Evaluation of Data Placement Method in Database Run-Time Processing Considering Energy Saving and Application Performance

Naho Iimura
Ochanomizu University
2-1-1 Otsuka, Bunkyou-ku, Tokyo, 112-8610 JAPAN
Email: naho@ogl.is.ocha.ac.jp

Norifumi Nishikawa
Hitachi, Ltd., Yokohama Research Laboratory
292, Yoshida-cho, Totsuka-ku, Yokohama, 224-0817 JAPAN
Email: norifumi.nishikawa.mn@hitachi.com

Miyuki Nakano
Shibaura Institute of Technology
3-7-5 Toyosu, Koto-ku, Tokyo, 135-8548 JAPAN
Email: miyuki@shibaura-it.ac.jp

Masato Oguchi
Ochanomizu University
2-1-1 Otsuka, Bunkyou-ku, Tokyo, 112-8610 JAPAN
Email: oguchi@computer.org

***Abstract*—In recent years, the scale of datacenters has become larger due to the explosive increase in the amount of digital data. As a result, the growth of energy consumption is an important factor in the management use cost of datacenters. Storing and processing such large volumes of data by database applications are the core technologies in this Big Data era. However, storage accounts for a significant percentage of a datacenter's energy consumption. Therefore, we try to reduce the energy of storage to save on the total cost of datacenters. The purpose of this study is to reduce the energy consumption of storage while minimizing the deterioration of application performance. Thus far, many methods for storage energy reduction have been discussed. However, because it is difficult to control storage energy reduction efficiently only at the storage level, we have investigated the storage power control mechanism on middleware (database) layer. In this paper, we use TPC-H (a database benchmark) as an application example of data processing. We evaluate a data placement control method of storage proposed for energy saving in the database run-time processing suitable for a large-scale environment.**

Keywords*—*TPC-H; storage; energy saving; data placement control

I. Introduction

In recent years, the scale of datacenters has become larger due to an explosive increase of digital data. The volume of digital data ten years from now is estimated to be approximately 44 times larger than that of the present day. Because the amount of storage is increasing, management and operation costs of storage should not be overlooked, and efficient management of data has been focused on.

The energy consumption of datacenters in the world in 2050 is estimated to be about three times larger than the total amount of power generation in Japan, where more than 120 million people are living in 2010. In this society, where energy saving is needed, it is urgent to reduce the energy consumption of datacenters in which huge volumes of data are stored [1]. Storing and processing such huge volumes of data by database applications are the core technologies in this Big Data era. Because storage accounts for 13% of datacenter energy consumption, reducing the power consumption of storage is an efficient way to save energy in datacenters.

978-1-4799-6177-1/14/$31.00

In terms of energy saving of storage, it is possible to reduce electric power consumption by shifting hard disks from an active state to a standby one when they are not accessed. However, if hard disks are accessed when they are at the standby state, they must be shifted back to the active state first, so that application performance should be degraded. Additionally, shifting hard disks between the active state and standby one consumes electric power more than that of keeping them to be the active state. Therefore, shifting hard disks to the standby state should be executed at an appropriate timing, and its decision is crucial for energy saving of storage.

In order to reduce electric power consumption of storage, there are several approaches in some layers of the system. First, it is possible to make decisions at an upper layer, the application layer. Because applications know when I/O is issued and when it is not, it is possible to decide the timing to shift hard disks to standby state by analyzing applications for achieving energy saving of storage. However, for this approach, we must analyze all applications executed on the system and it is not realistic. On the other hand, I/O can be observed at a lower layer, storage level. If hard disks are shifted to the active state when I/O is issued and shifted to the standby state when it is not at this layer, electric power consumption of storage is reduced. However, it is extremely difficult to decide the timing of I/O issues at the storage level.

Compared with these approaches, it is realistic to make a decision at middleware, the database layer. Because SQL queries are analyzed to achieve optimal execution of applications at the database layer, the timing of I/O issues can be decided when it is executed. Therefore, it is possible to reduce electric power consumption of storage by shifting hard disks to the standby state based on this observation in database run-time processing.

