A Video Replacement Policy based on Revenue to Cost Ratio in aMulticast TV-Anytime System *

Xiaobo Zhou   Cheng-Zhong Xu
Department of Computer Science
Department of Elec. & Computer Engineering
Wayne State University
Detroit, MI48202, USA
{zbo, czxu}@ece.eng.wayne.edu

Lars O.Burchard   Reinhard Lüling
Paderborn Center for Parallel Computing(PC2)
University of Paderborn
D-33102, Paderborn, Germany
{baron, rl}@upb.de

Abstract

This paper analyzes a tree hierarchical network architecture employing video caching and multicasting capacity to support a large scale Video-on-Demand service called TV-Anytime. The host servers are connected to other host servers which collectively store all the videos in the system. The proxy servers are located close to customers and are able to store the most popular videos in an adaptive way. Considering many uncertainties in the future demands for videos, we propose an on-line video replacement policy based on a revenue to cost ratio, with an objective of maximizing the overall revenues generated by the system during the runtime. Simulation results show that this policy leads to an efficient TV-Anytime system and makes the system more adaptive to changes in video popularity which is typically the case in the TV industry. The simulation results also show that multicasting significantly improves the system throughput during the high-load periods and makes the system more scalable.

1 Introduction

In recent years, Video-on-Demand (VOD) applications have moved from concept to reality due to major advances in disk, network and data compression technologies. TV-Anytime is a VOD application that aims to realize interactive and on-demand video services [1, 2]. Its main idea is to record broadcasted television programs and make them available independently from the time when they are broadcasted.

In large scale Video-on-Demand applications, a central server for storage and delivery of broadband video streams keeps administrative effort low but has an obvious disadvantage of limited scalability, concerning both network and storage resources. In fact, the network-I/O bottleneck has been observed in many VOD applications [3, 4]. In earlier work, we proposed a tree hierarchical network architecture to support a large-scale TV-Anytime system [5]. As to the network structure, the convenience of considering tree hierarchical networks is related to the need to reduce the costs through at least partial use of the existing network infrastructure. The server network, as depicted in Figure 1, is managed by a distributed server management system called DSMS [6]. The root of this tree architecture is referred to as a center server. The center server usually is large enough to store all the available videos. The servers reside between the root and leaf level are called host or backup servers. A host server is connected to other host servers which collectively store all the videos in the system. The servers at the leaf level of the hierarchy are referred to as proxy or cache servers. They are used to reduce the network traffic

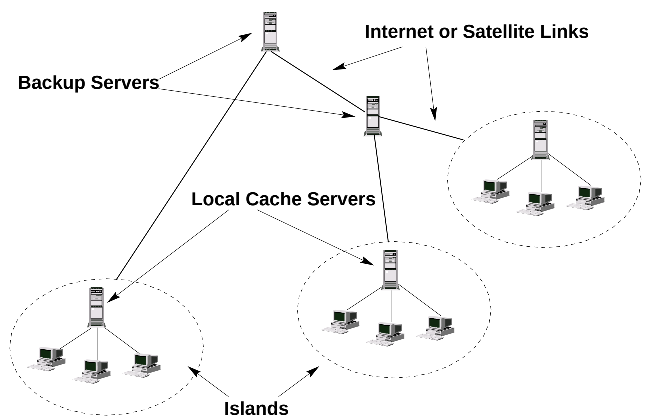

Figure 1: Structure of a TV-Anytime Server Network

*This research was partly supported by a grant from Siemens AG, München and Axcent Media AG, Paderborn in the framework of the project HiQoS - High Performance Multimedia Services with Quality of Service Guarantees.

in such a way that the customers are not accessing the center or host servers directly. Proxy servers store a subset of the overall videos and are used as caches for a single *island*. The island is an environment which provides the necessary network infrastructure for streaming videos to a number of customers. This distributed approach is effective in terms of manageability, reliability, QoS quarantees and especially bandwidth saving.

