An Online Video Placement Policy based on Bandwidth to Space Ratio (BSR)

Asit Dan and Dinkar Sitaram
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
E-mail:{asit, sitaram}@watson.ibm.com

Abstract

In a video-on-demand server, resource reservation is required to guarantee continuous delivery. Hence any given storage device (or a striping group treated as a single logical device) can serve only up to a fixed number of client access streams. Each storage device is also limited by the number of video files it can store. For the reasons of availability, incremental growth, and heterogeneity, there may be multiple storage devices in a video server environment. Hence, one or more copies of a particular video may be placed on different storage devices. Since the access rates to different videos are not uniform, there may be load imbalance among the devices. In this paper, we propose a dynamic placement policy (called the Bandwidth to Space Ratio (BSR) Policy) that creates and/or deletes replica of a video, and mixes hot and cold videos so as to make the best use of bandwidth and space of a storage device. The proposed policy is evaluated using a simulation study.

1 Introduction

Recent progress in digital video technology has led to large scale multimedia systems that store videos on disks or other storage devices and provide video-on-demand services to many clients over a network [9, 14, 8, 13]. Each storage device is limited not only by the number of concurrent streams it can support (i.e., *bandwidth*) but also by the number of video objects it can store (i.e., *space*). Hence, different devices are characterized by the combination of these two parameters: bandwidth and space. We define a *fast* or a *slow* device as a device with relatively larger bandwidth or space, respectively. A popular video object may be viewed by many clients simultaneously, and hence, multiple replicas of that video may need to be created on multiple storage devices to support a very high bandwidth requirement. In a general environment, there are both large video objects (e.g., long movies) and short video clips (e.g., objects in interactive applications such as shopping, medical, etc.). The viewing frequency of these objects is independent of their sizes, i.e., some movies as well as short clips will be viewed more frequently (and are referred to as *hot* objects) while others are viewed less frequently (and are referred to as *cold* objects).

The video objects have to be placed carefully on these devices in order to avoid load imbalance across devices as well as to achieve maximum utilization of both space and bandwidth of these devices. If large cold objects are placed on a fast device, the higher bandwidth of that device will be unused since only a limited number of objects can be stored by that device. On the other hand, if short hot clips are placed on a slow device, the space on that device will be unused as the lower bandwidth will not permit accessing all objects stored on that device. Therefore, large and short, hot and cold objects have to be mixed properly. These mixes should also match the (bandwidth and space) characterization of these devices (See Figure 1). In this paper, we propose a policy referred to as the *bandwidth to space ratio* or *BSR policy* for achieving this objective. Under the proposed policy, each device is characterized by its bandwidth to space ratio. Similarly, at the time of placement, each video object is characterized by the ratio of the expected bandwidth requirement on that replica (based on the *expected number of concurrent viewers*) to the space required to store that replica. The BSR policy determines how many replicas of a video should be created as well as the devices on which the replicas should be placed. The proposed policy is an online video placement policy, i.e., as the expected number of viewers on a video object changes, it dynamically creates and/or deletes existing replicas to better match the BSRs of the devices and their stored objects. The policy also takes into account many other practical considerations and the details are described in Section 3.

SIGMOD ' 95,San Jose , CA USA

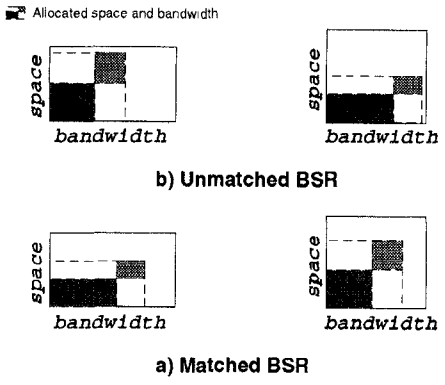

Figure 1: Illustration of the BSR policy

Multiple smaller devices can be grouped together to create a larger logical device where video objects are striped across the physical devices of a striping group. Note that the BSR of this new logical device is the same as the BSR of its component physical devices, since the total bandwidth and space of the logical device is approximately the sum of those of the physical devices.[1] However, a larger striping group can provide a large aggregate bandwidth, and hence, creation of multiple replicas of a hot video object can be avoided. In addition, a single striping group implies all the video objects are placed on a single logical device, and placement becomes trivial. Unfortunately, in a general video server environment all devices cannot be grouped into a single striping group due to various factors [7]. The issues are discussed briefly in Section 2.

The remainder of this paper is organized as follows. Section 2 describes the relationship of this work to other complementary earlier works in the area of design of video-on-demand servers, such as dynamic load balancing. Section 3 discusses the issues in the design of the BSR policy as well as provides an overview. The implementation details are provided in Section 4. In Section 5 using a detailed simulation, the performance of the BSR policy under various workloads is studied. Finally, Section 6 summarizes the results.

2 Related Work

In order to reduce the storage costs of a multimedia system, videos may be stored on tape and staged into disk as necessary [11, 10]. Such systems require a replacement policy that specifies which video on disk to replace if there is no free space. For example, a heat-based policy that replaces segments of relatively unpopular videos with segments of currently referenced videos as needed is studied in [10]. This policy exploits the fact that different videos may have different expected demands [16]. However, staging videos in from tape incurs various overheads and may not be economical for all videos [11]. First, while tape storage reduces the cost of storing videos, it increases the cost of playback since the video has to be read in from tape to disk and then played [11]. For playing large objects expensive tape device need to be occupied for a long time. Even for short video objects, the overhead of mounting the tape and advancing the tape to the correct position may be very high compared to the time required to actually retrieve the object. In addition, the queueing delay for accessing the tape may be significant if enough tape storage are not available or if load imbalance occurs across tape devices. Therefore, if the system cost is dominated by the cost of the resources needed to play back streams, it may be cost-effective to pre-stage popular videos to disks.

