TRACER: A Trace Replay Tool to Evaluate Energy-Efficiency of Mass Storage Systems

Zhuo Liu[1], Fei Wu[1 *], Xiao Qin[2], Changsheng Xie[1], Jian Zhou[1], and Jianzong Wang[1]

[1]Wuhan National Laboratory for Optoelectronics
[1]Key Laboratory of Data Storage Systems,Ministry of Education of China
[1]School of Computer Sci. and Tech., Huazhong University of Sci. and Tech., Wuhan, Hubei 430074, P.R.China
[2]Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849,USA
[2]xqin@auburn.edu http://www.eng.auburn.edu/∼xqin
[*]Corresponding Author: wufei@hust.edu.cn

***Abstract*—Improving energy efficiency of mass storage systems has become an important and pressing research issue in large HPC centers and data centers. New energy conservation techniques in storage systems constantly spring up; however, there is a lack of systematic and uniform way of accurately evaluating energy-efficient storage systems and objectively comparing a wide range of energy-saving techniques. This research presents a new integrated scheme, called TRACER, for evaluating energy-efficiency of mass storage systems and judging energy-saving techniques. The TRACER scheme consists of a toolkit used to measure energy efficiency of storage systems as well as performance and energy metrics. In addition, TRACER contains a novel and accurate workload-control module to acquire power varying with workload modes and I/O load intensity. The workload generator in TRACER facilitates a block-level trace replay mechanism. The main goal of the workload-control module is to select a certain percentage (e.g., anywhere from 10% to 100%) of trace entries from a real-world I/O trace file uniformly and to replay filtered trace entries to reach any level of I/O load intensity. TRACER is experimentally validated on a general RAID5 enterprise disk array. Our experiments demonstrate that energy-efficient mass storage systems can be accurately evaluated on full scales by TRACER. We applied TRACER to investigate impacts of workload modes and load intensity on energy-efficiency of storage devices. This work shows that TRACER can enable storage system developers to evaluate energy efficiency designs for storage systems.**

***Index Terms*—Load Intensity, Energy-efficiency, Evaluation.**

I. Introduction

When an increasing number of information infrastructures are in normal operation, all available evidence shows that power dissipation in data centers is far beyond imagination. According to statistics, storage subsystems constitute a 27% proportion of total power consumption in an entire data center[24]. It is worth noting that this 27% rate does not include energy used for cooling storage components. Even worse, the primary energy demand of storage systems is increasing by 60% every year[24]. Storage systems contribute a significant portion of the total initial cost and energy usage of modern supercomputer and data centers. Energy-conservation techniques are increasingly becoming environmental and financial concerns in the design of mass storage systems. Data center developers need to pay particular attention to building an energy-efficient storage systems. As a result, a wide spectrum of energy-conversation techniques for storage systems spring up. There are lots of representative energy-saving techniques in storage systems including MAID [6], PDC [16], DRPM [9], eRAID [12], and BUD [18]. Unfortunately, there is a lack of systematic and uniform way of accurately evaluating energy-efficient storage systems and objectively comparing a wide range of energy-saving techniques. Without such a method of measuring energy efficiency, it is difficult for system designers to choose among various energy-saving techniques for storage devices. Although there are several enterprise standards for power evaluation of computer systems, most of the standards are applicable to CPU-intensive applications. Integrated, portable, and practical energy-efficiency measurement tools are still in their infancy. To address this problem, we implemented a new integrated framework - called TRACER - for the evaluation of energy-conservation techniques in large-scale storage systems.

TRACER is a load-controllable energy-efficiency evaluation framework, which facilitates a trace replay mechanism for mass storage systems. TRACER consists of performance and energy metrics as well as a toolkit used to measure energy efficiency of storage systems. Additionally, TRACER contains a novel and accurate workload-control module to acquire power varying with workload modes and I/O load intensity. Relying on existing trace-collecting and replaying techniques, the workload-control module can replay an I/O trace file according to a dynamic configuration of load generators and their parameters (e.g., 10%, 20%, 70% load levels).

To seamlessly integrate the measurement of power consumption with I/O throughput in the context of storage systems, we introduced MBPS/Kilowatt and IOPS/Watt as main metrics in TRACER for the evaluation of energy-efficient storage systems. The new metrics allow us to objectively analyze correlations between I/O performance and energy efficiency in any storage system. This work shows that TRACER can enable storage system developers to accurately evaluate performance and energy efficiency of mass storage systems on full scales. Applying TRACER to evaluate energy efficiency of a general RAID5 enterprise disk array, we observed that power consumption of a storage system is closely correlated with I/O throughput performance and workload affecting factors (e.g., request size, random rate and read rate) representing various I/O-intensive applications. To measure energy efficiency of a storage system in active mode, I/O load intensity and characteristics of workload must be taken into account.

Our contributions of TRACER are summarized as follows:

1) TRACER is a fully integrated framework that consistently and systematically evaluate energy-efficiency of mass storage systems. The flexibility of TRACER is high in the sense that TRACER can be implemented in a wide range of storage environments where energy-conservation techniques are employed. The TRACER framework includes a trace-depository collector coupled with a trace replay tool and, therefore, TRACER allows systems developers to compare among various energy-saving techniques integrated into modern storage systems.
2) A simple yet effective load control scheme was implemented in TRACER to generate workload by manipulating a given trace file in accordance with a specified intensity level. We measured the accuracy of the load control scheme; our experiments show that TRACER is highly accurate when it comes to the workload control of real-world traces as well as those collected by us.
3) To demonstrate the usage of TRACER, we made use of TRACER to measure the power consumption of two RAID systems. While the first RAID array was built with hard disk drives (HDD), the second one is comprised of solid state disks (SSD). Results obtained by TRACER show that the HDD-based RAID array is more energy-efficient under higher I/O load with more I/O sequential access patterns (see Section VI for more details on effects of read/write ratio and request size). The SSD-based RAID system is more energy-efficient than its HDD-based counterpart. Moreover, the energy efficiency of the SSD-based RAID system is greatly affected by the read/write ratio. These experimental results produced by TRACER provides constructive information on how to develop future energy-efficient storage systems. For example, system developers can rely on results obtained from TRACER to incorporate I/O load balancing mechanisms in mass storage systems to reduce power consumption.

The remainder of this paper is organized as follows: First, we present the architecture of TRACER. Then, we outline the implementation of a trace reply tool in TRACER. After describing a testbed and experimental settings, we validate the accuracy and usage of TRACER in the context of RAID systems.

