# Dynamic Power-Aware Disk Storage Management in Database Servers

Peyman Behzadnia[1], Wei Yuan[2], Bo Zeng[3], Yi-Cheng Tu[1(✉)], and Xiaorui Wang[4]

[1] Department of Computer Science and Engineering, University of South Florida,
Tampa, FL, USA
{peyman,ytu}@cse.usf.edu

[2] Department of Industrial and Management Systems Engineering, University of South Florida,
Tampa, FL, USA
weiyuan@mail.usf.edu

[3] Department of Industrial Engineering, University of Pittsburgh, Pittsburgh, PA, USA
bzeng@pitt.edu

[4] Department of Electrical and Computer Engineering, The Ohio State University, Columbus,
OH, USA
wang.3596@osu.edu

**Abstract.** Energy consumption has become a first-class optimization goal in design and implementation of data-intensive computing systems. This is particularly true in the design of database management system (DBMS), which was found to be the major consumer of energy in the software stack of modern data centers. Among all database components, the storage system is the most power-hungry element. In this paper, we present our research on designing a power-aware data storage system. To tackle the limitations of the previous work, we introduce a DPM optimization model to minimize power consumption of the disk-based storage system while satisfying given performance requirements. It dynamically determines the state of disks and plans for inter-disk fragment migration to achieve desirable balance between power consumption and query response time. We evaluate our proposed idea by running simulations using several synthetic workloads based on popular TPC benchmarks.

## 1 Introduction

Data centers, criticized as the SUVs of the IT world, consume massive and growing amount of energy. A recent report shows that, in 2013, data centers in the Unites States consumed an estimated 91 billion kilowatt-hours (kWh) of electricity (which costed roughly 7.5 billion US dollars) and are on-track to reach 140 billion kWhs by 2020 [1]. In a typical data center, Database Management System (DBMS) is the largest power consumer among all software modules deployed. And, among all components of a database server, storage system is the most energy hunger constituent. Disk storage system is estimated to consume 25–35 % of total energy consumption in a data center [2]. Another report [3] shows that power consumed by storage in large online transaction processing (OLTP) systems is more than 70 % of the total power of all IT equipment. Power consumption rate of storage systems will grow even larger in the next years - an

S. Hartmann and H. Ma (Eds.): DEXA 2016, Part II, LNCS 9828, pp. 315–325, 2016.
DOI: 10.1007/978-3-319-44406-2_25

annual growth of 60 % has been reported in [4]. Given this strong demand for energy reduction in storage systems, we tackle the problem of designing a power-aware database disk storage system in this paper. Note that the use of SSD drives simplifies the problem since they are highly energy efficient compared to HDDs, but, as of today, SSDs are still not in a position to replace all magnetic disks in large-scale database systems, especially those handling today's big data applications. In previous work, Dynamic Power Management (DPM) algorithms are normally used to save energy in disk storage systems. Such algorithms make real-time decisions on when to transition magnetic disks to lower-power modes with the price of longer response time to data access requests. Many modern hard disks have two power states: active and stand-by. Disks in stand-by mode stop rotation completely thus consume significantly less energy than in active state. However, it incurs a remarkable energy and time cost to spin up to active mode in order to serve a request. Figure 1 shows the detailed specifications related to the power and transition time among different states of a typical multi-mode disk (model Ultrastar 7k6000 from IBM) [5]. In order to amortize the aforementioned penalty cost of disk state change, effective DPM techniques extended the idle period of disks by either controlling the I/O intervals [6–10] or migrating data among disks [11–16]. The first set of works usually considers single-disk systems and utilizes energy-efficient caching or pre-fetching techniques to prolong the idle periods in the I/O trace. The second set of works basically consolidates the most frequently accessed data (called "hot" data in literature) on subset of disks to allow "cold" disks sleep longer. Therefore, they perform corresponding inter-disk data migration in order to achieve the hot data consolidation goal.

