Advances in Computational Research
ISSN: 0975-3273 & E-ISSN: 0975-9085, Volume 4, Issue 1, 2012, pp.-61-65.
Available online at http://www.bioinfo.in/contents.php?id=33

A SURVEY ON LOAD BALANCING TECHNIQUES IN VIDEO ON DEMAND (VoD) SYSTEM

DHAGE S.N.* AND MESHRAM B.B.

Department of Computer Technology, VJTI (Autonomous Institute), Matunga, Mumbai-400 019, India.
*Corresponding Author: Email- sudhirdhage@gmail.com

Received: February 21, 2012; Accepted: March 06, 2012

Abstract- Telecommunication industry believes that the video-on-demand market will expand heavily in the near future. The service providers are interested in the deployment of video-on-demand (VoD) systems with large numbers of videos. As the number of videos and customers increases at a VoD system, two problems are faced. The first problem is the manual video allocation of multiple copies of videos to the disks, called video placement. The second problem is where to forward a newly arrived request to play a video so that the dynamic loads amongst disks or arrays of disks of video servers are balanced. Video-on-demand (VoD) has become one of the essential parts in telecommunication industry and the most crucial service of the broadband network applications. An important issue in the design and operation of the VoD system is to distribute the movie traffic load evenly across the disks in the system. In this paper, we provide a survey on the existing Load Balancing architectures, their performance and identify their tradeoffs, based on the analysis of an existing deployment of a Video on demand service.
Keywords- Load balancing, Video on demand, Network Architecture model, Load Balancing Algorithms

Citation: Dhage S.N. and Meshram B.B. (2012) A Survey on Load Balancing Techniques in Video on Demand (VoD) System. Advances in Computational Research, ISSN: 0975-3273 & E-ISSN: 0975-9085, Volume 4, Issue 1, pp.-61-65.

Introduction

Video-over-IP applications have recently attracted a large number of users over the Internet. A video-on-demand system can be designed using any of the three major network configurations – centralized, networked and distributed. In a centralized system configuration, all the clients are connected to one central server which stores all the videos. All the client requests are satisfied by this central server. In a network system configuration, many video servers exist within the network. Each video server is connected to a small set of clients and this video server manages a subset of the videos. In a distributed system configuration, there is a central server which stores all the videos and smaller servers are located near the network edges. When a client requests a particular video, the video server responsible for the requests ensures continuous playback for the video [1][3].

The best solution for streaming video over the Internet from the above three models is the client-server service model. A client sets up a connection with a video server and video content is streamed to the client directly from the server. Similar to many other service systems, an important issue in the design and operation of the VoD system is to distribute the movie traffic load evenly across the disks in the system [2]. Load balancing actually means distributing the traffic evenly across the network along with maintaining the response time. Load balancing can be achieved by many ways out of which the ways which will be discussed in this paper are Dynamic Load Balancing Using Agents, Global Wait Queue Load Sharing Algorithm, Popularity and partial Replication Load Sharing, Adaptive Viewers Bias Based Algorithm, Least Connection First Load Balancing Algorithm[6][7].

The rest of the paper is organized as follows:Section 2 gives the Network Architecture Models, Section 3 describe Performance of each Algorithms, Section 4 gives the differentiating parameters and Finally, the paper is concluded with some final remarks.

Network Architecture Models

The different architectures used to implement various load balancing techniques mentioned above are

A. Least connection first load balancing algorithm

This architecture has two main parts namely Load Balancing Server and Video Servers. Load balancing Server: is responsible for sending and receiving packets between client and video servers and also responsible for load balancing strategies, Include video information database with serving video states in each video server and connecting load information database with currently load states in each video server [6]. Video servers are Sending video information with serving video states in each video server to load balancing server and stored video files [2].

