

Conserving Energy in Conventional Disk based RAID Systems

Dong Li, Jun Wang

Computer Science and Engineering Department
University of Nebraska Lincoln, Lincoln, NE 68588

Peter Varman

Department of Electrical and Computer Engineering
Rice University, Houston, TX 77005

Abstract—Energy-efficiency is becoming increasingly important for storage systems to reduce the total cost of ownership (TCO). In this paper, we propose an energy saving policy named eRAID for conventional disk based RAID-1 systems using redundancy. In particular, we develop a dynamic performance control scheme with the help of a performance predictor based on queueing network theory. Moreover, we discuss alternative data layout schemes that are more energy-efficient than traditional disk mirroring. Experimental results show that eRAID can save up to 30% energy without violating predefined performance constraints.

I. INTRODUCTION

Energy-efficiency has become AN extremely important consideration for storage systems due to the large energy-related portion in its total cost of ownership (TCO). Because of the intensity of workloads, simple disk spin down/up policy is considered infeasible in an I/O intensive environment. Most current energy saving policies for server side storage systems resort to multi-speed disks [1], [2], [3], [4]. Since current storage systems are still built with conventional disks, it is important to provide a workable solution for conventional-disk-based storage systems. The most fundamental method to save energy in a conventional disk based system is to put a disk in standby state, in which

the disk cannot serve requests. Since request inter-arrival time of server side workloads are usually too short to supply long idle periods, such periods have to be artificially created. A possible way to create long idle disk periods is to unbalance the disk workload within the storage system.

There are two approaches to unbalance workloads for conventional disk based systems to save energy: *relocating data* and *redirecting requests*. The common feature of relocating data is to intentionally unbalance disk loads by migrating data across disks according to the changing access patterns. However, the representative works of this approach, MAID [5] and PDC [6], showed that they can save energy only when the load on the server is extremely low. The second approach, *redirecting requests*, intentionally bypasses a data target (e.g. a standby disk) by redirecting requests to other data target(s). The precondition is that the alternative data target(s) can provide the same information to users. Given their inherent redundant encoding of information, RAID systems satisfy this precondition well. Our group is among the first to exploit this redundancy to save energy in both RAID-1 and RAID-5 based systems in literature [2] using multi-speed disks. We also developed a request transformation scheme for conventional disk based RAID-5 systems [7]. The impact of redirecting or transforming re-

This work is supported in part by the NSF CSR Award CNS-0509480, CPA Award CCF-0429995, Department of Energy ECPI Award DE-FG02-05ER25687 and NSF Award CCF-0105565.

quests on performance has however, not been adequately studied.

In this paper, we develop an energy saving policy named eRAID using request redirecting for conventional disk based RAID-1 systems. We also discuss an alternative data layout scheme to further save energy. Experimental results show that eRAID can save up to 30% energy without violating predefined performance constraints. Compared with previous server-side energy saving policies, eRAID has the following salient advantages: (1) it is a soft solution so that it does not require current storage systems to add extra hardware (e.g. cache disks) or to replace all their conventional disks with multi-speed disks; (2) it does trade-off between energy-saving and performance degradation by taking both performance metrics — average response time and throughput into consideration.

II. MOTIVATION

There are three major limitations in current server-side disk energy management policies.

- No workable solutions on conventional disk based systems.
- Single performance measurement: Current disk power management algorithms trade-off energy saving for a single performance metric, either throughput or response time. However, energy-saving policy may degrade both, and the degradation of both metrics are not always proportional.
- No differentiation of workload time criticality: based on time criticality, I/O workloads can be roughly divided into two classes: synchronous load and asynchronous. Since both kinds of workloads have different time criticality, performance variations impact them differently.

The aforementioned observations motivate us to develop an energy-saving policy for conventional disk based RAID system.

- With the help of redundant information, it is possible to generate long idle periods for server disks. In RAID-1, by redirecting read requests to primary disks and deferring write update in a controller cache

(NVRAM), the idle period of the mirror disks can be dramatically stretched.

