# Shingled Magnetic Recording

## Areal Density Increase Requires New Data Management

TIM FELDMAN AND GARTH GIBSON

Tim Feldman works on drive design at the Seagate Technology Design Center in Longmont, Colorado. His current work focuses on object storage. He also spends time randonneuring, Nordic skiing, and logging.
timothy.r.feldman@seagate.com

Garth Gibson is Professor of Computer Science at Carnegie Mellon University and the co-founder and Chief Scientist at Panasas Inc. He has an MS and PhD from the University of California at Berkeley and a BMath from the University of Waterloo in Canada. Garth's research is centered on scalable storage systems, and he has had his hands in the creation of the RAID taxonomy, the SCSI command set for object storage (OSD), the PanFS scale-out parallel file system, the IETF NFS v4.1 parallel NFS extensions, and the USENIX Conference on File and Storage Technologies.
garth@cs.cmu.edu

Shingled Magnetic Recording (SMR) is the next technology being deployed to increase areal density in hard disk drives (HDDs). The technology will provide the capacity growth spurt for the teens of the 21st century. SMR drives get that increased density by writing overlapping sectors, which means sectors cannot be written randomly without destroying the data in adjacent sectors. SMR drives can either maintain the current model for HDDs by performing data retention behind the scenes, or expose the underlying sector layout, so that file system developers can develop SMR-aware file systems.

The hard disk drive industry has followed its own version of Moore's Law, known as Kryder's Law [1], for decades. While gate density has increased for integrated circuits, bit density has increased at a similar compound annual growth rate of about 40% through the application of a sequence of technologies from inductive to magneto-resistive to perpendicular recording. Technologies that are still in development include Heat-Assisted Magnetic Recording and bit-patterned media, each of which has its own innovative method of packing bits even more closely together. Preceding those technologies, however, the industry is faced with the challenge of increasing areal density of perpendicular recording.

Conventional recording, shown schematically in Figure 1, uses a track pitch that is sized to match the writer gap width such that tracks do not overlap, and the reader gap width is sized such that the signal from only one track is read. Conventional recording has scaled by decreasing both the reader and writer gap sizes, which allows bits to be packed more densely in the down track direction as well as the track pitch in the cross track direction. Further decrease of the writer gap size is extremely difficult. Small write gaps do not produce enough flux density to record the magnetic domains effectively on the disk surface. But reader gap widths can continue to be scaled to narrower dimensions.

SMR, shown schematically in Figure 2 with less than one track of overlap, enables higher areal density by recording at a track pitch appropriate for the as-narrow-as-possible reader. Recording a sector at this track pitch with an as-wide-as-necessary writer means that neighboring sectors are affected. SMR records in a strict sequence and with overlap in only one direction, leaving previously recorded data in the other direction in a readable state. This overlapping is like the placement of shingles on a roof, hence the name Shingled Magnetic Recording.

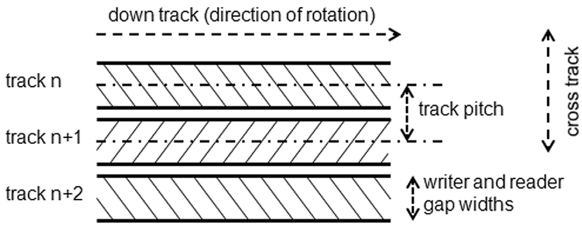

down track (direction of rotation)
track n
track n+1
track n+2
track pitch
cross track
writer and reader gap widths

**Figure 1:** Schematic of conventional magnetic recording

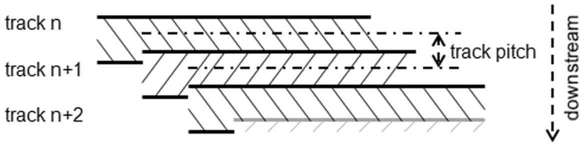


**Figure 2**: Schematic of Shingled Magnetic Recording

## SMR Data Management Challenge

Historically, magnetic recording has used isolated areas of media for each sector such that sectors can be updated without overwriting neighboring sectors. Down track each sector is spaced sufficiently to accommodate spin speed fluctuation, and cross track they are spaced so that writes do not affect neighboring tracks. This is a match with the random block access model of the interface to disk drives. SMR breaks this model of independently writable sectors.