Obviously, the placement of videos in the server network plays an important role in the management of the system. Uploading of videos is a lengthy process. The videos have to be predictively placed onto the server network in such a way that the proxy servers store the videos most likely being accessed from customers within corresponding islands. Thus, each customer may have easy and cheap access to videos stored in the proxy server of its island. Customers inside an island also have access to the videos stored on the servers located in remote islands or backup servers in the backbone network. However, streaming from a remote server to this island is expensive and might be impossible due to insufficient bandwidth availability or network connections that are available only for a certain time such as satellite links. We have formulated this static mapping problem as a combinatorial optimization process and proposed a set of effective solutions for the problem [7]. This predictive placement is based on knowledge about characteristics of videos and customers preferences, e.g. subscriptions of content categories or user profiles. Supposing the availability of a perfect predictor of access patterns, the predictive placement module can hence be performed periodically (e.g., per day). Since customer behaviors change over time, accurate predictions are rather difficult. A successful TV-Anytime system must offer customers a large selection of videos. Some of the videos will be extremely popular and accessed frequently. The most frequently requested videos may change not only on a daily or weekly, but even on an hourly basis. For example, children's videos are likely to be popular early in the evening or in weekend mornings, but less popular late at night. Thus, a TV-Anytime system must be able to adapt rapidly to a widely varying and highly dynamic workload.

This paper documents our efforts in three areas. First, we develop a workload model that may reflect customer behaviors over a 24-hour period. Second, we propose an on-line video replacement policy based on a revenue to cost ratio. Its objective is to maximize the overall revenue that the system can generate. Third, we study the effect of multicasting and the scalability of the system.

2 Related Work

A number of distributed server architectures have been proposed [8, 9]. The authors concentrated on the communication aspects of these distributed server architectures. A primary objective of this paper is to develop methods and policies for the effective management of videos in the hierarchical architecture of a TV-Anytime system. Moreover, in our work the encoding bit rate of a video is supposed to be scalable. It means that the quality of services is also scalable.

A well-known technique to reduce communication traffic is to allow customers to share multicast data. To the end, a number of approaches have been proposed, including Batching, Pyramid Broadcasting, and, perhaps most notably, Skyscraper Broadcasting [10, 11, 12]. Interested readers are referred to these papers for the details. In this paper, we study the impact of batching and multicasting on a large scale video server network, rather than a single video server.

In contrast to our approach, most of the work in the field of multimedia caching does not focus on revenue based replacement strategies. For example, in [13], a prefix caching mechanism was proposed, storing the first few seconds of a video, independent from its popularity. The proposed architecture concentrates on lower bit rate streams in the Internet environment and only takes a single proxy cache into account. In [14], several caching strategies for distributed server systems were discussed. However, the authors did not use any revenue based metrics to evaluate their algorithms. In this paper, we discuss a revenue to cost ratio based replacement strategy for a distributed server system for high-quality video streams.

3 A Multicast TV-Anytime System

In typical proposals for VOD applications, customers are served individually by allocating a dedicated transmission channel and a set of server resources to each customer. This approach leads to an expensive-to-operate, non-scalable system. The resource savings can be higher if the network switches are able to provide multicasting features, as it happens in ATM networks [15]. In fact, the probability of having in the same time period a higher rate of requests addressing the same video increases with the number of customers connected to the server network. In this case, video multicasting can be of interest in a wider geographical area.

We consider a TV-Anytime system that satisfies customer requests using delayed server response and multicast communication. Advantages are gained by *batching* technique in both servers and the network when several requests for the same video occur at nearly the same time. The time window size is referred to *batching interval* [10]. In the servers, requested videos need only be accessed once. In the network, multicast communication can be used to reduce required bandwidth. Note that some interactivity and

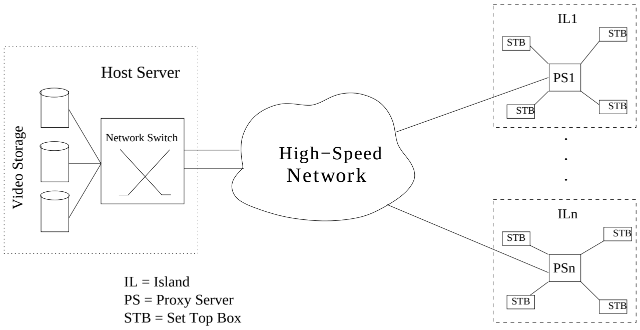

Figure 2: A Multicast TV-Anytime network structure

part of the on-demand nature are sacrificed to achieve cost-effectiveness and other objectives. For the purpose of our evaluation, a basic reference network architecture can be adopted, with a host server connected to different proxy servers. The main components of the architecture, as depicted in Figure 2, include:

- a host server, where videos are stored on hard disk arrays according to different schemes and used as a resource for proxy servers;
- a high-speed core network (e.g. ATM), connecting the host server to proxy servers;
- proxy servers, connecting customer islands to the TV-Anytime system;
- access networks (e.g. ADSL), connecting set-top boxs of customers in islands to proxy servers;
- customer set-top boxs (STB), including decompression and decoding devices, which enable video signals to be displayed at the customer TV. Note that the buffer of STB can also be used to implement limited interactivity.

