DESIGN AND IMPLEMENTATION OF A MULTIMEDIA PERSONALIZED SERVICE OVER LARGE SCALE NETWORKS

Xiaorong Li, Terence Hung Gih Guang

The Institute of High Performance Computing
Division of Advanced Computing
1 Science Park Road, Singapore 117528
Email: {lixr,terence}@ihpc.a-star.edu.sg

Bharadwaj Veeravalli

The National University of Singapore
Dept. of Electrical and Computer Engineering
10 Kent Ridge Crescent, Singapore 119260
Email: elebv@nus.edu.sg

ABSTRACT

In this paper, we proposed to setup a distributed multimedia system which aggregates the capacity of multiple servers to provide customized multimedia services in a cost-effective way. Such a system enables clients to customize their services by specifying the service delay or the viewing times. We developed an experimental prototype in which media servers can cooperate in streams caching, replication and distribution. We applied a variety of stream distribution algorithms to the system and studied their performance under the real-life situations with limited network resources and varying request arrival pattern. The results show such a system can provide cost-effective services and be applied to practical environments.

Key words: Video distribution, cache capacity, link bandwidth, time constraint, stream caching.

1. INTRODUCTION

Multimedia personalized services (MPSs) [1] are increasingly becoming promising technologies to provide customized services to the subscribers. Clients connected to the multimedia servers in the network can order a large amount of multimedia documents (MMDs) under this service facility at any favorable times and with affordable prices. In addition, such services can provide a complete flexibility in presentation by controlling the playback, i.e., clients can alter the presentation by using Video Cassette Recorder (VCR)-like control facilities such as rewind, fast-forward, pause, etc.

The concept of providing *multimedia personalized services* (MPSs) according to client preferences was firstly mentioned in [1, 2]. In the literature [1], the idea of personalized multimedia services was introduced to consider managing the resources on the network (such as bandwidth, storage, cache capacity) by reserving network resources. Clients submit requests with their requirements (e.g. the name of certain Multimedia Documents (MMDs) and the playback time, etc.) and the system decides either to accept or drop the requests depending on the availability of resources. Once a request is accepted, system should guarantee the service under its requirement. Such a mechanism is to provide user-level QoS [3], where a service "agreement" is settled between the client and the service provider before the service begins.

In this paper, a prototype of a network-based multimedia system is designed and implemented to render multimedia services according to client requirements. To guarantee the QoS, our experimental system thrives to manage the network resources to distribute streams under time-constraints. Our focus is geared towards designing a large scale distributed multimedia system and *maximize* the number of users and *minimize* the total caching cost and transmission cost of servicing user requests demanding a set of heterogenous video streams. We examine the performance of a variety of algorithms and study the combined effects of different algorithms in term of various influencing parameters concerning the network environments.

The remainder of the paper is organized as follows. Section 2 describes our system model and formulates the problem. Section 3 describes the hardware and software of our experimental system. In Section 4, experiments are conducted to examine the performance of various stream distribution strategies. Section 5 concludes the paper.

2. SERVICE MODEL AND PROBLEM FORMULATION

We envisage the multimedia system as a generic network, which comprises multiple Video Warehouses (VWHs) connected by communication links with limited bandwidth as shown in Fig. 1. Each VWH is primarily repository of Multimedia Documents (MMDs) or videos with finite cache capacity. Clients submit their requests demanding certain MMDs to be played at future viewing times. To guarantee QoS, the system should ensure that the requested MMDs will be played at the requested viewing times; Otherwise, the requests will be dropped. Fig. 2 describes the interaction among clients, VWHs and the scheduler.

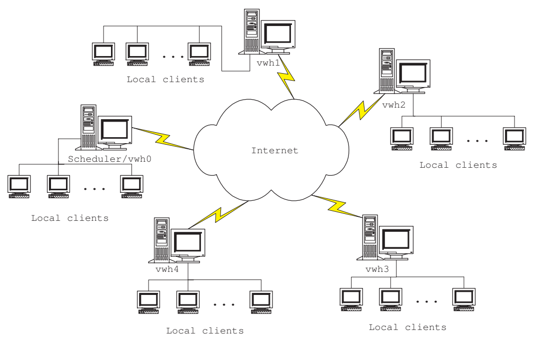

Fig. 1. A distributed multimedia system

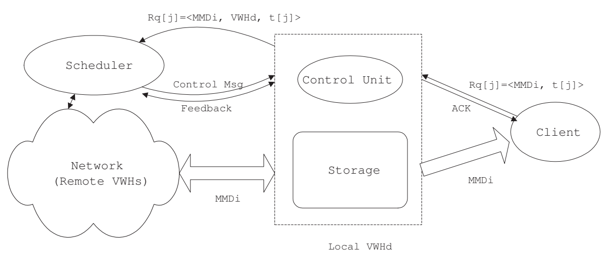

Fig. 2. Interaction among system components

2.1. Some Definitions

1. *Time constraint*: This is defined as the time interval between the arrival of request and its requested viewing time. For example, given a request $Rq[j] =< MMD_i, vwh_d, t[j] >$, which arrives at time $t_{arrival}[j]$ and demands to view MMD_i on vwh_d at viewing time $t[j]$. Thus, the time constraint T is given by,