- Server disks usually have spare service capacity most of the time, and system workloads vary over time [8]. Therefore, redirecting requests of the mirror disks to primary disks does not necessarily result in a large performance loss.
- Queueing model is a widely adopted method to model RAIDs for performance analysis and prediction. It can be used to estimate the throughput and response.

III. ERAID DESIGN

The main idea of eRAID is to spin down, partially or in entirety, the mirror group to standby for energy conservation based on workload changes. Read requests are served by the data copies in the primary group, and write requests to the standby disks are held in the controller cache or on active disks and flushed to the standby disks after they are spun up. It may be noted that using redundancy information to spin down disks to save energy can be applied to other RAID organizations like RAID-5. More details can be found in our technical report [7].

eRAID needs to strike a balance between energy savings and meeting performance based service level agreements (SLA) by ensuring that a certain number of disks are active. eRAID employs a time-window based approach to perform the control. At the beginning of each time window, we do performance/energy prediction to find out 1) how many disks can be spun down to save energy and 2) how many disks should be spun up in order not to violate predefined constraints. The problem is formalized below.

$$\text{Object: maximize } S_E = \frac{E_{base} - E_{eRAID}}{E_{base}}$$

$$\text{Subject to } \left\{ \begin{array}{l} \text{System Throughput Constraint:} \\ D_T = \frac{|T_{eRAID} - T_{base}|}{T_{base}} \leq Limit_T \\ \text{Response Time Constraint:} \\ D_X = \frac{|X_{eRAID} - X_{base}|}{X_{base}} \leq Limit_X \end{array} \right.$$

Here E , T and X denote energy consumption, mean response time and mean throughput.

”base” represents the baseline system that employs no energy-efficiency policy. $Limit_T$ and $Limit_X$ are user-defined performance degradation parameters for mean response time and mean throughput respectively. Next, we show how we address this problem.

A. Solving for Energy Saving S_E

We develop a power model for multi-disk systems to estimate the energy saving S_E . Since we mainly focus on the disk energy consumption, other components (e.g. CPU and cache) in RAID systems are not examined here. A single disk power model considering the spin down/up policy has been extensively studied. However, little research work has been conducted on modeling energy for multi-disk systems. Disk spin down/up schemes are considered an impractical method to save energy, due to the short idle periods typically encountered in server workloads. However, as we discussed in Section II, with the help of redundancy in RAIDs, we have the full control over the disk idle period distribution, by trading energy efficiency against a small performance penalty.

Given a time-window with length T and an N -disk RAID-1 system, assuming there are R requests being served and the average service time is t . We have the system utilization $\rho_1 = \frac{R*t}{N*T}$. Let P_a, P_i, P_s and P_w denote the power consumption of a disk in active, idle, standby, and mode-switching states respectively. Let T_s and T_w denote the time that the disk spends in standby and mode-switching respectively. To get the upper bound of energy saving, we do not consider energy consumption for data coherence from writes. The energy consumption of baseline system is $E_{base} = E_{active} + E_{idle} = P_aNT\rho_1 + P_iNT(1 - \rho_1)$.

For asynchronous loads, the request arrival rate is independent of the number of active disks, while for synchronous loads, the request arrival rate could be reduced because fewer active disks provide smaller system throughput. That is, in synchronous loads, the number of requests in T may be less than R . We use ρ_2 ($\rho_2 \leq \rho_1$) to denote the new utilization.

Suppose i disks spin down to standby state at the beginning of T , and spin up to active state at the end of T . The energy consumption of the i disks is $i(P_sT_s + P_wT_w)$. The energy consumption of the $(N - i)$ disks is $P_aNT\rho_2 + P_i[(N - i)T - NT\rho_2]$. Then the new energy consumption can be described as: $E_{eRAID} = i(P_sT_s + P_wT_w) + P_aNT\rho_2 + P_i[(N - i)T - NT\rho_2]$.