In our research, we save power from database processing in the cloud through efficient management of data, and the purpose of this study is to reduce the energy consumption of storage while minimizing the deterioration of application performance. Although energy saving for storage has been discussed in many literatures, it is difficult to predict the behavior and control it efficiently only at the storage level. In addition, although static analysis of the behavior of applications is also studied intensively, it is hard to predict their dynamic behavior during the execution. Therefore, we have investigated the dynamic storage power control mechanism during the execution on middleware, the database layer.

In this report, we use TPC-H as an application example of data processing, which is a widely used database benchmark that executes typical decision support processing on data [2]. First, power saving during run-time processing is investigated. Next, based on an analytical result, data placement control method is proposed in which data allocation is changed depending on the access frequency. We evaluate the control method of storage proposed for energy saving in the database run-time processing on datacenters suitable for a large-scale environment.

The remainder of this paper is organized as follows. Section II explains related works of our research. Section III introduces the experiment environment. Power consumption characteristics of HDDs are evaluated in Section IV. Section V describes the measurement of the power consumption of HDDs during TPC-H runtime processing. Section VI shows that the energy savings and performance of the storage are improved by our proposed method using data placement control. Section VII presents our concluding remarks.

II. Related work

Thus far, many methods for storage energy saving have been proposed [3],[4],[5],[6],[7]. In these studies, various methods that suspend disks according to the I/O interval for storage are proposed to realize energy savings in storage. However, it is not easy to predict the storage level I/O behaviors.

In addition, there are many studies about static methods for power saving of service by an analysis of applications before their execution. In practice, however, the power-saving method during the execution of the service has not been studied. While this cannot be solved by the platform provider, tailored power control to suit a specific application on the user side with different applications is also highly expensive. Therefore, by putting the power saving function on middleware layer (database in this study) that can monitor the control of input and output, we have tried the storage power saving control in run-time processing that does not depend on the application.

An energy saving method with efficient usage of storage by cooperative applications is proposed [8],[9]. To construct an energy efficient storage management system combined with data-intensive applications, a power saving method for storage is proposed that utilizes application level I/O behaviors. The power consumption of the storage can be reduced by using the proposed method.

We are interested in the performance of data-intensive applications on datacenters in addition to the power savings of the system. Therefore, we focus on the Service Level Agreement (SLA) that copes with both energy savings and the performance of storage. The goal of this study is to reduce energy consumption of storage while minimizing the deterioration of application performance. Thus, in this paper, we evaluate the proposed method suitable for larger environments.

In the case that power saving in storage is discussed, to replace hard disks to Solid State Drives (SSDs) is one of candidate methods. Many literatures have discussed to save energy by using SSD [11],[12],[13]. It depends on the cost if it is possible to replace hard disks to SSDs or not, and it is expected so in the future. Anyway, our approach can be applied to SSD case. Additionally, the power consumption of SSD for shifting between the active state and the standby one is much less than that of hard disks. Therefore, our proposed method should be promissing even in the era of SSD.

III. Experiment environment

We used a storage server and power meter to construct an experimental environment, which is supposed to emulate a part of datacenters. Table I shows the specifications of the storage server and power meter used for the measurements. This experimental environment can be accessed and executed remotely.

The power meter is connected to the HDDs of the server and controlled by a dedicated computer. The storage server, power meter, and computer to control the power meter can be controlled remotely for the experiments.

TABLE I. The specifications of the storage server and power meter.

OS	CentOS 5.10 64bit
CPU	AMD Athlon 64 FX-74 @ 3GHz(4 cores) x2
Memory	8 GB
HDD	Seagate Barracuda 7200 series 3.5 inch SATA 6 Gb/s 3 TB 7200 rpm 64 MB 4K sector x 11
DBMS	HITACHI HiRDB Single Server Version 9
Power Meter	YOKOGAWA WT1600 Digital Power Meter

IV. Power consumption characteristics of HDDs

In this section, we investigate the variety of transition states of HDDs, as well as the measured power consumption of each state, which is supposed to be used on datacenters. On the basis of this investigation, we calculate the Break-Even Time, which is a measurement that indicates the possibility of power-saving.