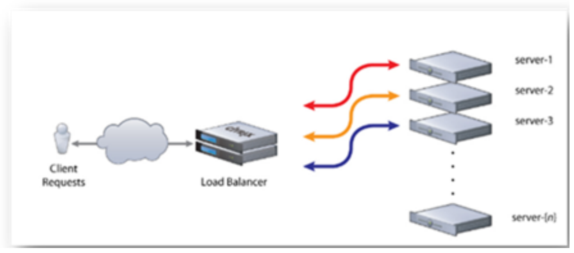

Fig. 1- load Balancing Architecture

B. Global Wait Queue Load Sharing Algorithm and Popularity and partial Replication Load Sharing

Both the methods Popularity and partial Replication Load Sharing and global wait queue load sharing have the same system architecture consisting of loosely coupled N servers connected with each other through an ATM switch.

The bandwidth of the ATM switch is Bswitch and each link that connects one server with the switch has a Blink bandwidth on each direction. Each server has attached Nterm clients, and can retrieve concurrently Nstream streams. When a server has not enough local bandwidth to attend a local request, it initiates a dialog with the other servers. If there is another server with idle retrieval capacity, then that server can help to service the request [4]. The model is a distributed system of loosely coupled servers where some objects (videos) are replicated, and eventually a video can be retrieved from alternative servers, depending on their respective load[1][4].

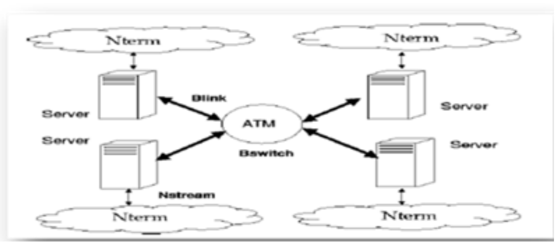

Fig. 2- load sharing Architecture

C. Adaptive Viewers Bias Based Algorithm

This technique uses Berkley VoD system architecture. The Berkeley system comprises a metadata Database which stores information about videos in the system. Usually, the client will query and browse via a user interface the metadata DB for information about the video. The client will then select a video of his interest and submit his request to the system. The system will first check if a copy of the requested video is already in one of the CMS's. If a copy exists and if server bandwidth is available, the system will require the CMS containing the copy to deliver the video in real time. If a copy does not exist, the system will make a request to load the requested video from AS to a CMS. The CMS serves as a cache between the AS and the client [5].

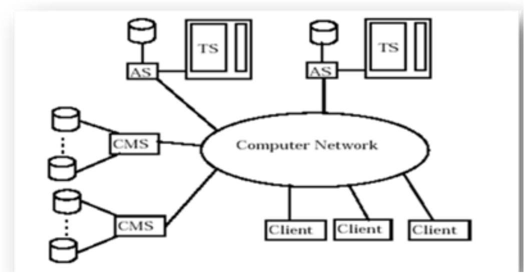

Fig. 3- Berkley VoD Architecture

D. Dynamic Load Balancing Using Agents

A group of Proxy Servers are connected together in the form of a ring called Local Proxy Server Group [LPG]. A set of clients (users) are connected to each Proxy Server. This group of local Proxy Servers is connected to the Central Multimedia Server [CMS] through fiber optic cables. One of the Proxy Servers in the LPG acts as a coordinator and maintains a database which contains the information of the videos present in each Proxy Server in that LPG and also the popularity of the videos in that LPG[3].

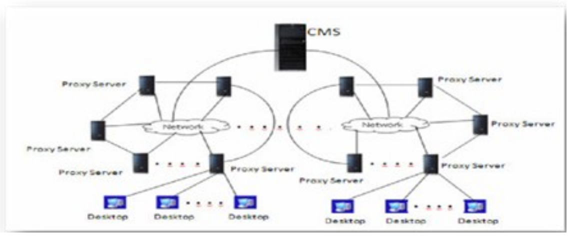

Fig. 4- Dynamic Load Balancing using agents

Performance of Algorithms

Performance of the Algorithms is as follows:

A. Dynamic Load Balancing Using Agents

First, all the N videos are arranged based on their popularity. We select the first m videos from the popularity based sorted list and stored in each proxy server. The remaining videos are stored depending on the local popularity in the proxy servers. When a request for a video arrives at the PS, the following cases happen:

- The requested video is present at the Proxy Server (PS): Then the real time transmission of the video starts immediately
- The requested video is not present at the Proxy Server, but is present either in Left neighboring Proxy Server (LPS) or Right neighboring Proxy Server (RPS), Then we check the number of channels allocated for popular videos bandwidth LPS & PS and CMS & PS. If more numbers of channels are allocated for popular videos bandwidth LPS & PS, then path LPS-PS is selected, otherwise the path CMS-PS is selected. The same is done if it is found in RPS
- The requested video is present in both LPS and RPS : Then we find the shortest path between LPS-PS, RPS-PS, CMS-PS and start the transmission.

- The requested video is not present in both LPS and RPS, but is present in one of the Proxy Servers in that LPG. Then we find the optimal path between local proxy server and the proxy server having the content
- The requested video is not present in any of the Proxy Servers in the LPG. Then the path CMS-PS is selected.

B. Global Wait Queue Load Sharing Algorithm

Global Wait Queue(GWQ) is an algorithm for load sharing in a distributed system of servers. It maintains all servers pending requests in a global queue from which idle servers obtain additional jobs. The implementation of GWQ requires two queues on each server: a Local Queue for requests generated by customers attached at the server(Local requests) and a Remote Queue for requests received from other servers fully loaded (remote requests). We describe the main characteristics of the GWQ algorithm:

- A server generates a remote request only if it receives a local request and is fully loaded, that is, has not enough resources to attend this request.
- The first priority of every server is to service requests from the Local Queue. Requests from the Remote Queue are attended only when the Local Queue is empty.
- Each request is serviced by the same server throughout its lifetime.

C. Popularity and Partial Replication Load Sharing

The implementation of Popularity and Partial Replication Load Sharing(PRLS) requires two queues on each server: a Local Queue for requests generated by customers attached at the server (local requests) and a Remote Queue for requests received from other servers (remote requests). Next we describe the main characteristics of the PRLS algorithm:

- A server generates a remote request when it has not enough resources to attend this request or when it receives a request to a non local video. In both cases, the server dialogs with the other servers to assign the remote service, but (due to the fact of partial replication) only with the servers that maintain a copy of the requested video. To this end, we assume that each server knows the localization of each video in the system through a localization table. We suppose that this table is fixed, that is, there is no video migration.
- The first priority of every server is to service requests from the Local Queue. Requests from the Remote Queue are attended only when the Local Queue is empty.
- Each request is serviced by the same server throughout its lifetime**.**

D. Adaptive Viewers Bias Based Algorithm

Adaptive viewers bias based algorithms replace the videos from the Continuous media server (CMS) efficiently to make space for new videos.

The mode of operation is as follows:

The metadata DB contains the priority decay rate, the initial priority level for each CMO(Continuous Media Object). Each time a CMO is loaded from an AS(Archive Server) to a CMS(Continuous Media Server) , the latter inserts for the newly cached CMO at the CMS, an entry comprising its identifier, its priority decay rate and its initial priority level[12]. Besides, for each cached CMO in the CMS, the termination times of old requests are recorded. The CMS, in addition, stores in each CMO entry the cumulative priority decay rate and the cumulative initial priority level, which is used to calculate the priority of the CMO at the current time. When a CMO just terminates playback for the current request, its cumulative priority decay rate and its cumulative initial priority level are modified.

To select which CMO in the CMS to be the victim, the following steps are undertaken:

- If the CMS has enough free disk space, the CMO is simply loaded from the AS.
- Else, each CMO entry in the CMS is traversed, and the CMO's priority is calculated .The CMO with the lowest priority is the candidate to replace and is thus marked can be deleted.
- Repeat previous step until enough disk space is available to accommodate size of newly requested CMO.
- If enough space has been freed, the CMO's marked can be deleted are removed and the requested CMO is loaded. If enough disk space cannot be freed, the newly requested CMO cannot be loaded in this CMS for playback and the marked CMO's are unmarked.