The need for multiple striping groups are discussed in detail in [7]. There may be various types of devices, i.e., devices with different BSR, in the same system. It is difficult to use space and bandwidth of all devices effectively by grouping into a single striping group. For example, under a simple striping policy such as round-robin policy, the successive blocks of a video are stored in different disks in order. Here; the blocks of the video files are evenly distributed over the disks. Hence it would not be possible to store more blocks than the capacity of the smallest disk on any disk, leading to wastage of storage space on the larger disks. Alternative striping schemes may be more efficient in their use of storage but lead to imbalances in the load among different disks in the striped set since the blocks are no longer evenly distributed over the disks. Hence, such heterogeneous environments require complex block allocation algorithms that can utilize bandwidth and space effectively as well as respond to the changes in workload. However, for a striping group with a large number of devices, the overhead of implementing such an algorithm at the file system block allocation level may be significant. Thus it is natural to group different types of disks into separate striping groups. Multiple striping groups are desired from other considerations as well, such as the complexity of system reconfiguration (e.g. adding new disks), high replication overhead in order to achieve availability and a complex recovery procedure [2, 3], etc..

Static replication of all or read-mostly files for high availability and also for load balancing has been proposed in the context of file-server [12, 15]. However, this may not be desirable in multimedia systems due to the large size of video objects. Note that placement of video objects is based on expected load, which in general will not be known with certainty. Thus

[1] Achievable realtime bandwidth of a striping group is somewhat smaller than the sum of the bandwidths of the individual devices, since instantaneous load across the devices are not always balanced.

simple heuristics are preferred over complex precise optimizations. Additionally, there will be statistical fluctuations of instantaneous load. A dynamic load balancing policy can shift load from a replica on a heavily loaded device to a replica on a lightly loaded device. However, this is possible only if enough number of video objects of a heavily loaded device is replicated elsewhere.

A dynamic approach that uses partial replication of video files (referred to as the *Dynamic Segment Replication (DSR)* policy) for load-balancing in multimedia systems is proposed in [7]. It is based on the observation that if there were a number of consecutive requests for the same video and the blocks read in by the first request are copied to another device, it would be possible to switch the following requests to the partial replica just created. Such an approach should be viewed as a complementary runtime policy. The DSR policy creates a replica of a hot video only if enough space is available on a device with available bandwidth. In other words, a good initial placement based on BSR matching is required for the DSR policy to handle dynamic load fluctuations. In a real system, some space and bandwidth may be set aside for the purpose of load balancing and dealing with uncertainty in expected load.

The detailed design of the storage subsystem has been considered in [1, 13]. Policies for increasing the capacity of the disk subsystem by the use of buffering and multicasting are considered in[5, 4, 6].

3 Overview of the Proposed Policy

We first discuss some of the underlying design issues for the BSR policy and will then provide an overview of the policy.

3.1 Issues in Algorithm Design

In a multimedia system, the size of a video object may be very large. Hence, migration and deletion of video objects in general takes a very long time. Careful placement of video objects in advance based on the anticipated load is thus very important. The expected load on a video object can be projected by external agents on the basis of historical data together with human input (e.g. the knowledge that demand for a newly released video will be high). The actual load on an object however, may be affected by factors that are hard to quantify. Examples of such factors are favorable reviews or discounted pricing when viewing of a movie is offered at a discounted rate. Such unpredictability can be better handled by the use of an external agent that monitors the demand for the video objects in the system and initiates on-line changes in placement of video objects when needed. Since the system should adjust quickly to the newly available data on expected load, an important input to the placement policy is the quickness with which changes in placement have to be made.

It is important to note that the *expected load* for a video object is different from the *current load* on that object. The expected load on a replica is maintained by the system only for the purpose of placement. The current load, in fact, interferes with the operation of the placement policy since it prevents creation of replicas on a device that is currently heavily loaded but its expected future load will subside. Since the placement policy needs to maintain the expected load for all video objects in the system (i.e., residing on disks), it is necessary that all placement of video files be carried out under the BSR policy. For example, deletion of a video object is carried out by invoking the BSR policy with a lower expected load on that video object, and the BSR policy may decide to delete some or all the replicas of that video object. The BSR policy also attempts to re-allocate expected load (not actual load) among the remaining replicas so as to match the BSRs of the devices. Similarly a new replica of a video object is created if the expected load on the existing replicas of that video can not be increased. The details are provided later in the section.

Two important decisions have to be made when placing a video object: estimating the number of replicas that are needed, and deciding where to place the replicas. At the time when enough space and expected (not actual) bandwidth is available in the system, deciding the number of replicas to be created may be somewhat arbitrary. Creating a large number of replicas incurs a lot overhead, and may always not be possible if the system is currently busy. However, increasing the number of replicas of a video at this time increases the ability of the system to balance both the expected load as well as the actual load among the devices in the future. This is because for a later placement of another video object the expected load may be shifted amongst replicas of this video without any cost. Due to a large uncertainty in the future demands for video objects, the increased flexibility in reconfiguration provided by multiple replicas is important. Additionally, since clients can view any replica of a video object, multiple replicas increase the dynamic load-balancing ability of the system. The presence of multiple replicas also increases the availability since viewers of a failed replica can be redirected to view another replica.

