High Performance and High Capacity Hybrid Shingled-Recording Disk System

Jiguang Wan[12]*, Nannan Zhao[2]†, Yifeng Zhu[3]‡, Jibin Wang[2], Yu Mao[2], Peng Chen[2] and Changsheng Xie[12]§
[1]*Wuhan National Laboratory for Optoelectronics, Wuhan, China*
[2]*Computer science and technology, Huazhong University of Science and Technology, Wuhan, China*
[3]*Electrical and Computer Engineering, University of Maine, USA*
**jgwan@mail.hust.edu.cn, †nnzhaocs@hotmail.com, ‡zhu@eece.maine.edu, §cs_xie@mail.hust.edu.cn*
Corresponding author: Changsheng Xie

***Abstract*—Areal density scaling in magnetic hard drives is in jeopardy as magnetic particles become unstable when they are sufficiently small. Shingled recording holds great promise to mitigate the problem of density scaling cost-effectively by overlapping data tracks. However, this innovative technology suffers severely from slow small writes. This prevents shingle recording from being widely adapted in practice. This paper presents a new hybrid storage architecture that combines a shingled-recording magnetic disk and a fast SSD cache to achieve a high-capacity storage system without any compromise to performance. We propose a new wave-like shingled recording that overlaps adjacent tracks from two opposite radial directions. This new schemes doubles the areal density of conventional circular log-based shingled recording. We also design a new replacement strategy to manage the hybrid system to effectively eliminate the performance degradation. We evaluate our design based on a prototype implementation. Experimental results under 12 I/O workloads show that our hybrid system exhibits a sustained performance comparable to a disk with no shingled-recording.**

***Keywords*-Shingled recording; SSD; Hybrid system; Replacement policy; Data layout**

I. Introduction

The areal density of magnetic disks is reaching its length-scale limitation. The capacity of a magnetic disk has increased 30%-50% per year for almost 50 years and currently disks can store up to 400Gbits/in^2 [12]. Disk areal density is quickly approaching to 1Tbit/in^2, a limit caused by superparamagnetic effect [5]. The magnetic direction of a sufficiently small particle can be randomly flipped under the influence of ambient thermal energy. New recording technologies have been proposed to scale up the areal densities, such as bit-patterned media [13] and heat-assisted magnetic recording [4]. Compared with them, shingled recording technology is the most promising one since it can notably increase the grain density without changing underline storage media. By partially overlapping tracks to reduce track width, the areal density can be improved to 2-3Tb/inch^2 [3]. Combined with 2-D readback and new signal processing techniques [10] [6], the areal density can be further enhanced. The inherent weakness defect of shingled recording is slow small writes, also called random writes, because writing to a given data track requires rewriting to its neighbor tracks. The amount of data actually written is larger than the write request size. The inferior performance for small writes, also called write amplification, is one of major factors restricting its widespread deployment in real systems.

The key challenge of extending the application of shingled writing and integrating it into magnetic disk system is to lower the write amplification of shingled writing disk. Cassuto et al. [11] constructed an indirection system for shingled recording disk and introduced a data layout that organize data into circular log named S-block architecture. However, the circular log-based data layout has a large data immigration overhead during garbage collection. Moreover, it has to maintain a large amount of metadata for tracking the dynamic mapping between a logical address and its physical location. Park et al. [16] proposed a H-WSD architecture by adopting a hot data identification mechanism to reduce the garbage collection overhead. But this unbalanced garbage collection still relies on data immigration, because hot data that are frequently updated are mostly located in the head of circular log and many valid tail data need to be moved to head to collect the invalid head data during garbage collection.

A different approch to alleviate the random write issue is hybrid systems that leverage a small non-volatile RAM or SSD for effectively caching hot data and reducing data immigration and improve performance. Currently SSD outperform disks significantly in both read and write, particularly in random read [17]. Using SSD for caching random reads is very helpful to improve the performance of shingled disk system.

This paper proposes a hybrid wave-like shingled recording disk system (HWSR) to improve both the performance and the capacity of a shingled recording disk. HWSR contains three different storage media: memory, SSD, and hard disk. The memory has a very small capacity, such as 100MB, in our design to reduce the overall cost. It is used to buffer hot writes. The SSD is used as a disk cache to improve random read performance.

HWSR consists of three key components: (1) a new data layout based on segmentation for shingled recording disk to reduce random write amplification; (2) a new shingled track layout named wave-like shingled recording (WSR) to

978-0-7695-4807-4/12 $26.00 © 2012 IEEE
DOI 10.1109/CLUSTER.2012.21

further improve its capacity; (3) a new replacement policy based on least write amplification that effectively reduces the miss rate and data rewritten amount.

The key contributions of this paper are as follows.

- Limited random write amplification to a single segment by using a new data layout based on segmentation, which is much smaller than a region.
- A wave-like shingled recording that reduce half of wasted space and improve disk utilization rate.
- Design and implementation of a new replacement policy based on least write amplification that greatly reduces the miss rate and data immigration.
- A hybrid storage system that integrates these components to improve both the capacity and the performance.

The rest of this paper is organized as follows. Section II gives an overview of HWSR system and presents our data layout, address mapping structure and replacement policy. Section III discusses three key issues: write amplification, disk utilization, metadata amount. The experiment and evaluation are presented in Section IV followed by related work in Section V. Section VI concludes this paper.

II. Hybrid Wave-like Shingled-Recording Disk

In this section, we describe our system model and give an overview of HWSR.

