Future Generation Computer Systems 29 (2013) 196–207

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An adaptive energy-conserving strategy for parallel disk systems

Mais Nijim [a], Xiao Qin [c,*], Meikang Qiu [b], Kenli Li [d]

[a] Department of Electrical Engineering and Computer Science, Texas A&M University-Kingsville, Kingsville, TX 78363, United States
[b] Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506-0046, United States
[c] Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, United States
[d] School of Computer and Communication, Hunan University, Changsha, 410082, China

ARTICLE INFO

Article history:
Received 20 April 2010
Received in revised form
30 January 2012
Accepted 14 May 2012
Available online 11 June 2012

Keywords:
Energy saving
Parallel disks
Adaptive strategy

ABSTRACT

Although various parallel disk systems have been developed to achieve high I/O performance and energy efficiency, most existing parallel disk systems lack an adaptive way to conserve energy in dynamically changing workload conditions. To solve this problem, we develop an adaptive energy-saving scheme or DCAPS in parallel disk systems. We show that adaptability in energy conservation can be achieved through the integration of a dynamic disk scheduling scheme and power management in parallel disk systems. DCAPS consists of a data partitioning mechanism, a response time estimator, and an adaptive energy-conserving mechanism. The Data partitioning mechanism allows DCAPS to adjust the parallelism degrees of write requests based on dynamic workload conditions. Apart from supporting the data partitioning mechanism, the response time estimator makes it possible for the adaptive energy-conserving mechanism to dynamically adjust voltage supply levels while guaranteeing desired response times. We conducted extensive experiments to quantitatively evaluate the performance of the proposed energy-conserving strategy. Experimental results consistently show that DCAPS significantly reduces energy consumption of parallel disk systems in a dynamic environment over the same disk systems without using DCAPS.

Published by Elsevier B.V.

1. Introduction

Since the performance gap between processors and disks has widened over the last two decades [1], the performance of data-intensive applications has been significantly affected by disk systems [2,3]. Because parallel disks are highly scalable, parallel disk systems can alleviate I/O bottleneck problems in many data-intensive systems like video surveillance [4], remote-sensing database systems [5,6], digital libraries [7,8], simulation tools [9], and long running simulations [10].

Although parallel disks play an important role in achieving high-performance for data-intensive applications, a substantial amount of energy consumed in data-intensive systems is contributed by parallel disk systems. A recent industry report reveals that storage devices account for almost 27% of the total energy consumed by a data center [11]. For example, the power consumption of today's data center ranges from 75 to 200 W/ft^2. Since this trend will undoubtedly continue in the near future [12], the energy-consumption problem in data centers will become even more serious. Therefore, we are motivated to extensively investigate energy-conservation software techniques for parallel disk systems.

Modern parallel disk systems have increasingly become energy efficient (see, for example, [13]); however, there is a lack of approaches for adaptively conserving energy in parallel disks. Adaptive energy-saving techniques are important in parallel disk systems because of two reasons. First, many data centers have dynamically changing workload characteristics. For example, I/O loads of web servers are known to dynamically change with time [14]. Second, real-world data-intensive applications tend to have performance and resource requirements. For example, disk requests issued by data-intensive applications need to be completed within specified response times [15]. Adaptively conserving energy in parallel disk systems becomes particularly critical for data-intensive applications running in dynamically changing computing environments.

In this study, we are inspired by the needs of data-intensive applications to develop a way of flexibly and adaptively reducing energy consumption caused by the data-intensive applications. We show that adaptability in energy conservation can be achieved through the integration of a dynamic disk scheduling scheme and power management in parallel disk systems. We focus

* Corresponding author.
E-mail addresses: mais.nijim@tamuk.edu (M. Nijim), xqin@cs.nmt.edu, xqin@auburn.edu (X. Qin), mqiu@engr.uky.edu (M. Qiu), lkl510@263.net (K. Li).
URL: http://www.eng.auburn.edu/~xqin (X. Qin).

0167-739X/$ – see front matter. Published by Elsevier B.V.
doi:10.1016/j.future.2012.05.003

on disk scheduling in this research, because disk scheduling algorithms (e.g., shortest seek time first (SSTF) and SCAN) can significantly improve disk performance [16–20]. For example, the SSTF algorithm aims to minimize seek time of disk requests [21]; the SCAN algorithm solves the unfairness problem and reduces seek time [21]; and Reist and Daniel developed a parameterized generalization of SCAN and SSTF to seamlessly integrate the two algorithms [22]. Most existing disk scheduling algorithms are inadequate for improving energy efficiency of parallel disk systems. To address this problem, we develop an adaptive energy-conservation technique or DCAPS that incorporate power management with disk scheduler in parallel disks systems. More importantly, our DCAPS scheme can offer significant energy savings while guaranteeing desired response times of disk requests.

Our DCAPS scheme can manage two types of disk I/O parallelisms, namely, inter-request parallelism and intra-request parallelism. The inter-request parallelism allows multiple independent disk requests to be served simultaneously by multiple disks, whereas the intra-request parallelism enables a single disk request to be responded by an array of disks in parallel. The parallelism degree of a data request is the number of disks in which the requested data is residing [23]. DCAPS adjusts the parallelism degrees of write requests based on dynamic workload conditions (see Section 4.1). After determining parallelism degrees, DCAPS optimizes disk supply voltage levels to reduce power consumption while guaranteeing requests' desired response times by utilizing the disk scheduling mechanism (see Sections 4.2 and 4.3).

Experimental results show that DCAPS significantly reduces energy consumption of parallel disk systems in a dynamic environment over the same disk systems without using DCAPS. In addition, DCAPS improves energy efficiency of parallel disks without reducing satisfied ratio of requests having desired response times.

The rest of the paper is organized as follows. We summarize related work in the next section. Section 3 describes the system architecture for energy-efficient parallel disk systems. In Section 4, we propose the adaptive energy-conservation scheme called DCAPS. Section 5 evaluates the performance of the proposed energy-saving technique by comparing an existing approach. Section 6 concludes the paper with summary and future directions.

2. Related work

Disk I/O has become a performance bottleneck for data-intensive applications due to the widening gap between processor speeds and disk access speeds [1,24,25]. To help alleviate the problem of disk I/O bottleneck, a large body of work has been done on parallel disk systems. For example, Kallahalla and Varman designed an on-line buffer management and scheduling algorithm to improve performance of parallel disks [26]. Scheuermann et al. addressed the problem of making use of striping and load balancing to tune performance of parallel disk systems [23]. Rajasekaran and Jin developed a practical model for parallel disk systems [27]. Kotz and Ellis proposed investigated several write back policies used in a parallel file system implementation [28]. Our research is different from the previous studies in that we focused on energy savings for parallel disk systems. Additionally, our strategy is orthogonal to the existing techniques in the sense that our scheme can be readily integrated into existing parallel disk systems to substantially improve energy efficiency and performance of the systems.

Abundant research has been done to improve energy efficiency of mobile devices (e.g., smart phones) to increase battery life of the devices. Unlike mobile devices where battery life is critical, disk systems have to be energy efficient in order to lower electricity bills in data centers. In the past decade, much attention has been paid to the development of energy-efficient disk systems. For example, the conventional wisdom of saving energy in disks is to place idle disks into the low-power (e.g., standby) mode. Because significant energy in disks is consumed by spindle motors, dynamic power management schemes are proved to be very effective [29]. Therefore, the dynamic power management techniques have been widely applied to reduce energy dissipation in disk drives of both PCs and high-performance computers [30].

Most of the previous research regarding conserving energy focuses on single disk system in laptops and mobile devices to extend the battery life. Recently, a handful of techniques proposed to save energy in disk systems include dynamic power management [30–32], I/O workload-skew schemes [33,34], power-aware cache management [35–37], power-aware perfecting schemes [38–40], software-directed power management techniques [41], redundancy techniques [41], multi-speed disks [42–44], and data placement technique [45]. However, the research on energy-efficient parallel disk systems is still in its infancy. It is imperative to develop new energy conservation techniques that can provide significant energy savings for parallel disk systems while maintaining high performance.

The dynamic voltage scaling technique or DVS is a widely adopted approach to conserving energy in processors. The DVS technique can dynamically reduce the voltage supplies of processors to conserve energy consumption in processors (see, for example, [46,47]). Thus, processor voltage supplies are scaled down to the most appropriate levels, thereby quadratically reducing power whenever possible. Compared with traditional computer systems with fixed voltage supplies, DVS-enabled systems can achieve high energy efficiency. Our approach differs from the conventional DVS methods, because ours is the first technique of its kind designed exclusively for energy-efficient parallel disk systems aiming to guarantee desired response times requests issued by of data-intensive applications. Our adaptive energy-conserving strategy seamlessly incorporates DVS-enabled dynamic power management into the disk scheduler to achieve high energy efficiency in parallel disk systems while satisfying desired response times.

3. System architecture and model

In this section, we first present a framework within which we develop an adaptive energy-conservation technique for parallel disk systems. Then, we describe the power consumption model used to estimate the power consumption of large-scale parallel disk systems.