The expected load on video objects change with time and can be reflected periodically. It is important to note that system operation cannot be interrupted for placement and thus the change in video object placement has to be performed on-line. A change in expected load on more than one video object can happen at the same time. Changes to the expected load for multiple video objects can be accomplished by

repeatedly executing the same algorithm for different videos in a certain order. First, the BSR policy is invoked for changing the expected load for video objects whose demand has decreased. This results in releasing bandwidth and possibly space on the devices in the system. After this, the change in expected load for other video objects, i.e., for those whose expected load has increased are performed in decreasing order of expected load. In Section 5, it is shown that the above ordering can indeed produce better placement. Initial placement of the video objects on the system is a special case of the above and can be done by placing the video objects on the system in decreasing order of expected load.

The actual load on a video will vary with time according to some statistical distribution. Hence, the expected load may be specified in terms of either the peak or average anticipated load. Since multiple videos will share the same device the combined anticipated load will show less variation. Therefore, when the expected load is expressed in terms of peak load on individual videos, the device bandwidths may be *overcommitted*, i.e., the sum of expected load may be greater than the actual device bandwidths. Even in this case the overcommitment across devices needs to be balanced.

3.2 Overview of the Algorithm

The BSR policy takes three inputs: the identifier of the video object to be placed, the new expected load for the video object, and the required bandwidth for creating the replica. The bandwidth required for creation will depend on how quickly the video has to be placed. The algorithm consists of four phases. In the first phase, the policy determines if additional replicas are needed. If more replicas are required, in the second phase the BSR policy selects additional devices on which new replicas are to be created. Selection of additional devices has to be made keeping in mind the available current bandwidth on these devices in order to satisfy the quickness criteria. In the third phase, the expected load is then allocated to all the selected devices so as to match the BSRs of the devices. A parameter to the BSR policy, called the availability parameter, specifies the maximum number of viewers that may be allocated to any replica of any video. This parameter specifies an upper bound on the number of viewers that are lost on a failure of a storage device since it may be possible to move some of the viewers to another replica. A replica consolidation phase is entered only if the policy is unable to place all of the expected load. In the consolidation phase, some of the existing replicas of the other videos are deleted in order to place the new load. The execution of the above phases result in the creation of one or more replicas of the video. The above steps are followed for increase in expected load on a video object, i.e., for creating additional replicas. For a decrease in the expected load, only the third step (i.e., reallocation step) is executed. The unused replicas are deleted only when the space is required.

4 Implementation Details

In the following section, we first describe the data structures necessary for the policy, followed by a detailed description of the BSR policy.

4.1 Data Structures

Figure 2 shows the *video table* data structure that maintains an entry for each video object. The entry for each video object, v_i, contains the bandwidth necessary to play the video object, b_i^v, the size of the video object, s_i^v, and a pointer to a list of *replica entries* for the video object. The j^{th} replica entry consists of the replica identifier, r_{ij}, (which is a unique name, e.g. file name, for the replica), the device identifier of the device on which the replica resides, d_{ij}, and the expected load for this replica, e_{ij}.[2]

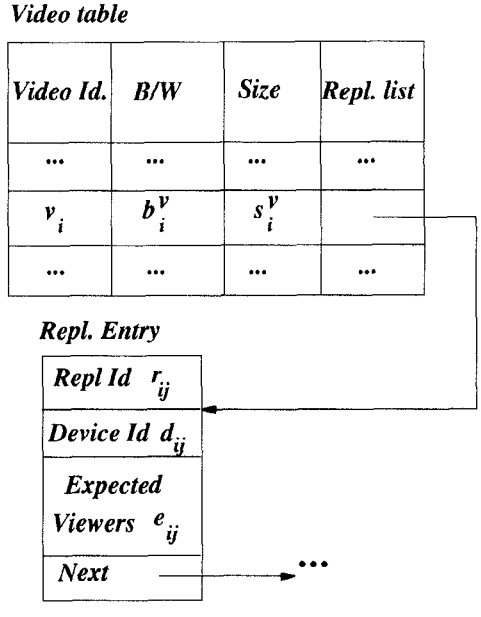

Figure 2: Video data structure

Figure 3 shows the *device table* data structure that maintains an entry for each logical device. As mentioned earlier, each logical device may consist of a number of physical devices striped together. The k^{th} device entry consists of the device identifier, D_k, the maximum

[2]In a system which supports dynamic retrieval of video objects from tape, there may be a replica entry indicating that the video object is on tape.

space, S_k, and free space, s_k^d, the maximum bandwidth, B_k, and the expected free bandwidth, b_k^d, of the device, and a pointer to a list of the *device replica entries* residing on this device. The device replica entry contains the video id of the video object, v_{kl}, and the replica id, r_{kl}. The expected free bandwidth of the device, b_k^d, is given by $b_k^d = B_k - L_k$, where L_k is the total allocated expected load on the replicas placed on this device. The expected load on each replica on this device, e_{kl}, can be found from the video table using the video id, v_{kl}, and replica id, r_{kl}. L_k can thus be computed as the sum of expected load over all the replicas on this device. Initially, when the devices are added to the system, they contain no video objects. Therefore, the free space and bandwidth are set to the maximum space and bandwidth, respectively, of the device. These values are updated whenever video objects are loaded onto the device.