A. Design Overview

Figure 1 shows the HWSR system architecture consisting of a SSD cache, a MEM buffer and a shingled-recording disk. While the MEM buffer mainly stores write requests, the SSD cache mainly stores read requests. When a write request arrives, MEM buffer stores the incoming data and update the address mapping table. When MEM buffer is full, MEM buffer evicts out some cold data to make space for new data. All evicted data is written to the disk. When a read request arrives, MEM buffer looks up the address mapping table. The read request accesses SSD if MEM buffer does not hold the target data. The shingled disk serves all misses of the SSD cache.

B. Data layout

1) Wave-like shingled recording disk: In a traditional shingled disk, the tracks are laid out with partial overlap in the radial direction as shown in Figure 2(b). We assume that the number of tracks that are overwritten by a shingled write is κ, where κ is typically 4-8 [12]. Theoretically, if there are no wasted space (guard band), the maximum capacity of a shingled disk is κ times traditional disk as shown in Figure 2(a). Via overlapping tracks, the average track width is significantly reduced.

Data tracks are organized into bands called regions, which are separated by a collection of p following tracks called guard band, where p is at least κ. Guard bands are to prevent the interference between regions and do not store any valid data. Thus guard bands creates significant spatial overhead. The tracks in wave-like shingled recording, as shown in Figure 2(c), are laid out with partial overlap in two opposite radial directions like waves. There is only one guard band shared by every two regions, which means that on average only half guard band is wasted for each region. Compared with traditional shingled recording, our Wave-like approach reduces the spacial overhead of traditional shingled disk by half, resulting in significantly improve the utilization rate. We will discuss in detail the disk utilization ratio of wave-like shingled recording to traditional shingled recording in Section III.

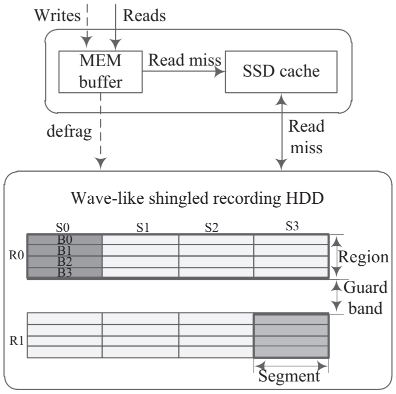
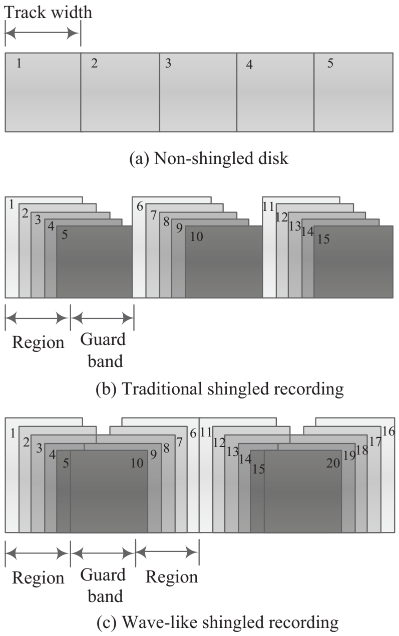


Figure 1. HWSR system architecture



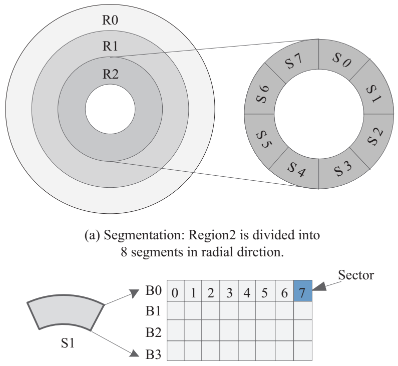
(a) Non-shingled disk

(b) Traditional shingled recording

(c) Wave-like shingled recording

Figure 2. Track layout

2) Segment-based data layout: The number of tracks in each region influences significantly the disk performance, since a random write to a given track may require rewriting



(a) Segmentation: Region2 is divided into 8 segments in radial dirction.

(b) Blocks: In each segment, the sectors in the same track constitute one block. There are four blocks in Segment1and each block contains 8 sectors.

Figure 3. Data layout based on segmentation

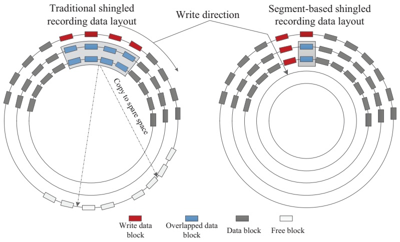


Figure 4. Comparison of write amplification in tradition layout and our proposed segment-based layout

the whole region. We divide a region into segments in the radial direction to reduce write amplification. The size of data rewritten is then limited to a segment, which is much smaller than a region. We assume four contiguous tracks on the same surface constitute a region. Figure 3 shows an example of three regions, with eight segments in each region. With in a segment, all sectors in the same track constitute a data block. In this example, a segment has four data blocks and each block has eight sectors. If data in segment S1 is updated, its neighbor segments S2 and S0 are not affected.

An example of write amplification is provided in Figure 4, in which κ is set as two. when four sequential data blocks marked in red are written to a traditional shingled disk, eight data blocks in the adjacent tracks have to be rewritten, i.e., we need to copy the eight data blocks to some spare space and then rewrite them back. If we divide each region into multiple segments in the radical direction and assume each segment holds three data blocks, then only three sequential data blocks are laid out in the radical direction and only two data blocks in the two adjacent tracks are rewritten. This segment-based shingled recording reduces the effect of write amplification to a quarter of traditional shingled data layout. Generally, the write amplification of data layout based on segmentation is 1/*n* of traditional data layouts, where *n* is the total number of segments in a region.

