A Simulation Study of Shadowed Disks

Athena Vakali and Theocharis Malamatos

Department of Informatics
Aristotle University
54006 Thessaloniki, Greece
email: avakali@csd.auth.gr

Abstract. Multiple disks have been introduced for use in I/O subsystems to increase data availability and fault tolerance. Identical disks (comprising the shadowed disk) sets have been adopted as an effective secondary storage replication scheme. In the present paper, we introduce a shadowed disk model and mathematically analyze its theoretical expected performance measures by applying the *nearer-server* rule to satisfy the requests. In addition, we introduce the corresponding simulation model, where each disk is modeled by means of a prototype disk configuration and specific commercial disks metrics. Furthermore, the mirrored disks model is studied extensively as a boundary disk topology of the shadowed disks set. The developed simulation model is experimented and tested under various workloads produced by real disk I/O traces. The performance of the simulation model is compared to the analytical results and, thus, it is validated since analysis and simulation give close results.

1 Introduction

Storage subsystems are vital components of modern computer systems. According to [6], the amount of storage sold has been almost doubling each year, whereas magnetic disks are the most dominant and most widely used secondary storage devices. Typical accesses to disks are much slower than accesses to the main memory of a computer system. This fact has created the so called *access gap problem*, which has attracted a lot of attention to devise methods to overcome or minimize the difference between processor speed and disk servicing time.

Redundant inexpensive/independent disk arrays (RAIDs) have been proposed to increase reliability and improve system performance [3]. In such systems, an immediate backup service is supported, while data are accessible whenever at least one disk is available. Considerable interest has been shown in redundancy schemes based on the RAID level-1, i.e. in the mirroring disk topology. As an extension to the mirrored disk configuration, a more generalized scheme of shadowed parallel disks has been proposed, where a number of identical disks is considered. Under these configurations, each disk may be viewed as an exact copy of the others, i.e. all disks store exactly the same data [1, 2, 8, 15].

Storage system simulators have been developed to verify the performance modeling of various disk configurations. The *Pantheon* storage simulator of Hewlett-Packard Laboratories is an effective tool for performance modeling of parallel disk arrays as well as of a wide range of I/O subsystems for both uniprocessors and parallel computers [16]. The *DiskSim* storage simulator was developed at the University of Michigan and supports most secondary storage components including disk and device drivers, buses, controllers and adapters [4, 5]. A detailed simulation model of the HP 97560 disk has also been developed supporting one or more disks attached to one or more SCSI buses [7]. Data reorganization on the disk surface could improve the service process of a request by reducing the head traveling distance. In particular, a disk shuffling simulator has been developed to implement a shuffling technique to move frequently accessed data toward the disk center [10].

This paper presents a storage simulator for a shadowed disk subsystem with emphasis on the mirrored disk topology. The developed simulator supplements and validates theoretical statements and expected performance measures from previous research efforts. The design and implementation of the simulation model aims to the effective I/O servicing by exploiting redundant data placed on multiple disks. The remainder of the paper is organized as follows. The next section describes the shadowed disk subsystem topology, analyzes the service rules and states the performance metrics. Sections 3 and 4 present the implemented simulation model and the simulation library, respectively. Section 5 presents the experiments and the validation tests, whereas Section 6 points some conclusions and discusses potential future work.

2 The model and its performance analysis

2.1 The shadowed disk subsystem

The shadowed disk subsystem comprises of k disks, all being identical copies of one another. Figure 1 depicts the shadowed disk topology assumed for the present simulation model. Each disk is equipped with read/write heads, whereas at any point in time a specific head resides on top of a cylinder c_i in each of the i=1,2,...,k disks. Since disks represent identical images and must remain consistent at all times, a request is served differently depending on its type (i.e. read or write). A read operation is satisfied by accessing any disk, since they all store exactly the same data. The choice of the disk to be accessed is made by applying the *nearer-server rule*, i.e. the disk on which the appropriate head is closest to the requested cylinder is chosen. A write operation must be satisfied by all disks, since they all have to be identical copies. Thus, it is obvious that shadowed disk subsystems support a type of an immediate backup. In addition, they could serve many requests in parallel by using specific controllers.

Given the arrangement of data surfaces and heads, the time required for a particular disk operation involves mainly the following actions [9]: *move* the appropriate head to the appropriate cylinder (*seek time*), *wait* for the required sector to rotate around to the location of the activated head (*rotational latency*), and *read* the required sector from the disk surface (*block transfer time*). Additionally, the overall service time involves the controller overhead, the external transfer time, the head-switch time, the track-switch time and other factors described thoroughly in [12]. These metrics are quite small and they are not measured in previous research efforts. Therefore, the overall service time is evaluated by the expression [13, 14]:

$$T_{Service} = T_{Seek} + T_{Rotation} + T_{Transfer}$$

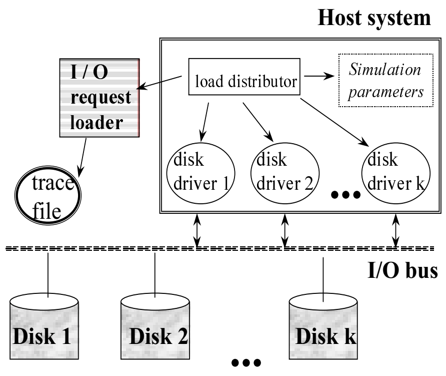

Fig. 1. The shadowed disk topology.