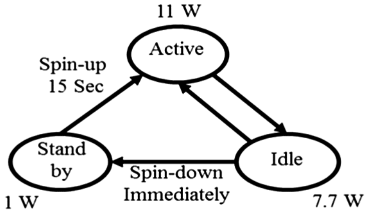


**Fig. 1.** Power modes and their power consumption of the IBM Ultra-Star 7k6000

As the major limitation, work of this type cannot efficiently handle the dynamic I/O traces where arrival rate of data requests changes significantly with respect to time. Furthermore, they do not provide efficient disk state configuration or inter-disk data migration. In this paper, we tackle the limitations of the previous work. The best known algorithm that tries to handle dynamic environment is named Block Exchange (BLEX) presented in [15]. However, we believe BLEX, again, does not efficiently adapt to dynamicity in the workload. The reason is that it maintains some data in stand-by disks and therefore, it incurs significant penalties related to spinning stand-by disks up and down in order to adapt to dynamic changes in data request arrival rates. We address this issue by introducing an optimization model that integrates the Model Predictive Control (MPC) strategy to accommodate dynamic scenarios by enabling optimization actions in an online fashion. Our experimental results clearly show that our proposed model outperforms the BLEX algorithm significantly in terms of both energy savings and response time.

Our contributions and roadmap are summarized as follows: (1) We introduce an integrated DPM optimization model extended with MPC strategy that dynamically determines state (power mode) adjustment and efficient fragment migration to achieve the optimal tradeoff between power consumption and the query response time; (2) We conduct experimental simulations using extensive set of synthetic workloads based on popular TPC benchmarks to evaluate our solution in terms of power saving and response time compared with those of the BLEX algorithm. Our proposed DPM optimization model outperforms the BLEX algorithm significantly in terms of both energy savings and response time in data access. The remainder of this paper is organized as follows: Sect. 2 provides a survey on the related work in the literature; Sect. 3 illustrates the proposed DPM optimization model in detail; Sect. 4 discusses our experimental evaluation; and Sect. 5 concludes the paper.

## 2 Related Work

DPM algorithms are the most popular techniques to achieve energy savings in disk storage systems. Intuitively, the core idea of an effective DPM algorithm is to prolong the idling period of disks in order to allow them sleep longer in the lower-power mode and thus, boost power saving opportunity. We classify algorithmic techniques extending disks idleness period into three different categories: (1) the *first* approach taken in DPM algorithms is *data packing* that consolidates the frequently accessed data (hot fragments) into fewer disks (hot disks) in order to help other disks stay in idle mode longer. An efficient algorithm named Block Exchange (BLEX) is introduced in [15] that dynamically achieves load consolidation and performs block exchange between disks. To the best of our knowledge, BLEX is the most effective algorithm in literature that tries to handle the dynamic I/O traces. Therefore, we will frequently make comparisons to BLEX in describing our solutions in the remainder of this paper. Our experiments will also use BLEX as the baseline. Other similar proposals that exploit data packing are found in [12–14, 16]. They assume RAID layouts which is not the focus of our work; (2) the *second* approach to extend disk inactivity period is to manage I/O intervals via power-aware caching and prefetching algorithms. The main idea is to deploy energy-aware policy in cache data management algorithm (or in prefetching techniques) to redirect some I/O requests to cache in order to change I/O intervals towards longer idle times. Work presented in [6, 7, 10] tackles this method to achieve energy conservation; (3) the *third* class of research works extending disk idleness period tackle energy proportionality in data parallel computing clusters whose files systems maintain a set of replicas for each data block. Papers in [21–23] are classified under this category for energy savings in data parallel clusters. Some other miscellaneous research proposals along with more details on the aforementioned related work are provided in our more thorough survey over the literature in [20].