4 A Video Replacement Policy

4.1 Overview

In the TV-Anytime environment, there are both large videos (e.g., movies) and short videos (e.g., news clips). The system strives to minimize communication cost by caching videos closer to customers in islands. In the predictive placement module, we assume a priori knowledge about the popularity of videos before their uploading to the system. This anticipated popularity is used to decide the residence and encoding bit rates of videos [7]. However, customer behaviors change over time which means accurate predictions are rather difficult. In a large scale TV-Anytime system with a rich selection of videos, some of the videos will be extremely popular and accessed frequently. The video popularity changes not only on a daily or weekly, but even on an hourly basis. Thus, a successful TV-Anytime system must be able to adapt rapidly to a widely varying and highly dynamic workload.

Hence, during the interval between two consecutive predictive placements, whenever customer behaviors change leading to a modified access pattern of their proxy server, an adaptive video reallocation is necessary to improve the utilization and performance of the system. Due to a large uncertainty in the future demands for videos, the replacement of videos has to be performed on-line and adaptively. We check it every time that a video is requested by the proxy children of a host server between two consecutive predictive placements. Based on the accumulated access frequencies for the videos in the current time frame, we consider to remove some videos and replace them with some others that are expected to generate more revenues. The objective is to maximize the total revenues of the system. Servicing the requests for different videos requires different resources, but also generates different revenues. Thus, we present the revenue/cost based video replacement policy.

4.2 Characterization of Resource and Cost

We consider a hierarchical server network with k servers, $S_1, S_2, \cdots, S_k$. Every server $S_j, 1 \leq j \leq k$ has a given storage capacity. This server network currently stores a set of videos, $M = \{V_1, V_2, \ldots, V_m\}, \mid M \mid = m$. D_i is the duration of each video V_i in seconds, $1 \leq i \leq m$. Let b_i be the encoding bit rate of video V_i. $D_i \cdot b_i$ is the storage space requirement of video V_i. Furthermore, let R_i denote any request for a video V_i. Let SB_i, NB_i, MB_i and CPU_i denote the storage I/O bandwidth, external network bandwidth, memory buffer space and CPU cycles, respectively, which are necessary to service request R_i for video V_i. Likewise, denote $AvaSB_j$, $AvaNB_j$, $AvaMB_j$ and $AvaCPU_j$ the storage bandwidth, external network bandwidth, memory buffer space and CPU cycles, respectively, which currently are available in the video server S_j.

In the following, we use $Map(V_i, S_j)$ to denote whether or not video V_i corresponding to request R_i is available in a video server S_j.

$$Map(V_i, S_j) = \begin{cases} 1 & \text{iff } V_i \text{ has been mapped onto } S_j. \\ 0 & \text{otherwise.} \end{cases}$$

The first step in the development of the adaptive video reallocation module is to define the cost for servicing a request R_i on each server S_j. Capability of a server to support additional requests for videos is not only dependent on the available resources, but also on the resource requirements for servicing a customer for a video that has been mapped on this video server. These values are video specific. Considering the disk bandwidth, memory buffer, network bandwidth and processing resources of the video server S_j and those required for retrieval and delivery of a video V_i, we define the cost for supporting a request R_i to the video server S_j in terms of its cost function $Cost(R_i, S_j)$ [17]. The cost function is defined as:

$$Cost(R_i, S_j) = Max(\frac{SB_i}{AvaSB_j}, \frac{MB_i}{AvaMB_j}, \frac{NB_i}{AvaNB_j}, \frac{CPU_i}{AvaCPU_j})$$

From the above equation, it is obvious that the cost factor helps to detect critical resources in a video server (i.e., the resources that limit the capacity of the video server). The difference between the different ratios in fact gives an idea of the nature and extent of the performance bottleneck in the video server. Since requests for different videos have different resource requirements, the cost function may vary from request to request. It depends on the type of critical resources in the server. Generally, the lower the cost function of the request, the greater the capacity of the video server to support additional similar requests will be.

4.3 Revenue/Cost based Reallocation

We formulate adaptive reallocation as an optimization process with the objective of maximizing total revenues that a server network is expected to generate. Each request R_i is associated with a revenue Rev_i. For a server S_j, the revenue, due to a service for request R_i relative to other requests, is dependent on: i) the storage space, communication bandwidth and other resources that are available in server S_j and required to store and deliver the stream of video V_i, and ii) the popularity of video V_i, i.e, accumulated access frequencies in server S_j for V_i in the current time frame. By assigning a popular video in server S_j, S_j is likely to satisfy more requests than it is assigned a less popular video. Furthermore, proxy servers may use *batching* technique to reduce communication traffic by allowing customers to share multicast data. The basic idea is to delay the requests for the different videos for a certain amount of time that is defined as *batching interval*, so that more requests for the same video object arriving during the current *batching interval* may be serviced using the same video stream.