SMR mandates that tracks be written in the shingled direction. Sectors are still the atomic unit of media access, but SMR requires that the overlapped sectors on downstream tracks that get overwritten do not contain data of interest to the system. Either drive firmware, host software, or a combination of the two must take on the data management challenge of dealing with the data in the overlapped sectors.

Data management for SMR poses an emerging challenge for storage systems and drive design. This article covers the challenge of data placement in disk drive design, the range of solutions, and some of their issues. There are two major solution spaces. Drive-managed SMR retains the current random block write model where the most recently written data for every logical sector is retained regardless of accesses to any other sector. This is referred to as data retention in this article. Host-managed SMR, in contrast, shifts data retention responsibility to the host. This article further introduces a third SMR data management type that attempts to blend some drive- and host-managed characteristics, an approach we call cooperatively managed SMR.

### *Contribute to the Discussion*

Host and cooperatively managed SMR are still in definition. This article serves as a notice to the systems community on the various ways SMR may impact storage design.

The industry will be defining standards for interfaces to SMR disk drives in the traditional committees: T10—SCSI Storage Interfaces and T13—ATA Storage Interface of the International Committee for Information Technology Standards (INCITS). A T10 SMR Study Group exists as a forum for discussion.

## The Disk Physical Layout Model

Hard disk drive media is organized as a set of surfaces, each having at least one read/write head and each consisting of a set of tracks. The tracks are organized in concentric circles. Each track is a set of non-overlapping sectors. The sector constitutes the atomic unit of access; partial sector reads and writes are not supported.

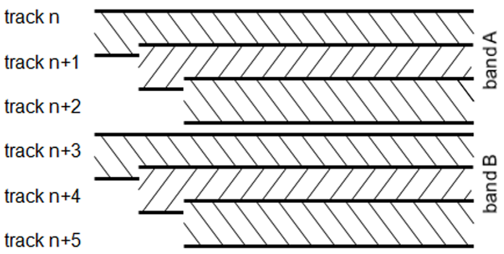


**Figure 3:** Schematic of Shingled Magnetic Recording with two 3-track bands

The sectors of a track are accessed consecutively as the disk rotates with respect to the head. One sector on each track is designated as being logically the first sector of the track, with subsequent sectors in turn being logically the next.

Often SMR is organized as sets of tracks that overlap each other; these are physically isolated from other sets of tracks by a gap so that there is no overlap between sets. Such a set of tracks is often called a “band.” We will use this nomenclature in this article. Figure 3 shows this schematically.

Within a band the shingling happens in a single direction. Thus, the tracks of a band are overlapped much like the overlapping shingles on a roof.

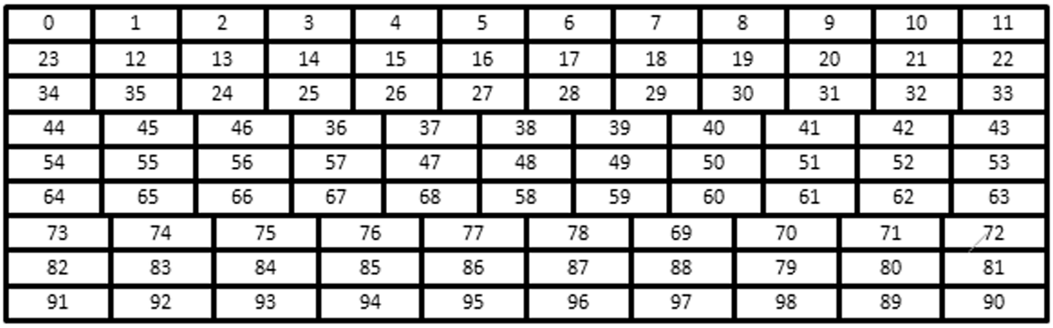
## Logical to Physical Mapping

Modern block command sets, notably ATA and SCSI command sets used by SATA and SAS, use a linear sector address space in which each addressable sector has an address called a logical block address, or LBA. This obfuscates the physical, three-dimensional characteristics of the drive: number of surfaces, tracks per surface, and sectors per track. It allows drives to manage defects without perturbing the host using the drive. Decoupling of logical and physical mapping has allowed drives to evolve without being synchronized to changes in host software.

A particular expectation needs to be acknowledged: LBA x and LBA x+1 are related in such a way that if LBA x is accessed, then accessing LBA x+1 is very fast. This is not an absolute requirement, and is not true 100% of the time, but it is generally the case for conventional drives.