C. Address mapping structure

To reduce the amount of metadata and ensure consistence, we construct the address mapping structure between cache, buffer and disk. The logical address space of MEM buffer and SSD cache is divided into independent blocks which are the same size with the blocks of shingled disk as shown in the right side of Figure 5. There are only one hash table which mapping the logical address of requests to the cache or buffer space. If a request is on a miss, it accesses the disk directly using the following address transform formula:

$$\begin{aligned} N_{reg} &= LBA/S_{reg} && (1)\\ N_{seg} &= (LBA \bmod S_{reg})/S_{seg} && (2)\\ N_{blk} &= (LBA \bmod S_{seg})/S_{blk} && (3)\\ N_{offset} &= LBA \bmod S_{blk} && (4)\end{aligned}$$

$$PBA = N_{reg}*S_{reg}+N_{blk}*S_T+N_{seg}*S_{blk}+N_{offset} \quad (5)$$

where LBA and PBA are the logical and physical address respectively(in sectors); N_{reg}, N_{seg}, N_{blk} and N_{offset} are the logical section number, segment number, block number and sector number respectively; S_T is the track size (For the convenience of representation, we assume all tracks have the same size); S_{reg}, S_{sec} and S_{blk} are region size, section size and block size respectively (in sectors).

We construct a segment hash table (SHT) which stores segment information in each hash node. Each segment node contains two types of arrays that store block locations. *SBmap* stores the block location in the SSD cache and *MBmap* stores the block location in MEM buffer as shown in the middle of Figure 5.

SHT is used to speed up the lookup in SBmap and MBmap. If an element in one array is valid, then the requested data content is stored at the location of corresponding devices. If invalid, then the requested data is not in SSD and Mem, and then the disk will be accessed. Figure 5 gives an example of data lookup operations for two read requests (LBA1 and LBA3) and one write request (LBA2) in HWSR. We assume that the number of blocks per segment is four and the segment hash value is $N_s\%4$, where N_s is the segment number. In case of read request LBA1 (S5, B3, R), the content of bracket mean that the read request accesses the block 3 in the segment 5, MBmap[3] and SBmap[3] are both NULL, then the request accesses shingled recording disk and the desired data is prefetched to SSD cache. In the case of LBA2, the content of MBmap[3] is valid, so the location of the target block is stored in block 4 of the MEM buffer. In the case of LBA3, the content of SBmap[2] is SB0, which mean the block required is stored in block 0 in SSD cache.

D. Replacement policy

On a miss, cache or buffer must select a block to be replaced. HWSR uses different replacement policies for SSD

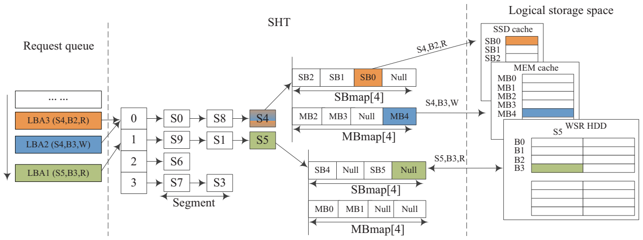


Figure 5. Address mapping translation

cache and MEM buffer. Note that MEM buffer is mainly used to buffer writes and SSD cache is only used to cache reads because of its limited write cycles. In addition evicted blocks out of MEM buffer must be written to the disk to maintain consistence while the replaced blocks out of SSD cache are not written to disk because the data isn't changed.

As price per byte of SSD continues to decrease. SSD with a relative small capacity (several hundred Mbytes) does not incur a significant cost. However, such as a small SSD can effectively capture most hot data and have a high hit rate, as proved by our experimental results presented later. For simplicity, we use LRU to manage the SSD cache.

There are two LRU queues in MEM buffer: one for blocks and the other for segments. The LRU block queue takes advantage of temporal locality but it replaces only one block at a time. This increases the number of defrag and write amplification because of rewriting about a segment per defrag. The LRU segment queue replaces a segment each time and reduce write amplification. But it wastes a lot of cache space. For example, some blocks are hot while the other blocks are cold in a segment. Then this segment is in the front of LRU queue and its cold blocks will never be replaced.

We introduce an new LRU algorithm based on the least write amplification for MEM buffer as shown in Figure 6. We construct an LRU circular linked list of blocks. An incoming write is added to the head of the circular list. During a miss, the victim block is chosen within a predefined window starting from the tail of the circular link. The victim block should belong to the same segment as MB13. MB15 and MB12 belong to the same segment (segment1) as MB13. Because replaced blocks often belong to the same segment, which limits the rewrites to one segment, this algorithm not only maintains the most hot data and discards the cold data, but also reduces write amplification.

III. Discussion

In this section, we discuss some key issues existing in our HWSR and S-block architecture which is a typical example of conventional shingled recording disk based on circular log. Those key issues have a significant impact on the performance of shingled recording disk.