By use of the revenue/cost based method for reallocation, the server provides higher priority to requests that generate higher revenue, thereby ensuring that videos reallocation is in such a way that the resources in the server are utilized to generate the maximum revenue.

During the adaptive reallocation process, each server S_j maintains a revenue/cost table. Assume that an absent video V_a is requested by a proxy server S_A. Suppose we somehow located a replica of V_a in host server S_B [1] We consider the communication in the server network takes only place between servers that have a common path from the root to the proxy server. Node S_A, as well as every other node in the path from S_B to S_A, compares the metric of assignment of V_a to itself (if V_a is not available at that node) with the metrics of all videos that are currently resident at that node. The metric of assignment of video V_a to the server S_j is computed as follows:

$$Metric(V_a, S_j) = \frac{Freq_a \cdot Rev_a}{Cost(R_a, S_j)} \qquad 1 \le a \le m, 1 \le j \le k.$$

Let C_j be the currently available storage space in server S_j that resides in the path from S_A to S_B. If $Map(V_a, S_j) = 0$ and $D_a \cdot b_a \le C_j$, video V_a is cached in server S_j. Otherwise, a video replacement takes place in server S_j, only if three following conditions are met: i) $Map(V_a, S_j) = 0$, ii) the $metric(V_a, S_j)$ exceeds the metric of one or more video(s) that are available at the server S_j, iii) the removal of the video(s) that generates less revenue frees enough resources in S_j to accommodate video V_a and support the requests for it. If one of these conditions is violated, video V_a is transferred without caching, i.e., a replica of video V_a is not materialized at the server S_j. Note that we do not account for the overhead of caching, i.e., extra resources.

This revenue to cost ratio based adaptive reallocation process in server S_j is formulated as a optimization problem:

Given: A server S_j with C_j currently available storage space. Let V be the set of videos that are currently stored on server S_j. Consider an absent video V_a, $V_a \notin V$.

Question: Find a subset V' of V, $V' \subseteq V$, such that $C_j + \sum_{i \in V'} D_i \cdot b_i \ge D_a \cdot b_a$ and $\sum_{i \in V'} Metric(V_i, S_j) \le Metric(V_a, S_j)$.

Optimization: $\sum_{i \in V - V' + \{a\}} Metric(V_i, S_j) \rightarrow Maximum.$

The adaptive reallocation module makes video replacement decisions based on the exact arrival time of requests. Note that the real dereplication of a video can only be done after its current services are finished. In order to not impact the performance of the video server, the adaptive reallocation module has to be simple so that the overheads of executing this module are nominal.

We propose a dynamic programming algorithm to solve this optimization problem. Obviously, if the duration of each video is the same, the cost of the algorithm for above optimization is nominal. If their durations are different and are not integers, it seems that the optimization problem is a NP-hard problem like the classic knapsacking problem. Luckily, supposing that the duration of different videos in the system is an integer (in second), the complexity of the algorithm is $O(\mid V \mid \cdot \mid B \mid \cdot D_a)$. B is the set of possible encoding bit rates.

[1]For details on video locating policy in the hierarchical server networks, please refer to our paper [7].

5 Performance Results

5.1 Policies under Study

We analyze the performance of the following video management policies.

- **Policy PP** (Predictive Placement): According to the predicted access patterns, predictive assignment algorithms presented in [7] are used to map the videos to the server network. Subsequently, a customer request is either served immediately, or rejected if its required resources can not be allocated. The time frame between two consecutive predictions is a 24-hour period.
- **Policy LM** (Local Multicasting): The requests for the videos that are currently mapped in the proxy servers are batched and multicasted. However, the requests for videos that are mapped in host servers and hence have to be served by remote accesses are not batched.
- **Policy GM** (Global Multicasting): Not only the requests that are served locally by the proxy servers are batched and multicasted, but also the requests that have to be served remotely via the backbone networks and the host servers are batched and multicasted.
- **Policy AR** (Adaptive Reallocation): Adaptive reallocation is used to reallocate the videos that are currently mapped on the server network, according to their revenue/cost during the runtime of the system.

