Using Heterogeneous Disks on a Multimedia Storage System with Random Data Allocation *

Jose Renato Santos†, Richard Muntz
UCLA Computer Science Department
{santos,muntz}@cs.ucla.edu

Abstract

RIO (Randomize I/O) is a multimedia object server which manages a set of parallel disks and supports real-time data delivery with statistical delay guarantees. While most multimedia storage systems proposed on the literature are designed to support just one type of application, namely video on demand, RIO can simultaneously support multiple applications like video on demand and 3D interactive virtual worlds as well as non real-time applications. In RIO data objects such as videos or 3D virtual worlds models are divided in constant size data blocks and then blocks are randomly placed on the multiple system disks. The randomness of data placement ensures long range load balancing, since each disk is equally likely to be the target of each disk access. To provide short term load balancing RIO uses limited redundancy by replicating some fraction of the data blocks. This data redundancy gives the disk scheduler some flexibility in scheduling disk block reads, allowing the system to reduce the short term load imbalance.

In this paper we analyze the performance of RIO when the set of disks used to store data blocks is not homogeneous, having both different bandwidths and different storage capacities. The basic problem to be addressed for heterogeneous configurations is that, on average, the fraction of the load directed to each disk is proportional to the amount of data stored on it, which may not be proportional to the disk bandwidth. This may cause some disks to be overloaded, with long queues and delays, even though bandwidth is available on other disks. This reduces the system throughput or increases the delay bound that can be guaranteed. This problem arises whenever the bandwidth to space ratio (BSR) is not uniform across all disks. In RIO this problem is addressed by replicating a fraction of data blocks. Replication allows some of the load of the disks with smaller BSR to be redirected to the disks with higher BSR, reducing or eliminating the long range load imbalance, depending on the amount of replication. While, on a homogeneous system, replication is used just for short term load balancing, on a heterogeneous system, replication can be effectively used for both long and short term load balancing.

*This research was supported in part by grants from Sun Microsytems, Intel Corp., and NSF grant IRI-9527178.

†Also with University of Sao Paulo; his research is partially supported by a fellowship from CNPq (Brazil)

1 Introduction

A prototype real-time storage system that can store and deliver both real-time and non real-time multimedia data to a large number of simultaneous users has been designed and implemented. The system, named RIO (Randomized I/O), supports high data rates and many simultaneous data streams by using the aggregate bandwidth of a collection of parallel disks [12]. RIO is part of the VWDS (Virtual World Data Server) Multimedia System that can simultaneously support many types of multimedia applications such as video on demand, 3D interactive virtual world navigation, interactive scientific visualization, etc., as well as traditional non real-time applications. Our current prototype is implemented on a SUN E4000 SMP, having 10 Ultrasparc processors, 1.25 GB of shared memory, fourteen 4 GB disks dedicated to multimedia storage, with ATM and Ethernet connections to client machines. We are currently porting RIO to a distributed architecture implemented on a cluster of Pentium based workstations [8].

Both the centralized and distributed versions of RIO, have previously assumed a homogeneous set of disks, where the storage space and bandwidth provided by each disk is the same. It's reasonable to assume that a new system will be configured with a set of homogeneous disks. However, if the system needs to be upgraded after a period of time, to support more users, higher bandwidth applications or to store more or larger multimedia data objects, new disks will have to be added to the system. Since disk technology is changing very rapidly, most probably the new disks will have different characteristics than the original ones, with respect both to the storage capacity and data rates supported. Typically, the new disks will be faster and with larger capacity. Heterogeneous disks introduce asymmetries in the system which may cause load imbalances across the disks and thus may reduce the effective data rate that can be supported in order to continue satisfying the real-time requirements of applications. The purpose of this paper is to analyze the real-time performance of a parallel storage system composed of heterogeneous disks.

The rest of this paper is organized as follows. In section 2, we briefly describe the RIO storage system, summarize the performance results for a system with homogeneous disks, and describe related works. Section 3 analyses the effect of replication on long range load balancing for heterogeneous disk configurations. In section 4 we analyze, through simulations, the real-time performance of RIO for heterogeneous disk configurations. Finally in section 5 we present our conclusion.

2 The RIO Storage System

2.1 System Overview

In this section we describe the principal characteristics of RIO. For a more detailed description of RIO the reader is referred to [12]. Although RIO supports both read and write operations, we assume that most of the time users are reading data, while writing is done infrequently and when it is done it is usually a non real-time operation (e.g., loading a new object). We thus consider a scenario of read only operations in the rest of this paper.

In RIO, each multimedia object is composed of a sequence of constant size data blocks. Each data block is stored on a randomly selected disk at a random physical disk block. When a set of homogeneous disks is used, the randomness of data placement ensures long range load balancing, since each disk is equally likely to be the target of each disk access, and thus, in the long run, each disk will have the same average share of the load. However, it is possible that during a short period of time data requests are concentrated on a few disks, causing a short term load imbalance, with larger request queues on a few disks than on others. This increases the probability of violating the real-time constraints when compared to a system with perfectly balanced disk queues. To provide short term load balancing, RIO replicates some fraction of the data blocks on a different disk, also randomly selected. This data redundancy gives the disk scheduler some flexibility in scheduling disk block reads, allowing the system to reduce the short term load imbalance, and thus reduces the probability of violating the real-time constraints of request response times. Note that any sequence of application requests will generate a sequence of requests to data stored at random disks or random disk pairs, independently if the requests are generated by one or multiple streams, accessing the same or different data objects.

However, when the system is composed of heterogeneous disks, the random placement of data blocks on disks does not ensure long range load balance by itself. Since we assume random allocation of data blocks on disks, the fraction of the load directed to any disk is, on average, proportional to the amount of data stored on it, and thus also proportional to the disk storage capacity. However, this load may not be proportional to the disk bandwidth. In general, some of the disks may operate at a higher utilization than others, having longer queues and delays. This problem arises whenever the ratio of disk bandwidth and storage capacity of each disk, called BSR (Bandwidth to Space Ratio) [5], is not the same in all disks. In this paper we investigate the effectiveness of addressing this problem using data replication. Replication allows some of the load to the disks with lower BSR to be redirected to the disks with higher BSR, reducing or even eliminating the long range load imbalance, depending on the amount of replication. Note that, on a homogeneous system, replication is used for short term load balancing, while on a heterogeneous system, replication is used for both long and short term load balancing.

Each disk has an associated queue that holds requests that need to access data on that disk. Requests are ordered on all queues according to their deadlines, where non real-time requests are assumed to have infinite deadlines. New requests are routed to the appropriate disk queue at the time they arrive, using an online load balancing algorithm. Requests for non replicated blocks are routed to the only possible disk while requests for replicated blocks are routed to the least loaded of the two possible disks (the one with the least expected delay). Requests on disk queues are serviced in asynchronous cycles. Whenever, the disk finishes reading requests from a previous cycle and new requests are waiting on the disk queue, a new cycle starts. At the beginning of each cycle the first requests on the disk queue, up to a maximum predefined number of requests, are selected to be serviced in the current cycle. The selected requests are then reordered according to their location on the disks, and serviced according to a conventional SCAN algorithm that minimizes seek-time overhead.