# 3 Proposed DPM Optimization Model

In this section, we show the design of a DPM optimization model towards balance between energy consumption and performance impact. It is well-known that the arrival rate of data requests changes significantly in respect to time in I/O traces of database servers. This is particularly true in scientific database servers and OLTP servers. The SSDS SkyServer is a famous scientific database server that clearly shows significant changes in the server traffic rate [24]. Also, [17] shows workload changes in an OLTP trace that demonstrates notable arrival rate changes in respect to time. The major problem of previous contributions is that they cannot efficiently adapt to dynamic I/O workloads. We solve this issue by integrating Model Predictive Control (MPC) strategy in an optimization model to enable optimization actions in an online fashion. Section 3.4 describes in detail how our optimization model integrates the MPC technique in order to capture the dynamic changes in data access frequency. Given such significant arrival rate changes in dynamic I/O workloads, we partition the planning horizon into multiple periods where the arrival rate in each period can be modeled by a constant. We formulate a model as a (nonlinear) mixed integer program (shown in Sect. 3.1) where the objective function is the overall cost from all energy consumption elements in the storage system during one epoch. At the beginning of each epoch, based on the observed I/O and the predicted workload for the epoch, the model configures the optimal disk state setting and corresponding inter-disk fragment migration that minimize the energy consumption (aforementioned objective function) during the epoch while maintaining query response time quality. In order to avoid the disk overloading problem, the model performs load balancing between the overloaded disk (s) and other active disks at the beginning of each epoch. In addition to the MPC strategy implemented in our model, another advantage of the DPM model is that we explicitly include fixed charge penalty on disk status change to avoid excessive spin up and down operations (with expensive response time and energy costs), while it is rather considered subjectively in BLEX.

The length of the epoch should be short enough to capture changing arrival rates and also long enough to accommodate disks transition cost and data migration periods, and to impose tolerable number of on/off actions on disks in order to not damage their lifetime services. Considering arrival rate change patterns existing in database I/O traces, we

**Table 1.** DPM model parameters

| Name | Description | Name | Description |
|---|---|---|---|
| $i$ | Index of disks, $i = 1,\ldots,I$ | $ed_i$ | Energy to spin down disk $i$ |
| $j$ | Type of fragmentation, $j = 1,\ldots,J$ | $ep_i$ | Energy to spin up disk $i$ |
| $\lambda_{j,t}$ | Hotness level/popularity of fragment type $j$ in period $t$ | $p^t_{i,k}$ | power consumption of disk $i$ at $k$ spinning state in period $t$ |
| $k$ | State of disk | $\Gamma$ | Response time penalty parameter |
| $Sc_i$ | Storage capacity of disk $i$ | $maxfrag$ | Disk maximum no. of fragments |
| $c_j$ | Migration cost of fragment type $j$ | $\lambda^{max}$ | Maximum fragment popularity |
| $b_j$ | Block size of fragment type $j$ | $M$ | Maximum no. of blocks in a disk |

verified different epoch length values to determine an efficient value that fulfills the above requirements. Based on our sensitivity analysis in [20], the energy saving ratio is insensitive to the epoch lengths larger than 30 min. Therefore, we determined the epoch length to be 30-min long since it captures arrival rate changes effectively while exploiting energy savings. Table 1 introduces the main parameters and indices used in the model development. Table 2 introduces the list of decisions variables used in our DPM optimization model including binary, integer, and continuous variables.

**Table 2.** Decision variables

| Name | Type and description |
|---|---|
| $x^t_{i,j}$ | Integer- Quantity of *j* type fragment on disk *i* in period *t* |
| $y^t_{j,i_1,i_2}$ | Integer- Quantity of *j* type fragments migrated from *i1* to *i2* at the end of period *t* |
| $s^t_{i,k}$ | Binary- Equals to 1 if disk *i* is in state *k* in period *t* |
| $u^t_i$ | Binary-Equals to 1 if disk *i* should be spun up in period *t* |
| $d^t_i$ | Binary- Equals to 1 if disk *i* should be spun down in period *t* |
| $t^k_i$ | Continuous- Response time of disk *i* |
| $T^k_i$ | Continuous- Response time penalty of disk *i* |

### 3.1 Formulation of DPM Optimization for Multi-state Disks

Our objective is to minimize the energy consumption within each epoch period. The total energy consumption during an epoch consists of four elements. The first part is the basement energy that relates to disk state (rotation speed) and number of disks spinning in each state. It is independent of the migration operations. The second part is the energy consumed during the migration time which strictly depends on the total fragment size of migration. And, the rest of energy consumption includes energy costs for disk spin-up and spin-down operations. The objective function is shown in the following equation:

$$\begin{aligned} min \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{k} p^t_{i,k}s^t_{i,k} + \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i_1=1}^{I}\sum_{(i_2\in I, i_2\neq i_1)} c_j y^t_{j,i_1,i_2} + \sum_{t=1}^{T}\sum_{i=1}^{I} ep_i u^t_i \\ + \sum_{t=1}^{T}\sum_{i=1}^{I} ed_i d^t_i + \sum_{t=1}^{T}\sum_{i=1}^{I}\sum_{k} \Gamma \cdot s^t_{i,k} T^k_i \end{aligned} \tag{1}$$

