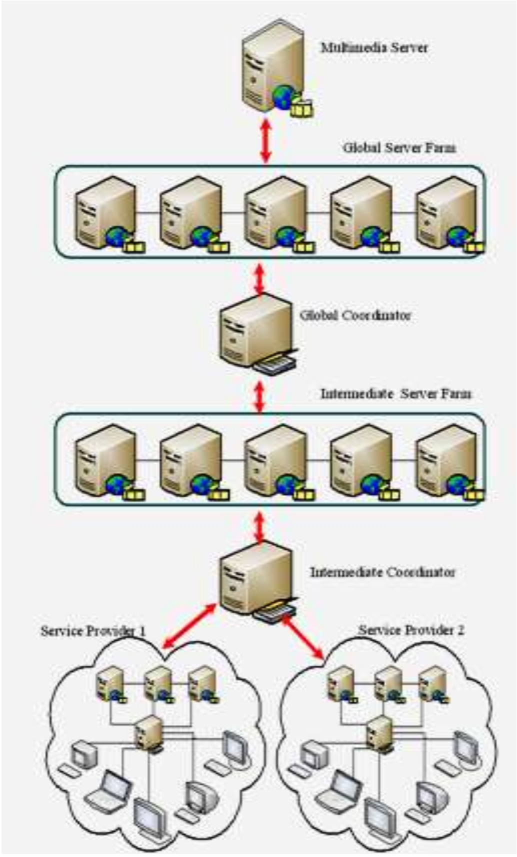


Figure 2: Hybrid Video-on-Demand (VoD) System Model

Table 1: Nomenclatures Used

SG	:	Server Group
S_i	:	Server in a server group
C_i	:	Client
W_i	:	Weight of a server S_i in SG
V_{req}	:	Video Request
W	:	Vector containing server id with their weights
SBW_i	:	Total available bandwidth in server S_i
SSP_i	:	Processing speed of the server S_i
SAL_i	:	Access latency
SMZ_i	:	Memory size
S_c	:	Strong cluster group
S_w	:	Weak cluster group
SPS_i	:	Super Peer Server in SG
λ	:	Threshold value for W_i
W_{avg}	:	Average workload
W_{th}	:	Workload threshold (5% of W_{avg})
SPS_{LK}	:	List of k^{th} super peer server

B. CLUSTERING

Let us consider a collection of N server nodes that are part of a server group (Local, Intermediate and

Global). It is assumed that each server always stores one copy of its own content item which it serves to clients and that it has additional storage space to store *k* replicated content items from other server which it can also serve. Each video is associated with an authoritative *origin server* (OS) in the network. The video copy located at the origin server is called the *origin copy* and a video copy at any remaining server is called a *replica*. The servers in the server group are grouped into strong and weak clusters based on their weight vector which comprises the following parameters: available capacity, processing speed, memory and access latency.

Procedure for clustering

```
Create_Cluster()
{
1. for each $S_i$ in $SG$ calculate weight $W_i$ of server as
            $W_i = (SBW_i + SSP_i + SMZ_i) / SAL_i$

2. $W = \{S_i, W_i\}$ in descending order of weights.

3. for each $S_i$ in $W$
     if $W_i \geq \lambda$
            append $S_i$ into $S_c$
     else
            append $S_i$ into $S_w$
}
```

C. REPLICATION

The videos are classified as Class I and Class II, based on their popularity. The most frequently accessed videos are ranked as Class I and the less frequently accessed videos as Class II. Then more copies of Class I videos are replicated in strong clusters (servers having high weights). Routing is performed hierarchically by broadcasting the request to the strong clusters.

Procedure for replica placement

```
Create_Replica()
{
1. for each $V_{req}$ arriving from $C_i$,
             Identify $V_{req} \in C_I$ or $C_{II}$

2. if $V_{req} \in C_I$
            Assign $V_{req}$ to $S_c$
    else
            Assign $V_{req}$ to $S_w$

3. if Replica of $V_{req}$
            Serve request
   else
            If $SBW_i$ && $SSP_i$ && $SMZ_i$ available
            Create replica
   else
            Put $V_{req}$ into Queue
}
```