where the service time of a request by the disk mechanism ($T_{Service}$) is a function of the seek time (T_{Seek}), the rotational latency ($T_{Rotation}$) and the block transfer time ($T_{Transfer}$) for the specific request size. Seeking involves successively the operations of speedup, coast, slowdown and settle to reach the requested location. Therefore, seeking has been considered as the most dominant performance factor and seek distance has been accepted as an indicative performance metric.

Analytic models have been developed to study the performance behavior of seeking in shadowed disks. These models are studied by adopting certain simplifications. For instance, it is assumed that the requests follow a uniform distribution with respect to the cylinder positions, as well as that successive seeks are independent. Thus, the expected seek distance has been evaluated by [1] :

$$\mathrm{E}[\text{seek\_distance}] \;=\; r\ \mathrm{E}[\text{read}] \;+\; w\ \mathrm{E}[\text{write}] \tag{1}$$

where r is the read rate, w is the write rate (evidently, r+w=1 holds), and

$$\mathrm{E}[\text{read}] \;=\; \frac{C}{2k+1} \qquad\qquad \mathrm{E}[\text{write}] \;=\; C\ (1 - I_k)$$

where C is the total number of cylinders and

$$I_k \;=\; \begin{cases} \frac{2k}{2k+1} I_{k-1} & \text{if } \;k > 1 \\ \frac{2}{3} & \text{if } \;k = 1 \end{cases}$$

On the other hand, assuming a model of dependent seeks the expected seek distance has been evaluated by the same Equation (1) and by considering a Markov chain model resulting in the following partial expected seeks [8, 15]:

$$\mathrm{E}[\text{read}] \;=\; \sum_{i=1}^{k} \pi_i\ \frac{C}{2i+1} \qquad\qquad \mathrm{E}[\text{write}] \;=\; \sum_{i=1}^{k} \pi_i\ C\ (1 - I_i)$$

with π_i be the long run proportion of the time the process spends in a certain state.

The next table gives all the parameter symbols of the present study. The case of k=2 corresponds to the mirrored disk case, otherwise defined as the RAID level-1 configuration. The mirrored disk storage system has been studied in detail, since it is the most common topology adopted in various system configurations.

Symbol	Definition
$T_{Service}$	disk service time
T_{Seek}	seek time
$T_{Rotation}$	rotational latency
$T_{Transfer}$	transfer time for a specific request size
k	number of disks
C	number of cylinders per disk
r	read ratio
w	write ratio
E[seek_distance]	expected total seek distance
E[read]	expected seek distance for reads
E[write]	expected seek distance for writes
E[D_i]	expected total seek distance for a particular disk i=1..2

Table 1. Parameter symbols.

2.2 The nearer-server rule

Assume a request arrival for a cylinder x, when heads lie on top of cylinder c_i in each of the i=1,...,k disks and suppose that D_i is the distance of the head in disk i from the requested location, i.e. $D_i = |c_i - x|$

Definition: According to the nearer-server rule, the read request will be served by disk m traveling distance D_m with $D_m = min\{D_1, D_2, ..., D_k\}$, whereas the write request must be satisfied by all disks resulting in traveling distance D_M with $D_M = max\{D_1, D_2, ..., D_k\}$.

Proposition: The nearer server rule is optimal for the shadowed disk storage system.

Proof: Consider the special case of k=2 disks of a mirrored disk set. Suppose that D_1, D_2 are random variables for the seek distances involved in the corresponding two disks, respectively. The mean seek distance is:

$$\text{E[seek\_distance]} \;=\; \text{E}[D_1] \;+\; \text{E}[D_2] \qquad (2)$$

We define $p_x(i,j)$ the probability to request cylinder x, when the head of Disk-1 lies on top of the i-th cylinder, whereas the head of Disk-2 is on top of the j-th cylinder. For each of the two disks it holds that:

$$D_1 = \begin{cases} 0 & \text{when Disk 2 serves} \\ |i-x| & \text{when Disk 1 serves} \end{cases}$$
$$D_2 = \begin{cases} 0 & \text{when Disk 1 serves} \\ |j-x| & \text{when Disk 2 serves} \end{cases}$$

We introduce a binary function $f_x(i,j)$ to distinct the service of request referring to x by:

$$f_x(i,j) = \begin{cases} 1 & \text{if Disk 1 serves } x \\ 0 & \text{if Disk 2 serves } x \end{cases}$$

