1

On the performance of D-redundant disk systems

E.Bachmat

***Abstract*— In this paper we formally introduce the notion of a D-redundant disk system. The class of D-redundant systems includes the class of physically mirrored disks with D copies as well as the more recently introduced SR systems. We provide a very general lower bound on the average access time of any D-redundant system. Using the lower bound we show that that the SR systems of Yu et el. are within 15 percent of optimal performance. We also show how SR systems can be combined with anticipatory head movement policies (AHM) to provide systems which are optimal within 5 percent.**

I. INTRODUCTION

In this paper we study the following problem which was proposed in [10].

Problem: Given D disks and a data set whose capacity is that of a single disk what is the best way to organize and manage the system in order to maximize performance?

We call systems which satisfy the assumptions of the problem *D-redundant systems*. The motivation for considering such systems for large values of D is the large price difference between disks and RAM memory. The idea is to replace a RAM device by a large number D of disks in such a way that the replacement will still be cost effective and that the deterioration in performance will be relatively small, see Yu et al., [10]. In this paper, following some initial computations and ideas from [10], we provide a mathematical analysis of this problem in the case where D is large and the I/O activity is read activity. To fully describe the systems considered in the problem we have to list all the different hardware and software aspects of such systems. Such a list leads us to the notion of a D-redundant disk system. We use this notion as the formal basis for our mathematical analysis of the problem. It also allows for a fairly concise and unified description of prior results and their underlying assumptions. The classical approach to improving read performance in D-redundant systems was to spread many copies of each data item on different disks and to cleverly choose which data copy will be used to service the request depending on the current position of the disk heads. More recently a new family of systems known as SR arrays were introduced, [10]. These systems use relatively few data copies all contained within the same disk (most of the system's disk capacity is not used). The latter feature vastly simplifies the performance analysis of these systems. Initial computations in [10] suggested that SR systems should display good behavior in read environments and performance which is superior to other more classical systems (shadowed systems) in write intensive applications since they use few data copies. It was not clear however, how far away is the performance of such systems from being optimal. To settle this issue we need good upper bound estimates on the performance of an optimal system.

We establish an elementary upper bound on the performance of any D-Redundant system which closely resembles the volume bound for error correcting codes, see [6].

Comparing this bound to the computed performance of an SR system we will show that the read performance of an SR system is at most 1.22 times slower than that of an optimal system. When we make some realistic assumptions on the physical characteristics of the disk drives in the system we obtain an even better performance ratio of 1.15.

Another strategy which has been considered in an attempt to improve read performance is to move the location of idle heads to better disk positions in anticipation of future requests. This is analogous to the movement of racket ball players which position themselves near mid court while the other player attempts to hit the ball to an undisclosed location. The addition of such strategies can improve the performance of the system. We construct anticipatory head movement (AHM) strategies which Combined with SR systems (or shadowed systems) yields a D-redundant system whose performance is at most 1.07 times slower than optimal. for realistic disks drives the performance ratio is at most 1.05. Constructing an AHM as above is easy. With some more work it is possible to construct a more efficient AHM which moves only a single head at any given time. This construction is somewhat more elaborate. We need to study the typical behavior of a fairly complicated Markov chain associated with our head movement policy. The trick is to design the anticipatory head movement policy in such a way that the associated Markov chain closely resembles the behavior of Hammersley's particle process which was introduced in [5] and thoroughly studied in [1]. We can then leverage the results of [1] to obtain a reasonable understanding of our Markov process and to ultimately justify our computations. This construction will be outlined in the paper.

The performance ratios quoted above assume that the requests to the data set are spread uniformly. We also compute the performance in case there are differences in the popularity of data items leading to a non uniform access pattern. All relevant quantities are multiplied by some factor which depends on the non uniform distribution. Performance ratios are not affected.

Our upper bound on the performance of an optimal system is not tight. The problem of producing tighter bounds is equivalent to a densest packing problem in "disk geometry". SR systems with anticipatory head policies correspond to certain lattice packings. We conjecture in fact that SR systems

Department of Computer Science, Ben-Gurion University, Beer Sheva, Israel. Email:ebachmat@cs.bgu.ac.il. Research partly supported by an IBM faculty award.

coupled with the anticipatory head movement policy yield asymptotically optimal read performance. We present some mild evidence in support of this claim by showing that the lattice packings corresponding to SR systems are densest among lattice packings (for linear seek functions).

II. Definitions and preliminaries

In this paper we consider the performance of *D-redundant* disk systems in a read only setting. A D-redundant disk system consists of the following elements

A) A set of D identical disks containing a data set. The physical characteristics of a disk are described by the rotational latency and seek parameters R, and S and a seek function f.

B) A data set composed of many small data units called *data blocks*, each such block is composed of a fixed (relatively small) number of bytes, a typical example being 512 bytes. We assume that the total size in bytes of the data set equals the capacity of a single disk drive.

C) A configuration

D) A service policy

E) An anticipatory head movement policy

We now describe these different software and hardware components which together make up the D-redundant system in more detail. We begin with the disks of the system.