II. Related Work

The goal of TRACER is to facilitate a uniform way of evaluating performance and energy-efficiency of modern storage systems. In what follows, we highlight three research fields closely related to energy-efficient storage systems.

A. Benchmarks and Metrics of Energy-Efficient Computer Systems

Benchmarks representing various characteristics of practical applications are commonly used to measure the performance of computer systems. In the past few years, an array of new benchmarks have been developed to evaluate energy-saving techniques in computer systems. Energy Star[1] provides standards for energy efficient consumer products as well as computer systems. For example, energy efficiency of electric appliances are required to be over 80% according to the Energy Star's standard. The focus of the Energy Star's standard is energy saving rather than performance. Unlike Energy Star, SPECpower_ssj benchmarks consist of a group of applications used to evaluate energy efficiency of computer servers [21]. The new metric introduced in SPECpower_ssj is overall ssj_ops/watt - operation numbers of a Java application per watt when CPU utilization of the server varies from 10% to 100%. SPECpower_ssj is focused on the energy-saving issues in CPU and memory [21]. In the case of data centers, a novel metric - Data Center Density - is used to measure the power of all equipment on a raised floor per area of the raised floor [22]. Similarly, Rivoire developed JouleSort, which is a benchmark focusing on I/O [17]. The metric used in the JouleSort benchmark is the number of sorting operations per Joule [17]. SNIA has been making an effort to promote the specification of green storage [20], which includes a green storage taxonomy classifying data storage products based on energy-dissipation characteristics and application domains in addition to a standard for collecting idle power consumption in storage systems. Power of devices in the active mode has not been included in the green storage specification until October 2009. The SUN Microsystems introduced the SWaP metric - Performance/(Space*Watts) - to measure energy efficiency of SUN servers [13]. Energy-Bench is a new benchmark suite in which throughput per Joule is used as a metric to evaluate energy efficiency [8].

B. Benchmarking, Modeling, and Testing

Benchmarking tools for file and storage systems can be classified into three categories, namely, macrobenchmarks, microbenchmarks, and trace replay tools [23]. Macrobencmarks aim to test I/O performance using a particular workload condition which represents certain real-world I/O load. TPC-C [7], TPC-H and SPC-1, SPC-2 are commonly used macrobenchmarks. Microbenchmarks are commonly used to generate synthetic I/O workloads. For example, IOmeter - an I/O microbenchmark - was developed by Intel in 1998 to generate synthetic I/O workloads for the Windows operating system [14]. The parameters in IOmeter used to control synthetic workloads mainly include: request size, read/write ratio, random/sequential access ratio and the like. It is very effective to apply IOmeter to measure the peak I/O performance of a storage device in the Windows environment. Trace replay tools record file or data block access logs into trace files which later can be replayed to represent real-world workloads. For example, Blktrace, developed by Axboe *et al.*, is a trace-replay tool running in the Linux operating systems [3]. Blktrace

TABLE I
EXISTING APPROACHES TO EVALUATING ENERGY CONSERVATION TECHNIQUES IN STORAGE SYSTEMS.

Paper	Simulation or Test	Benchmark Descriptions	Metrics for Evaluation
MAID[6]	Test	Cello96 & synthesized supercomputer center trace	Response time& energy savings
PDC[16]	Simulation	a synthetic trace & two real traces: HUM & POP	Response time& energy savings
PA[27]/PB[28]	Simulation	Self connected OLTP trace& cello96& synthetic traces	Response time& energy savings& throughput
PARAID[24]	Test	A web replay trace & Cello99-postmark	Response time& energy savings& throughput& reliability
DRPM[9]	Simulation	Pareto & Exponential& TPC-C& TPC-H	Response time& energy savings
eRAID[12]	Simulation	SPC-SE, Cello99,TPC-C& four synthetic traces	Response time& energy savings& throughput
Hibernator[26]	Simulation	A OLTP trace& Cello99	Response time& energy savings
BUD	Simulation	Synthetic traces& eight real trace	Response time& energy savings

collects and replays I/O traces at the block level for certain storage devices. In process of implementing TRACERS, we seamlessly integrated Blktrace with our load-control module.

Recently, Zedlewski designed and implemented the Dempsey tool to model the power consumption of hard disks [25]. Like Dempsey, the Drive-Thru tool developed by Peek and Flinn can be used to evaluate power management policies of file systems by replaying file traces [15]. Allalouf *et al.* built the STAMP tool, aiming to estimate the power of stand-alone disks and disk arrays [2]. Hylick *et al.* provided an analysis of hard drive energy consumption through measurements rather than simulations [10]. In year 2008, Seo *et al.* tested and evaluated the energy efficiency of flash-based solid state disks [19]. TRACER is entirely different from the aforementioned approaches in the sense that TRACER provides an integrated means of testing, benchmarking, and evaluating energy-efficient disk arrays.

C. Evaluating Energy Conservation Techniques

In the past few years, much attention has been paid to the development of energy conservation techniques for storage systems. Table I summarizes the existing solutions to evaluate performance and energy efficiency of novel energy saving schemes. The second column of the table dictates how the technique is evaluated and validated. ”Test” means measurements on real storage devices while ”Simulation” means measurements in simulation environment, like DiskSim. Table I shows that most evaluation methods rely on synthetic and real-world traces to test the proposed energy-saving schemes. The metric used to measure energy efficiency and performance are typically energy saving rate, I/O throughout, and response time. We observed that the benchmarks and traces used in the previous studies are quite diverse. In most works, simulations are a powerful research tool to validate the designed energy saving techniques for storage systems. Before validating the design of an energy-efficient storage system, one has to choose appropriate I/O traces representing certain application environments. It is essential and challenging to systematically generate useful traces for the evaluation purpose. Although real I/O traces have been widely adopted to test energy-saving schemes for storage systems, traces with extremely high or low I/O loads are unable to demonstrate the strengths of most novel energy conservation techniques. Synthetic I/O traces, on the other hand, are effective to measure peak I/O performance. However, many synthetic I/O trace tools lack the ability to dynamically control workload under given I/O intensity levels. We conclude that non-standard evaluation methodologies and metrics make comparison studies very difficult in the realm of energy-efficient storage systems. Therefore, there is a pressing need to develop a uniform and general methodology coupled with novel metrics for evaluating various energy-conservation techniques applied in storage systems in general and in disk arrays in particular. In an attempt to address this need, we developed the TRACER framework where energy efficiency can be evaluated in conjunction with performance measurements.