2.2 The RIO Simulator

In order to evaluate the performance of RIO under different configurations, we implemented a simulator. The simulator basically route requests, generated by a traffic generator, to disk queues according to the online load balance algorithm; service requests according to the cycle based scan algorithm, using a model of the disk behavior; and measure the delay of each individual request, when it is completed. The simulator is driven by a traffic generator that generates a sequence of requests to randomly selected disks. Each request is randomly selected to be replicated with probability of replication set according to the desired level of replication (between 0% and 100%). We assume that the total input traffic is the sum of the traffic generated by many independent streams and thus can be modeled as a Poisson process. This is a reasonable assumption, since RIO is designed to support many simultaneous users. Therefore, we use a Poisson arrival process to generate the input traffic to the simulator.

To model the disk performance we did a set of experiments on the disks used on our prototype, Seagate Barracuda ST151150W. The disk performance for random data access is dependent of the size of data block selected, with larger data blocks providing better performance, since the overhead of seek, rotational latency, etc., is amortized over longer data transfer times. However larger data blocks require larger memory buffers, have longer I/O times and increase the probability of reading superfluous data for applications with unpredictable access patterns. The selection of the right block size needs to consider these tradeoffs but is beyond the scope of this paper. In our simulations we assumed a block size of 128 KB, but since we present our results normalized by the mean I/O time, the performance of other block sizes can be easily obtained by scaling the results by the corresponding mean I/O time.

Although we could use a detailed model of the disk performance, considering seek time, rotational latency, disk transfer rate, SCSI bus transfer rate, OS overhead, etc., we observed in our experiments that a simple normal distribution model is a good approximation for the disk I/O time, and thus we used this model in our simulator. We selected a block size of 128 KB and a cycle size of 1, for all simulations in this paper. In this case the mean disk I/O time for our disks is $\mu = 35.25$ms and the standard deviation is $\sigma = 5.20ms$. In the simulations presented in this paper we used these values for simulating the slower disks and a mean and standard deviation for the faster disks decreased proportionally to the assumed increase in bandwidth.

In each experiment we simulate the system for a period of time sufficiently large to generate approximately 10^7 requests. We then measure the delay of each request and estimate the delay distribution by computing a delay histogram. The delay bound that is "guaranteed" by RIO is defined such that the probability of a request being delayed by a value greater than the delay bound is less than a very small probability, which is set to 10^{-6} in the performance results presented in this paper.

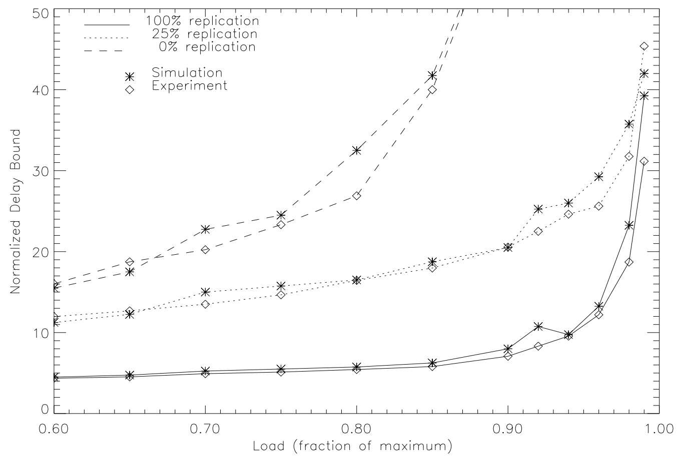

Figure 1: Simulation performance results compared to experimental results

In order to validate our simulation we compare the simulation results with experiments done in our prototype. Figure 1 shows the delay bound that can be "guaranteed" with probability $(1 - 10^{-6})$ as a function of the relative load (i.e.. the disk utilization), obtained both for experiments on the prototype and for simulations. These results assume 128 KB blocks and cycles of size 1. Clearly the results of the simulation track very closely the measured results, confirming that our simulation is modeling the system performance and disk behavior accurately. The results also show that higher levels of replication allow lower delay bounds, due to better load balancing.

2.3 Related Work

Most of the work on multimedia storage servers has concentrated on video and audio playback, which has highly predictable access patterns. Once a video playback stream is started, data is sequentially read from the storage system at the playout rate. This predictability is exploited in many video servers that carefully layout data on disk to achieve good load balance and high real-time performance. The most widely used scheme is data stripping [1][4][15].

We on other hand, have proposed a random allocation of data to disks as a mean to support more general workloads [2] [12]. Nevertheless, we have shown in [12] that even for predictable workloads like video playback, despite the randomness of data allocation, our system has performance equivalent to, and in some cases even better than, conventional stripping schemes if replication is combined with random allocation. More recently, other authors have also adopted random data allocations for supporting video playback [14][3][10]. Tewari et all [14] analyze the performance of a clustered video server with random allocation of data blocks. However, they do not consider data replication and do not address the problem of short term load imbalances. Birk [3] describe a high performance video server that uses random distribution of blocks on RAID systems. The major difference of his work and ours, is that he explores redundancy of parity groups for load balancing, as opposed to replication. Korst [10] compares different static algorithms for optimally distributing requests to randomly replicated data blocks. However the author only considers static load balance algorithms which assume that all requests to be balanced on a given cycle are known. This assumes synchronized cycles on all disks which reduces performance since disk I/O time is variable and in each cycle some disks will finish their load before others. Therefore the cycle duration has to be specified considering worst case I/O times causing most of the disks to be idle towards the end of the cycle. This is avoided in our online load balancing scheme, since disk cycles are asynchronous. We believe that random data allocation will become very popular in the next years, since random allocation solve many practical problems associated with careful layout and scheduling on video servers. For example, variable bit rate on compressed video, variable I/O times on disks, multiple playout rates of different objects, use of heterogeneous disks, etc., all complicate regular layout and scheduling.

The routing of requests for replicated blocks to the least loaded disks is easily mapped to a problem that has been studied in the computer science literature for load balancing in distributed systems [7], that is often referred to as "random probing". Rather than disk accesses and disks, the problem is normally expressed in terms of tasks and processing nodes. The mapping of this problem to our parallel disk subsystem is clear. Disk requests are the tasks and random disk block allocation with random replication is equivalent to randomly probing for the least loaded server. The most recent work related to this problem is that of Mitzenmacher [11] which has a detailed analysis of several variations of a multi-server queuing model in which customers can randomly probe d servers and join the one with the shortest queue. The results show that an exponential improvement in the mean waiting time of customers is obtained in the system when multiple choices are provided as compared to the case where customers have only one choice. In particular, most of the improvement is obtained when customers only have two choices. Our experimental results confirm that replicating data on multiple disks can significantly reduce the variation of individual disks loads, providing a more balanced and efficient system. However there are significant differences between our system and the models analyzed in [11] with respect to service time distributions, arrival processes, etc. Since we require accuracy on the tail of the response time distribution we resorted to simulation and measurement. Models, however, are useful for configuration planning and a first order approximation of the delay bounds that can be guaranteed.