The physical and logical constraints are as follows: (1) Fragments stored in a disk can never exceed the disk capacity; (2) Disks during an epoch period must stay in a certain state; (3) During any epoch *t*, there must be at least one active disk serving the data requests; (4) Any fragment can only migrate once in a certain epoch *t*; (5) A disk in stand-by mode is not considered as source or destination for data migration; (6) There is a limit for data migration (*H*) that represents the data transfer limit for any disk within an epoch. The migration limit by default is set to half of the epoch. The following equations represent the aforementioned constraints respectively:

$$\sum_{j=1}^{J} b_j x_{i,j}^t \le Sc_i \quad \forall i, t \tag{2}$$

$$\sum_{k=1}^{K} s_{i,k}^t = 1 \quad \forall i, t \tag{3}$$

$$\sum_{i=1}^{I} s_{i,k=1}^t \le I - 1 \quad \forall t \tag{4}$$

$$\sum_{i_2} y_{j,i,i_2}^t \le x_{i,j}^t \quad \forall i, t, j (i_2 \ne i) \tag{5}$$

$$\begin{gathered} y_{j,i_1,i_2}^t \le M \cdot \sum_{k=2}^{K} s_{i_1,k}^t \quad \forall i, t, j (i_2 \ne i_1) \\ \sum_j (\sum_{i_2} y_{j,i,i_2}^t + \sum_{i_2} y_{j,i_2,i}^t) \le H \quad \forall i, t, j (i_2 \ne i) \end{gathered} \tag{6}$$

Also, the migration equation that links $x_{i,j}^t$ and $y_{j,i_1,i_2}^t$ is:

$$x_{i,j}^t + \sum_{i_1 \in I, i_1 \ne i} y_{j,i_1,i}^t = x_{i,j}^{t+1} + \sum_{i_2 \in I, i_2 \ne i} y_{j,i,i_2}^t \quad \forall i, t \ge 1, j \tag{7}$$

And, in order to determine the binary indicating variables related to spin up and down of disks, the following equations are used in the model:

$$\sum_k k s_{i,k}^t - \sum_k k s_{i,k}^{t+1} \le u_i^t \tag{8}$$

$$\sum_k k s_{i,k}^{t+1} - \sum_k k s_{i,k}^t \le d_i^t \tag{9}$$

### 3.2 Two-State Optimization Model

It is easy to obtain the model formulation for two-state disk (active and stand-by) storage by setting two values for parameter *k* (1 or 2) in the general formulation provided in the previous section for multi-mode disk. The equations related to two-mode optimization model are provided in detail in [20]. The general DPM optimization model assumes 10 levels of popularity (hotness) for data fragments based on the observed data request arrival rate. We believe that having 10 levels is sufficient to accurately classify data blocks based on the hotness level (if more resolution would be needed, the model can certainly have more levels that indeed reduce the MPC computational time). An important feature of two-state model is that the least and the second least popular data stay in original disks. This will help to minimize the migration cost.

### 3.3 Response Time Modeling

The expected response time of a disk is a function of its spinning state and the total arrival rate. Thus, if we consider the state of disk constant, the response time of the disk is a convex function with respect to its hotness level with increasing first derivative order. We modeled this function by using Piecewise linear (PWL) functions for our optimization model since they are widely used to approximate any arbitrary function (specially convex functions) with high accuracy. The input of PWL function is relative hotness of a disk. The relative hotness of a disk is calculated by following equation:

$$\lambda_{t,i} = \frac{\sum_{j} \lambda_j x_{i,j}}{\lambda^{max} maxfrag} \tag{10}$$