III. TRACER ARCHITECTURE

Figure 1 depicts the architecture of TRACER, which mainly contains the following components: an evaluation host, a workload generator, a power analyzer, and a storage system with interconnects. The evaluation host is connected with the power analyzer and the workload generator by the Ethernet. The workload generator and the storage devices communicate with each other via either the fiber channel or the Ethernet depending on the type of tested storage system. Mass storage systems like NAS and SAN are compatible with TRACER.

A. Components in TRACER

1) Evaluation Host: The evaluation host is a kernel control part of the entire system, which comprises five separate modules, including a graphic user interface (GUI), a communicator, a database, a parser, and a messenger module. GUI receives test's configurations (i.e., I/O load levels) from users who can control test processes in a real-time manner while monitoring performance and power consumption in a storage system. The users are able to send queries to the database to access results after the testing processes are done. After each test, energy efficiency and performance results are stored as records in the database for for future retrievals.

Each record in the database contains information on energy efficiency and performance (e.g., time of the test, workload modes, energy dissipation data (or power data), performance result, and energy-efficiency result). Each workload mode is a vector that consists of request size, random rate, read rate, and load proportion value. Note that load proportion values are used by the workload generator to proportionally control I/O load intensity levels. The energy dissipation data in each record contains average electrical current measured in amperes, voltage measured in volts, and power measured in watts. The performance result of each record includes average I/O operations per second (or IOPS for short), MBytes per second (or MBPS for short), and response time.

The messenger module is responsible for both passing control information to the power analyzer and receiving energy efficiency results from the power analyzer. The parser is a middle layer sitting between GUI and the messenger module. Since data protocols used in GUI and the messenger module are different, the parser has to maintain the consistency between the two protocols and avoid unnecessary conflicts. TRACER is able to support various types of power analyzer devices with some modification on the messenger module.Command information is delivered from GUI to initialize the power analyzer or finalize a power test. The communicator in the evaluation host interacts with the communicator in the workload generator through the TCP socket channel. Test control information mainly includes workload modes and I/O intensity levels configured by the users.

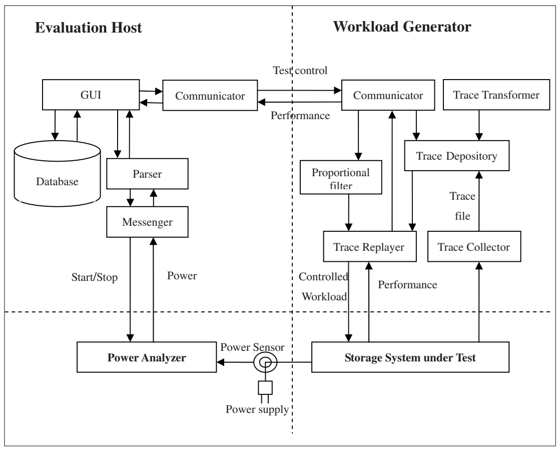


Fig. 1. The architecture of TRACER.

2) Workload Generator: The workload generator is in charge of collecting I/O traces and replaying the traces to reflect appropriate I/O workloads configured by the evaluation host. The communicator passes on request size, random/sequential rate, and read/write rate to the trace replay module, which loads a corresponding trace file from a trace repository. The proportional filter selects and replays a certain percentage of I/O requests in the loaded trace file according to a load intensity value. The filter algorithm is detailed in Section IV. We leveraged the IOmeter tool to generate peak synthetic workloads with specified request sizes, random/sequential ratios, and read/write ratios. The trace collector is a low-overhead module that performs I/O tracing for storage systems under the peak workloads. Collected trace files are stored in the trace repository. The name of each trace file implies important information such as storage device type, request size, random rate, and read rate. The trace format transformer is a useful tool developed to change the HP trace format(i.e., trace files with the extension name srt)[11] into the blktrace format (i.e., trace files with the extension name replay). Note that TRACER can only load trace files with the blktrace format and; therefore, the trace files with the HP trace format must be converted to the blktrace format before being loaded by TRACER. Apart from performing I/O tracing under synthetic workloads, TRACER stores real-world trace files into the trace repository for future retrievals. If one replays a trace file under a certain load level, he or she needs to launch the trace replay tool in TRACER that monitors and tracks performance information like I/O throughput (measured in MBPS and IOPS) and average response time. Such performance information are transmitted by the communicator to the evaluation host for further energy-efficiency analysis. The flexibility of TRACER is high, because many parameters can be configured by users. For example, the sampling cycle period - whose default value is 1 Second - is fully configurable.

3) Power Analyzer and Storage System under Test: The power analyzer keeps track of electrical current and voltage in a storage system when the trace reply tool is issuing I/O requests to the system. Energy dissipation data are transmitted to the evaluation host in a real-time manner. The power sensor used in our experiments is a magnetic loop that acquires power related data based on the Hall effect. The power analyzer has multiple channels that allow the energy efficiency of multiple storage systems to be tested simultaneously. The power analyzer has different power testing channels for both DC and AC power supplies. It is worth noting that TRACER can be used to test a wide range of storage systems including hard drives, solid state disks, disk arrays, and storage area networks (SAN).

B. Evaluating Energy Efficiency

An entire process of testing and evaluating energy efficiency of a target storage system includes following steps:

1) **Setting up the TRACER evaluation environment.** The major components to be connected in this step include the evaluation host, the power analyzer, the workload generator, the storage devices and the interconnects (see Section III-A for the description of each component).
2) **Building a trace repository.** One may use any third-party synthetic workload generator to produce peak I/O workloads with specific request size, read/write rate, and random/sequential access rate. Then, the trace collector performs I/O tracing based on a specified workload mode. Normally, it takes us approximately two minutes to collect one trace file. The trace collector is able to collect a full range of trace files automatically without users' manipulation. In addition to collecting trace files under synthetic workloads, series of real-world traces are stored in the trace repository.
3) **Testing energy efficiency.** Before a test carried out, users must configure TRACER by providing workload information. The users are allowed to view real-time energy dissipation, I/O throughput(IOPS and MBPS),

and energy-efficiency values of a tested storage system using the graphic user interface. Additionally, the users can query the database for completed tests. See Figure 2 for more details about the testing progress. In Figure 2, we can see that the function of scaling inter-arrival times between requests is also added into TRACER as a supplement for trace entries filtering scheme. In this way, I/O load intensity of a trace replay can be scaled either to 10%, 20%, 30% or 200%, 1000%, 1% of original intensity.