Heterogeneous disks on video servers is considered in [5]. They explore popularity of data objects for dealing with different BSR of disks. Object popularity is in also considered in homogeneous configurations, when object copies are completely stored in just one or in a subset of the disks. In this case, disks storing very popular or "hot" objects may be overloaded and additional copies of these objects are made on other disks, to improve load balance [16]. In [5] the authors address the heterogeneity problem by placing a higher percentage of "hot" data on disks with higher BSR than on disks with lower BSR, such that the predicted load on disks are made proportional to their bandwidths. The drawback of this approach is that predicting object popularity is not always easy and in general imperfect. Moreover, the actual load suffers fluctuation over time and predicted load can at best estimate the average load, but not the actual load. Dynamically replicating objects based on current demand has been proposed as a way to reduce this problem [6], but this requires additional disk bandwidth for copying data online. Our approach avoids these problems, since it does not rely on data popularity. Moreover, as we will show, distributing load proportionally to disk bandwidth is not necessarily the best thing to do, but biasing the load more towards faster disks should produce better performance.

3 Long Range Load Balance

As discussed previously, using heterogeneous disks in RIO without replication will limit the maximum data rate that can be supported if storage capacity is fully utilized. A simple alternative to solve this problem would be to use only part of the storage capacity on some disks in order to balance the load. Another alternative, which we adopt in RIO, is to replicate a fraction of data blocks and give the system some flexibility to shift some of the load from disks with lower BSR to ones with higher BSR and have an effective long range load balance, and thus effectively use the total bandwidth supported by the collection of disks. Of course, to make statistical guarantees on the maximum delay we have to limit the system load to a value lower than the maximum supported data rate, such that the delays are kept below a given delay bound with high probability. A system operating close to the saturation point would have very long queues and delays. However in this section we are only interested in evaluating the maximum throughput that could be supported by the system if there were no constraints on the delays. In section 4 we analyze the performance of RIO when real-time constraints have to be satisfied. Our purpose in this section is just to analyze how replication can reduce or eliminate long range imbalances and how much replication is required to achieve the maximum throughput offered by the collection of disks.

3.1 Determining the minimum replication necessary to achieve a given bandwidth utilization

Consider a system with M sets of disks, where each set is homogeneous, i.e., contains one type of disk. Let set i ($1 \leq i \leq M$) have N_i disks which can store a total of S_i blocks and have an aggregate bandwidth of B_i blocks/sec. Note that our definition of disk bandwidth, used in the rest of this paper, is the effective disk bandwidth in terms of the number of data blocks per unit of time that can be read from the disk, and therefore includes I/O overheads such as seek time and rotational latency. Of course, different block sizes and disk scheduling algorithms (FIFO, Scan, etc) will provide different effective bandwidths B_i, but we assume that the block size and disk scheduling algorithm are fixed and determine the effective bandwidth B_i for each disk set.

Let $S = \sum_{i=1}^{M} S_i$ be the total space and $B = \sum_{i=1}^{M} B_i$ be total bandwidth supported by the system. We then define the relative space and bandwidth of each disk set as $s_i = \frac{S_i}{S}$ and $b_i = \frac{B_i}{B}$. We also define the relative bandwidth space ratio of disk set i as $bs_i = b_i/s_i$. A system with uniform BSR has $bs_i = 1$, for all disks, and can fully utilize the total available disk bandwidth B. However, if the disk BSR is not uniform some disks will be relatively fast ($bs_i > 1$) and some relatively slow ($bs_i < 1$). If no replication is used, the disks with the lowest bs_i will be the most stressed disks and will be the first to saturate. Note that a disk with the lowest bs_i is not necessarily the slowest disk in system, since it may have higher bandwidth and still saturate before others because it also stores larger amounts of data and has to support a higher share of the load than other slower disks. In fact, most likely, the fastest disks on the system will be the relatively slowest ones, since disk technology trends shows that disk capacity is increasing faster than disk bandwidth [9]. Without loss of generality, we will assume through the rest of the paper that disk sets are ordered according to their relative bandwidth space ratio, i.e $bs_i \leq bs_j$ for $i < j$.

Let r be fraction of data that is replicated and $D = S/(1+r)$ be the total number of data blocks (without considering replicas). Let $\lambda_{max} \leq B$ be the desired maximum load to be supported corresponding to an average disk utilization $\rho = \lambda_{max}/B$. A necessary condition to use the full bandwidth of disk set i is to be able to direct to disk i, a load equal to the disk set bandwidth B_i. The maximum load that can be directed to any disk set i is achieved when all disk blocks on this disk set contain distinct data blocks (i.e. there is no replicated block with both copies on disk set i), and all requests to data for which there is a copy on disk set i are directed to disk set i. This maximum load can thus be computed as $\lambda_{i_{max}} = \lambda_{max} \frac{S_i}{D} = \rho B(1+r)s_i$.

Case 1: Full bandwidth utilization ($\rho = 1$)

To use all the available disk bandwidth on the system, we need to have $\lambda_{i_{max}} \geq B_i$ for all disk sets, when $\rho = 1$. This condition is satisfied when $bs_i \leq (1+r)$, for all i ($1 \leq i \leq M$). Therefore the minimum required replication to enable the use of the full aggregate disk bandwidth is

$$r \geq max(bs_i) - 1 = bs_M - 1 \qquad , \text{for } \rho = 1 \tag{1}$$

Assuming we have at most two copies of each data block, i.e. $r \leq 1$, a required condition to be able to use the full bandwidth of the system is $bs_M \leq 2$.

Case 2: Partial bandwidth utilization ($\rho < 1$)

Now, consider the case where the desired disk utilization is $\rho < 1$. In this case, some disk sets may have a maximum directed load less than their bandwidth, $\lambda_{i_{max}} < B_i$. Let u_i be the normalized unused bandwidth on disk set i when the maximum possible load is directed to disk set i, i.e. $u_i = max(0, (B_i - \lambda_{i_{max}})/B) = max(0, b_i - \rho(1+r)s_i)$. To achieve a disk utilization ρ we must ensure that the total unused bandwidth is limited to $\sum_{i=1}^{M} u_i \leq 1 - \rho$. Let $rmin_j$ be the minimum replication that ensures that the first j disk sets do not have unused bandwidth, i.e. $u_i = 0$ for $i \leq j$, and thus, from the definition of u_i, $rmin_j = (bs_j/\rho) - 1$. For a given replication r, $0 \leq r \leq 1$, the number of disk sets that have the full bandwidth available is given by

$$nf(r) = \begin{cases} 0 & r < rmin_1 \\ j & rmin_j \leq r < rmin_{j+1}, 1 \leq j \leq M-1 \\ M & r \geq rmin_M \end{cases} \tag{2}$$

The condition required for the system being able to achieve a desired utilization ρ is that the available normalized bandwidth be greater or equal the utilization ρ, i.e. $\rho \leq 1 - \sum_{i=nf(r)+1}^{M} (b_i - \rho(1+r)s_i)$. Rewriting this condition in terms of r, we obtain the following condition on the required replication for a given utilization ρ.

$$r \geq \frac{\sum_{i=1}^{nf(r)} (s_i - b_i/\rho)}{1 - \sum_{i=i}^{nf(r)} s_i} \qquad , \text{for } \rho < 1 \tag{3}$$

3.2 Determining how the minimum replication should be distributed on the disks

Conditions 1 and 3 provide a lower bound on the replication necessary to achieve a disk utilization ρ. To show that this condition is also sufficient we need to show that there is a appropriate distribution of the replicated data on the disks, for this minimum replication level, that ensures that the system achieves the desired disk utilization ρ. We show that this is always possible, once the minimum replication condition is satisfied, by determining how the replicated blocks should be distributed in all possible situations.