We capture the physical characteristics of a particular disk drive by the following quantities. choose some ray to represent points whose angle is 0. In order to specify the relative location of data on different drives we assume that at a certain starting time $t = 0$ all disk heads are at angle 0, given their identical nature and hence identical rotational speeds the disk heads will then remain in the same angular position at all times. As noted in [10], in practice disks will not remain synchronized unless forced to, however this will be of little relevance to our ensuing discussion. The angle of a disk location is measured with respect to angle zero in the direction of the disk's rotational motion. Since disks rotate at a constant speed we may measure angles by time. we let R be the time it takes the disk to complete a full circle, then R will be the size of a full circular angle (360 degrees). Also, we define S to be the time needed for excecuting a full stroke (maximal) seek. All other angles and radial locations are computed from R and S using linear extrapolation, for example, the coordinate of the disk location which is a third of a rotation away from angle zero and a quarter of the way from the inner circle of the disk towards the outer circle is $(R/3, S/4)$.

Let $f(\theta)$ be the function which describes the radial distance that the disk head can seek in time θ. We note that the seek has to start and finish with no radial motion since the disk head must stabilize itself on a given track in order to read or write. f is known as the *Seek function*. Since the disk rotates at a constant speed we can choose time units so that angles measure time and hence $f(\theta)$ is the radial distance that the disk head can seek in a given amount of time θ.

f is linear if the seek speed is constant. More realistically, since the seek starts and finishes without radial motion, the head will accelerate, reach some maximal speed and then decelerate when approaching the targeted track. This scenario implies that the seek function is convex. We will consider the family of seek functions of the form $f(\theta) = c\theta^a$ for $c > 0$ and $a \geq 1$. We will only be interested in short seeks hence we assume that this formula holds only for small angles θ (otherwise c would be determined by R,S and a). In fact it is common in the disk modeling literature to use different formulas for small and large angles, see the widely quoted formulas of [9] for example.

Remark: In [9] and many other references the value $a = 2$ is used for short seeks with c depending on the specific disk being modeled. We also note that other references including [9] describe f^{-1} rather than f.

If the disk head is at location $m = (r_1, \theta_1)$ and the request is to $n = (r_2, \theta_2)$, then to service the request the disk will have to seek from r_1 to r_2, this takes $t_s = f^{-1}(|r_1 - r_2|)$ time, where f^{-1} denotes the inverse function of f. After completion of the seek the head is at angle $\theta_1' = \theta_1 + t_s$ where the angular addition is done modulo R. The rotational latency time t_l it takes the disk to rotate from θ_1' to θ_2 is $t_l = \theta_1' - \theta_2$ if $\theta_1' \geq \theta_2$ and is $t_l = R - (\theta_2 - \theta_1')$ otherwise.
The *access time* is $d(m, n) = t_s + t_l$. Note that $d(m, n)$ is not symmetric due to the non symmetric definition of t_l and hence is not a metric. It does satisfy the triangle inequality.
In addition to seek and rotational latency, service time also includes transfer time. Block transfer time is system independent hence in line with all previous studies we will ignore transfer time in our analysis and equate service time with access time which may be viewed as the system overhead.

We can use the fact that the total capacity of the data set residing on the disks of a D-redundent system is that of a single disk to index the elements of the data set by laying out the data set on a disk which is not part of the system and using (r, θ) as an index to the data element which is layed out in that location. In a physically mirrored system these indexes do correspond to actual physical locations but for more general configurations this is not the case. We shall index elements of the data set using this convention throughout the paper.

A *configuration* consists of a choice of *multiplicity* for each data block, multiplicity being the number of copies of the data block, and a choice of physical locations on the various disk drives for all these copies for all data blocks. We allow different copies of the same block to reside on the same disk. We also allow portions of some or all disks to be empty. In this paper we will assume that the choice of a configuration is fixed once and for all.

The most famous example of a D-redundant disk system configuration is that of a *D-shadowed system* (physical mirroring). In such systems each data item has D copies, one per disk, residing at identical corresponding locations on the different disk drives. Up to index permutations of data blocks such a configuration is unique.

A *service policy* is an algorithm which decides, given a read request, which copy of the requested block of data will be read, assuming the multiplicity of the block is more than 1. In general the choice of a copy will depend on the state of the system at the time when service to the I/O request is initiated.

A very commonly studied service policy is the *Nearest server policy* (NS). In this policy, given a request to some data block, one considers which data block copy has the smallest access time, given the current positions of the disk heads and chooses that copy to service the request.

An *anticipatory head movement policy* is a policy for moving the position of some disk heads which are not currently in the process of servicing user I/O requests in anticipation of future requests. Such policies again will usually depend on the current locations of the various disk heads. We will refer to such policies as AHM policies.

In order to study the performance characteristics of D-redundant systems we also need to describe the user I/O pattern. In this paper we will restrict ourselves to the situation where the data block content is fixed, in other words all I/O activity will be read activity. Read only systems are not uncommon. In many database applications periods of intensive write updating alternate with extensive periods of read only queries, hence the system is exclusively read only, for extended periods.

In line with all related analytical work, see [4],[7],[8],[10], we assume in this paper a *synchronous* timing model in which the user issues a new I/O request once the previous one completes. this is a serial applications model. We note that the synchronous assumption implies that there are no queues at the disks.

Given this timing assumption we can make the following definition. We say that an AHM policy is *admissible* if all the anticipatory head movement completes before the arrival of the next incoming request. Thus in the synchronous timing model all heads must complete movement before the head servicing the current request completes its service.

As in all related analytic work we assume that request locations are drawn independently of each other in accordance with some fixed density function $q(r, \theta)$. Previous analytical studies usually assumed that q is the uniform distribution.

