

DiskGroup: Energy Efficient Disk Layout for RAID1 Systems*

Lanyue Lu and Peter Varman
Rice University, USA
{ll2@rice.edu, pjv@rice.edu}

Jun Wang
University of Central of Florida, USA
{jwang@eecs.ucf.edu}

Abstract

Energy consumption is becoming an increasingly important issue in storage systems, especially for high performance data centers and network servers. In this paper, we introduce a family of energy-efficient disk layouts that generalize the data mirroring of a conventional RAID1 system. The scheme called DiskGroup distributes the workload between the primary disks and secondary disks based on the characteristics of the workload. We develop an analytic model to explore the design space and compute the estimated energy savings and performance as a function of workload characteristics. The analysis shows the potential for significant energy savings over simple RAID1 data mirroring.

1. Introduction

The rise in Internet based services have led to the deployment of increasing numbers of large-scale storage systems in data centers. Energy costs and cooling infrastructures are a growing component of the total cost of ownership of these facilities. For a data center, the storage subsystem can consume 27% of the energy, and this fraction has been increasing by 60% annually [13]. The energy consumed by disks in a web server accounts for 24% of the overall energy, while for web proxy servers, the energy needed grows up to 77% of the total energy [2]. From the viewpoint of thermal effects, the increasing energy consumption leads to higher temperatures, increasing the possibility of disk-head crashes and off-track errors, which degrade the reliability of the storage system [6]. The cost of cooling systems also increases with energy consumption [16]. Motivated by these critical issues, many solutions have been proposed to reduce

the energy consumption in storage systems.

Some researchers [2, 5] have investigated the use of multi-speed disks whose power consumption can be controlled by setting the speed. When the load is light, the disks will be switched to low speed status to save energy. The cost and general availability of specialized disks are a factor in their adoption as an exclusive solution. In [17, 18], power-aware cache replacement strategies to reduce disk energy consumption were presented. Given an input request sequence, they generate a replacement strategy that minimizes the energy consumed by the disks. Simulation results based on multi-speed disks are also presented. Other efforts [8] implement special file systems, which rearrange the data layout on the disk to decrease the disk-access latency to save energy. Due to the popular use of disk arrays in current large storage systems, much research has focused on disk array layouts for energy efficiency [4, 9, 12, 14, 15, 16]. These proposals concentrate on a specific layout and do not systematically study the tradeoffs between data layouts and energy reduction in the larger design space.

In this paper, we present and analyze a family of energy-efficient layouts for data-replicated systems (like RAID 1 for instance). Our scheme called DiskGroup generalizes the standard RAID1 mirroring scheme. Like RAID 1 systems, DiskGroup partitions the array into primary and secondary (or redundant) disks. Unlike RAID 1 where all data on a primary disk is mirrored on one secondary disk, DiskGroup uses a more flexible assignment of data to the redundant disks, with the aim of improving energy consumption characteristics. We develop an analytic model to relate the characteristics of the workload to a grouping scheme, so as to minimize the energy consumption without sacrificing performance. DiskGroup partitions the primary (and secondary) disks into several groups: a primary (secondary) group contains 2^k primary (secondary) disks. Each primary group is associated with a unique secondary group. The data on a primary disk is replicated on the associated secondary group by striping the data on the primary disk

*Supported in part by the National Science Foundation under grant NSF 0615376.

among the 2^k secondary disks in that group.

An example of a system with 8 primary and 8 secondary disks is shown in Figures 3 to 7. Figure 3 shows the data layout of eight blocks on each of the primary disks. In this case, four groupings are possible corresponding to $k = 0, 1, 2$ and 3 respectively. The case $k = 0$ is the standard RAID 1 mirrored organization as shown in Figure 7. There are 8 groups of size 1 each, and the data on a primary disk is exactly mirrored on a secondary disk. The case $k = 3$ results in 1 group of 8 disks as shown in Figure 4; here data on a primary disk is striped across all 8 secondary disks. (The reliability properties of a similar organization has been studied under the name chain declustering [7].) The two remaining cases corresponding to $k = 2$ and $k = 1$ are shown in Figure 5 and Figure 6 respectively.

When the load on the system is light the primary disks are sufficient to satisfy the workload requests. DiskGroup can spin down all the secondary disks to save energy. When a primary disk gets overloaded, its requests can be redirected to one or more disks in its secondary group, while the other disks in the group as well as the disks in the other secondary groups will remain spun down. The workload is characterized by its I/O request distribution that specifies the number of block I/O requests per second (IOPS) on each primary disk. The performance of a particular grouping depends on the characteristics of the workload. For instance, a uniform load will distribute the requests for blocks uniformly among the primary disks, while hot-spotting might result in these requests being concentrated on just a few disks. We develop an analytic model to evaluate the energy savings that can be achieved by different grouping schemes as a function of the workload (number of IOPS and its distribution).