3.1. System architecture

Fig. 1 outlines the framework of energy-efficient parallel disk systems. The framework is general enough to accommodate a wide range of storage systems, including both network attached storage devices (NAS) and storage area networks (SAN). The framework in Fig. 1 embraces a parallel disk system, networks, and the DCAPS scheme. In this study, we focus on the development of DCAPS that consists of a data partitioning mechanism (see Section 4.1), a response time estimator (see Section 4.2), and an adaptive energy-conserving mechanism (see Section 4.3).

The data partitioning mechanism (see Section 4.1) is geared to divide a large amount of data into fixed-size of data units stored on a number of disks. We consider file striping – a generic method for various data types. To optimize the parallelism degree (a.k.a., stripe unit size) for each write request, the data partitioning mechanism relies on the response time estimator to predict requests' response times.

The response time estimator (see Section 4.2) not only supports the data partitioning mechanism, but also is indispensable for the adaptive energy-conserving mechanism. The response time esti-

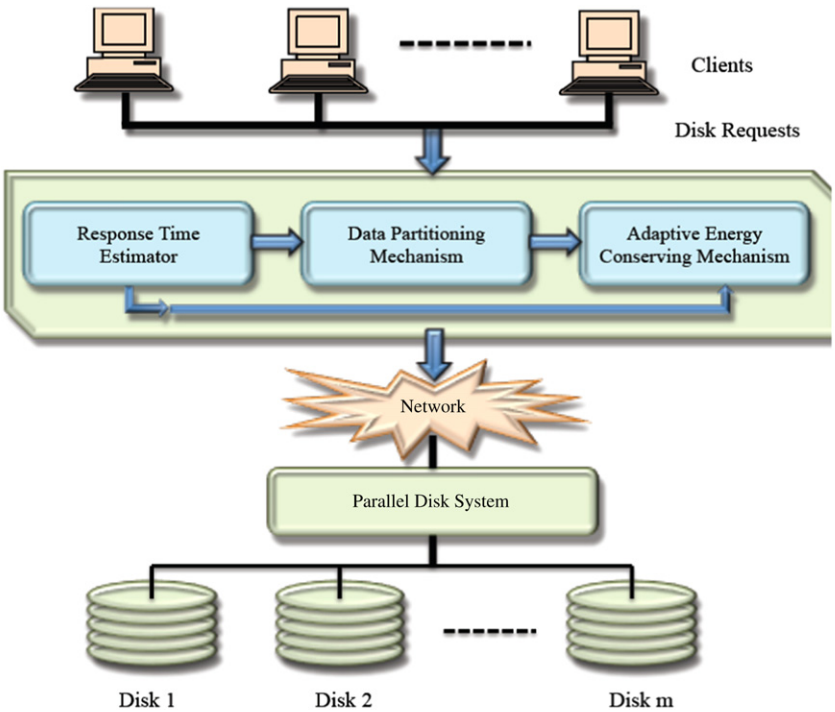

Fig. 1. The framework of energy-efficient parallel disk systems.

mator is important, because estimating response times makes it possible for the adaptive energy-conserving mechanism to dynamically adjust voltage supply levels while guaranteeing desired response times.

The energy-conserving mechanism (see Section 4.3) – at the heart of the proposed DCAPS – is responsible for adaptively saving energy in parallel disks without violating desired response times of disk requests. Thus, the energy-conserving mechanism aims to achieve the best tradeoff between energy efficiency and performance. More specifically, the adaptive energy-conserving mechanism reduces energy consumption by making use of the dynamic voltage scaling technique to judiciously lower voltage supply levels of disks as long as specified performance requirements can be met.

3.2. Energy consumption model

Before developing the adaptive energy-conserving mechanism, we first introduce a power consumption model for parallel disk systems. We consider a sequence of disk requests $R = \{r_1, r_2, \ldots, r_n\}$ submitted to a parallel disk system. Each disk request $r_i \in R$ has an arrival time a_i, a desired response time t_i, and data size d_i. Ideally, request r_i needs to be completed within the desired response time t_i.

A multiple-voltage disk system has a number of discrete voltages; the disk system can instantaneously switch from one voltage to another. Without loss of generality, we assume the parallel disk system can be operated at a finite set $V = \{v_1, v_2, \ldots, v_{\max}\}$ of voltage supply levels. Given a disk voltage v_i, we can accordingly determine the bandwidth b_i of the disk.

Because energy dissipation in disks quadratically proportional to supply voltages, voltage scaling can achieve significant energy savings for disks. Thus, the energy consumption rate P_i of the ith disk can be expressed as below:

$$P_i = C_1 \cdot v_{i,dd}^2 \cdot \frac{\left(v_{i,dd} - v_t\right)^{\alpha}}{C_2}, \quad v_{i,dd} \in V, v_{i,dd} \geq v_t; \tag{1}$$

where C_1, C_2, and $\alpha \in [4,48]$ are constants depending on physical characteristics of disk devices, $v_{i,dd}$ is the supply voltage, and v_t is the threshold voltage. Let D_i denote a set of disk requests to be processed by the ith disk in the parallel disk system. Given a disk request r_j to be processed by the ith disk, we can calculate the energy consumption of the request as below:

$$E_{i,j} = P_i\left(v_{i,dd}^j\right) \cdot \theta_j\left(v_{i,dd}^j\right), \tag{2}$$

where $v_{i,dd}^j$ is the voltage supply level determined for the disk request, $P_i\left(v_{i,dd}^j\right)$ is the disk's energy consumption rate, and $\theta_j\left(v_{i,dd}^j\right)$ is the processing time of the disk request. Both $P_i\left(v_{i,dd}^j\right)$ and $\theta_j\left(v_{i,dd}^j\right)$ largely rely on the supply voltage $v_{i,dd}^j$ of the disk; $P_i\left(v_{i,dd}^j\right)$ can be straightforwardly derived from Eq. (1).

The energy consumption E_i of disk i is written as a summation of energy consumption caused by each disk request handled by the disk. Thus, we have

$$E_i = \sum_{r_j \in D_i} E_{i,j} = \sum_{r_j \in D} P_i\left(v_{i,dd}^j\right) \cdot \theta_j\left(v_{i,dd}^j\right). \tag{3}$$

Suppose there are m disks in the parallel disk system, the total energy consumption E of the disk system can be expressed as:

$$E = \sum_{i=1}^{m} E_i = \sum_{i=1}^{m} \sum_{r_j \in D_i} E_{i,j} = \sum_{i=1}^{m} \sum_{r_j \in D} P_i\left(v_{i,dd}^j\right) \cdot \theta_j\left(v_{i,dd}^j\right). \tag{4}$$

We can now obtain the following non-linear optimization problem formulation to compute the energy consumption of a parallel disk system

$$\begin{aligned} &\text{Minimize} \quad E = \sum_{i=1}^{m} \sum_{r_j \in D} P_i\left(v_{i,dd}^j\right) \cdot \theta_j\left(v_{i,dd}^j\right), \\ &\text{Subject to} \quad (a)\ v_{i,dd}^j \in \{v_1, v_2, \ldots, v_{\max}\}, \\ &\qquad\qquad\quad (b)\ f_j \leq t_j, \end{aligned} \tag{5}$$

where f_j is the response time of the jth disk request. $f_j \leq t_j$ in Expression (5) signifies that the desired response time constraints must be met.

4. Adaptive energy-conservation strategy

The proposed adaptive energy-conserving strategy encompasses three components, namely, a data partitioning technique, response time estimation method, and an adaptive DVS algorithm. In this section, we describe the design of these three components in more detail.

4.1. Data partitioning

The proposed DCAPS framework (see Section 3.1) depends on the data partitioning scheme to optimize parallelism degrees for write requests. The adaptability of DCAPS is made possible by dynamic data partitioning that helps in minimizing write request response times. Therefore, the data partitioning scheme in DCAPS offers great opportunities to improve energy efficiency by scaling down disk supply voltages. In the first phase, DCAPS utilizes the dynamic data portioning scheme to substantially shorten response times by adaptively determining the parallelism degrees of each write request (see Step 3 in Fig. 2 in Section 4.3).

We denote the parallelism degree and data size of a request r_i by p_i and d_i, respectively. Before proceeding to the analysis of optimal parallelism degrees, let us first formally derive the disk service time $T_{disk}(d_i, p_i)$ of request r_i. Thus, the disk service time can be computed as

$$T_{disk}(d_i, p_i) = T_{seek}(p_i) + T_{rot}(p_i) + T_{trans}(d_i, p_i), \tag{6}$$

where $T_{seek}(p_i)$, $T_{rot}(p_i)$, and $T_{trans}(d_i, p_i)$ are the seek time, rotation time, and transfer time of the disk request. Seek time $T_{seek}(p_i)$ can be approximated as Eq. (7) [23]:

$$T_{seek}(p_i) = eC\,(1 - a - b\ln(p_i)) + f \tag{7}$$

where C is the number of cylinders on a disk, a and b are two disk-independent constants, whereas e and f are disk-dependent constants.