### *Static Mapping*

The conventional approach to mapping LBAs to physical sectors is to map the lowest LBA to the first sector on the outermost track and follows the sector progression—leaving known defective sectors unused in the mapping—and then follows the track



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 23 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 34 | 35 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 44 | 45 | 46 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | |
| 54 | 55 | 56 | 57 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | |
| 64 | 65 | 66 | 67 | 68 | 58 | 59 | 60 | 61 | 62 | 63 | |
| 73 | 74 | 75 | 76 | 77 | 78 | 69 | 70 | 71 | 72 | | |
| 82 | 83 | 84 | 85 | 86 | 87 | 88 | 79 | 80 | 81 | | |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 89 | 90 | | |

**Figure 4:** An example of a static mapping layout with tracks shown as rows of sectors labeled with their LBA

progression to map all of the rest of the LBAs. A rotational offset from the last sector on one track to the first sector of the next track is called "track skew" and allows a seek to complete in the rotational time so as to optimize sequential throughput. This mapping does not change dynamically, say in response to a new write command. There are rare exceptions to the static nature of this mapping in conventional disk drives such as when a grown defect is discovered and the LBAs for the affected sectors are remapped to spare sectors that are part of a small over-provisioning of the media for the purposes of defect management.

Figure 4 shows an example of static mapping for a drive with three tracks of 12, three tracks of 11, and three tracks of 10 sectors per track. In this example the track skew is one sector. For simplicity, this example is a single surface and has no skipped defects.

Static mapping on an SMR drive has some critical implications. The first is that an arbitrary LBA cannot be updated without affecting the data retention of the LBAs assigned to sectors overlapped by the sector to which the LBA to be updated is mapped. Accommodating this means making either a significant change in the disk's data retention model, because writing one sector modifies the data in one or more other sectors, or a significant change to write performance, because firmware must pre-read all the collaterally impacted sectors and rewrite them in downstream order. Caches and other techniques can be used to moderate either or both of these effects.

Note that with static mapping, each LBA has a couple of key characteristics determined by the set of LBAs that it overlaps. One characteristic is the distance from the written LBA to the largest overlapped LBA. We refer to this as the Isolation Distance as it describes the minimum number of unused LBAs that will isolate the written LBA from all LBAs further away. The magnitude of this distance depends on the downtrack overlap of the write, number of sectors per track, track skew, and skipped defects. Another characteristic is that for each written LBA there is an extent of contiguous LBAs that it does not overlap, ending at the largest higher LBA that the LBA does overlap. We refer to the size of this extent as the No Overlap Range as it describes a range within which writes do not affect other LBAs. The size again depends on the number of sectors per track, track skew, and skipped defects. These distances can be used by the data management scheme as is described later in the section on Caveat Scriptor.

Figure 5 repeats the layout example of Figure 4, with a writer overlap of two neighboring tracks and with LBAs increasing in the downstream direction. This means, for example, that LBA 0 overlaps LBAs 23 and 34; thus, its Isolation Distance is 34. The extent following LBA 0 that is not overlapped extends to LBA 11; thus, its No Overlap Range is 12. In contrast, LBA 68 overlaps LBAs 76, 77, 85, and 86 for a Isolation Distance of 18. The extent following LBA 68 that is not overlapped goes through LBA 75 for a No Overlap Range of 8.

Figure 5 shows that for static mapping, maintaining the data in every LBA requires all downstream LBAs to be read and then rewritten. For instance, a write to LBA 68 not only requires LBAs 76, 77, 85, and 86 to be read and then rewritten, but also LBAs 94 and 95 because writes to LBAs 76 and 85 overlap LBA 94, and writes to LBAs 77 and 86 overlap LBA 95. A simpler data retention algorithm is to read and then rewrite all higher LBAs to the end of the band; thus, a random write may, on average, cause half of its band to be read and rewritten. Alternatively, if data retention is not required, then LBAs can be updated in place. A simple model is that writing an LBA can cause loss of data retention in all higher LBAs to the end of the band.

### *Dynamic Mapping*

An alternative to static mapping is to allow LBAs to be mapped to physical sectors dynamically by drive firmware. This is analogous to the Flash Translation Layer (FTL) model for solid state drives (SSD).