This paper makes the following contributions. First, we propose a new pattern of data placement between the primary disks and secondary disks that generalizes the standard RAID 1 mirroring to achieve better energy savings. Second, we develop an analytic model to compute the estimated energy savings and performance as a function of workload characteristics.

The rest of this paper is organized as below. Related work is discussed in Section 2. Section 3 presents the details of the design and analysis of DiskGroup. Evaluation results are presented in Section 4. Finally, discussion and future work are presented in Section 5.

2. Related work

Current commodity disks have only two power modes: active, when the disk spins at full speed and standby, when the disk stops spinning completely [16]. Active disks can serve requests, while standby disks cannot. However active disks consume much more energy than standby disks.

Motivated by the potential benefits of intermediate disk power modes, Gurumurthi *et al* [5] and Carrera *et al* [2] presented a multi-speed disk model, which can dynamically modulate disk speed. During times of light workload, the multi-speed disk can decrease its rotational speed to save energy while continuing to serve requests. If the workload becomes heavy, it will increase its speed correspondingly. Although this scheme can save significant energy as shown in their paper, it is based on non-commodity specialized disks.

Zhu *et al* [17] advocated a power-aware cache replacement policy that caches more data blocks from lightly-loaded disks, to keep these disks longer in low power mode. In [18] a partition-based cache replacement algorithm, which divides the entire cache into separate partitions, one for each disk, was proposed. It can dynamically adapt to workload changes without tedious parameter tuning. Both experiments use multi-speed disks, and do not consider the potential benefits of data placement.

To reduce disk-head positioning latencies, Huang *et al* [8] implemented a file system named FS2, which dynamically places copies of data in free disk space based on the run-time disk access patterns. Therefore, several copies of a data block can be accessed, potentially saving positioning time and energy. When the disk space utilization is high, FS2 will free the space occupied by these copies, affecting energy and performance gains.

Nightingale and Flinn [10] suggested a distributed file system BlueFS, which uses a flexible cache hierarchy that adaptively decides when and where to access data, based on the energy characteristics of each device. This file system is focused on mobile storage systems and mobile computing.

Colarelli and Grunwald [4] introduced Massive Array of Idle Disks (MAID), which uses a small number of extra active disks as cache disks to hold recently used data, while keeping other non-cache passive disks in low power mode to save energy. However, MAID needs additional cache disks, which is the system overhead. Furthermore, if the data requested are spread on many non-cache disks, MAID will spin up and spin down these disks, increasing energy consumption and latency.

Pinheiro and Bianchini [12] presented Popular Data Concentration (PDC), which dynamically migrates the popular data to a subset of the disks. PDC puts the most popular data on the first disk, the second most popular on the second disk, and so on. It skews the load to a few disks, and keeps others in low power modes. Energy benefits accrue for multi-speed disks but not for conventional disks. Furthermore, PDC gains degrade substantially for long migration intervals.

Zhu *et al* [16] combined several techniques: using multi-speed disks, dynamically adjusting the speed settings of the disks at fixed intervals, and migrating the data to an

appropriate-speed disk while guaranteeing performance.

Yao and Wang [15] proposed a redundancy-based, two-level I/O cache architecture for RAID5, called RIMAC. It enables data on the standby disk to be recovered by accessing data in the two-level I/O cache or on currently active / idle disks. The theoretical upper bound of energy savings in a RIMAC-assisted N-disk RAID5 is $1/N$.

Li and Wang [9] proposed an energy saving policy, eRAID, for conventional disk-based mirrored RAID1 systems. It saves energy by spinning down a partial or entire set of secondary disks, while trying to limit performance degradation. They also developed dynamic performance control schemes for two performance measures, response time and throughput. The mechanisms depend critically on the accuracy of queuing network models and the load predictor. While that paper considers only conventional RAID mirrored layouts, we extend the design space and analyze different layout configurations as a function of the workload characteristics.

Weddle *et al* [14] introduced the gear-shifting power-aware RAID (PARAID). PARAID studies different data layouts for RAID0 and RAID5. It exploits unused storage to replicate and stripe data blocks in a skewed fashion. It assumes that the workload does not change frequently, and is low enough to be handled without performance degradation by the fewer number of disks in a gear.