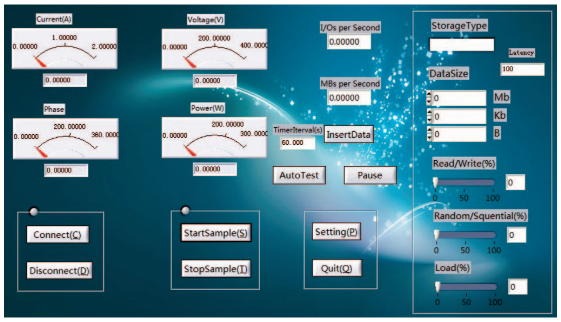


Fig. 2. The graphics user interface of TRACER environment

C. *TRACER in a Distributed System*

Figure 3 shows how to implement TRACER for mass storage systems in a distributed computing environment. In our experiment, we implemented TRACER to evaluate a storage system with a large number of disk arrays. Figure 3 demonstrates that we can make use of TRACER to test a large-scale storage system where multiple evaluation hosts, power analyzers and mass amount of storage are efficiently connected. More specifically, multiple storage devices and workload generator machines are connected with the Fiber channel to form a Fiber-Channel-based storage area network or FC-SAN. The multi-channel power analyzers in Figure 3 can monitor power dissipation in multiple storage devices in parallel. Please refer to Section V for the description on how to leverage TRACER to measure the energy efficiency of a large-scale storage system with full-scale workloads.

IV. Load Controllable Trace Replay

A. *Dynamic Load Control*

I/O traces can be replayed by a trace replay tool to represent the behaviors of I/O-intensive applications running in real-world computing environments. In this study, we paid particular attention to the implementation of a tool that replays real I/O traces in storage devices rather than disk simulators. Suppose a trace is collected by monitoring I/O requests issued to a storage system with bandwidth B, the trace can be used to test any disk device whose bandwidth is equal to or smaller than B. To evaluate performance of storage systems with bandwidth smaller than B, one can replay the trace by reducing idle periods between consecutive requests to increase I/O intensity

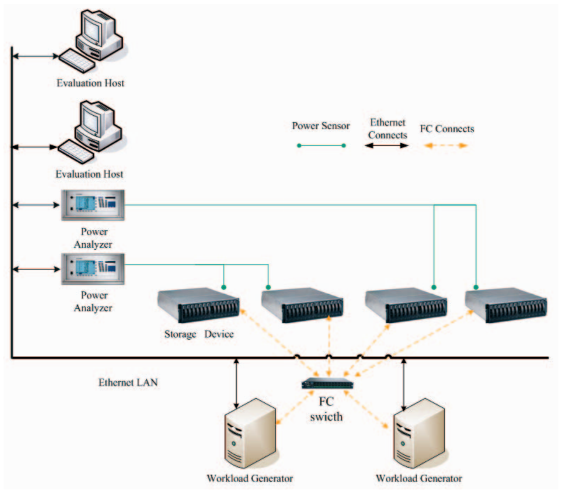


Fig. 3. The implementation of TRACER in a distributed computing environment

levels. In addition to manipulating idle periods in a trace, accurately controlling workload of the trace allows storage system developers to study the impact of I/O load intensity on the energy efficiency of storage systems. In recognition that it is critical to control load intensity levels of traces, we implemented a load control mechanism to automatically replay I/O traces.

The main goal of our trace replay tool is to uniformly select a certain proportion of trace entries from a trace file and to synchronously replay the selected trace entries. In the light of the load control mechanism, I/O load intensities of traces can be straightforwardly scaled to a specified level without significantly changing the characteristics of the original I/O traces. Fig. 4 shows the file structure of a blktrace file, where each I/O request is represented in the form of an IO_package containing the starting sector of the request, the request size (measured in Bytes), and an I/O operation type (i.e., read or write). An array of concurrent IO_packages are organized in form of a bunch, which is modeled in the file structure by the total number of IO_packages in the bunch and time stamp (i.e., arrival time of the IO_package). Note that each trace file includes thousands of bunches and IO_packages. For example, we collected a 2-minute trace file in a RAID-5 disk arrays. The trace file contains approximately 50,000 bunches and 400,000 IO_packages. Apart from our I/O traces collected under peak synthetic workloads, other real-world traces (e.g., BORG traces[4], HP cello99, cello96[11]) can be replayed by properly configuring our trace replay tool.

To implement the trace replay tool, we developed a novel filter algorithm to select and replay partial bunches from a blktrace file. The filter algorithm carries out the following four steps. First, bunches in a trace file are partitioned in groups. In other words, every 10 consecutive bunches are placed into one group. Then, one can choose the percentage of I/O requests in a trace to be replayed. For example, the filter algorithm enable the trace replay tool to perform only a portion (e.g., 10%, 20%,

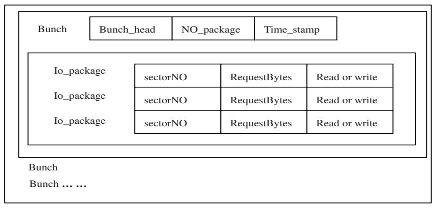


Fig. 4. Structure of a Blktrace file[3]

30%, …100% of the total bunches) of a trace by ignoring unselected trace entries. Next, the filter algorithm a portion of bunches within each bunch group. For example, if the specified portion is 20%, then two out of ten bunches in a bunch group must be uniformed selected. Finally, chosen I/O bunches by the filter algorithm are replayed based on the original time stamps, whereas unselected bunches are completely ignored by the trace replay tool.

Concurrent I/O requests in a selected bunch must be replayed in parallel. Since equal number of bunches in each bunch group are chosen by the filter algorithm, the trace replay tool is able to preserve the main accessing characteristics of the original trace file. The total bunch number in a trace file is normally very large and; therefore, in our experiments the number of replayed bunches in a trace is set to 10%, 20%, 30% … 100% of the total bunches. It is important to point out that given a bunch group, the filter algorithm uniformly rather than randomly select I/O bunches. This is mainly because random filtering bunches can possibly lead to distorted features of replayed traces due to many wave crests and troughs of workloads.