Suppose we use the minimum replication r that satisfies the necessary condition to achieve disk utilization ρ. In this situation, if the system load is set at the maximum possible average rate $\lambda_{max} = \rho B$, all requests to replicated data blocks will be directed to the copy on the relative faster disk set, which we refer to as primary copy; while the other copy, which we refer to as secondary copy, will never be used. A careful reader may observe that this situation is equivalent to having unused space on the relative slower disks, since the secondary copies are never used. Although this is true in this particular situation, note that for any load less than λ_{max} or replication greater than the minimum, secondary copies will be used for dynamic load balancing and will reduce short term load imbalances. Therefore, having replicated blocks is in fact better than unused space. Nevertheless, for the purpose of analyzing the long range load balance described here one may think of secondary data copies as unused space. Moreover, although the selection of blocks in which to store primary copies and non replicated copies may be important for short term load balance, they are not important for the purpose of long range balance, since they will be responsible for the same amount of long range load on the disk they are stored, and thus it will not be considered here. Our problem now is reduced to determining how secondary copies of data blocks should be distributed. More precisely, we want to determine the fraction of data blocks rs_i that is replicated and have their secondary copy stored on disk set i.

Condition 3 above, was obtained assuming that the load directed to disk sets $i > nf(r)$ is set to the maximum possible to reduce the amount of unused bandwidth, and thus these disk sets should not store secondary copies of data blocks, but only primary copies or non replicated data blocks. Therefore $rs_i = 0$ for $i > nf(r)$. Now, let's consider disk sets $i \leq nf(r)$. Note that for $\rho = 1$, nf(r) = M.

For the maximum load λ_{max} and minimum replication that satisfy condition 3, all disk sets $i \leq nf(r)$ will have a load equal to their bandwidth, since we assumed that these disk sets do not have unused bandwidth. On other hand the load directed to any disk set i is proportional to the number of data blocks on it that do not store secondary copies of replicated data blocks. Therefore the following condition must be satisfied

$$\rho \frac{(S_i - rs_i D)}{D} = \frac{B_i}{B} = b_i \tag{4}$$

From the above equation we can compute the relative number of secondary copies rs_i for each disk set i

$$rs_i = (1 + r)s_i - \frac{b_i}{\rho} \qquad , i \leq nf(r) \tag{5}$$

Note that, if we sum the number of secondary copies rs_i over all disk sets, noting that $\sum_{i=1}^{M} rs_i = r$, we obtain a replication level r equal to the one defined by condition 3 if $\rho < 1$, as expected. Now, let's verify that this distribution of secondary data copies is always possible.

case 1: $\rho = 1$

The distribution of secondary copies as described in Eq.5 is feasible if and only if the number of required secondary copies in each disk set is non negative and at most equal to the total number of blocks in the disk set, i.e. $0 \leq rs_i D \leq S_i$. Since $b_i > 0$, $rs_i D < (1+r)s_i D = S_i$ and the first condition on rs_i is satisfied. If the minimum replication defined in condition 1 is satisfied, we obtain $rs_i \geq bs_i s_i - b_i = 0$, and thus the second condition on rs_i is also satisfied. Therefore, the secondary copies of replicated data can always be distributed as defined in Eq.5 for $\rho = 1$, once the minimum replication condition 1 is satisfied, and thus the system can fully utilize the aggregate disk bandwidth.

case 2: $\rho < 1$

Since $rs_i = 0$ for $i > nf(r)$, we just need to show that rs_i is feasible for $i \leq nf(r)$. For the same reason as in the case $\rho = 1$ we always have $rs_i D < S_i$. From the definition of $nf(r)$ given in Eq.2, we have $r \geq (bs_{nf(r)}/\rho) - 1$ and thus $r \geq (bs_i/\rho) - 1$, for $i \leq nf(r)$. Therefore $rs_i \geq (bs_i/\rho) - b_i/\rho = 0$, and both conditions on rs_i are satisfied. We then conclude that for the case $\rho < 1$, it is also always possible to distribute the secondary copies of replicated data blocks as defined in Eq.5, and thus achieve an aggregate disk bandwidth of ρB.

As a last comment, to ensure that the load is equally distributed inside a particular disk set, we must equally distribute the secondaries copies among the disks of that disk set.

3.3 The particular case of a system with only two types of disk

Now, consider the particular case where there is only 2 disk sets ($M = 2$). Define $\alpha = S_1/S_2 = s_1/s_2$ and $\beta = B_1/B_2 = b_1/b_2$ as the ratio of space and bandwidth, respectively, provided by each disk set. From this definitions we can also write $bs_1 = \beta(\alpha+1)/\alpha(\beta+1)$ and $bs_2 = (\alpha+1)/(\beta+1)$. Note that, since $bs_1 \leq bs_2$, we have $bs_1 \leq 1$, $bs_2 \geq 1$, and $\beta \leq \alpha$.

Now, consider the space (α, β), $\alpha > 0$, $\beta > 0$, and $\beta \leq \alpha$. We are interested in the region of this space where it is possible to use replication to balance the long range load and what is the minimum replication required for each point in this region. For a desired maximum disk utilization, $\rho = 1$, the condition to be satisfied is $r \geq bs_2 - 1 = (\alpha - \beta)/(\beta + 1)$. From this condition we can compute the region in the (α, β) space where the long range load balance allows full utilization of the disk bandwidths as

$$\frac{\alpha - r}{r+1} \leq \beta \leq \alpha \qquad , \rho = 1 \tag{6}$$

If the desired maximum disk utilization is $\rho < 1$, then the condition to be satisfied is condition 3. Since

$\rho < 1$, some bandwidth of disk set 2 is not used and thus $nf(r) \leq 1$. However, we are interested in the point where the system reaches saturation which correspond to using all bandwidth on the relative slower disk set $i = 1$, and thus at saturation we must have $nf(r) = 1$. In this situation, condition 2 is reduced to $r \geq (s_1 - b_1/\rho)/s_2 = \alpha - \beta(\alpha+1)/\rho(\beta+1)$. From this condition we can compute the region on the (α, β) space where the long range load balance allows disk utilization $\rho < 1$ as

$$\frac{\rho(\alpha - r)}{\alpha + 1 - \rho(\alpha - r)} \leq \beta \leq \alpha \qquad , \rho \leq 1 \tag{7}$$

Note that the region defined in Eq.7 is reduced to the region defined in Eq.6 when ρ is set to 1.

