Improving Disk Performance Via Latency Reduction

Spencer W. Ng, *Member, IEEE*

Abstract—**Due to the great disparity between the access time of main memory and the average response time of disk storage, there has always been a strong need to improve the performance of disks. Even for large computing complexes where system queueing delays may overshadow the disk service time, the disk's basic service time is the gating factor. This paper illustrates the fact that in many of today's environments the rotational latency and the RPS miss delay are the major contributors to a disk's basic service time. Then, by doing a sensitivity study using a simple analytical queueing model, this paper demonstrates that a reduction in these two components (both of which are related to the rotation of disk drives) has the greatest impact in reducing the disk's basic service time and in turn produces the greatest improvement in overall subsystem performance.**

While the most straightforward way to reduce latency and RPS miss penalty would be to increase the disk's rotation speed, this paper explains some of the limitations to such an approach. Several alternatives to reducing latency and RPS miss penalty are proposed and explored here and their performance is analyzed using analytical queueing models.

Index Terms—**Data storage systems, dual copy, fast write, performance analysis, queueing model, rotation latency, synchronous disks.**

I. Introduction

SINCE the 1950's, when the magnetic drums and disks were introduced as secondary storage to augment the main memory of computing systems, there has always been a great disparity between the data access time of the main memory and the response time provided by the secondary storage devices. This disparity is mainly due to the inherent mechanical nature of the secondary storage devices. While CPU's memory speed has increased from about 10 μs in the 1950's to about 100 ns today for 2 orders of magnitude in improvement [13], disk access time has only improved from hundreds of milliseconds to tens of milliseconds during the same period for one order of magnitude in improvement [5], [15]. This trend is expected to continue in the near foreseeable future.

As a result of the slow response time of the secondary memory, many application programs are I/O-bound. To improve the CPU's utilization, the standard technique is to introduce multiprogramming so that other programs can execute in the CPU while some programs are waiting for the completion of their I/O's. While this technique helps to improve resource utilization and total job throughput, it does not decrease the job completion time of a program. The only way to achieve that is to improve the response time of the storage device.

For quite some time the main focus of research in reducing a storage device's response time has been on improving the seek time. The approaches include both hardware (faster mechanical actuators) [5] and software (disk arm scheduling) [4], [12], [16]. Recently there are also research activities on increasing the data transfer rate [8], [11]. However, there does not seem to be much attention given to reducing a rotational storage device's latency time. It is the purpose of this paper to demonstrate that in many of today's computing environments reducing rotational latency time is the most important factor in improving the I/O response time. Several methods for achieving this goal, other than the obvious one of increasing the rotational speed, are considered in this paper. Their performance characteristics are investigated and the practicality of their implementations are discussed. While other effective methods, such as caching, track buffering, and adding extra paths or controllers, are known for improving the overall performance of I/O systems, they are not discussed here for two reasons. First, the focus of the paper is on the importance of latency reduction. Second, all those methods are orthogonal to the latency reduction techniques discussed in this paper, and can, therefore, be applied on top of these techniques.

While magnetic storage has been used exclusively in the past 30 years as secondary memory, optical storage is starting to come on the scene. It is expected that, at least initially, the optical devices will typically have a slower seek time and rotation speed compared to magnetic devices. While the parameters used in the analysis in this paper are those of a typical magnetic disk drive, most of the conclusions should also apply to optical disks. Furthermore, some writable optical disks require an erasure before any write. For such devices the write latency is particularly long since one revolution must be added to the normal latency and data transfer. Thus, reducing this long latency is also critical in improving an optical disk's performance.

A. Method of Study

A simple M/M/1 analytic queueing model [1] of a single channel/control unit storage subsystem [19], [20] is used as the primary study tool for this investigation. Although a high-end storage subsystem is typically more complex than this simple model, having multiple channels and multiple control units, such a model is quite adequate for the purpose of studying *qualitatively* the effects of changing a disk drive's various characteristics.

For a disk subsystem, an I/O operation consists of six phases, as shown in Fig. 1. The six phases are:

1) Wait in queue—time spent in a queue waiting for the drive to be free to handle the I/O.
2) Wait for channel—time spent waiting for the channel to be free so that seek and sector information can be sent down.
3) Seek—time for positioning the arm to the target cylinder.
4) Latency—time for the desired point on a track to rotate under the head.
5) RPS miss—time lost if the disk must rotate extra revolutions if the channel is not available when the desired point is under the head.
6) Overhead and data transfer—time for control unit overhead and the actual data transfer.

For each I/O operation in this model, the disk drive is busy during phases 3–6 whereas the channel is busy (for this I/O) only during phase 6. While all six phases constitute a disk's *response time* for an I/O, only phases 3–6 are a disk's *basic service time.* Most of the above items will be discussed in greater detail in the next section.

Manuscript received June 13, 1988; revised August 24, 1989.
The author is with the IBM Almaden Research Center, San Jose, CA 95120.
IEEE Log Number 9040680.

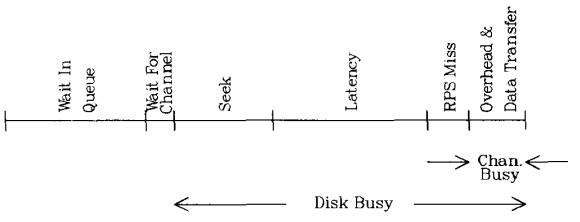


Fig. 1. Time events of an I/O.

The analytic queueing model for the disk subsystem is as shown in Fig. 2. Each disk drive in the subsystem is an individual server and has its own queue. The channel/control unit is a common server whose service is required by all I/O's. An exponential I/O interarrival distribution (Poison process) is assumed for each drive, with λ_i the arrival rate for drive i. It is generally known that in a real system not all drives have the same I/O rate. Instead, I/O requests are skewed such that some drives are much more heavily used than others. A common mathematical way to characterize this skewing is termed "Sth degree of skew" in which if the least busy device has one I/O per second, then the k least busy devices has $k^{(S+1)}$ I/O's per second. A 0 degree skew means flat or uniform distribution. Finally, the overall response time of the subsystem is simply the average of the response times of all the drives weighted by their skews.