The value of rotation time can be expressed as Eq. (8):

$$T_{rot}(p_i) = \frac{p_i}{p_i + 1} \cdot T_{ROT} \tag{8}$$

where T_{ROT} is the nominal rotation time of a disk. The disk request transfer time can also be given as:

$$T_{trans}(d_i, p_i) = \frac{d_i}{p_i} \cdot \frac{1}{B_{disk}} \tag{9}$$

where B_{disk} denotes the disk bandwidth.

Substituting Eqs. (7)–(9) into Eq. (6), we obtain the value of disk service time as:

$$T_{disk}(d_i, p_i) = eC\,(1 - a - b\ln(p_i)) + f + \frac{p_i}{p_i + 1} \cdot T_{ROT} + \frac{d_i}{p_i} \cdot \frac{1}{B_{disk}}. \tag{10}$$

Now we are positioned to calculate the optimal parallelism degree of request r_i by determining the minimum of the function $T_{disk}(d_i, p_i)$. Thus, we can obtain the optimal value of p_i by solving Eq. (11).

$$\frac{dT_{disk}(d_i, p_i)}{d\,(p_i)} = \frac{T_{ROT}}{p_i + 1} - \frac{p_i \cdot T_{ROT}}{(p_i + 1)^2} - \frac{eCb}{p_i} - \frac{d_i}{p_i^2} \cdot \frac{1}{B_{disk}} = 0. \tag{11}$$

The parallelism degree determined by Eq. (11) cannot exceed the number (i.e., m) of disks in the parallel disk system. Consequently, the optimal parallelism degree is the minimum value of pi and m (i.e., $\min(p_i, m)$).

4.2. Response time estimator

The response time estimator is of importance to both the data partitioning mechanism and the adaptive energy-conserving mechanism. The maximum response time of a disk request is defined as the time interval between submission and completion of the request by the parallel disk system. The response time for a newly issued disk request r is the sum of queuing delay, data partitioning time, and processing delay. Thus, we have:

$$T(r, p, \sigma) = T_{queue} + T_{partition} + \max_{i=1}^{p}\left\{T^i_{proc}(r, p, \sigma_i)\right\}, \tag{12}$$

where p is the parallelism degree of disk request r, T_{queue} is the queuing delay at the client side, $T_{partition}$ is the time spent in data partitioning, and T^i_{proc} is the system processing delay experienced by the ith stripe unit of the request. With respect to the ith stripe unit of the request, the system processing delay T^i_{proc} can be expressed as

$$T^i_{proc}(r, p, v_i) = T^i_{network}(r, p, v_i) + T^i_{disk}(r, p, v_i), \tag{13}$$

where $v = (v_1, v_2, \ldots, v_p)$ is the request's vector of the supply voltage for p stripe units, $T^i_{network}$, and T^i_{disk} are the delays at the network subsystem, and parallel disk subsystems, respectively.

We assume that when the ith stripe unit of a request arrives at the network queue, there are k stripe units waiting to be delivered to the parallel disk sub-system. Suppose stripe units are transmitted in a first-in-first-out order, all the stripe units that are already in the queue prior to the arrival of the ith stripe unit must be transmitted earlier than the ith stripe unit. Hence, the delay in the network subsystem $T^i_{network}(r, p, v_i)$ can be written as

$$T^i_{network}(r, p, v_i) = \frac{i \cdot \frac{d}{p} + \sum_{j=1}^{k} d_j}{B_{network}}, \tag{14}$$

where d_j is the data size of the jth stripe unit in the network queue, and B_{network} is the effective network bandwidth. It is worth noting that k in Eq. (14) is the optimal parallelism degree determined by the data partitioning mechanism (see Eq. (11) in Section 4.1).

Similarly, it is assumed that when the ith stripe unit of the request arrives at disk j, there are k disk requests must be processed by disk j before handling the stripe unit. Thus, the delay in the disk subsystem $T^i_{disk}(r, p, v_i)$ is given by the following formula

$$T^i_{disk}(r, p, v_i) = T_{disk,j}(d/p) + \sum_{l=1}^{k} T_{disk,j}(d_l), \tag{15}$$

where $T_{disk,j}(d)$is the disk processing time of a request containing d bytes of data. We can quantify $T_{disk,j}(d)$as follows

$$T_{disk,j}(d) = T_{seek} + T_{rot} + \frac{d}{B_{disk}}, \tag{16}$$

where T_{seek} and T_{rot} are the seek time and rotational latency, and $\frac{d}{B_{disk}}$ is the data transfer time depending on the data size d and disk bandwidth B_{disk}.

4.3. The adaptive energy-conservation algorithm

The adaptive energy-conservation algorithm dynamically optimizes parallel disk voltage levels to reduce energy consumption caused by each disk request. The algorithm adaptively chooses the most appropriate voltage for stripe units of a disk request while guaranteeing the desired response time of the request.

The proposed algorithm fully utilizes the dynamic data partitioning mechanism (see Section 4.1) and the response time estimator (see Section 4.2) to adaptively control voltage levels. The algorithm outlined in Fig. 2 attempts to minimize the supply

Input: r: a newly arrived disk request
t_i: desired response time of the ith request
v_{max}: the maximum supply voltage
v_{min}: the minimum supply voltage
Q, a waiting queue at the client side
1. Insert r into Q based on the earliest desired response time first policy
2. **for** each request r_j in the waiting queue Q **do**
/* Phase 1: dynamic data partitioning */
3. Calculate the optimal parallelism degree p_i of r_i
4. Partition the request into p_i stripe units
5. **for** each stripe unit of r_i **do**
6. Initialize v_{ij} of the jth stripe unit to the maximum supply voltage v_{max}
/* Phase 2: response time estimation */
7. Apply Eqs. (15) and (16) to estimate the response time of the jth stripe unit;
/* Phase 3: Dynamic Voltage Scaling */
8. **while** (estimated response time < desired response time t_i) and (**if** $v_{ij} > v_{min}$) **then**
/* v_{ij} can be further reduced */ (see **property 1**)
10. scale the voltage v_{ij} down to the next level;
11. Apply Eqs. (15) and (16) to estimate the response time of the jth stripe unit;
12. **end while**
13. Deliver the jth unit through the network subsystem to the parallel disk system;
14. **end for**

Fig. 2. The adaptive energy-conserving strategy (DCAPS) for parallel disk systems.

voltage levels of a disk request to improve the energy efficiency of the disk systems.

The algorithm takes the following three steps to reduce voltage levels. First, the algorithm dynamically calculates the optimal parallelism degree of the disk request to effectively shorten response time of the request in the parallel disk system (see Eq. (11)). Second, the algorithm estimates the response time of each stripe unit of a disk request (see Eq. (12)). Third, based on the estimated response time and desired response time, the algorithm adaptively lower down the supply voltage for each stripe unit provided that the request can be completed before its desired response time.

When a disk request is issued to the system, the DCAPS strategy inserts the newly arrived requests into the waiting queue based on the earliest desired response time first policy (see Step 1). After the data portioning of each request in the queue, DCAPS initializes the voltage of all the stripe units of request r_i to the maximum supply voltage $v_{\max}$ (see Step 6). In doing so, DCAPS are more likely to guarantee desired response times under heavily loaded conditions. Using the dynamic voltage scaling technique, the DCAPS strategy adaptively makes disks operate at low voltage supply levels for all the stripe units to conserve the total energy consumption of the parallel disk system. Assume that the maximum supply voltage $v_{\max}$. is 3.3 V, the supply voltage can be reduced as long as the disk request can be accomplished within its desired response time or the supply voltage reach the minimum voltage $v_{\min}$. In this study, we assume that the threshold voltage is 0.8 [online]. In an effort to steadily reduce the voltage of stripe units, DCAPS guarantees that all requests will be completed before their desired response times. Thus, the following property needs to be satisfied in DCAPS.

(1) The reduced supply voltage v_{ij} is greater than the minimum voltage $v_{\min}$;
(2) $T_j(r_i, p_i, \sigma_i) \le t_i$, where T_j is the response time of the jth stripe unit, t_i is the desired response time of the request, and $T_j(r_i, p_i, \sigma_i) = T_{queue} + T_{partition} + T_{proc}^{ij}(r_i, p_i, \sigma_{ij})$.

Steps 10–11 are repeatedly performed to scale down disk voltage until a request's desired response time cannot be guaranteed (see Step 12) or the supply voltage approaches the threshold voltage. Consequently, DCAPS adaptively reduces the supply voltage while making the best effort to complete all the disk requests before their desired response time.

Table 1
Disk parameters of the simulated parallel disk system.