where $0 \leq \lambda_{t,i} \leq 1, 1 \leq \lambda_j \leq 10$ is the popularity (arrival rate) of fragment type $j$, $maxfrag$ is maximum number of fragments in a disk and $\lambda^{max}$ is upper bound for popularity. We define $L$ as the number of linear functions to approximate the response time. It is well known that PWL functions can represent arbitrary functions to any accuracy by simply increasing the number of segments ($L$) to the point of desired accuracy. Therefore, we verified different $L$ values for approximation of the response time convex function. We decided to use 9-piece-linear function shown in Fig. 2 for two-state disk storage system since it approximates the convex function with high accuracy.

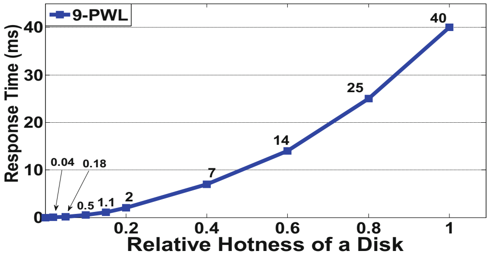


**Fig. 2.** 9 PWL function of response time model

### 3.4 Model Predictive Control (MPC)

The presented optimization model is rather static while our real system works in a dynamic online enviroment. Thus, we extend the model to accommodate dynamic scenarios by using Model Predictive Control (MPC) techniques to solve this issue. MPC, also known as receding horizon control (RHC) or rolling horizon control, is a form of control strategy to integrate optimization. Specifically, the current control action is obtained in an online fashion where, at each sampling instant, a finite horizon optimization problem (which is (1)–(10)) is solved and its optimal solution in the first stage is applied as the current control decision while remaining solutions will be disregarded. Such procedure repeats along the whole control process. Therefore, all controllable variables (such as disk status and response time) for the first period are implemented in MPC.

It has been observed that MPC is a very effective control strategy with reasonable computational overhead [18]. The prediction information on workload arrival rate is provided to the MPC optimization model. This plays a key role in developing an accurate underlying mixed integer program model since any mis-prediction on data request arrival rates could cause the model to produce a solution with a less desired quality. However, as observed in many other applications of MPC, since only the first stage solution will be implemented and remaining parts will be ignored, MPC is robust to poor predictions and has a strong adjustment capability [26].

### 3.5 Solving Strategy

Our initial attempt to find solutions to the two-state model is to implement and solve the model in the well-known Cplex solver. The solver is installed on a server which is connected to the server running the widely used disk simulator, *Disksim* [19], which is utilized as an accurate and reliable simulation platform by many related works. In other words, the model solution is integrated in the storage system simulated in Disksim. Technical details regarding the experimental simulations are provided in Sect. 4.

## 4 Empirical Evaluation

We conducted simulations under extensive set of dynamic I/O workloads to validate our proposed method. We have compared our results in terms of energy saving ratio and average response time with those of the BLEX algorithm. The simulated disk storage system in Disksim consists of an array of 15 conventional hard disks; each disk is configured as in independent unit of storage. The hard disk model used in simulations is IBM Ultrastar 7K6000 [5] whose main specifications are provided in [20].

### 4.1 Synthetic Workload Generator

We developed a workload generator written in C to synthesize I/O workloads for disks based on popular database TPC benchmarks. We follow the well-known $b/c$ model in generating a workload of a series of random data read operations ($b$ % of all read operations is against $c$ % of the data) [25]. It is well known that database tuple access pattern is highly skewed and can be described as an 80/20 or even a 90/10 model [20]. Zipf probability distribution is used in the generator to produce $b/c$ model. The default $b/c$ model used in simulations is set to 80/20. We have used Gamma distributions in our workload generator to reflect the dynamic behavior of database I/O disk trace. Given the data correlations among database tuples in queries, the access frequency change pattern of each data fragment type is represented by a Gamma distribution.