Then the percentage of energy saving, S_E , in the period T can be expressed as:

$$S_E = \frac{(E_{base} - E_{eRAID})/E_{base}}{P_i + (P_a - P_i)\rho_1} = \frac{(P_a - P_i)(\rho_1 - \rho_2) + \frac{i}{N}(P_i - \frac{P_sT_s + P_wT_w}{T})}{P_i + (P_a - P_i)\rho_1}$$

Here ρ_1, ρ_2 can be resolved by a performance prediction technique in the following section. Taking IBM 36Z15 [8] disk parameter as inputs, and assuming $T \gg T_w$, we draw Figure 1 to show how the energy saving is impacted by i and R_{si} .

The number of spun down disks is the dominant factor for energy savings. We then remove the assumption $T \gg T_w$ and set $T = kT_w, k > 1$. The right picture of Figure 1 shows that, when k is larger than six, the energy saving is near that of the ideal condition ($T \gg T_w$).

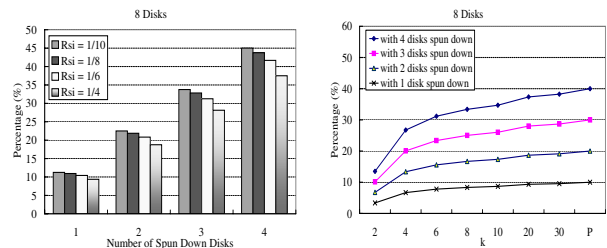


Fig. 1. The Impact of $R_{si} = \frac{P_s}{P_i}$ and k on Energy Saving.

B. Solving for D_X and D_T

We use queueing models to calculate the performance degradation factors D_X and D_T . Our basic approach can be described as follows: (1) use queueing models to model RAID-1 (2) examine how the input parameters are changed after spinning down some mirror disks (3) calculate new performance numbers (4)

compare original and new results for decision-making. We assume the workload changes little between the current time window and the next window. This assumption is reasonable in a short-term while workloads do have significant change in a long-term manner.

Since read and write operation in RAID-1 systems have different features, they are modelled differently. Based on time criticality, we use different model for synchronous and asynchronous load. To isolate the difference, we examine the performance impact of spinning down mirror disks under synchronous read (SR), asynchronous read (AR), asynchronous write (SW) and asynchronous write (AW) loads individually.

In this paper, we choose two queueing models introduced by Varki et al. in the literature [9] to model RAID-1 system with synchronous read and synchronous write loads. Based on both models, we develop another two queueing models for asynchronous read and asynchronous write loads. Our queueing models are built based on the main features of a real RAID system, HP SureStore E Disk Array FC60 [10].

1) *Read load*: The left diagram of Figure 2 shows the model of a RAID-1 system with a synchronous read load using a closed queueing network. It is assumed there are M processes in this system while each generating an I/O stream (array read requests). Array read requests are first submitted to controller cache. In case of cache misses they are directed to the disks. After the requests are finished, they are returned to the processes. Only after the previous request is returned does a process issue another request following a further process-delay time.

All the processes are modelled as a delay server. The average performance measures of this closed queueing network can be computed by the Mean Value Analysis (MVA) technique [11]. For asynchronous loads, the RAID system is modeled as an open queue network shown at the right of Figure 2. Since the load balance policy is not considered here, the disk

array is treated as a bunch of M/M/1 queue nodes. Table I shows the model input parameters. The disks are named from subsystem 1 to subsystem N . The cache and RAID controller are named as subsystem 0. Note that $P_0 = 1$ and $\sum_{i=1}^N P_i = 1 - \text{cache_hit_rate}$.