Number of disks	16
Block size	1 KB
Number of tracks per cylinder	11
Number of cylinder per disk	1435
Capacity	300 GB
Average seek time	8 ms
Spindle speed	7200 RPM
$V_{\max}$	3.3 V
$V_{\min}$	1.2 V
Three-level multiple voltages	1.2, 2.4, and 3.3 V

5. Experimental evaluation

To evaluate the performance of the DCAPS strategy in an efficient way, we simulated a parallel disk system with all the functions that are necessary to implement our system. Table 1 summarizes important parameters used to resemble real world disks. In addition, we implemented a data-partitioning algorithm to optimize parallelism degrees of large disk I/O requests. We will first compare the performance of a parallel disk system with DCAPS against that of another system without employing DCAPS. In this study, a system that does not apply DCAPS is a standard parallel disk system with a fixed voltage level. We will then study effects of varying arrival rates, data size, and disk bandwidth on the performance of the two disk systems. Next, we will compare and evaluate the two disk systems based on varying the voltage. Finally, we will also analyze the performance impacts of parallelism degrees on the parallel disk systems.

In our simulation experiments, we made use of the following three performance metrics to demonstrate the effectiveness of the DCAPS scheme.

(1) Satisfied ratio is a fraction of total arrived disk requests that are found to be finished within their desired response times.
(2) Energy consumption is the total energy consumed by the parallel disk systems.
(3) Energy conservation ratio.
(4) Average overhead measured in seconds.

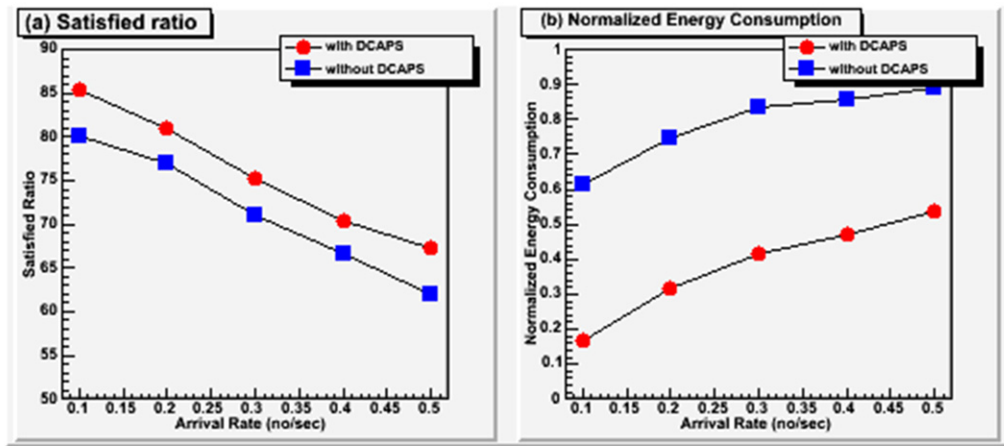

Fig. 3. Impact of request arrival rate on satisfied ratio and normalized energy consumption when disk bandwidth is 30 MB/s.

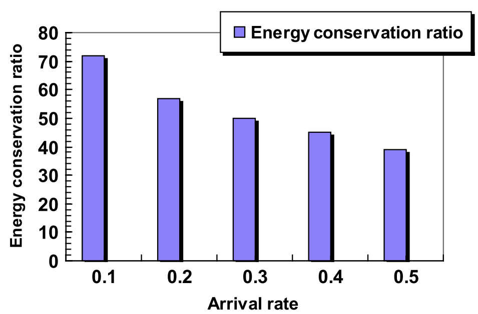

Fig. 4. Impact of request arrival rate on energy conservation ratio.

5.1. Impact of arrival rate

In this experiment, we evaluate the impacts of disk request arrival rate on the satisfied ratio and normalized energy consumption. We compare a DCAPS-enabled parallel disk system with a non-DCAPS-enabled parallel disk system where voltage supply levels are fixed. We vary the arrival rate from 0.1 to 0.5 No./s with an increment of 0.1 No./s.

Fig. 3 plots the satisfied ratios and normalized energy consumption of the two evaluated parallel disk systems. Fig. 3(a) shows that the non-DCAPS-enabled system's satisfied ratio is slightly lower than that of the DCAPS-enabled system. This intriguing result can be explained by the fact that non-DCAPS-enabled system attempts to conserve energy by placing disks into the standby mode at extra power-state-transition overhead. Rather than relying on power-state-transitions, DCAPS conserves energy by lowering the power voltage. Fig. 3(b) shows that the DCAPS algorithm significantly reduces the energy consumption in the parallel disk system by up to 71% with an average of 52.6%. The improvement in energy efficiency can be attributed to the fact that DCAPS reduces the disk supply voltages while making an effort to completes requests before their desired response times.

Fig. 4 shows the energy conservation ratios offered by DCAPS when the arrival rate is increased. The results plotted in Fig. 4 confirm increasing I/O load leads to decreased energy-conservation rate. This trend is reasonable, partially because heavy I/O load forces DCAPS to boost disk voltages to complete requests before their desired response times. Increasing voltage levels in turn reduces chances of saving energy in parallel disk systems.

Fig. 5 shows the energy-conservation overhead introduced by DCAPS. Fig. 5 indicates that increasing arrival rates leads to increased energy conservation overhead, which gives rise to a low energy-conservation rate. In addition, Fig. 5 shows that the DCAPS-enable system has higher energy-conservation overheads than those of the non-DCAPS system, because DCAPS pays extra overhead to adjust voltage levels.

5.2. Impact of disk bandwidth

In the second experiment we measure performance impacts of disk bandwidth on the energy efficiency of DCAPS. We vary varying the disk bandwidth from 10 to 50 MB/s, with an increment of 10 MB/s. Fig. 6(a) clearly shows that as one increases the disk bandwidth, the satisfied ratios of the two parallel disk systems gradually climb up. This performance trend can be explained by the fact that increasing disk bandwidth results in shortened data

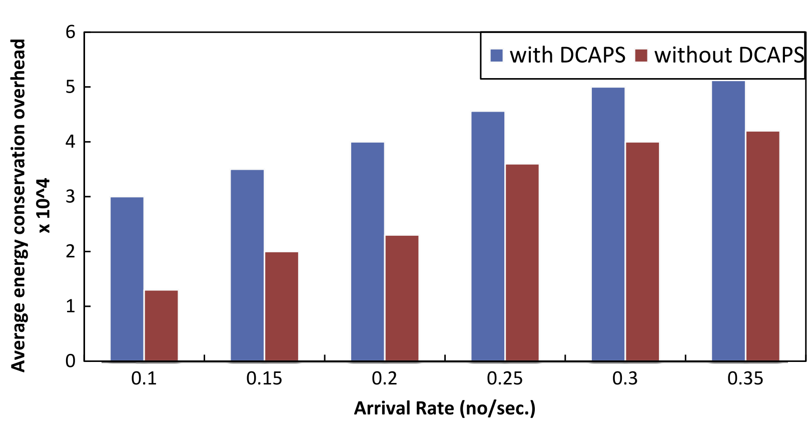

Fig. 5. Impact of arrival rate on energy conservation overhead.

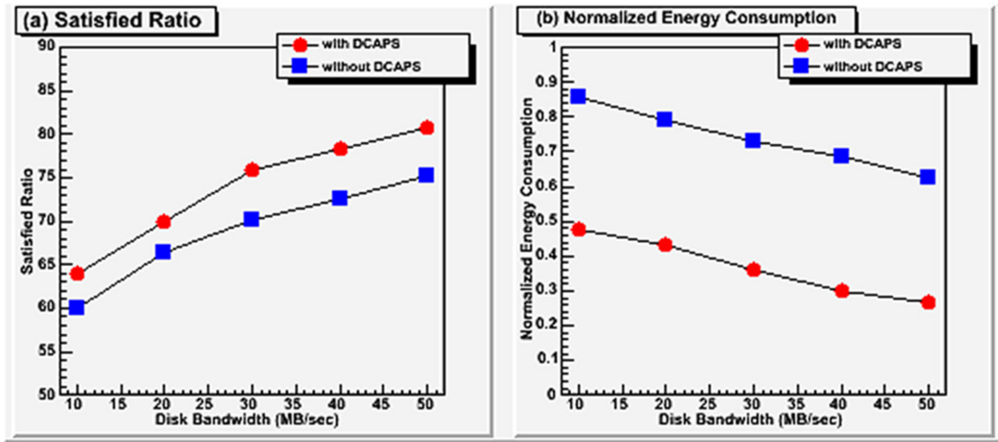

Fig. 6. Impact of disk bandwidth on satisfied ratio and normalized energy consumption. Request arrival rate = 0.3 no/s.

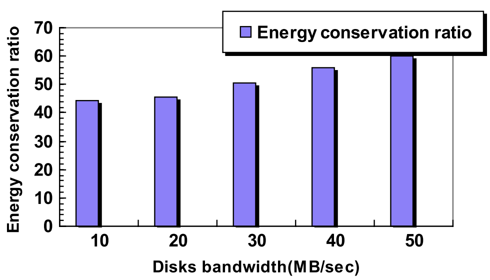

Fig. 7. Impact of disk bandwidth on energy conservation ratio.

transfer times, which in turn lead to decreased I/O processing times of disk requests. Consequently, an increasing number of disk requests can be accomplished within their desired response times. Fig. 6(b) shows that the normalized energy consumption decreases as the disk bandwidth is increases. This is mainly because energy consumption is a product of power and I/O processing time, which is noticeably reduced with the increasing disk bandwidth.

