Emulating a Shingled Write Disk

Rekha Pitchumani[1,*,†], Yulai Xie[3,*], Andy Hospodor[1], Ahmed Amer[2], and Ethan L. Miller[1]

[1]Storage Systems Research Center, University of California, Santa Cruz, CA
[2]Santa Clara University, Santa Clara, CA
[3]Huazhong University of Science and Technology, China

1 Introduction and Motivation

Any further dramatic improvements to the capacity of hard disk drives is bound to come with some major changes to the currently employed techniques, as they are reaching their limitations imposed by the laws of physics. Of the new technologies being explored, Shingled writing promises an areal density increase of about 2.3x [1], with minimal changes to the manufacturing process.

Since Shingled Write Disk(SWD) layers the tracks on top of each other, a random write will destroy the data on adjacent tracks. This forces the SWD to be a largely sequential write device with unrestricted random reads. Although it is possible to treat a SWD like a virtual tape with random reads, exploring its potential to replace Hard Disk Drives in their traditional roles is more desirable. Recent research [1] [3] has explored the design issues in a shingled write disk system and proposed solutions ranging from data layout management to system software changes.

Further development and assessment of the proposed solutions are hindered by the unavailability of a SWD. This need for a Shingled Write Disk Drive is the motivation behind this work in progress. Our solution to this problem is to emulate a Shingled Write Disk Drive, by implementing a device driver to mimic the operations of a SWD on a regular Hard Disk Drive.

2 Emulating a Shingled Write Disk

Our implementation makes use of the Device Mapper infrastructure available in the Linux 2.6 kernel. We have implemented 'dm-shingle', a device mapper target module which lets one create a shingled disk block device on top of any existing block device. The resulting I/O stack is represented in the figure 1. Figure 2a illustrates the *read* and *write* operations in a SWD and figure 2b shows how this is captured in our emulation.

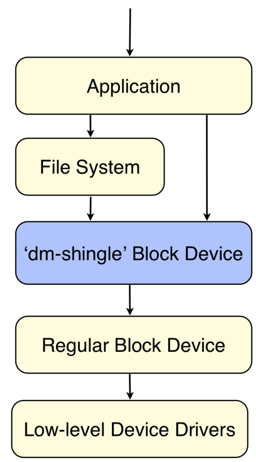
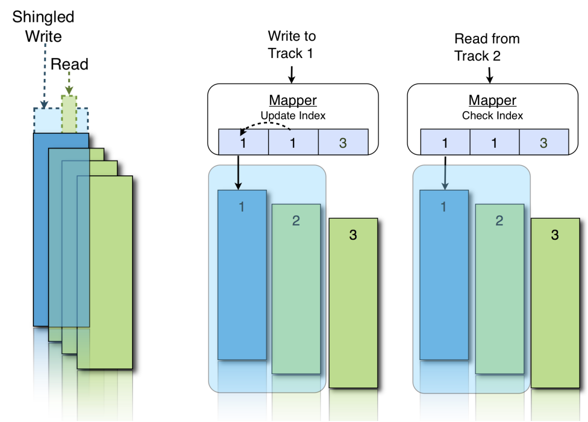


Figure 1: I/O Stack

A shingled write to a track overwrites k adjacent tracks. We maintain a mapping table to indicate whether a track has been overwritten by data from another track. On a write, we write to the current track and mark the sectors on k adjacent tracks as being overwritten with data from the current track. For example, in the figure a write to track 1 is written to track 1 and track 2 is marked as overwritten by track 1. Every read operation first checks the map table to determine to determine which track to read from.

We implement the mapping at the LBA level and hence our driver takes the following disk parameters into account: number of platters, sector layout mechanism, number of zones, number of tracks per zone, number of sectors per track per zone and the track skew. To keep our driver relatively simple, we chose to work with a single platter 160GB Seagate SATA drive. We used Disk Geometry Analyzer(DIG) [5] utility to determine the drive's parameters, and hard coded the geometry information in our driver.

*Student Author
†Presenter



(a) SWD operations

(b) SWD operations on a Regular HDD

Figure 2: Shingled Write Disk(SWD) Emulation

3 Why emulate?

An alternate approach to solve the problem at hand would be to build a simulator such as DiskSim [2] to simulate a Shingled Write Disk Drive. But the best that simulation could do is model the behavior of a couple of disks and cannot be as good as being able to test the solutions on a real disk.

We are the first to offer a solution that can be used to test the shingled disk management schemes on a real disk. Since the underlying medium and the read and write mechanisms of a shingled write disk will be very similar to existing hard disk drives, the read/write performance measurements on our emulated disk should also be very similar to a real SWD. An emulated disk will not report any synthetic numbers as it actually reads and writes data to an underlying real disk.

Modern hard disk drives have a complex physical geometry that is tailored to individual drives. Even the disk controllers *learn* the drive's layout during the final manufacturing process. Since most SWD layout management solutions, be it a simple *read-modify-write* scheme, or grouping of tracks into *bands*, or a complex block allocation policy, has to be aware of the underlying disk's geometry, customizing the driver with the parameters extracted from individual disks is a much better approach.

4 Current Status and Future Work

We are currently evaluating our driver, not just to ensure an error free driver, but also to verify and validate the captured physical disk geometry and the shingled disk operations. Once the evaluation is complete, we plan to make it available online to facilitate other research as well.

Borrowing from the NAND Flash device management, a Shingled Translation Layer(STL) approach [4] has been proposed. But the STL management calls for implementing dynamic bands and garbage collection/defragmentation in addition to performing a block translation in the SWD firmware for better performance. It is counter intuitive to let the host filesystem do block allocation while the firmware does all the above work. It makes more sense to let the firmware deal with block allocation too, which brings us to the Object based Storage Device(OSD) interface.

In this work in progress talk, we will talk about our experiences in emulating a Shingled Write Disk, the lessons we learned doing this work, and why we believe hiding the details of specific storage media from the host operating system behind an object interface is the right way to go.

References

[1] Amer, A., Long, D. D., Miller, E. L., Paris, J.-F., and Thomas Schwarz, S. Design issues for a shingled write disk system. In *Proceedings of the 26th IEEE Conference on Mass Storage Systems and Technologies* (May 2010).

[2] Busy, J. S., Schindler, J., Schlosser, S. W., Ganger, G., and Contributors. The disksim simulation environment version 4.0 reference manual. Tech. Rep. CMU-PDL-08-101, Carnegie Mellon University, May 2008.

[3] Cassuto, Y., Sanvido, M. A., Guyot, C., Hall, D. R., and Bandic, Z. Z. Indirection systems for shingled-recording disk drives. In *Proceedings of the 26th IEEE Conference on Mass Storage Systems and Technologies* (May 2010).

[4] Gibson, G., and Ganger, G. Principles of operation for shingled disk devices. Tech. Rep. CMU-PDL-11-107, Carnegie Mellon University, 2011.

[5] Gim, J., and Won, Y. Extract and Infer Quickly: Obtaining Sector Geometry of Modern Hard Disk Drives. *ACM Transactions on Storage 6*, 2 (2010).