Specifically, an SMR drive can employ dynamic mapping in which it maintains a logical to physical map, sometimes called a

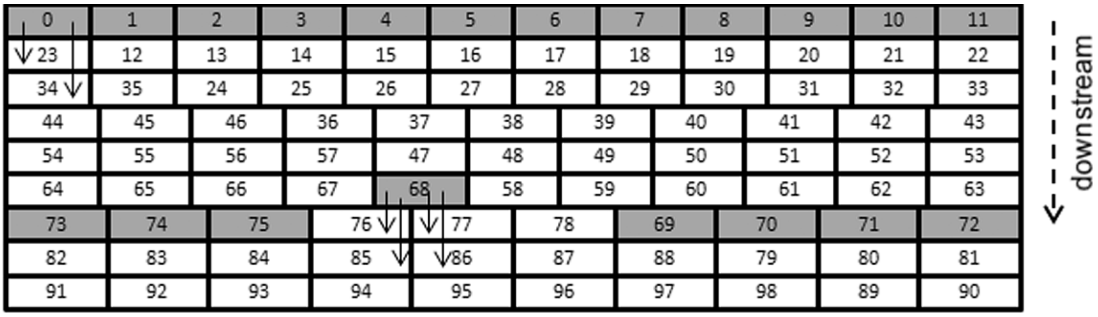


**Figure 5:** The static mapping layout example with shading indicating selected no overlap ranges and arrows indicating selected overlaps for a two-track overlap width

forward map in SSD, and assign the LBAs for write commands based on the drive's internal mapping policies. The forward map must then be referenced to satisfy read requests to determine which sectors are currently assigned to the requested LBAs.

Any and all of the techniques in an SSD FTL can be leveraged on an SMR drive with dynamic mapping. This includes policies for write performance so that new data can be placed in sectors that do not overlap data that must be retained, policies for read performance so that a minimum number of media accesses are required, and garbage collection so that data requiring retention can be relocated before media is reused.

## Data Management for SMR

Handling data accesses, their performance, power, data retention impact, and restrictions on access patterns collectively are the data management done by a drive. This section covers the solution space for SMR.

### *Choices in SMR Data Management*

SMR data management makes specific choices in data retention, restrictions on data accesses, the physical sector layout, and the logical to physical mapping. Specific examples of data management choices are described later in the sections on drive- and host-managed SMR.

Conventional drives deliver complete data retention for all LBAs at all times to a specified error rate, such as 1 nonrecoverable read error per $10^{15}$ bits read. SMR drives can deliver the same data retention model, or explicitly embrace a different model in which LBAs may not have data retention depending on the sequence of writes to other LBAs.

Conventional drives allow an LBA to be accessed at any time, either read or write accesses. SMR drives can deliver the same data access model, or explicitly embrace a different model in which only specific LBAs may be written and specific LBAs may be read depending on the state of the drive. The pertinent state is expected to be dependent on the sequence of writes and, possibly, temporal separation between the writes.

SMR data management often makes use of many mutually isolated bands of tracks. The bands may be constant in the number of tracks, or might be constant in the number of sectors. The specifics of where band boundaries are in the physical sector layout are a choice of SMR data management.

SMR data management has choices of what logical to physical mapping to employ. Static or dynamic mapping can be used. Dynamic mapping has a wide range of choices that include examples from Flash Translation Layers and other innovations [2].

### *Drive-Managed SMR*

In drive-managed SMR, the drive autonomously delivers the conventional data retention model of maintaining the data of every LBA without any restrictions on the host access patterns. No changes are needed to the interface for drive-managed SMR. Choices of layout and mapping do not need explicitly to be exposed externally, but the choices do impact the performance and power profiles. Drive-managed SMR is responsible for garbage collection if the mapping choice can leave unmapped data in physical sectors. Drive-managed SMR is likely to be stateful in that the performance may be dependent on the state of the drive as determined by the usage history. Provisioning of additional memory and storage resources typically provides a choice of better performance at the cost of the expense of those resources.

Drive-managed SMR is a data management approach that can most directly leverage the technologies developed for SSD and FTLs. This includes over-provisioning of media. For instance, sometimes an SSD is populated with N gibibytes of Flash media but delivers N billion bytes of usable host capacity, in which case the SSD has the approximately 7% difference between $2^{30}$ and $10^9$ as over-provisioning. Similar over-provisioning is possible in an SMR drive.

Drive-managed SMR allows an SMR drive to be used in any existing storage stack, albeit with a different performance and power profile compared to conventional drives.

### *Host-Managed SMR*

The term "host-managed SMR" is defined to mean an SMR drive that does not autonomously deliver data retention for every LBA or restricts host accesses. Changes to the interface may be needed for host-managed SMR.

