行政院國家科學委員會專題研究計畫　成果報告

如何設計擴充性佳的連續性媒體系統

計畫類別：個別型計畫
計畫編號：NSC91-2218-E-002-034-
執行期間：91年12月01日至92年07月31日
執行單位：國立臺灣大學資訊工程學系暨研究所

計畫主持人：周承復

計畫參與人員：周承復.紀婉容.林光龍

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 92年10月30日

行政院國家科學委員會專題研究計畫成果報告

如何設計擴充性佳的連續性媒體系統

Design of Scalable Continuous Media Systems

計畫編號：NSC 91-2218-E-002 –034-
執行期限：91 年 12 月 1 日至 92 年 7 月 31 日

主持人：周承復　　台灣大學資訊工程系
e-mail: ccf@csie.ntu.edu.tw
http://www.csie.ntu.edu.tw/~ccf

一、中文摘要

關鍵字：連續性媒體伺服器、擴充性、可靠性、門檻式方法、廣域資料帶解、負擔平衡、服務品質、動態重製

多媒體應用（如視訊傳輸、數位博物館及遠距教學系統等等）對其所儲存媒體系統與傳輸網路上的服務品質（QoS）、系統效能與可靠性有相當程度的要求。除此之外，如此嚴苛的需求更使得如何設計一個低成本又具擴充性的連續性媒體（Continuous media，CM）伺服器更加困難。因此，本計畫試著提供一個能快速動態調整的資源管理技術，輔以資料配置技術，以求能增進整個系統的效能、擴充性與可靠性。

對一個大規模、分散式的多媒體伺服器而言，資料配置技術攸關著整個系統有效使用資源的能力。在本計畫中我們根據採用折衷資料配置方案，也就是說在一個節點（是指分散式的多媒體伺服器之一）內使用廣域資料帶解(wide data striping) 的方式以解決負擔量失衡的問題；而在節點間應用資料複製（replication）的技術。

為了系統可以及時因應當資料存取模式的改變，我們將透過門檻式方法（Threshold-based）以及簡單且有效率的數學統計模型來判定(a)系統應在何時調整物件重製的數目？(b) 如何調整? 等問題。經由我們的結果也顯示我們的方式可以使的整個分散式多媒體伺服器系統的負擔平衡、使用率提高並且提供良好的服務品質。

Abstract

Keywords: Continuous Media Server, Scalability, Reliability, Threshold-based approach, Wide-data striping, Load balance, Quality of Service and Dynamic replication.

Multimedia applications (such as video stream delivery, digital libraries, and distance learning systems) place high demands for quality-of-service (QoS), performance, and reliability on storage servers and communication networks. These, often stringent, requirements make design of cost-effective and scalable continuous media (CM) servers difficult. Therefore, the main focus of this project is to provide efficient adaptive and dynamic resource management techniques in conjunction with data placement techniques in order to improve performance, scalability and reliability of such systems.

In a large-scale distributed CM server, the choice of data placement techniques can have a significant effect on the ability of such system to utilize resources efficiently. In this project, we choose a hybrid data placement according our previous research results. That is, Instead of striping each object across all the nodes of the system, we take a hybrid approach and constrain the striping to a single node while replicating popular objects on several nodes in order to provide sufficient bandwidth capacity to service the demand for these objects.

In addition, in order to react more frequent changes in data access patterns, we plan to address (1) when the system should alter the number of copies of a CM object through a threshold-based approach, and use dynamic replication policies in conjunction with a mathematical model of user behavior to address how to accomplish this change. Our results show that the hybrid data placement with dynamic replication techniques result in a load-balance, cost-effective and better QoS Continuous media system.

二、Introduction & Objective

With the rapid growth of multimedia applications, there is a growing need for a large-scale continuous media (CM) servers that meet user demand. Multimedia applications (such as video stream delivery, digital libraries, and distance learning systems) place high demands for quality-of-service (QoS), performance, and reliability on storage servers and communication networks. The scalability of a CM server's architecture depends on its ability to: (a) expand as user demand and data sizes grow; (b) maintain performance characteristics under degradation of system resources; and (c) maintain performance characteristics under growth or re-configuration. In particular, the choice of data placement techniques can have a significant effect on the scalability of a CM server and its ability to utilize resources efficiently. In this project, we consider a hybrid approach, where the main focus on the tradeoffs between striping and replication.

However, some interesting and important questions rise such as (i) how many copies of each CM object should the system maintain, (ii) on which nodes these copies should be placed, and (iii) (possibly) how to migrate users from one node to another, in mid-stream, in order to admit more users through adjustments to current load allocations. More frequent changes in data access patterns lead to the following additional important questions: (1) when the system should alter the number of copies of a CM object, and (2) how to accomplish this change.

Threshold-based Approach & Dynamic Replication

We use a threshold-based approach to decide when the system should adjust the number of copies of an object and some dynamic replication policies in conjunction with a mathematical model of a user behavior to accomplish replication of an object. Since the number of copies of object i partly determines the amount of resource available for servicing requests for that object, we adjust the number of replicas maintained by the system dynamically. In other words, we use a threshold of an object to represent its popularity to decide if the current resource of object i in the system is enough to meet the current customers' requests. If not, the system will trigger a replication of object i until there are sufficient number of copies for object i.