5.2 Simulation Model

Before we can discuss the video replacement policy for the TV-Anytime system, we have to characterize the workload the server network will support. Unfortunately, there is little reliable information available on the workload for these applications. Many companies are building prototypes to gather such data but are reluctant to share the information because of the fiercely competitive nature of the VOD industry [16]. In the absence of workload measurements and reliable predictions, we need to make an estimate of the workload, which includes predicted sizes for video objects, possible encoding bit rates of video streams for typical compression schemes and estimated of locality of requests based on video store rental patterns.

Table 1 summarizes the workload that will be used in the simulation model in this paper. We assume a MPEG-II compression scheme with the set of possible encoding bit rates $\{2, 4, 6, 8\}$ (Mbits/s). The set of video display lengths is $\{30, 60, 90\}$ (minutes). Based on these data, video capacity ranges from 0.45 GBytes for a 30-minute television program encoded with 2 Mbits/s rate to 5.4 GBytes for a 90-minute movie encoded with 8 Mbits/s rate.

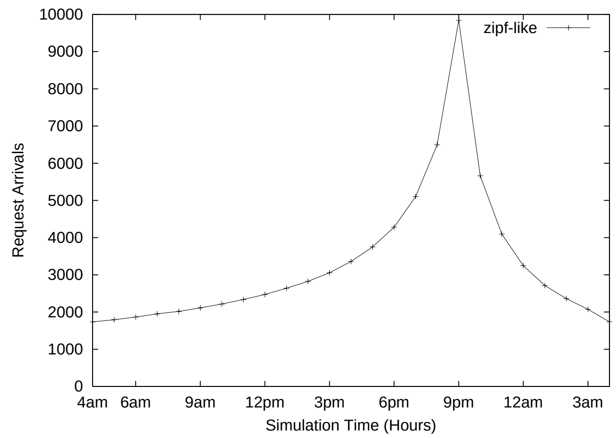

Figure 3: The pattern of request arrivals

Encoding bit rates	$\{2, 4, 6, 8\}$ Mbits/s
Length of videos	$\{30, 60, 90\}$ minutes
Capcity of videos	0.45 to 5.4 GBytes
Locality of accesses	Zipf's Law Distribution

Table 1: Summary of workload characteristics

Further, we assume that all videos are requested in their entirety and played sequentially. We assume that VCR-like operations such as pause, rewind and fastforward can be partly implemented by the buffer of the set-top box. Actually, how to provide interactivity in multicasting Video-on-Demand applications is an interesting issue. For simplicity, we ignore it in this study. The peak period of request arrivals is assumed to be 9pm and the lowest period is assumed to be 4am for each day. Assuming that the request arrivals per hour follow a Zipf-like distribution [18, 19], we calculate the probability of request arrival in hour i to be $p_i = c/(i^{1-\phi 1})$, for $4am \leq i \leq 9pm$; $p_i = c/(i^{1-\phi 2})$, for $9pm \leq i \leq 4am$, where $\phi 1$ and $\phi 2$ are the degrees of skew and are assumed to be 0.4 and 0.2 respectively. The constant $c = 1/(\sum_{4am \leq i \leq 9pm}(1/i^{1-\phi 1}) + \sum_{9pm \leq i \leq 4am}(1/i^{1-\phi 2}))$. Figure 3 depicts a typical request arrival pattern with a total of 80, 000 requests during a 24-hour period.

Based on video store rental patterns, we assume that access to videos will be highly localized, with a small number of videos receive most of the accesses. We assume that predicted popularity of videos follows Zipf-like law with the skew factor of 0.6. The probability of choosing the jth most popular of movies is $P_j = K_n/j^{0.6}$, where $K_n = (\sum_{1 \leq j \leq n} 1/j^{0.6})^{-1}$. Figure 4 depicts a predicted selection of 300 videos via a proxy server during the 24-hour period. Furthermore, we introduce a *pattern skew* to express the difference in percentages between the predicted access patterns and realistic access patterns observed during

Level	Vertices	Servers		Links	
		NIC (Mb/s)	Storage (GB)	edges	NB (Mb/s)
1	V_1={1}	560	270	$\{(i,2i),(i,2i+1) \mid i \in V_1\}$	280
2	V_2={2,3}	800	162	$\{(i,2i),(i,2i+1) \mid i \in V_2\}$	400
3	V_3={4, $\cdots$, 7}	1280	97.2	$\{(i,2i),(i,2i+1) \mid i \in V_3\}$	640
4	V_4={8, $\cdots$, 15}	640	54	ϕ	∞

Table 2: Simulation parameters

the runtime of the system. Figure 5 depicts the realistic selections of videos in the 24-hour period when *pattern skew* is 40%.