Interestingly, Fig. 7 shows that the energy conservation ratio continues increasing as the disk bandwidth increases. This effect is apparent because the shortened I/O processing times allows DCAPS to leverage the scaled-down voltages to substantially conserve the energy consumption by up to 60% with an average of 51%.

Similar to Fig. 5, Fig. 8 confirms that DCAPS reduces energy dissipation in parallel disks at the expense of energy-conservation overheads. Fig. 8 shows that the energy-conservation overhead for high-speed disks is larger than those of low-speed disks. Although DCAPS introduces larger energy-conservation overheads for disks with high bandwidth, DCAPS still can offer better energy-conservation rate for high-speed parallel disks.

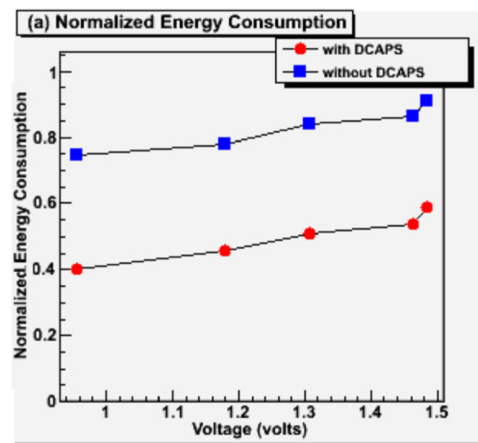

Fig. 9. The impact of minimum voltage level on energy efficiency of DCAPS.

5.3. Impact of the minimum supply voltage level

In the third experiments, we focus on the impact of the minimum voltage supply level on energy efficiency and the overhead of DCAPS applied in parallel disk systems. Fig. 9 shows that when one increases the minimum supply voltage, energy dissipations in both parallel disk systems go up. Fig. 10 illustrates that the energy conservation ratio slightly drops as the minimum voltage increases. The implication of this result is that parallel disk systems can take greater advantage from DCAPS when the minimum voltage level is lower.

Fig. 11 illustrates the impacts of minimum voltage level on energy-conservation overhead. The results show that the energy-

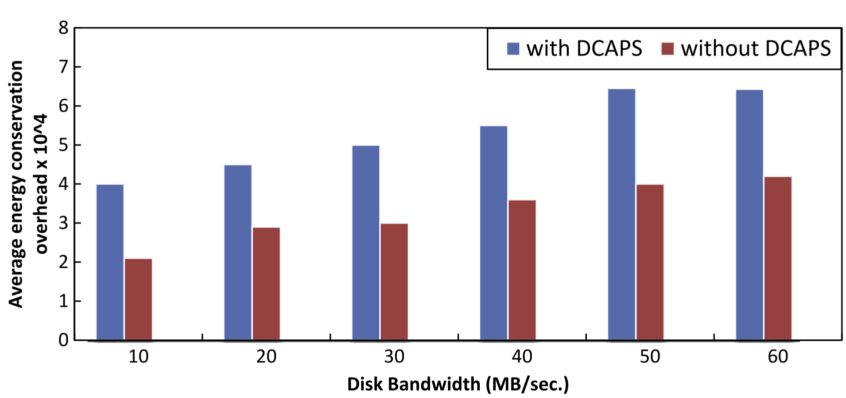

Fig. 8. Impact of disk bandwidth on energy conservation overhead.

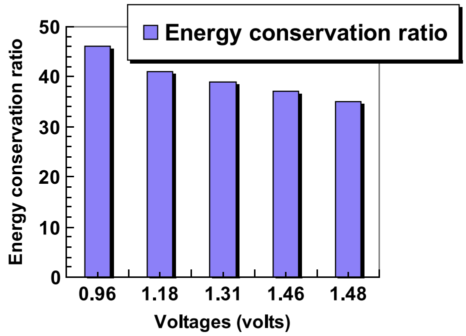

Fig. 10. The impact of minimum voltage level on energy-conservation ratio.

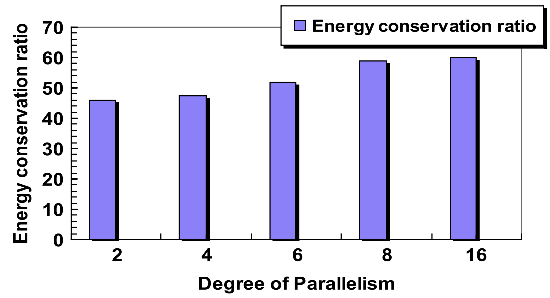

Fig. 13. Impact of parallelism degree voltage on energy conservation ratio.

conservation overhead is slightly increased when the minimum voltage is increased from 1 to 1.5 V. The increased energy-conservation overhead partially contributes to the decrease in energy-conservation ratio when the minimum voltage goes up.

5.4. Impact of parallelism degrees

Recall that two types of I/O parallelisms are inter-request and intra-request parallelism. In this experiment let us consider the intra-request parallelism, because the inter-request parallelism can be treated as a special case of the intra-request parallelism. Please note that DCAPS can be readily extended to deal with inter-request parallelisms. In this experiment, we vary the parallelism degree from 2 to 16. Fig. 12(a) shows that the performance in terms of satisfied ratio is improved with the increasing parallelism degree. The rationale behind this trend is that increasing parallelism degrees allow a large number of disk requests to simultaneously served by multiple disks in the systems. Thus, high parallelism degrees help to substantially enhance throughput of the parallel disk systems, thereby making more requests to be completed within their desired response times.

It is intriguing to observe from Fig. 12(b) that the high parallelism degrees lead to low energy consumption. We attribute this performance trend to positive impacts of the large striping widths substantially reduce the response times, which in turn make it possible to efficiently scale down the supply voltage. Fig. 13 reveals that the energy conservation ratio increases with the increasing value of parallel degree, because high parallelism degrees help to substantially enhance system throughput, which provides space for further energy savings.

Fig. 14 shows that a high parallelism degree leads to low energy-conservation overhead, which in turn contributes to the

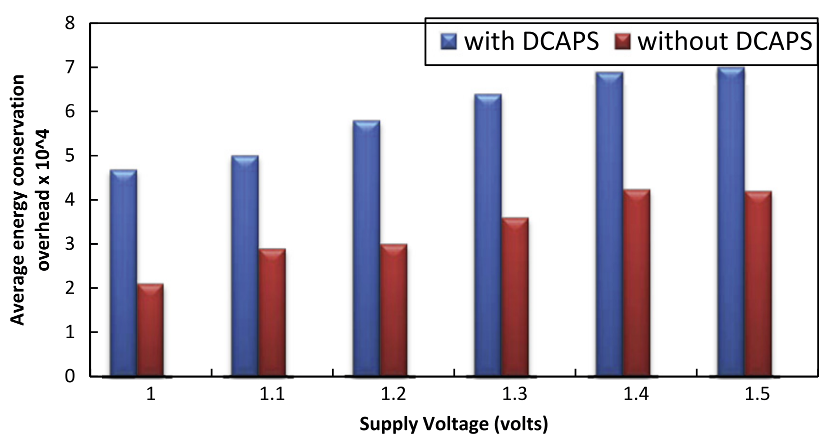

Fig. 11. The impact of minimum voltage level on energy-conservation overhead in DCAPS.

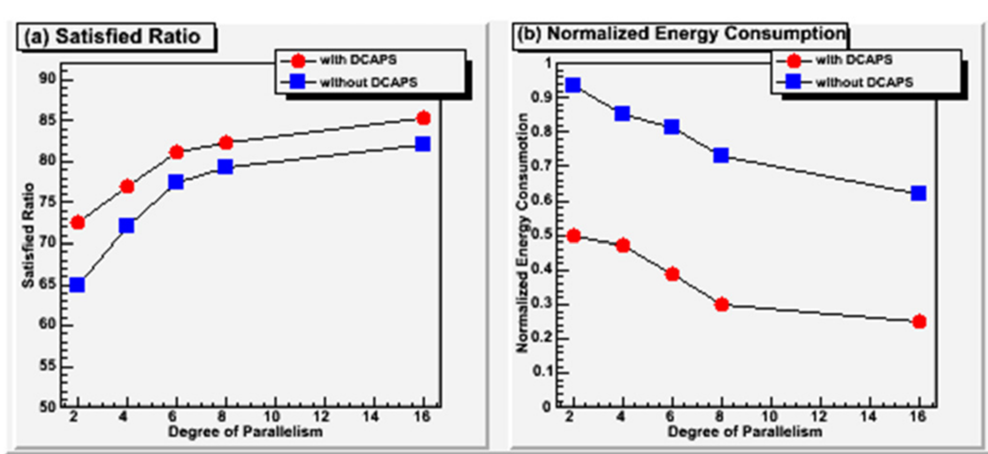

Fig. 12. Impact of parallelism degree on satisfied ratio and normalized energy consumption. Request arrival rate = 0.3 no/s, disk bandwidth = 30 MB/s.