TABLE I
READ LOAD MODEL INPUT PARAMETERS

Para.	Description
Common Parameters ($0 \leq i \leq N$)	
N	number of all disks in RAID-1
μ_i	service rate of subsystem i
P_i	access probability of subsystem i
Synchronous Read Model	
M	the number of processes
O_p	mean process delay
Asynchronous Read Model	
λ	mean request arrival rate

After some mirror disks are spun down to save energy, two input parameters of the queueing model could be affected: disk access probability and disk service time. When i mirror disks are spun down, the load of the mirror disks will be directed to their primary disks. Access probabilities of other disks are not affected. Disk service time depends on a large number of factors. We measured the service time of three physical disks directly and found the disk service time (for both read and write operations) has little difference when the disk queue length is less than three¹.

Based on above discussion, we can compute the performance measures for both the base system and eRAID. For ease of presentation, given an N -disk RAID-1, we define disk 1 to $N/2$ to be mirror disks and disk $(N/2+1)$ to N to be primary disks. Let disk n and disk v be a disk group (mirror disk and primary disk), with $v=N/2-n+1$. For synchronous read load, we use $T_i(M)$ and $T'_i(M)$ to denote the mean response time of subsystem i having M requests in the baseline system and eRAID

¹We only spin down mirror disks when the system utilization is low. Even when the disk utilization is 35%, which means the queue length is $\rho/(1-\rho) = 35\%/(1-35\%) \approx 0.54$ according to queueing theory, doubling the disk load would increase the queueing length to $70\%/(1-70\%) \approx 2.33$.

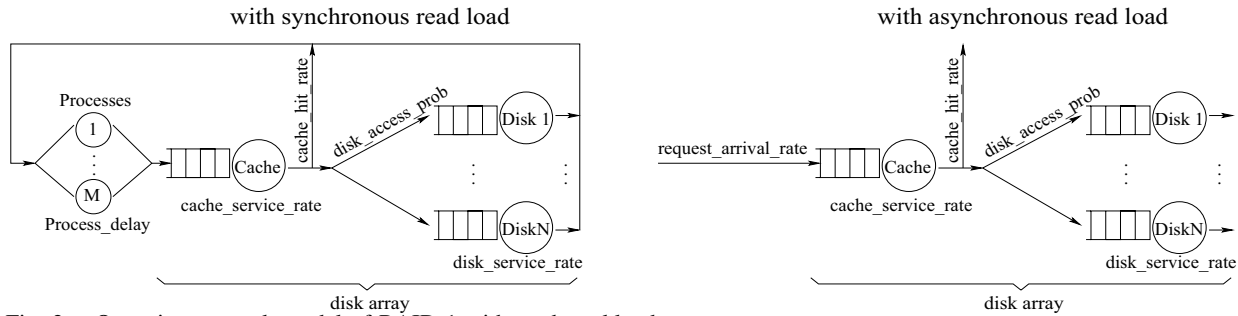


Fig. 2. Queuing network model of RAID-1 with read workloads

respectively. $T_i(M)$ and $T'_i(M)$ can be easily computed by MVA technique [11], [12]. The final mean response time of the disk array and mean throughput of the baseline system can be expressed as: $T_{base}(SR) = \sum_{n=0}^N [T_n(M)P_n]$ and $X_{base}(SR) = \frac{M}{T_{base}(SR)+O_P}$. The performance measures of eRAID are: $T_{eRAID}(SR) = T_0(M) + \sum_{n=i+1}^{N-i} [T'_n(M)P_n] + \sum_{n=N-i+1}^N [T'_n(M)(P_n + P_v)]$ and $X_{eRAID}(SR) = \frac{M}{T_{eRAID}(SR)+O_P}$.