We have carried out event-driven simulations of a hierarchical TV-Anytime network. The hierarchical architecture is a 4-level 2-ary complete tree. The simulation parameters are detailed in Table 2. The communication bandwidth of the external networks that connect customers to proxy servers is assumed to be sufficient to stream all requested videos encoded with different bit rates. However, for each proxy server, its realistic communication bandwidth is limited by its Network Interface Card (NIC). For each host server, the bandwidth of NIC corresponds to its backbone bandwidth (NB).

The high cost of on-demand broadband video applications is mostly due to the network costs. For instance, the cost of networking contributes to more than 90% of the hardware cost of the Time Warner's *Full Service Network* project in Orlando. Therefore, we currently assume that CPU, disk bandwidth and memory resources do not constitute bottlenecks. The encoding bit rate of a video determines its requirement of bandwidth resource, and hence determines its cost factor. The server network redundantly stores 300 videos encoded with the different bit rates. The residence and encoding bit rates of these videos are determined by the predictive placement (PP). The root node contains all 300 videos encoded with 2 Mbit/s bit rate.

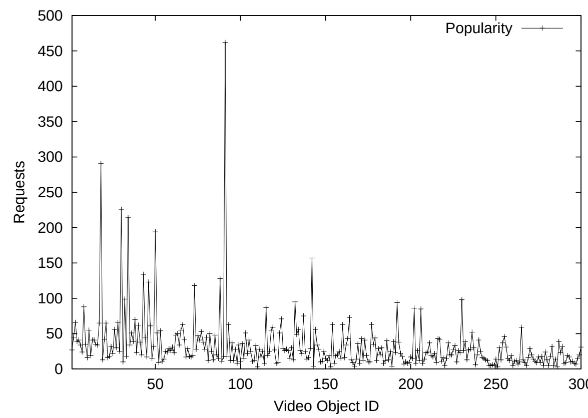

Figure 4: Predicted selection of videos.

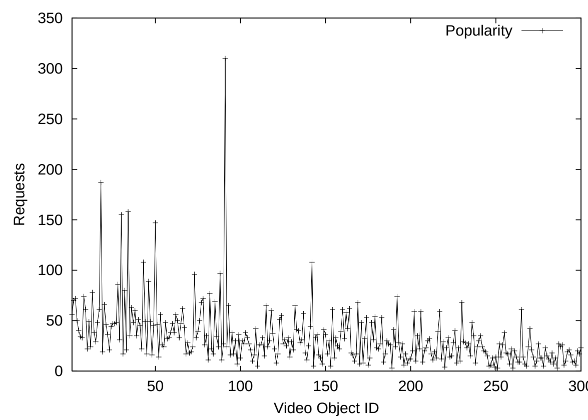

Figure 5: Realistic selection of videos(40% skew).

5.3 Effect of Multicasting

A video can be encoded in different bit rates, and hence provide different presentation quality. It also uses some different storage capacity and communication resources. Moreover, the longer the duration of a video is, the larger its storage space requirement will be. Therefore, we use overall revenues of the system, rather than the ratio of satisfied or rejected requests, as the performance metric of the system. Each request for a video is associated with a revenue. The realistic revenue is dramatically affected by prices and marketing practices. We define the logical revenue by servicing a request with duration 30 minutes and bit rate of 2 Mbit/s is 1. Thus, the generated logical revenue by servicing a request with duration 60 minutes and bit rate 6 Mbit/s is 6.

Figure 6 shows the performance of the predictive placement (PP) with different multicasting capacities. The *batching interval* is 6 minutes. The *pattern skew* is 0, which means that the predicted access pattern perfectly matches the realistic access pattern. During the low load periods, the benefit of multicasting is trivial due to the limited number of requests for the same video arriving in the *batching interval*. Conversely, during the high load periods, the benefit of multicasting is obvious. Without any batching and multicasting, during high load periods the generated revenues of the system is almost a constant. Clearly, the batching

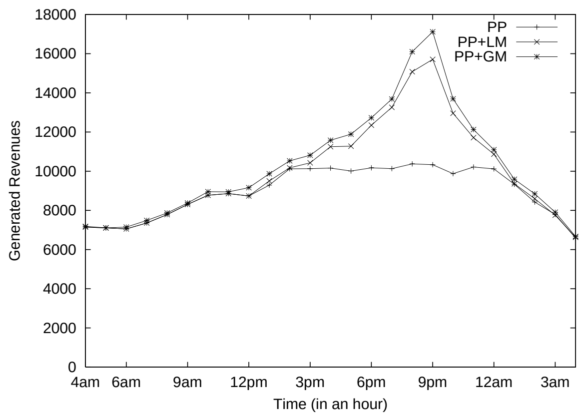

Figure 6: Benefit of multicasting.