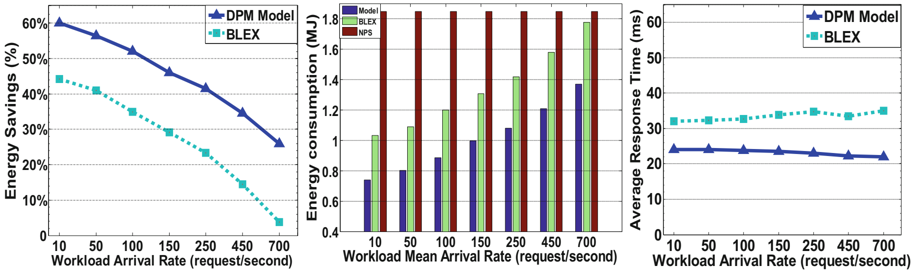
### 4.2 Experimental Platform

Our model is integrated in the disk simulator as well as BLEX, as the comparison target. We enhanced Disksim with a multi-speed disk power model where the power consumption rate is proportional to disk rotation speed. Also, it is augmented with extra features such as dynamic disk spin up (and down), disk state adjustment and inter-disk data migration during the simulation. The predicted access frequency (hotness level) for each fragment type for the next *k* epochs is provided to the model along with the observed fragment type frequencies in the previous epoch. The prediction is performed by the prediction and autoregressive modeling methods in MATLAB. In particular, based on the observed data access frequency, autoregressive modeling tool develops an identified model. Then, the prediction method forecasts fragments access frequency for *k* epochs ahead based on the identified model and the observed fragments frequency.

### 4.3 Simulation Results and Comparisons

In this section, we describe our experimental results in terms of energy saving and average response time under extensive set of dynamic traces.

**Energy Saving Results.** Figure 3(a) shows energy saving for various I/O traces with different mean arrival rates. Figure 3(a) clearly shows that the DPM optimization model significantly outperforms the BLEX algorithm by saving energy up to 60 %. The proposed model outperforms BLEX with the difference of minimum 16 % and up to 23 % in energy savings. Based on the results, it saves 19 % more energy on average than BLEX. Figure 3(b) shows the total power consumption of the disk storage system for each power saving method compared to that of no power saving (NPS) method applied, where all disks constantly run in active mode. Such results are shown for several I/O traces. We can conclude that DPM optimization model is dominant in power saving.



(a) Energy Saving Results (b) Total Power Consumption (c) Average Response Time

**Fig. 3** Experimental results under dynamic I/O traces with different mean arrival rates

**Average Response Time Results.** It is important to measure the response time effected by power saving schemes to ensure high quality of service for queries. Figure 3(c) shows the average response time for DPM model and BLEX algorithm under several workloads with various mean arrival rates. Note that the computational time to obtain the solution for both power saving schemes is up to a second, which is apparently ignorable comparing to the epoch length (30 min), and thus it is excluded from the response time computations above. The results show that optimization model provides significantly better response time than BLEX. The reason, in addition to response time consideration in its optimal power-performance tradeoff, is that it takes into account the predicted information on data access frequency for the next epoch in its solutions.

## 5 Conclusion

Power consumption has increased greatly in data centers, and DBMS is the major energy consumer. Disk storage systems are the most power-hungry components among all in DBMS. Thus, we present our proposals in this paper on designing a power-aware disk storage system that improves on the limitations of previous contributions. We introduced a DPM optimization model extended with the MPC strategy that can be adapted to any multi-speed disk storage system. We developed the two-state DPM optimization model for two-mode disk storage systems since most of the modern disks in the market run in two modes. We evaluated our proposed method by experimental simulations using extensive set of synthetic I/O traces based on popular TPC benchmarks.

**Acknowledgments.** The project described was supported by grants IIS-1117699 and IIS-1156435 from the National Science Foundation (NSF) of USA. Equipments used in the experiments are partially supported by a grant (CNS-1513126) from NSF.