Strictly Append is a type of host-managed SMR that restricts host writes to occur only at the append point of a band. The append point is the ending position of the last write to the band; that is, an appending write implicitly moves the append point. Strictly Append also restricts reads to occur only before the append point of a band. That is, only written LBAs can be read. In its simplest implementation, Strictly Append presents a single band, and the drive may be written once in strictly sequential LBA order. ShingledFS [3] is a file system for Hadoop that uses this model. More complexity and versatility can be added by supporting multiple bands and thus multiple append points, and by allowing reuse of a band after an explicit event that moves the append point.

Exposed SMR is a different type of host-managed SMR where the host is aware of the layout and mapping. By specification or query through the interface, the host knows the details of the location of bands. Static mapping is the obvious choice such that each band is a consecutive set of LBAs. With this information, a host can know that writing an LBA obviates the data retention of all subsequent LBAs to the end of the band. Exposed SMR does not restrict host accesses, but instead moves the ownership of the data retention model to the host. This has further impact on defect management and other reliability constraints that are beyond the scope of this article. A specific Exposed SMR proposal, Caveat Scriptor, is described in a later section.

The logical to physical mapping for host-managed SMR does not have to be static; however, within a band, LBAs must be mapped to sectors that do not overlap sectors mapped to lower LBAs. This blurs the distinction between logical blocks and physical sectors. Nonetheless, the LBA is retained as the address semantic, which, for instance, allows dynamic mapping of defects.

Host-managed SMR can include a small fraction of unshingled space, some unshingled bands, for random writes.

### *Cooperatively Managed SMR*

Cooperatively managed SMR is a type of SMR data management that is not purely drive or host managed, but has characteristics of each. For instance, bands may have append points but perhaps not require all writes to be at the append point. Band locations may be exposed to the host and explicit methods may need to be invoked to move the append point. A specific cooperatively managed SMR proposal, Coop, is described in a later section.

## Alignment of Drive-Managed SMR to Applications

Drive-managed SMR delivers drives that have performance profiles that are notably different from conventional drives. Write performance is commonly sensitive to the availability of safe-to-write sectors, which in turn can be a function of the number and location of stale sectors. A drive that does internal garbage collection may sometimes be ready to accept a burst of new writes, or may have to proceed in its garbage collection to service new writes. This is the scope of a file system problem brought into the domain of the disk drive.

Read performance is sensitive to data layout. If dynamic mapping is part of the drive-managed SMR policies, LBA x and LBA x+1 can frequently not be proximate to each other, causing the read of a single LBA extent to require multiple disk media accesses. This read fragmentation issue is the typical file fragmentation problem brought into the domain of the disk drive. Drive-managed SMR includes the memory and storage resources and the embedded computing costs for over-provisioning and the FTL-like firmware.

Despite the performance differences with respect to conventional drives, drive-managed SMR is well aligned to many applications. Not only can it be deployed without modifications to the host, it is also a good match to the requirements in a lot of markets. This section describes the alignment of selected use cases to drive-managed SMR.

### *Personal External Drives, Backup Storage and Archive*

External drives for personal use and backup or archival storage are suitable applications for drive-managed SMR. The ingress of data is very bursty and sequential enough that the drive can handle writes efficiently. Long idle times between writes and reads allow the drive to defragment appropriately and prepare for subsequent write workloads. Low duty cycle and low performance requirements help the introduction of new technology, too. A paper on deduplication of desktop VMs [4] discovered that as much as 85% of desktop data collected from Microsoft developers disk traces is write-once.

### *Log-Structured Files Systems and Copy-on-Write*

With log-structure file systems (LFS) and copy-on-write (COW) policies in databases, file systems and other applications create a drive workload that is purposefully dominated by sequential writing. Drive-managed SMR can be optimized to handle a sequential write stream efficiently, making these applications a good match.

### *Applications with Writes Primarily Small or Spatially Dense*

Natural workloads always have some spatial locality. Sufficient spatial locality makes a limited amount of over-provisioning useful for drive-managed SMR just as it does for SSD. Many workloads are dominated by relatively small random writes of 4 KiB or so. Databases, online transaction processing, and many other applications commonly exhibit these traits. Such workloads are a good match for FTL-style technologies, and in fact can lead to drive-managed SMR performance that is superior to conventional drives—if the writes are small enough and/or the spatial density is high enough.