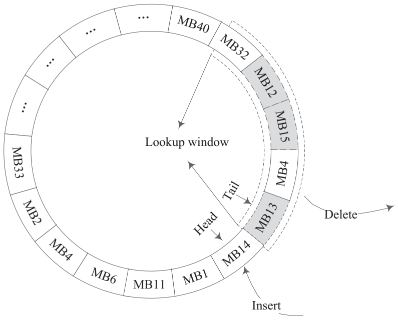


Figure 6. LRU algorithm based on least data immigration

A. Write Amplification

The SSD and MEM works similar to a two-level hierarchical cache to the shingled disk. Upon a miss, it leads to prefetching data from the disk or defraging data to disk. Compared with the S-block architecture [11] that collects garbage when the request is on a miss, our HWSR system only rewrites one segment at most. The average number of valid Sblocks that have to be immigrated to the head in garbage collection is ω as shown in Table I. Note that the Sblock in S-block architecture is the same size as segment in HWSR, so we replace Sblock with segment for convenience. For our HWSR, we also need to rewrite some blocks in defrag operation when the request is a miss. However, the amount of rewrite data of HWSR is limited to the size of one segment when defrag occurs. As a result, the ratio of the average data immigration of S-block to HWSR is ω.

Assume the number of segments in a region is n, and the spare segments in each region is β*n. The spare segments are used to reduce ω in S-block architecture. Generally, $\beta \neq 0$. If $\beta = 0$, the entire segments of circular log are immigrated to the head to free one invalid S-block when there are only one invalid segment in the head of circular log. Write amplification is the total number of segments in circular log. Because each incoming update write is added to the head of circular log and the head segment are frequently updated, which means that in most cases invalid segments are mostly concentrated to the head of circular log and almost all segments have to be immigrated to the head in order to free the invalid head segment. Approximately, ω can be considered as $\frac{1}{\beta}$, because we write β*n to a region and it will cause approximately n Sblocks immigrated when the whole segment is filled with n Sblocks. Note that there are always β*n invalid segments (spare segments) in each region. Moving all segments in circular log can release at least β*n invalid segments. We neglect the cases that there are some invalid segments in the tail or near the tail of circular log. This is reasonable because after a long term of writing, there are less invalid segments in the tail of circular log in the whole process.

Figure 7 compares the average segment immigration of

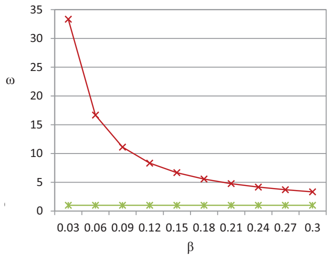
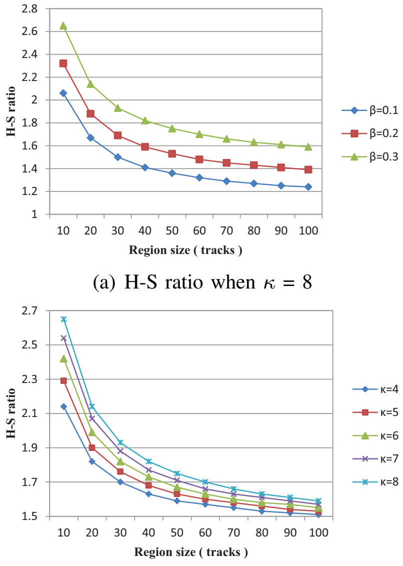


Figure 7. Average segment immigration of S-block architecture in garbage collection

Table I
KEY PARAMETERS OF WORKLOADS

Parameters	
ω	The average segment immigration of S-block architecture in garbage collection.
κ	The number of tracks which are overlapped with the adjacent track
β	The spare capacity rate per region
n	Each region consists of n segments (or Sblocks)
H-S ratio	The disk utilization ratio of HWSR system to S-block architecture
S_{reg}	Region size (in tracks)



(a) H-S ratio when κ = 8

(b) H-S ratio when β = 0.3

Figure 8. H-S ratio

S-block architecture with different spare capacity rate and the average segment rewritten of HWSR. As β increases, the average segment immigration decreases.

B. Disk Utilization

As discussed previously, we assume that each region consists of n segments. The guard band and internal guard band both contains κ tracks, where κ is the number of tracks which are overlapped with the adjacent track. There are internal guard bands in circular log-based data layout. Internal guard band is used to prevent the written data blocks from affecting existing blocks in each region. There are β*n spare segments that aim to mitigate the data immigration in S-block architecture. The region utilization rate of HWSR is 1 because HWSR doesn't need any spare segments in each region. The region utilization rate of S-block architecture is $\frac{S_{reg}-\beta*S_{reg}}{S_{reg}}$. The region utilization ratio of HWSR to S-lock architecture is $\frac{1}{1-\beta}$. We also take the guard band and internal guard band into consideration.

The track utilization rate of S-block architecture which uses traditional shingled recording is $\frac{S_{reg}}{S_{reg}+2\kappa}$. The track utilization of HWSR is $\frac{S_{reg}}{S_{reg}+\kappa/2}$. The track utilization ratio of HWSR to S-block architecture is $\frac{2*(S_{reg}+2*\kappa)}{2*S_{reg}+\kappa}$. So the disk utilization ratio of HWSR system to S-block is $\frac{2*(S_{reg}+2*\kappa)}{(2*S_{reg}+\kappa)(1-\beta)}$.

Figure 8(a) compares the disk utilization ratio of HWSR to S-block when κ is eight and the region size varies. We observe that the disk utilization of HWSR is 2.65 × of S-block architecture when S_{reg} is 10 and β is 0.3. Figure 8(b) studies the disk utilization ratio when β is fixed to 0.3. The H-S ratio has a maximum value of 2.65 and it decreases gradually as S_{reg} increases. The H-S ratio increases if κ and β increases. As shown in Figure 8, disk utilization rate of HWSR doubles the traditional circular log-based shingled recording disk. As a result, HWSR improves the capacity of a shingled writing disk by around 2 times.

