Tailor-made SSD Using a Genetic Algorithm

Hanchan Jo*, Hyunchan Park, and Chuck Yoo, *Member, IEEE*
*Samsung Electronics Co., Ltd.
Korea University, Seoul, Korea

***Abstract*--We present a design system for SSDs that utilizes a genetic algorithm. To maximize the performance in a specific system with a workload trace, the technique efficiently determines customized architectural parameters for the SSD. Compared to general SSDs, our proposed scheme reduces the average response time by up to 30%.**

I. INTRODUCTION

NAND flash memory has become the most popular storage for consumer electronics because of its many attractive characteristics: high access speed, low power consumption, light weight, and small size. Solid state drives (SSDs) are the representative NAND flash storage, and are widely used in mobile phones, personal computers, and enterprise server systems. In particular, enterprise servers that demand high I/O performance tend to adopt SSD as a secondary storage or cache [5]. Such servers require customized SSDs that are optimized to the peculiar workload of each server. For example, a database server requests many I/O operations at randomized positions over a short period, whereas a video streaming server requests large and sequential I/O operations periodically. To satisfy these various I/O performance requirements, SSD manufacturers adjust the parameters of the embedded I/O controller [3], [2].

The embedded I/O controller in an SSD manages the behavior of the SSD internals. The controller exploits flash chips in parallel to achieve high SSD performance. In addition, it manages the wear-leveling of flash chips to extend the SSD lifetime. There are several adjustable parameters in the controller that determine the performance characteristics of SSDs. Many studies have attempted to clarify the relationship between these parameters and the SSD performance. Agrawal et al. presented the design trade-offs that are influenced by the hardware parameters, such as the number of channels, planes, and dies on multiple chips [3]. Chen researched the effect of the read-ahead operation, and concluded that the technique is effective for sequential, but not randomized, I/O operations [1].

The problem is that no total solution exists for an SSD design that provides the best set of parameters for a specific workload. Previous studies have not determined the effectiveness of, or relationships between, all relevant parameters. This is actually very difficult, because there are so many parameters in an I/O controller. Another approach is to test every set of parameters for a specific workload. Although

This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2010-0029180) with KREONET and in part supported by Seoul R&BD Program(WR080951)

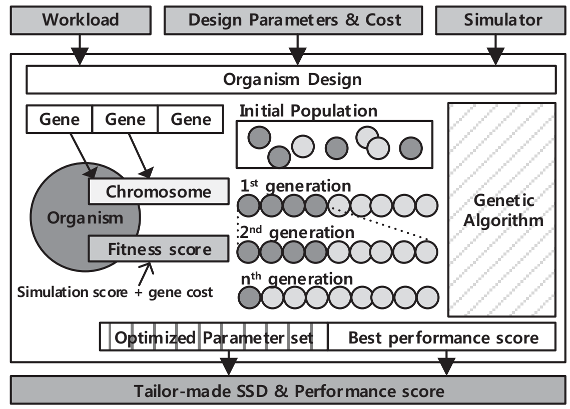

Fig. 1. Architecture of SSD-Tailor.

this will certainly find the best set, it consumes a lot of time and cost, and is therefore not considered a viable solution to the problem.

In this paper, we propose SSD-Tailor, a customization system for SSDs. SSD-Tailor determines the best set of parameters for a specific workload using a genetic algorithm (GA). GAs are search heuristics that emulate the process of natural evolution [4]. We apply a GA to optimize the SSD design by creating a chromosome containing the SSD parameters, such as the mapping and cleaning policies. The GA reduces the number of calculations needed to obtain the optimal parameter set, thus decreasing the development cost of an SSD with customized performance for an enterprise server.

We implement SSD-Tailor using DiskSim, a simulator for storage systems, and evaluate SSD-Tailor with real-world workloads collected by an enterprise server. As a result, our GA-based SSD-Tailor determines the best parameter set 288 times fewer simulations than a brute-force algorithm. In addition, the parameter set obtained by SSD-Tailor reduces the average response time by up to 30% compared to an SSD with a general parameter set.

II. SSD-TAILOR

SSD-Tailor is a GA-based system for designing customized SSDs for enterprise servers. The system input is the workload of the specific enterprise server, and the output is the tailor-made SSD that exhibits optimal performance. We apply the GA to determine the best set of SSD parameters for the input server workload. We expect the GA to reduce the number of tests needed to obtain the best solution, decreasing the manufacturing cost of customized SSDs.

SSD-Tailor receives two inputs: the workload to be optimized, and the parameter set, including information about

TABLE I
SSD DESIGN PARAMETERS

Parameter	Range	Cost
Reserved Pages (%)	1~20	Added
Minimum Free Blocks (%)	1~10	Added
Clean Policy	Greedy, Wear-leveling	
Plane Block Mapping	Simple concatenation, Full stripping	
Copy-back	Enable, Disable	
Number of Parallel units	Entire element, Two dies, Four plane-pair	
Max Queue depth	1, 16, 64	Added

their range and cost, as shown in Fig. 1. SSD-Tailor makes an organism that represents an SSD. This organism has a chromosome and a fitness score. The chromosome consists of genes, each of which represents a parameter with a cost and range. SSD-Tailor simulates the organism according to the genes in the chromosome. The fitness score is calculated by combining the performance score with the cost of the genes being used.