D. LOAD BALANCING

A typical Video-on-Demand (VoD) system has a number of servers working independently with each other. Some of them are linked by communication channel, while some are not. Each server possesses an initial load i.e. video request, which represents the amount of work to be performed. Each server may have a different processing capacity. To minimize the time to perform all tasks or to serve all the incoming video requests from the clients, an efficient load balancing algorithm is required. Thus, the VoD systems need efficient load balancing of the requests mainly for achieving high throughput, availability, and scalability. A number of load balancing algorithms for VoD system as stated in related work have been proposed. The current ubiquitous environments are quite different from those assumed by the classical load balancing algorithms. An effective load balancing scheme requires correct and timely knowledge on the global system state (e.g., the workload distribution, video popularity, execution time, etc.). However, the global state of a large-scale VoD system swiftly and dynamically changes, and it is very difficult to accurately model the system using a typical approach.

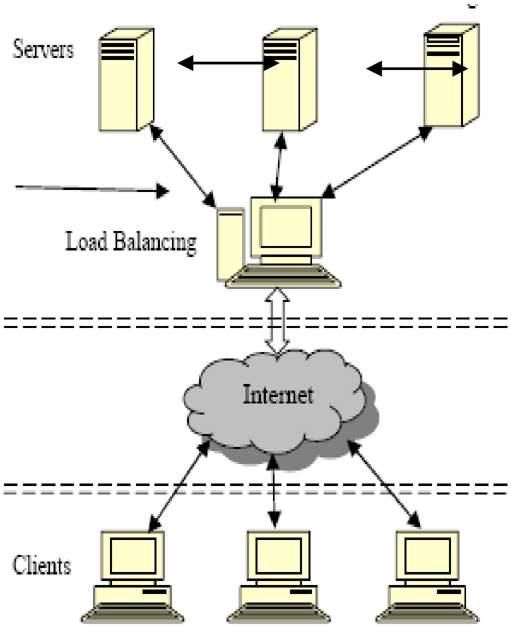


Figure 3: Load Balancing Model

The proposed load balancing implements intra-cluster and inter-cluster load balancing through the replication of videos. Balancing the load within a particular cluster is called as intra-cluster load balancing and balancing the load among the clusters is called as inter-cluster load balancing. Thus complete load balancing across the system is achieved. The proposed load balancing algorithm is also a dynamic load balancing algorithm because it makes decision at run-time for both intra-cluster and inter-cluster. The load balancing is achieved through replication strategy. Every cluster monitor called Super Peer Server (SPS) observes its server availability over a period of time. A video will be replicated only under the following circumstances:

- if the disk space at the server having the required video is very low.
- if the disk space of the server is larger than that of the requested video

The servers periodically checks the popularity of the video on a weekly basis and removes the less popular ones to utilize disk space for storing other videos that are more popular. However, the original copy is stored at the server in which replication was done first.

1) Intra-Cluster Load Balancing

The Super Peer Server (SPS) receives the information periodically regarding the current loads W_i and available disk space SMZ_i of the servers. Based on W_i the tracker SPS creates a sorted list SPS_L of the servers. For instance consider that are n servers in the list. Among the last [n/2] servers in the list, the servers whose corresponding values of SMZ_i less than the threshold MZ_{Th} are deleted. Thus, load balancing is achieved by replicating the hot video from the first server in the list to the last server and the second server to the second last server and so on. If the load difference between the servers exceeds a threshold λ, then the video is replicated.

Procedure for Intra-Cluster Load Balancing

```
IntraCluster_Loadbalancing()
{
1. for each $S_i$ in $S_c$ and $S_w$,
        Periodically communicate $W_i$ and $SMZ_i$ to SPS.

2. SPS appends $W_i$ and $SMZ_i$ to its list $SPS_L$

3. for each last (n/2) $S_i$ in $S_c$ / $S_w$
        if $SMZ_i < MZ_t$
            Delete $S_i$ from $SPS_L$

4. for each remaining $S_i$ in $SPS_L$
        if $W_i (S_i) - W_i (S_{i+1}) > \lambda$
            Replicate videos from first half of
            $SPS_L$ to last half (ascending to
            descending order)
}
```