To the best of our knowledge, previous research works do not consider alternative design characteristics that affect the scalability of CM servers in an end-to-end setting under changes in access patterns (i.e., taking into consideration both the network and the storage resource constraints). Moreover, in this project, we not only do a quantitative study of such issues and the cost/performance and reliability characteristics but also consider the control overhead of different methods to collect and update the threshold for each object i, i.e., how to predict the data access pattern and their effect on the system performance.

Different Methods to Predict Data Access Pattern

We consider three methods to predict the data access patterns. The first one is (i) LA: the latest inter-arrival time of an objects, i.e., $p_i(t) = 1/(t - at_i)$. (ii) AR: first order autoregressive function of the inter-arrival time of an object, i.e. $p_i(t) = 1/(\alpha(t-at_i) + (1-\alpha)p_i(t-1))$. (iii) SS: simple access statistics within a fixed window T_w(time interval). For instance, if window size is 60 min, we compute the $p_i = N_i/T_w$, the popularity of an object, where the Ni is the total number of access for object i within the past 60 min.

*Global vs. Local Control Protoco*l

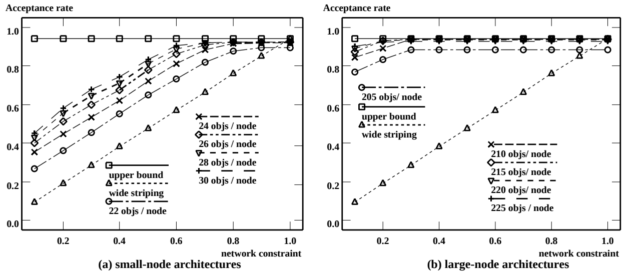
In this project, we survey two types of control protocols of collecting the information to predict data access patterns. One is each node just use its local information to update the threshold of object i. The other one is the system will collect all the information from all sub-nodes to a master server and use that information to update the threshold for each object i.

三、Results Summary

In this section, we will consider two types of hybrid systems: (i) one is called small-node architecture, i.e., each node has small service capacity including storage space and local switch size. (ii) the other one is called large-node architecture, i.e., each node has large service capacity. Therefore, if the total service capacity is the same for both small-node and large-node architectures, the small-node system has more nodes than a large-node system does.

Wide Data Striping System vs. Hybrid System

Under lower network capacities, a hybrid system[1] has better overall performance as well as performance degradation characteristics than the wide data striping system. That is, we could tradeoff capacities of various system resources in a hybrid system in order to achieve a more cost-effective system overall. Specifically, the system can tradeoff some local storage space ad local switch capacities with global switch capacities and achieve nearly the same performance characteristics. Even when we take the control overhead into consideration, the hybrid system still performs better than the wide data striping system.



(a) small-node architectures (b) large-node architectures

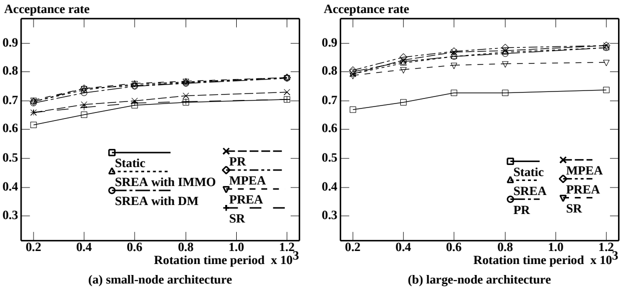
Fig. 1 wide data striping vs. hybrid systems

Reliability

We use the mean time to failure (MTTF) as our reliability metric, defined as the mean time until some combination of disk failures results in loss of some data that can no longer recovered through the use of redundant information. Our results clearly show that higher reliability can be achieved by the hybrid systems, even for objects that only a single copy, as compared to wide data striping. Moreover, the MTTF in our any hybrid system is 2 to 20 times longer than the wide data striping system. This increases in reliability is due to the isolation of fault affects, i.e., the wider we strip an object, the more disk failures can affect it.

Static vs. Dynamic Replication

According to our extensive simulations, our results (as depicted in fig.2) show that dynamic replication with early acceptance does result in significantly better performance as compared to static replication policy although there is little overhead for bookkeeping and communication between different nodes. Therefore, we conclude that dynamic replication schemes with threshold-based approach could make good decisions about: (i) which object is popular and (ii) what is the right time to trigger a replication for such popular object. Moreover, it is worth keeping little history information within a node since the needed storage space and communication bandwidth overhead is small compared to those needed by a continuous media object.