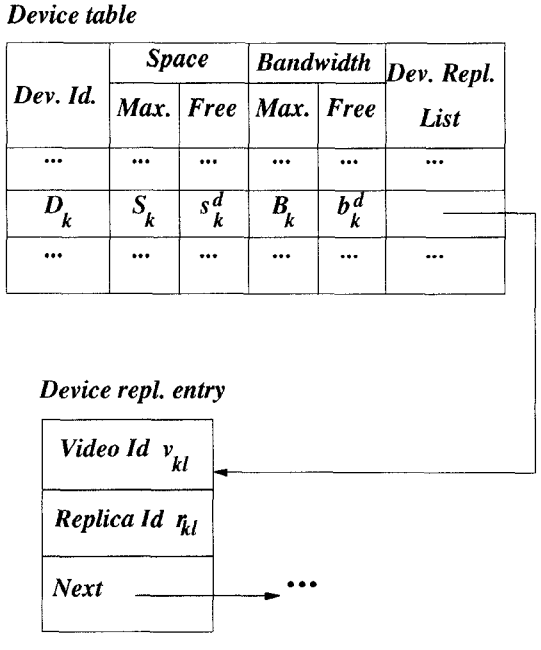

Figure 3: Device data structure

As described in Section 3, the BSR policy is a four-phase policy. These four phases are described in detail below.

4.2 Details of Implementation

The first phase of the BSR policy determines if additional replicas of the video object being placed are necessary, for an increase in expected load. The policy first determines the maximum additional load that can be placed on the existing replicas of this video. This is estimated by taking into account the free bandwidth of the devices on which the replicas reside, and the limit imposed by the availability parameter. The limit may also be imposed by the BSR deviation threshold. New replicas are needed if the total additional expected load that can be placed on the existing replicas is smaller than the increase.

The second phase is executed if additional replicas are needed. The BSR policy selects the devices on which to place new replicas as follows. The BSR of the video objects on a device is defined as the ratio of the total allocated bandwidth to the total allocated space of that device. As illustrated in Figure 1, this is different from the BSR of the device. The *BSR deviation* of a device is defined as the deviation of the BSR of the video objects on the device from the BSR of that device. The devices without a replica of this video are then considered in decreasing order of BSR deviation. A device is classified as hot if the BSR of the video objects on that device is greater than the BSR of the device and as cold otherwise. Depending on the additional expected load to be placed and the size of the video, hot or cold devices are selected to reduce the BSR deviations. A hot device is selected if placing the video object on that device will make the device less hot and hence, will reduce the its BSR deviation. Otherwise the video object is placed on a cold device. Of course, a device is selected only if enough free space is available on that device to store the video, and enough current bandwidth to create the replica satisfying the quickness criteria. The maximum additional expected load that can be placed on the new replica is computed as in phase one. Selection of devices stops when either i) sufficient devices have been chosen to satisfy the additional expected load or ii) no more devices can be selected. In the latter event, an emergency scan is initiated to select additional devices where the availability limit and the classification of the devices into hot and cold are ignored.

In the third phase, the BSR policy attempts to allocate expected load so as to match the BSR of the devices. Note that the first two phases are skipped for a decrease in expected load since the existing replicas are sufficient to satisfy the demand. The total expected load on the video is allocated over all replicas (including newly created replicas) so as to minimize the BSR deviation across devices. The BSR policy then executes a rebalancing operation as follows. It first orders all devices in decreasing order of BSR deviation. For the device with the highest BSR deviation, the policy shifts expected load to/from devices with lower BSR deviation until no more load can be shifted.

The fourth phase of the BSR policy is executed only if the policy is unable to place any additional expected load in the first three phases. This implies that all the devices have exhausted either their bandwidth capacity or their space capacity. However, devices that have

exhausted their bandwidth may have space available and vice versa. The policy now consolidates multiple replicas into one through shift in expected load in order to free up space on a device with available expected bandwidth. For each such replica, the policy determines if the replica can be deleted by shifting all the expected load to other replicas. If all the expected load cannot be shifted, it may still delete this replica for creating a new replica of the new video if this results in increased expected bandwidth utilization of the system.

5 Results

In this Section, the performance of the on-line placement policy is studied using simulation. There are various ways to measure the performance of the policy. For example, after every invocation of the policy the deviation of the BSRs could be measured. However, the effectiveness of the policy is really demonstrated under the stress cases. A good measure of placement is the expected amount of unutilized bandwidth that can not be used when new video replicas cannot be placed due to the lack of available space. This bandwidth is referred to as the *unutilized bandwidth* in this paper. Recall from Section 3 that the expected load is not the actual load, and a dynamic load balancing policy attempts to balance the actual load across all devices. Hence, all of the bandwidth may actually be used if the dynamic load balancing policy can balance this load. However, unutilized bandwidth is a reflection of the difficulty in achieving this. Another related measure is the number of replicas deleted due to bad earlier placements.

Unless otherwise specified, two configurations of striping sets are used. The first one is referred to as the *base configuration* consisting of four striping sets of 8, 16, 32 and 64 equal size disks, respectively. Each individual disk is assumed to provide 2 GB of space and support 6.25 unit of load. Hence, the bandwidth and space of the four striping sets are (50, 16GB), (100, 32GB), (200, 64GB) and (400, 128GB), respectively. Note that even though the bandwidth and space of different striping sets are different, the BSR is the same for all sets. The second configuration consists of four equal size striping sets with (188, 60GB) amount bandwidth and space, respectively, and is referred to as the *symmetric configuration.* Note that the two configurations have the same amount of total space and bandwidth. Each video is assumed to be 3.6GB in size. Therefore, approximately 64 replicas can be placed in the entire system. To create stress case, it is further assumed that the total expected load on all videos are the same as the total system bandwidth and the number of videos to be placed as 64. Therefore, there is just enough space to store a single replica of all the videos. Note that if the load on a particular video is very high multiple replicas may need to be created, and hence, the system will not be able place all videos. In other words, there will be some unutilized bandwidth as defined above. Only in the ideal situation, all videos can be placed and the unutilized bandwidth will be zero. The expected load on the videos are given by the Zipf distribution with the parameter 0.271 [5] (also see Figure 4). The quickness criteria is always satisfied as the simulated system has no actual load. The availability parameter is assumed to be 30, i.e., no more than 30 unit of load per replica is affected by a failure of the replica.