B. Uniformly Filtering Trace Entries

Figure 5 shows how bunches in each group are selected based on a specified percentage or I/O load level. Each line in Figure 5 represents a group of ten consecutive bunches in a trace. The bunches marked in yellow are bunches selected by the filter, whereas the bunches marked in grey are bunches unselected and ignored by the trace replay tool. For example, to make the load level be 10% of the peak load intensity, the trace replay tool simply selects and replays the tenth bunch of in each group. As for a load level of 20%, both the fifth and tenth bunches in each group are replayed. As a result, the filter algorithm attempts to uniformly select bunches from the original trace file. It can be easily proved that for trace files with fixed size of IO_packages (i.e., the I/O request size is fixed), this filter algorithm can manipulate I/O throughput as user demands by filtering I/O trace entries. In addition, empirical results (see Section VI) demonstrate that the filter algorithm can accurately choose I/O trace entries from a trace file where I/O requests have variable sizes.

V. Evaluation Methodology

A. Experimental Setup

Figs. 3 and 6 illustrate the testbed used to implement and evaluate TRACER. In this testbed, we connected an evaluation

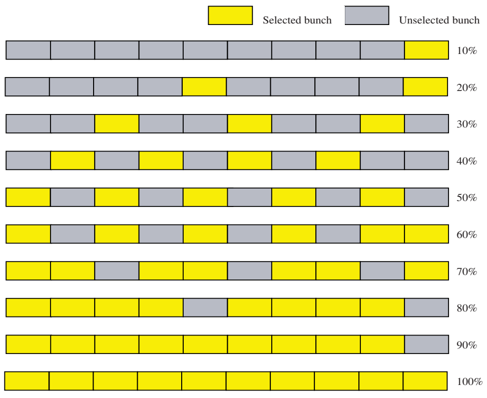


Fig. 5. Trace entries filter algorithm for load control

Fig. 6. Photo of the measurement environment

host, a workload generator, and a power analyzer through the fast Ethernet. A disk array is linked to the workload generator machine with a 4Gbps fiber channel. The disk array controller's cache is disabled during the experiments to assure direct access to disks. The input power supply of the disk array is 220V AC voltage. The details of the configuration for the testbed are listed in Table II below.

TABLE II
Configuration of the Testbed

Evaluation Host	OS: Windows 2003 Server sp1 CPU: Intel Celeron(R) 3.06G Memory: 1.5G Bytes DDR
Workload generator	OS: RHEL 4.1.1, Linux 2.6.18 CPU: Intel Dual-Core E5200 2.5GHz Memory: 2G Bytes DDR2 FC-HBA: 4Gbps Emulex LPe111
Disk array	Hard disks: 500G Seagate 7200.12 *6 SSD disks: 32G Memoright SLC *4 Cache: 300M controller cache (disabled) Power supply: 220V AC FC-HBA: 4Gbps HPFC6700

The power analyzer in the testbed (see Fig. 6) is a Kingsin KS706 multifunctional power meter, which uses a magnetic loop to enclose the 220V AC power supply of the tested disk array. The power meter measures current values by analyzing magnetic changes. The meter also measures voltage values with two probes inserted into the power socket, which is in parallel connected with the power supply of the disk array.

B. Evaluation Metrics

Recall that the goal of TRACER is to dynamically measure both energy efficiency and performance of disk arrays. To quantitatively evaluate TRACER, we have to introduce objective evaluation metrics. The power consumption is measured by the metric in terms of Watt and Kilowatt. The performance of the disk array is evaluated in term of throughput, i.e., IO/second (IOPS) and MByte/second (MBPS). Any energy conservation techniques aim to improve energy efficiency while maintaining high I/O performance and; therefore, we propose two new metrics - IOPS/Watt and MBPS/Kilowatt - to quantify both energy efficiency and performance. IOPS/Watt can be utilized to decide, within one second, how many IO requests can be processed per Watt. Similarly, MBPS/Kilowatt represents, within one second, the amount of data processed per Kilowatt. These two metrics can be used to evaluate both I/O performance and energy efficiency.

C. I/O Workloads

We conducted extensive experiments using both synthetic and real-world I/O traces. In what follows, we describe how to generate a wide variety of traces to evaluate our framework.

1) Synthetic Traces: Using IOmeter [14], we generated 125 synthetic traces to represent a large range of I/O workloads. Note that we outlined in Section III-B an approach to building a repository to store these trace files. Each trace is generated using three important parameters, namely, request size, read ratio, and random ratio. We chose to test five request sizes, five read ratios, and five random ratios, thereby generating 125 trace files. While generating each synthetic trace file, we run IOmeter for approximately two minutes. We used TRACER to replay each trace under 10 load levels to collect 1250 groups of results, each group of which includes the average power in watts (i.e., a product of current and voltage), the average throughput in IOPS and MBPS, and energy efficiency in terms of IOPS/Watt and MBPS/Kilowatt. We made use of the 125 traces to evaluate the accuracy of our load-control algorithm and the energy efficiency of the tested disk array as a function of request size, read ratio, random ratio, and load level.

TABLE III
CHARACTERISTICS OF THE WEB SERVER TRACE

File System Size (GB)	DataSet (GB)	Read ratio	Average Req_size(KB)
169.54	23.31	90.39%	21.5

2) Real-world Traces: In addition to synthetic traces, we converted the HP cello96 and cello99 traces [11] into the blktrace files before evaluating the energy efficiency of the target disk array. Note that the HP's traces are low-level disk I/O traces collected on an HP UNIX server. In our experiments, we chose a cello99 trace file, in which the read ratio is 58%. Apart from the HP traces, we replayed web server traces containing web requests for a week on the O4 machine of a web server in the Department of Computer Science, Florida International University[4]. The above traces represent real-world I/O workloads wihout need to build and run I/O-intensive applications. Table III shows the statistical characteristics of the web server trace.

VI. Experimental Results

We configured a RAID-5 storage array in which strip size is 128 KBytes. We classify our experiments into three groups: (1) synthetic workloads,(2) real-world workloads, and (3) solid-state disks. As such, we conducted the experiments in the following three steps.

1) We evaluated 125 synthetic I/O traces, each of which was replayed ten times with load proportions varied from 10% to 100%. Although we had to performed more than 1250 experiments (10 experiments for each trace), TRACER can automatically collect results after having each trace replayed.
2) After the evaluations of synthetic traces, we tested real-world traces - the web server trace and the HP cello96 trace. Like synthetic traces, each real-world trace was replayed ten times with load proportions varied from 10% to 100%. We recorded results in terms of MBPS and IOPS (see Section V-B) in each one-minute interval.
3) We built a RAID-5 disk array using solid state disks or SSD. Then, we evaluated both energy efficiency and performance of this SSD-based RAID-5 by replaying the synthetic traces on the RAID-5 disk array.

