The Peril and Promise of Shingled Disk Arrays

(how to avoid two disks being worse than one)

Quoc M. Le   JoAnne Holliday   Ahmed Amer
Department of Computer Engineering, Santa Clara University, Santa Clara, CA
{*qle, jholliday, aamer*}*@scu.edu*

Shingled Disks and Arrays – Disk drives have seen a dramatic increase in storage density over the last five decades, but to continue the six orders of magnitude growth seems difficult if not impossible. One promising approach to overcome the impending limit is shingled magnetic recording (SMR). It's particularly appealing as it can be adopted while utilizing essentially the same physical recording mechanisms. Current disks offer recording densities of $400Gb/in^2$, but with shingled writing $1Tb/in^2$ would be an achievable goal [7]. And yet, the logical behavior of such drives is just as important as their physical realization. Because of its manner of writing, a shingled write disk would be unable to update a written track without overwriting neighboring tracks, potentially requiring the rewrite of all the tracks to the end of a "band" (where the end of a band is an area left unwritten to allow for a non-overlapped final track). Both our prior work, as well as that of other researchers has explored techniques to alleviate the effects of this restriction, typically through some form of log-structuring of writes to defer the need to update data in-place [2, 3, 4]. Casutto *et al.* [3] offered one of the first practical solutions to managing a log-structured layout in the presence of limited metadata storage capacity, while Amer *et al.* [2] explored a spectrum of design parameters for shingled-write disks, including alternative interfaces such as object-based stores, or file system-based approaches to addressing the new disk behavior. We describe our recent experiences evaluating the behavior of shingled disks when used in an array configuration or when faced with heavily interleaved workloads from multiple sources.

Workload-Based Evaluation – Should shingled write disks be used in a server environment, it is likely that they would be organized as part of an array or subjected to workloads that involve writes originating from multiple sources. We evaluated the impact of data striping and workload interleaving on such disks. Whether an array of shingled disks is arranged as a simple spanning arrangement, or a striped arrangement (aimed at increasing effective bandwidth), our initial results suggest that such arrangements can dramatically increase the amount of data relocation and re-writing required to maintain a shingled write drive. We have also found that a workload that originates from a heavily interleaved mix of sources is also detrimental to shingled write disk performance. We have reached these preliminary conclusions through the replay of recorded workload traces. To evaluate the impact of shingled-writing when employed on disks arranged in array, we evaluated several recorded workloads and replayed them against a simulated drive to measure the number of track-to-track movements that would be incurred under different conditions.

In prior work we have evaluated the performance of a single Shingled Write Disk, SWD, against a variety of workloads. The workload types collected were Block I/O level traces [6] drawn from a variety of system types, block traces reconstructed from web server HTTP request logs [1], and new block-level traces which we collected from general file system usage over several months. From workload traces, we have also been able to generate workloads representative of specialized applications. For example, one of our traces was drawn from a filesystem being used to host the image files of a local VMWARE installation, while others were reconstructed web server workloads. These workloads were part of a larger collection we compiled from a pool of 35 different real-world workloads. These workloads varied greatly in the total number of operations observed, and in the mix of reads, writes and updates, and from them we extracted four disparate workloads as representatives for use in the current experiments.

Figure 1 shows the logical arrangement of blocks we evaluated, while Figure 2 shows a sample of the preliminary results we observed for the amount of inter-track movement resulting from a total of eight different configurations of block arrangement and workload interleaving. All the results in Figure 2 were based on a shingled write disk utilizing a log-structured write scheme to minimize the need to copy overlapped blocks when an in-band update was required. The **pure** workload shows the total amount of disk activity across four disks arranged in sequence, with workloads replayed sequentially and including no interleaving. In other words, four consecutive traces were each replayed in their entirety, and consecutively, against a disk array employing a spanning layout. This effectively simulated the behavior of a workload that varied over time, but which at no point included requests interleaved with others of a different workload. The **striped** workload combines four different workloads, and replays the composite workload against a striped organization of disk blocks across