A. Variety of transition states and power consumption of HDDs

In this paper, we use four varieties of transition state: Standby 1, Standby 2, Idle, and Active. Spindown means switching state from Idle or Active to Standby 1. Spinup 1 means switching state from Standby 1 to Idle or Active. Spinup 2 means switching state from Standby 2 to Idle or Active.

[9] uses three varieties of transition state: Standby, Idle, and Active. We examined the detailed transition states of the disk used in this study. As a result, the duration of the two

different power consumptions during Standby was observed. Therefore, we distinguish them into two types of states during standby: Standby 1 and Standby 2.

We measure the power consumption of the disk in each state. In this measurement, the same disk as the previous measurement is used. Table II shows the power consumption of each state. The values of Standby 1, Standby 2, Idle, and Active states are the maximum. The values of Spindown, Spinup 1, and Spinup 2 states are the average.

TABLE II. POWER AND ENERGY CONSUMPTION OF DISK STATES.

Standby 1 (W)	Standby 2 (W)	Idle (W)	Active (W)
1.05	0.88	5.22	7.25

Spindown (J)	Spinup 1 (J)	Spinup 2 (J)
6.79	108.5	105.5

B. Break-Even Time

Break-Even Time is the amount of time to continue the Standby state that satisfies the following condition. The amount of energy needed for the spinup or spindown of the disk is equal to that of the energy saved by remaining in the Standby state during Break-Even Time. We define the parameters as follows:

E_d: the amount of energy needed for Spindown
E_{u2}: the amount of energy needed for Spinup 2
P_{s1}: the power comsumption of the HDD during the state of Standby 1
P_{s2}: the power comsumption of the HDD during the state of Standby 2
P_i: the power comsumption of the HDD during the state of Idle
T_d, T_{u2}: the amount of time required to Spindown or Spinup 2
T_{s1}, T_{s2}: the amount of time remaining for Standby 1 or Standby 2

Using these parameters, Break-Even Time T_{be} is calculated as follows:

$$T_{be} = \left(E_d + E_{u2} - P_{s2} * T_d - P_{s2} * T_{u2} + T_{s1} * (P_{s1} - P_{s2})\right) / (P_i - P_{s2})$$

We distinguish them into two types of states during Standby. Therefore, we calculated the Break-Even Time by referencing [15]. The Break-Even Time of HDDs used in this measurement was approximately 24 seconds. According to this result, to reduce power consumption by using the Standby state, an I/O interval of approximately 24 seconds or more is needed. Figure 1 shows the transition of disk power consumption used in this measurement. The state transition is: Idle to Standby 1 to Standby 2 to Idle.

V. RUNTIME POWER CONSUMPTION WHILE USING THE ENERGY SAVING STATE OF THE DISK

In this section, we measure the power consumption of HDDs during TPC-H runtime processing, which is supposed

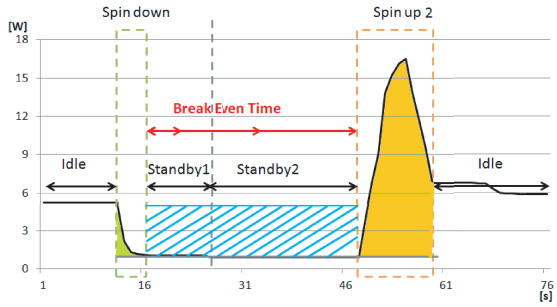

Fig. 1. Power consumption of the transition of the disk and the Break-Even Time.

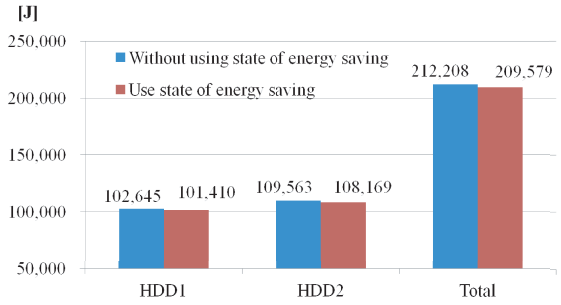

Fig. 2. Comparison of energy consumption during runtime TPC-H while using the energy saving state.

to be used on datacenters, while using the energy saving state of the disk. We compare the energy consumption with and without using the energy saving state. "Using the energy saving state of the disk" means that the state of the disk is switched to Standby. The scale factor (SF) of DB is 3, and two HDDs are used in this measurement. The placement of data on DB is the same as that of the previous experiments. We use the average value of three measurements.