In this paper, when comparing the performances of different subsystems or configurations, only subsystems or configurations with equal amounts of storage capacities will be compared. In particular, we will start with a base device type and a disk subsystem with a given number of those devices. Then, any low-latency device configuration considered will be configured in a subsystem such that the total capacity matches that of the base subsystem. The performance of this new subsystem will then be compared to that of the original base subsystem.

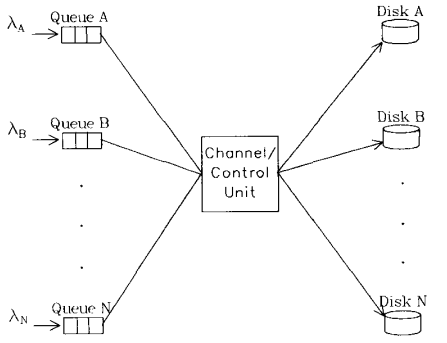


Fig. 2. Disk subsystem model.

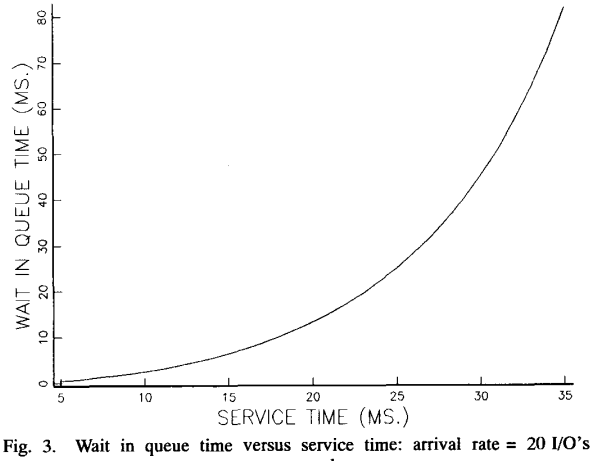


Fig. 3. Wait in queue time versus service time: arrival rate = 20 I/O's per second.

II. Disk Basic Service Time Components

In the previous section, the major components of a disk drive's response time were introduced. In this section, we will examine these components more closely. A high-end IBM mainframe type of environment such as TSO, VM, or IMS will be assumed, and a disk drive with IBM 3380 [3], [5] like characteristics will be used in the discussion and analysis.

The wait in queue time accounts for a very substantial portion of an I/O's overall response time [14]. This component is mainly a function of the disk's I/O rate and its basic service time for each I/O. Fig. 3 shows the relationship between queueing time and service time for a single device at an I/O rate of 20 I/O's per second, assuming a simple M/M/1 queueing model. Thus, for a given subsystem configuration and I/O rate, the best way to reduce the queueing time is to reduce the basic service time. The wait for channel time is a function of the subsystem's I/O rate and the channel busy time (overhead and data transfer) for each I/O. In general, this component is not a big contributor to the overall response time. In the following, we will look at the components of the disk's basic service time and evaluate their relative importance.

A. Seek Time

Seek time has traditionally been the most important area of research effort for the reduction of disk service time. In the early days of disk design, when the average seek time for random seeks (also called nominal seek time) was one or two orders of magnitude larger than all the other components [5], this was the right area to attack. As a result of research and development efforts in this area, the nominal seek time for a 3380 type of DASD is now down to around 16 ms. Furthermore, this number assumes that seeks are totally random in nature. However, recent studies [9], [14] discovered that in a real environment the disk arm does not need repositioning the majority of times. Therefore, the actual mean seek time per I/O is even lower than the nominal seek time. For this study, a mean seek time of 1/3 of the nominal seek time of 16 ms, i.e., 5.33 ms, will be assumed, as observed by both [9] and [14].

Research was also done on disk arm scheduling policies as an alternative means of lowering the mean seek time [4], [12], [16]. The basic principle of such policies is to reorder I/O requests waiting in a queue so as to reduce the amount of arm movement. In order for such a scheme to work, there obviously needs to be a queue of reasonable size. However, in reality a properly tuned and managed storage system should not have long queues. In fact, [6] reports that an average queue length of about 0.3 should be the limit for heavily used devices. Under such a condition, application of any arm scheduling is not going to be productive. The low percentage of arm movement cited earlier also contributes to this conclusion [18]. For systems with a small

number of disks (one or two), disk arm scheduling may produce a more noticeable impact. If so, then the actual mean seek time is further reduced.

B. Overhead and Data Transfer Time

The overhead time is simply the time required by a control unit to take care of various housekeeping work such as table lookups, locking and unlocking resources, etc. This is typically a small number and for this study we will assume a mean value of 1.5 ms.

Data transfer time is dependent on two factors, namely the data transfer rate and the amount of data that needs to be transferred. Data rate of DASD's has steadily been improving over the years, increasing from 8.8 kilobytes/s for the IBM 350 in 1957 to 3 megabytes/s for today's 3380 [5]. This trend is expected to continue. In recent years, a method of increasing data rate by interleaving data across multiple disk drives has been proposed [8], [11], [17]. For example, four 3 megabyte/s disk drives operating in parallel on the same data set can yield a combined data rate of 12 megabytes/s. This technique, called disk striping, is effective in applications such as scientific computation or image processing in which a large volume of data need to be moved to/from storage at a time. On the other hand, in many of today's typical TSO, VM, or IMS applications the average amount of data being transferred is about 3 to 5 kilobytes per I/O. For such environments the benefit of a higher data rate is then somewhat limited. However, it is safe to assume that the trend is for transfer block sizes to increase in the future, making higher data rate an attractive approach for reducing a disk's basic service time. For this study, an average of 4 kilobytes of data is assumed to be transferred at a data rate of 3 megabytes/s, giving a mean data transfer time of 1.33 ms.