$$T = t[j] - t_{arrival}[j], \qquad (T \geq 0) \tag{1}$$

Time constraint T is no less than 0, and $T = 0$ means the requested MMD should be played immediately as the the request arrives.
2. *Service cost*: Sum of the transmission cost and the caching cost along the paths from the source VWHs to the destination VWHs [4, 2]. We assume that the bandwidth in LANs is wide enough and the cost for delivering MMDs from destination VWHs to clients is not considered in this paper.
3. *Acceptance ratio* and *Average cost per request*: The performance evaluation is based on the two fundamental metrics, *Acceptance ratio* (α) and *Average cost per request* ($\overline{C}$). They are defined as

$$\alpha = \frac{No.\ of\ requests\ successfully\ served}{Total\ no.\ of\ requests} \tag{2}$$

and

$$\overline{C} = \frac{\sum_{j=0}^{k-1} C[j]}{k} \tag{3}$$

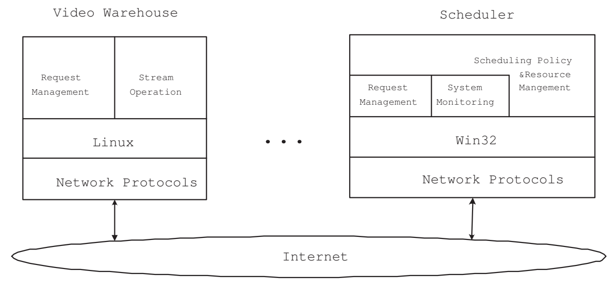

Fig. 3. Software architecture of the experimental system.

where $C[j]$ is the service cost for j-th request and k is the total number of requests successfully served.

2.2. Problem statement

Given a topology $G = (V, E)$, where V is a set of VWHs and E is the set of directed links in the system, let $Rq = \{Rq[0], Rq[1], \cdots, Rq[n], \cdots\}$ be a sequence of requests, which request to view MMDs in T minutes. We seek a feasible schedule S that maximizes the acceptance ratio defined by (2) and minimizes the total cost defined by (3) while satisfying the constraints on the amount of cache space available on VWHs, link availability, and the reserved bandwidth for transmitting MMDs, respectively.

3. EXPERIMENTAL SYSTEM

3.1. Hardware and software

In our experimental system, video warehouses (VWHs) are media servers, which are in charge of executing stream operations, such as transmitting, receiving, caching video streams. The scheduler is the central manager of the system and it acts as an information and command center which generates commands for each VWH to serve requests according to different scheduling algorithms. In our system, the scheduler allocates and reserves network resources by sending command to respective VWHs. These commands are sent out from the scheduler to VWHs by sockets supporting TCP/IP protocols. The hardware configuration of the system is listed in Table 1.

Fig. 3 shows the software architecture of our experimental system. The functions of media server VWHs include two modules: *Requests Management*, and *Stream Operation*. Each of them is based on multi-threaded programs. Requests Management module is mainly in charge of collecting the local requests to pass on to the central scheduler. The Stream Operation module is responsible for executing the commands of the scheduler such as transmitting, receiving, caching, and flushing video streams. This module is programmed as a multi-threaded entity to enable each VWH to receive/transmit media streams simultaneously. The system was made to support a variety of media formats of MMDs, including Motion

Table 1. Hardware Configuration.

	CPU	MEM (MB)	Hard Disk (GB)	OS	Number
VWH	Pentium 3	256	150	Linux	14
Scheduler	Pentium 4	1000	30	WinXp	1

Picture Experts Group (MPEG-2, or MPEG-4), Audio-Video Interleaved (AVI), MPEG Audio Layer-3 (MP3) and WAV files. Moreover, the functions of error detection, retransmission and error statistics are available for high-precise stream transmission/receiving which depends on the clients' requirements.

The scheduler acts as a central information and command center which generates commands for each VWH to render media services cooperatively. The program on scheduler consists of three modules: *Resource management* & *Scheduling process*, *Command management*, and *System monitoring*. Scheduling algorithms are being run in the module of resource management & scheduling process.