In the remainder of this section, we first study the impact of the number of disks in a disk array on energy consumption. Then, we evaluate the impacts of load proportion, random ratio, read ratio, and request size on energy efficiency of RAID-5 disk arrays including a disk array equipped with solid-state disks.

A. Number of Disks in a Disk Array

Energy dissipation in a disk array is contributed by both disks and non-disk components. Non-disk components include controller, fan, motherboard, and the like. Now we study how the number of disks in a disk array affects total power consumption of a disk array. In this set of experiments, we increased the number of disks from zero to six.

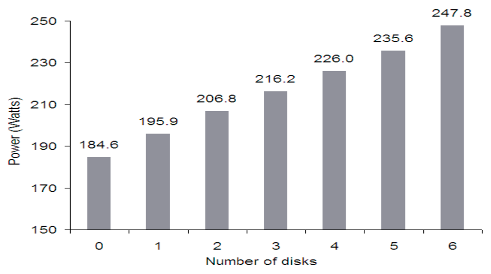


Fig. 7. Power consumptions of a RAID with increasing number of disks

Fig. 7 shows the power consumption of a disk array as the function of the numbers of disks in the idle mode (i.e., no I/O request is served). Note that power consumed by the non-disk components is equivalent to the power consumption of the disk array without any disk. We observed from Fig. 7 that the power consumption of the disks in the disk array is proportional to the number of disks. More importantly, when the number of disks exceeds three, power consumption of disks dominates the total power dissipation in the disk array.

B. Accuracy of Load Proportion Control

Recall that TRACER allows us to control I/O workload of any trace by setting load proportion level. The load level of a trace is configurable, because TRACER selects a certain percentage of trace entries from a trace file to replay. Evaluating the accuracy of load proportion control is important, since we measured impacts of various workload conditions on energy efficiency and performance by changing load levels using the load proportion control.

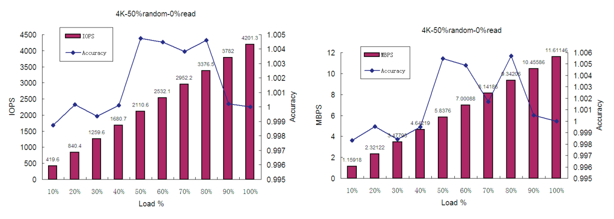
We tested 125 collected trace files, each of which was evaluated under ten configured load proportions varying from 10% to 100%. Given a trace file f and its manipulated trace f', we define load proportion used to generate f' from f as:

$$LP(f, f') = T(f')/T(f) \tag{1}$$

where $T(f)$ and $T(f')$ are the throughputs of the original trace f and the manipulated trace f'. Both $T(f)$ and $T(f')$ are measured in terms of IOPS or MBPS (see Section V-B). We measure the accuracy of the load control by comparing a measured load proportion with its configured load proportion. Thus, we define the load control accuracy with respect to trace f and its manipulated trace f' as:

$$A(f, f') = LP(f, f')/LP_{config} \tag{2}$$

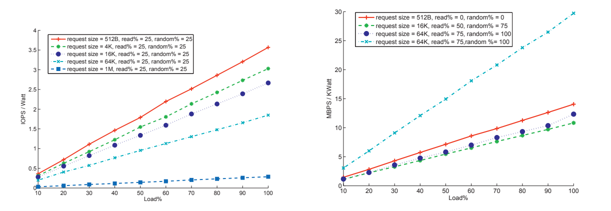
where $LP(f, f')$ is the measured load proportion and LP_{config} is the corresponding configured load proportion.



(a) Throughput: IOPS (b) Throughput: MBPS

Fig. 8. Bars show the I/O throughput (i.e., a: IOPS and b: MBPS) as functions of configured load proportion. Curves show the load control accuracy. Request Size = 4 Kbytes, Random Ratio = 50%, Read Ratio = 0%.

Fig. 8 shows the I/O throughput as functions of configured load proportion when average request size is 4 Kbytes, random ratio is 50%, and read ratio is 0%. The results plotted in Fig. 8 show that the measured load proportions are very close to the configured ones, indicating that the load control accuracy is extremely high (with error rate smaller than 0.5%). High load control accuracy is achieved because size of I/O requests in the collected traces is a constant.



(a) Request Size: 512B...1MB; Read Ratio: 25%; Random Ratio: 25%. (b) Request Size: 512B...64KB; Read Ratio: 0%...75%; Random Ratio: 25%.

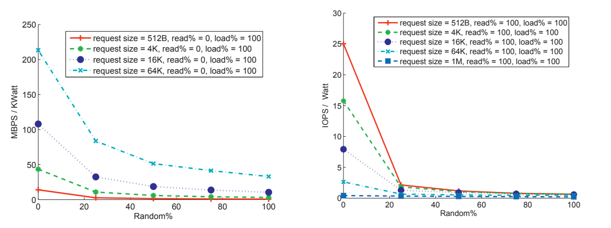
Fig. 9. Impacts of I/O load on energy efficiency measured in terms of (a) IOPS/Watt and (b)MBPS/Kilowatt.

C. Impacts of I/O Load on Energy Efficiency

Fig. 9 shows the impacts of I/O load on energy efficiency measured in terms of IOPS/Watt and MBPS/Kilowatt. We observe that energy efficiency in disk arrays is linearly proportional to I/O load, indicating that high disk utilization leads to high energy efficiency. High I/O load improves energy efficiency of disk arrays, because less idle time can be achieved by higher I/O load level. A second observation obtained from Fig. 9 is that the IOPS/Watt values for small requests are higher than those for large requests, since the disk array can process more small requests per Watt it consumes.

D. Impacts of Random Ratio on Energy Efficiency

Fig. 10 shows the MBPS/Kilowatt and IOPS/Watt values as functions of random ratio when read ratio is set to 0% and 100%, respectively. The results illustrate that when request size, read ratio and load proportion are fixed, energy efficiency in terms of MBPS/Kilowatt and IOPS/Watt reduces with the increasing value of random ratio. This general trend is observed because throughput in terms of MBPS and IOPS sharply drop with the increase of random ratio whereas energy dissipation in the disk array does not dramatically decrease as random ratio goes up.