For asynchronous read loads, as discussed in Section II, system throughput is not affected if there no disks are saturated after spinning down the mirror disks. The mean response time is computed as *mean_cache_response_time* + *mean_disk_response_time*. That is,

$$T_{base}(AR) = \frac{1}{\mu_0 - \lambda} + \sum_{n=1}^N \frac{P_n}{\mu_n - \lambda P_n} = \sum_{n=0}^N \frac{P_n}{\mu_n - \lambda P_n}.$$

After spinning down mirror disks 1 to i , the mean response time of eRAID is $T_{eRAID}(AR) = \frac{1}{\mu_0 - \lambda} + \sum_{n=i+1}^{N-i} \frac{P_n}{\mu_n - \lambda P_n} + \sum_{n=N-i+1}^N \frac{P_n + P_v}{\mu_n - \lambda(P_n + P_v)}$.

2) *Write load*: Modern disk arrays usually implement write-back caching in the array controller. In this case, unlike reads, a write request is completed once the data is written to cache. Dirty data are flushed back to disks according to cache replacement policies. FC-60 uses a two-threshold write-back policy [10], [9], *destage_threshold* and *max_dirty_blocks*. Destaging operations starts if the number of saved dirty blocks is larger than *destage_threshold*. The maximum dirty blocks that can be held in the cache is determined by *max_dirty_blocks*. For write workloads, the RAID system can be modeled as a Markov birth-death M/M/1/K process with $K = (\text{max_dirty_blocks} - \text{destage_threshold} + 1)$

[9]. Practically, K is the maximum number of requests that can be held in cache. Thus, the disk array can be modeled by the cache alone with input parameters listed in Table II.

 TABLE II
 WRITE LOAD MODEL INPUT PARAMETERS

Para.	Description
P_{miss}	array cache write miss probability
K	maximum queue length
d_λ	arrival rate of dirty blocks to cache
d_μ	rate at which dirty blocks are written from cache to disks
μ	mean disk service rate
Synchronous Write Model	
χ	maximum array throughput

Since all data must be written to both mirror and primary disks, an N -disk RAID-1 can serve $N/2$ requests at the same time. Thus the service rate d_μ is $\frac{N}{2\mu}$ [9]. In the synchronous write model, χ denotes the maximum array throughput, which is the array's throughput when the disk array has an infinite cache, and *chi* be computed by MVA technique. The arrival rate of dirty blocks is different for synchronous and asynchronous write loads. For a synchronous write load, d_λ can be approximated by $\chi * P_{miss}$, while for asynchronous loads, $d_\lambda = \lambda * P_{miss}$ where λ can be directly measured.

In current design of eRAID, we use NVRAM to save inconsistent data. By spinning down i disks, the service rate d_μ is $\frac{N-2i}{2\mu}$. Another two input parameters may be affected when some of the mirror disks are spun down: K and P_{miss} . Since deferring the write of dirty blocks of sleeping disks in the cache for a longer period may increase the *destage_threshold*, the queue length K may be

decreased. For the other parameter P_{miss} , by restricting the amount of accumulated dirty blocks of sleeping disks (e.g. within the second half of LRU table), the cache pollution problem can be ignored. Through experiments we found eRAID made little impact on P_{miss} .

For synchronous write load, the mean throughput of the disk array is computed by $\chi * (1 - P_{max_dirty})$. Here P_{max_dirty} is the steady state probability when the cache has number of maximum dirty blocks. According to M/M/1/K queueing model [11], $P_{max_dirty} = \frac{(1-a)*a^K}{1-a^{K+1}}$ with $a = \frac{d_\lambda}{d_\mu} = \frac{2\mu\chi P_{miss}}{N}$. That is, $X_{base}(SW) = \chi * (1 - \frac{(1-a)*a^K}{1-a^{K+1}})$, $a = \frac{2\mu\chi P_{miss}}{N}$. The mean response time can be computed using Little's Law on the entire system $T_{base}(SW) = \frac{M}{X_{base}(SW)} - O_p$. $X_{eRAID}(SW)$ and $T_{eRAID}(SW)$ can be computed in the same way as computing $X_{eRAID}(SW)$ and $T_{eRAID}(SW)$ but replacing K by K' and $a = \frac{2\mu\lambda P_{miss}}{N-2i}$. Here K' is the maximum queue length in the queueing model of eRAID.