3. DiskGroup

3.1. Conventional RAID1 System

RAID (Redundant Array of Inexpensive Disks) [11, 3] is a special disk array architecture, which employ two orthogonal concepts: data striping for improved performance and redundancy for improved reliability. RAID1 is a mirroring or shadowing disk array, which maintains two copies of the a data block, in the primary disk and a secondary disk. The primary disk and the secondary disk can satisfy read requests in parallel to enhance performance in terms of throughput and response time. RAID1 is frequently used in large servers and data centers, where availability and transaction rates are more important than storage efficiency. Figure 1 shows the basic disk data layout in a RAID1 system, which has primary disks $P1$ to $P4$, and secondary disks $M1$ to $M4$. A_i, B_i, C_i, D_i represent the data fragments stored on these disks.

A conventional RAID1 system is not designed for energy efficiency. Regardless of the characteristics of the workload, all the disks are running all the time. If the primary disks are sufficient to service the workload, then keeping the secondary disks active is a waste of energy without any performance benefits. In order to save energy, we can spin



Figure 1. Basic mirrored RAID1 system

down some redundant disks to a low power state if that does not violate the performance requirements.

3.2. Disk Load Model and A Simple Optimization

There are several ways to model the disk load, such as request rate, queue length, and response time. Every disk has a maximum load threshold L^T depending on its capacity. Below this load threshold, the disk can satisfy performance requirements such as throughput and response time. If the workload arriving at a primary disk is higher than L^T , we consider this disk to be *overloaded* and must switch part of its load to a secondary disk.

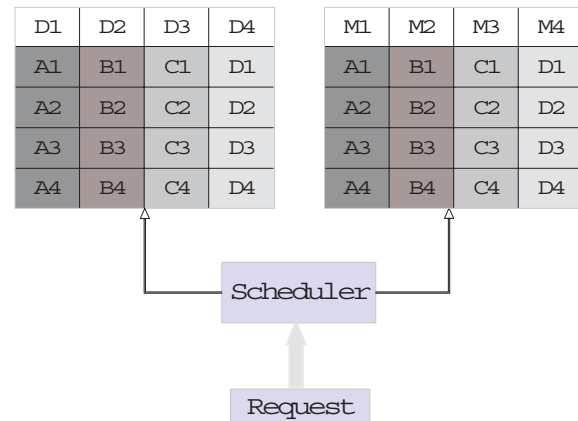


Figure 2. Disk load model, with a simple optimization for energy savings

Figure 2 shows a simple optimization of RAID1 (similar to eRAID [9]) for saving energy. If the load on all disks is smaller than L^T , then the primary disks are enough to serve the requests and all the secondary disks can be spun down. However, if the load on a primary disk is beyond L^T , then we have to spin up the corresponding secondary disk to share the load. In this scheme for every overloaded primary disk, a secondary disk needs to be activated. If the primary disks are only a little overloaded, this results in low utilization of the secondary disks and a waste of energy. Motivated by this, we propose and analyze a family of data layout schemes called DiskGroup. Based on the char-

acteristics of the load, DiskGroup arranges the data on the secondary disks in different configurations to minimize the number of activated secondary disks.

3.3. DiskGroup Data Layout

Before introducing the detailed design of DiskGroup data layout, several assumptions must be noted. Currently, DiskGroup assumes that the workloads contain a high percentage of read requests. Workloads with lots of writes must update data blocks in the powered-off disks, leaving few opportunities to save energy. In our scheme, the occasional writes to a disk in the low power state are saved in a NVRAM and written when the disk is powered up, or in a batch when the write buffer of that disk is full. Second, DiskGroup does not dynamically change the disk layout, but uses a static disk layout configuration over long periods.

The main idea underlying DiskGroup is partitioning the secondary disks into several groups. The data blocks from a primary disk are distributed evenly across all the secondary disks in one group. So, each secondary disk contains data blocks from all the primary disks in the same group. We use a RAID1 system with n primary disks and n secondary disks as an example. For $n = 8$, the partial data layout of 8 successive blocks of each primary disk is shown in Figure 3.

D1	D2	D3	D4	D5	D6	D7	D8
A1	B1	C1	D1	E1	F1	G1	H1
A2	B2	C2	D2	E2	F2	G2	H2
A3	B3	C3	D3	E3	F3	G3	H3
A4	B4	C4	D4	E4	F4	G4	H4
A5	B5	C5	D5	E5	F5	G5	H5
A6	B6	C6	D6	E6	F6	G6	H6
A7	B7	C7	D7	E7	F7	G7	H7
A8	B8	C8	D8	E8	F8	G8	H8

Figure 3. Primary disks data layout

Different group sizes lead to different layouts on the secondary disks. When there is only one secondary group as shown in Figure 4, all the data from one primary disk is striped to all the secondary disks; thus each secondary disk has $1/8$ of the data blocks from each primary disk. When there are two secondary groups, each group consists of four secondary disks. Only data blocks from $P1$ to $P4$ are stored on $M1$ to $M4$, and data blocks from $P5$ to $P8$ are spread on $M5$ to $M8$. Figure 5 shows that each secondary disk has $1/4$ of the data of each primary disk in its group. Figure 6 and Figure 7 show the data layouts when the number of groups is 4 and 8. Notice that RAID1 is only one of the possible configurations of DiskGroup, in which the number of groups is equal to number of secondary disks.