Using this condition, we plot in Figure 2 the boundaries of the regions where it is possible to achieve a disk utilization $\rho = 0.9$ and $\rho = 1$ for different levels of replication. Each solid curve in the figure defines the minimum value of β to achieve the desired bandwidth utilization, while there is no limit on the maximum value of β, except that all systems are assumed to have $\beta \leq \alpha$ (assumption of disk sets ordered according to their BSR). As we can observe, a significant range of the (α, β) space $(\alpha > 0, \beta > 0, \beta \leq \alpha)$ is covered if we use some replication $r > 0$ and/or sacrifice a small fraction of the available bandwidth.

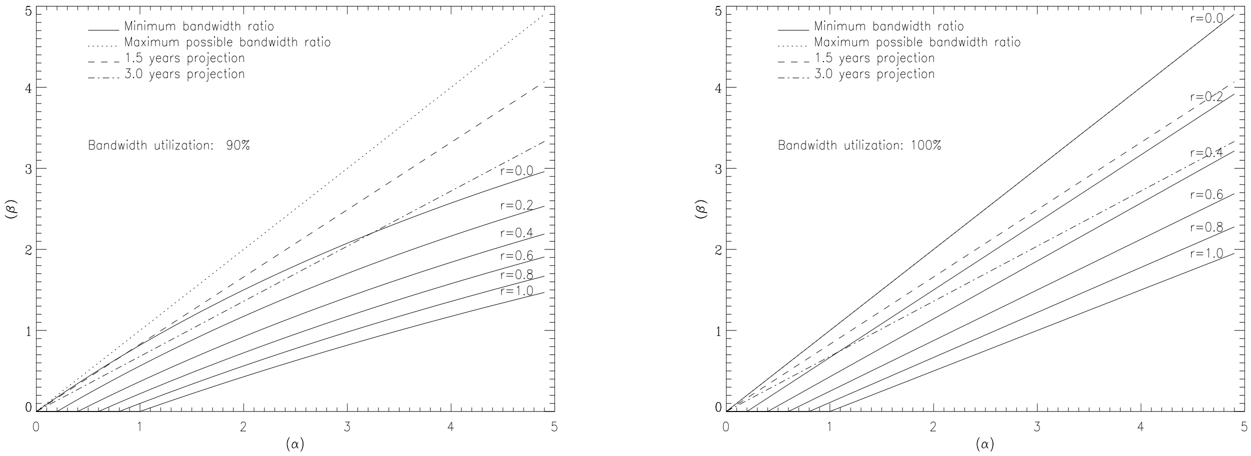

Figure 2: Regions in the space (α, β) that achieve bandwidth utilizations of 90% and 100%

3.4 System Upgrade Example

Now consider a particular example of a system with a uniform set of disks that will be upgraded after a period of time by adding some new disks. Since disk technology is constantly improving, at the time of the upgrade the disks available on the market will most likely have different storage space and bandwidth characteristics, resulting in a heterogeneous disk system. Storage capacity in disk drives has been increasing at a rate of approximately 60% a year, while disk bandwidth has been increasing at a rate of only approximately 40% a year [9]. Assuming that this trend will continue in the next few years, we predict the bandwidth and storage capacity of the new disks at the time of an eventual upgrade, assuming the system is initially configured with disks currently available. Since storage space is increasing at a higher speed than bandwidth, the new disks will have BSR lower than the old disks, and will be the more stressed ones. In a year and a half, disk drive capacity is projected to have increased two fold while bandwidth increased only 1.66 times. In three years the increase should be four fold for drive capacity and only 2.74 times for bandwidth. Of course, at any point in time there is a range of disk drives available on the market with different BSR, and one could select the ones that have BSR closer to the original ones. For our example however, we will simply assume disks with characteristics projected from the originally used disk.

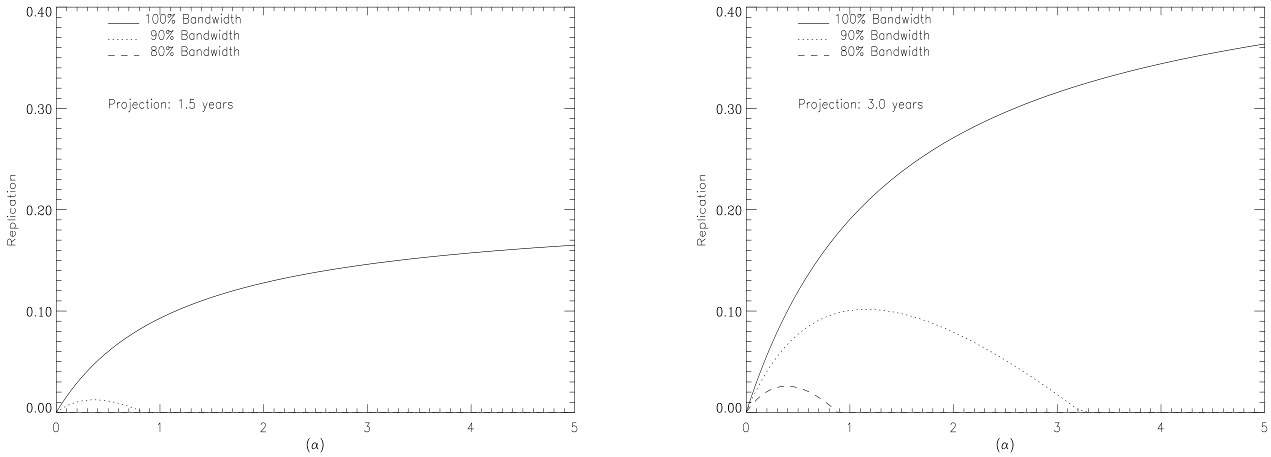

Figure 3: Minimum required replication

Assuming these projected disks characteristics, an upgrade in 1.5 and 3 years will generate heterogeneous systems with $\beta = 0.83\alpha$ and $\beta = 0.68\alpha$, respectively. These relations of α and β for the two cases are also plotted in Figure 2. The exact point in these lines for a particular configuration is determined by the ratio of the number of new and old disks; the larger the number of new disks, the larger is α. The intersection of these curves with the solid line curves define the minimum required replication to achieve the desired bandwidth utilization as function of the the relative additional storage space α added to the system. Figure 3 show this minimum required replication for 1.5 and 3 years projection. We observe that for an upgrade in three years it is possible to have 100% bandwidth utilization for any amount of added storage capacity, using less than 40% of replication, and if we just double the storage space ($\alpha = 1$) we just need about 20% replication (Although Figure 3 shows only values for $\alpha \leq 5$, the curves for 100% bandwidth asymptotically approach $(1-\beta/\alpha)/(\beta/\alpha)$ for $\alpha \rightarrow \infty$) . If we are willing to sacrifice a small fraction of the total bandwidth, the amount of replication can be significantly reduced, as observed in Figure 3 for the cases of $\rho = 0.8$ and $\rho = 0.9$. For example, for 90% bandwidth utilization we do not need any replication if we add more than three times the current storage space or we just need about 10% replication if we double the system capacity. For a 1.5 year upgrade the required replication is much less. For upgrades done after a period much longer than three years, having 100% bandwidth utilization may not be be possible for large values of α, but it will always be possible to achieve a high bandwidth utilization with enough replication. However, as the new drive characteristics start becoming much different from the original ones, after more than three or five years, it will be more cost effective to simply substitute the original disks by new ones, since the relative cost of storage capacity or I/O bandwidth of new disks will be much lower than the original ones and they could be replaced for a fraction of their original cost. Replacing the original disks has the benefit of a simpler system that is homogeneous and additionally will provide a lower delay bound guarantee, since the new disks will be able to service requests with smaller delays (This will be clearer when we analyze real-time performance guarantees of the system in the next section). Therefore, replacing the original disks by new ones would be preferred after long periods of time, where the replacement cost is a small fraction of the original disk cost.