Figure 2 shows the comparison of the energy consumption during TPC-H runtime processing when using the energy saving state of the disk. Without using the energy saving state of the disk, the amount of energy used was 212,208 J. When using the state, however, the amount of energy was 209,579 J, and the reduction rate was approximately 1.2%. The delay time of query processing was 125 seconds, representing a delay rate of approximately 0.8 %.

The delay of query processing has occurred due to the overhead of starting the disk. According to this result, it is possible to save on energy consumption during runtime TPC-H processing when using the I/O interval and the energy saving state of the disk.

VI. DATA PLACEMENT CONTROL

We showed that TPC-H run-time power saving is possible when using the I/O interval and the Break-Even Time. It is possible to change the state to the energy saving one after a short period of timeout, as no I/O has occurred during that period. However, this energy saving method is too naive because, in this method, we use the simple behavior of the disk without respect to applications.

In this section, we investigate the I/O frequency of data, tables, and indexes of TPC-H during runtime processing of a TPC-H query. Next, we discuss the placement of data on the disk to control the I/O interval during TPC-H runtime processing. We prepared the environment for cases where the number of used HDDs is 3, and then we evaluated our proposed method.

A. The investigation of I/O frequency

First, we investigate the I/O frequency of data, tables, and indexes of TPC-H during runtime processing of a TPC-H query to evaluate our proposed method. We used three patterns of scale factor: 1, 2, and 3. I/O interval is obtained by the `pdbufls` command [16] (DB buffer statistical information retrieval tool that comes with the DBMS) for every second. DB is placed on the raw device. The number of investigated buffers is 23. The survey period is from the beginning to the end of the query execution.

In this investigation, we focus on the actual number of times the HDD READs from the obtained data items. The purpose of this experiment is to investigate and analyze I/O frequency. In general, DBMS is used in the state in which a part of the DB resides in the buffer (called the Hot state). Therefore, the DB is in the Hot state in our experiment. The DB buffer size is approximately 0.58 GB for the table data and approximately 0.21 GB for the index data. The size of the DB varies based on the scale factor. Table III shows the size of the DB for each scale factor. According to the result of the investigation, the number of buffers containing data that have I/O was 13, whereas the number without I/O was 10.

TABLE III. SIZE OF DB (GB)

SF	1	2	3
table	1.38	2.75	4.13
index	0.29	0.57	0.86

B. The control of data placement

Based on the results of the investigation in Section VI-A, we modify the data placement. In all cases, the scale factor is 1 to 3, and the data are classified into two types: the data that has the actual number of instances of READ and that that does not have it. HDD1, 2 and 3 are the same HDDs we have used in the previous experiments.

First, we placed data such that the amount of data in each HDD to be evenly. The frequency of the data I/O is not considered in this case. We call this condition "without the control of data placement." In this case, the ratio of the amount of arranged data is HDD1:HDD2:HDD3 = 1:1:2. The amount of data on HDD3 is 2 times that of HDD1 and HDD2. This is because the amount of data stored on a certain buffer is almost half of the total amount of TPC-H data, and these data are placed on HDD3.

For the first evaluation of our proposed method, we classified the data into three types: (1) the data that has no actual number of instances of READ, (2) half the amount of data that has the actual number of instances of READ, and (3) All of the rest of the data. (1) is placed on HDD1, (2) is placed on HDD2, and (3) is placed on HDD3. We call this condition "with control of data placement 1."

Fig. 3. The I/O frequency of specified data run-time TPC-H.