Therefore, the definition of the distances D_1, D_2 becomes:

$$D_1 \;=\; f_x(i,j) \cdot |i-x| \qquad\qquad D_2 \;=\; (1 - f_x(i,j)) \cdot |j-x|$$

The overall expected distance D_1 will be:

$$\mathrm{E}[D_1] = \sum_{i=1}^{C}\sum_{j=1}^{C}\sum_{x=1}^{C} p_x(i,j) \cdot f_x(i,j) \cdot |i-x| \tag{3}$$

whereas D_2 will be:

$$\mathrm{E}[D_2] = \sum_{i=1}^{C}\sum_{j=1}^{C}\sum_{x=1}^{C} p_x(i,j) \cdot (1 - f_x(i,j)) \cdot |j-x| \tag{4}$$

The application of Equations (3) and (4) in the overall expected seek distance of Equation (2) results in:

$$\begin{aligned} \mathrm{E}[\text{seek\_distance}] &= \sum_{i=1}^{C}\sum_{j=1}^{C}\sum_{x=1}^{C} p_x(i,j) \cdot f_x(i,j)|i-x| \\ &+ \quad \sum_{i=1}^{C}\sum_{j=1}^{C}\sum_{x=1}^{C} p_x(i,j) \cdot ((1 - f_x(i,j)) \cdot |j-x|) \\ &= \sum_{i=1}^{C}\sum_{j=1}^{C}\sum_{x=1}^{C} p_x(i,j) \cdot |j-x| \\ &+ \quad \sum_{i=1}^{C}\sum_{j=1}^{C}\sum_{x=1}^{C} p_x(i,j) \cdot f_x(i,j) \cdot (|i-x| - |j-x|) \end{aligned}$$

It is obvious that E[seek_distance] will be minimized if the second part in the above expression is minimum, since the first part of the summation is always positive and independent of any service rule. Quantity $dif = |i-x| - |j-x|$ in the second part is positive only when the disk with its head on cylinder i serves the request. Minimizing $p_x(i,j) \cdot f_x(i,j) \cdot (|i-x| - |j-x|)$ will be achieved if $f_x(i,j) = 0$ for all positive dif and $f_x(i,j) = 1$ for all negative dif, i.e. minimization occurs in the case of the nearer-server rule.

The extension of this reasoning to a shadowed disk system with $k > 2$ disks is quite straightforward by using induction. Supposing that the proposition is true for $k{=}n{-}1$ disks, then it is obvious that it will still hold for $k{=}n$ disks since:

$$\mathrm{E}[\mathrm{seek\_distance}] \;=\; \mathrm{E}[D_1] + \ldots + \mathrm{E}[D_{n-1}] + \mathrm{E}[D_n]$$

3 The Simulation model structure

The simulation model developed applies the nearer-server rule in a shadowed disk subsystem and it introduces four main components for each disk :

1. *the Controller:*
   The disk controller includes a microprocessor, a buffer and a bus connection. Its job is to receive, process and control all the requests to the disk and guide the head to the appropriate track. In the simulation model, the controller is implemented by an algorithm resembling the actual disk controller algorithm.
2. *the Disk Mechanism (DM):*
   The disk mechanism models the process of data transfer between the cache memory and the storage medium. It is represented by a thread and it is independent of the inner data transfer.
3. *the Direct Memory Access Engine (DMA Engine):*
   The Direct Memory Access mechanism is governed by the disk controller to allow data transfer between main memory and disk with no CPU involvement. The CPU is notified by the controller only after data transfer completion. The DMA is supported by appropriate registers and connection protocols. It is represented by a thread and it is independent of the external data transfer.
4. *the Cache memory:*
   The cache memory is an intermediate RAM type memory for the temporary storage of data moving from/to disk to/from main memory. In case of a write operation, data are temporarily stored in cache with a higher speed than the disk storage, whereas disks get the data at their own rate later. In case of a read operation, it is quite helpful to get data from cache when the bus is busy.

The implementation includes other supportive components such as a request queue, a bus connection mechanism, a connection with the specific disk driver, disk parameter specifications as well as tools for the proper disk functioning. Requests are served by the FCFS scheduling policy, i.e. according to their arrival time.

Every request consists of basic information about its type, its initial sector and the number of consecutive sectors requested. There are three types of requests: read, write and write immediate-reported. Disk status is either idle (free), or serve (one of the 3 request types). The events affecting the disk status are: arrival of a new request, notification of either DMA or DM job completion, which are mutually excluded.

All possible controller states depending on the current status and the upcoming event are presented in Table 2. If disk is busy, the request is added to the queue, otherwise the controller acts according to the request type :

Disk Status	Event		
	new request arrival	DMA job completion	DM job completion
free	√	X	X
read	√	√	√
normal write	√	√	√
write IR	√	√	√

Table 2. Controller possible states (X = does not occur).