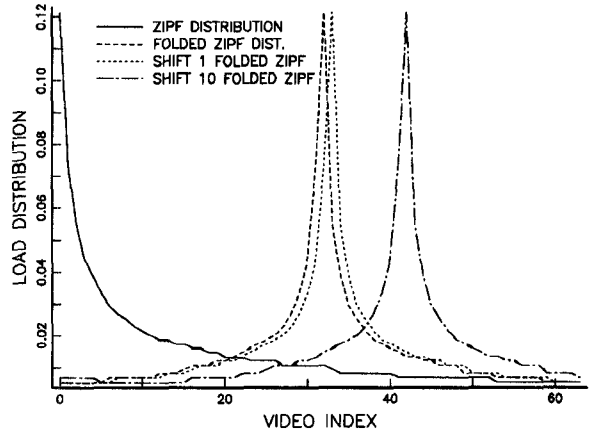

Figure 4: Illustration of Zipf, Folded Zipf and K-Shifted Folded Zipf distribution

5.1 Initial Placement Sequence

The performance of the on-line policy is first studied assuming no prior placement of videos. Figures 5 and 6 show the capacities and allocated bandwidth and space of the four sets after attempting to place all 64 videos. Two different allocation schemes are used corresponding to the availability parameters 30 and 50, respectively. As can be seen from the figures that excellent matching of BSRs are achieved. This is further reflected in the small amount (4.8%, 4.8%, respectively) of expected unutilized bandwidth. As mentioned earlier, under this stress case, some bandwidth will be expected to be unutilized as multiple replicas for the popular videos leaves no space for unpopular videos. Even all the space can not be fully utilized since the sizes of the video replicas do not always add up to exactly the sizes of the devices. The algorithm also deletes a small number of replicas (14,6, respectively) during the entire placement sequence. Most real systems will be configured either with more available space than the space required for placing a single replica of all videos, or the device bandwidths will be overcommitted. Hence, very little bandwidth will be unutilized. Figures 7 shows the bandwidth allocation after initial placement sequence for the symmetric configuration for two different availability parameters

(30 and 50). There is also an excellent match for this case.

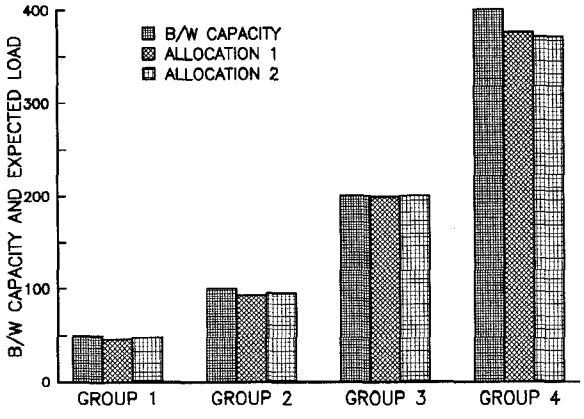

Figure 5: Expected load across striping groups (base configuration)

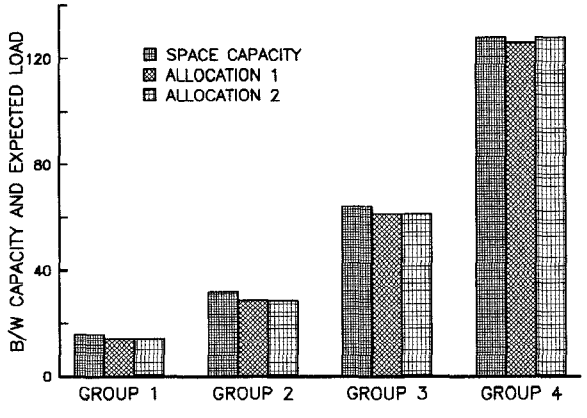

Figure 6: Allocated space across striping groups (base configuration)

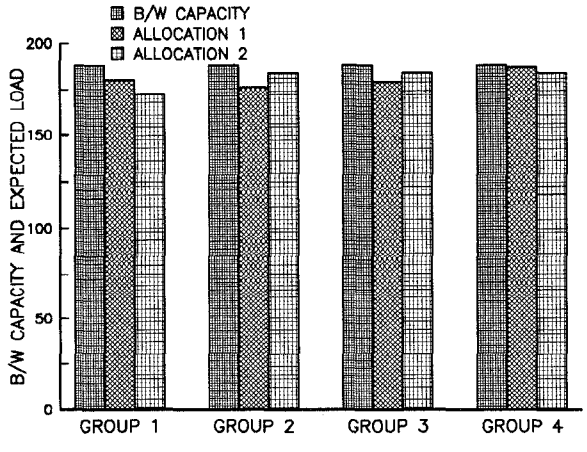

Figure 7: Expected load across striping groups (symmetric configuration)

Different placement sequences of the videos with different expected loads are next studied. Figure 8 shows the unutilized bandwidth for three different placement sequences for the two configurations. The curves are very similar for both the configurations. Under a decreasing load sequence, the video with the highest expected load is placed first. The reverse is the case for the increasing load sequence. For the case of random placement sequence, the average over five different placement sequences is estimated. The experiments are repeated for different load skew parameters (i.e., Zipf parameter) that determine load skews across different videos. As expected, even under such a stress case, the placement under an decreasing load sequence wastes only a small amount of bandwidth. The increasing sequence on the other hand, places the videos with the lowest load first, and runs out of space for placing videos with higher loads. Unless a policy decision is made that a replica with a higher load to be placed later can replace another video with a lower expected load, the later placement attempt is aborted. Deletion of a replica may not always be possible if there are ongoing streams on that replica. In the experiments, no such replacement is assumed. For a lower load skew across videos, the placement sequence has very little effect on the unutilized bandwidth. The unutilized bandwidth for the random case is between the two extremes. Figure 9 shows the corresponding number of deleted replicas. In the algorithm, the availability constraint is violated whenever the replica consolidation is considered. Hence, for the decreasing placement sequence, some additional replicas of the hot videos are created when the space is available, but subsequently are deleted when the system is under stress. Increasing sequence on the other hand does not face this situation.