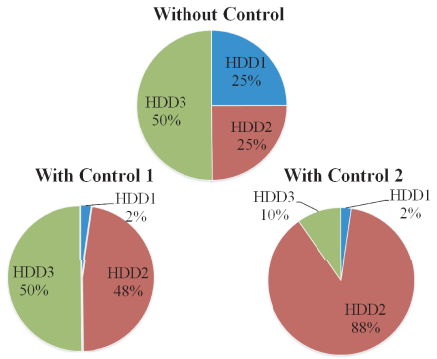

Fig. 4. The ratio of data allocated in each HDDs(3-Disks).

For the second evaluation of our proposed method, we classified the data into three types: (1) the data that has no actual number of instances of READ, (2) the data which has the actual number of instances of READ and has the specific I/O interval (9800 seconds or more before the end of executing Q8), (3) ALL of the rest of the data. We call this condition "with control of data placement 2." Figure 3 shows the I/O frequency of run-time TPC-H processing of (2), and the scale factor is 3. Only 4 types of data are used in 3 periods primarily, and they are not used during the other period. In this condition, the data that has the actual number of instances of READ and has the specific I/O interval (9800 seconds or more before the end of executing Q8) is classified. With using control of data placement 2, we can expect more energy saving. Because the HDD in which (2) is placed has longer I/O interval, and we can use HDD's energy saving state.

Figure 4 shows the ratio of data allocated in each HDDs. With control of data placement 1, HDD1 and HDD2 have equal amaount of data that have actual number of instances of READ. With control of data placement 2, the amount of data is biased to HDD2.

As the same with the previous experiments, we measured energy consumption of each HDD and the query processing time. We compared each item of the measurement value with and without the control of data placement. The scale factor is 1 to 3, as the same with that of experiments we have measured thus far. We applied the energy saving state of the disk to HDD1.

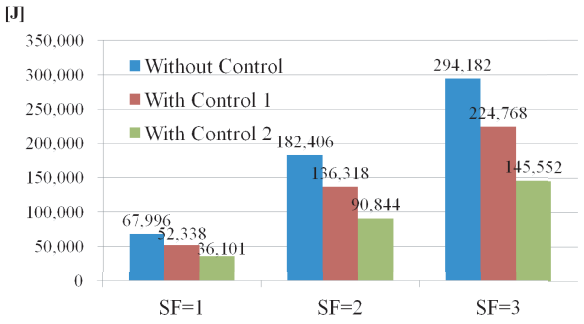

Fig. 5. Power consumption with changing data placement (3-Disks).

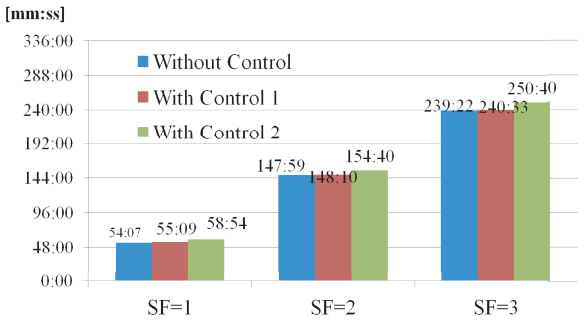

Fig. 6. The amount of runtime with changing data placement (3-Disks).

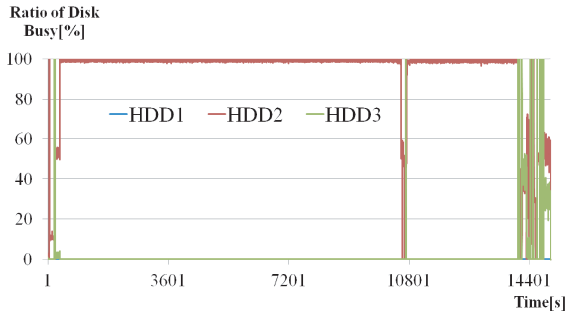

Fig. 7. The disk-busy rate with changing data placement.