For asynchronous load, the mean response time is $\frac{\bar{k}}{d_\lambda}$ according to M/M/1/K model. Here \bar{k} is the average queue length and $\bar{k} = \frac{a}{1-a} - \frac{(K+1)a^{K+1}}{1-a^{K+1}}$ with $a = \frac{d_\lambda}{d_\mu} = \frac{2\mu\lambda P_{miss}}{N} < 1$. That is, $T_{base}(AW) = \frac{a}{1-a} - \frac{(K+1)a^{K+1}}{1-a^{K+1}}$, $a = \frac{2\mu\lambda P_{miss}}{N}$. In the same way, $T_{eRAID}(AW)$ can be computed via replacing K by K' and $a = \frac{2\mu\lambda P_{miss}}{N-2i}$.

C. Control Algorithm

The control algorithm is used to automatically adjust the number of spun-down disks to tradeoff energy-efficiency and performance. Disk mode switch decision is made at the beginning of each time window. Since solving the multi-constraint (S_E , D_T and D_X) problem involves much computing overhead, we use a simple but effective method: giving higher priority to less frequently used (LFU) disks to spin down. Alg.1 shows how to find the mirror disks to spin down. Experiment shows that Alg.1 can give the optimal solution for most cases. The conservative control algorithm is summarized as Alg.2, which gets executed at the beginning of each time-window. For write

load, once there are mirror disks spun down, their data updates are expected to be deferred in cache. The new destage_threshold in eRAID is set based on: $\frac{destage_threshold*cache_size}{data_write_rate_of_sleepdisks} > 6T_w$.

Alg.1: find mirror disks to spin down

```

1: Put active mirror disks into set S;
2: Order all mirror disks according to LFU
3: for( $i = 1; i \leq N/2; i++$ ) {
4:   Predict  $D_X$  and  $D_T$  of spinning down
   first  $i$  disks of S;
5:   If any constraint is violated, the first
    $i - 1$  disks in S are disks to spin down;
6: }
```

Alg.2: conservative control algorithm

```

1: Get real  $D_X$  and  $D_T$  of last time-window;
2: if(real degradation  $\leq$  limitation) {
3:   spin down the disks found by Alg. 1;
4: }else {
5:   spin up all sleep disks;
5.1: update all deferred dirty data of
   sleep disks (for write load);
6: }
```

IV. EVALUATION

We evaluated our policy by using trace-driven simulations. IBM Ultrastar 36Z15 is chosen as the disk power model. We generate synchronous read (SR) load and synchronous write (SW) load from the real trace Cello99² with the assumption that 30% of all requests are synchronous requests. To make Cello99 trace intensive enough for current hardware conditions, we merged its twenty-disk load into an eight-disk load. Asynchronous read (AR) load and asynchronous write (AW) load are derived from another real trace, TPC-C20³. The time-window size is set to 10 minutes. We used DiskSim [13] to configure an 8-disk RAID-1 system based on Hewlett-Packard SureStore E FC-60 disk array [10]. Last, we augmented DiskSim with the disk power model and our energy-efficient policy.

Table III shows the preliminary results of eRAID for two scenarios. It can be seen that, the performance control algorithm is effective in both Case I and II and performance degradation constraints, $Limit_T$ and $Limit_X$, are not

²http://tesla.hpl.hp.com/public_software

³<http://traces.byu.edu/new/Tools>

$f = 2/3$

D1	D2	D3	D4	D5
P1	Q1	R1	S1	T1
P2	Q2	R2	S2	T2
P3	Q3	R3	S3	T3
P4	Q4	R4	S4	T4
P5	Q5	R5	S5	T5
P6	Q6	R6	S6	T6

M1	M2	M3	M4	M5
P1	P2	P3	Q1	Q2
P4	P5	P6	Q4	Q5
Q3	R1	R2	R3	S1
Q6	R4	R5	R6	S4
S2	S3	T1	T2	T3
S5	S6	T4	T5	T6

Fig. 3 A new Data Layout Scheme

	Mirroring	This scheme	Probability
N	Rm	Rv	P
0	0	0	$(1-p)^5$
1	1	1	$5p(1-p)^4$
2	2	1	$10p^2(1-p)^3$
3	3	2	$10p^3(1-p)^2$
4	4	2	$5p^4(1-p)$
5	5	3	p^5