3.2. Stream Distribution Strategies

3.2.1. Stream Distribution algorithms

In real-life situations, network resources (link avasilability and cache space) are limited in amount. To achieve higher acceptance ratio, multicast algorithms (e.g., SPT[5], SBS [6], DBS [6]) consider constructing cost-effective multicast trees spanning distributed VWHs so as to optimize the global service cost of distributing media streams. Shortest Path Tree (SPT) algorithm constructs shortest-path multicast trees and delivery MMDs as soon as possible to the destinations. In Source-based Stream Scheduling (SBS) algorithm, the searching for a cost effective path begins from a primary source and it terminates after reaching the destination; Destination-based Stream Scheduling (DBS) algorithm [6] considers constructing a cost-effective path inversely from the destinations to the nearest source.

3.2.2. Video Partitioning

Video partitioning algorithms (e.g., UVP [7], WAVP [7]) consider partitioning video streams and caching media objects partially on the caches. The advantages of video partitioning is that it can save the cache space, reduce the bandwidth consumptions, speed up the setup delay, avoid frame jitters, and etc. In Window-Assisted Video Partitioning Strategy (WAVP) [7], videos are partitioned into portions of a same size and video portions are cached and flushed according to adaptive windows so as to optimize the service cost associated with the network bandwidth and the cache space. The frequency of flushing a video portion is determined by the values of the delay bound T, the available cache space, and the prices for stream caching and transmission.

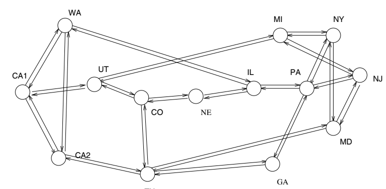

Fig. 4. Logical topology of the experimental network.

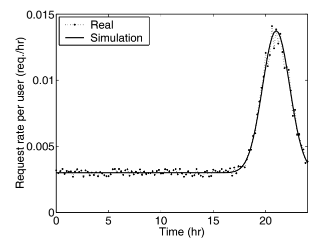

Fig. 5. Request rate per user over 24 hours.

4. EXPERIMENTAL RESULTS AND ANALYSIS

Parameters

Our experimental system is based on the NSF network, which consists of 14 nodes and 21 links representing a generic communication network connecting 14 main cities in US, as shown in Fig. **??**. Each node is equivalent to a VWH, supporting 5000 local clients in sub-networks. The cost for caching an MMD on each VWH is assumed to be $\$0.015$ per-min and the cost for transmitting an MMD via a link, varies from $\$1.5$ to $\$27.0$ depending on the distance between different nodes in the network. Our system has 100 MMDs and all the MMDs are of size 1.5 GB and of length 100 minutes. The popularity of each MMD is set according to Zipf. distribution [8] with a skew constant 0.271.

Pattern of request arrival

We consider a request rate reported in [9], see Fig. 5, where clients submits requests to VWH in a 24-hour period. Requests arrive at each VWH in *Poisson*, and each request arrives at least T minutes before its viewing times.

Results and analysis

For performance comparison, we try to reflect the average effect of all the algorithms. Every point on each graph represents the average values taken when the system runs for more than 30 days. The performance evaluation is based on two metrics: *Acceptance Ratio* and *Average Service Cost*. To examine the performance in a 24-hour period, we sample 48

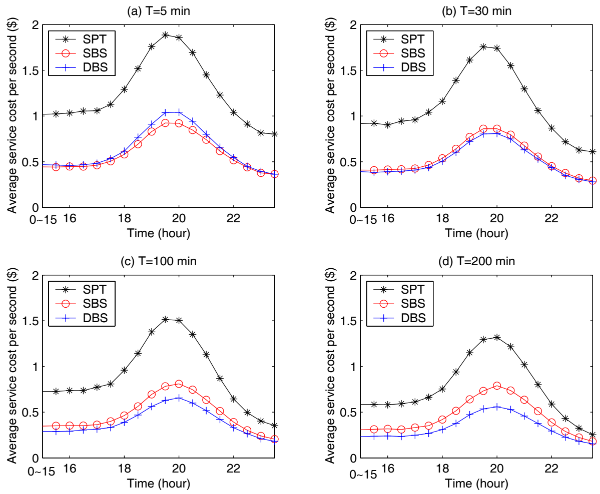

Fig. 6. Performance comparison of different algorithms. System parameters: Cache=100%, B_a=100Mbps.