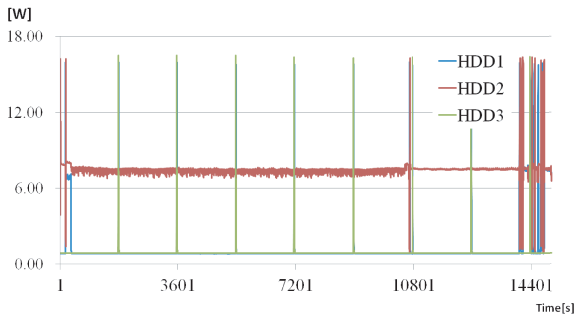

Fig. 8. The power consumption with changing data placement.

Figure 5 shows the comparison of energy consumption with and without the control of data placement. Figure 6 shows the comparison of query processing time. The reduction rate of power consumption in the condition of with control of data placement 1 is approximately 22-25%, whereas the delay rate is approximately 0.5-1.5 %. On the other hand, the reduction rate of power consumption in the condition of with control of data placement 2 is approximately 47-50%, whereas the delay rate is approximately 4-8 %. Compared the control of data placement 1 with the control of data placement 2, the latter can save more energy and the delay rate is larger. This is because using control of data placement 2, HDD3 has longer I/O interval and HDD2 has bigger load.

We acquired and analyzed the I/O traces of disks by `blktrace` and `btrecord` (tools for I/O trace analysis) [14] after changing the data placement. According to the analysis, HDD1 (stored no I/O data) does not seem to perform I/O during query processing, and data I/O is concentrated on HDD2 and HDD3 (stored I/O data).

Furthermore, we investigate the disk busy rate by the command `iostat`. Table IV shows the result of this investigation. Figure 7 shows the changes in each disk busy rate when the scale factor is 3. According to Table IV, it seems that, after changing the data placement, the disk busy rate of HDD1 (no I/O data placed) is 0 %, whereas that of HDD2 and HDD3 (I/O data placed) is constant or increased. However, in regard to energy consumption after changing the data placement, HDD1 remained constant, and HDD2 and HDD3 are always Idle or Active.

Figure 8 shows that power consumptions of HDDs per second when the scale factor is 3. According to Figure 8, it seems that, HDD1's state is always Standby, HDD2 is Idle or Active, and HDD3 is almost Standby. The power consumption of HDD1 and HDD2 is increased per 30 minutes, this means spin up occured by check of disk's state (smartd.conf[17]). According to Figure 7, it is obvious that disk I/O is not done on the time of this spin up. According to the above result, the data control method in this case is also effective for energy savings of storage during the execution of run-time applications.

Thus, it is possible to reduce power consumption effectively using our proposed method when the number of disk is three, as well. Our proposed method is also effective in a large-scale environment that includes a large number of disks because this approach is scalable in terms of the number of disks.

TABLE IV. RATIO OF DISK BUSY(%)

		HDD1	HDD2	HDD3
Without control	Average	64.3	1.43	29.4
	Maximum	80.8	55.1	50.8
With control 1	Average	0	65.8	29.3
	Maximum	0	95.8	51.1
With control 2	Average	0	97.2	0.17
	Maximum	0	100	100

VII. CONCLUSION

We consider energy savings of datacenters as possible by reducing the energy consumption of storage through the efficient management of data. In this paper, the evaluation of a data placement control method we proposed suitable for a large-scale system environment is shown.

Based on existing research, we analyzed the performance and energy consumption during runtime disk access. Next, in consideration of two patterns of standby (the energy saving

state of the disk), we calculated Break-Even Time precisely. Furthermore, as an evaluation of our proposed method, we use TPC-H (a database benchmark) as a data-intensive application and evaluate the control method of storage we proposed for energy savings during the runtime database benchmark. We succeeded in the saving energy about maximum 50% by data placement control with suppressing the rate of delay about 8%. The data placement control method is shown to be effective for energy savings during runtime application processing. Our proposed method is also effective in a large-scale environment that includes a large number of disks because this approach is scalable in terms of the number of disks.

Future works include an examination of more detailed data placement on three or more disks for energy savings. Dividing data appropriately is considered to be useful to obtain a longer I/O interval. We will perform these examinations.

ACKNOWLEDGMENT

This work is partly supported by the Ministry of Education, Culture, Sports, Science and Technology, under Grant 24300034 and 25280022 of Grant-in-Aid for Scientific Research.