2) *Inter-Cluster Load Balancing*

Inter-cluster load balancing is necessary in order to prevent load imbalance among the clusters. Here, the load current loads W_i ΣSPS_i is exchanged from the SPS with their neighbouring SPS periodically. The SPS_k checks whether its load exceeds the average load of the set SPS_i of its neighbouring cluster leaders by more than workload threshold W_{th} of the average load. If it exceeds, then it determines the hot video which should be moved. It sends a message about the disk space requirements SMZ of each video to each SPS_i to transfer some part of its load to them. Each SPS_i check the available disk space in each of their cluster servers. They send a message to SPS_k about the total loads and their available disk space if the space limits are satisfied. Then, SPS_k arranges the servers which are ready in SPS_i to the list SPS_{LK} so that the first element of the SPS_{LK} is the least loaded cluster. The hot videos are assigned to the willing servers in RR fashion. Once the new videos are received all SPSi sort the list SPS_{LI} in descending order of load of its servers and performs intra-cluster load balancing.

Procedure for Inter-Cluster Load Balancing

```
InterCluster_Loadbalancing()
{
1. for each $SPS_i$ in the network
        Periodically communicates $W_i$ and $SMZ_i$ to
        each other.

2. for each $SPS_i$ in the network
        Compute the average load $W_{avg}$ of the $\Sigma SPS_i$

3. for each $SPS_i$ in the network
        if $W_i > (W_{avg} + W_{th})$
        {
            Calculate the exceeded $W_i$ and $SMZ_i$
            Broadcast the new $SMZ_i$ to all
            remaining SPS.
            Collect the $W_i$ and $SMZ_i$ information
            from the responding $SPS_i$ and sort in
            ascending order.
        }

4. Do until $W_i$ of heavily loaded $SPS_i$ is equal to $W_{avg}$
        for each $SPS_i$ in the new list created
        {
            Transfer videos starting from to the
            first $SPS_i$
            Each $SPS_i$ performs intra-cluster load
            balancing.
        }
}
```

VI. RESULTS AND DISCUSSIONS

The simulation of the proposed load balancing strategy yielded in the results as depicted in the following graphs. The simulation experiments were conducted for the server of LSG1 and LSG2 (Service Provider Servers). Figure 4 to 6 illustrates the load on individual servers of LSG1. To illustrate that the load among the servers of LSG1 are balanced a graph depicting the load on all three servers is drawn (figure 7). Figure 8 to 10 illustrates the load on individual servers of LSG2 and figure 11 illustrate that the load among the servers of LSG2 are balanced. The intra-cluster and inter-cluster load balancing algorithms yielded in results depicted in the following graphs, the load on the servers of LSG1 and LSG2 are balanced up to 90%. Slight variation s in the load is due to the bandwidth availability or video replication time.