Note that the results presented in this section consider only long range load balance and the bandwidth utilization achieved for a given replication level is just the maximum load that will cause some of the disks to reach saturation. However, to ensure real-time delay bounds we need to operate at a maximum load that is lower than this maximum throughput, such that the probability of delays greater than the delay bound is very small. In the following section we will analyze the performance of the system and evaluate the effect of the replication level on the delay bound that can be guaranteed with high probability as a function of the load.

4 Performance Analysis of RIO with Heterogeneous Disks

In this section we evaluate the real-time performance of RIO with heterogeneous disk configurations, analyzing results obtained using the simulator described in Section 2. We first present simulation results for a system with uniform BSR and then consider systems with non uniform BSR corresponding to upgrades in

1.5 and 3.0 years.

4.1 Disks with Uniform BSR

Let's consider a heterogeneous disk system with two types of disk. We already know, as observed in the performance results for homogeneous disks configurations shown in Section 2, that higher levels of replication allow smaller delay bounds for the same input load, due to better load balance among the disks. In a heterogeneous system, however, given the amount of replicated data, we still need to evaluate what is the best way to distribute the replicated data blocks among the disks. A replicated block can be replicated either on two disks of the same type, or on disks of different types. We call these two types of replication, homogeneous and heterogeneous replication, respectively. The basic question we want to answer is, what is the best combination of heterogeneous and homogeneous replication that provides the best real-time performance?

To answer this question we consider systems with uniform BSR and 100% replication, and evaluate the effect of different combinations of homogeneous and heterogeneous replication on system performance. Figure 4 shows performance results for systems with uniform BSR for two different disk capacity ratios, 2 and 4. For each case we consider two possible configurations for the number of disks, 16/8 and 16/16. These configurations covers the range $1 \leq \alpha \leq 4$. The delay bound on the y axis of all graphs in this figure, as well as on the rest of the performance graphs in the paper, is normalized by the average time to read a single data block on the slower disk. For each configuration we simulated the system with different distributions of replicated data corresponding to 0%, 50% and 100% of the maximum heterogeneous replication possible. Note that having maximum heterogeneous replication does not mean having no homogeneous replication, since whenever $\alpha \neq 1$, there are a different number of blocks on each disk set and some of the blocks on the larger set will have to be replicated on the same set, for lack of additional blocks on the other set. Therefore 100% of the maximum heterogeneous replication corresponds, in our examples, to the case where all data blocks on the slower disks are replicated on the faster disks, with the extra blocks on the faster disks being homogeneous replicated. Of course, for each type of replication, homogeneous or heterogeneous, all combinations of homogeneous or heterogeneous disk pairs, respectively, are equally likely.

As we observe in Figure 4, higher levels of heterogeneous replication provides better performance, i.e.. allows lower delay bounds for the same load and probability of satisfying the delay bound, for all configurations. This can be explained as follows. If no heterogeneous replication is used, the system behaves as

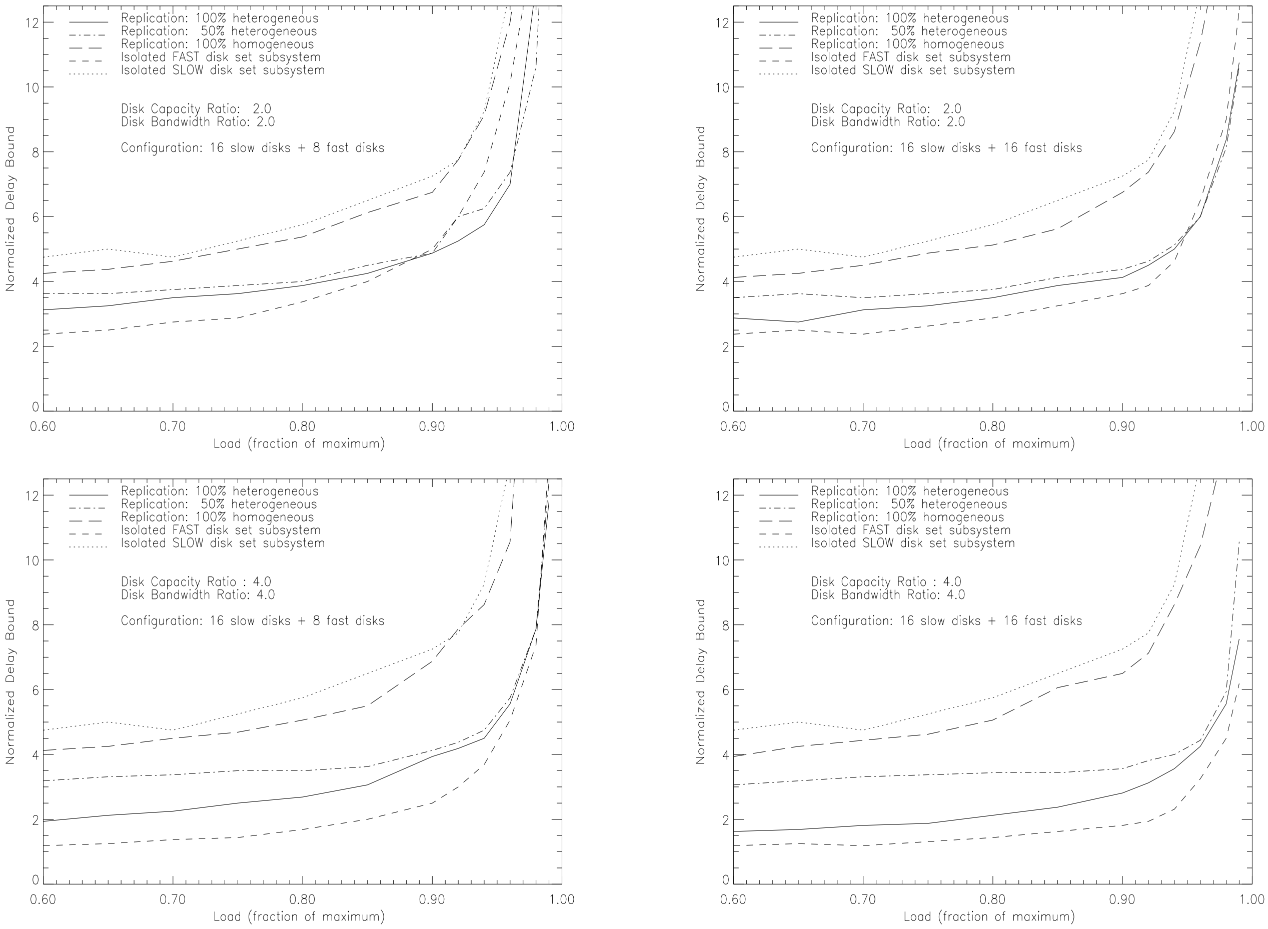

Figure 4: Performance of heterogeneous disk system with uniform BSR

two independent systems (one for each disk set), with the same relative load. Since the relative load is the same in both sets, they have approximately the same disk queue size distributions. But, since the average time to service a request is different in both sets, request delays on the slower set is twice or four times (depending on the bandwidth ratio of disks) longer than request delays on the faster set. Thus, the delay bound is mostly determined by the slower disk set, while the faster set could satisfy requests with much lower delay bounds. If heterogeneous replication is used, this asymmetry is reduced, since the system can now direct a relatively higher load to the faster disk set than to the slower disk set, and thus have more uniform distribution of delays on both sets. This behavior is a consequence of the online load balancing algorithm that always directs requests to the disk with lower mean expected delay.