Let $l = (r, \theta)$ be the index of a request. Assume that there are k copies of l with locations $(r_i(l), \theta_i(l))$ on disks $D_{j(i)}$, $1 \leq i \leq k$. Recall that a service policy A for a D-redundant system is a rule which given the current positions $\bar{H} = ((r_1, \theta_1), ..., (r_D, \theta_D))$ of the disk heads, the current request l and the request density q, decides which copy of l will service the request. Let us denote the copy which services the request by $A(q, \bar{H}, l)$. In addition the AHM policy, which we will denote by B, may define given $\bar{H}$, l and q, to position the heads in new positions $\bar{H}'$. Any such rule together with the density q defines a Markov chain, M, whose states are the head position vectors $\bar{H}$ and with some appropriate transition densities $\nu_{\bar{H}}(\bar{H}')$. Let $dp(\bar{H})$ denote the stationary distribution of the head positions and let $d\mu$ denote the normalized uniform measure in the (r, θ) rectangle, namely $\frac{1}{RS} dr d\theta$. The the average access time of a D-redundant system with service policy A and AHM policy B, given a request distribution $q(r, \theta) d\mu$ is given by

$$E_{A,B,q}(D) = \int_{\bar{H}} (\int_{(r,\theta)} d(\bar{H}, (r, \theta)) q(r, \theta) d\mu) dp(\bar{H})$$

Where the internal d refers to the disk access time required to service $l = (r, \theta)$ given the head positions $\bar{H}$ and the service policy and the external d's are differentials.

In this paper we will be concerned with the asymptotic performance of D-redundant systems. Our basic object of study will be families of D-redundant systems with increasing numbers of disks D but with the same disk characteristics, configuration, service and AHM policies. Let W denote such a family. For such families the average access time will be a function of the number of disks and we shall denote this function by $E_W(D)$. We will be concerned with the asymptotic behavior of $E_W(D)$, namely given two different families W and W' we will try to compute $lim_{D \to \infty} E_W(D)/E_{W'}(D)$. We note that this type of analysis is insensitive to lower order terms and we will use this flexibility to simplify computations.

A. SR systems

Let k, l be integers and let $D = kl$. A (k, l) SR system is a D-redundant system with the following configuration.

Each block has multiplicity l. If the data is indexed by (r, θ) then it is placed in the (k, l) SR system on disk $[r/(kl)]$ at locations $(l(r - ([r/(kl)]/kl)), (\theta/l + i) mod l)$ for $i = 0, ..., l - 1$.

We note that only the range $0 \leq r \leq 1/k$ of each disk is occupied by data, this serves to reduce the seeks. In addition each data has l evenly spaced copies in the angular direction and this reduces the rotational latency. An important feature of SR systems is that each piece of data has all it's copies on a single disk hence only one disk head can service a given request. The service policy is nearest server.

Remark: The definition of an SR systems above is a formal description of the systems which were introduced in [10]. As defined above, these systems suffer from several drawbacks

(1) Their performance with respect to sequential workloads is poor since it uses only a single disk head at a time.

(2) SR systems provide no data protection.

(3) The construction of SR systems is insensitive to user access patterns, therefore if the access pattern is not uniform the performance will not be as good.

Item (1) is easily addressable by shuffling consecutive data tracks (small groups of consecutive blocks) onto different disks. This shuffling procedure is also commonly known as *striping*.

A solution for problem (2) is suggested in Yu et al. [10]. The authors suggest creating a hybrid system which mixes SR and mirroring. This solution seems acceptable.

item (3) seems to be much more difficult to solve, one has to either rearrange the system in reaction to changes in access patterns, a costly operation or know in advance the type of pattern which will be used.

B. related work

Several papers have dealt with the calculation of the average seek time for physically mirrored systems with various service and AHM policies. Generally speaking they assume linear seek functions and read only, uniform distribution I/O streams. Most of them deal with the nearest server policy.

With the above assumptions and with an empty AHM policy the case $D = 2$ has been analyzed in great detail by Calderbank et al. in [4]. For $D = 2$, Hofri studies the same problem assuming a non admissible AHM policy which moves a single head at a time, [7]. King considered the case of $D = 2$ with a general non admissible AHM policy, [8]. All (the many) other studies of the nearest server policy contain a basic error. They assume that when the access distribution q is uniform the stationary distribution $dp(\bar{H})$ is also uniform. This is not true as already shown in [4]. See [2] for a more detailed discussion. The best known of the incorrect studies is the widely cited reference [3]. To date there is no correct analytic or experimental study of the behavior of physically mirrored systems with $D > 2$. The analysis seems to be very difficult.

All the above references consider seek time only. As noted before access time minimization, including rotational latency, has recently been considered by Yu et al. in [10], still under the assumption of uniformity of requests and a linear seek function. In that paper SR systems were introduced and studied experimentally with several benchmarks on which SR systems displayed good performance. Reference [11] deals with write intensive appliocations.