C. Latency

For a conventional disk subsystem, the head is at some random position on a track at the completion of a head seek. The average latency, which is the mean time for the disk to rotate from that random position to the one in which the desired point is under the head, is therefore half a revolution. Thus, the latency is dependent on the rotational speed of the disk drive. Most of today's drives, including the IBM 3380, rotate at 3600 RPM. Hence, the average latency is 8.3 ms.

The obvious approach to reducing latency is to spin the disk at a higher RPM. However, the rotational speeds of drives have not changed significantly during the past 20 years [5]. This is mainly due to three factors:

1) To rotate at a higher speed requires much more power (relationship is much worse than linear).
2) More heat is generated due to increased air friction which must be dissipated.
3) The rotational speed of a disk drive is intimately tied to its storage capacity (linear density) and the data rate. If the capacity is held fixed while the RPM is increased, the data rate also goes up proportionally. The analog recording channel and the channel to which the disk drive is attached may have to be redesigned to handle this increased data rate. If a constant data rate is to be maintained while the RPM is increased, then the capacity must be reduced accordingly. This option has historically been avoided.

Other effects such as the head's flying height, track miss registration (TMR), track servo bandwidth, signal-to-noise ratios, etc., also need to be considered when attempting to reduce latency by increasing the RPM.

D. RPS Miss Time

As seen from the previous discussion, in a conventional disk drive the mean latency is several times larger than the mean data transfer time. It is, therefore, not efficient for the disk drive to hold the channel/control unit during this time when other drives could be using the channel/control unit to transfer data or receive command to start a seek. To allow such concurrency to take place, today's DASD's such as the 3380 have hardware to detect the rotational position of the disk drive. Called rotational position sensing (RPS) [3], this is a hardware feature that allows the disk drive to disconnect from the control unit and the control unit to release the channel connection during latency. When the disk drive becomes rotationally ready, it attempts to reconnect with the control unit which in turn attempts to reconnect with the channel.

This works very well when the overall I/O rate of the storage subsystem is low. As the I/O rate increases, then it becomes more and more likely that when a disk drive is ready, the channel/control unit is busy servicing another drive. When this happens, which is called an *RPS miss,* the disk must rotate one full revolution (16.7 ms) before it is ready to attempt reconnection again. The mean RPS miss time for a storage subsystem with one channel/control unit can be mathematically modeled as follows. Let

- p_i probability of an RPS miss for disk i
- λ_i the I/O rate for all disks (in number of I/O's per second)
- λ the total I/O rate for all disks (in number of I/O's per second)
- T the mean channel busy time, i.e., overhead and data transfer time.

Then (note that $\lambda \leq \frac{1}{T}$)

$$p_i = \frac{(\lambda - \lambda_i) \times T}{1 - \lambda_i \times T}. \tag{1}$$

For disk i, the average RPS miss time is

$$\frac{p_i}{1 - p_i} \times \text{time for 1 RPS miss.} \tag{2}$$

It has been observed that (1) may be underestimating the probability of RPS miss [2]. However, for this study (1) is sufficient for the purpose of illustrating the effect of RPS miss delay on the total disk service time.

As can be seen in (1) and (2), the average RPS miss delay time is dependent on the I/O rate, the distribution (skew) of I/O's among the drives, the channel busy time, and the penalty time of one RPS miss. At this time we will assume the penalty time of one miss is 16.7 ms. Fig. 4 shows the weighted average RPS miss time for a subsystem with eight devices and no skew, a subsystem with eight devices and fifth degree skew, and 16 devices with fifth degree skew.

The preceding discussion is for a storage subsystem with a single channel/control unit. When there are multiple channels, then dynamic path reconnection technique as used in the 370/XA channel architecture may be applied. With such a technique, when a disk drive is rotationally ready to do data transfer it can attempt to establish a path back to the CPU through any one of the available channels. This helps to reduce the probability of an RPS miss. However, the analysis of the RPS miss probability is more complex and is beyond the scope of this paper, which assumes a single channel/control unit storage subsystem.

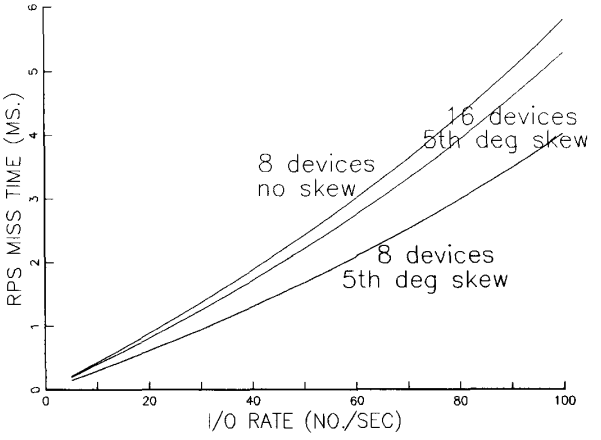


Fig. 4. Average RPS miss time.

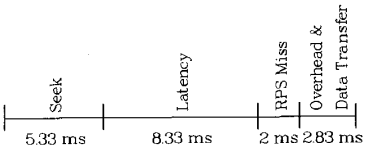


Fig. 5. Mean values of a disk's service time components.

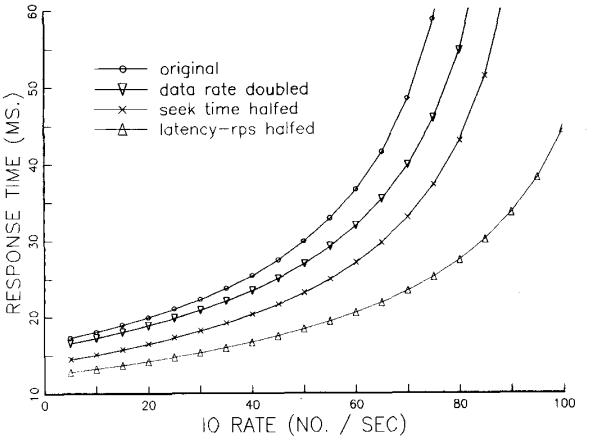


Fig. 6. Disk subsystem performance improvements.