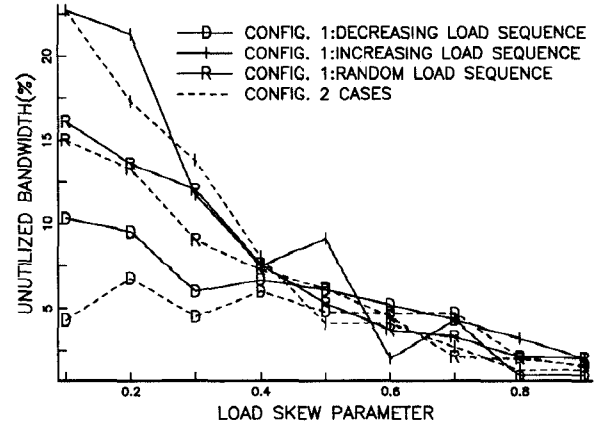

Figure 8: Effect of placement sequence on unutilized bandwidth

Since the available space is a key factor in determining the expected unutilized bandwidth, the number of videos to be placed in the system is varied next (see Figure 10). The total load across all videos are

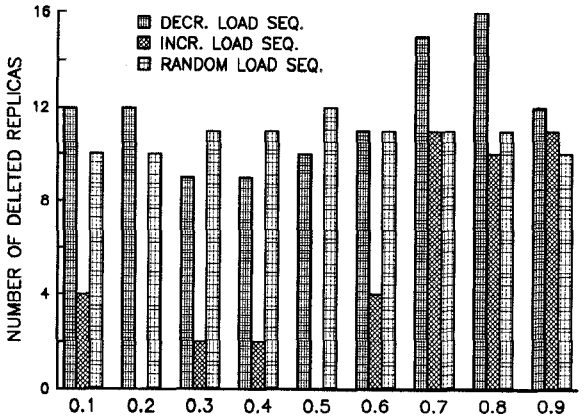

Figure 9: Effect of placement sequence on the number of deleted replicas

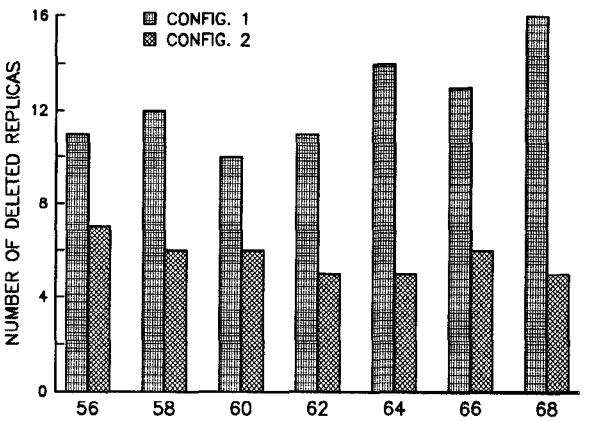

Figure 11: Effect of available space on the number of deleted replicas

made equal to the total system bandwidth capacity in all cases. As expected, with a smaller number of videos the unutilized bandwidth becomes smaller. Note however that with a smaller number of videos a larger load is expected on the hot videos, and hence, a larger number of such videos require multiple replicas. Figure 11 shows the corresponding number of deleted replicas. With a larger number of videos space becomes an extremely valuable commodity, and replica consolidation is enforced. Hence, the number of deleted replicas increases with a larger number of videos. For the base case with unequal striping set sizes, a better matching of BSRs is achieved by deleting a larger number of replicas.

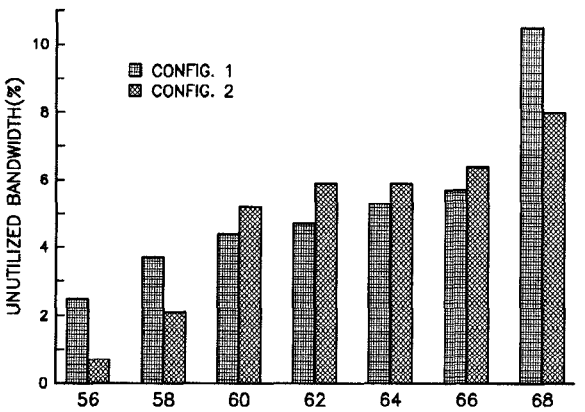

Figure 10: Effect of available space on unutilized bandwidth

5.2 Performance under Various Configurations

To demonstrate the robustness of the policy, the performance of the video placement policy in various other configurations is studied. In all experiments, the total number of disks and hence, the total space and bandwidth of the system are kept constant. In the first set of experiments, 2, 4, 6 and 10 equal size striping sets are considered. With a larger number of striping groups a larger amount of bandwidth and space are unutilized due to fragmentation (see Figure 12). Overall, for the 10 group case, the unutilized bandwidth is much larger. It is important to note that the dynamic segment replication policy [7] can make use of the unused space to better balance the actual load across the striping groups. The number of deleted replicas also increases with a larger number of striping groups (see Figure 13) indicating a larger number of initial replica creation and a larger instances of shifting of expected load across replicas.