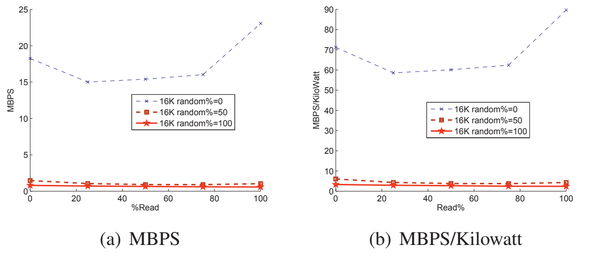
(a) Request Size: 512B...64KB; Read Ratio: 0%; Load Proportion: 100%. (b) Request Size: 512B...1MB; Read Ratio: 100%; Load Proportion: 100%.

Fig. 10. Impacts of random ratio on energy efficiency measured in terms of (a) MBPS/Kilowatt and (b) IOPS/Watt.

I/O access pattens with low random ratio help in improving the energy efficiency of disk array because disks consume more power when serving random requests than sequential requests. Compared with serving sequential requests, processing random requests can increase disk energy consumption since voice-coil actuators of hard disks consume additional energy to perform seek operations for random requests. This observation is consistent with a recent study conducted by Allalouf *et. al* [2]. Importantly, these results suggest that one can improve energy efficiency of disk arrays by aggregating random requests to form sequential access pattens to reduce energy consumption caused by unnecessary seek operations. Fig. 10 also reveals that the energy efficiency is less sensitive to random ratio when the random ratio is larger than 30%.

E. *Impacts of Read Ratio on Energy Efficiency*

Fig. 11 shows impacts of read ratio on both throughput measured in MBPS (see subfigure a) and energy efficiency measured in MBPS/Kilowatt (see subfigure b). In this set of experiments, request size is 16 Kbytes; we set the random ratio to 0%, 50%, and 100%, respectively.



(a) MBPS (b) MBPS/Kilowatt

Fig. 11. Impacts of read ratio on throughput and energy efficiency. Request Size: 16KB; Random ratio: 0%, 50%, and 100%.

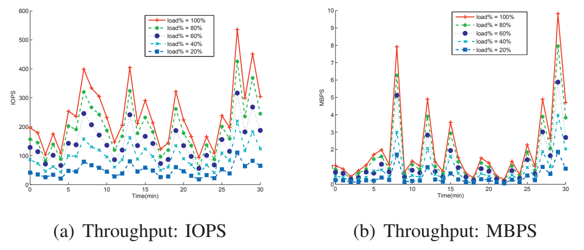
The results plotted in Fig. 11 reveal that when random ratio is 50% and 100%, both the throughput and energy efficiency of the test disk array are not very sensitive to read ratio. When it comes to low random ratio (e.g., 0%), the throughput and energy efficiency are noticeably affected by read ratio. More specifically, there is a U-shaped relationship between read ratio and throughput. We also observe a U-shaped relationship between read ratio and energy efficiency. The results suggest that applications with read-intensive or write-intensive I/O access patterns can achieve relatively higher performance and energy efficiency on disk arrays under a condition that most I/O accesses are sequential (i.e., random ratio is low).

F. *Real-World I/O Workloads*

Now we make use of TRACER to evaluate the load control accuracy and the performance of disk arrays using real-world I/O workloads - a web server trace and the HP cello99 trace (see Section V-C2 for details on these two traces). Table IV shows the accuracy of controlling I/O workloads of the real-world web server trace by illustrating both configured and measured load proportion in IOPS and MBPS. The configured load proportion is varied from 10% to 100%. Table IV confirms that TRACER can accurately control I/O workloads for real-world traces like the web server trace, because the maximum error is around 7%.

Table V compares the configured load proportion (i.e., from 10% to 100%) for the HP cello99 traces with the corresponding measured load proportion with respect to MBPS. The error rate for HP's cell99 traces is slightly higher than that for the web server trace, partially because of the uneven request sizes in the HP's cell99 traces.

Fig. 12 show throughput in IOPS (see subfigure a) and MBPS (see subfigure b) of the tested disk array when the load proportion is varied from 20% to 100%. The results illustrate that the I/O workload trend remains unchanged when the load proportion is reduced from 100% down to 10%.



(a) Throughput: IOPS (b) Throughput: MBPS

Fig. 12. A real-world web server trace. Average throughput of RAID5 under load proportions of 20%, 40%, 60%, 80%, and 100% during 30-minute replay of the web server trace

G. *Solid State Disks*

To evaluate energy impacts of solid state disks or SSD on storage systems, we built a RAID-5 disk array using four 32 Gbyes SLC SSDs with the strip size of 128KB. The idle power of an SSD is on the average of 3.5 Watts; the idle power of the SSD-based disk array is 195.8 Watts. We observed that power consumption in the active mode largely depends on the random ratio and read ratio of I/O workloads. For example, a high random ratio gives rise to low energy efficiency. This trend is consistent with the results described in Section VI-D. A low read ratio leads to relatively high energy efficiency; the trend is similar to that discussed in Section VI-E. We conclude that SSDs can improve energy efficiency in disk arrays while maintaining reasonably high I/O performance.

VII. Conclusions and Future Work

There is a lack of systematic and uniform approaches to accurately evaluating both energy efficiency and performance of storage systems. To address this technological gap, we described in this paper an integrated framework - TRACER - aiming to evaluate energy conservation techniques in mass storage systems. The TRACER framework consists of performance and energy-efficiency metrics coupled with a toolkit used to measure energy dissipation in storage systems. At the heart of TRACER, there is a load-controllable trace replay mechanism for scaling I/O load intensities (a.k.a., load proportion) of traces without significantly distorting the characteristics of original I/O traces. We implemented the TRACER framework and experimentally validated TRACER using a RAID-5 disk array. The usage of TRACER is demonstrated by performing a case study on a disk array on which both synthetic and real-world traces were replayed. The experimental results show that TRACER can be used to evaluate energy efficiency of disk arrays by accurately controlling load proportions to represent a wide range of workload conditions. In addition, we applied TRACER to investigate the impacts of

TABLE IV

ACCURACY OF LOAD PROPORTION CONTROL FOR THE WEB SERVER TRACE.

Configured Load %	10	20	30	40	50	60	70	80	90	100
Measured Load % of IOPS	9.9266	20.0126	30.2813	40.2814	50.3797	60.0663	70.6157	80.8538	88.9732	100
Accuracy of IOPS	0.99266	1.0006	1.00938	1.00704	1.00759	1.00111	1.00880	1.01067	0.98859	1
Measured Load % of MBPS	10.7035	20.2518	30.2881	39.6042	49.7253	59.847	69.343	80.3676	88.9451	100
Accuracy of MBPS	1.0704	1.0126	1.00960	0.99011	0.99451	0.9974	0.99061	1.00460	0.98828	1

TABLE V

ACCURACY OF LOAD PROPORTION CONTROL FOR THE HP CELLO99 TRACES.