Due to the data layout of DiskGroup, a single secondary disk can contain data blocks from several primary disks. This creates a chance to spin up only a few secondary disks to satisfy requests from several primary overloaded disks. In addition to increasing the utilization of the secondary disks there is potential to save energy consumption since fewer disks need to be activated. For example, if the number of groups is 1, then each secondary disk can serve $1/8$ of the load of every primary disk. Therefore, if disks $P1$ to $P8$ all have a load of $1.125L^T$, then DiskGroup may only need to spin up one of $M1$ to $M8$ to take up the extra $0.125L^T$ load from each primary disk, while RAID1 needs to spin up all 8 disks in this situation. If there are two groups, as shown in Figure 5, DiskGroup has to spin up two disks, one from $M1$ to $M4$, and the other from $M5$ to $M8$. Now consider another case where $P1$, $P2$, $P3$ and $P4$ all have a load of $1.25L^T$. When there is only one group, DiskGroup will wake up two secondary disks, because only $1/8$ of the load of a primary disk can be served by one secondary disk. If there are two groups, then only one secondary disk is enough, one of $M1$ to $M4$.

From the above examples, it can be seen that for different numbers of overloaded disks and the amount of load on the disks, different group configurations require different number of disks to be spun up. We therefore construct an analytic framework to explore this design space and investigate the relationships between the configuration, load and energy conservation.

M1	M2	M3	M4	M5	M6	M7	M8
A1	A2	A3	A4	A5	A6	A7	A8
B1	B2	B3	B4	B5	B6	B7	B8
C1	C2	C3	C4	C5	C6	C7	C8
D1	D2	D3	D4	D5	D6	D7	D8
E1	E2	E3	E4	E5	E6	E7	E8
F1	F2	F3	F4	F5	F6	F7	F8
G1	G2	G3	G4	G5	G6	G7	G8
H1	H2	H3	H4	H5	H6	H7	H8

Figure 4. Secondary disks form 1 group. Blocks from each primary disk are striped across the 8 secondary disks.

3.4. Analysis

For the analysis we begin by considering the requests directed to the primary disks assuming all secondary disks are in the low power state. Let n denote the number of primary (secondary) disks, g be the number of groups, and $s = n/g$ the number of disks in a group. A primary disk

M1	M2	M3	M4	M5	M6	M7	M8
A1	A2	A3	A4	E1	E2	E3	E4
A5	A6	A7	A8	E5	E6	E7	E8
B1	B2	B3	B4	F1	F2	F3	F4
B5	B6	B7	B8	F5	F6	F7	F8
C1	C2	C3	C4	G1	G2	G3	G4
C5	C6	C7	C8	G5	G6	G7	G8
D1	D2	D3	D4	H1	H2	H3	H4
D5	D6	D7	D8	H5	H6	H7	H8

Figure 5. Secondary disks are partitioned into 2 groups with 4 disks in a group

M1	M2	M3	M4	M5	M6	M7	M8
A1	A2	C1	C2	E1	E2	G1	G2
A3	A4	C3	C4	E3	E4	G3	G4
A5	A6	C5	C6	E5	E6	G5	G6
A7	A8	C7	C8	E7	E8	G7	G8
B1	B2	D1	D2	F1	F2	H1	H2
B3	B4	D3	D4	F3	F4	H3	H4
B5	B6	D5	D6	F5	F6	H5	H6
B7	B8	D7	D8	F7	F8	H7	H8

Figure 6. Secondary disks are partitioned into 4 groups with 2 disks in a group

M1	M2	M3	M4	M5	M6	M7	M8
A1	B1	C1	D1	E1	F1	G1	H1
A2	B2	C2	D2	E2	F2	G2	H2
A3	B3	C3	D3	E3	F3	G3	H3
A4	B4	C4	D4	E4	F4	G4	H4
A5	B5	C5	D5	E5	F5	G5	H5
A6	B6	C6	D6	E6	F6	G6	H6
A7	B7	C7	D7	E7	F7	G7	H7
A8	B8	C8	D8	E8	F8	G8	H8

Figure 7. Secondary disks are partitioned into 8 groups with 1 disk in a group as in RAID1

i is said to be overloaded if its load L_i is greater than the threshold load L^T . The overload factor of primary disk i is defined as $\rho_i = (L_i - L^T)/L^T$. We will be concerned with the case where $0 \leq \rho_i \leq 1$, which corresponds to the maximum load possible on a disk without exceeding the threshold of the baseline RAID 1 system. The system load is represented by an *overload vector* $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$ where λ_i equals 1 if $\rho_i > 0$ and 0 otherwise. Let ν be the number of overloaded disks. Note that this is the number of 1's in the overload vector Λ .