REFERENCES

[1] GIPC Survey and Estimation Committee Report FY2009 (Summary), http://www.greenit-pc.jp/activity/reporting /100707/index.html, 2009

[2] TPC-H: http://www.tpc.org/tpch/default.asp

[3] Jorge Guerra, Himabindu Pucha, Joseph Glider, Wendy Belluomini, and Raju Rangaswami: Cost Effective Storage using Extent Based Dynamic Tiering, In Proc. 9th USENIX Conference on File and Storage Technologies, pp.1-14, 2011.

[4] Dushyanth Narayanan, Austin Donnelly, and Antony Rowstron: Write Off-Loading: Practical Power Management for Enterprise Storage, In Proc. 6th USENIX Conference on File and Storage Technologies, pp.253-267, 2008.

[5] Athanasios E Papathanasiou and Michael L Scott: Energy Efficient Prefetching and Caching, In Proc. the annual conference on USENIX Annual Technical Conference, 2004.

[6] Akshat Verma, Ricardo Koller, Luis Useche, and Raju Rangaswami: SRCMap : Energy Proportional Storage using Dynamic Consolidation, In Proc. 8th USENIX Conference on File and Storage Technologies, 2010.

[7] Charles Weddle, Mathew Oldham, Jin Qian, An-I Andy Wang, Peter Reiher, and Geo Kuenning: PARAID: A Gear-Shifting Power-Aware RAID, In Proc. 5th USENIX Conference on File and Storage Technologies, Vol. 3, pp. 245-260, October 2007.

[8] Norifumi Nishikawa, Miyuki Nakano, and Masaru Kitsuregawa: Runtime Disk Energy Saving Method Using Application I/O Behavior and Its Evaluation : Energy Saving Efficiency for Online Transaction Processing, The IEICE transactions on information and systems, Vol.J95-D, No.3, pp.447-459, March 2012.

[9] Norifumi Nishikawa, Miyuki Nakano, and Masaru Kitsuregawa: Energy Efficient Storage Management Cooperated with Large Data Intensive Applications, In Proc. 28th IEEE International Conference on Data Engineering (IEEE ICDE 2012), pp.126-137, April 2012.

[10] Naho Iimura, Norifumi Nishikawa, Miyuki Nakano, and Masato Oguchi: A Proposal of Storage Control Method for Energy Saving on Runtime Database Processing, In Proc. Multimedia, Distributed, Cooperative, and Mobile Symposium 2013, pp.1646-1652, 7C-1, July 2013.

[11] Jian Ouyang, Shiding Lin, Zhenyu Hou, Peng Wang, Yong Wang, and Guangyu Sun. Active SSD design for energy-efficiency improvement of web-scale data analysis, IEEE International Symposium on Low Power Electronics and Design (ISLPED 2013), pp.286-291, September 2013.

[12] Peng Li, Gomez, K., Lilja, D.J. Exploiting free silicon for energy-efficient computing directly in NAND flash-based solid-state storage systems, IEEE High Performance Extreme Computing Conference (HPEC 2013), pp.1-6, September 2013.

[13] Devesh Tiwari, Sudharshan S. Vazhkudai, Youngjae Kim, Xiaosong Ma, Simona Boboila, and Peter J. Desnoyers. Reducing Data Movement Costs Using Energy-Efficient, Active Computation on SSD, USENIX Workshop on Power-Aware Computing and Systems (HotPower '12), October, 2012.

[14] Alan D. Brunelle: btrecord and btreplay User Guide, http://www.cse.unsw.edu.au/ aaronc/iosched/doc/btreplay.html, 2007.

[15] Y.H. Lu, G.D.Micheli: Comparing System-Level Power Management Policies, IEEE Design & Test of Computers, Vol.18, No.2, pp.10-19, March 2001.

[16] pdbufls: http://www.hitachi.co.jp/Prod/comp/soft1/ manual/pc/d635540/W3550027.HTM

[17] smartd.conf: http://smartmontools.sourceforge.net/ man/smartd.conf.5.html