SEA-SSD: A Storage Engine Assisted SSD with Application-Coupled Simulation Platform

CHAO SUN†, ASUKA ARAKAWA† and KEN TAKEUCHI†

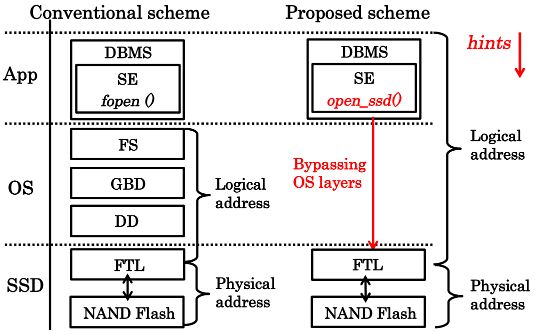
In this paper, a storage engine assisted solid-state drive (SEA-SSD) has been proposed to improve the storage performance for the database application by co-optimizing the SSD controller and database storage engine. Data with different activities are predicted, classified and aggregated into the same block of the NAND flash memory. From the experimental results, maximum 24% speed boost, 16% power reduction and 19% endurance enhancement are achieved, without requiring a cache layer for the SSD.

1. Introduction

Solid-state drives (SSDs) are replacing the hard disk drives (HDDs) as the primary storage due to the advantages in speed, reliability and power. In the conventional database application, SSDs are used as the cache for HDDs to balance the performance benefits and costs. Since the bit-cost of NAND flash continuously decreases, the SSDs are proposed as the storage for the database. Since the storage engine (SE), a software component in the database management system (DBMS), controls the data storage. The data information inside the SE is utilized to improve the SSD write performance, which is relatively slow due to the inherent characteristics of the NAND flash memories.

2. SSD Based DBMS

The garbage collection (GC) overhead of the SSD is the write performance bottleneck of the SSD [1]. One approach to reduce the GC overhead is clustering data with similar activities in the same block of the NAND flash. To realize it, upstream information from the SE is utilized. As illustrated in Fig. 1, each layer of the storage stacks of the operating system (OS) has to be modified to pass the hint information from SE to the flash translation layer (FTL) in the SSD controller, conventionally [2]. It introduces additional overhead and engineering efforts. Hence, the OS is bypassed in the proposed SSD-based DBMS. Hint information passes from the SE to the FTL directly.



App: Application, DBMS: database management system,
SE: Storage engine, FS: File system, GBD: Generic block device,
DD: Device driver, FTL: Flash translation layer (SSD controller)

Fig. 1. The proposed SSD-based DBMS [2].

3. Storage Engine Assisted SSD (SEA-SSD)

Figure 2 describes the overall architecture of the proposed SEA-SSD [2]. All the data from the database are classified into the dynamic and static data. Thus, the SSD is partitioned into two segments: *Segment_dyn* for the dynamic data and *Segment_stat* for the static data. Three kinds of hint messages are passed from the InnoDB SE. The first hint passes the database setting hints such as the redo log and buffer pool size to initialize the size of *Segment_dyn* in the SSD. The second hint preliminarily classifies the data in the SE with a dynamic data model represented by the SE internal information such as the log sequence number. As shown in Algorithm I of Fig. 3, when the second hint is '*SEG_dyn*', or logical value 1, the data is written to *Segment_dyn* [2]. Otherwise, it is written to *Segment_stat*. Last, the third hint predicts the

† Faculty of Science and Engineering, Chuo University

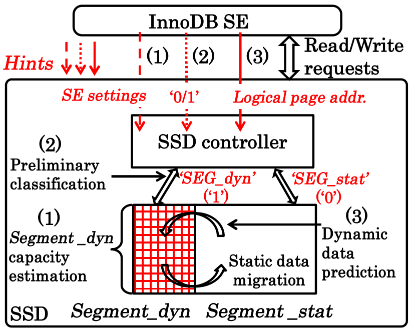


Fig. 2. The overview of the SEA-SSD [2].