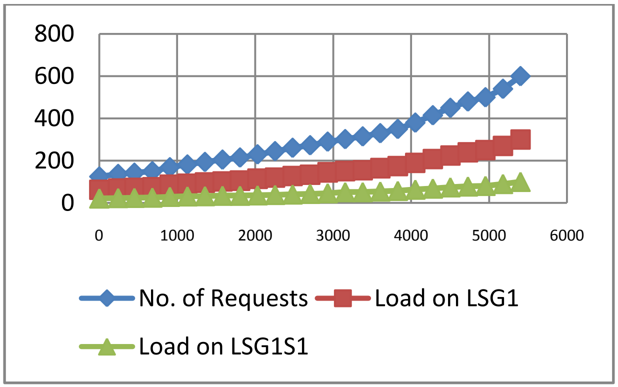


Figure 4: Load on LSG1S1

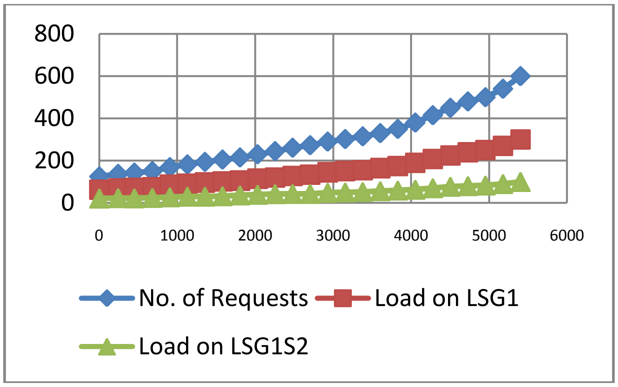


Figure 5: Load on LSG1S2

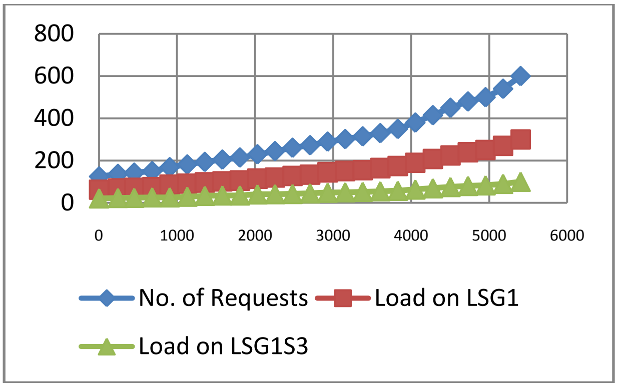


Figure 6: Load on LSG1S3

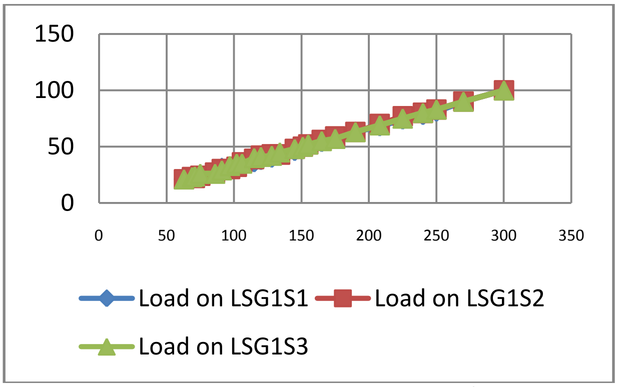


Figure 7: Load balanced among servers of LSG1

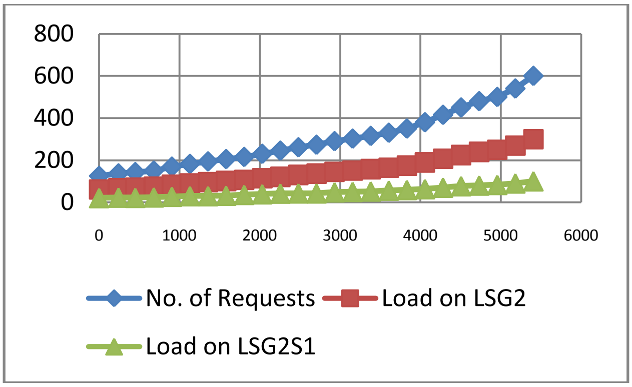


Figure 8: Load on LSG2S1

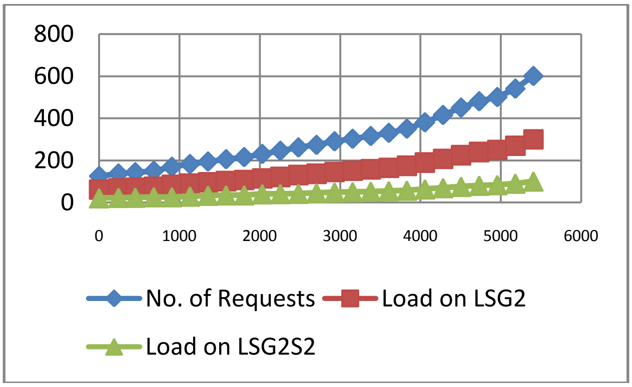


Figure 9: Load on LSG2S2

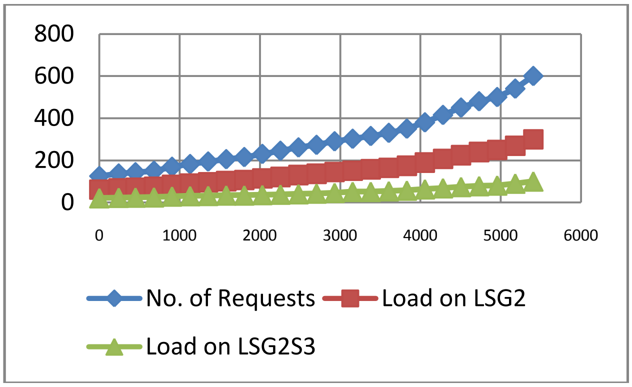


Figure 10: Load on LSG2S3

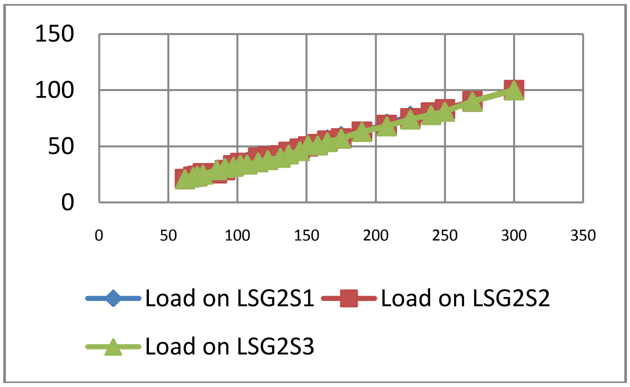


Figure 11: Load balanced among servers of LSG2