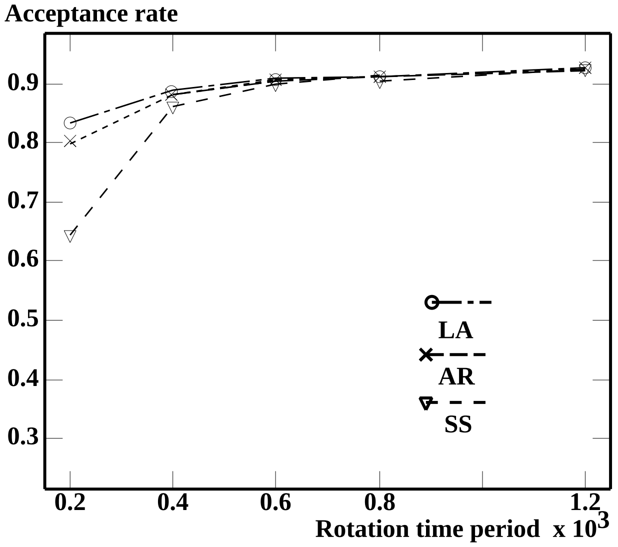


(a) small-node architecture (b) large-node architecture

Fig. 2: static vs. dynamic replication

Different Predicted Methods

We would like to investigate the effect of different data collection methods to predict the access patterns on the system performance. In general, when the rotation time, i.e., the time between two different data access patterns, is smaller, either we use the latest access time or its first order of autoregressive function for object i to represent its popularity works better than the static simple statistic access method. However, when the rotation time becomes large, all three methods have almost the same performance as shown in fig.3 as below.



(a) small-node architecture

Fig.3 comparison of different predicted methods

Global vs. Local Control Protocol

Since the control overhead compared to the storage and communication bandwidth needed by a CM object is pretty small, the centralized control protocol performs better than the local control protocol, i.e., in our testing cases, the performance improvement could be up to 37%. However, when the node capacity increases (large-node architecture), the local control protocol has a comparable performance as that of the global control protocol.

四、Conclusion

From the above discussions, we know the use of the hybrid design allows us to tradeoff resources and this should helps a system designer to make better system sizing decisions by making appropriate tradeoffs. In addition, this hybrid design with dynamic replication scheme result in a higher system reliability, system acceptance rate and very low control overhead. Therefore, we believe that these techniques could be applied to a wide range of applications of continuous media servers.

五、Bibliography

[1] S. Berson, L. Golubchik, and R.R. Muntz, “Fault tolerant design of multimedia servers”, in Proc. of the ACM SIGMOD Conf. on Management of Data, San Jose, CA, May 1995, pp.364-375.

[2] Bolosky et al., “The Tiger video fileserver”, Technical Report MSR-TR-96-09, Microsoft Research, 1996.

[3] M.-S Chen, D.D. Kandlur, and P.S. Yu, “Support for fully interactive playout in a disk-array-based video server”, in Proc. of 2nd ACM intl. Conf. on Multimedia, Oct. 1994, pp.391-398

[4] A.L. Chervenak, “Tertiary storage: An evaluation of new applications”, Ph.D. Thesis, UC Berkeley, 1994.

[5] A. Dan and D. Sitaram, “An online video placement policy based on bandwidth to space ratio (BSR)”, in Proc. of ACM SIGMOD, 1995.

[6] A. Dan, M. Kienzle, and D. Sitaram, “A dynamic policy of segment replication for load-balancing in video-on-demand severs”, ACM Multimedia System, Vol. 3, pp. 93-103,1995

[7] S. Ghandeharizadeh and R.R. Muntz, “Design and implementation of scalable continuous media servers”, Parallel Computing Journal, pp. 123-155,1998

[8] R.L. Haskin, “Tiger Shark: a scalable file system for multimedia”, Technical Report, IBM Research, 1996

[9] K.D. Jayanta, J.D. Sallehi, J.F. Kurose, and D. Towley, “Providing VCR capacities in large-scale video servers”, in Proc. ACM Intl. Conf. on Multimedia, 1994, pp. 25-32

[10] P. Lie, J.C. Lui, and L. Golubchik, “Threshold-based dynamic replication in large-scale video-on-demand systems”, Multimedia Tools and Applications, Vol. 11, No. 1, pp. 35-62, 2000

[11] N. Venkatasubramanian and S. Ramanathan, “Load management in distributed video servers”, in Proc. of ICDCS, Baltimore, MD, May 1997, pp. 528-535

[12] J. Wolf, H. Shachnai, and P. Yu, “DASD dancing: A disk load balancing optimization scheme for video-on-demand computer

systems", in ACMSIGMETRICS/Performance Conf., 1995

[13] B.Wang, S.Sen, M.Adler and D.Towsley, "Optimal Proxy Cache Allocation for Efficient Streaming Media Distribution" To appear in Proceedings of IEEE INFOCOM, 2002.

[14] O. Verscheure, P. Frossard and J. Boudec, "Joint Smoothing and Source Rate Selection for Guaranteed Service Networks", INFOCOM, 2001, pp. 613-620.

[15] Soam Acharya and Brain Smith, "MiddleMan: A Video Caching Proxy Server", Proceeding of NOSSDAV, June 2000.

[16] P. Frossard and O. Verscheure, "Batched Patch Caching for Streaming Media", IEEE Communication Letters, 6(4), April 2002.