A primary group is said to be an *overloaded group* if one or more of its disks is overloaded. Let $\Omega_i, i = 1, 2, \dots$ denote the i^{th} overloaded group induced by Λ , and let Ω'_i denote

the corresponding secondary group. An overloaded group Ω_i will require one or more disks in Ω'_i to be activated to keep the load on the primary disks below L^T ; however all disks in a secondary group can remain in the low power (idle) state if the corresponding primary group is not overloaded.

As an example suppose $n = 8, g = 4$ as in Figure 6 and assume that $\nu = 3$. If $\Lambda = [11001000]$ then Ω_1 is the group $\{1, 2\}$ and Ω_2 is the group $\{5, 6\}$. The other groups are not overloaded. On the other hand if $\Lambda = [11001001]$ then there are three overloaded groups $\Omega_1 = \{1, 2\}, \Omega_2 = \{5, 6\}$ and $\Omega_3 = \{7, 8\}$.

For a given value of $\nu, 1 \leq \nu \leq n$, where $n = sg$, there are $\binom{n}{\nu}$ distinct overload vectors Λ . We partition these vectors into equivalence classes ${}^s\mathcal{C}_\nu^g(1), {}^s\mathcal{C}_\nu^g(2), \dots$ where ${}^s\mathcal{C}_\nu^g(k)$ consists of the vectors that induce exactly k overloaded groups. Note that s is the size of a group, g is the number of groups, and n is the number of primary disks.

Observation 1: Consider a system of n primary disks, g groups and group-size s , where $n = sg$. Suppose that ν of the primary disks are overloaded. Assuming that every primary disk is equally likely to be overloaded, the probability that there are k overloaded groups is given by: $|{}^s\mathcal{C}_\nu^g(k)| / \sum_j |{}^s\mathcal{C}_\nu^g(j)|$.

As an example let $n = 8, g = 4$ and $\nu = 3$. In this configuration the groups are of size $s = 2$ and hence an overloaded group has either 1 or 2 overloaded disks. There are $\binom{8}{3} = 56$ possible overload vectors. Since there are 3 overloaded disks and 2 disks per group there must be either 2 or 3 overloaded groups. That is ${}^2\mathcal{C}_3^4(1) = \Phi$ and ${}^2\mathcal{C}_3^4(k) = \Phi$ for $k > 3$. To find $|{}^2\mathcal{C}_3^4(2)|$ note that having two overloaded groups requires 1 group (out of the 4) to have 2 overloaded disks and another group to have one: this can be done in $\binom{4}{1} \times \binom{3}{1} \times \binom{2}{1} = 24$ ways. The first term chooses the the group with 2 overloaded disks, the second chooses the group with 1 overloaded disk among the remaining 3 groups, and the last term is the number of ways to choose the overloaded disk from the 2 disks in the group. Similarly, $|{}^2\mathcal{C}_3^4(3)| = \binom{4}{3} \times \binom{2}{1} \times \binom{2}{1} \times \binom{2}{1} = 32$.

Observation 2: Given n, s, g and ν , let $\mathcal{R}_{s,g,\nu}$ be a random variable equal to the number of overloaded groups. Then its expected value, $E(\mathcal{R}_{s,g,\nu})$ is given by:

$$E(\mathcal{R}_{s,g,\nu}) = \sum_{1 \leq k \leq g} k \times \frac{|{}^s\mathcal{C}_\nu^g(k)|}{\sum_j |{}^s\mathcal{C}_\nu^g(j)|} \quad (1)$$

The computation of ${}^s\mathcal{C}_\nu^g(k)$ can be expressed by the following recursive equation.

Lemma :

$$|{}^s\mathcal{C}_\nu^g(k)| = \binom{g}{k} \left[\binom{ks}{\nu} - \sum_{1 < j < k} |{}^s\mathcal{C}_\nu^g(j)| \right] \quad (2)$$

Proof: There are $\binom{g}{k}$ ways of choosing k overloaded groups from the g possible groups. The total number of disks in these k groups is $s \times k$, and among them there are ν overloaded disks. Thus there are $\binom{s \times k}{\nu}$ different ways to select these overloaded disks. However, the product of these two parts includes the cases when the number of overloaded groups j is smaller than k ; j ranges from 1 to $k - 1$. Hence we need to subtract these doubly counted portions as shown in the expression.