III. Performance upper bounds

In this section we present an upper bound on the performance of D-redundant systems. Equivalently we establish a lower bound on the average access time of such systems. The bound is very general and applies essentially to all D-redundant systems even when one allows dynamic configuration changes, varying data multiplicities and any AHM policy. Throughout this section we assume that q is uniform. The results will be generalized to all density functions in section V. Let W denote a D-redundant system and let $E_W(D)$ be its average access time function.

theorem: *Let ρ be such that*

$2\int_0^\rho f(\theta)d\theta = \frac{RS}{D}$

and let

$L = \frac{2D}{RS}\int_0^\rho f(\theta)\theta d\theta$

then

$E_A(D) \geq L$

Proof: Assume that at the time of arrival of a request $l = (r,\theta)$, the D disk heads are at locations $p_1, ..., p_D$, $p_i = (r_i, \theta_i)$. Assume that there are k copies of data item l, residing on disks $D_{i(1)}, ..., D_{i(k)}$ at positions $s_j = (r(j), \theta(j))$, $j = 1, ..., k$. The disks $D_{i(j)}$ need not be distinct. Let $d_j = d(p_{i(j)}, s_j)$ be the access time from the disk head position of disk $D_{i(j)}$ to the physical location of the j'th data copy of l. Let $d = Min_j d_j$ be the minimal distance from any disk head position to a data copy location on the same disk. It is obvious that the shortest service time for the given request will be achieved if the request is serviced by the nearest head and the service time will be denoted by $d = d(p_1, ..., p_D, l)$

Denote by $e(p_1, ..., p_D)$ the average of $d(p_1, ..., p_D, l)$ over all possible l. Recalling that q is uniform we obtain,

$e(p_1, ..., p_D) = \frac{1}{RS}\int_0^S\int_0^R d(p_1, ..., p_D, (r,\theta))drd\theta$

Let $d\mu = \frac{1}{RS}drd\theta$. We wish to show that L is a lower bound for $e(p_1, ..., p_D)$ for all $p_1, ..., p_D$. This will establish our theorem since $E_W(D)$ is a weighted average of $e(p_1, ..., p_D)$ over all choices of head positions.

We define the Voronoi diagram of $p_1, ..., p_D$ and the system W, to be the partition of the set (space) of data elements $l = (r, \theta)$ into sets V_i', $i = 1, ..., D$, consisting of all data elements l which are serviced by a copy residing on disk i, given that the disk heads are in positions $p_1, ..., p_D$ and that we employ the service policy of W.

The sets V_i' are called the Voronoi cells of the diagram attached to $p_1, ..., p_D$ and W. All data elements l which are in V_i' have copies on disk i. If an element l has more than one copy on disk l we consider the copy which is closest to the disk head position p_i. We let V_i denote the set of physical locations on disk i of the data copies of elements $l \in V_i'$.

Since V_i is the set of locations of the closest copies of data elements serviced by the head located at p_i we have

$e(P_1, ..., P_D) \geq \frac{1}{RS}\sum_i\int_{V_i} d(p_i, (r,\theta))d\mu.$

Inequality may arise if the service policy of W decides not to use the closest copy for service. We define for any location p on a disk

$B_{p,\rho} = \{(r,\theta) \mid d(p, (r,\theta)) \leq \rho\}$

where (r, θ) refers to a physical disk location. Stated otherwise, $B_{p,\rho}$ consists of all disk locations whose access time from p is at most ρ.

Choose ρ_i such that $\mu(B_{p_i,\rho_i}) = \mu(V_i)$, namely ρ_i is chosen so that $B_(p_i, \rho_i)$ has the same disk capacity as V_i. Denote B_{p_i,ρ_i} by B_i. We claim that

$\int_{B_i} d(p_i, (r,\theta))d\mu \leq \int_{V_i} d(p_i, l)d\mu.$

To establish the claim, let $A = B_i \cup V_i$, $C = B_i - V_i$ and $E = V_i - B_i$. By definition B_i is the disjoint union of A and C, and V_i of A and E. By the choice of ρ_i we have

$\mu(A) + \mu(C) = \mu(B_i) = \mu(V_i) = \mu(A) + \mu(E),$

hence $\mu(C) = \mu(E)$. For any point (r,θ) in C we have $d(p_i, (r,\theta)) \leq \rho_i$ and for any (r, θ) in E we have $d(p_i, (r, \theta)) \geq \rho_i$. We conclude that

$\int_{B_i} d(p_i, (r,\theta))d\mu$

$= \int_A d(p_i, (r,\theta))d\mu + \int_C d(p_i, (r,\theta))d\mu$

$\leq \int_A d(p_i, (r,\theta))d\mu + \mu(C)\rho_i = \int_A d(p_i, (r,\theta))d\mu + \mu(E)\rho_i$

$\leq \int_A d(p_i, (r,\theta))d\mu + \int_E d(p_i, (r,\theta))d\mu$

$= \int_{V_i} d(p_i, (r,\theta))d\mu$

as claimed. We conclude that

$e(p_1, ..., p_D) \geq \sum_i \int_{B_i} d(p_i, (r,\theta))d\mu = L_1$

Let p be some arbitrary point in the disk. Let $\rho_1 > \rho_3 > 0$, and let ρ_2 be such that $\mu(B_{p,\rho_1}) + \mu(B_{p,\rho_3}) = 2\mu(B_{p,\rho_2})$ then we claim that

$$\int_{B_{p,\rho_1}} d(p,l)d\mu + \int_{B_{p,\rho_3}} d(p,l)d\mu \geq 2\int_{B_{p,\rho_2}} d(p,l)d\mu.$$