- *normal write.* The controller asks for the cache to be empty and to halt any prefetching action. Then the DMA engine is asked to transfer the number of requested sectors whereas the disk is asked to write an equal number of sectors starting at the logical sector included in the request pattern.
- *write IR.* The same actions (as in normal write) are carried out here, but the last sector to be written on the disk is noted additionaly.
- *read request.* The controller asks for the number of sectors in the cache, whereas the DMA engine is asked to transfer the requested sectors. At the same time, the disk is asked to transfer the requested sectors not included in the cache.

Routine	Description
`notifyRequestDone()`	notification that the driver's last sent request is completed
`acquire()`	a disk driver asks for the bus
`release()`	the bus is released
`TransferBytes(sectorSize)`	time involved at the transfer of the data identified by `sectorSize`
`waitUntil...`	various methods for the waiting time for the action specified after the `waitUntil..` keywords.
`setSector`	set / place sectors from cache
`getSector`	get / retrieve sectors from cache

Table 3. The basic routines / methods.

The main routines and methods used in the DMA engine and Disk models are described in Table 3. The simulation model uses a DMA engine algorithm to implement proper DMA functioning.

The Disk Mechanism algorithm of the simulation model considers sector transferring as the head moves forward and supposes that there are no delays when advancing from a sector to its next logical sector (i.e. 1:1 interleaving).

A pseudo-code version of the DiskMechanism-Work-Loop algorithm follows:

```
//DiskMechanism-Work-Loop algorithm //
while (true) do
  Controller.getWork()
  if (work=read) then     // Prefetching enabled in case of a read
    doPrefetch <- true
    for each sector to be transferred do
      if (sector=first sector)
         time <- StartSectorComputeTime(sector,now)
      else
         time <- NextSectorComputeTime(sector,now)
         Delay(time)            // delay until head is on the right sector
         while not Cache.canSetSector()
         Delay(oneRotation); Delay(timeToReadSector);
         Cache.setSector()
         Controller.notifyWorkDone()      // work done
         // prefetching stage
         while (doPrefetch and Cache.canSetSector())
            time=NextSectorComputeTime(sector,now)
            Delay(time)                   // simulating delay
            while not Cache.canSetSector()
            Delay( oneRotation )
            Delay(timeToReadSector); Cache.setSector()
  else                                   // we have a write request
     for each sector to be transferred do
        time=NextSectorComputeTime(sector,now); Delay(time)
        while not Cache.canGetSector()
          Delay(oneRotation)
          Delay(timeToWriteSector); Cache.getSector()
          if (work=normal write)
             Bus.claim(); Bus.transferBytes(doneMessageSize)
             Driver.notifyRequestDone(); Bus.release();
             Controller.notifyWorkDone()
```

The Caching of the simulation model is performed by sector data transfer, whereas it stores an area of consequent sectors at a time. Cache memory is common for reads and writes and either upon completion of a read request or upon start of a write request (except write IR) it gets empty. If the cache is full, then loading new sectors is aborted until all its sectors are requested as in [7].

4 The Simulation library

The model was simulated by using discrete event simulation. The Simulation Library was developed in C++ under Windows NT 4.00 and included:

- *simulation threads* of the various simulated processes. Threads are treated as in Operating Systems synchronization problems (e.g. two threads for the consumer-producer type of processes).
- *synchronization objects* used for the proper thread synchronization. For example, a synchronization object is needed in cache management.

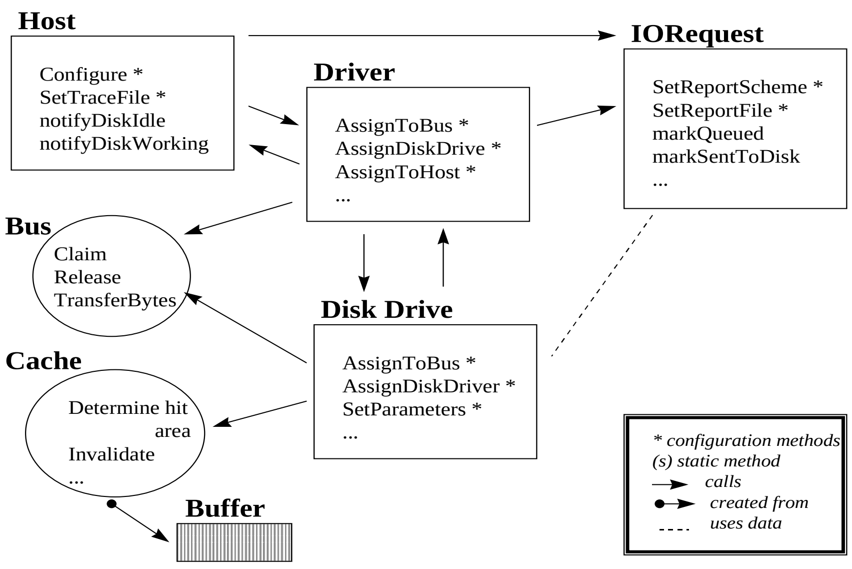

Fig. 2. The main classes and methods of the Simulation Library.