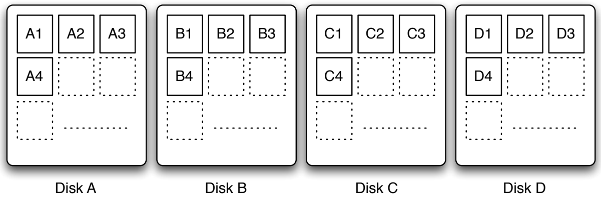

Figure 1. *Logical view of a simple array of disks. In the* **striped** *arrangement, blocks 0, 1, and 2 are arranged as A1, B1, and C1. In* **pure** *arrangements, blocks 0, 1, and 2 are arranged as A1, A2, and A3.*

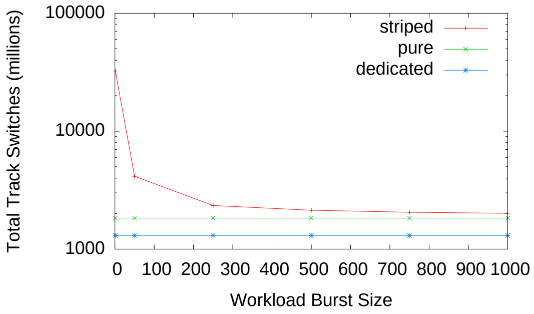

Figure 2. *Disk activity when replaying multi-source traces against a simulated array of shingled write disks.*

four disks. The workload was generated by randomly interleaving the operations from each of the four workloads in limited bursts. The x-axis of the figure represents each burst size, increasing from a minimum of one (where the interleaving is maximized) up to bursts of a thousand operations. Finally, the **dedicated** results represent the behavior of the shingled write disks when each disk is dedicated to an individual source workload.

Figure 2 shows that as the degree of interleaving in the composite workload traces is reduced (the burst sizes increase), for the array we see a reduction in the amount of disk activity that approaches that of the **pure** configuration. This is predictable and expected, as replaying a sequence of traces without any interleaving is exactly what is done by that configuration, and is the ultimate destination of extending burst sizes until they encompass an individual workload trace in its entirety. The surprising observations are just how much more activity results when unrelated operations are finely merged into a composite trace, and how further improvement can be achieved by separating workloads from different sources to individual dedicated disks. For a workload created from interleaving operations from multiple sources into small bursts, the amount of movement caused by relocating disk bands rises dramatically (up to forty times in this instance, though quickly dropping as the burst size increases to the level of 50 and 250 operations per burst). We attribute this behavior to the increased likelihood of unrelated data being written in adjacent positions that increases the likelihood of an update being required that is unrelated to much of the data on the same band. This problem is alleviated as the burst sizes increase, and eliminated entirely when individual dedicated disks are used. The difference in the **dedicated** configuration is that, unlike the **pure** configuration, it will never result in the writing of data from different data sources to the same device. Because the **dedicated** configuration avoids this risk entirely, we see a further drop in disk activity of around 25%.

Conclusions – We have offered our first experimental results evaluating the behavior of shingled write disks when used as part of an array, and the impact of increasingly interleaved workloads from different sources. While our initial results show a potentially dramatic negative impact when dealing with heavily interleaved workloads, they also demonstrate the positive effect of reducing such interleaving. This can be achieved either by rethinking a traditional array layout and dedicating disks and bands, or by directing independent workloads to different devices/bands. Directing different workloads to different devices can be aided by existing efforts on workload differentiation and tagging [5].

References

[1] "http://ita.ee.lbl.gov/html/contrib/nasa-http.html."

[2] A. Amer, D. D. E. Long, E. L. Miller, J.-F. Paris, and T. Schwarz, "Design issues for a shingled write disk system," in *Proceedings of IEEE MSST*, 2010.

[3] Y. Casutto, M. Sanvido, C. Guyot, D. Hall, and Z. Bandic, "Indirection systems for shingled-recording disk drives," in *Proceedings of IEEE MSST*, 2010.

[4] G. Gibson and M. Polte, "Directions for shingled-write and two-dimensional magnetic recording system architectures: Synergies with solid-state disks," tech. rep., Carnegie Mellon PDL, May 2009. CMU-PDL-09-014.

[5] M. Mesnier, F. Chen, T. Luo, and J. B. Akers, "Differentiated storage services," in *Proceedings of ACM SOSP*, 2011.

[6] D. Narayanan, A. Donnelly, and A. Rowstron, "Write off-loading: Practical power management for enterprise storage," in *Proceedings of FAST'08*, 2008.

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