## References

1. http://www.nrdc.org/energy/data-center-efficiency-assessment.asp
2. Gurumurthi, S., Sivasubramaniam, A., Kandemir, M., Franke, H.: Reducing disk power consumption in servers with DRPM. J. Comput. **36**(12), 59–66 (2003)
3. Poess, M., Nambiar, R.O.: Energy cost, the key challenge of today's data centers: a power consumption analysis of tpc-c results. In: VLDB 2008 Proceedings. ACM Press (2008)
4. Moore, F.: more power needed. In: Energy User News, November 2002
5. www.hgst.com/hard-drives/enterprise-hard-drives/enterprise-sas-drives/ultrastar-7k6000
6. Zhu, Q., Zhou, Y.: Power-aware storage cache management. IEEE Trans. Comput. **54**(5), 587–602 (2005)
7. Papathanasiou, A.E., Scott, M.L.: Energy efficient prefetching and caching. In: USENIX Annual Technical Conference, Boston (2004)
8. Li, D., Wang, J.: Eeraid: power efficient redundant and inexpensive disk arrays. In: 11th Workshop on SIGOPS European Workshop, Belgium (2004)
9. Yao, X., Wang, J.: RIMAC: a novel redundancy-based hierarchical cache architecture for energy efficient, high performance storage systems. In: ACM SIGOPS OS Review (2006)
10. Zhu, Q., David, F.M., Devaraj, C.F., Li, Z., Zhou, Y., Cao, P.: Reducing energy consumption of disk storage using power-aware cache management. In: IEEE Proceedings of Software (2004)
11. Pinheiro, E., Bianchini, R.: Energy conservation techniques for disk array-based servers. In: Proceedings of ICS 2004 (2004)
12. Colarelli, D., Grunwald, D.: Massive arrays of idle disks for storage archives. In: ACM/IEEE Conference on Supercomputing, pp. 1–11 (2002)
13. Weddle, C., Oldham, M., Qian, J., Wang, A., Reiher, P., Kuenning, G.: PARAID: A gear-shifting power-aware RAID. ACM Trans. Storage (TOS) 3(3) (2007). 13

14. Verma, A., Koller, R., Useche, L., Rangaswami, R.: SRCMap: energy proportional storage using dynamic consolidation. In: Proceedings of FAST 10, vol. 10
15. Otoo, E., Rotem, D., Tsao, S.-C.: Dynamic data reorganization for energy savings in disk storage systems. In: Gertz, M., Ludäscher, B. (eds.) SSDBM 2010. LNCS, vol. 6187, pp. 322–341. Springer, Heidelberg (2010)
16. Guerra, J., Pucha, H., Glider, J., Belluomini, W., Rangaswami, R.: Cost effective storage using extent based dynamic tiering. In: Proceedings of FAST, pp. 273–286 (2011)
17. Zhu, Q., Chen, Z., Tan, L., Zhou, Y., Keeton, K., Wilkes, J.: Hibernator: helping disk arrays sleep through the winter. In: ACM SIGOPS Operating Systems Review (2005)
18. Garcia, C.E., Prett, D.M., Morari, M.: Model predictive control: theory and practice- a survey. In: Automatica 1989
19. http://www.pdl.cmu.edu/DiskSim/
20. Behzadnia, P., Tu, Y.-C., Zeng, B., Yuan, W., Wang, X.: Dynamic Power-Aware Disk Storage Management in Database Server. Technical report CSE/15-123., Department of Computer Science and Engineering, University of South Florida (2015). http://msdb.csee.usf.edu/E2DBMS/tech-report-123.pdf
21. Kim, J., Chou, J., Rotem, D.: iPACS: power-aware covering sets for energy proportionality and performance in data parallel computing clusters. J. Parallel Distrib. Comput. Elsevier **74**(1), 1762–1774 (2014)
22. Kim, J., Chou, J., Rotem, D.: Energy proportionality and performance in data parallel computing clusters. In: 23rd SSDBM Conference, July 2011
23. Chou, J., Kim, J., Rotem, D.: Energy-aware scheduling in disk storage systems. In: Proceedings of ICDCS 2011 (2011)
24. The SDSS DR1 SkyServer. http://skyserver.sdss.org/dr1/en/skyserver/paper/
25. Nicola, M., Jarke, M.: Performance modeling of distributed and replicated databases. IEEE Trans. Knowl. Data Eng. (TKDE) **12**(4), 645–672 (2000)
26. http://cepac.cheme.cmu.edu/pasilectures/lee/LecturenoteonMPC-JHL.pdf