– *several services*, such as the Delay time supporting tool, methods for the simulation start and completion, information about current system status etc.

Figure 2 depicts the main classes and methods of the Simulation Library.

We have defined an object *SIM* to guide and control the simulation process. The main methods for the SIM object are as follows :

– `void SIM.RegisterThread(StartRoutine, PArguments arglist)` for the simulation thread resistration. Its arguments are the name of the routine to start the execution (StartRoutine) and the pointer (PArguments) to pass all parameters for the thread initialization.
– `void SIM.Activate()` for the simulation initialization.
– `void SIM.Terminate()` for the end of the simulation. All threads will be destroyed and return at the point after SIM.Activate().
– `void SIM.Delay(double time)` for delay of the current thread from its execution until time goes on for `time` units of time.
– `double SIM.GetTime()` returns the simulation time at the time of its call.

5 Experiments - Results

As mentioned in previous, our simulation model was examined by using HP-97560 disk I/O traces provided by Hewlett-Packard Laboratories and described in [11, 12]. Each trace record includes timings of enqueue time, disk number, partition and device

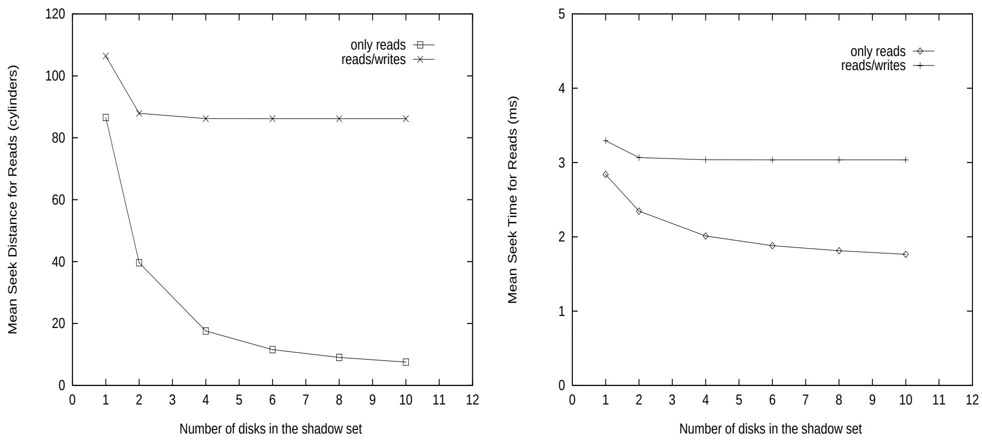

Fig. 3. Mean seek distance and time for reads.

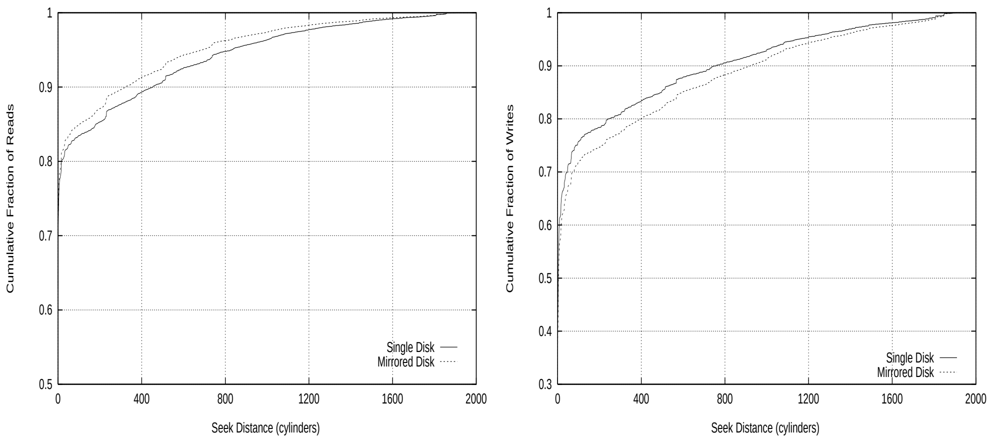

Fig. 4. cdf seek distance for reads and writes.

driver type, start location, transfer size as well as other information details for each I/O request. Results presented in this section include the most important performance metrics such as seek time and distance, access and service times as well as cumulative distribution functions.

5.1 Shadowed disk subsystem

For our shadowed disk subsystem model, we assume that each disk is a HP-97560 device of Hewlett-Packard. This particular disk has most of the basic characteristics needed for our simulation model, real traces of its I/O traffic for a certain period are available from Hewlett-Packard Laboratories and has been studied in the past [11, 12]. The main characteristics of this model are described in Table 4.