The argument is very similar to that of the previous claim. Let $A = B_{p,\rho_1}$, $B = B_{p,\rho_2}$ and $C = B_{p,\rho_3}$. By construction $\mu(B - A) = \mu(C - B)$. The difference between the two sides of the inequality is

$$\int_{C-B} d(p,(r,\theta))d\mu - \int_{B-A} d(p,(r,\theta))d\mu \geq \mu(C-B)\rho_2 - \mu(B-A)\rho_2 = 0$$

as claimed. By successively applying the claim to pairs of summands in the expression

$$L_1 = \sum_i \int_{B_i} d(p_i,(r,\theta))d\mu$$

whose corresponding radii ρ_i differ, we obtain

$$L_1 \leq D\int_{B_{p,\rho}} d(p,(r,\theta))d\mu.$$

Here ρ satisfies $\mu(B_{p,\rho}) = \frac{1}{D}$

since the total area of the balls involved in the claim doesn't change. Finally, denote the "circle" of radius ρ around some point $p = (r_0, \theta_0)$ by $s_{p,\rho}$. By circle we mean the locus of all disk locations whose access time from p is equal to ρ. $s_{p,\rho}$ is the interval of points of the form

$$(r_0 + r, \theta_0 + \rho),\ -f(\rho) \leq r \leq f(\rho).$$

For a given ρ we have

$$\int_{B_{p,\rho}} d(p,(r,\theta))d\mu$$
$$= \int_0^{\rho}\int_{-f(\theta)}^{f(\theta)} \theta d\theta dr$$
$$= \int_0^{\rho} 2f(\theta)\theta d\theta \ \ q.e.d$$

IV. Average access time calculations for $f(\theta) = c\theta^a$ and a uniform access distribution

Throughout this section we assume that q is uniform.

A. The performance of SR systems under uniform distribution

Consider a (k,l) SR with $D = kl$ disks. At any given moment any given head is positioned at the location of the latest request it has serviced. The r coordinate of that location is uniform in an interval domain of size $1/k$. Since the next request will also have a uniform radial location in that range the average seek will be given by

$$E_{seek} = \int_0^{1/k}\int_0^{1/k} f(|x-y|)dxdy$$

After seeking to the radial position of the new request, the disk has to rotate to the angular location of the new request. Since that location is uniform and independent of the radial location the radial positions of the data in the SR system will form a set of l equidistant angles shifted by some uniformly distributed value in the range of $[0, R]$. Since the previous request had an arbitrary shift the same holds after the seek and hence the average rotational latency time is $E_{rot} = R/(2l)$. We may now minimize $E = E_{seek} + E_{rot}$ over all pairs k, l such that $D = kl$. To avoid issues of integrality we may minimize over k, l real, that is over pairs $x, D/x$ and optimize by setting the derivative of E to zero. We then choose $k = [x]$ and $l = [D/[x]]$. this way a relatively small number of disks will never be used, however it is easy to show in all our examples that the error in using the continuous approximation has no effect on the asymptotics for large D.

We compute the performance of an optimized SR system. The simple procedure follows the one outlined in [10]. We write $D = x\frac{D}{x}$, $x \in [0, D]$ ignoring integrality as noted in the previous section. To compute the seek time of a radial movement of r units we must invert the seek function to obtain $g(r) = f^{-1} = \frac{1}{c^{1/a}}r^{\frac{1}{a}}$. Since each disk head is responsible for a region of size S/x in which requests are uniformly distributed, the average seek is

$$\frac{x^2}{S^2}\int_0^{S/x}\int_0^{S/x}\frac{1}{c^{1/a}}|r_1 - r_2|^{\frac{1}{a}}dr_1dr_2$$
$$= \frac{2}{c^{1/a}}\int_0^1\int_0^{r_1}\left(\frac{(r_1-r_2)S}{x}\right)^{\frac{1}{a}}dr_2dr_1$$
$$= \frac{2a^2}{c^{1/a}(a+1)(2a+1)}\left(\frac{S}{x}\right)^{\frac{1}{a}}.$$

Since each disk head is responsible for an angular sector of size $R/(D/x) = \frac{xR}{D}$ the average latency is simply $\frac{xR}{2D}$. We add the two expressions, to obtain the average total access time, differentiate and set the derivative to zero to obtain the optimal x. Plugging this x into the average total access time expression yields

$$I_f(D) = 2^{\frac{a-1}{a+1}}(a+1)^{\frac{1}{a+1}}a^{\frac{a}{a+1}}(2a+1)^{-\frac{a}{a+1}}\frac{RS}{cD}^{\frac{1}{a+1}}$$

B. Calculation of the lower bound on average access time

We calculate the lower bound for seek functions of the form $f(\theta) = c\theta^a$ for $a \geq 1$.

The equation $2\int_0^{\rho_0} f(\theta)d\theta = \frac{RS}{D}$ implies

$$\rho_0 = \left(\frac{RS(a+1)}{2cD}\right)^{\frac{1}{a+1}}$$

Computing

$$L_f(D) = \frac{D}{RS}\int_{B_{P,\rho_0}} d(P,R)dR = 2\int_0^{\rho_0} c\theta^{a+1}d\theta$$
$$= \frac{2cD}{(a+2)RS}\rho_0^{a+2} = \frac{2cD}{(a+2)RS}\frac{RS(a+1)}{2cD}\left(\frac{RS(a+1)}{2cD}\right)^{\frac{1}{a+1}}$$
$$= \frac{a+1}{a+2}\left(\frac{RS(a+1)}{2cD}\right)^{\frac{1}{a+1}}.$$

C. Comparison

Let K_a denote the ratio of I_f the performance of the optimized SR system and L_f the lower bound on the average access time of any D-redundant system. We have

$$K_a = \frac{2a}{2a+1}^{\frac{a}{a+1}}\frac{a+2}{a+1}$$

We note that K_a is a decreasing function whose asymptotic limit is 1.