E. Summary and Motivation

Fig. 5 summarizes the mean time of each of the components of a disk's basic service time. The RPS miss time is for a subsystem of eight devices with fifth degree of skew at an I/O rate of 60 per second. It now becomes clear that in today's typical environment, of which Fig. 5 is intended to represent, the combined effect of latency and RPS miss (both of them are dependent on the rotation of the disk) dominates the total service time. Together they represent 56% of the total 18.5 ms, and this percentage increases as the I/O rate goes up. In comparison, seek time and data transfer time represent only 29% and 7%, respectively, of the total. It follows that reducing the latency and RPS miss time has the greatest impact on improving a disk's service time.

This conclusion can be more clearly illustrated when the previously discussed numbers are analyzed with our queueing model so as to include the effects of various queueing delays. The results are as shown in Fig. 6 which plots the response time versus throughput for a subsystem consisting of eight devices with a fifth degree of skew. We first compute the performance curve using today's device characteristics. We then compute the effect of 1) doubling the data rate, 2) reducing the average seek time by half, and 3) reducing by half the latency and the penalty time for 1 RPS miss. It can clearly be seen that the last approach produces the greatest improvement in performance. For example, at a moderate rate of 60 I/O's per second, the storage subsystem average response time is decreased from 36.6 ms to 31.9 ms (−13%) with a doubled data rate, to 27 ms (−26%) with a faster seek, and to 20.5 ms (−44%) with reduced latency and RPS miss penalty. Alternatively, at the given average response time of 25 ms the original subsystem can sustain 39 I/O's per second, while doubling the data rate allows it to sustain 45 (+15%), a faster seek permits 55 (+41%), and reducing the latency/RPS time enables 74 (+90%) I/O's per second.

Some of the difficulties of reducing latency time by increasing a disk drive's rotational speed have already been discussed. In the remainder of this paper, four alternative approaches to reducing the latency and RPS miss time will be considered and their performance implications examined. Three configurations were modeled for each of these approaches, namely, eight devices with fifth degree skew, eight devices with no skew, and 16 devices with fifth degree skew. Although the three configurations generate somewhat different numerical results, qualitatively they are consistently in agreement with each other. Hence, for this paper, only the results for the configuration with eight devices and fifth degree of skew will be presented.

III. Dual Copy Approach

Dual copy, also known as duplex disks and many other different names, is a commonly used method for improving a disk subsystem's reliability and availability [7], [10]. In such a subsystem, data are duplicated, usually on two different drives, so that if one copy is not accessible for any reason a second copy can be used. The costs of this approach are that the subsystem's storage capacity is reduced in half (or twice as many drives are needed to maintain the same capacity) and that every data write must be done twice. The cost of the extra write depends on the read-to-write ratio of the subsystem. For a subsystem with high read-to-write ratio this cost is minimal.

As pointed out in [10], the fact that a piece of data is now available from two different places offers the opportunity of improving performance by judiciously selecting which copy to read. One such approach is to always move the heads of the two duplex disks together to the same corresponding cylinder. It is assumed that both heads will arrive at the target cylinder at the same time. The control unit will then let the RPS hardware decide which disk is rotationally ready first and direct the I/O operation to that drive. In other words, select the drive with the smallest latency. Any additional control unit overhead required to make this selection is assumed to be negligible compared to the other service time components. It can be shown (see Appendix) that the average latency is now reduced to $1/3$ of a revolution (assuming the rotational positions of the two disks are randomly out of phase), instead of the original $1/2$ revolution for a single drive. Furthermore, if the selected drive encounters an RPS miss, the other drive can be used for the next reconnection attempt. The

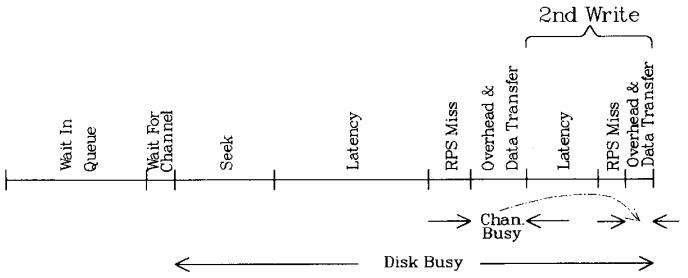


Fig. 7. Time events of a dual copy I/O with serial write.

average RPS penalty time for a miss is therefore reduced to only 1/2 revolution.

There are a number of different ways to handle the duplex write. We will first assume that the second write is to be done serially, immediately after the first write. It can be assumed that the seek for the second write has already been completed. The service time components of this second write are then the latency, RPS miss delay, overhead, and transfer time. The latency of the second write is 1/2 revolution and the penalty for an RPS miss is a full revolution. However, these second write components must be reduced by a factor of $1/(R+1)$ where R is the read-to-write ratio of the subsystem. The time events of an average I/O for our model are now as shown in Fig. 7. The wait for channel delay and the RPS miss delay are now higher than the simplex case since the control unit must handle the extra writes. In particular, the probability of an RPS miss as given in (1) is now adjusted to $\left(\text{now } Q \times \lambda \leq \frac{1}{T}\right)$

$$p_i = \frac{Q \times (\lambda - \lambda_i) \times T}{1 - Q \times \lambda_i \times T} \tag{3}$$

where

$$Q = \frac{(R+2)}{(R+1)}.$$

The CPU will not be notified that the I/O is completed until the second write is finished.

Fig. 8 shows the performance impact of using the dual copy approach as a means to reduce latency and RPS miss penalty for a subsystem of eight devices with fifth degree skew. The results for read-to-write ratios of 2, 4, and 8 are plotted. As can be expected, for a R/W ratio of 2, the dual copy approach performs worse than the original subsystem due to the frequent need to do an extra write. At a R/W ratio of 4, dual copy performs about the same as the original subsystem, being slightly better at low I/O rate and slightly worse at high I/O rate. When the R/W ratio is 8, the dual copy approach becomes clearly superior as the advantage of latency and RPS miss reduction outweighs the cost of having to do extra writes which are now rather infrequent.