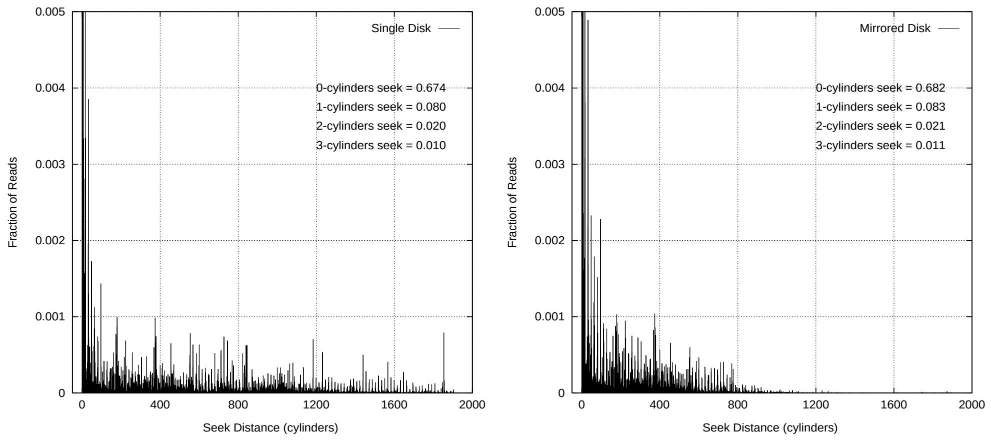

Fig. 5. pdf seek distance, reads only.

Parameter	Value	Comments
Sector size	512 bytes	
Cylinders (C)	1962	1935 for data
tracks per cylinder	19	
sectors per track	72	
revolution speed	4002 rpm	
controller interface	SCSI-II	
controller reads overhead writes	2.2 msec 2.2 msec	
seek time (ms)	$3.24{+}0.400\sqrt{d},\ d < 383$ $8.00{+}0.008d,\ d \geq 383$	d is distance in cylinders
disk buffer, cache size	128	

Table 4. HP-97560 model characteristics and parameters.

The simulation model was tested for different number of disks in the shadowed disk subsystem. More specifically, shadowed disk sets of k=2,4,6,8,10 disks were simulated for traces in the time period between 1992.04.28 to 1992.05.01. There were 352,826 requests processed, where a portion of 46.24% concerned read and the remaining 53.76% concerned write operations.

First, we tested our simulation model with the above period traces after isolating the read requests only. Figure 3 depicts the mean seek distance and time, when we consider either only reads or reads/writes. We notice that in case of considering only reads we have a benefit of a 54.5% improvement in seek distances, between the single disk and the mirrored disk model (Figure 3). This reduces to a 17.2% improvement with respect to the corresponding seek times of these two models (Figure 3). This variation in the reductions of the seek distance and time is explained by the non-linear relationship of seek time/distance of the HP-97560 model (see expression given in Table

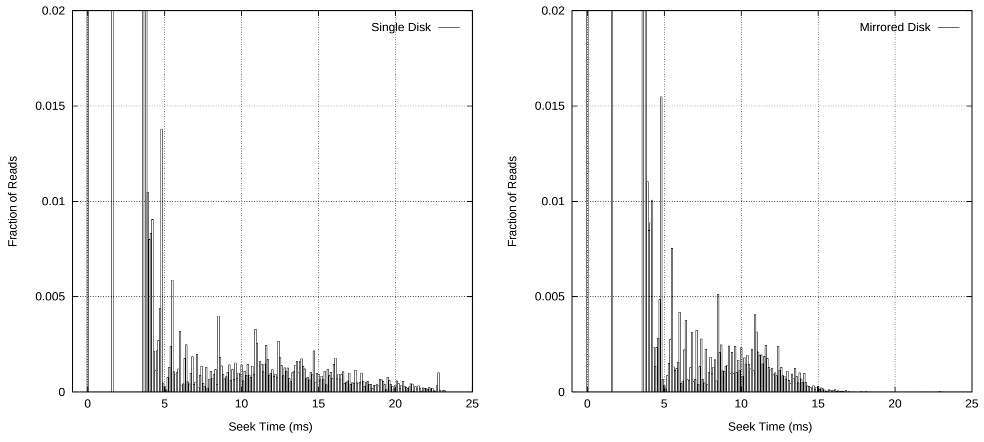

Fig. 6. pdf seek time, reads only.

4). In case of having read/write traces the corresponding improvements between the single disk and the mirrored disk model are 25.4% for the seek distances and 9% for the seek times. It is obvious that reads benefit the most by the shadowed disk set.

Lastly, we included both reads and writes of the trace data in our simulation. The seek reduction with more than two disks in the set is not impressive and seems to be stabilized for over than five disks, since the write percentage influences the overall performance. It is, also, interesting to note the improved simulation model behavior in comparison to the theoretical expected seek distance based on Expression (1). For example, in case of mirrored disks (k=2) and only reads (r=1) we expect a seek distance of $\frac{C}{5} = 392.4$ for the considered HP-97560 disk model ($C = 1962$ cylinders). The resulted seek distance for this example is almost 1/5 of the analytical expected seek distance. This is explained by the fact that theoretical results assume independence between consecutive requests, which does not happen in practice. The improvements gained by the mirrored disk set have been investigated in detail and discussed in the next subsection.