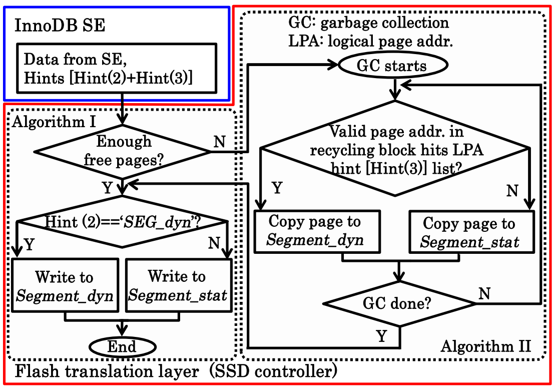


Fig. 3. SSD management algorithm flow chart [2].

dynamic data with the flush list. When new data enters the flush list, it indicates that this data will be flushed to storage in the near future. Hence, the logical page address (LPA) of the data is sent to the SSD controller. When GC is triggered, the data of the hinted LPAs will migrate from *Segment_stat* to *Segment_dyn*, described in Algorithm II of Fig. 3.

4. Evaluation

An application-coupled platform has been designed to evaluate the proposed scheme. In this work, the database is directly coupled with the SSD for optimizing the SSD controller. SSD is virtualized for the simulation acceleration. As mentioned earlier, the OS is bypassed in the interaction between the SSD and database. Table I describes the SSD workload settings [2]. Online transaction processing (OLTP) benchmarks, Sysbench [3] and TPC-C [4], are used. From the experimental results, maximum 24% SSD speed boost, 16% power reduction and 19% endurance enhancement are achieved at the SSD capacity of 3 GB (33% over-provisioning) [2]. As illustrated in Fig. 4, the SEA-SSD improves the SSD performance by 7%-24% at a reasonable over-provisioning range of less than 50%.

Table I. Benchmark statistics for evaluation [2].

MySQL benchmark	SSD workload	Buffer pool size (MB)	Redo log size (MB)	Write size (GB)	Read size (GB)	Write request ratio
TPC-C	T1	256	4	25.1	41.7	37.5%
	T2	512	8	21.9	15.7	58.2%
	T3	1024	16	19.3	4.07	82.6%
Sysbench	S1	256	4	11.1	34.1	24.5%
	S2	512	8	8.93	14.1	38.8%
	S3	1024	16	7.94	0.97	89.1%

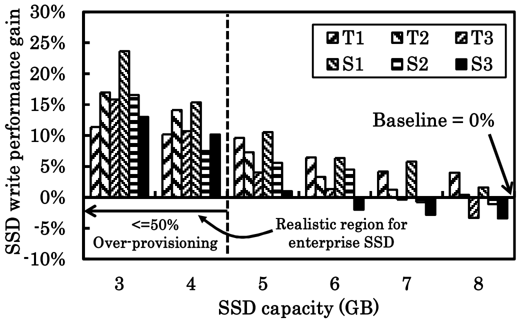


Fig. 4. SEA-SSD write performance evaluation [2].

5. Conclusion

A SE assisted SSD has been proposed to improve the SSD performance for the database application. By obtaining the hint messages from the upstream layer, specifically SE of the database, the data are better classified and more efficiently stored in the SSD. Consequently, the garbage collection overhead of the SSD is reduced and maximum 24% SSD performance improvement is achieved.

Reference

[1] K. Takeuchi et al., "A 56 nm CMOS 99 mm^2 8 Gb multi-level NAND flash memory with 10 Mbyte/sec program throughput," *IEEE J. of Solid-State Circuits (JSSC)*, vol. 42, no. 1, pp. 219-232, 2007.

[2] C. Sun et al., "SEA-SSD: a storage enegine assisted SSD with application-coupled simulation platform," *IEEE Trans. on Circuits and System I (TCAS-I): Regular Papers, 2014.* (accepted)

[3] http://sysbench.sourceforge.net/.

[4] http://www.tpc.org.