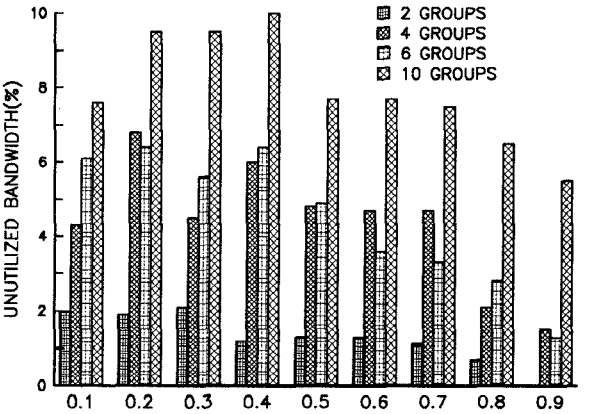

Figure 12: Effect of number of groups on unutilized bandwidth

In the second set of experiments, the number of striping groups is kept fixed at four, but the BSR of various groups are made different by varying the bandwidth and/or space across various groups. In real systems, this represents striping groups made of different types of disks. The four cases are considered. The first case is the same as the symmetric configuration. In the second case, the space of all four groups are kept the same, but

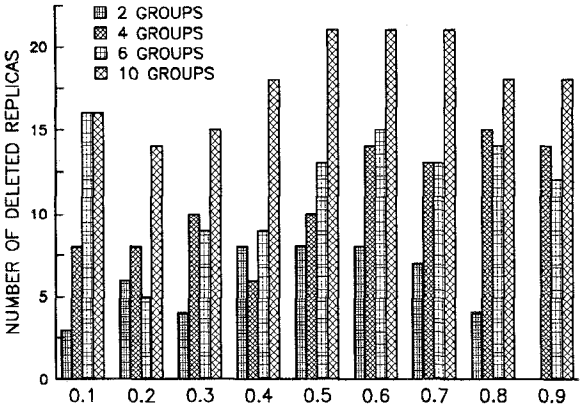

Figure 13: Effect of number of groups on the number of deleted replicas

the bandwidth of different groups are changed. The first and second groups have respectively 20% and 10% more bandwidth than the average, while the third and fourth groups have respectively 10% and 20% less bandwidth. The third case is similar to the second case except that the bandwidth increase or decrease is doubled compared to the second case. In the fourth case, groups with very different BSRs are created by applying 10% space difference on top of 20% bandwidth difference. The space reduction is highest for the group that has the highest bandwidth. Figure 14 shows the unutilized bandwidth for all these configurations as a function of load skew parameter. For the first two cases, the unutilized bandwidth is smaller than 7%. With a lower load skew (i.e., a higher load skew parameter) the third and fourth cases where the BSR of the devices are very different, it is inherently hard to find a good match of the BSRs.

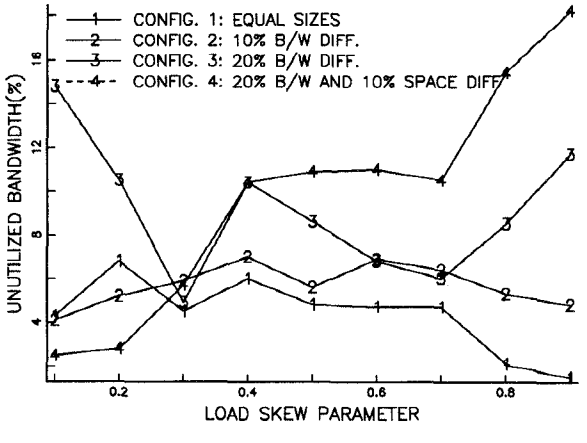

Figure 14: Effect of BSR difference on unutilized bandwidth

5.3 Change in Expected Load

The on-line policy performs equally well even when the system already has a lot of existing videos. This is probably the normal mode of operation for most systems since the system is assumed to be operational at all times with a lot of ongoing access streams. The load change across all videos is gradual rather than abrupt, i.e., the previously hotter videos become slightly colder while the previously colder videos become somewhat hotter. The change in expected load is modeled as K-shifted Folded Zipf distribution (illustrated in Figure 4). First the load across all videos are generated using an Zipf distribution. Assume the videos are indexed such that the video with the highest load is in the middle. The next highest load is in its immediate left, the next highest is in its immediate right, and so on. This creates a load curve where the load on each side of the video with the highest load is approximately the same. Next, the indices are shifted (rotated) by the amount of K. For a shift of 1, the video on the immediate right of the previously hottest video becomes the now hottest video. A larger shift of indices represent a quick change in load.

Figure 15 shows the capacity and allocated bandwidths after an initial placement sequence, and after a subsequent shift of 1 and 10, respectively. In all cases, the BSR match is excellent. Note that, only a small number of replicas of the previously hot videos (less than 9) are deleted and the space is used to create the new replicas for the now hottest videos.