For comparison, Figure 4 also plots the performance of homogeneous systems composed by only one of the disk sets. We observe that, in fact with 0% of heterogeneous replication, the performance of the heterogeneous system is dominated by the slower disk set, indicated by the fact that its performance is close to the performance of the slower disks. The performance of the heterogeneous system is slightly better than the slower disks, in this case, because on the heterogeneous system the fraction of the load directed to the faster disks has much lower delays and reduces the fraction of total requests that exceed any particular delay bound. As the level of heterogeneous replication is increased the performance of the heterogeneous system approaches the performance of the faster disks. Therefore, for optimizing performance we should maximize the amount of heterogeneous replication, since in this case more requests will have a chance to be serviced by a faster disk and have a lower delay.

4.2 System Upgrade Example

Now consider the system upgrade examples of Section 3.4 that have non uniform BSR. As discussed previously, some amount of replication is needed for long range balance to allow potential use of the full aggregate bandwidth. The basic question we want to answer in this section is whether the use of replication for long range load balance reduces the capability of the system to achieve short term load balance. In other words, is the short term load balancing, and thus the delay bound guarantees, a function of *only* the additional replication in excess of the replication necessary for long range load balance, or the replication necessary for long range load balance also helps the short term load balance. To answer this question we simulated the performance of systems upgraded in 1.5 and 3.0 years, with three different amounts of added storage capacity, 100% ($\alpha = 1$), 200% ($\alpha = 2$) and 400% ($\alpha = 4$). We assumed an initial configuration of sixteen slower disks. Since the storage capacity should increase two fold in 1.5 years and four fold in three years, the number of new disks added for the three values of α is 8, 16, and 32 for a 1.5 year upgrade, and 4, 8, and 16 for a 3.0 years upgrade. To evaluate the effect of replication for long range load balance on the delay bounds that can be guaranteed, we compare the performance of the upgraded system with a hypothetical system with the same number of disks and the same bandwidth per disk that has a uniform BSR. We simulated the system with three different levels of replication: 25%, 50% and 100%. In the case of partial replication, we distributed the non replicated blocks on the disks, in amounts proportional to their storage capacity. The remaining space in each disk is used to store replicated data blocks and the maximum level level of heterogeneous replication is used, since the last section showed that this provides the best performance.

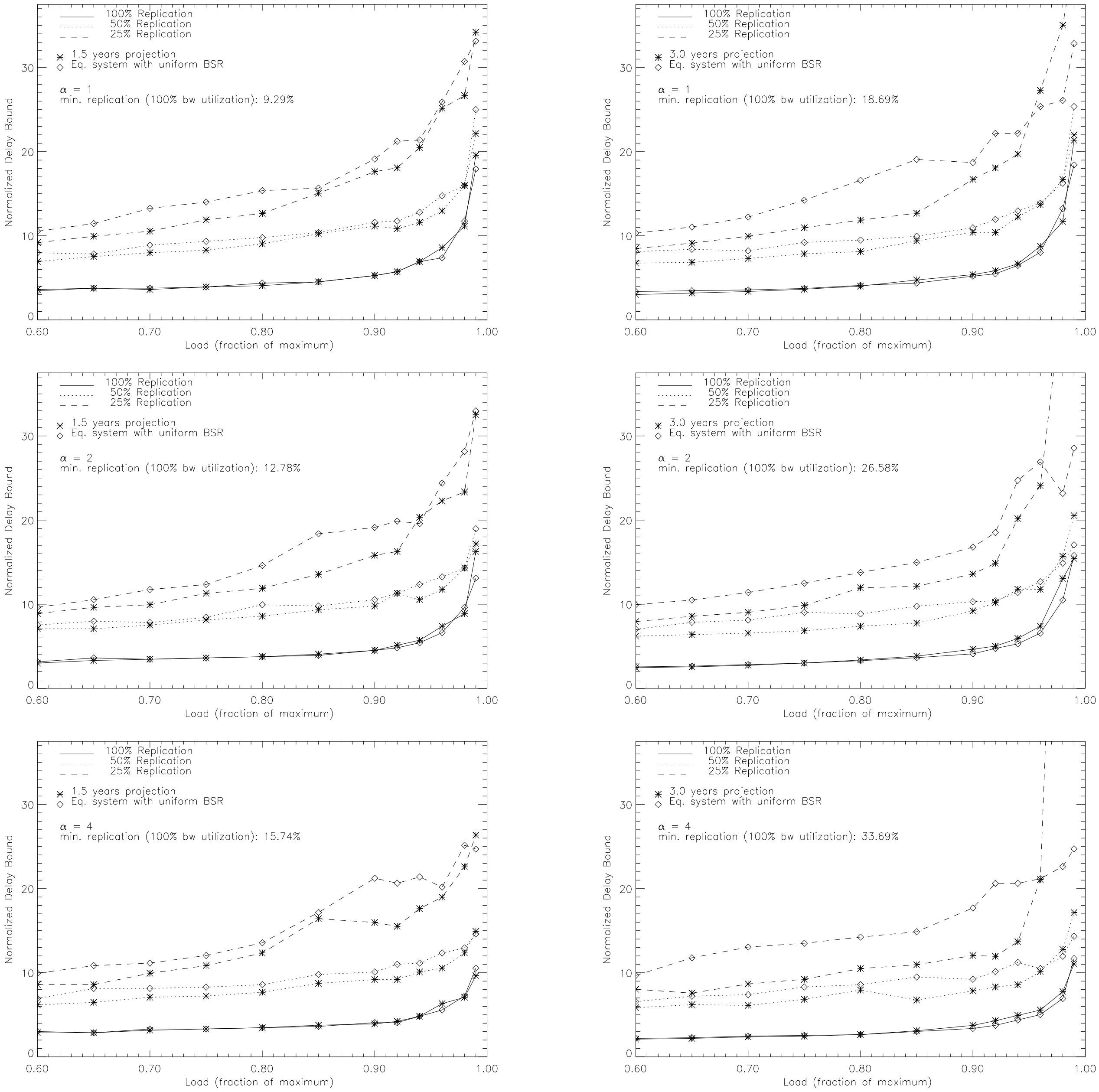

Figure 5: Performance of heterogeneous disk system obtained by an upgrade in 1.5 and 3.0 years