The modeling results have shown that a simple smallest-latency-first dual copy approach is not a very effective way to improve a subsystem's performance, especially for low read-to-write ratios. This is because the small reductions in latency and RPS miss penalty are not sufficient to compensate for the penalty of having to do duplex writes. However, the performance of handling duplex writes can be improved if nonvolatile storage (NVS) is added to the control unit. The need of nonvolatility is to guarantee against data loss in case of power failure. With such an arrangement, data for the first write are sent to both the disk drive and the NVS in parallel. As soon as the first write is

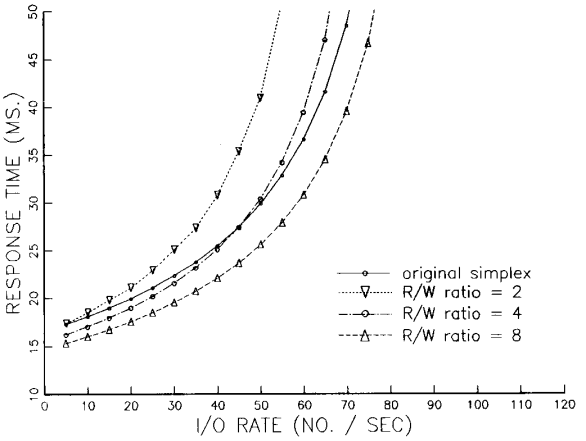


Fig. 8. Dual copy: eight logical devices, fifth degree skew.

completed the CPU is notified of completion of the I/O. At the same time, the second write is initiated internally by the control unit. As before, it can be assumed that the seek for the second write has already been completed. During the latency and RPS miss delay of this second write the control unit can handle other I/O's. However, if another I/O request arrives for this pair of dual copy disks it must be held up (we will call this a *write delay*) until the second write is completed (in order to avoid moving the disk arm). We will call this method of doing the duplex write "*fast second write.*" Thus, the time events of this model are now as shown in Fig. 9. The wait for channel delay and the RPS miss delay are again higher than the simplex case. The mean write delay time for disk i can be expressed as

$$\text{Prob(write delay)} \times \text{average time for a delay.} \tag{4}$$

Let B_i be the basic service time (seek + latency + RPS miss + overhead and data transfer) for disk i and W_i be the mean time to complete the second write. Then the average time of a delay is simply $W_i/2$ (because it varies uniformly between 0 and W_i), and

$$\text{Prob(write delay)} = \frac{1}{R+1} \times \frac{\lambda_i \times W_i}{1 - \lambda_i \times B_i} \tag{5}$$

Fig. 10 shows the performance improvement of using fast second write for a system of eight devices and fifth degree skew. The results for R/W ratios of 2, 4, and 8 are plotted. As can be expected, the response time of disk duplexing is independent of the R/W ratio at very low I/O rate. This is because at low I/O rate

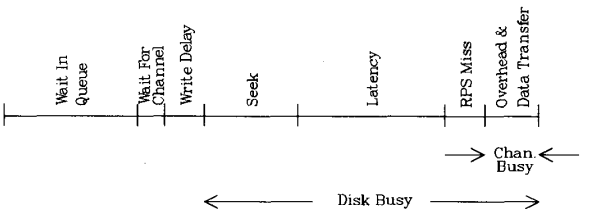


Fig. 9. Time events of a dual copy I/O with fast second write.

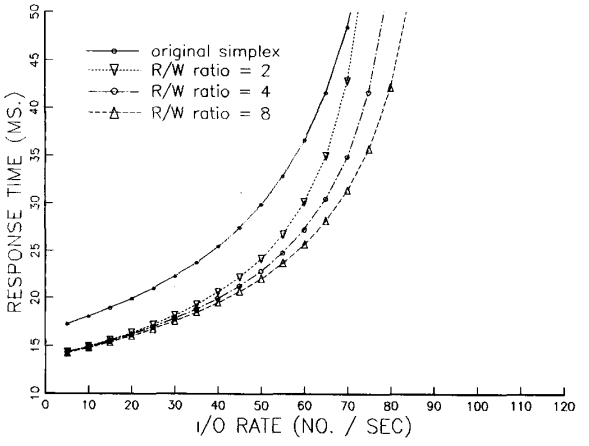


Fig. 10. Dual copy with fast second write: eight logical devices, fifth degree skew.

the chance of a write delay is very small and therefore the second write can be done in the background without interfering with any other I/O. As the I/O rate increases, the performance for small R/W ratio deteriorates faster than that for large R/W ratio. This is due to the fact the write delays caused by the second write is gradually becoming a factor. Nonetheless, the performance of the fast second write approach seems to be superior to the original simplex subsystem regardless of the R/W ratio.

IV. Synchronized Dual Copy Approach

This second approach is an enhancement of the first approach. Let us assume that we have a means of synchronizing the rotations of two disk drives. We will then synchronize the two drives of a pair of dual copy disks so that they are 180 degrees out-of-phase. In other words, when the index mark of one drive is under the head, the index mark of the other drive is half a revolution away from the other drive's head. The advantage of such an arrangement is obvious. Instead of an average latency of 1/3 revolution for the rotationally closest drive in the nonsynchronized case, the average latency is now 1/4 revolution. The penalty for one RPS miss is exactly 1/2 revolution.

The performance of a synchronized dual copy approach is as shown in Fig. 11 for a subsystem of eight devices and fifth degree skew. A simple serial second write is assumed. As is expected, the performance shows an improvement over the nonsynchronized approach. For a R/W ratio of 2, the synchronized dual copy approach is a little better than the original simplex subsystem at low I/O rate and becomes worse as the I/O rate increases.

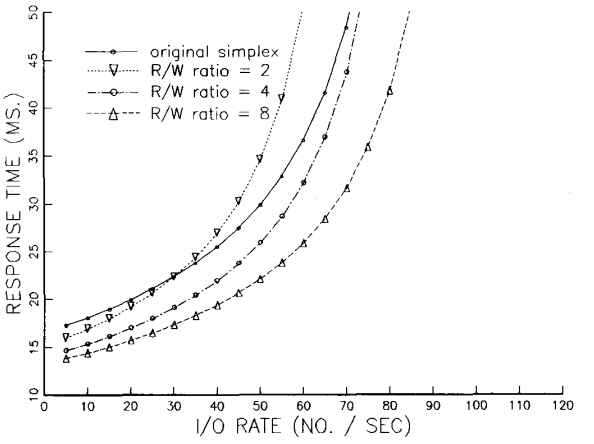


Fig. 11. Synchronized dual copy: eight logical devices, fifth degree skew.