Configured Load %	10	20	30	40	50	60	70	80	90	100
Measured Load % of MBPS	13.223	22.3816	27.8819	41.9044	50.3581	57.4004	67.2765	78.4304	94.3809	100

load proportion, random/sequential ratio, read/write ratio, and request size on energy-efficiency of storage systems.

We intend to bring in temperature as new metric of TRACER evaluation framework, as temperature has obvious influences on energy, performance and reliability of storage systems. We will leverage TRACER to make further measurements on mainstream energy-conservation techniques for comprehensive evaluation and comparisons. In addition, we intend to seamlessly integrate TRACER with Disksim [5] - an efficient, accurate, and highly-configurable disk system simulator.

ACKNOWLEDGMENTS

This work was supported by both the National Natural Science Foundation of China (Grant No. 60933002) and the Natural 863 Project of China (Grant No. 2009AA01A402). Xiao Qin's work was supported by the US National Science Foundation under Grants CCF-0845257 (CAREER), CNS-0757778 (CSR), CCF-0742187 (CPA), CNS-0917137 (CSR), CNS-0831502 (CyberTrust), CNS-0855251 (CRI), OCI-0753305 (CI-TEAM), DUE-0837341 (CCLI), and DUE-0830831 (SFS).

REFERENCES

[1] U. S. E. P. Agency and E. S. Program. *Energy Star program requirements for computers: Version 4.0*. http://www.energystar.gov/, Online, 2007.

[2] M. Allalouf, Y. Arbitman, M. Factor, R. Kat, K. Meth, and D. Naor. Storage modeling for power estimation. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*. ACM New York, NY, USA, 2009.

[3] J. Axboe and A. Brunelle. blktrace User Guide, 2007.

[4] M. Bhadkamkar, J. Guerra, L. Useche, S. Burnett, J. Liptak, R. Rangaswami, and V. Hristidis. BORG: block-reORGanization for self-optimizing storage systems. In *7th USENIX Conference on File and Storage Technologies*, 2009.

[5] J. Bucy and G. Ganger. The DiskSim simulation environment version 3.0 reference manual. 2003.

[6] D. Colarelli and D. Grunwald. Massive arrays of idle disks for storage archives. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–11. IEEE Computer Society Press Los Alamitos, CA, USA, 2002.

[7] T. P. P. Council. *TPC-C*. http://www.tpc.org/tpcc/, Online.

[8] T. E. M. B. C. (EEMBC). *Energy-Bench. version 1.0 power/energy benchmarks*. http://www.eembc.org/benchmark/power_sl.php, Online.

[9] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, and H. Franke. DRPM: dynamic speed control for power management in server class disks. In *Annual International Symposium on Computer Architecture*, volume 30, pages 169–181. IEEE Computer Society; 1999, 2003.

[10] A. Hylick, R. Sohan, A. Rice, and B. Jones. An Analysis of Hard Drive Energy Consumption. In *MASCOTS*, pages 103–112. IEEE Computer Society, 2008.

[11] H. Labs. *Tools and traces*. http://www.hpl.hp.com/research/ssp/software/, Online, 2007.

[12] D. Li and J. Wang. EERAID: energy efficient redundant and inexpensive disk array. In *Proceedings of the 11th workshop on ACM SIGOPS European workshop*. ACM New York, NY, USA, 2004.

[13] S. Microsystems. *SWaP (Space, Watts and Performance) Metric*. http://www.sun.com/servers/coolthreads/swap/, Online, 2007.

[14] OSDL. *Iometer project*. http://www.iometer.org/, Online, 2004.

[15] D. Peek and J. Flinn. Drive-Thru: Fast, Accurate Evaluation of Storage Power Management. In *Proceedings of the Annual USENIX Technical Conference*, 2005.

[16] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. In *Proceedings of the 18th annual International Conference on Supercomputing*, pages 68–78. ACM New York, NY, USA, 2004.

[17] S. Rivoire. *Models and Metrics for Energy-Efficient Computer Systems*. PhD thesis, Stanford University, 2008.

[18] X.-J. Ruan, A. Manzanares, S. Yin, Z.-L. Zong, and X. Qin. Performance Evaluation of Energy-Efficient Parallel I/O Systems with Write Buffer Disks. In *Proc. 38th International Conference on Parallel Processing*.

[19] E. Seo, S. Park, and B. Urgaonkar. Empirical Analysis on Energy Efficiency of Flash-based SSDs. In *Proceedings of the USENIX HotPower'08*. USENIX Association, 2008.

[20] SNIA Green TWG. SNIA Green Storage Measurement Technical Specification Working Draft Version 0.0.18. Technical report, 2009.

[21] S. P. E. C. (SPEC). *SPECpower ssj2008*. http://www.spec.org/specpower/, Online.

[22] The Green Grid. *The Green Grid opportunity: Decreasing datacenter and other IT energy usage patterns*. http://www.thegreengrid.org/, Online, 2007.

[23] A. Traeger, E. Zadok, N. Joukov, and C. Wright. A nine year study of file system and storage benchmarking. *ACM Transactions on Storage (TOS)*, 4(2):5, 2008.

[24] C. Weddle, M. Oldham, J. Qian, A.-I. A. Wang, P. Reiher, and G. Kuenning. Paraid: a gear-shifting power-aware raid. *ACM Transactions on Storage*, 3(3):1–13, 2007.

[25] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, and R. Wang. Modeling Hard-Disk Power Consumption. In *2th USENIX Conference on File and Storage Technologies*, 2003.

[26] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, and J. Wilkes. Hibernator: helping disk arrays sleep through the winter. *ACM SIGOPS Operating Systems Review*, 39(5):177–190, 2005.

[27] Q. Zhu, F. David, C. Devaraj, Z. Li, Y. Zhou, and P. Cao. Reducing energy consumption of disk storage using power-aware cache management. In *Proceedings of the 10th International Symposium on High Performance Computer Architecture*, volume 118. IEEE Computer Society Washington, DC, USA, 2004.

[28] Q. Zhu, A. Shankar, and Y. Zhou. PB-LRU: a self-tuning power aware storage cache replacement algorithm for conserving disk energy. In *Proceedings of the 18th annual international conference on Supercomputing*, pages 79–88. ACM New York, NY, USA, 2004.