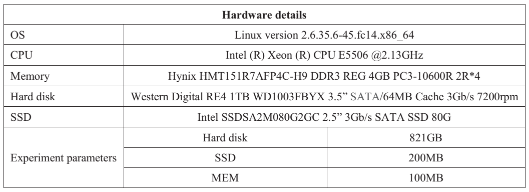
C. Metadata

HWSR directly uses address translation and it doesn't require a mapping table to map a logical address to it physical location on the disk. The data layout based on circular log requires a translation table to map LBAs to PBAs because their mapping information are not fixed. A large translation table creates significant overhead in a circular log-based shingled disk. According to Ref. [16], 1TB shingled writing disk requires over 24GB space to store the metadata, which will certain require a large memory space. In addition, data lookup in such a large table is often very slow.

D. Design issues

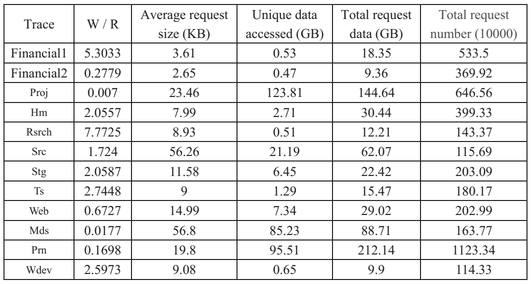
Higher average seek time is a key problem in our HWSR because of segmentation along the radical direction. Sequential data blocks are distributed on three adjacent tracks as shown in Figure 4. If the required data is located on multiple blocks, the traditional sequential data layout shingled recording has less seek time than the data layout based on segmentation. In our design, we make a tradeoff between write application and average seek time. There are three reasons. (1) Based on the observation in our experiments, the writes latency contributes to a large proportion of total latency, and the segment-based data layout optimizes the write performance due to lower write amplification. (2)

Table II
HARDWARE DETAILS



Hardware details		
OS	Linux version 2.6.35.6-45.fc14.x86_64	
CPU	Intel (R) Xeon (R) CPU E5506 @2.13GHz	
Memory	Hynix HMT151R7AFP4C-H9 DDR3 REG 4GB PC3-10600R 2R*4	
Hard disk	Western Digital RE4 1TB WD1003FBYX 3.5" SATA/64MB Cache 3Gb/s 7200rpm	
SSD	Intel SSDSA2M080G2GC 2.5" 3Gb/s SATA SSD 80G	
Experiment parameters	Hard disk	821GB
	SSD	200MB
	MEM	100MB

Table III
PARAMETER LIST



Trace	W / R	Average request size (KB)	Unique data accessed (GB)	Total request data (GB)	Total request number (10000)
Financial1	5.3033	3.61	0.53	18.35	533.5
Financial2	0.2779	2.65	0.47	9.36	369.92
Proj	0.007	23.46	123.81	144.64	646.56
Hm	2.0557	7.99	2.71	30.44	399.33
Rsrch	7.7725	8.93	0.51	12.21	143.37
Src	1.724	56.26	21.19	62.07	115.69
Stg	2.0587	11.58	6.45	22.42	203.09
Ts	2.7448	9	1.29	15.47	180.17
Web	0.6727	14.99	7.34	29.02	202.99
Mds	0.0177	56.8	85.23	88.71	163.77
Prn	0.1698	19.8	95.51	212.14	1123.34
Wdev	2.5973	9.08	0.65	9.9	114.33

HWSR use SSD to cache reads, which effectively reduces the number of random reads to disk, resulting in better read performance. (3) Compared with circular log-based data layout, which moves valid blocks from the tail to the head during garbage collection and thus generate a lot migration data traffic, HWSR exhibits stable performance even under the workload with a lot random access as shown in the following section.

IV. Experimental Evaluation

We implemented a prototype of HWSR. This shingled disk is emulated by using a conventional disk. The capacity of each track on the disk is 8×1024 sectors (512 bytes in one sector). We constructed S-block architecture by organizing data as circular log. Except the data layout of shingled disk, the parameters of buffer or cache of S-block architecture is the same as HWSR. The hardware details and our experiment parameters are shown in Table II. We directly replay 12 different I/O traces in our prototype implementation. The key characteristics of these traces are summarized in Table III. The first two traces (Financial1 and Financial2) are collected from OLPT applications which run at two large financial institutions [19]. The other ten traces are collected from enterprise servers at Microsoft Research Cambridge [20].

A. Performance of random Access

In our experiments, a block has 64 sectors and a segment has 20 tracks. In addition, we used a traditional disk without shingled tracks in our prototype. The traditional disk also has a MEM buffer which has the same size in S-block architecture and HWSR.

We show the response time of each 10,000 requests for HWSR and S-block in Figure 9(a). The first 80,000 requests of Financial1 exhibit a very low response time and a high performance, until the MEM buffer is full and defrag operations start, the performance drops dramatically as shown by red curve of S-block.