The number of disks in the secondary group of Ω_i that must be activated can be computed as follows. The analysis assumes that requests to a primary group disk j in Ω_i are uniformly spread among the disks of the corresponding secondary group. For instance if requests to disk j are to random locations on the primary disk in Ω_i then the requests would be randomly distributed among the secondary disks in Ω'_i . On the other hand, typical workloads exhibit considerable spatial locality in accessing objects (*e.g.* an entire file object is accessed); in this case since the objects are striped across the disks in the secondary group, we once again have load balance among these disks.

Lemma: Let $\rho^i = \max_{j \in \Omega_i} \rho_j$ be the maximum overload factor among all the disks in Ω_i . Then the maximum number of disks in Ω'_i that must be activated is $\lceil s \rho^i \rceil$.

Proof: Consider a disk $j \in \Omega_i$ with load factor ρ_j . By definition of load factor, the load L_i on disk j equals $L^T(1 + \rho_j)$. The load from disk j that is transferred to the s disks in Ω'_i is therefore $L^T \rho_j$. At most the s disks belonging to Ω_i can contribute to the load on the disks in Ω'_i . Hence the total load on the disks in Ω'_i is bounded by $\sum_{j \in \Omega_i} L^T \rho_j \leq s L^T \rho^i$. Since each disk in Ω'_i can handle a load of L^T , the maximum number of secondary disks that need to be spun up is: $\lceil s L^T \rho^i / L^T \rceil = \lceil s \rho^i \rceil$.

Finally, we upper bound the expected number of redundant disks that need to be activated.

Theorem: Let $\rho = \max_{\Omega_i} \rho^i$ be the maximum load on any primary disk. Then the expected number of secondary disks that need to be activated is upper bounded by:

$$Spin_up = \lceil \rho \times s \rceil \times E(\mathcal{R}_{s,g,\nu}) \quad (3)$$

Using equations (1) and (2) to compute $E(\mathcal{R}_{s,g,\nu})$ we compute the upper bound on the expected number of secondary disks to be activated using equation (3), for different values of s , g and ν . For example, if $s = 2$, $\nu = 2$ and $g = 4$, then there are two possible configurations: one where there is one overloaded group and one where there are 2 overloaded groups.

There are 4 configurations with one overloaded group: $\Omega_1 = \{1, 2\}$, $\Omega_1 = \{3, 4\}$, $\Omega_1 = \{5, 6\}$ and $\Omega_1 = \{7, 8\}$. There are several configurations with two overloaded groups. For instance, if $P2$ and $P5$ are overloaded disks, then groups $\Omega_1 = \{1, 2\}$ and $\Omega_2 = \{5, 6\}$ are overloaded groups. The total number of such configurations is:

$\binom{4}{2} \times 2 \times 2 = 24$. So, the expected number of overloaded groups is: $1 \times \frac{4}{4+24} + 2 \times \frac{24}{4+24} = 1.857$. Using this we can get the expected number of secondary disks that needed to be activated based on the overload factor.

DiskGroup provides an analytic model to compute and compare the average number of secondary disks to be powered up, based on the number of overloaded disks and the load factor, for different groups sizes g . For a given number of total primary disks n , the number of groups g is successively chosen as powers of 2, (1, 2, 4 and so on until up to n). That is, DiskGroup chooses $(\log n + 1)$ sizes of g for a fixed n . For a given g and ρ , we can compute the expected value of *Spin_up* as shown in the Theorem.

4. Evaluation of DiskGroup

In order to evaluate our DiskGroup scheme, we simulate a simple storage system assuming data on a primary disk is striped across the secondary disks in the group. At the same time, the load on a primary disk is simulated as random block numbers uniformly distributed across the blocks on the disk.

In our simulation, there are 16 disks in total with 8 primary and 8 secondary disks. For each disk, there are 16000 data blocks, which are stored sequentially. For different group configurations, the layout of secondary disks is shown in Figures 4 to 7.

The simulation varies three parameters for the DiskGroup system, the number of groups, the number of overloaded disks and the amount of overload for each disk. The simulator accepts these three parameters as input, and then simulates the disk access process, then outputs the number of disks that have to be spun up. The experiment evaluates three group sizes: 1, 2, and 4. We varied the number of overloaded disks from 1 to 8, while varying the overload factor from 1/8 to 7/8. We compared the simulation results with the theoretical results from the analytic model of Section 3.4. The results showed that the simulation and analysis are in very close agreement, indicating the high accuracy of the analytical DiskGroup model.

Figures 8 to 14 show the performance comparison of different group configuration in DiskGroup and RAID1, when the overload factor varies from 1/8 to 7/8 and the number of overloaded disks varies from 1 to 8. When the size of the group is 8, the configuration is the same as the mirrored RAID1 system. DiskGroup will consider all the group configurations and calculate the average number of disks spun up for each configuration.