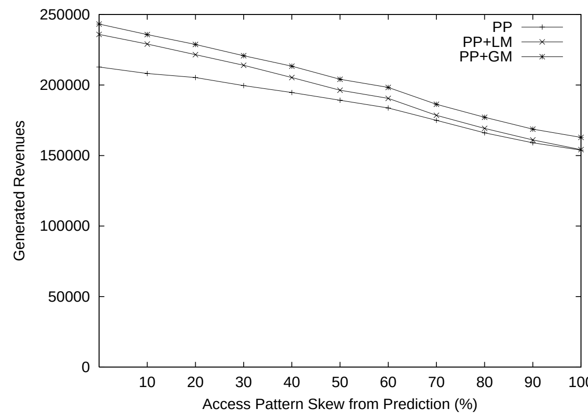

Figure 8: Performance of pattern skews.

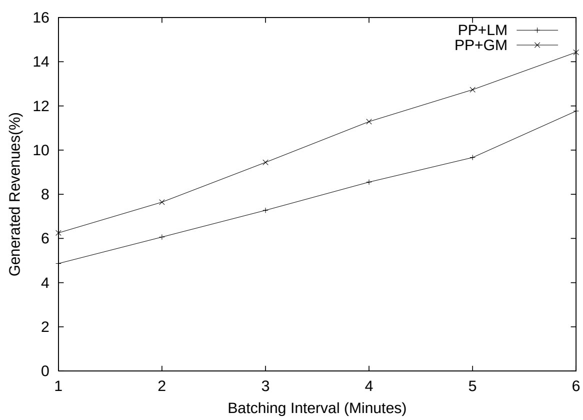

Figure 7: Effect of batching intervals.

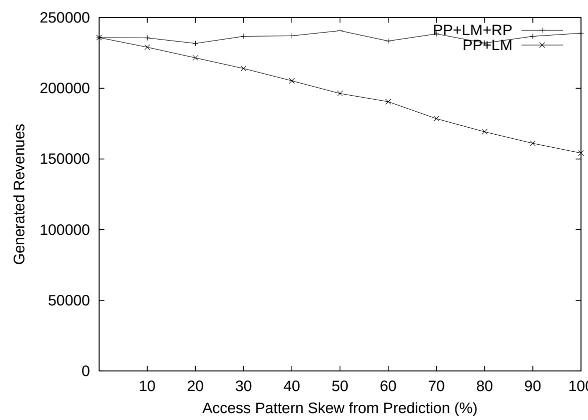

Figure 9: Performance of adaptive replacement.

and multicasting can yield a more scalable system. The small difference between the benefit of *local multicasting* and *global multicasting* exactly demonstrates the desirable effect of the predictive placement algorithm given in [7]. According to the predicted access patterns, more popular videos are mapped with higher encoding bit rates in servers nearer to the proxy servers. The most popular videos are mapped in the proxy servers with the highest bit rates.

Customers making the early requests are likely to renege if they are kept waiting too long. In our simulations, we assume that there is no latency due to the storage systems and communciation networks, and the access latency is due to batching interval. Supposing customers can accept an access latency up to 6 minutes, Figure 7 shows the improved overall revenues (%) by batching and multicasting approaches under different batching intervals (from 1 minute to 6 minutes). It shows the batching effect is almost linear with the batching interval.

5.4 Performance of Replacement

Figure 8 shows the performance of predictive placement without adaptive replacement under different *pattern skews*. The *Batching interval* is 6 minutes. It is shown that the performance degrades considerably along with the increasing pattern skews between predicted and realistic access patterns. Overall generated revenue loses almost 40% when realistic access patterns have nothing to do with the predicted ones. The benefit of local multicasting (LM) diminishes because the popularity of videos originally placed in proxy servers decreases. The realistic popularity of videos originally placed in host servers increases. Thus, the effect of global multicasting (GM) is more obvious.

Since the performance of policy "PP+LM" degrades most drastically with increasing pattern skews, we make use of it to demonstrate the effect of the revenue to cost ratio based replacement (RP). It can be seen in Figure 9 that this revenue/cost based replacement policy enables the system to function more stablely under different pattern skews. The total generated revenue of the system is almost not affected by the dynamically changing access patterns and video popularity that is typically the case in the TV industry.