However, for R/W ratios of 4 or greater, the synchronized approach outperforms the original at all I/O rates of interest.

If NVS and the fast second write technique (as described in the previous section) are used in conjunction with the synchronized dual copy approach, then the performance far exceeds that of the original subsystem for any R/W ratio. This is as shown in Fig. 12.

V. Dual Copy On A Single Disk

This approach is a variation of the synchronized dual copy approach discussed in the last section. Instead of using two rotationally synchronized devices, one for each set of the dual copied data, this approach stores both copies of the data on the same device. Each piece of data is arranged such that it and its dual copy are 180 degrees out-of-phase with respect to each other. In other words, if a piece of data is currently under the head, its second copy is half a revolution away from the head. There are a number of ways with which this can be implemented. One way is to pair up the heads (assuming there are an even number of them) in the actuator of the device. Data on the resulting odd numbered tracks are then duplicated on the even numbered tracks on the same cylinder, with the index marks (beginning of track) of the odd tracks offset by 180 degrees from the index marks of the even tracks. With this method the number of logical tracks in a cylinder is reduced in half. Another method of implementation is to divide every track into two halves. Data on the first of each track are then repeated on the second half of the same track. With this approach, the number of logical tracks in a cylinder remains unchanged; however, the capacity of each track is now reduced by a factor of two.

The latency and RPS miss penalty of this approach are identical to those of the synchronized dual copy approach, viz., 1/4 and 1/2 revolution, respectively. However, from a subsystem point of view, these two approaches behave differently. In the synchronized dual copy approach, two physical devices are combined to create one new logical device with the same total capacity as that of one original physical device. With the dual copy on a single disk approach, one physical device becomes one new logical device. However, because the new logical device has only half the capacity of that of the original physical device, two new logical devices are needed to maintain the original capacity. Thus, while both the synchronized dual copy approach and the

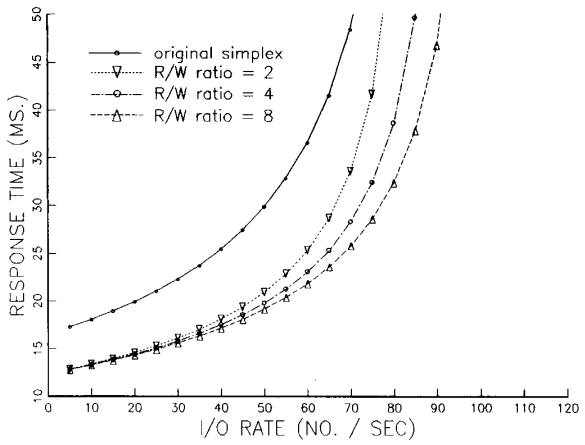


Fig. 12. Synchronized dual copy with fast second write: eight logical devices, fifth degree skew.

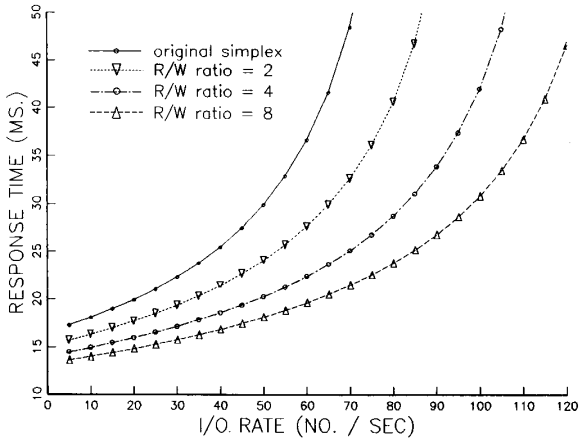


Fig. 13. Dual copy on a single disk with fast second write: 16 half-size logical devices, fifth degree skew.

dual copy on a single disk approach require the same number of physical devices to provide the same amount of capacity, the second approach presents itself as having twice as many logical devices. This should have a favorable impact on the subsystem's performance. The amount of improvement depends on how data are split from one device into two devices. Assuming the degree of skew remains the same after the split, Fig. 13 shows the performance of a subsystem using the dual copy on a single disk approach with an equivalent total capacity of eight original devices and with a fifth degree skew. It can be seen that the performance is better than that of the synchronized dual copy approach shown in Fig. 11. This is because now there are twice as many servers in the subsystem as before and, therefore, the time an I/O spent in a queue waiting is reduced.

Once again, if NVS and the fast second write technique (as described in the Section III) are used in conjunction with the dual copy on a single disk approach, then the performance is even better and far exceeds that of the original subsystem for any R/W ratio. This is as shown in Fig. 14.

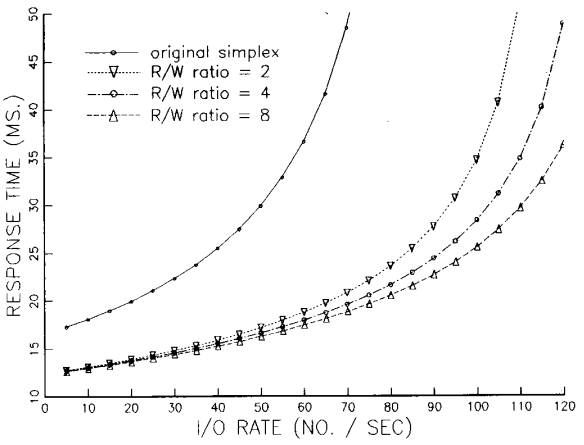


Fig. 14. Dual copy on a single disk with fast second write: 16 half-size logical devices, fifth degree skew.