### *Applications Dominated by Reads*

Drive-managed SMR that bounds the read fragmentation can have read performance that is at or near parity with conventional drives. Applications such as content distribution, Web servers, and reference material hosting such as wikis are dominated by reads. These applications are a good match for drive-managed SMR.

### *Legacy and Installed Base*

The most important quality of drive-managed SMR is that it conforms to the same protocol and data retention model as conventional drives, albeit with a different performance profile. Drive-managed SMR allows the areal density increase of SMR to be employed in a legacy application and serves the entire installed base of disk-based storage.

## Alignment of Host and Cooperatively Managed SMR to Applications

Acknowledging that drive-managed SMR has different performance means that some applications, if unmodified for SMR, will have performance sensitivities for which drive-managed SMR is not always an optimal match. This is the main motivation for considering host and cooperatively managed SMR and its attendant impact to host implementations.

### *Sequential Write Workloads*

While drive-managed SMR can be optimized for sequential writes, it does not always deliver conventional drive performance. In particular, if a sequential write does not go all the way from LBA 0 to LBA max, and in natural workloads sequential writes never span the whole capacity of the drive, there is a start and end to each sequential write. When the start and end do not align with band boundaries for the logical to physical mapping of the drive, there is work required in the drive to “mend” the data at the edges of the write. Host and cooperatively managed SMR provide the context in which sequential writes can be restricted to start and end at band boundaries. These schemes additionally deliver read performance with fragmentation only at band boundaries, which closely approximates conventional read performance.

### *Log-Structured Files Systems and Copy-on-Write*

While the LFS and COW are generally a good match for drive-managed SMR, they eventually have a garbage collection requirement so that space can be reused. Garbage collection on an undifferentiated LBA space is likely to produce the same sort of performance challenges just described for sequential write workloads in general. Host and cooperatively managed SMR are an opportunity for garbage collection that is optimized for SMR.

### *High Performance Storage*

Lastly, given the opportunity to purpose-build a storage system for SMR, host and cooperatively managed SMR enable the system to be optimized for performance. Such systems may further optimize the over-provisioning and other attributes that contribute to cost, power, and reliability.

## Caveat Scriptor: An Exposed SMR Proposal

Caveat Scriptor is Latin for “let the writer beware” and is used here as a name for a more specific proposal for Exposed SMR. The layout model for Caveat Scriptor is static mapping with critical drive parameters exposed.

### *Drive Parameters*

As described in the section on static mapping, above, each LBA has two notable parameters:  No Overlap Range and Isolation Distance.

Remember that No Overlap Range is the minimum distance of contiguous, non-overlapping LBAs that follow each written LBA, and Isolation Distance is the maximum LBA distance in which some LBA might be overlapped. An Exposed SMR drive could simply make these parameters available to the host for every LBA. A Caveat Scriptor drive instead exposes a single No Overlap Range and Isolation Distance value that apply to every LBA. It determines the possible drive parameters as follows:

- Drive No Overlap Range <= minimum (No Overlap Range for all LBAs)
- Drive Isolation Distance >= maximum (Isolation Distance for all LBAs)

For a given model of Caveat Scriptor drives, all will have the same DNOR and DID values. That is, Caveat Scriptor selects a Drive No Overlap Range (DNOR) to be small enough for all drives of the model, and a Drive Isolation Distance (DID) to be large enough for all drives of its model. This allows software to be specialized to a model and not to individual drives.

For example, for a model of drives in which all layouts are described by Figure 5, the minimum No Overlap Range is at LBA 68 where the following no overlap extent goes through LBA 75, so DNOR is 8, and the maximum Isolation Distance is at LBA 0 as described previously, so DID is 34.

### *Host Band Construction*

With the DNOR and DID parameters, the determination of band boundaries is left up to the host. Leaving at least DID LBAs unused between bands is sufficient to provide isolation. In the example of Figure 3, 34 LBAs is the amount of unused space required to isolate bands; LBAs 0 to 29 could constitute a 30-sector band, LBAs 64 to 98 a second 35-track band, with LBAs 30 to 63 as the 34 unused sectors that isolate the two.