Table IV(a)

p	AVE(Rm)	AVE(Rv)	AVE(Rm) / AVE(Rv)
1/3	1.67	1.08	1.54
1/2	2.50	1.50	1.67
2/3	3.33	1.92	1.74
4/5	4.00	1.86	2.15
5/6	4.17	1.88	1.74
1	5.00	3.00	1.67

Table IV(b)

 TABLE III
 Overall Results

CASE I: $Limit_T \leq 10\%$ & $Limit_X \leq 3\%$			
Load	overall D_T	overall D_X	S_E
AR	7.5%	0.0%	10.2%
SR	7.1%	1.5%	11.8%
AW	4.3%	0.0%	13.3%
SW	0.0%	0.0%	0.0%
CASE II: $Limit_T \leq 50\%$ & $Limit_X \leq 6\%$			
Load	overall D_T	overall D_X	S_E
AR	29.5%	0.0%	30.0%
SR	25.9%	4.6%	27.7%
AW	14.3%	0.0%	23.5%
SW	41.4%	1.6%	7.4%

violated. The maximum energy saving, 30%, is achieved in AR load. For SW load, with a tighter constraint in CASE I, energy cannot be saved, while in CASE II giving a looser constraint, 7% energy can be saved.

V. LEVERAGING ERAID

We present an alternative data layout and block distribution scheme for RAID-1 that are more energy-efficient than traditional disk mirroring. The layout shares with RAID-1 the property that each block of data is replicated exactly once, and either the disk holding the original block or the replica may be accessed on a read operation. However, unlike RAID-1 the system does not maintain an exact mirror of any disk. Instead the blocks on a disk are spread out over a set of redundant disks which collectively maintain a replica the original disk.

With each disk we associate a threshold f , $1/2 < f < 1$, which is the maximum normalized load that can be comfortably tolerated on the disk without degrading performance. When the load on any disk exceeds f , its replica is powered up, and the load on the original disk is now shared between the two.

We consider the operation of the system under read loads. Writes are handled as described in the earlier sections. In normal operation under light load we assume that d original disks are active and d replicas are in standby mode. When the load on a disk exceeds the threshold its replica is awakened. Hence if k , $1 \leq k \leq d$ disks exceed the threshold load, then in a RAID-1 (mirrored) system, there will be $d+k$ disks in active mode and $d-k$ in standby.

Our modified layout works as follows. We stripe blocks on a disk D_i across a set of $\min(s, d)$ redundant disks where $s = |S_i|$ and $S_i = \lceil \frac{1}{1-f} \rceil$. As a consequence, if d_i and any disk belonging to S_i are simultaneously in the active state, then independent of the request sequence, both disks will have a load less than f . The stripe for d_{i+1} begins on the next redundant disk immediately after the disk where the stripe for d_i completed.

We illustrate the arrangement for the case of $d = 5$ and $f = 2/3$ in Figure 3. Table IV(a) shows the minimum number of redundant disks that need to be activated for the mirrored

scheme and the proposed data layout. If either one or two original disks exceeds their threshold, then the load can be maintained below $f = 2/3$ by activating just 1 redundant disk. For instance if D_3 and D_5 exceed the threshold, then redundant disk M4 is activated. The load on D_3 and D_5 drop by $1/3$ each, and M4 increases to $2/3$. For the mirrored scheme, two redundant disks M3 and M5 need to be activated. One may verify the remaining entries of Table IV. If either 3 or 4 (respectively) original disks exceed the threshold, then it is sufficient to activate 2 (respectively 3) redundant disks with the proposed layout, in contrast to 3, 4 (respectively 5) redundant disks activations in the mirroring scheme.