For linear seek functions, $a = 1$, which is the case studied in [10] we have $K_a = \sqrt{3/2} = 1.22$.

For $a = 2$ which is a popular choice in disk modeling, see [9], we have approximately $K_a = 1.15$. We conclude that the performance of optimized SR systems are at most 15 percent from optimal.

V. Allowing AHM policies

We consider the case in which heads, other than the one currently servicing a request, are allowed to move while service to the current request has not completed. We analyze an SR system with an added AHM policy but the results can also be obtained for systems which resemble physically mirrored (shadowed) systems with the nearest server policy. The advantage of physically mirrored systems is that they will be able to adapt to non uniform access patterns without configuration changes.

The idea is to place nearly all the heads in the middle of the radial strip they are responsible for, nearly all the time. Assuming we can do that, we can repeat the previous procedure for computing the performance under the assumption that the heads are in the middle of the strip. To optimize performance in the modified SR system we have to minimize the new access time function which assuming all disk heads are radially in the center of the strip is $\frac{xR}{2D} + c^{-1/a} a^{-1} \frac{S}{2x}^{1/a}$. Solving as before we obtain for the optimal total access time the value

$$J_f(D) = 4^{-\frac{1}{a+1}} (a+1)^{\frac{1}{a+1}} (\frac{RS}{D})^{\frac{1}{a+1}}$$

Denoting by H_a the ratio of J_f to L_f we obtain
$H_a = 2^{-\frac{1}{a+1}} \frac{a+2}{a+1}$
H_a is again a decreasing function with asymptotic value 1. For $a = 1$ we obtain approximately $H_f = 1.07$ and for $a = 2$ we obtain $H_f = 1.05$. We conclude that the performance of modified SR systems as at most 5 percent from optimal.

To justify our computation we have to show that we can find an AHM policy that will align almost all heads almost all the time almost at the center of their radial strip.

we normalize the radial strip of a given disk head to be the interval $[-1, 1]$ so that the distance from the center of the strip to the edges is 1. Let $d_i(t)$ denote the distance between the radial location of the i'th head and the center of the strip at some given moment. We say that a disk head is *centered* if $d_i(t) = 0$. We say that an AHM policy *centers* head i (at time t) if the policy moves the head to the center of the strip. Fix some $\varepsilon > 0$. We say that at a given time t a disk head i is *almost centered* if $|d_i(t)| < \varepsilon$. A state consisting of the radial positions of all the disks is said to be *strongly non centered* if there are at least εD heads which are not almost centered

Let x be the radial location of a disk head and assume that the head must service a request at radial location y. We say that the head performs a *long seek* if $|x - y| \geq 1$.

We consider the following admissible AHM policy.

When a long seek is performed, all heads which are not servicing the request are moved to the center of the radial strip, i.e., are centered.

Theorem: *The above AHM policy is admissible. When applied in conjunction with an SR system the probability of a strongly non centered state approaches zero, as D tends to infinity. Stated otherwise, asymptotically almost all heads are almost centered almost all the time. consequently J_f correctly computes the asymptotic performance of such modified SR systems.*

Proof: The policy is clearly admissible since the centering of a head requires a seek whose distance is at most 1, while a long seek covers a distance of at least 1. Consider the amount of time that the system spends in a strongly non centered state. We divide the time (measured in I/O) into cycles. A cycle begins when all heads but one are centered and ends when such a state is reached again. Consider a cycle which contains a strongly non centered state. Since at the beginning of a cycle at most one head is non centered it takes at least $\varepsilon D - 1$ I/O requests to reach such a state. We consider the probability of remaining for k consecutive I/O in a strongly non centered state. If $|D_i(t)| \geq \varepsilon$, then with probability at least $\varepsilon/2$ a request of service to disk i will result in a long seek. For example if $D_i(t) > 0$ then a call to an I/O with radial position in the range $[-1, D_i(t) - 1]$ will result in a long seek. Since by definition of a strongly non centered state there are at least εD heads for which $D_i(t) \geq \varepsilon$ we conclude that there is a probability of at least $\delta = \varepsilon^2/2$ for a long seek, whenever we are at a strongly non centered state. Therefore the probability of m consecutive strongly non centered states is at most $(1-\delta)^m$. In a similar way we may calculate the probability that the system spends $m + n$ steps in a given cycle, m of which are spent in strongly non centered states and the other n in between such states. The only way that a strongly non centered state can become strongly centered without cycle termination is when a head i which is not almost centered is called to execute an I/O request in the range $[-\varepsilon, \varepsilon]$. The probability of such an event is at most ε. We see that the probability of the system spending m interspersed steps in strongly non centered states in a sequence of $m + n$ steps is at most $(1 - \delta)^m \varepsilon^n$. We see that the expected number of I/O in a cycle in which the system is in a strongly non centered state is finite, while the length of any cycle containing such a state is un bounded, proving the theorem. We note that the same arguments can be used to show that the portion of cycles containing a strongly non centered state is exponentially small. *q.e.d*

It is also possible to achieve the same goal with a more minimalistic AHM policy which moves only a single head (not servicing a request) during any given I/O. We will outline the main ideas involved in the construction while avoiding the less illuminating details.

The AHM policy is given by the following rules

A) assume that a centered head is servicing a request whose distance from the center of the radial strip is U. consider the set of non centered heads whose current distance is smaller than U. Move to the center the one among them with largest distance.