Figure 5 shows the simulation results for the selected configurations. As expected, higher levels of replication allow lower delay bounds for the same load, due to better load balance. An interesting result though, is that the performance of the upgraded system does not seem to be degraded because the BSR is not uniform, since its performance is equivalent to the uniform BSR case with 100% replication, and even better than the uniform BSR case when partial replication is used. This can be explained as follows. Although, when the system is close to the saturation point the load has to be distributed in amounts proportional to disk bandwidths, to avoid an earlier saturation, at lower loads it is in fact better to have the fraction of the load directed to the faster disks higher than proportional to their bandwidth since they provide lower I/O times, as discussed in Section 4.1. Since the new disks on the upgraded system have proportionally more space than bandwidth, they will naturally tend to be used at a higher utilization than the slower disks. With 100% replication, the system adapts to the difference in the I/O times of each disk set and effectively distributes the load to achieve uniform delays, by having higher disk utilization on the faster disks. Therefore, for both uniform and non uniform BSR, the system will adapt the load on each disk set to have the best performance. This explains why the performance of the upgraded systems is similar to that of an equivalent system with uniform BSR. This equivalence shows that the replication used for long range load balance is also effectively used for short term load balancing and thus answers our basic question stated on the beginning of this section.

For partial replication, the fraction of requests directed to non-replicated data blocks are distributed to disks proportionally to their storage capacity. This causes the non replicated load to be proportional to the disk bandwidths on the system with uniform BSR, as opposed to the upgraded system case, where a higher relative load is directed to faster disks. This characteristic favors the upgraded system, since a higher number of requests are served on the faster disks, which have lower I/O times. This explain the better performance of the upgraded system, with partial replication, when compared to a system with uniform BSR. Of course, this could be a disadvantage if the disk utilization of the faster disks became much higher than the optimal, if the difference of the BSR was too high, but for the BSR of the examples analyzed this turned out to an advantage. Of course, in both cases, a non uniform distribution of non replicated data could improve performance. An analysis of the best way to distribute non replicated data is left as future research.

In summary, replication in RIO is effectively used for both long range and short range load balance and the fact that some replication is used to balance long range load, does not reduce the effectiveness of replication for short range load balance.

5 Conclusion

We studied the performance of the RIO storage system with heterogeneous disk configurations. We first analyzed how replication can be used to provide long range load balance in order to allow the potential use of the aggregate disk bandwidth. We showed that for a system with two disk types, it is possible to use a high percentage or even 100% of disk aggregate bandwidth on a significant range of the space of possible bandwidths, capacities ratios, and number of disks used of each type, if we use some amount of replication. We showed that for practical system upgrades, in a few years 10% to 40% replications is sufficient to allow the potential use of the full disk aggregate bandwidth.

We also showed that replication can be effectively used for short term load balance, reducing the delay bound that can be guaranteed on a heterogeneous system. Moreover, we showed that replication is not used exclusively for long range or for short range load balance, but the amount of replication necessary for long range load balance also helps the short term load balance, improving real-time performance. We also noted that having disks with non uniform bandwidth space ratios is not necessarily a disadvantage, but it may be an advantage in practical configurations. Better performance can be achieved if the faster disks operate at a higher disk utilization, since these disks provide lower response times, what naturally happens on an upgraded system, since new faster disks have proportionally more storage space than bandwidth.

To conclude, RIO can be effectively used with heterogeneous disk configurations, without considering popularity of data objects, which is the approach used by other multimedia storage servers [5]. Our approach thus avoid all problems involved in accurately predicting object popularity and avoids the cost of reconfiguring the system when data popularity changes. In summary, RIO provides a simple but effective way of dealing with heterogeneous disks.

References

[1] S. Berson, R. Muntz, S. Ghandeharizadeh, X. Ju "Staggered Striping in Multimedia Information Systems", *SIGMOD 94*, p.79-90, May 1994.

[2] S. Berson, R. Muntz, W. Wong, "Randomized Data Allocation for Real-time Disk I/O", *Compcon 96*, p.286-90, 1996.

[3] Birk, Y., "Random RAIDs with Selective Exploitation of Redundancy for High Performance Video Servers", *7th International Workshop on Network and Operating System Support for Digital Audio and Video Proceedings - NOSSDAV 97* , St. Louis MO, p. 13-23, May 1997.

[4] A.L. Chervenak, A.A. Patterson, R.H. Katz, "Choosing the Best Storage System for Video Service", *ACM Multimedia 95*, p.109-119, 1995.

[5] A. Dan, D. Sitaram,, "An Online Video Placement Policy based on Bandwidth to Space Ratio (BSR)" *SIGMOD 95*, San Jose CA, p.376-85, 1995.

[6] A. Dan, M. Kienzle, D. Sitaram, "A Dynamic Policy of Segment Replication for Load-Balancing in Video-on-Demand Servers", *Multimedia Systems*, vol.3, no.3, p.93-103, July 1995.

[7] D.L. Eager, E.D. Lazowska, J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems", *IEEE Trans. on Software Engineering*, pp. 662-675, 1986.

[8] F. Fabbrocino, J.R. Santos, and R. Muntz, "An Implicitly Scalable Real-Time Multimedia Storage Server" *UCLA Computer Science Department Technical Report #980009*, March 1998.

[9] E. Grochowski, R.F. Hoyt "Future Trends in Hard Disk Drives", *IEEE Transactions on Magnetics*, Vol. 32, No. 3, May 1996.

[10] J. Korst, "Random Duplicated Assignment: An Alternative to Striping in Video Servers" *ACM Multimedia 97*, Seattle WA, p.219-26, 1997.

[11] M.D. Mitzenmacher, "The Power of Two Choices in Randomized Load Balancing", *PhD Dissertation*, University of California at Berkeley, Computer Science Department, 1996.

[12] R. Muntz, J.R. Santos, S. Berson, "A Parallel Disk Storage System for Real-time Multimedia Applications", Special Issue on Multimedia Computing Systems of the International Journal of Intelligent Systems, 1998. (to appear)

[13] J.R. Santos, R. Muntz, "Using Heterogeneous Disks in RIO Multimedia Storage System" *UCLA Computer Science Department Technical Report #980010*, March 1998.

[14] R. Tewari, R. Mukherjee, D.M. Dias, H.M. Vin, "Design and Performance Tradeoffs in Clustered Video Servers" *Proceedings of the International Conference on Multimedia Computing and Systems*,Los Alamitos, CA, IEEE Comput. Soc.Press, p.144-50, 1996.

[15] F.A. Tobagi, J. Pang, R. Baird, M. Gang, "Streaming RAID - A Disk Management System For Video Files" *ACM Multimedia 93*, p.393-400, 1993.

[16] J.L. Wolf, P.S. Yu, H. Shachnai, "DASD Dancing: A Disk Load Balancing Optimization Scheme for Video-on-Demand Computer SYSTEMS". *SIGMETRICS 1995*,pp.157-166, 1995.