VI. Dual Actuator Approach

The main disadvantage of the three previous approaches is that the raw storage capacity of a subsystem has to be doubled. In the fourth approach, to be described here, only the same amount of storage as that of a simplex subsystem is required. This approach calls for adding an extra actuator with an extra set of heads for each device, placed diagonally opposite to the original set of actuator and heads. This concept was mentioned in [14]. Each piece of data is now accessible by two different heads located 180 degrees apart. On a seek, both actuators are moved to the same cylinder. The RPS hardware then decides which head is ready for data transfer first. As is the case for the synchronized dual copy, the average latency is reduced to $1/4$ revolution and the penalty of an RPS miss is exactly $1/2$ revolution. However, unlike the dual copy approaches, there is no need to do any duplex write and hence there is no performance penalty. The performance of the dual actuator approach is shown in Fig. 15 for a subsystem of eight devices with fifth degree skew. This curve is the same as the *latency–RPS halfed* curve of Fig. 6. It is reasonable to raise the question about whether the added actuator might be put to better use by splitting the platters of a device into two sets, using the added actuator for half of the platters and the original actuator for the other half. In other words, subdivide each device into two. This new configuration of subsystem would store the same data, support the same I/O rate, and will be assumed to have the same device skew as before. The intuitive advantage of such a configuration is that the number of servers is doubled and, therefore, the time an I/O spent in a queue is reduced. The performance curve of such a subsystem is also shown in Fig. 15. At very high I/O rate the performance of such a configuration deteriorates more slowly than that of the dual actuator approach. However, at the I/O rates of practical interest the dual actuator approach is clearly superior.

VII. Summary and Recommendations

In this paper, we have made the observation that rotational latency and RPS miss delay together account for the majority of a disk drive's service time in many of today's high-end computing environments. Both of these two components are related to the rotation of the storage device. Efforts directed at reducing these

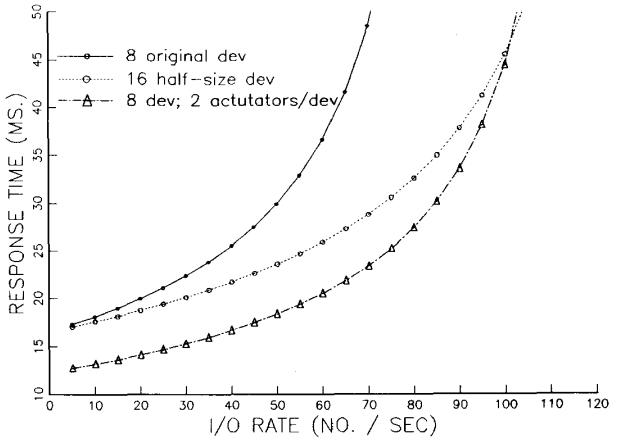


Fig. 15. Two actuators per disk: eight devices, fifth degree skew.

components should have the greatest impact on improving a disk storage's response time.

While the RPM of future generations of disk drives undoubtedly will be higher, there are certain practical limitations. In this paper, we have described several different alternatives for reducing the latency and RPS miss penalty time that do not require spinning the disk faster. The performance behaviors of these approaches are summarized in Fig. 16 for a subsystem with a fifth degree skew and with a total capacity equivalent to that of eight devices. For the various dual copy approaches the results shown are for a R/W ratio of 4. The main objective of these dual copy approaches is to improve the average I/O response time of a subsystem. Reliability and availability issues were not part of the consideration. For this reason some of the dual copy approaches discussed here are different from the conventional high availability dual copy approaches [10]. The preferred approach will be selected solely on a low response time basis.

As can be seen in Fig. 16, the dual actuator approach gives one of the best performance improvements and could have the smallest incremental cost among the different alternatives considered. However, for a multiple-arms-per-actuator disk drive, aligning the two set of heads such that all the corresponding heads are exactly over the same track may be difficult to achieve. Furthermore, in magnetic recording, aligning two heads so that they have the same skew with respect to a track is even more of a major problem. Therefore, despite its attractive performance, such a technique may not yet be practical for magnetic storage. However, this approach is a viable one for improving the performance of single-head-per-actuator optical storage devices. In fact, for those writable optical disks that require an erasure before any write, such an arrangement also reduces latency by choosing the smallest latency head to do the erasure and the other head to do the write.

While at low I/O rates the performance of the dual copy on a single disk approach is inferior to that of the dual actuator approach, it is still substantially improved over that of the original device (e.g., at 40 I/O's per second, the improvement is 30%). At high I/O rates, these two approaches perform about equally. Furthermore, if NVS and the fast second write technique are used in conjunction with the dual copy on a single disk approach the performance becomes even better than that of the dual

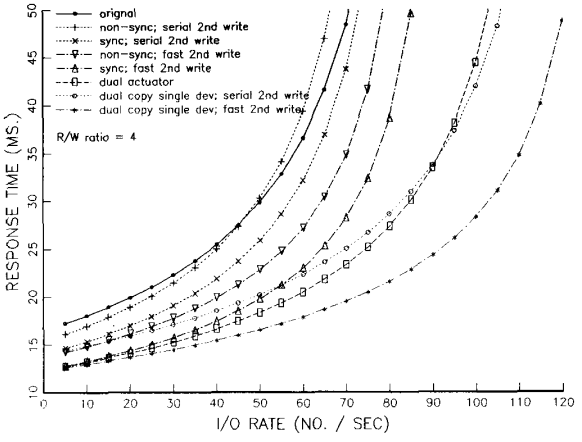


Fig. 16. Performance improvement summary: capacity = eight devices, fifth degree skew.

actuator approach. Thus, for magnetic storage, the dual copy on a single disk approach, with or without fast second write, is recommended as a means of improving the disk subsystem's overall performance. In general, the cost of this approach is the need of additional storage space due to data duplexing. However, there exist today performance conscious installations that are currently not using the full storage capacity of their busiest disk drives in order to reduce the amount of data under each actuator (i.e., have more actuators per gigabyte). For those installations, they have already paid for the extra raw capacity needed in this approach, and this approach should be particularly attractive to them. Even in the low-end such as the PC's where there is only a single disk drive in the storage subsystem, there need not be extra cost to implement the dual copy on a single disk approach, because the single disk's storage may not always be fully used, and the recommended technique can be selectively applied only to a subset of the most frequently used data. In fact, in some systems the most critical data such as volume table of contents or file allocation tables are already replicated on the disk for reliability reason. Why not replicate them in an intelligent manner so that performance can also be enhanced?