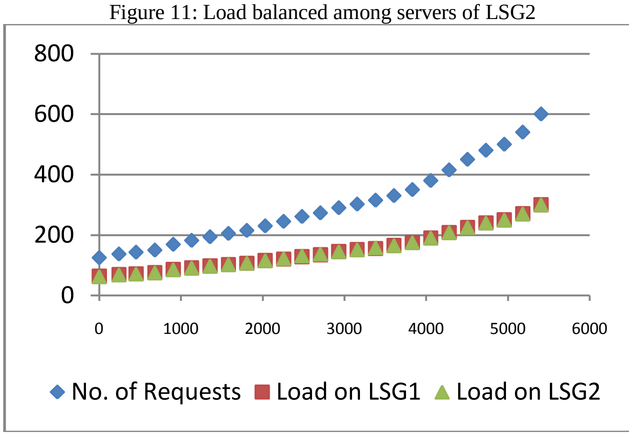


Figure 12: Load balanced between LSG1 and LSG2

The delay between request arrival time and response start-up time is illustrated in figure 13. Here, there is an acceptable amount of delay as load increases exponentially in the systems.

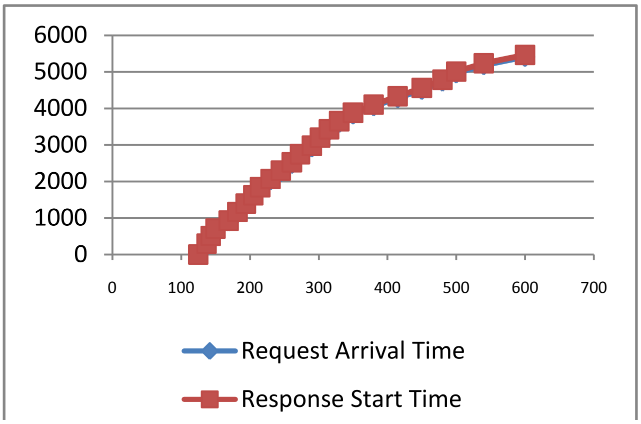


Figure 13: Request Arrival Vs. Response Start-up Time

Figure 14 and 15 depicts the total number of request being successfully served and rejected by the system respectively. Around 8% of the total requests are rejected because of non-availability of resources for serving them.