6 Conclusion

In this paper, we have discussed a revenue to cost ratio based replacement strategy for a distributed server system for high-quality video streams. Simulation results have shown that due to the highly changing customer behaviors, the single predictive placement approach may lead to under-utilization of system resources. It is also showen that the adaptive replacemet approach combined with multicasting can considerably increase the performance of the server network even in case that the realistic access patterns differ considerably from the predicted ones. We conclude that the adaptive reallocation policy can make the system more adaptive to changes in video popularity and system state which is typically the case in the TV industry. Moreover, batching and multicasting techniques significantly improve the system throughput during high load periods and make the system more scalable.

References

[1] TV-Anytime Forum: http://www.tv-anytime.org.

[2] ReplayTV: http://www.replay.com.

[3] T. S. Perry, *The Trials and Travails of Interactive TV*, IEEE Spectrum, Vol 33, No.4, Apr1996, 22-28.

[4] W. J. Bolosky, J. S. Barrera, R. P. Draves, R. P. Fitzgerald, G. A. Gibson, M. B. Jones, S. P. Levi, N. P. Myhrvold, R. F. Rashid. *The Tiger Video Fileserver*, Proc. of the 6th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'96), Zushi, Japan, April 1996, 86-92.

[5] Reinhard Lüling. *Hierarchical Video-on-Demand Servers for TV-Anytime Services*, Proc. of 8th International Conference on Computer Communications and Networks (ICCCN'99), Boston, IEEE Press, Oct 1999, 110-117.

[6] L. O. Burchard and R. Lüling. *An Architecture for a Scalable Video-on-Demand Server Network with Quality-of-Service Guarantees*, Proc. of 7th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS), Springer, LNCS.1905, 2000, 132-143.

[7] Xiaobo Zhou, Reinhard Lüling and Li Xie. *Solving a Media Mapping Problem with Parallel Simulated Annealing*, Proc. of 29th International Conference on Parallel Processing (ICPP'2000), IEEE Computer Society, Toronto, Aug 2000, 115-124.

[8] J. Nussbaumer, B. Patel, F. Schaffa, and J. Sterbenz, *Network Requirements for Interactive Video-on-Demand*, IEEE Journal on Selected Areas in Communications, Vol.13, No.5, 1995, 779-787.

[9] J. Hayes and M. Kerner. *Network Architecture for Multimedia Services*, Proc. of SPIE Advanced Networks and Services Symposium, Amsterdam, Mar 1995.

[10] C. C. Aggarwal, J. L. Wolf and Philip S. Yu. *On Optimal Batching Policies for Video-on-Demand Storage Servers*, Proc. of ACM Multimedia'96, 1996, 253-258.

[11] S. Viswanathan, T. Imielinski. *Metropolitan Area Video-on-Demand Service using Pyramid Broadcasting*. Multimedia systems, Vol 4, No.4, Aug 1996, 179-208.

[12] K. A. Hua, S. Sheu, *Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems*, Proc. of the ACM SIGCOMM'97 Conference, Cannes, France, Sep 1997, 89-100.

[13] S. Sen, J. Rexford and D. Towsley. *Proxy prefix caching for multimedia streams*, Proc. of IEEE INFOCOM'99, April 1999.

[14] A. Dan and D. Sitaram. *Multimedia caching strategies for heterogeneous application and server environments*, Multimedia Tools and Applications, vol.4, May 1997, 279-312.

[15] K. C. Almeroth and M. H. Ammar. *The Use of Multicast Delivery to Provide a Scalable and Interactive VoD Service*, IEEE JSAC, Vo.13, No.6, Aug 1996.

[16] A. L. Chervenak, D. A. Patterson and R. H. Katz. *Choosing the Best Storage System for Video Service*, Proc. of ACM Multimedia'95, San Francisco, 1995, 109-119.

[17] N. Venkatasubramanian, S. Ramanathan. *Load Management in Distributed Video Servers*. Proc. of International Conference on Distributed Computing Systems (ICDCS'97), Baltimore, 1997, 31-39.

[18] G. K. Zipf. *Human Behavior and Principle of Least Effort: an Introduction to Human Ecology*, Addison Wesley, Cambridge, Massachusetts, 1949.

[19] C. Bisdikian and B. V. Patel. *Issues on Movie Allocation in Distributed Video-on-Demand Systems*, Pro.c of the IEEE International Conference on Communications, June 1995, Vol.1, 250-255.