When the overload factor is 1/8 (see Figure 8), DiskGroup only spins up one disk no matter how many disks are overloaded. However, the number of disks that RAID1 needs to spin up increases linearly with the number

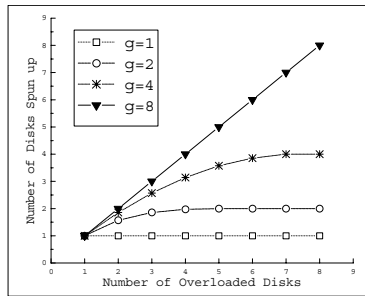


Figure 8. DiskGroup and RAID1, when workload is $1.125L^T$

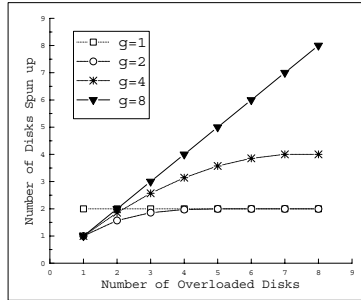


Figure 9. DiskGroup and RAID1, when workload is $1.25L^T$

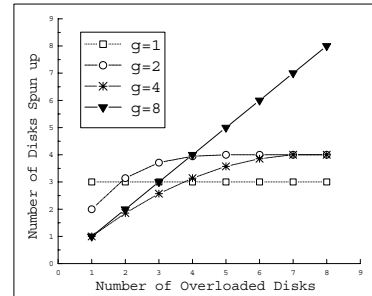


Figure 10. DiskGroup and RAID1, when workload is $1.375L^T$

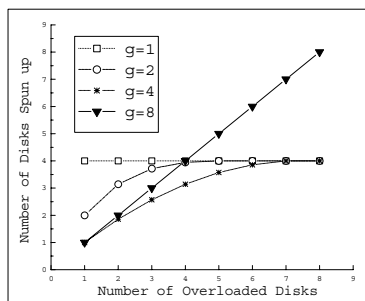


Figure 11. DiskGroup and RAID1, when workload is $1.50L^T$

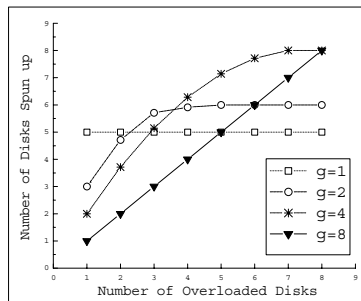


Figure 12. DiskGroup and RAID1, when workload is $1.625L^T$

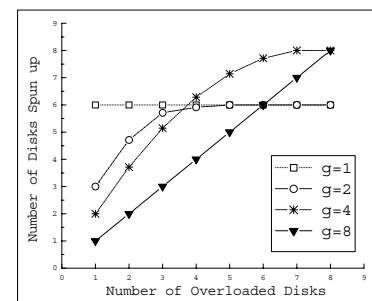


Figure 13. DiskGroup and RAID1, when workload is $1.75L^T$

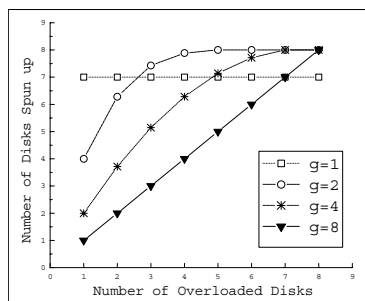


Figure 14. DiskGroup and RAID1, when workload is $1.875L^T$

of overloaded disks. Obviously, DiskGroup is much more energy efficient, spinning up only between 12.5% to 50% of the disks. In this configuration, we notice that since the workload is very light, one secondary disk can serve the load from several overloaded primary disks.

When the overload factor grows to $1/4$, the number of

disks spun up is shown in Figure 9. DiskGroup can save energy¹ from 21.5% to 75% compared with RAID1 if the configuration consists of two groups. In this configuration, DiskGroup still can save significant energy, because the load is still not very high, which leaves chances for DiskGroup to optimize the energy consumption.

As the amount of workload increases, DiskGroup has fewer chances to save energy, because one secondary disk will serve the extra workload from fewer primary disks. In Figure 10, the energy savings rate varies from 7.15% to 62.5%. When the number of overloaded disks is below 4, $g = 4$ will result in the best energy savings. After that, $g = 1$ is the best. In Figure 11, the energy saving rate varies from 7.15% to 50%, and $g = 4$ is always the best. When the workload grows up to $1.625L^T$, and the number of overloaded disks is more than 6, there are no energy savings beyond that obtained by RAID1. After that, DiskGroup saves energy from 16.7% to 37.5%. The same holds for Figure 13 and Figure 14, which respectively show energy savings from 14.3% to 25% for 7 or more overloaded disks,

¹The energy to spin a disk down and up is ignored in these estimates.

and 12.5% with 8 overloaded disks.