Fig. 1 illustrates the procedure of the GA. SSD-Tailor forms an initial population of organisms from randomly generated chromosomes, simulating each organism to calculate its fitness score. SSD-Tailor then selects a group of organisms with high fitness scores to be parents to the first generation. The chromosomes of the dominant organisms are passed on to the next generation. The characteristics of the parent organisms are passed to their children with a mixture of crossover and mutation, which occur with a fixed probability. Under crossover, parts of the parents' chromosomes are randomly selected and recombined to generate a new chromosome. Mutation generates a brand-new chromosome at random. This operation increases the likelihood of finding the global optimum. The generation process is repeated until there is no more improvement in fitness score between a given number of generations.

With SSD-Tailor, we reduce the number of simulations needed to obtain the optimal parameters with the GA. If the performance score of two parameter sets is the same, SSD-Tailor selects that with the lowest cost.

III. EVALUATION

We evaluate SSD-Tailor in terms of efficiency and performance by considering the number of simulations and response time, respectively. Enterprise server I/O workloads are used in our evaluation. Workloads traced from financial servers [6] are random-write intensive and comprise small-size requests. For the simulation, we use DiskSim + SSD Extension [3], [5]. Table I shows the set of SSD design parameters selected for this evaluation. We also insert range and parameter cost information into SSD-Tailor.

TABLE II
NUMBER OF SIMULATIONS

	Brute Force method	SSD-Tailor
Simulation count	14400	50 + 20(finish condition)

Finish condition is an exit that is no more improvement between next 2 generations

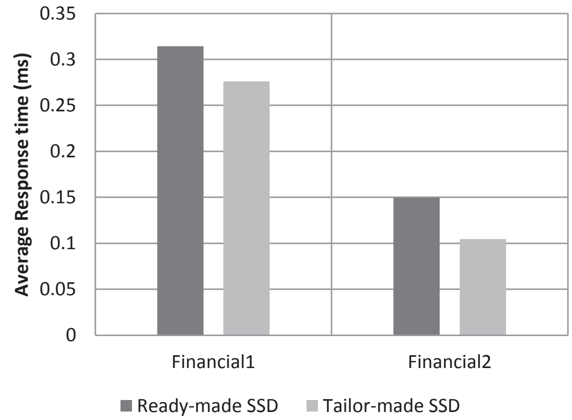

Fig. 2. Experiment results for SSD-Tailor and a ready-made SSD.

First, we demonstrate the efficiency of SSD-Tailor by comparing the number of simulations needed to find the best parameter set using the GA-based SSD-Tailor and a brute-force algorithm. For the given workloads, SSD-Tailor attempts to find the parameter set that achieves the lowest average response time. As a result, SSD-Tailor and the brute-force algorithm find the same parameter sets. The number of simulations is shown in Table II. Whereas the brute-force algorithm attempts all parameter combinations, SSD-Tailor simulates only a few dominant parameters.

Second, to demonstrate the performance of our tailor-made SSD, we compare it to a ready-made SSD with a pre-defined parameter set. As shown in Fig. 2, SSD-Tailor produces an SSD that is optimized for the financial workloads, reducing the average response time by 12% and 30% for the two sets of data.

IV. CONCLUSION

In this paper, we have proposed SSD-Tailor for customizing SSD design parameters using a GA. Enterprise servers have different performance requirements according to their different purposes and objectives. The tailor-made SSD satisfies these requirements by customizing the SSD parameters. Using our technology, consumer electronic devices will be able to fulfill various performance requirements.

EXAMPLES OF REFERENCE STYLES

[1] F. Chen, D. A. Koufaty, and X. Zhang, "Understanding Intrinsic Characteristics and System Implications of Flash Memory based Solid State Drives," ACM SIGMETRICS Performance Evaluation Review, Pages 181-192, June 2009

[2] G. Wu and X. He. "Reducing SSD Read Latency via NAND Flash Program and Erase Suspension," In 10th USENIX Conference on File and Storage Technologies (FAST12), 2012.

[3] N. Agrawal, V. Prabhakaran, T. Wobber, J. D. Davis, M. Manasse, and R. Panigrahy. "Design tradeoffs for SSD performance," USENIX 2008 Annual Technical Conference, pages 57-70, 2008

[4] R. L. Haupt and S. E. Haupt. "Practical Genetic Algorithms," Wiley-Interscience, 2004.

[5] S. Lee, T. Kim, K. Kim, and J. Kim. "Lifetime Management of Flash-Based SSDs Using Recovery-Aware Dynamic Throttling," In 10th USENIX Conference on File and Storage Technologies (FAST12), 2012.

[6] "OLTP Application I/O" http://traces.cs.umass.edu/index.php/Storage