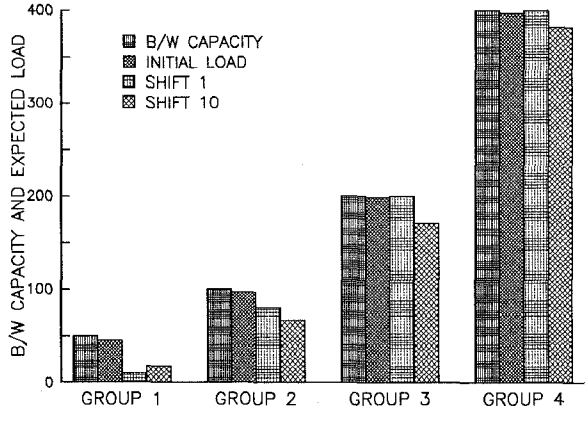

Figure 15: Expected load across striping groups after a change in expected load

6 Summary and Conclusions

In a multimedia server, each storage device is characterized both by its bandwidth (i.e., the number of streams that can be supported) and space (i.e., the number of videos that can be stored). Both long video objects such as movies and short video objects such as video clips may be present in the system. The viewing frequency, in general, will be skewed and independent of the size of the objects. Because of the limited bandwidth of each storage device, it may be necessary to create multiple replicas of popular video objects on dif-

ferent devices. Without careful placement of video objects on the storage devices, load imbalance and wastage of device bandwidth and space may occur. For example, placing many infrequently viewed large videos on a device with a large bandwidth but small space wastes the bandwidth of that device. It is therefore necessary to mix objects of different sizes (large and small) and different viewing frequencies (popular and infrequent) on each device. For avoiding wastage of device capacity, these mixes should match the bandwidth and space capacities of these devices.

In this paper, a policy called the *bandwidth to space ratio* or *BSR* policy is proposed for achieving this objective. The proposed policy characterizes both storage devices and video objects by their bandwidth to space ratio. The BSR of a video object is the ratio of its expected bandwidth (which is computed from the *expected load* on the object) to its size. The BSR policy computes the number of replicas needed as well as the devices on which to place the replicas. Under changing demands, the policy also dynamically changes the number and placement of replicas to better match the BSRs of the storage devices and video objects. Hence it is an on-line policy that can be used while the server is fully operational.

The performance of the BSR policy is studied using simulation. The experiments are selected to study performance under stress where the total storage requirement for a single replica of all video objects is equal to the total storage capacity of the devices and the total expected load is equal to the total bandwidth capacity. The effectiveness of the policy was studied by measuring the expected amount of bandwidth that could not be utilized and the number of replicas created initially and deleted later due to bad initial placement. Since the expected load is different from the actual load, the expected unutilized bandwidth may not actually be unused. Only a small fraction of the bandwidth is expected to be unutilized in all experiments and the BSR policy deleted only a small number of replicas that were created earlier. Decreasing the availability parameter (that limits the expected number of viewers on a single replica) increases the availability but forces additional replication, and thereby, increasing the amount of unutilized expected bandwidth. With fewer video objects to be placed, the BSR policy performs better as there is extra space for greater replication. The performance of the BSR policy after a change in expected load of the already placed videos is found to be equally good.

Acknowledgements: We would like to thank Tim Krein and Scott Marcotte for their feedback in the design of this algorithm, Swarna Dinkar for her encouragement in preparing the paper and anonymous referees for their help in improving the presentation.

References

[1] Anderson, D. P., Y. Osawa, and R. Govindan, "A File System for Continuous Media," *ACM Transactions on Computer Systems*, Vol. 10, No. 4, November 1992, pp. 311–337.

[2] Berson, S., L. Golubchik, and R. R. Muntz, "A Fault-Tolerant Design of a Multimedia Server", *UCLA Tech. Report CSA-940009*, Feb. 1994.

[3] Berson, S., S. Ghandeharizadeh, R. Muntz, and X. Ju, " Staggered Striping in Multimedia Information Systems", *ACM SIGMOD '94*.

[4] Dan, A., and D. Sitaram, "Buffer Management Policy for an On-Demand Video Server", *IBM Research Report, RC 19347*, Yorktown Heights, NY, 1994.

[5] Dan, A., D. Sitaram, and P. Shahabuddin, "Scheduling Policies for an On-Demand Video Server with Batching", *ACM Multimedia'94*, San Francisco, CA, October, 1994.

[6] Dan, A., P. Shahabuddin, D. Sitaram and D. Towsley, "Channel Allocation under Batching and VCR Control in Movie-On-Demand Servers", *IBM Research Report RC 19588*, Yorktown Heights, NY, 1994.

[7] Dan, A., M. Kienzle and D. Sitaram, "Dynamic Segment Replication Policy for Load-Balancing in Video-on-Demand Servers", *IBM Research Report, RC 19589*, Yorktown Heights, NY, 1994 (also to appear in *Multimedia Systems*).

[8] *Electronic Engineering Times*, March 15, 1993, pp 72.

[9] Fox, E. A., "The Coming Revolution in Interactive Digital Video," *Communication of the ACM*, Vol. 7, July 1989, pp. 794–801.

[10] Ghandeharizadeh, S., and C. Shahabi, "On Multimedia Repositories, Personal Computers, and Hierarchical Storage Systems", *ACM Multimedia'94*, San Francisco, CA, pp.407–416.

[11] Kienzle, M., A. Dan, D. Sitaram, and W. Tetzlaff, "Using Tertiary Storage in Video-On-Demand Servers", *Compcon'95*.

[12] Liskov, B., et. al., "Replication in the Harp File System", *ACM Symp. on Operating Systems Principles*, 1991, pp. 226-238.

[13] Rangan, P. V., H. M. Vin, and S. Ramanathan, "Designing an On-Demand Multimedia Service," *IEEE Communication Magazine*, Vol. 30, July 1992, pp. 56–65.

[14] Sincoskie, W. D., "System Architecture for a Large Scale Video On Demand Service," *Computer Networks and ISDN System*, Vol. 22, 1991, pp. 155–162.

[15] Sitaram, D., A. Dan, and P. Yu, "Issues in the Design of Multi-Server File Systems to Cope with Load Skew", *Second International Conference on Parallel and Distributed Information Systems*, January 1993.

[16] *Video Store Magazine*, Dec. 13, 1992.