As shown above, when the load is light, and many primary disks are overloaded, DiskGroup can save significant amounts of energy by spinning up much fewer number of disks compared with RAID1, up to 87.5%. When the load is very high, there are fewer opportunities for DiskGroup (or any dynamic power management scheme) to save much energy without sacrificing performance.

5. Conclusion and Future Work

In this paper, we introduce a novel energy efficient disk layout (DiskGroup) for conventional RAID1 systems. DiskGroup partitions the disks into several groups, where each group has its own primary disks and secondary disks. The data blocks of each primary disk are spread across evenly all the secondary disks in that group. We developed an analytic framework that enables us to compare the energy savings potential of different configurations as a function of the workload. Theoretical analysis and simulation shows that DiskGroup can spin up a fewer number of secondary disks when primary disks are overloaded to save energy.

In the future, we would like to enhance our data layout model for combining different group configurations into a super group, which contains several layouts. Furthermore, we are using the simulation tool DiskSim [1] to test our scheme with real workloads.

References

- [1] J. Bucy and G. Ganger. The DiskSim Simulation Environment Version 3.0. <http://www.pdl.cmu.edu/DiskSim/>, 2003.
- [2] E. Carrera, E. Pinheiro, and R. Bianchini. Conserving Disk Energy in Network Servers. In *Proceedings of the 17th International Conference on Supercomputing*, 2003.
- [3] P. M. Chen, E. L. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson. RAID: High-Performance, Reliable Secondary Storage. *ACM Computing Surveys*, 26, 1994.
- [4] D. Colarelli and D. Grunwald. Massive Arrays of Idle Disks For Storage Archives. In *Proceedings of the 15th High Performance Networking and Computing Conference*, 2002.
- [5] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPMs: Dynamic Speed Control for Power Management in Server Class Disks. In *Proceedings of the International Symposium on Computer Architecture*, 2003.
- [6] S. Gurumurthi, A. Sivasubramaniam, and V. Natarajan. Disk Drive Roadmap from the Thermal Perspective: A Case for Dynamic Thermal Management. In *Proceedings of the International Symposium on Computer Architecture*, 2005.
- [7] H.-I. Hsiao and D. J. DeWitt. Chained Declustering: A New Availability Strategy for Multiprocessor Database Machines. In *Proceedings of the IEEE International Conference on Data Engineering*, 1990.
- [8] H. Huang, W. Hung, and K. Shin. FS2: Dynamic Data Replication in Free Disk Space for Improving Disk Performance and Energy Consumption. In *proceedings of the 20th ACM Symposium on Operating Systems Principles(SOSP)*, 2005.
- [9] D. Li and J. Wang. eRAID: A Queuing Model Based Energy Conservation Technique. *14th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2006.
- [10] E. B. Nightingale and J. Flinn. Energy-efficiency and Storage Flexibility in the Blue File System. In *Proceedings of the 6th Symposium on Operating Systems Design and Implementation(OSDI)*, 2004.
- [11] D. Patterson, G. A. Gibson, and R. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1988.
- [12] E. Pinheiro and R. Bianchini. Energy Conservation Techniques for Disk Array-Based Servers. In *Proceedings of the 18th International Conference on Supercomputing*, 2004.
- [13] E. Pinheiro, R. Bianchini, and C. Dubnichi. Exploiting Redundancy to Conserve Energy in Storage System. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2006.
- [14] C. Weddle, M. Oldham, J. Qian, A. A. Wang, P. Reiher, and G. Kuenning. PARAID: The Gear-Shifting Power-Aware RAID. In *USENIX Conference Proceedings on File And Storage Technologies (FAST)*, 2007.
- [15] X. Yao and J. Wang. RIMAC: A Novel Redundancy-Based Hierarchical I/O Cache Architecture for Energy Efficient, High Performance Storage Systems. In *Proceedings of the 1st ACM EuroSys Conference (EuroSys)*, 2006.
- [16] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wikes. Hibernator: Helping Disk Arrays Sleep through the Winter. In *proceedings of the 20th ACM Symposium on Operating Systems Principles(SOSP)*, 2005.
- [17] Q. Zhu, F. M. David, Y. Zhou, C. F. Devaraj, P. Cao, and Z. Li. Reducing Energy Consumption of Disk Storage Using Power-Aware Cache Management. In *Proceedings of the 10th International Symposium on High-Performance Computer Architecture(HPCA)*, 2004.
- [18] Q. Zhu, A. Shankar, and Y. Zhou. PB-LRU: A Self-Tuning Power Aware Storage Cache Replacement Algorithm for Conserving Disk Energy. In *Proceedings of the 18th International Conference on Supercomputing*, 2004.