Reliable Energy-Aware SSD based RAID-6 System

Mehdi Pirahandeh, Deok-Hwan Kim*
Department of Electronic Engineering, Inha University, Incheon, South Korea
mehdi@iesl.inha.ac.kr, deokhwan@inha.ac.kr

***Abstract*—*This paper presents an improved approach for periodic estimating the reliability and energy consumption of SSDs. The reliability estimation has been considered to enhance the energy efficiency on the SSD based RAID-6 system. We also present additional layered architecture for reliable energy-aware RAID-6 system. The proposed method mainly differs from existing techniques in that data pages are segmented into packages and the proposed method uses the power switching of SSDs after writing or reading of current package is done. The experimental results show that the proposed method significantly reduces the energy consumption by controlling the power modes of SSDs.*

I. INTRODUCTION

Current trends in storage systems enforce SSD manufactures and researchers to promise breakthroughs in terms of energy consumption, reliability, and performance of SSDs [1]. Many methods have been proposed to achieve optimal energy consumption for HDD, SSD and large storage system [2-5]. These methods emphasize in three phases such as dynamic voltage measurement, auto power management and compiler directed energy optimization. It is important to understand and perform storage energy saving to use various RAID systems. In addition, reliability system costs for replication based schemes are expensive [3, 4]. However, these are challenges which SSD markets deal with. We propose the energy aware algorithm and model which enable SSDs to increase the performance and decrease the level of power consumption.

To identify performance and energy use of RAID systems with various type SSD storage devices, we used the analysis methods for reliability and energy flow measurement. Continuously, data should be archived along with "some redundant data" across many disks that if disk failures happen, then you still have enough data to repair disks [2]. For example, we calculate the repair transition rate at time *t*, which needs absolute time and relative time. The average data loss should be considered during *N* iterations of IO operations on the critically exposed sectors of RAID system [3]. The utilization of a given I/O phase equals to the ratio of the accumulative disk requirement and the bandwidth of the SSD. Lower utilization levels produce lower failure rates and the probability that a disk fails within a certain range between 20% and 60% is considered as the safe utilization zone [4].

II. PROPOSED RAID SYSTEM

In this study, we are focusing issue of power consumption for SSD based RAID system. Fig.1 shows the overall structure of proposed RAID system which consists of three main layers. In the core layer, we added three functions into the traditional RAID-6 controller. First function is an encoding and decoding scheduler. Second ones are the random and sequential read and write modules with an internal failure detector using erasure codes. In the third function, traces are updated using the log generator. Reliability-aware layer includes procedures for prediction of SSD reliability by using erasure codes generated from the core layer.

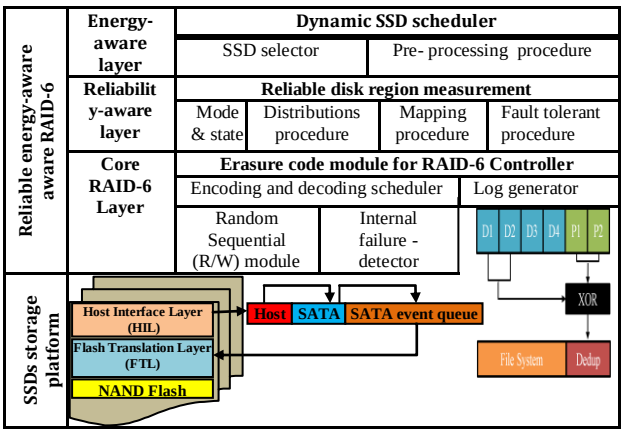

Figure 1: Overall structure of reliable energy-aware SSD based RAID System

Energy-aware layer consists of pre-processing procedure, SSD selector and dynamic SSD scheduler. The pre-processing procedure initializes the reliability measurement and utilization level. The SSD selector will set the status of selected SSD power mode to idle-sleep-active via imported traces. Finally, the dynamic SSD scheduler will visualize the statistics of SSD energy consumption. During each cycle, dynamic SSD scheduler will update the power mode into idle, sleep or active. In the remaining part, multi SSD storage platform consists of host interface layer, flash translation layer and NAND flash chips. For multi storage devices using total *M* SSDs, it can be represented as a matrix including six columns and *i* rows where *i**6=*M*. The six columns denote parallel IO sequence while *i* rows denote serial IO sequence. Fig.2 shows the energy flow of sequential/random read-write operation using proposed method. To reduce the energy consumption, we propose to use power switching of SSDs after writing of current package is done. Before read- write, data pages are segmented into packages. When read-write of current package is done in SSD S_j, IO operation of next package is performed in SSD S_{j+1}. RAID system measures SSD reliability and choose the parity SSD S_{j+4} and S_{j+5} with less utilization level. Normal read operation needs to access four disks from S_j to S_{j+3} sequentially and skips two parity SSDs as shown in Fig. 2(a) whereas write

* Corresponding Author

operation requires accessing six SSDs using power switching modes as shown in Fig. 2(b).