E. Least Connection First Load Balancing Algorithm

In this algorithm there is a cluster of servers which r allocated work by the load balancing server. Whenever user requests for a video the load balancing server redirects the requests to the server having least no of active processes running or none at all. If there are multiple numbers of servers at the same state i.e. running the same no of processes the load balancing server allocates requests in a round robin fashion. In short :CF algorithm searches for the server having the least number of active processes running and allocates or delegates work to them.

Differentiating Parameters

Following are the various differentiating parameters :

A. Nature of Load Balancing Techniques

Static load balancing assigns load to nodes probabilistically or deterministically without consideration of runtime events. It is generally impossible to make predictions of arrival times of loads and processing times required for future loads.

On the other hand, in a dynamic load balancing the load distribution is made during run-time based on current processing rates and Network condition.

i. Dynamic Load Balancing Using Agents: In this technique the load is assigned to nodes on runtime. The local proxy server searches for the video in its own database and if not found finds the best possible path and redirects the video from there to the client. Hence here the load balancing technique thus used is dynamic in nature.
ii. Global Wait Queue Load Sharing Algorithm: In this technique if the video is not present in the local server or if the local server is busy the request is inserted into the global queue from where the idle servers process it. As all this happens at runtime it is also dynamic in nature.
iii. Popularity and partial Replication Load Sharing: In this technique based on the popularity of the videos the videos are removed from the list to make space for new videos which are requested by the user. As this also happens on runtime it is Dynamic in nature

iv. Adaptive Viewers Bias Based Algorithm: Similar to PRLS this algorithm uses user's bias as a means to determine which video needs to be removed from the list to make space for new ones. This happens at runtime so is Dynamic in nature.

v. Least Connection First Load Balancing Algorithm: This algorithm decides which server to be allocated on receiving the request for video, considering the active processes in the various servers. As this is done only during runtime this is also dynamic in nature.

B. Resource Utilization

Resource utilization include automatic load balancing A distributed system may have unexpected number of processes that demand more processing power. If the algorithm is capable to utilize resources, they can be moved to under loaded processors more efficiently.

i. Dynamic Load Balancing Using Agents: This method uses resources optimally by utilizing maximum channels for popular videos between the client and the source server and also by finding the best possible path between the source and the client[18].

ii. Global Wait Queue Load Sharing Algorithm: This algorithm uses the resources of the local server and the other coupled servers by delegating work to them when the local server is busy.

iii. Popularity and Partial Replication Load Sharing: This algorithm uses resources efficiently by replicating popular videos selectively to increase the throughput and reduce the waiting time for the user. Also selective replication makes maximum use of disk space and avoids wastage which happens due to storage of less popular videos.

iv. Adaptive Viewers Bias Based Algorithm: This algorithm keeps or deletes videos from the archive servers' database based on user's demands. This algorithm constantly monitors user's demands and based on them keeps on modifying the contents of archiving servers.

v. Least Connection First Load Balancing Algorithm: This algorithm delegates works to least busy servers thereby increasing the total work output and increasing the overall efficiency of system.

C. Cooperative

This parameter gives that whether servers share information between them in making the process allocation decision other are not during execution. What this parameter defines is the extent of independence that each server has in concluding that how should it can use its own resources. In the cooperative situation all servers have the accountability to carry out its own portion of the scheduling task, but all servers work together to achieve a goal of better efficiency. In the non-cooperative individual servers act as independent entities and arrive at decisions about the use of their resources without any effect of their decision on the rest of the system.

i. Dynamic Load Balancing Using Agents: In this method all the proxy servers interact with each other continuously and whenever there's a need of video which is not present in the local server all the proxy servers in the group work together to find the most optimal solution to the problem. Thus they are highly cooperative.

