Fast Performance Estimation of Solid State Disk Reflecting Real Environments[1]

Do Yeun Kim[2]
Division of Computer and
Communication Engineering,
Korea University,
Korea
doyonee@korea.ac.kr

Jong Sung Lee
Division of Computer and
Communication Engineering,
Korea University,
Korea
sayhappy@korea.ac.kr

Chanik Park
SAMSUNG
Electronics, Co., Ltd,
Korea
ci.park@samsung.com

Eui-Young Chung
School of Electrical and
Electronic Engineering,
Yonsei University,
Korea
eychung@yonsei.ac.kr

Sung Woo Chung
Division of Computer and
Communication Engineering,
Korea University,
Korea
swchung@korea.ac.kr

Recently, Solid State Disk (SSD)s, based on NAND flash memories [1], dramatically improve storage performance. The performance of the SSD exceeds that of a similar-sized HDD by more than 150% and its weight is approximately 50% compared to a conventional HDD. Moreover, the SSD uses less power. Naturally, it comes into the spotlight as a next generation storage device [2]. However, though storage performance is enhanced, overall system performance may not increase proportionally without appropriate software support. For this reason, we need to explore optimal structures of systems to fully utilize the storage performance. Hence, it is necessary to emulate more advanced storage device that does not exist currently in the context of real OS or applications. Previous works such as [3][4] proposed storage emulation techniques [3][4]. However, they do not reflect real system environments or cause long evaluation time. In this paper, we propose a novel emulation approach, called 'Fast Device Emulation (FDE)', on the SSD. The main purpose of the FDE is to estimate the system performance with an enhanced SSD that currently does not exist. In addition, real environments are reflected minimizing the overhead of storage emulation.

We can determine the storage performance is improved when DMA read time for the storage becomes shorter. To emulate shorter DMA read time, we suspend a CPU for a certain time during each DMA read request to the storage. After execution, the total suspended time is subtracted from the total execution time, as illustrated in Figure 1.
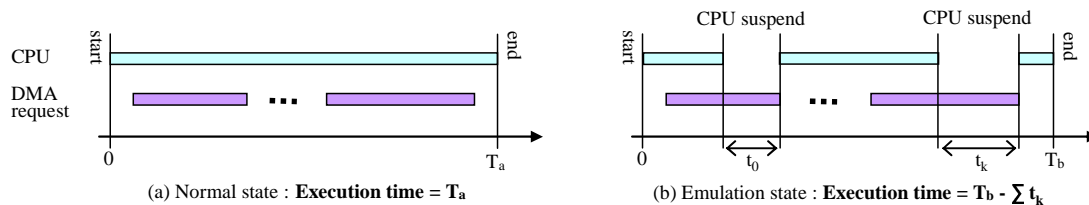

(a) Normal state : **Execution time = T_a**   (b) Emulation state : **Execution time = $T_b - \sum t_k$**

Figure 1: Emulation Methodology

In the case of a SSD, there is no spin-up time, seek time and rotational latency that a HDD suffers. Hence, DMA read time is directly proportional to the number of sectors requested: the variations among DMA times for the same number of sectors are negligible (2%, on average). From our experiment, we obtained the DMA time for the number of sectors on the SSD [1] as shown in Table 1.

The number of sectors (512B sector)	32	56	128	200
DMA read time	530μs	720μs	1320μs	1900μs

Table 1: DMA time for the number of sectors

For example, when the number of sectors for the DMA read request is 128, we can estimate the DMA time of the read request as 1320μs by Table 1. In this case, when we emulate faster SSD whose performance is improved by 30%, the CPU should be suspended for 396μs: this is 30% of 1320μs. We do the same procedure for each DMA read operation, depending on the number of sectors. After execution, we subtract the total suspended time (30% of all the read DMA operations on this example) from the total execution time.

At the first glance, the proposed FDE seems to be same as simple calculation by hand. However, there is a big difference. In most cases, there exist some data dependencies between CPU operations and DMA operations (e.g., when CPU is waiting for the result of the DMA read operation), which can not be considered in simple calculation by hand. Using the proposed technique, data dependencies are dynamically considered. Moreover, the FDE are applicable to special DMA operations that are related with OS operations such as boot-up or hibernation.

When we compare the result estimated from the FDE using the SSD (throughput 46.7MB/sec, CPU suspend 50.4%) to that of the real SSD (throughput 92.6MB/sec), the FDE predicts execution time with 94.5% accuracy, on average. The FDE has been studied on DMA read operations and further works are now in progress on DMA write operations. The FDE will be used to analyze the effects of the next generation SSD of Samsung Electronics on system performance. Moreover, the FDE enables exploration of operating systems optimized for the next generation SSD that does not exist currently.

References
[1] *SAMSUNG NAND Flash-based Solid State Drive*. Data sheet. Samsung Electronics, July 2007
[2] J. Janukowicz, and D. Reinsel. *Evaluating the SSD Total Cost of Ownership*. White paper. International Data Corporation(IDC), Oct. 2007.
[3] G. R. Ganger, B. L. Worthington, and Y. N. Patt. *The DiskSim simulation environment version 1.0 reference manual*, Technical report CSE–TR–358–98. Department of Computer Science and Engineering, University of Michigan, February 1998.
[4] J. L. Griffin, J. Schindler, S. W. Schlosser, J. S. Bucy, and G. R. Ganger. *Timing-accurate storage emulation*. Proceedings of the Conference on File and Storage Technologies (FAST). USENIX, January 2002

---

[2] M.S. course student since March 2007.