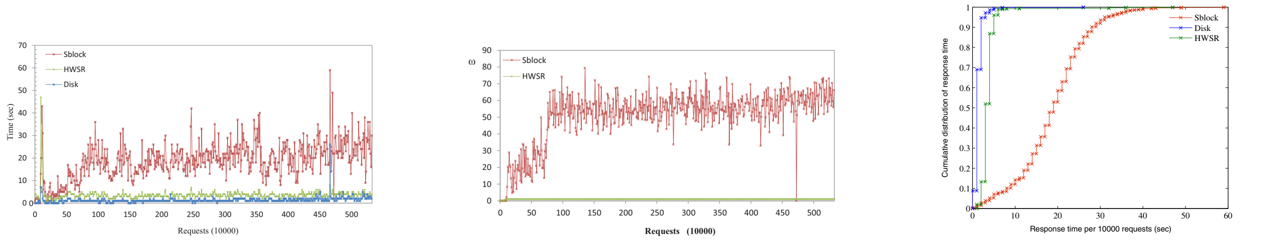
Figure 9(b) plots the number of Sblocks immigrated or segments rewritten per 10,000 requests on Financial1. The first 80,000 requests have a small number of Sblocks immigrated. Note that the average number of segment rewritten of HWSR is only 1. The sharp pattern of the curve of S-block architecture in Figure 9(a) is similar to Figure 9(b). This indicates that as more data is immigrated, the system performance is degraded more severely. Compared with S-block, the green curve of HWSR exhibits a stable and superior performance.

Traditional magnetic disk has the lowest response time as shown by blue curve. The performance of HWSR almost reaches the performance of magnetic disk. Figure 9(c) plot the cumulative distribution of response time with Financial1. About 90% response time (per 10,000 requests) of HWSR is less than 5 seconds while about 90% response time of S-block architecture is around 27 seconds. Figure 10 plots the response time, average number of Sblock immigration or segment rewritten and cumulative distribution of Financial2. S-block architecture has a lower data immigration and exhibits good performance until it compete the first 490,000 requests. It performs data immigration later than Financial1.

And with a read intensive workload of Financial2, the performance of HWSR exceeds magnetic disk because HWSR uses SSD cache while magnetic disk doesn't, which mean that combined with SSD, HWSR can almost reach the performance of a magnetic disk with no shingled recording.

B. Block size and segment size

We study the performance impact of block size and segment size in both HWSR and S-block. Figure 11 plots the read, write and total average response time with different block sizes. Solid line represents the average response time for both reads and writes. In financial1, the average response time of HWSR and S-block is 0.3ms and1.8ms respectively when the block size is 64 sectors. When the block size is reduced to 16 sectors, the average response time of HWSR and S-block increase to 0.8ms and 5.7ms respectively (see Figure 11(a)). We observe that when the block size is increased, the performance of HWSR and S-block become better in the workload of Financial1, which is opposite to Financial2. Comparing the two different traces, we observe that Financial 1 has more writes and larger request size. A larger block can capture more requests in Financial1 and accordingly less requests are separated to different blocks. For HWSR, the average seek time is reduced because less requested data are divided into different blocks which are mostly located on two adjacent tracks. For S-block

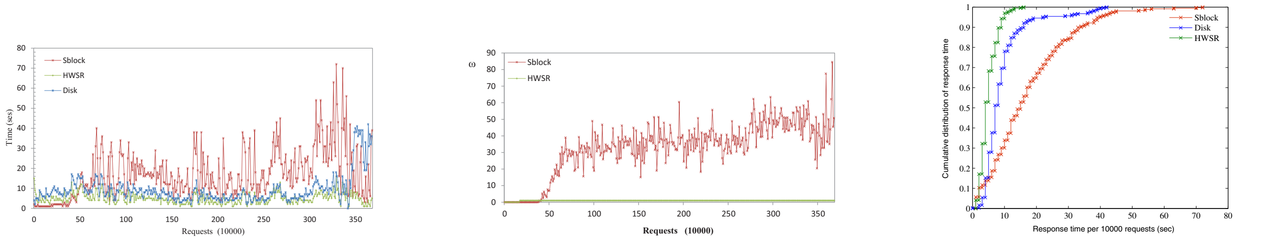


(a) Response time per 10000 requests

(b) The average segment immigration of S-block architecture in garbage collection and average segment rewritten of HWSR

(c) Cumulative distribution of response time

Figure 9. Financial1



(a) Response time per 10000 requests

(b) The average segment immigration of S-block architecture in garbage collection and average segment rewritten of HWSR

(c) Cumulative distribution of response time

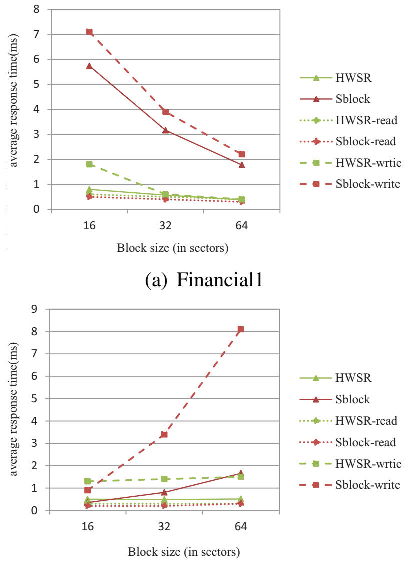
Figure 10. Financial2

architecture, the buffer hit rate is higher and thus less data immigrates occur.

HWSR exhibits a slightly lower performance on reads compared with S-block architecture as shown in Figure 11. The average read response time is only about 0.1ms more than S-block architecture, because some request data is on two blocks and read two blocks on adjacent tracks cost more seek time than reading two sequential blocks on one track. For financial2, the average write time of S-block dramatically increase when the block size increases. The maximum average time of S-block is 1.7ms when the block size is 64 sectors.