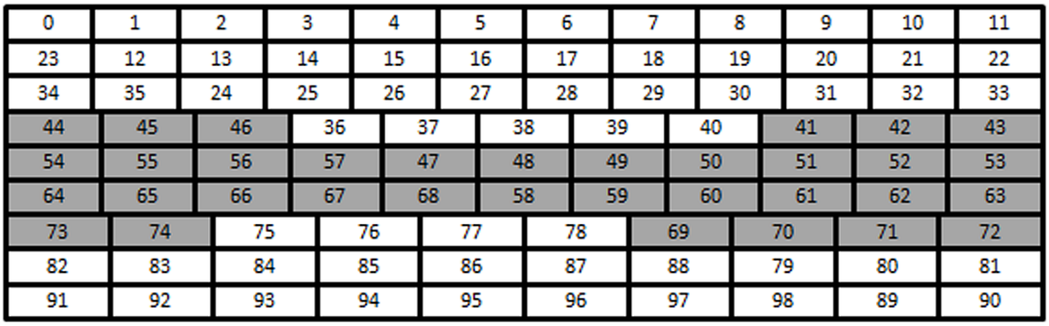
Three specific uses cases are described:

1. Random write band: Making a band no bigger than DNOR LBAs creates a random write band if the range is sufficiently isolated by DID LBAs on both ends. A band of this size has the attribute that no LBA in the band is overlapped by any in-use LBA—that is, LBAs that are used as part of the band isolation. Such a band is one whose LBAs can be randomly written without obviating the data retention of any in-use LBA. In the example of Figure 3, 8 LBAs is the maximum random write band size; LBAs 50 to 57, inclusive, can be a random write band. Note that DID will typically be much larger than DNOR, so random write bands are inefficient in their ratio of in-use to not-in-use LBAs.
2. Sequential write band: A band of any size that is sufficiently isolated by DID LBAs can be used as a sequential write band in which data is retained for LBAs that precede the most recent write. Such a band has no LBAs that are overlapped by LBAs in a different band, and no LBAs overlap any LBA in a different band.
3. Circular buffer band: A band can be managed as a circular buffer if a sufficient distance is maintained between the end and the start. The minimum required distance is DID. Thus the effective size of a circular buffer is less than its band by at least DID. A circular buffer could be used, for instance, to have intra-band garbage collection in which non-stale data is shuttled from the start to the end. In this instance, when stale data is present at the sta rt of the buffer the start position can traverse forward without a concomitant copying of data to the end, thus increasing the distance from the end to the start and allowing new data to be added to the buffer.
4. In the example of Figure 3, if all 99 sectors are used as a single circular buffer band and the end of the band is, say, at LBA 40, then the start must not be in the LBA range 41 to 74, inclusive. Figure 6 shows this state. Before data can be added at LBA 41, LBA 75 must become unused or stale to comply with the spacing requirement of DID = 34.

### *Value Proposition*

The Caveat Scriptor Exposed SMR proposal delivers the following value propositions.

- Performant: Fast, static mapping can be used with all accesses going straight to media.
- Predictable: There is a low probability of internal drive management operations causing response times that the host does not expect.
- Versatile: Circular buffers can be deployed as well as random and sequential bands.
- Efficient: Isolation occurs only where the host needs LBA extents to be isolated.
- Flexible: Hosts can construct bands of any size.
- Host-owned data retention: The data retention of logical blocks is determined by the host, matching the usage model of the storage stack.



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 23 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 34 | 35 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 |
| 44 | 45 | 46 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | |
| 54 | 55 | 56 | 57 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | |
| 64 | 65 | 66 | 67 | 68 | 58 | 59 | 60 | 61 | 62 | 63 | |
| 73 | 74 | 75 | 76 | 77 | 78 | 69 | 70 | 71 | 72 | | |
| 82 | 83 | 84 | 85 | 86 | 87 | 88 | 79 | 80 | 81 | | |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 89 | 90 | | |

**Figure 6:** The static mapping layout example deployed as a circular buffer with its start at LBA 40 and its end at LBA 95. The shading shows 34 LBAs that are unused between the start and end of the circular buffer.

## Coop: A Cooperatively Managed SMR Proposal

Coop is a specific proposal for  cooperatively managed SMR. It blends some characteristics of drive-managed SMR with some of host-managed SMR. Coop has the data retention model of drive-managed SMR and the performance characteristics of host-managed SMR when the host conforms to a constrained band reuse model.

Coop is targeted to applications that are dominated by sequential writes through large LBA extents, optionally with a small set of randomly written extents. Coop additionally targets applications where there may be infrequent exceptions to the sequential write behavior even outside the randomly written extents.