B) If a non centered head is servicing a request and the seek is long, move the head furthest from the center of the strip to the center.

The policy is obviously admissible.

Let us consider the evolution of a system with such an AHM policy. It will be convenient to think of the non centered heads as particles on the interval $[0, 1]$ according to their distance from the center of the strip, with the particles at time t being located at the set of points $\{d_i(t) > 0\}$. We also consider N the time dependent random variable which counts the number of particles present in the system (non centered heads).

Consider now the action taken in rule A of the AHM policy from the point of view of the particle system. The radial location U of the current request is uniform. A particle is created at U since a centered head which does not correspond to a particle becomes non centered at U. On the other hand the non centered head corresponding to the particle to the left of location U becomes centered, hence that particle disappears. The net effect of these two occurrences is the same as that of the particle to the left of U moving to position U. If there is no particle to the left of U a new particle appears at U. This description is essentially equivalent to the description of Hammersley's particle process which was introduced in [5] and studied in [1].

Let N be the number of non centered heads (particles). Let us try to analyze the rates in which N increases or decreases. Assume that the system currently has N particles. Consider rule A first. Rule A is employed when a centered head services the request. That happens with probability $(D-N)/D$. A particle is added to the system if the uniformly chosen point U is between 0 and the left most particle. if our particle system had evolved according to rule A alone as Hammersley's process the results of [1] would show that the left most particle has order of magnitude $O(1/N)$. Since rule B only adds to the number of points near 0 (It moves points from the denser part of hammersley's process near 1) this will hold in our system as well. We conclude that particles are created at a rate of $O((D-N)/ND)$ or, more simply, $O(1/N)$. On the other hand rule B is invoked with the complementary probability of N/D. As shown in the previous proof, the probability of a long seek is some positive constant.

We deduce that particles are eliminated at a rate of $O(N/D)$. Equating the rates we see that the particle system will reach equilibrium when the number of particles N is $O(\sqrt{D})$. Theorem 5 and lemma 9 of [1] are needed to make the argument rigorous.

As in the previous proof we have to deal with the rare, but possible, event in which the particle system strongly deviates from the equilibrium number of particles and N becomes greater than εD. Once the number of particles becomes linear in D the probability of particle elimination becomes constant. The only way the number of particles can grow with better than an exponentially small probability is if the left most particle is at a constant distance away from 0 (and hence particles are generated at a constant rate). We can consider the probability of a large empty interval in Hammersley's process. The large deviation results on page 212 and lemma 9 of [1] can easily be used to show that such an event occurs with probability smaller than, $e^{-\delta\sqrt{D}}$ for some positive $\delta > 0$. This means that starting with a small number of particles, $o(D)$ the system will reach a state in which $N > \varepsilon D$ only after $e^{\delta\sqrt{D}}$ requests on average.

Assuming we have reached the $N > \varepsilon D$ plateau we want to show that with very high probability we can recover quickly and bring the system back to a state with $o(D)$ particles. The idea is that if the AHM policy is turned off for a polynomial (in D) number of requests then the uniform location of requests will move the particles to nearly uniformly distributed locations. with a probability which approaches 1 exponentially fast (The convergence rate of the uniform Markov process). Once the distribution is nearly uniform we trim the number of particles using rule B until we reach the required $o(D)$ particles. Since there are plenty of particles near zero essentially no particles are created and the trimming works at a rate proportional to the large number of particles so it finishes quickly again with exponentially close to 1 probability. We conclude that in a polynomial number of I/O we can eliminate an exponentially rare event (being strongly non centered) and we are done.

VI. Non uniform distributions

Consider a continuous density function $q(r,\theta)$ in the r,θ coordinates and assume that the requests are independently drawn using the density function q.

Theorem: *Let*

$$C_q = (\int q(r,\theta)^{\frac{a+1}{a+2}} d\mu)^{\frac{a+2}{a+1}}$$

then C_q measures the effect of the request density distribution, that is, all performance computations remain valid if we multiply $L_f(D), I_f(D), J_f(D)$ by C_q. In particular performance ratios are independent of the distribution q.

Proof: We subdivide the r,θ rectangle into small rectangles of equal size and assume that q is constant with value q_i in the i'th rectangle E_i. Since q is continuous we can uniformly approximate it by such step density functions. Since we have assumed that q is uniform when restricted to E_i we may apply our previous calculations to that region. The functions I_f, J_f and L_f have the form $CD^{-\frac{1}{a+1}}$ for some constant C. Assume that we assign p_iD of the disks to service requests which fall in E_i then the average seek will be proportional to $\sum_i q_i p_i^{-\frac{1}{a+1}}$, subject to the condition that $\sum_i p_i = 1$. We minimize by setting the partial derivatives $\frac{\partial}{\partial p_i} - \frac{\partial}{\partial p_j}$ to zero. We obtain the equalities $p_i q_i^{-\frac{a+2}{a+1}} = p_j q_j^{-\frac{a+2}{a+1}}$ which implies that p_i is proportional to $q_i^{\frac{a+1}{a+2}}$. Passing to the continuous limit we conclude that the optimal head density in all three situations is $p(r,\theta) = q(r,\theta)^{\frac{a+1}{a+2}}/(\int q(r,\theta)^{\frac{a+1}{a+2}})d\mu$. The average seek is thus proportional to $\int q(r,\theta)p(r,\theta)^{-\frac{1}{a+1}}d\mu = (\int q(r,\theta)^{\frac{a+1}{a+2}}d\mu)^{\frac{a+2}{a+1}}$ which proves the statements of the theorem.