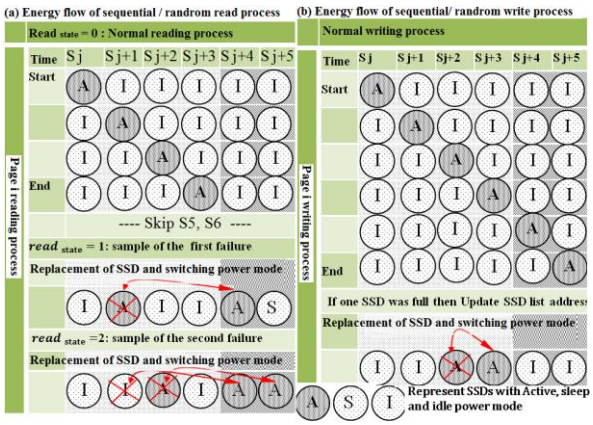

Figure 2: energy flow of sequential/random read-write operations

The rebuild process illustrated in Fig2.(a) has two states. When $\text{read}_{\text{state}} = 1$, 1st parity SSD is "active" and 2nd parity SSD is in "sleep" mode while $\text{read}_{\text{state}} = 2$, then both the SSDs are in "active" mode. Therefore, when one read operation fails, there is a strong likelihood that failure occurs in other SSD of the system which is called rare event [5]. Energy consumption of parallel SSDs can be reduced by dynamically switching three power states – active, idle and sleep. In addition, the method allows one of parity SSDs in safe zone to be sleep mode for saving the energy. The properties of proposed RAID system shown in Table 1 are as follows:

- We can estimate power consumption manually for each SSD using multiple power modes such as waken, active, sleep, idle and off. The estimation period is $\Delta t = t_2 - t_1$ which can be extended up to 30 minute.
- Average delay value of switching power mode Δt_{delay} can be calculated after *N* page write (line 27) where $t_{\text{write current package}}$ is used as writing time for one package.

$$(\sum_{i=1}^{6N} t_{\text{write current package}} - t_{\text{write previous package}}) / 6N$$

- Average data loss is calculated based on the Markov model and utilization level [3-4]. Rebuild time (t_{rebuild}) will be measured in line 14-21.
- Total energy estimation E_{Total} can be measured for *P* read and *N* write page through total idle energy $E_{\text{total idle}}$ and total read and write energy $E_{total\ Read\ \text{and write}}$ (line 30):

$$E_{total\ Read\ \text{and write}} = \sum_{i=1}^{P} t_{\text{write}} * E_{\text{active}} + \sum_{i=1}^{N} t_{\text{read}} * E_{\text{active}}$$

$$E_{total\ idle} = t_{\text{Clock}} - (\sum_{i=1}^{P} t_{\text{write}} + \sum_{i=1}^{N} t_{\text{read}})) * E_{\text{idle}}$$

$$E_{\text{Total}} = E_{\text{idle}} + E_{\text{Read and write}}$$

Where the t_{write}, t_{read}, t_{Clock} are writing, reading, total time and E_{active}, E_{idle} denote current SSD energy for active and idle mode. Reliable energy-fault aware algorithm is illustrated in Table 1.

Table 1: reliable energy-fault aware algorithm

1. Generate SSDs matrix
2. Initialize the estimated power modes and Set current mode as " idle"
3. Initialize utilization level and average data loss
4. Initialize Δt_{delay}
5. Define Page with Package Matrix: P [$p_0, p_1, p_2, p_3, p_4, p_5$]
6. Define Page address : $P_a[a_0, a_1, a_2, a_3, a_4, a_5]$
7. if (**current event** == **read**)
8. Initialize P_a[] list
9. While (**Not** end of P_a [] list) then
10. If normal state then Set $read_{\text{state}} = 0$
11. Switch current SSD power mode from " idle" to " active"
12 **read** data in $P_a[0] \ldots P_a[3]$ and store them into current Page *P*[]
13. Switch current SSD power mode from " active" to " idle"
14. If first failure happen then $read_{\text{state}} = 1$, Increase $t_{rebuild}$
15. Set power mode of 1st parity as active and 2nd parity as sleep
16. Set $read_{\text{state}} = 0$
17. Replace the failed SSD with 1st parity, Stop $t_{rebuild}$
18. If Second failure happen then Set $read_{\text{state}} = 2$,Increase $t_{rebuild}$
19. Set both 1st and 2nd parity SSDs power mode as active
20. Set $read_{\text{state}} = 0$
21. Replace the failed SSD with 2nd parity, Stop $t_{rebuild}$
22. if (**current event** == **write**)
24. Create P_a[]
25. While (**Not** end of P_a[] list) then
26. Switch the SSD power mode from " idle" to " active"
27 **Write** data in P_a [0]…P_a [5] into corresponding SSDs
28. Switch the SSD power mode from " active" to " idle"
29. if SSD corresponding to P_a[] is full, then update P_a[]
30. Estimate total power consumption(E_{Total})

III. EXPRIMENTAL RESULTS

The estimation periods are 2 sec, 2, 30 minute respectively. Fig. 3 shows the energy flow using power modes for IO operation. The result shows that the method significantly reduces the energy consumption by controlling the power switch of SSDs.

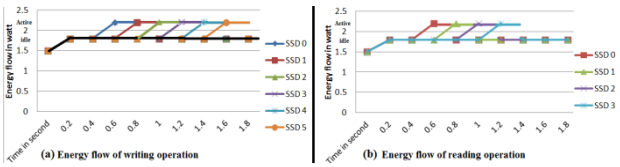

Figure 3: Energy flow in terms of read and write operation

ACKNOWLEDGMENT

This work was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0004114) and in part by Ministry of Knowledge Economy (MKE) and Korea Institute for Advancement in Technology (KIAT) through the workforce Development Program in Strategic Technology.

REFERENCES

[1]. D. Schall and et al. *Enhancing Energy Efficiency of Database Applications Using SSDs*. USENIX, 2007.

[2]. K. M. Greenan and et al. *A Spin-Up Saved Is Energy Earned: Achieving Power-Efficient, Erasure-Coded Storage*. USENIX, 2008.

[3]. N. Nishikawaand et al. *Energy aware RAID Configuration for Large Storage Systems*. IEEE, 2011.

[4]. E. Seo and et al. *Empirical analysis on energy efficiency of flash-based SSDs*. USENIX, 2008.

[5]. H J. Lee and et al. *Augmenting RAID with an SSD for energy relief*. USENIX HotPower, 2008.