Finally, for all the approaches discussed in this paper, a common side benefit is that if an uncorrectable error occurs during a read, the subsystem needs only to wait for 1/2 revolution before retrying the read again from the other disk/head, rather than one full revolution in today's subsystem. Thus, error recovery is also faster with the suggested approaches.

Appendix

Let X_1 and X_2 be two i.i.d. random variables each uniformly distributed between 0 and R. Let X be the $\min[X_1, X_2]$. Then the distribution function for X is

$$\begin{aligned} F_X(x) &= \text{Prob}(X < x) \\ &= \text{Prob}(X_1 < x \text{ or } X_2 < x) \\ &= 1 - \text{Prob}(X_1 \geq x \text{ and } X_2 \geq x) \\ &= 1 - \text{Prob}(X_1 \geq x) \times \text{Prob}(X_2 \geq x) \\ &= 1 - \left(1 - \frac{x}{R}\right) \times \left(1 - \frac{x}{R}\right) \\ &= \frac{2x}{R} - \frac{x^2}{R^2}. \end{aligned}$$

The density function of X is, therefore,

$$f_X(x) = F'_X(x) = \frac{2}{R} - \frac{2x}{R^2}.$$

The expected value of the $\min[X_1, X_2]$ can now be determined to be

$$E[X] = \int_0^R x f_X(x)\, dx = \frac{R}{3}.$$

Thus, if R is the time for one revolution of a disk drive and X_1 and X_2 are the latencies of two independent drives, the average latency of the first drive to be rotationally ready is 1/3 of a revolution.

References

[1] A. O. Allen, *Probability, Statistics, and Queuing Theory.* New York: Academic, 1978.

[2] Y. Bard, "A model of shared DASD and multipathing," *Commun. ACM,* vol. 23, no. 10, pp. 564–572, Oct. 1980.

[3] M. Bohl, "Introduction to IBM direct access storage devices," Science Research Associates, Inc. 1981.

[4] P. Denning, "Effects of scheduling on file memory operations," in *Proc. AFIPS 1967 Spring Joint Comput. Conf.,* vol. 30, 1967, pp. 9–21.

[5] J. M. Harker, D. W. Brede, R. E. Pattison, G. R. Santana, and L. G. Taft, "A quarter century of disk file innovation," *IBM J. Res. Develop.,* vol. 25, no. 5, pp. 677–689, Sept. 1981.

[6] D. Hunter, "Modeling real DASD configurations," IBM Res. Rep. RC 8606, Dec. 1980.

[7] *IBM 3990 Storage Control Introduction,* IBM Publication GA32-0098.

[8] M. Y. Kim, "Synchronized disk interleaving," *IEEE Trans. Comput.,* vol. C-35, no. 11, pp. 978–988, Nov. 1986.

[9] W. C. Lynch, "Do disk arms move?" *Perform. Eval. Rev.,* vol. 1, pp. 3–16, Dec. 1972.

[10] S. W. Ng, "Reliability, availability and performance analysis of duplex systems," in *Proc. IASTED Int. Symp. Reliability and Quality Contr.,* Paris, France, June 24–26, 1987, pp. 5–9.

[11] S. W. Ng, D. Lang, and R. D. Sellinger, "Trade-offs between devices and paths in achieving disk interleaving," in *Proc. 15th Annu. Int. Symp. Comput. Architecture,* Honolulu, HI, May 30–June 3, 1988.

[12] W. C. Oney, "Queueing analysis of the scan policy for moving head disks," *J. ACM,* vol. 22, no. 3, pp. 397–412, July 1975.

[13] E. W. Pugh, D. L. Critchlow, R. A. Henle, and L. A. Russell, "Solid state memory development in IBM," *IBM J. Res. Develop.,* vol. 25, no. 5, pp. 585–602, Sept. 1981.

[14] R. A. Scranton, D. A. Thompson, and D. W. Hunter, "The access time myth," IBM Res. Rep. RC 10197, Sept. 1983.

[15] L. D. Stevens, "The evolution of magnetic storage," *IBM J. Res. Develop.,* vol. 25, no. 5, pp. 663–675, Sept. 1981.

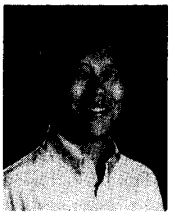
[16] T. Teorey and T. B. Pinkerton, "A comparative analysis of disk scheduling policies," *Commun. ACM,* vol. 15, no. 3, pp. 177–184, Mar. 1972.

[17] J. Voelcker, "Wichester disks reach for a gigabyte," *IEEE Spectrum,* vol. 24, no. 2, pp. 64–67, Feb. 1987.

[18] N. C. Wilhelm, "An anomaly in disk scheduling: A comparison of FCFS and SSTF seek scheduling using an empirical model for disk accesses," *Commun. ACM,* vol. 19, no. 1, pp. 13–17, Jan. 1976.

[19] ——, "A general model for the performance of disk systems," *J. ACM,* vol. 24, no. 1, pp. 14–31, Jan. 1977.

[20] J. Zahorjan, J. N. P. Hume, and K. C. Sevcik, "A queueing model of a rotational position sensing disk system," *INFOR,* vol. 16, no. 3, pp. 199–216, June 1978.



Spencer W. Ng (S'69–M'73) received the B.S. degree in electrical engineering from Washington State University, Pullman, in 1968, and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana, in 1970 and 1973, respectively.

From September 1973 to October 1983 he was a member of the Technical Staff with AT&T Bell Labs, Naperville, IL. Since October 1983 he has been a research staff member with the IBM Almaden Research Center, San Jose, CA.