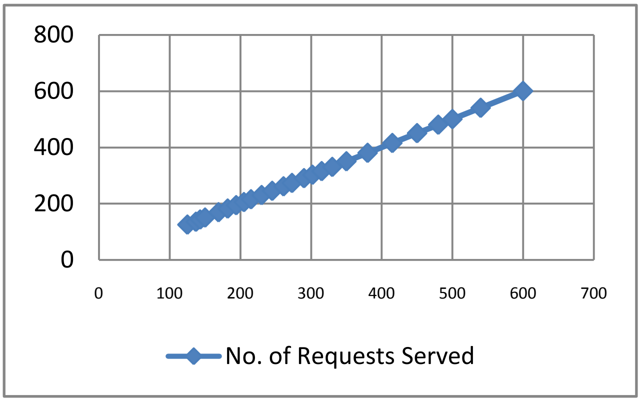


Figure 14: Total number of requests served

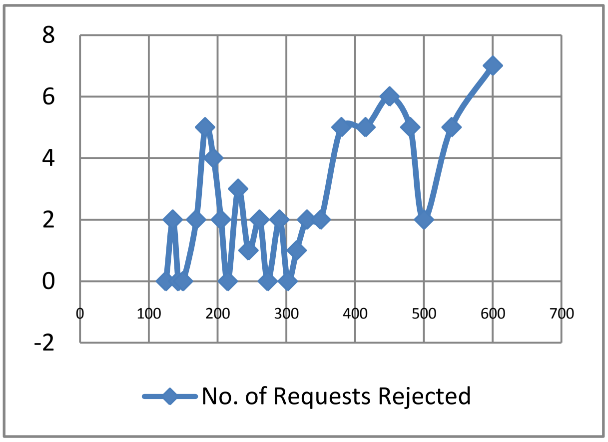


Figure 15: Number of requests rejected

VII. CONCLUSIONS & FUTURE WORK

This paper proposed a novel load balancing strategy for cluster based hybrid VoD systems. The usage of efficient hybrid architecture will good replication strategy, intra-cluster and inter-cluster load balancing strategies, use of coordinator (tracker) nodes, peer servers at various levels in the system hierarchy increases the system performance. Thus the system has features like availability and reliability.

Simulation of the proposed strategy yielded in fruitful results. Load among the servers of cluster (service provider's region is considered as a cluster) was up to 90%. Future work includes improving the system model to reduce the number of servers in the used in the architecture. The proposed load balancing strategy can be modified to provide system wide load balancing rather than just at the server groups. Pricing strategy for inter-cluster routing among different service providers can also be a novel strategy. Finally, the rejection rate must be reduced.

ACKNOWLEDGMENT

The authors would like to express heartfelt sincere gratitude to P E S Institute of Technology and S J C Institute of Technology for all the support provided for carrying out this research work.

REFERENCES

[1] D. Ligang, V. Bharadwaj, and C. C. Ko, "Efficient Movie Retrieval Strategies for Movie-On- Demand Multimedia Services On Distributed Networks," In the proceedings of Multimedia Tools Applications, vol. 20, no. 2, pp. 99–133, June 2003.

[2] Kai Fischbach Detlef Schoder and Christian Schmitt. "Core Concepts in Peer-to-Peer Networking", Idea Group Inc, Chapter 1, pp. 1–27, 2005.

[3] D. Ghose and H.J. Kim, "Scheduling video streams in video-on-demand systems: A survey", In the proceedings of Multimedia Tools and Applications, Chapter 11, pp. 167–195, 2000.

[4] Thomas D.C. Little and Dinesh Venkatesh, "Prospects for Interactive Video-on-Demand", In the proceedings of IEEE Multimedia, Vol 1, pp. 14–24, 1994.

[5] Notre-Dame de la Paix, Namur, "Video-on-Demand over Internet : A Survey of Existing Systems and Solutions", 2008.

[6] Z. Miao, A. Ortega, "Proxy Caching for Efficient Video Services over the Internet", In the Proceedings of Ninth International Packet Video Workshop (PVW '99), pp. 20-26, 1999.

[7] Liang-Teh Lee, Hung-Yuan Chang, Der-Fu Tao, and Siang-Lin Yang, " A Load Balanced PC-Cluster for Video-On-Demand Server Systems", *In the proccedings of International Journal of Grid and Distributed Computing*, pp. 63 – 70, 2005.

[8] W. Jaiphakdee, C. Srinilta, ""Dynamic Load Balancing of Short Videos in Heterogeneous Storage Environment", *In the proceedings of the International MultiConference of Engineers and Computer Scientist*, Vol I, pp. 1- 5, 2010.

[9] H S Guruprasad, H D Maheshappa, "Dynamic Load Balancing Architecture for Distributed VoD using Agent Technology", *In the proceedings of International Journal of Computer Science & Security (IJCSS)*, Volume (2) : Issue 5, pp. 13-22, 2008.

[10] Anant Nimkar, Chittaranjan Mandal, Chris Reade, "Video Placement and Disk Load Balancing Algorithm for VoD Proxy Server", *In the proceedings of 3rd IEEE international conference on Internet Multimedia Services Architecture and Applications*, pp. 141-146, 2009.

[11] Pushpendra Kumar Chandra, Bibhudatta Sahoo, "Performance Analysis of Load Balancing Algorithms for cluster of Video on Demand Servers", *In the proceedings of IEEE International Advance Computing Conference*, pp. 408-412, 2009.