HWSR exhibits a stable performance under these two workloads. The write latency accounts for most proportions of the total latency, thus HWSR often achieves a better performance than S-block due to our write optimization. When the block size is 16 sectors, S-block architecture is slightly better than than HWSR. As the block size decreases, more blocks can be cached in MEM cache and thus less data immigrations occurs in S-block architecture under the Financial2 workload. This shows that the performance of S-block is sensitive to the change of block size. HWSR exhibit a stable performance under different block sizes.

Segment size has a slight impact on the performance of HWSR as shown in Figure 12(a). The response time of S-block decreases when the region size gets bigger as shown in Figure 12(b) while the response time of HWSR increases.



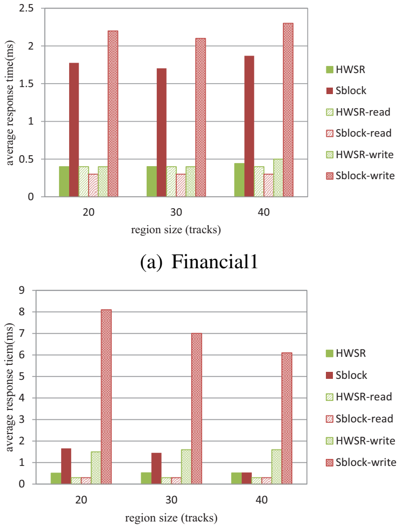
(a) Financial1

(b) Financial2

Figure 11. average response time with different block size

C. workload history impact

Figure 13 shows the average response time of HWSR system and S-block architecture with $\beta = 0.1$ on ten Microsoft traces. The size of MEM buffer and SSD cache we used



(a) Financial1

(b) Financial2

Figure 12. average response time with different segment

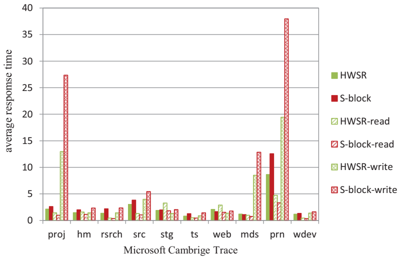


Figure 13. Average response time with a spare capacity of 0.2% on MSR Cambridge Traces

with Microsoft traces is same as Financial1 and Financial2. As shown in Figure 13, compared with S-block architecture, HWSR has a significant lower write average response time and total response time with most traces except web and mds because the read hit rate is very low. The average write response time of S-block is around twice HWSR with proj and prn. But the average read response time of HWSR is a little bit higher than S-block architecture.

V. Related Work

Recently many researches work on high density recording technology.

One approach of improving the areal density of magnetic disks is to change recording medium to avoid the superparamagnetic limit. Examples include Bit-patterned magnetic recording (BPMR), Heat-assisted magnetic recording (HAMR), and microwave assisted magnetic recording (MAMR). BPMR stores each recording bit in a fabricated magnetic island of around 10nm [13] [6] to extend the current superparamagnetic limit. HAMR softens the magnetic material to make it easier to magnetize [4]. MAMR [12] [8] make the thermally stable and hard-to-write media more writable by using microwaves focusing on small areas. All above technologies dramatically change the structure of underlying and mechanical design and disk head sensors of existing magnetic disk, which might introduce significant costs to disk manufacture.

A different approach to achieve high density is shingled recording technology that partially overlaps tracks to narrow the track width. Cross and Montemorra [15] demonstrated that by using a conventional disk head, the areal density of shingled recording disk can exceed 800Gb/in^2. And with the stronger write field, the areal density of shingled write disk can be increased to around 2Tb/in^2 [7] [3]. Miura [14] pointed out that high density can be attained when the reader is accurately placed on the center line of data tracks by analyzing different heads and media and estimating the maximum track density of shingled writing. Combined with 2-D readback and advanced signal processing, which referred to as 2-D magnetic recording (TDMR), can achieve an areal density of 10 Tb/in^2 [10] [6]. Reading from narrower tracks by using a wide reader causes inter-track interference (ITI). To address this problem, Ozaki et al. [18] proposed an ITI canceller to reproduce waveform from a shingled recording disk.

Shingled disks have attracted a lot of attentions due to its density improvement by at least 2T/in^2 [3] and its cost-effectiveness since it does not require to rework the storage media. But its inferiors performance, particularly for random writes, limit its application scope to minimal update workloads, such as archival workloads. Currently most existing research works tackle this problem by designing new data layout for shingled disks. Amer et al. [12] introduced a data layout of bands, with each band organized as circular log. All logs at each level store different data and take different clean strategies. Cassuto et al. [11] constructed an indirect system which contains two data layout methods. The first one is a disk cache based architecture that caches random writes and rewrites to native region through set associative mapping. The second one is S-block architecture which organizes data as a circular log. Park et al. [16] proposed H-SWD to reduce the garbage collection of circular log by using a hot data identification mechanism.

It is mentioned that SSD and NVRAM can be embedded as cache for delaying updates to shingled recording disks and reduce data rewritten [12] [6] [2]. Gibson and Ganger [1] proposed a shingled translation layer (STL) in an embedded controller to mask the random write restriction and integrated shingled writing into magnetic disks.

VI. Conclusions

In this paper, we presented a hybrid wave-like shingled recording disk system (HWSR) to address the problem of slow performance for small writes on shingled disks. We design the new data layout that can not only double the disk space utilization of conventional circular log-based shingled disks, but also effectively limit the write amplification to a small segment. Our hybrid system combines shingled disks with fast memory that works as buffer for writes and SSD that works as cache for reads. We design a new LRU algorithm based on least write amplification to optimize the overall I/O performance. Experimental evaluation on our prototype evaluation under a variety of I/O intensive workloads show that HWSR system improves the performance of small writes significantly. Results show that our design is not sensitive to many design parameters such as the block size and the segment size. While our new data layout slightly increase the average seek time as sequential data blocks are stored on adjacent tracks, such degradation can be effectively hide by the memory buffer and the SSD cache.