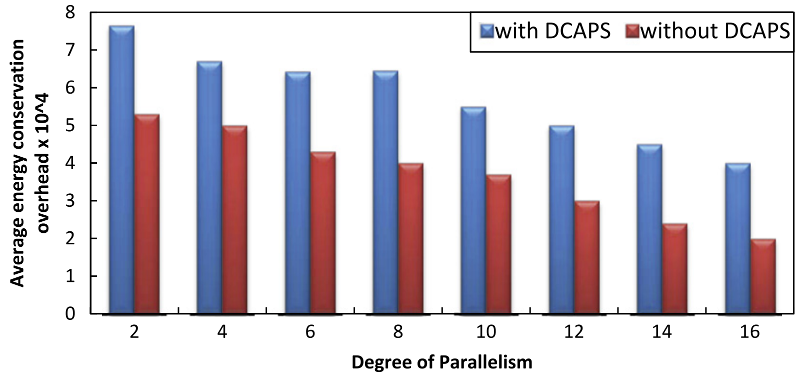

Fig. 14. The impact of parallelism degree on energy conservation overhead.

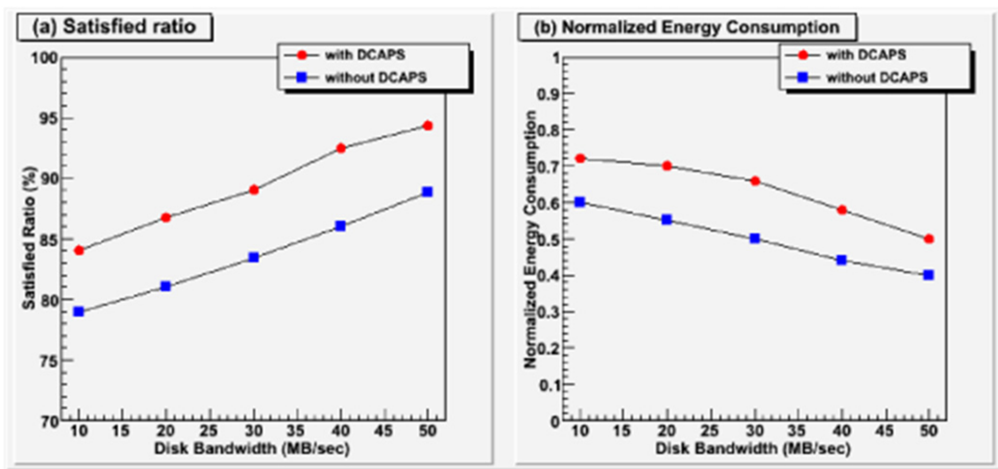

Fig. 15. Impact of disk bandwidth. Sparse Cholesky.

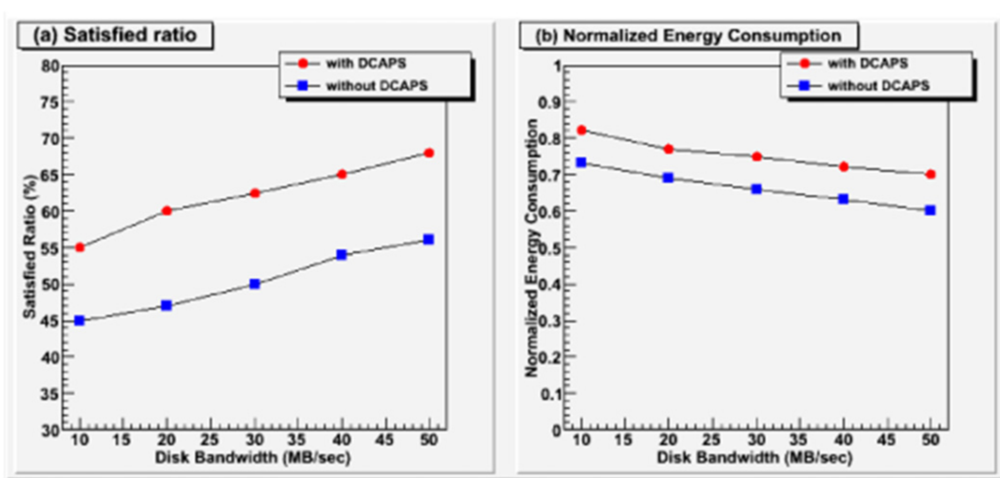

Fig. 16. Impact of disk bandwidth. LU decomposition.

high energy-efficiency of DCAPS (see Fig. 13). Fig. 14 suggests that energy-conservation overhead caused by DCAPS can be reduced by increasing parallelism degrees in large-scale parallel disk systems.

5.5. Real I/O intensive applications

To validate the results from the synthetic workload, we used disk traces of real world I/O-intensive applications to evaluate the performance of our strategy in comparison to the alternative one. We chose two common I/O-intensive applications: LU decomposition [49] and Sparse Cholesky [50], which have various I/O patterns. The LU decomposition application tries to compute the dense LU decomposition of an out-of-core matrix, whereas the sparse Cholesky application is used to calculate Cholesky decomposition for sparse, symmetric positive-definite matrices.

First, we evaluate the impact of disk bandwidth on the two real applications. In this group of experiments, the disk bandwidth varies from 10 to 50 MB/s with increment of 10MB/s. Figs. 15 and 16 depict the disk bandwidth effect on both the DCAPS-enabled and non-DCAPS-enabled systems when LU decomposition

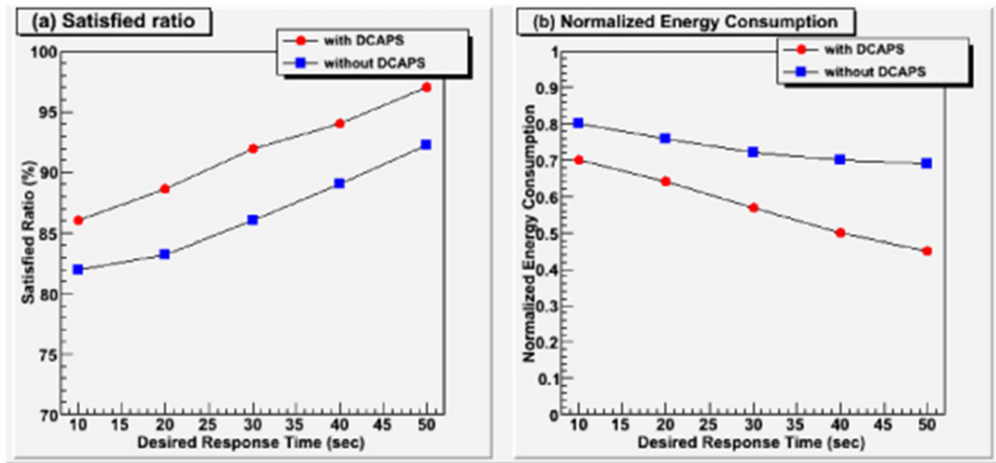

Fig. 17. Impact of desired response time. Sparse Cholesky.

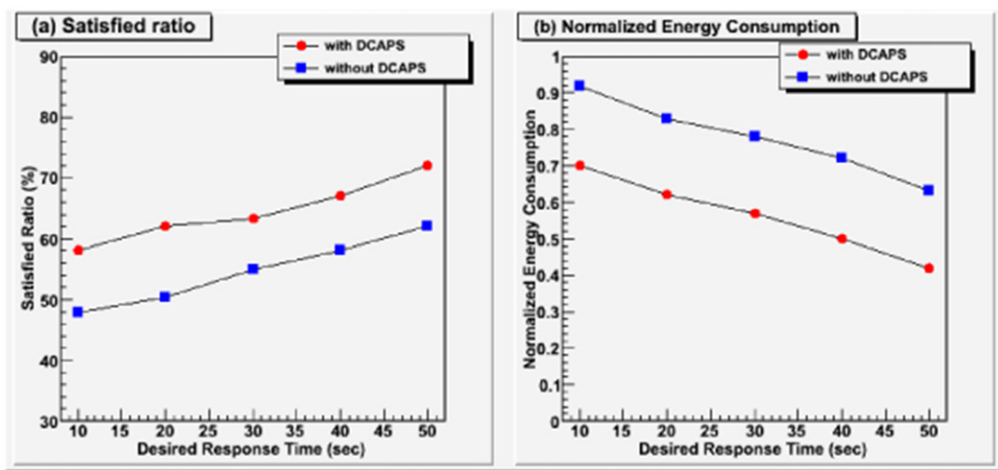

Fig. 18. Impact of desired response time. LU decomposition.

and sparse Cholesky are used as the workload. Figs. 16(a)–(c) and 17(a)–(c) illustrate that for all the cases we have examined, the increased disk bandwidth is a driving force of the improved satisfied ratios and the decreased of energy consumption. These results are consistent with the disk bandwidth impact observed in Figs. 6 and 7 (see Section 5.2).

