LARGE-SCALE MULTIMEDIA STORAGE SYSTEMS (LMSS) FOR LOAD BALANCING

R.ARULSELVI M.E, S.ELAMATHI(M.E)
Dept of cse,Muthayammal Engineering college,India
Arulselvi1985@gmail.com,selvarajelamathi@gmail.com

***Abstract*-A large-scale multimedia storage system (LMSS), which consists of a single server or multiple servers, manages the storage and retrieval of multimedia data from disk media. As a networked multimedia service is expected to serve a large pool of clients, it is impossible for a single server to meet all multimedia requirements such as continuous-time presentation, network bandwidth, etc. In a large-scale multimedia storage system where client requests for different multimedia objects may have different demands, the placement and replication of the objects is an important factor, as it may result in an imbalance in server loading across the system. Generally each object storage server responds to the requests that come from the centralized servers independently and has no communication with other OSSs in the system. But in this distributed load balancing strategy for LMSS, in which OSSs can cooperate to achieve higher performance, the OSS system can replicate the objects to and balance the requests among other servers to achieve a near-optimal average waiting time (AWT) of the requests in the system.**

Keywords- **Multimedia Storage System, Request Balancing, Distributed System, Average Waiting Time, Queuing theory**

I. INTRODUCTION

A large-scale multimedia storage system (LMSS), which consists of a single server or multiple servers, manages the storage and retrieval of multimedia data from disk media. Servers can generally be classified in two large categories: Centralized and Distributed. In a centralized system, a high-end scheduler monitors, serves, and manages the video streams among the servers. Distributed system workstations or PCs can work as servers and they cooperate with limited local information to achieve high system performance.

In LMSS if some servers remain idle, and others are extremely busy, then the system performance will drastically be affected. Load balancing is a technique to spread work between two or more computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, throughput, or response time. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. The balancing service is usually provided by a dedicated program or hardware devices.

To prevent this, load balancing is often used to distribute the request and improve performance measures, such as called average waiting time (AWT), which is the time difference between the time instant at which a request arrives to the system and the time instant at which the client starts receiving the first package of the multimedia system

In the literature, pertaining to LMSS, by and large, the issues are dealt in an isolated fashion. This means that studies would independently focus on balancing the requests, storing /replicating the media files, and optimal disk/storage utilization problems to quote a few. In this paper, we focus on designing a load balancing or PCs constituting an LMSS. We take a radically different approach in considering the problem in a collective manner and devise a distributed load balancing strategy to determine the placement and number of replications for the media objects. In addition, we attempt to balance the requests among the servers to achieve a minimized AWT of the requests.

A .*Dynamic Load-Balancing Strategies*

A fundamental issue affecting the performance of a parallel application running on message-passing parallel systems is the assignment of tasks to processors in order to get the minimum completion time (Senar.M.A 2004). Static mapping tools are used to do initial assignment of tasks on professors while dynamic balancing tools are used at execution time. Static task assignment is known to be NP-complete and, therefore, the obvious approach is to develop polynomial time algorithms that provide fast and near optimal solutions.

The aim of the mapping and dynamic load-balancing tools is to minimize the execution time of parallel programs on distributed memory machines by controlling the use of computation and communication resources.

The Storage Load Balancing is just a general load balancing problem in computer systems. It has storage controllers/processors, which act simultaneously and number of storage units which are known as volumes connected to those storage processors. The goal is to distribute those volumes, among the storage processors in order to achieve better performance.

Load balancing is a possible guiding principle for resource allocation, whereby the load is allocated across locations as evenly as possible. The network consists of a number of base stations and users. The required communication channels for users are available at the base stations. Each station may serve the users within its geographical range.

A reasonable allocation strategy for dynamic load balancing is the Least Load Routing (LLR) policy, which assigns each arriving consumer to a location with the least load in the associated neighborhood.

B**.** *A Distributed Media Server for the support of Multimedia Teaching*

One major problem of using multimedia material in lecturing is the trade-off between actuality of the content and quality of the presentations. A frequent need for content refreshment exists, but high qualities presentations can not be authored by the individual teacher alone at the required rate (Micheal Liepert 2012).

Several past and current projects have had the goal of developing so-called learning archives, a variation of digital libraries. On demand, these deliver material with limited structure to students. For lecturing, these systems provide an insufficient service as the unreliable WWW.