ii. Global Wait Queue Load Sharing Algorithm: In this method each of the servers gives priority to the local clients' request more than the global client request. All the resource allocation decisions are made by the individual server himself. After completing its work it decides to take in work of some other servers.

iii. Popularity and partial Replication Load Sharing: In this method similarly like Global wait Queue Load Sharing Algorithm the servers first concentrate on getting their local work done before helping other servers with their requests

iv. Adaptive Viewers Bias Based Algorithm: In this the archiving servers work in total cooperation with each other to increase the throughput and efficiency.

v. Least Connection First Load Balancing Algorithm: This is the most cooperative system because all the work is delegated to servers in a fashion to increase the throughput. The load balancing server considers the state of the servers and allocates work to the server having least number of active processes.

D. Centralized and Decentralized

Centralized schemes store global information at a designated node or a set of nodes at a central location. All receiver nodes access the designated node to calculate the amount of load-transfers and also to check the videos that are to be sent. In a distributed load balancing, every node executes balancing separately. The idle nodes can obtain load during runtime from a shared global queue of processes.

i. Dynamic Load Balancing Using Agents: In this a proxy servers are spread over a region and incase the video which is demanded is not found request is routed to the central server. In short the scheme used is distributed.

ii. Global Wait Queue Load Sharing Algorithm: In global wait queue load sharing method the servers are loosely coupled and they are distributed in nature. They after completing processes allocated to them obtain load during runtime from a shared global queue of processes.

iii. Popularity and partial Replication Load Sharing: Similar to global wait queue load sharing algorithm the servers are loosely coupled and share work independently and share the load from a global process queue.

iv. Adaptive Viewers Bias Based Algorithm: It has a centralized architecture consisting of many continuous media servers which keep on transmitting the videos that are demanded. In case a copy of video that has been demanded does not exist in the CMS then Archiving servers are requested to transfer a copy. In short the entire architecture is centralized and depends upon the AS and cluster of CMS.

v. Least Connection First Load Balancing Algorithm: Even in this method there is a centralized cluster of servers which is delegated work by the load balancing server depending upon the status of the servers. It follows centralized architecture scheme.

E. Overload Rejection

If Load Balancing is not possible additional overload rejection measures are needed. When the overload situation ends then first the overload rejection measures are stopped. After a short guard period Load Balancing is also closed down.

i. Dynamic Load Balancing Using Agents: The overload rejection is not implemented by this algorithm. In case of a request which cannot be processed the server will cater to it till the server crashes.
ii. Global Wait Queue Load Sharing Algorithm: In this method as each processor works Independently and for each of them the local processes are of higher priority overload rejection is taken care of.
iii. Popularity and partial Replication Load Sharing: Similar to global wait queue algorithm the server can handle the overload rejection very well.
iv. Adaptive Viewers Bias Based Algorithm : This algorithm handles overload rejection very well as every time a request is received the CMS searches for the video in its internal library but before starting transmission also checks for the available bandwidth which if available transmission begins. In case of insufficient system resources the request is rejected or is kept waiting till resources are freed up.
v. Least Connection First Load Balancing Algorithm: Least connection first algorithm allots requests based on their sizes to the servers which are free or have the least amount of active requests pending. In case of overloading request the request will be rejected by the load balancing server itself as it concludes that processing it would not be possible for any of the video servers