Acknowledgment

The authors would like to thank Yingjian Wang for the help in programming the emulation system, and Ding Li, Dong Wu and Weijiao Gong, for their constructive comments. The authors would also like to thank the anonymous reviewers of this paper.

This work is sponsored in part by the National Basic Research Program of China (973 Program) under Grant No.2011CB302303 and the National Natural Science Foundation of China under Grant No.60933002, and Natural Science Foundation of Hubei province (Grant NO. 2010CDB01605), and the HUST Fund under Grant No.2011QN053,No.2012QN101 and No.2011QN032, and the Fundamental Research Funds for the Central Universities, and the National Science Foundation (CNS #1117032, EAR 1027809, IIS #091663, CCF #0937988, CCF #0737583, CCF #0621493, EPS #0904155).

References

[1] G. Gibson, and G. ganger, "Principles of Operation for Shingled Disk Devices," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529-551, April 1955.

[2] G. Gibson and M. polte, "Directions for shingled-write and two-dimensional magnetic recording system architectures: Synergies with solid-state disks," Carnegie Mellon University Parallel Data Lab, Tech. Rep., May 2009, CMU-PDL-09-014.

[3] S. Greaves. Y. Kanai, and H. Muraoka,"Shingled recording for 2-3 Tbit/in^2", IEEE Transactions on Magnetics, Vol. 45, no. 10, PP. 3823-3829, Oct. 2009.

[4] M. Kryder, E. Gage, T. McDaniel, W. Challener, R. Rottmayer, G. Ju, Y.-T. Hsia, and M. Erden, "Heat assisted magnetic recording," Proceedings of the IEEE, vol. 96, no. 11, pp. 1810-1835, Nov. 2008.

[5] H. Richter, A. Dobin, O. Heinonen, K. Gao, R. veerdonk, R. Lynch, J. Xue, D. Weller, P. Asselin, M. Erden, and R. Brockie, "Recording on bit-patterned media at densities of 1Tb/in^2 and beyound," IEEE Transactions on Magnetics, vol. 42, no. 10, pp. 2255-2260, Oct. 2006.

[6] Y. Shiroishi, K. Fukuda, I. Tagawa, S. Takenoiri, H. Tanaka, and N. Yoshikawa, "Future options for HDD storage," IEEE Transactions on Magnetics, Vol. 45, no. 10, Oct. 2009.

[7] I. Tagawa and M. Williams, "High density data-storage using shingle-write," in Proceedings of the IEEE International Magnetics Conference, 2009.

[8] J. -G. Zhu, X. Zhu, and Y. Tang, "Microwave assisted magnetic recording," IEEE Transactions on Magnetics, Vol. 44, no. 1, pp. 125-131, Jan. 2008.

[9] R. Wood, "The feasibility of magnetic recording at 1 terabit per square inch," IEEE Transactions on magnetics, Vol. 36, no. 1, pp. 36-42, Jan. 2000.

[10] R. Wood, M. Williams, A. Kavcis, and J. Miles, "The feasibility of magnetic recording at 10 terabits per square inch on conventional media," IEEE Transactions on Magnetics, Vol. 45, no. 2, pp. 917-923, Feb. 2009.

[11] Y. Cassuto, M. A. A. Sanvido, C. Guyot, D. R. Hall, and Z. Z. Bandic, "Indirection systems for shingled-recording disk drives," in Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies(MSST), 2010.

[12] Amer, A. ,D. D. E. Long, E. L. Miller, J. -F. Paris, T. Swarz S. j. , "Design Issues for a Shingled Write Disk System," 26th IEEE Symposium on Massive Storage Systems and Technologies, May 2010.

[13] R. L. White, R. M. H. New, and R. F. W. Pease,"Patterned media: A viable route to 50 Gbit/in^2 and Up for magnetic recording?," IEEE Transactions on Magnetics, Vol. 33, no. 1, pp. 990-995, Jan. 1997

[14] K. Miura, E. Yamamoto, and H. Muraoka,"Estimation of maximum track density in shingled writing," IEEE Transactions on Magnetics, Vol. 45, no. 10, pp. 3722-3725, Oct. 2009.

[15] R. W. Cross, M. Montemorra,"Drive based recording analyses at $>$ 800Gb/in^2 using shingled recording," PMRC 2010, Sendal, Japan paper 19pB1, 2010.

[16] D. Park, C. -I. Lin and D. H. C. Du,"H-SWD: A Novel Shingled Write Disk Scheme based on Hot and Cold Data Identification," 10th USENIX Conference on File and Storage Technologies (FAST12), Feb. 2012.

[17] N. Jeremic, G. M*ü*hl, A. Busse and J. Richling,"The Pitfalls of Deploying Solid-State Drive RAIDs," 4th Annual International Systems and Storage Conference (SYSTOR), May, 2011.

[18] K. Ozaki, Y. Okamoto, Y. Nakamura, H. Osava and H. Muraoka, "ITI canceller for reading shingled-recorded tracks," 9th Perpendicular Magnetic Recording Conference. 2011.

[19] Financial1.spc and Financial2.spc. http://traces.cs.umass.edu/index.php/Storage/Storage

[20] MSR Cambridge Traces. http://iotta.snia.org/traces/388