One of the goal of the project is the creation of an infrastructure for the support of multimedia-enhanced teaching in schools and universities. Especially for the education in technical areas, it is today hardly possible for the individual teachers to keep their teaching material continuously up-to-date, if this teacher works for himself and does not exchange material with others. An education offer that is not up-to-date anymore, however, will not be accepted by the students.

Using the rapidly evolving presentation tools of the World Wide Web for access, the project will create a distributed system through integration of the approaches for production, distribution and consistency maintenance, which will grow from a central node that operates as a production center and archive for the end user through a flexible, decent rally organized system of regional nodes.

The future teaching support system is envisioned to support the creation of presentations from raw material rather than being only a material archive for lessons. In order to do this, it is relevant to note that material can rarely be used without modifications.

C**.** *Design of a Large Scale Multimedia Server*

Large scale multimedia storage servers will be an integral part of the emerging distributed multimedia computing infrastructure (Buddhikot 2006). However, given the modest rate of improvements in storage transfer rates, designing servers that meet the demands of multimedia applications is a challenging task that needs significant architectural innovation.

The primary requirements of future network based large scale storage servers are

- Large number of clients,
- Large storage capacity,
- Large network and storage system bandwidth,
- Real time service,
- Cost-effective.

In addition to high network and storage throughput, a multimedia server may have to provide a lot of computing power to support real time media processing.

D. *SAN Storage Provisioning*

The entire process of planning and serving the SAN storage is called Storage Administration. Storage Administration consists of two phases: storage capacity planning and storage provisioning (Krasimir Miloshev 2012).

1) *Storage capacity planning*

In order to do precise capacity planning, the following parameters are figured out. I/O load, disk load and number of disks.

2) *Storage provisioning*

Once the capacity planning is done and certain storage capacity is provided, the next step for the storage experts is to optimize the usage of that capacity in order to keep best performance in our SAN. The usage of that capacity is implemented via storage provisioning requests. Optimizing storage provisioning tasks with regard to the best SAN provisioning practices means to distribute the requested storage load the best possible way from performance perspectives. One of the best known approaches would be searching for optimal load balance between the storage processors.

II. PROPOSED SYSTEM STRUCTURE

A. *Overview*

In LMSS as shown in figure 3.1, if some servers are idle, and others are extremely busy, then the system performance will be drastically affected. To prevent this, load balancing is often used to distribute the requests and improve the performance by reducing average waiting time (AWT). AWT is the time difference between the time instant at which a request arrives to the system and the time instant at which the client starts receiving the first package of the multimedia stream. In the proposed strategy, the workloads of load balancing among all the OSSs in the system are distributed and let the MDSs focus on searching for the locations of the requested objects only. In the proposed system the OSS model is based on M/G/m probability system. Each OSS can determine the replications of the objects and distribute the requests.

B. *Balancing the Requests among the Replication*

Some replications for object O_i existing in the system, attempt to balance the requests for O_i among the OSSs. The primary OSSs may not be the close to other OSSs with a replication, and there may be multiple links along the path between them.

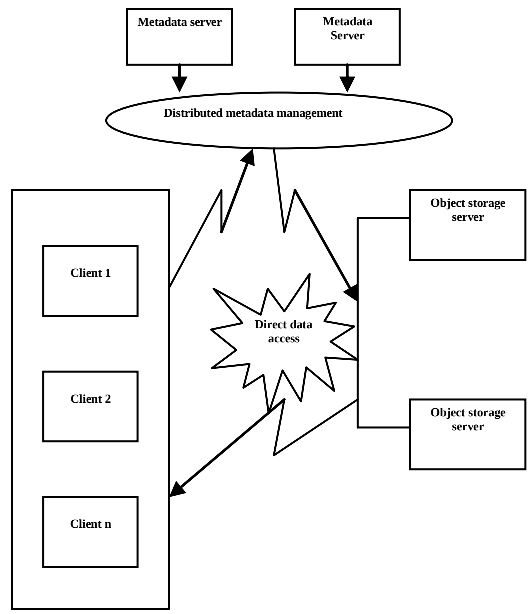

Fig. 1 System Architecture of our proposed LMSS

C. *Load Balancing Strategy*

Initially some objects are assigned to a server, and this server becomes their primary server. All the requests for these objects come to the primary server first, and the server takes charge of replicating the objects and distributing the requests among the system. Since it is a distributed system, the load balancing solution can be computed offline. Each server can obtain some system information, which can be computed offline, and can operate to achieve a global optimal solution.