VII. Lattice configurations

In this section we assume again that the distribution q is uniform.

The lower bound on access time which was proved is not achievable by any algorithm since it assumes a tiling of the (r,θ) rectangle by "balls" of equal radius, however such a tiling does not exist. A better bound may be obtained by considering $Min_{(p_1,\ldots,p_D)} e(p_1,...,p_D)$.

Among all head positions we may consider *lattice head positions*. In a lattice head position the points $(p_1,...,p_D)$ are the intersection points of the (r,θ) rectangle with a lattice which consists of all vectors of the form $mv_1 + nv_2$, where m and n are integers and v_1, v_2 some fixed vectors in the real plane. The performance of the modified SR system of section 4 is equal to $e(p_1,...,p_D)$ of the lattice head position given by the vectors $v_1 = (S/x, 0)$, $v_2 = (0, Rx/D)$, with x the optimized value for the SR system. We shall refer to this lattice as the *SR lattice*.

Since, by definition, a lattice is doubly periodic via shifts by v_1 and v_2, so is it's Voronoi diagram, hence all Voronoi cells away from the boundary of the (r,θ) rectangle are isometric, hence the computation of $e(p_1,...,p_D)$ for a lattice head position reduces to the computation of the average access time over a single Voronoi cell

Theorem: *Assume that the seek function is linear. The SR lattice minimizes $e(p_1, ..., p_D)$ over all lattice head positions.*
proof: In classical Euclidean geometry, Voronoi cells meet at lines representing points which are equally distant from two different points. In disk geometry such lines occur only if the two points have the same θ coordinate. The other boundaries of the Voronoi cell are formed by boundaries of balls centered at a point, such boundaries have the slope of the seek function f or of $-f$. A simple analysis now shows that given these constraints and double periodicity, the Voronoi cell corresponding to a lattice can be only of the types represented in figure 1. Type (a) is a parallelogram and represents the generic case, type (b) appears in the SR lattice, while type (c) appears in other lattices containing a vertical vector.

Figure 2 presents a transformation which preserves the average access time of the cell which transforms type (c) cells into type (b) cells. The computation of the optimal modified SR system actually computes the optimized rectangular lattice, hence the best possible cell of type (b). It is easy to show by direct computation that the best Voronoi cell of type (a) is given by an equilateral parallelogram. Finally one verifies that the operation of figure 2 changes the SR lattice Voronoi cell into an equilateral parallelogram, hence it is optimal (but not uniquely optimal) among lattices. *q.e.d*

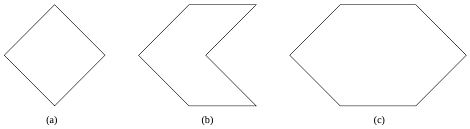

Fig. 1.

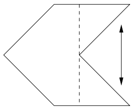

Fig. 2.

VIII. Conclusion and future work

We consider the general notion of a D-redundant system. This notion captures both, physically mirrored systems with D data copies and the SR systems which wre considered by Yu et al. We provid a very general lower bound for the average access time in any D-redundant system which is modeled on the volume bound for error correcting codes. Using the bound we show that SR systems are not far from being optimal. We show how to enhance SR systems with AHM policies and show that the combination is near optimal (in fact, it may be optimal). We also show that these results are independent of the distribution q of the I/O requests. In the future it would be nice to allow the queuing of requests in the system. here the main problem seems to be that of constructing a mildly reasonable model for the interplay between queue size and the number of disks. We hope to come back to these issues in future work.

References

[1] Aldous D. and Diaconis P., Hammersley's interacting particle process and longest increasing subsequences, *Probability theory and related fields*, vol 103, 199-213, 1995.

[2] Bachmat E. and Lam T.K., On the effect of a configuration choice on the performance of a mirrored storage system, *Journal of Parallel and Distributed Computing*, Vol. 65, 382-395, 2005.

[3] Bitton D. and Gray J., Disk shadowing, *Proceedings of the 14th VLDB conference*, 331-338, 1988.

[4] Calderbank A.R, Coffman E.G. and Flatto L., Sequencing problems in two server systems, *Math. Operations research*, vol 10, 585-598, 1985.

[5] Hammersley J.M., A few seedlings of research, *in Proceedings of the 6th Berkley symp. of math. stat. and probability*, vol 1, 345-394, U. of California press, 1972.

[6] Hamming R.W., Error detecting and and error correcting codes, *Bell system technical J.*, vol 29, 147-160, 1950.

[7] Hofri M., Should the two headed disk be greedy?-Yes, it should, *Information processing letters*, vol 16, 83-85, 1983.

[8] King R.P., Disk arm movement in anticipation of future requests, *ACM transactions on computer systems*, vol 8, 215-228, 1990.

[9] Ruemmler C. and Wilkes J., an introduction to disk drive modeling, *IEEE computer*, vol 27(3), 17-28, 1994.

[10] Yu X., Gum B., Chen Y., Wang R.Y, Li K., Krishnamurthy A. and Anderson T.E., Trading capacity for performance in a disk array, *in proceedings of the 4th Symposium on operating system design and implementation*, October 2000.

[11] Zhang C., Yu X., Krishnamurthy A. and Wang R.Y., Configuring and scheduling an eager-writing disk array for a transaction processing workload, *in proceedings of the second usenix conference on file and storage technologies, FAST 2002*, 289-304, 2002.