Conclusion

In this paper, we conducted a survey on the existing Load Balancing Techniques used in Video on Demand services. We described several key Load Balancing Algorithms like Dynamic Load Balancing Using Agents, Global Wait Queue Load Sharing Algorithm, Popularity and partial Replication Load Sharing, Adaptive Viewers Bias Based Algorithm, Least Connection First Load Balancing Algorithm. Current deployments on the Internet demonstrate that load balanced VOD systems are capable of streaming video to a large user population at low server cost. However, there are several fundamental limitations of existing load balanced video streaming methods. First of all, the users Quality of experience in current systems are still not comparable to the traditional TV services provided by cable and satellite broadcasting companies. Sometimes streaming users experience much longer channel start -up and channel delays. Video playback starts tens of seconds after a user selects a channel. There are also large playback lags. Consequently, users only receive low resolution videos. In addition, the video streaming quality is poor and unstable when the number of viewers watching the same program is small as these algorithms tend to replace them with more popular ones. This makes it challenging to serve long-tailed unpopular contents in such streaming systems. Those issues are interesting and challenging research problems that need to be addressed to make load balanced video streaming services a real competitor for traditional broadcast TV services. Out of the various techniques discussed LCF is the one that is most efficient when used for a centralized set of servers as the work delegation and response time is most efficient in this case whereas in case of a distributed system where the servers are distributed over a huge area popularity and partial replication load sharing or load balancing using agents technique can be used.

References

[1] Sonia Gonzalez, Angeles Navarro, Juan Lbpez, Emilio Zapata (2002) *Multimedia and Expo*.

[2] Liang-Teh Lee, Hung-Yuan Chang, Der-Fu Tao and Siang-Lin Yang, *International Journal of Grid and Distributed Computing.*

[3] Guruprasad H.S., Maheshappa H.D. (2008) *International Journal of Computer Science and Security*.

[4] Sonah B., Ito M. (1998) 6*th International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems IEEE*.

[5] Thouin F. and Coates M. (2007) IEEE- Network, 42-48.

[6] Thouin F., Coates M. and Goodwill D. (2006) *IEEE Network Computing and Applications*.

[7] Wang B., Sen S., Adler M. and Towsley D. (2004) *Multimedia IEEE,* 366-374.

[8] Kim K. and Park S. (1999) 5*th International Computer Science Conference on Internet Applications*, 373-378.

[9] Cheng C., Lee M. and Oyang Y. (1997) *Consumer Electronics, IEEE Transactions*, 1220-1228.

[10] Tsai C., Edward T., Chu H. and Huang T. (2004) *Fourth International Conference on Embedded Software*.

[11] Huang Y. and Huang J. (2004) *Computers, IEEE Transactions,* 77-82.

[12] Kim K., Hwang J., Lim S., Cho J. and Park K. (2003) International Parallel and Distributed Processing Symposium 124.1.

[13] Venkatramani C. and Chiueh T. (1995) *ACM SIGCOMM*'.

[14] Chiueh T., Venkatramani C. and Vernick M. (1997) *Software - Practice & Experience*, 27(2), 139-154.

[15] Vernick M. (1995) "*Design and Implementation of the Stony Brook Video Server", PhD Thesis, Computer Science Department, State University of New York at Stony*.

[16] Venkatramani C. (1996) "*Design, Implementation and Evaluation of RETHER:A Real-Time Ethernet Protocol", Ph.D. Dissertation, Computer Science Department, State University of New York at Stony*.

[17] Niranjan T., Chiueh T. and Schloss G. (1997) IEEE International Conference on Multimedia Computing Systems.

[18] Lee B., Cao P., Fan L., Phillips G. and Shenker S. (1999) *INFOCOM- IEEE.*

[19] Chakchai S. (2005) *A Survey of Proxy Caching Mechanisms for Multimedia Data Streams", Project Thesis, Department of Computer Science, Washington University*.

[20] Tang K., Chan S. and Wong E. (2000) *IT-IEEE*, 1, 672 - 676.

[21] Tang K., Chan S. and Wong E. (2001) *IEEE Transactions on Industrial Electronics*, 48(5).

[22] Wolf J., Philip S. and Shachnai H. (1997) *Multimedia Systems*, 5(6), 358 - 370.

[23] Birk Y.(1995) *Fourteenth IEEE Symposium on Mass Storage Systems*.

[24] Dan A. and Dinkar S. (1995) *SIGMOD*, *ACM*.

[25] Xugang Y., *"A Heuristic Method for A Rostering Problem with The Objective of Equal Accumulated Flying Time", MSc thesis, Florida state university college of arts and sciences*.