The probability, P of Tables IV(a) expresses the probability of k disks being overloaded. It assumes that each disk has an independent probability p of being overloaded. We can compute the expected number of active redundant disks under this probability model for the two placement schemes. Table IV(b) shows the average number of redundant disks activated for different values of p , for each of the placement schemes and load thresholds. Table IV(b) shows that the percentage of extra redundant disks activated by the mirrored scheme varies between 50% and 115%. Since idle disks translate directly into energy savings, this shows significant reduction in energy consumption of the scheme while bounding the maximum load on any disk to $2/3$.

A complete analysis of the behavior of the system and simulation results will be presented in the complete paper. In this paper we show that there are significant energy savings to be accrued by varying the data placement among the redundant disks. Besides the mirrored situation discussed here, we also show benefits of other erasure coding schemes in improving energy consumption. The details are deferred to the full paper.

VI. CONCLUSIONS

In this paper, we propose an energy saving policy named eRAID that employs request

redirecting for conventional disk based RAID-1 systems. Experimental results show that eRAID can save up to 30% energy without violating predefined performance constraints.

REFERENCES

- [1] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke, "DRPM: dynamic speed control for power management in server class disks," in *Proceedings of the 30th Annual International Symposium on Computer Architecture (ISCA-03)*, (New York), pp. 169–181, June 9–11 2003.
- [2] D. Li and J. Wang, "EERAID: Energy-efficient redundant and inexpensive disk array," in *Proceedings of 11th ACM SIGOPS European Workshop*, (Leuven, Belgium.), September 20-22, 2004.
- [3] X. Li, Z. Li, F. David, P. Zhou, Y. Zhou, S. Adve, and S. Kumar, "Performance-directed energy management for main memory and disks," in *Proceedings of the Eleventh International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'04)*, October 2004.
- [4] Q. Zhu and Y. Zhou, "Power-aware storage cache management," *IEEE Transactions on Computers*, vol. 54, pp. 587 – 602, May 2005.
- [5] D. Colarelli and D. Grunwald, "Massive arrays of idle disks for storage archives," in *Proceedings of Super Computing'2002 Conference CD*, (Baltimore, MD), pp. 312–317, IEEE/ACM SIGARCH, Nov. 2002.
- [6] E. Pinheiro and R. Bianchini, "Energy conservation techniques for disk array-based servers," in *Proceedings of the 18th International Conference on Supercomputing*, pp. 68–78, June 26 - July 01 2004.
- [7] D. Li, H. Cai, X. Yao, and J. Wang, "Exploiting redundancy to construct energy-efficient, high-performance RAIDs," Tech. Rep. TR-05-07-04, Computer Science and Engineering Department, University of Nebraska Lincoln, 2005.
- [8] E. V. Carrera, E. Pinheiro, and R. Bianchini, "Conserving disk energy in network servers," in *Proceedings of the 2003 International Conference on Supercomputing (ICS-03)*, (New York), pp. 86–97, ACM Press, June 23–26 2003.
- [9] E. Varki, A. Merchant, J. Z. Xu, and X. Z. Qiu, "Issues and challenges in the performance analysis of real disk arrays," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, pp. 559–574, June 2004.
- [10] H.-P. Company, "HP SureStore E Disk Array FC60 User's Guide," vol. Pub.No.A5277-90001, Dec. 2000.
- [11] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. New York: John Wiley & Sons, Aug. 1998.
- [12] E. Varki, "Response time analysis of parallel computer and storage systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, pp. 1146–1161, Nov. 2001.
- [13] J. S. Bucy and G. R. Ganger, "The disksim simulation environment version 3.0 reference manual," Tech. Rep. CMU-CS-03-102, Carnegie Mellon University, School of Computer Science, Jan. 2003.