Not surprisingly, the disk bandwidth effect on the two disk systems relies in part on the data-intensive applications. More specifically, the energy conversation ratio achieved by DCAPS is much more pronounced for the workload with the sparse Cholesky application as compared to the workload with LU decomposition. This can be attributed relatively to the small arrival rate and data size of the workload with sparse Cholesky compared to LU decomposition, which induces lower energy conservation.

Now we evaluate the impact of desired response time on the two real applications. In this group of experiments, the desired response time varies from 10 to 50 s with increment of 10 s. Throughout this set of experiments; the disk bandwidth was set to 30 MB/s. The results for the LU decomposition and sparse Cholesky applications are plotted in Figs. 17 and 18, respectively. Figs. 17(a) and 18(a) show that for the two I/O-intensive applications, satisfied ratios yielded by DCAPS are higher than those of the alternative strategy. This observation is especially true when the desired response time is long. Figs. 17(b) and 18(b) demonstrate that as the desired response time increases, the normalized energy consumption decreases conspicuously. This is mainly because when the desired response time is enlarged, the possibility of reducing the supply voltage without violating timing constraints increases.

By comparing the two applications, we observe that LU decomposition is more sensitive to desired response time while sparse Cholesky is less sensitive. The cause of this performance difference can be explained as follows. The disk request arrival rate and data size are two dominant factors for satisfied ratio and average security level. High arrival rate and large data size of disk requests give rise to LU decomposition's small satisfied ratios and reduction to high-energy consumption (See Figs. 17(a) and 18(b)). Consequently, the increase in desired response time provides more opportunities for LU decomposition to improve both satisfied ratio and energy conservation.

Now it is time to compare the performance of the two strategies based on the supply voltage. In this group of experiments, the supply voltage varies from 1 to 1.5 V. Throughout this set of experiments; the disk bandwidth was set to 30 MB/s. Figs. 19 and 20 plot the results for the LU decomposition and sparse Cholesky applications, respectively. Figs. 19 and 20 reveal that DCAPS can reduce energy consumption caused by the two data-intensive applications.

6. Conclusions and future work

Parallel disk systems play an important role in achieving high-performance for data-intensive applications, because the high parallelism and scalability of parallel disk systems can alleviate the

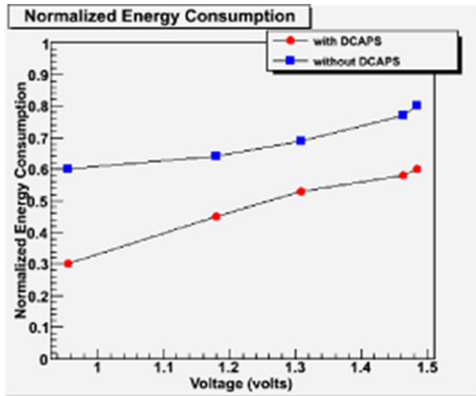

Fig. 19. Impact of minimum supply voltage on sparse Cholesky.

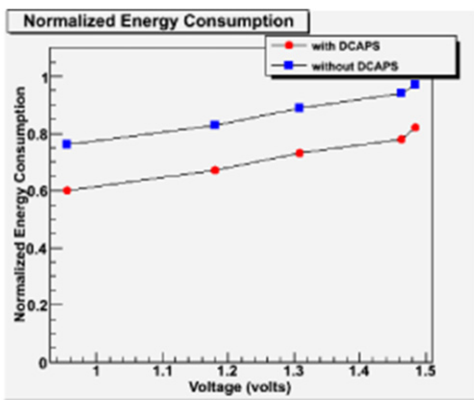

Fig. 20. Impact of minimum supply voltage on LU decomposition.

disk I/O bottleneck problem. However, growing evidence shows that a substantial amount of energy is consumed by parallel disk systems in data centers. It is therefore highly desirable to design energy-efficient parallel disk systems by extensively investigate energy-conservation software techniques. Adaptively conserving energy in parallel disk systems becomes particularly critical for data-intensive applications in which disk requests need to be completed within specified response times or desired response times. In this paper, we focused on the design of novel parallel disk systems that can achieve both great energy efficiency and high guarantee of specified performance. Specifically, we developed an adaptive energy-conserving strategy, which dynamically scaled down disk voltage supplies to the most appropriate levels, thereby significantly reducing energy dissipation in parallel disk systems. The experimental results have confirmed that our scheme can achieve up to 70% energy savings compared with standard parallel disk systems with fixed supply voltage.

Our approach is the first technique of its kind designed exclusively for energy-efficient parallel disk systems aiming to guarantee specified performance of data-intensive applications. As a future direction, we will implement a prototype of an energy-efficient storage system where the DCAPS scheme is incorporated. In addition, we will test a set of data-intensive applications on the energy-efficient storage systems to evaluate the energy efficiency and performance impact of DCAPS on real-world storage systems.

Acknowledgments

The work reported in this paper was supported by the US National Science Foundation under Grants CCF-0845257 (CAREER), CNS-0917137 (CSR), CNS-0757778 (CSR), CCF-0742187 (CPA), CNS-0831502 (CyberTrust), CNS-0855251 (CRI), OCI-0753305 (CI-TEAM), DUE-0837341 (CCLI), DUE-0830831 (SFS), and CNS-1048432.

References

[1] S. Rajasekaran, Selection algorithms for parallel disk systems, in: Proc. Int'l Conf. High Performance Computing, Dec. 1998, pp. 343–350.
[2] X. Qin, Performance comparisons of load balancing algorithms for I/O-intensive workloads on clusters, J. Netw. Comput. Appl. (2007).
[3] X. Qin, Design and analysis of a load balancing strategy in data grids, Future Gener. Comput. Syst. 23 (1) (2007) 132–137.
[4] D. Avitzour, Novel scene calibration procedure for video surveillance systems, IEEE Trans. Aerosp. Electron. Syst. 40 (3) (2004) 1105–1110.
[5] C. Chang, B. Moon, A. Acharya, C. Shock, A. Sussman, J. Saltz, Titan: a high-performance remote-sensing database, in: Proc. 13th Int'l Conf. Data Eng., Apr 1997.
[6] Y. Xue, et al., Workload and task management of grid-enabled quantitative aerosol retrieval from remotely-sensed data, Future Gener. Comput. Syst. 20 (2009) 589–598.
[7] T. Sumner, M. Marlino, Digital libraries and educational practice: a case for new models, in: Proc. ACM/IEEE Conf. Digital Libraries, June 2004, pp. 170–178.
[8] J. Trnkoczy, V. Stankovski, Improving the performance of federated digital library services, Future Gener. Comput. Syst. (2008).
[9] A. Breckenridge, L. Pierson, S. Sanielevici, J. Welling, R. Keller, U. Woessner, J. Schulze, Distributed, on-demand, data-intensive and collaborative simulation, analysis, Future Gener. Comput. Syst. 19 (6) (2003) 849–859.
[10] T. Tanaka, Configurations of the solar wind flow and magnetic field around the planets with no magnetic field: calculation by a new MHD, J. Geophys. Res. (1993) 17251–17262.
[11] Power, heat, and sledgehammer. White paper Maximum institution Inc., 2002. http://www.max-t.com/downloads/whitepapers/SledgehammerPowerheat20411.pdf.
[12] B. Moore, Taking the data center power and cooling challenge. Energy User News, August 27th, 2002.
[13] A. Manzanares, X. Qin, X.-J. Ruan, S. Yin, PRE-BUD: prefetching for energy-efficient parallel I/O systems with buffer disks, ACM Trans. Storage 7 (1) (2011) Article 3 June.
[14] A. Chandra, W. Gong, P. Shenoy, Dynamic resource allocation for shared data centers using online measurements, in: Proc. the ACM SIGMETRICS Int'l Conf. Measurement and Modeling of Computer Systems, 2003.
[15] M. Nijim, X. Qin, T. Xie, Modeling and improving security of a local disk system for write-intensive workloads, ACM Trans. Storage 2 (4) (2007) 400–423.
[16] J. Coffman, M. Hofri, Queueing models of secondary storage devices, in: Hideaki Takagi (Ed.), Stochastic Analysis of Computer and Comm. Sys., North-Holland, 1990.
[17] D. Jacobson, J. Wilkes, Disk scheduling algorithms based on rotational position, Technical Report, HPL-CSP-91-7, February 1991.
[18] M. Seltzer, P. Chen, J. Ousterhout, Disk scheduling revisited, in: Proc. USENIX Technical Conf., 1990, pp. 313–323.
[19] P.S. Yu, M.S. Chen, D.D. Kandlur, Grouped sweeping scheduling for DASD-based multimedia storage management, ACM Multimedia Syst. 1 (3) (1993) 99–109.
[20] Quantum Corp., Storage Basics, ORCA.
[21] P.J. Denning, Effects of schuling on file memory operations, in: Proc. AFIPS Conf., April 1967, pp. 9–21.
[22] R. Reist, S. Daniel, A continuum of disk scheduling algorithms, ACM Trans. on Computer Sys. (1987) 77–92.
[23] P. Scheuermann, G. Weikum, P. Zabback, Data partitioning and load balancing in parallel disk systems, VLDB Journal 7 (1998) 48–66.
[24] V. Pandey, J. Zhou, R. Bianchini, DMA-aware memory energy management, in: Proc. The 12th International Symposium on High-Performance Computer Architecture, 2006, pp. 133–144.
[25] H. Ou, A case study: performance evaluation of a DRAM-based solid state disk, in: Frontier of Computer Science and Technology Workshop, 2007, pp. 57–60.
[26] M. Kallahalla, P.J. Varman, Improving parallel-disk buffer management using randomized writeback, in: Proc. Int'l Conf. Parallel Processing, Aug. 1998, pp. 270–277.
[27] S. Rajasekaran, X. Jin, A practical realization of parallel disks parallel processing, in: Proc. Int'l Workshop Parallel Processing, Aug. 2000, pp. 337–344.
[28] D. Kotz, C. Ellis, Cashing and writeback policies in parallel file systems, in: Proc. IEEE Symp. Parallel and Distributed Processing, Dec. 1991, pp. 60–67.
[29] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, H. Franke, Dynamic speed control for power management in server class disks, in: Proc. 30th Annual Int'l Symp. Computer Architecture, 2003.
[30] S. Sankar, S. Gurumurthi, M. Stan, Sensitivity based power management of enterprise storage systems, in: Proc. IEEE International Symposium on Telecommunication Systems, 2008, pp. 1–10.
[31] E.V. Carrera, E. Pinheiro, R. Bianchini, Conserving disk energy in network servers, in: Proc. Int'l Conf. Supercomputing, 2003, pp. 86–97.
[32] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, H. Fanke, DRPM: dynamic speed control for power management in server class disks, in: Proc. Int'l Symp. of Computer Architecture, June 2003, pp. 169–179.
[33] A.E. Papathanasiou, M.L. Scott, Power-efficient server-class performance from arrays of laptop disks, in: WIP USENIX 2004 Annual Technical Conference, Boston, 2004.
[34] A. Manzanares, A. Roth, X.-J Ruan, S. Yin, M. Nijim, X. Qin, Conserving energy in real-time storage systems with I/O burstiness, ACM Trans. Embedded Comput. Syst. 9 (3) (2010).