5.2 Mirrored disk subsystem

The mirrored disk simulation model was tested with traces of the time period between 1992.04.29 to 1992.05.02. There were 393,937 requests processed, out of which 43.7% were reads and the remaining 56.3% were writes. As in the shadowed disk subsystem, our simulator was tested under traces collected at this period.

Firstly, we have run the simulator for a working set of traces having isolated the reads. Figure 4 depicts the cumulative distribution function (cdf) for reads and writes as a function of the seek distance. These figures indicate that the mirrored disk benefit is uniformly distributed for long requests, whereas differences between single and mirrored disk set are more intensive for short seek distances. This was expected since shorter seeks could be handled more effectively in a mirrored disk subsystem due to the availability of the second drive.

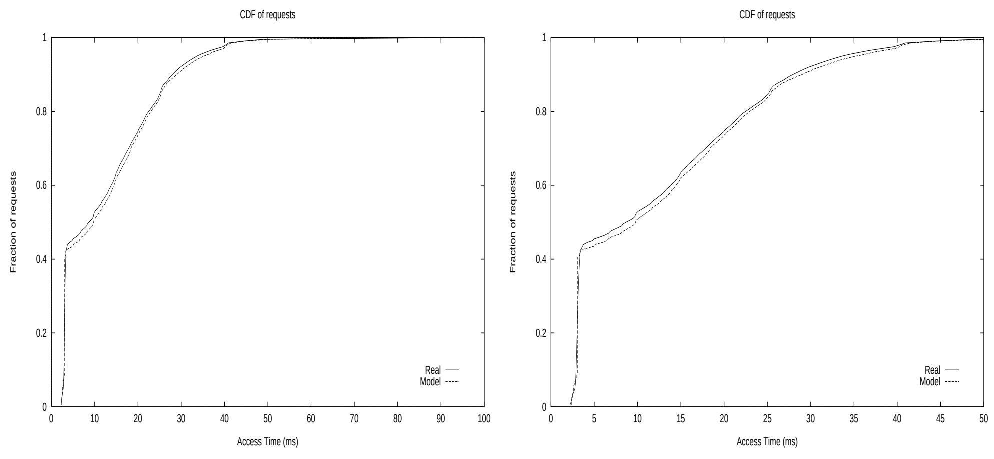

Fig. 7. cdf access time up to 100 ms and up to 50 ms.

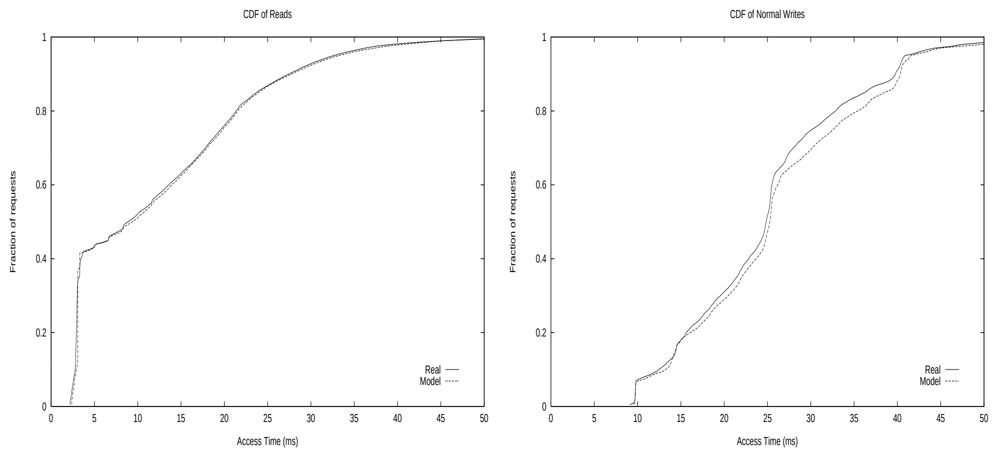

Fig. 8. access time up to 50 ms, reads and writes.

Furthermore, the probability distribution function (pdf) for seeking is presented in Figure 5 for the single and mirrored disk model, left and right part respectively. As indicated in these figures, the percentage of long seeks is eliminated in the mirrored disk set followed by an increase in the rate of moderate size seeks. Figure 6 represents the probability distribution function for the seek time for both the single and mirrored disk subsystem. As in the case of seek distances, the long-time seeks tend to reduce in the mirrored disk model.

5.3 Simulation model verification

The accuracy of the developed simulation model is essential in producing good and realistic simulation results. To validate the developed simulation model we have evaluated the demerit figure of the model, as suggested in [12]. The demerit figure serves as the metric for the performance differences between the real disk drive and the simulation model. This metric is evaluated by the root mean square of the horizontal distance between the modeled and the real time cdf's (measured in ms).