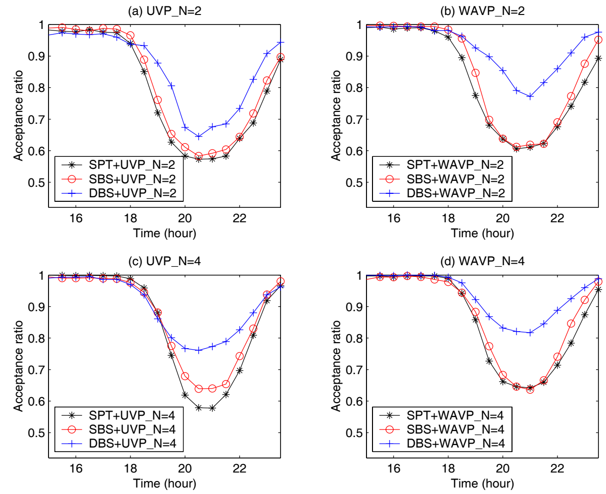

Fig. 7. Acceptance ratio of different algorithms over 24-hour. System parameters: T = 5 min, Cache=20%, B_a=50Mbps.

data points over a 24-hour period and the points of acceptance ratio and average service cost in the figures represents the average value over every 30 minutes.

Fig. 6 (a) to (d) compare the average service cost of various algorithms under various time constraint T. When $T = 5$ as in Fig. 6 (a), algorithm SBS has lower service cost than DBS; When T increases from 30 minutes to 200 minutes as shown in Fig. 6 (b) to (d), the average service cost of DBS is lower than that of the other algorithms. This is because DBS achieves the cost efficiency by receiving the MMDs as late as possible; however, it may cause more bandwidth consumption when the time constraint is too tight.

Fig. 7 (a) to (d) compare the acceptance ratios of different algorithms with respect to different video partitioning strategies, where N denotes how many portions that an MMD is partitioned. When DBS joints with WAVP, the acceptance ratio can be much improved. This is because WAVP takes advantage to update each portions due to their popularity, length, and positions in an MMD, which improves the unitization of both cache space and link bandwidth.

5. CONCLUDING REMARKS

In this paper, we developed a large scale experimental prototype of network based multimedia system which consists of a pool of media servers, a schedule server, and clients. During the 24 hours in a day, the request arrival rate may vary from time to time. We examined the performance of various algorithms and their combination effects under the real-life situations with limited network resources and varying request arrival patterns. The experiment results show that such a system can provide cost-effective services and be applied to practicable environments.

6. REFERENCES

[1] C.H. Papadimitriou; S. Randmanathan, P. Venkat Rangan, and S. Sampathkumar, "Multimedia Information Caching for Personalized Video-on-Demand," *Computer Communications*, Vol. 18, No. 3, pp. 204-216, Mar. 1995.

[2] Y.J. Won, and J. Srivastava, "Strategic Replication of Video Files in a Distributed Environment," *Multimedia Tools and Applications*, Vol. 8, No. 2, pp. 249-283, Mar. 1999.

[3] Abdelzaher, T.F.; Shin, K.G.; Bhatti, N.; "User-level QoS-adaptive resource management in server end-systems", *IEEE Trans. Computers*, Vol. 52, No. 5, pp. 678-685, May 2003.

[4] Bharadwaj Veeravalli, "Network Caching Strategies for a Shared Data Distribution for a Predefined Service Demand Sequence," *IEEE Trans. Knowledge and Data Engineering*, Vol. 15, No. 6, pp. 1487-1497, Nov.-Dec. 2003.

[5] M.D. Theys, Min Tan, N.B. Beck, H.J. Siegel, and M. Jurczyk, "A Mathematical Model and Scheduling Heuristics for Satisfying Prioritized Data Requests in an Oversubscribed Communication Network," *IEEE Trans. Parallel and Distributed Systems*, Vol. 11, No. 9, pp. 969-988, Sep. 2000.

[6] Xiarong Li, and Bharadwaj Veeravalli, "Design and Performance Analysis of Multimedia Document Retrieval Strategies for Networked Video-on-Reservation Systems," *Computer Communications*, Vol. 28, No. 17, pp 1910-1924, 2005.

[7] Xiaorong Li and Bharadwaj Veeravalli, "A Novel Stream Partitioning Strategy for Real-time Video Delivery in Distributed Multimedia Systems," *Proc.IEEE Consumer Communications and Networking Conference*, pp. 313-318, Jan 2005.

[8] Serpanos, D.N., Karakostas, G., and Wolf, W.H., "Effective caching of Web objects using Zipf's law," in *Proc. IEEE ICME*, Vol. 2, pp. 727 -730, Aug. 2000.

[9] PG de Haar et al., "DIAMOND project: Video-on-demand system and trials", *Eur Trans Telecommun.* (8)4:337-244, 1997.