### *Band Life Cycle*

Coop bands go through a cycle of state transitions from empty to filling to full and back to empty. The full to empty state transition occurs due to an explicit host command such as Trim that unmaps the entire LBA extent of a band. This command can also be issued to a band in the filling state, moving it to the empty state.

### *Well-Known Bands and High Water Marks*

Coop is built on a layout model of same-sized bands and regularly placed band boundaries. Band boundaries are at strict integer multiples of the band size. Each band has a High Water Mark that represents the highest address written since the most recent empty state. The High Water Mark is the optimum write location, but is not an enforced append point.

It is proposed that the band size is standardized to either 256 MiB or 1 GiB. These power-of-two sizes are sufficiently large to allow for a minimum of space to be devoted to band isolation.

### *Host Policies*

The host write behavior on a Coop drive should be dominated by writes at the High Water Mark of the respective band. Writes at the High Water Mark can be serviced by conventional policies since higher LBAs are “trimmed” and do not require data retention. Writes not at the High Water Mark, at either lower or higher LBAs, are allowed and impact the drive policies as described in the next subsection.

Host read behavior is not restricted. Hosts may read trimmed LBAs.

Before reusing a band, it is incumbent on the host to issue the appropriate command to unmap the whole band. Before issuing this command the host must first copy any non-stale data to some other band. Garbage collection in a Coop drive is the responsibility of the host.

### *Drive Policies*

Writes not at the High Water Mark may need to be serviced with drive-managed-style data management techniques. Note that writes not at the High Water Mark but within the No Overlap Range can potentially be optimized with policies that are similar to conventional data management.

Support for a small set of randomly written extents is also provided through drive-managed-style data management, possibly with an appropriate amount of over-provisioning. The amount of over-provisioning is likely to determine the amount of randomly written space that can be handled with higher performance.

Reads comply with the full data retention model of Coop. Reads of mapped sectors return the most recently written data. Reads of unmapped sectors return the appropriate unmapped-sector data pattern, possibly all zeros. For bands that have been written in strict sequential order, reads of LBAs below the High Water Mark of the respective band return the most recently written data, and reads above the High Water Mark return the appropriate unmapped-sector pattern.

### *Value Proposition*

The Coop proposal for cooperatively managed SMR delivers the following value propositions:

- Performant: Fast, static mapping can be used for bands that are sequentially written with sequential writes and all reads below the High Water Mark serviced directly from media. Drive performance for sequential writes at the respective High Water Mark will be like that of a conventional drive.
- Tolerant: Not all random writes have to be eliminated, just minimized. Software can be deployed without 100% removal of random writes.
- Versatile: The targeted applications represent a diverse set of common use cases.
- Efficient: The amount of over-provisioning can be bounded by the amount of randomly written space and the frequency of writes that are not at a High Water Mark.
- Low barriers to adoption: The conventional data retention model and standard commands allow straightforward adoption.
- Flexible: Random write extent locations can be anywhere in LBA space and can be non-stationary.
- Standardized: Current standard command sets continue to be used, albeit likely with a few additional queries for discovery of parameters and High Water Mark values.

## Further Work

### *Areal Density Gains*

Shingled Magnetic Recording offers the opportunity for disk drives to continue to deliver increasing areal density. The recording subsystem and the head, disk, and channel designs need to evolve to take maximum advantage of SMR.

Harvesting the areal density requires more than recording subsystem work. Storage systems need to prepare file systems, application software, and utilities to be well suited to SMR data management at the drive.

### *Call to Action*

Caveat Scriptor and Coop are two proposals for SMR interfaces. These proposals and others will be discussed at the T10 SMR Study Group, the open forum where changes to the SCSI standard are being discussed. Now is the time to add your voice to help move the technology in the best possible direction.

### *References*

[1] C. Walter, "Kryder's Law," Scientific American, July 25, 2005: http://www.scientificamerican.com/article.cfm?id=kryders-law.

[2] Tim Feldman, Seagate, US Patent Application 20070174582.

[3] Anand Suresh, Garth Gibson, Greg Ganger, "Shingled Magnetic Recording for Big Data Applications," Carnegie Mellon University, Parallel Data Laboratory, May 2012: http://www.pdl.cmu.edu/PDL-FTP/FS/CMU-PDL-12-105.pdf.

[4] Dutch T. Meyer, William J. Bolosky, "A Study of Practical Deduplication": http://www.usenix.org/event/fast11/tech/full_papers/Meyer.pdf.