To evaluate the real and the simulation models, disk I/O traces of the time period between 1992.04.26 to 1992.05.01 were used. More specifically, in Figure 7 the cumulative distribution of access times up to 100 ms and 50 ms (left and right part respectively) shows a demerit figure of 0.705 ms, corresponding to a 5.63% difference between the real and the simulation model. Figure 8 depicts the demerit figure estimation, for the cumulative distribution of access times up to 50 ms for reads and writes. The demerit figure percentage for reads up to 50ms is 2.63% whereas the corresponding percentage for writes is 5.28%. These demerit figures verify the accuracy of our simulation model, since other disk simulators have shown similar demerit figures (e.g. 5.7% in [12] and 3.9% in [7]).

6 Conclusions

The simulation model presented here applies the nearer-server rule to a shadowed disk subsystem. The simulation process included almost all the necessary parameters. Real disk I/O traces were used and certain conclusions were raised about shadowed disk sets. The mirrored disk was studied in detail as an indicative disk subsystem which could be easily implemented in a computer system. Mirrored disk model has been proven to outperform the conventional disk subsystem, since it has shown considerably improved seeking and service times under real I/O workload. Our simulation results complemented the theoretical background for the shadowed disk subsystem and documented performance metrics such as access and service times, cumulative distribution and probability density functions.

This simulation model could be extended by introducing shadowed disks of different configurations to study the shadowed disk set performance under varying disk topologies. Scheduling and data placement techniques could be imposed to the developed simulator to further improve performance and decide on a specific request scheduling and servicing. Various scheduling algorithms could be applied to the simulation model in conjunction with disk rearrangement and tested under different workloads to expand and integrate the disk simulation model.

Acknowledgments

The authors thank Dr. John Wilkes (Hewlett-Packard Laboratories, Storage Systems Program) for providing the disk I/O traces and helping in their introduction to the present work.

References

1. D. Bitton and J. Gray: "Disk Shadowing", *Proceedings 14th VLDB Conference*, pp.331-338, 1988.

2. D. Bitton: "Arm Scheduling in Shadowed Disks", *Proceedings IEEE COMPCON 89 Conference*, pp.132-136, 1989.
3. P. Chen, E. Lee, G. Gibson, R. Katz and D. Paterson: "RAID - High Performance, Reliable Secondary Storage", *ACM Computing Surveys*, Vol.26, No.2, pp. 145-185, 1994.
4. G. Ganger: "System-Oriented Evaluation of Storage Subsystem Performance", University of Michigan, Technical Report CSE-TR-243-95, Jun 1995.
5. G. Ganger, B. Worthington and Y. Patt: "The DiskSim Simulation Environment", University of Michigan, Technical Report CSE-TR-358-98, Feb 1998.
6. G.A. Gibson, J.S.Vitter, J. Wilkes et al.: "Strategic directions in Storage I/O Issues in Large-Scale Computing", *ACM Computing Surveys*, Vol.28, No.4, pp.779-763, Dec 1996.
7. D. Kotz, S.B. Toh and S. Radhakrishnan: "A Detailed Simulation Model of the HP 97560 Disk Drive", Dartmouth College, Technical Report PCS-TR94-220, Jul 1994.
8. R.W. Lo and N.S. Matloff: "Probabilistic Limit on the Virtual Size of Replicated Disc Systems", *IEEE Transactions on Knowledge and Data Engineering*, Vol.4, No.1, pp.99-102, Jan 1992.
9. Y. Manolopoulos: "Seek Time Evaluation", *Encyclopedia of Microcomputers*, Vol.15, pp.227-245, Marcel Deker, 1995.
10. C. Ruemmler and J. Wilkes: "Disk Shuffling", Hewlett-Packard Laboratories, Technical Report HPL-91-156, Oct 1991.
11. C. Ruemmler and J. Wilkes: "UNIX Disk Access Patterns", *Proceedings USENIX 1993 Winter Conference*, pp.405-420, Jan 1993.
12. C. Ruemmler and J. Wilkes: "An Introduction to Disk Drive Modeling", *IEEE Computer*, Vol.27, No.3, pp.17-28, Mar 1994.
13. E. Shriver: "Performance modeling for realistic storage devices", Ph.D. dissertation, Department of Computer Science, New York University, May 1997.
14. E. Shriver, A. Merchant and J. Wilkes: "An Analytic Behavior Model for Disk Drives with Readahead Caches and Request Reordering, *Proceedings ACM Sigmetrics 98 Conference*, pp.182-191, Jun 1998.
15. A. Vakali and Y. Manolopoulos: "An Exact Analysis on Expected Seeks in Mirrored Disks", *Information Processing Letters*, Vol.61, No.6, pp.323-329, 1997.
16. J. Wilkes: "The Pantheon Storage System Simulator", Hewlett-Packard Laboratories, Technical Report HPL-SSP-95-14, May 1996.