D. *Object Replication Determination*

In order to obtain an optimal combination of servers to minimize the AWT function and to test every potential combination of the servers and find an optimal set of v_i for every object o_i. This incurs a complexity of 0((N-

3

1)!) for each o_i, and it is impractical in real- life situations. To handle this overly complex scenario, and propose a heuristic method to determine an optimal set of v_i for object o_i in s_j. In the worst case, the complexity of our method is 0(N log N).

E. *Performance Evaluation*

Multimedia servers are connected by an Intranet or Internet. It is assumed that the queue discipline of the requests in each server is FCFS. Multimedia servers can be normal PCs or workstations, and the system can be homogeneous or heterogeneous. Each object is assigned to a server by a simple hashing function using the object ID as the key, and the server will become the primary server of the object. In order to guarantee the real-time delivery requirement of the continuous media, the servers, have to provide each request with a minimum bandwidth for playback.

III. CONCLUSION

A distributed load balancing strategy has been proposed for large-scale multimedia storage systems, which consist of a large number of OSSs. In traditional distributed file systems, request balancing is handled by the centralized MDSs, and the MDSs are the potential bottlenecks in achieving higher performance. In our proposed strategy, distribute such workloads of load balancing among all the OSSs in the system and let the MDSs focus on searching for the locations of the requested objects only. In this model of OSS as an M=G=m system, based on which formulate our objective function for the AWT of the requests that arrive at the system. Each OSS can determine the replications of the objects and balance the requests for these objects among their replications independently according to the near-optimal solution of the objective function.

IV. REFERENCES

[1] M.M Buddhikot "Design of a large scale multimedia server" , Proceedings of the IEEE Compcon Volume 9, no 5 , February 2006 , Pages 663.1 – 663.10.

[2] Carsten Griwodz, Michael Liepert, Giwon On, Michael Zink, Ralf Steinmetz "A distributed media server for the support of multimedia teaching" , IEEE transaction on Computer Graphics and Applications, Volume 15 , No 7 , August 2004 , Pages 1 – 12 .

[3] Krasimir Miloshev "SAN Storage Provisioning with External Load Balancing Mechanism" , IEEE transaction on Knowledge and Data Engineering , Pages 79 – 86 , Volume 58 , No 4 , June 2007.

[4] Senar.M.A, Ripoll.A, Cortés.A and E. Luque.E " Evaluation of Mapping and Dynamic Load-Balancing Strategies in SEPP-HPCTI" , IEEE transaction on Parallel and Distributed systems , Volume

[5] Bharadwaj Veeravalli and Zeng Zeng "On the design of distributed object placement and load balancing strategies in large-scale networked multimedia storage systems", IEEE transaction on Knowledge and Data Engineering , Volume 20 , No 3 , March 2008, Pages 369-382 .

42 , No 8 , September 2004, Pages 1 – 14 .

[6] Z.. Zeng and B. Veeravalli "Design and Analysis of a Non-Preemptive Decentralized Load Balancing Algorithm for MultiClass Jobs in Distributed Networks " , Computer Comm. , Volume 27 , 2004, Pages 679 – 694 .

[7] A.T. Stephanos and D. Spinellis " A Survey of Peer – to – Peer Content Distribution Technologies " , ACM Computing Surveys , Volume 36 , No 4 , December 2004 , Page 335 – 371 .

[8] K.Shen, T. Yang and L. Chu cluster support and replication management for scalable network services , IEEE Trans. Parallel and distributed systems, Volume 47 , No 3 , November 2003.

[9] L.Jie and Kameda "Load Balancing Problems for Multiclass Jobs in Distributed/ Parallel Computer Systems", IEEE Trans. Computers, Volume 47 , No 3 , March 1998 , Page 322 – 332 .

[10] L.F.Zeng, D.Feng, F.Wang and .Zhou "Object – Replication and Migration Policy Based on OSS " , Proc. Fourth Int'l Conf. Machine Learning and Cybernetics (ICMLC ' 05), August 2005.