[35] Q. Zhu, F.M David, C.F. Devaaraj, Z. Li, Y. Zhou, P. Cao, Reducing energy consumption of disk storage using power aware cache management, in: Proc. High Performance Computer Framework, 2004.
[36] M. Lee, S. Euiseong, Joonwon Lee, Jin-soo Kim, PABC: power-aware buffer cache management for low power consumption, IEEE Trans. Comput. 56 (4) (2007) 488–501.
[37] Q. Zhu, Y. Zhou, Power aware storage cache management, IEEE Trans. Comput. 54 (5) (2005) 587–602.
[38] S.W. Son, M. Kandemir, Energy aware data perfecting for multi-speed disks, in: Proc. ACM International Conference on Computing Frontiers, Ischia, Italy, May 2006.
[39] A. Manzanares, K. Bellam, X. Qin, A prefetching scheme for energy conservation in parallel disk systems, in: Proc, IEEE International Symposium on Parallel and Distributed Processing, 2008, pp. 1–5.
[40] Y. Guo, P. Narayanan, M. Bennaser, S. Chheda, C. Moritz, Energy-efficient hardware data prefetching, in: IEEE Transactions on Very Large Scale Integration Systems, 2011, pp. 250–263.
[41] S.W. Son, M. Kandemir, A. Choudhary, Software-directed disk power management for scientific applications, in: Proc. Int'l Symp. Parallel and Distr. Processing, April 2005.
[42] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, H. Fanke, DRPM: dynamic speed control for power management in server class disks, in: Proc. Int'l Symp. of Computer Architecture, June 2003, pp. 169–179.
[43] D.P. Helmbold, D.D.E. Long, T.L. Sconyers, B. Sherrod, Adaptive disk spin-down for mobile computers, Mob. Netw. Appl. 5 (4) (2000) 285–297.
[44] P. Krishnan, P. Long, J. Vitter, Adaptive disk spindown via optimal rent-to-buy in probabilistic environments, in: Proc. Int'l Conf. Machine Learning, July 1995, pp. 322–330.
[45] T. Xie, Y. Sun, No more energy-performance trade-off: a new data placement strategy for RAID-structured storage systems, in: The 14th Annual International Conference on High Performance Computing, HiPC 2007, in: Lecture Notes in Computer Science (LNCS), vol. 3834, 2007, pp. 35–46. Goa, India, December 18–21.
[46] V. Kaenel, P. Mackson, M. Degrauwe, A voltage reduction technique for battery operated systems, IEEE J. Solid-State Circuits 25 (1990) 1136–1140.
[47] T. Kuroda, et al., Variable supply voltage scheme for low power high speed CMOS digital design, IEEE J. Solid-State Circuits 33 (1998) 454–462.
[48] Ali Manzak, Chaitali Chakrabarti, Variable voltage task scheduling algorithms for minimization energy/power, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 11 (2) (2003).
[49] B. Hendrickson, D. Womble, The torus-wrap mapping for dense matrix calculations on massively parallel computers, SIAM J. Sci. Comput. 15 (5) (1994).
[50] A. Acharya, et al. Tuning the performance of I/O-intensive parallel applications, in: Proc. of the 4th IOPADS, Philadelphia, PA, May 1996, pp 15–27.

Mais Nijim is an Assistant Professor in the Department of Electrical Engineering and Computer Science Department at Texas A&M University-Kingsville. She received her master degree in computer science from New Mexico State University in 2005. Dr. Nijim received the Ph.D. in Computer Science from New Mexico Tech in 2007. Her research interests include parallel and distributed systems, real time computing, storage systems, fault tolerance, and performance evaluation. Her research is funded from National Science Foundation and Dotterweigh College of Engineering at Texas A&M University Kingsville. She has been on the program committees of various international conferences.

Xiao Qin is an associate professor at the department of computer science and software engineering at Auburn University. He received the B.S. and M.S. degrees in Computer Science from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree in Computer Science from the University of Nebraska-Lincoln, Lincoln, in 1992, 1999, and 2004, respectively. For three years, he was an Assistant Professor with the New Mexico Institute of Mining and Technology, Socorro, NM. Currently, he is an Associate Professor with the Department of Computer Science and Software Engineering, Auburn University, Auburn, AL. His research interests include parallel and distributed systems, storage systems, fault tolerance, real-time systems, and performance evaluation. His research is supported by the U.S. National Science Foundation, Auburn University, and Intel Corporation. Dr. Qin was a recipient of the U.S. National Science Foundation (NSF) Computing Processes and Artifacts Award and the NSF Computer System Research Award in 2007 and the NSF CAREER Award in 2009. He was a Subject Area Editor of the IEEE Distributed System Online in 2000–2001. He has been on the program committees of various international conferences, including IEEE Cluster, IEEE International Performance, Computing, and Communications Conference, and International Conference on Parallel Processing.

Meikang Qiu [SM'07] received the B.E. and M.E. degrees from Shanghai Jiao Tong University, China. He received the M.S. and Ph.D. degrees of Computer Science from University of Texas at Dallas in 2003 and 2007, respectively. He had worked at Chinese Helicopter R&D Institute and IBM. Currently, he is an assistant professor of ECE at University of Kentucky. He is an IEEE Senior member. He has published 140 papers, including 15 IEEE/ACM Transactions papers. He is the recipient of the ACM Transactions on Design Automation of Electronic Systems (TODAES) 2011 Best Paper Award. He also received three other best paper awards (IEEE EUC'09, IEEE/ACM GreenCom'10, and IEEE CSE'10) and one best paper nomination. He also holds 2 patents and has published 3 books. He has also been awarded SFFP Air Force summer faculty in 2009. He has been on various chairs and TPC members for many international conferences. He served as the Program Chair of IEEE EmbeddCom'09 and EM-Com'09. His research interests include embedded systems, computer security, and wireless sensor networks.

Kenli Li received the Ph.D. degree in computer science from Huazhong University of Science and Technology, China, in 2003, and the M.S. degree in mathematics from Central South University, China, in 2000. He has been a visiting scholar at University of Illinois at Champaign and Urbana from 2004 to 2005. He is now a professor of Computer science and Technology at Hunan University and the deputy director of National Supercomputing Center in Changsha. He is a senior member of CCF and has published more than 80 peer reviewed papers. His major research contains parallel computing, Grid and Cloud computing, and DNA computer.