Flex Dynamic Recording

TIMOTHY FELDMAN

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

Hard disk drives are capable of various recording methods without changing the hardware. For example, shingled magnetic recording (SMR) is a technique in which higher track density is obtained, but with a tradeoff of requiring sequential writing through a band of tracks. I introduce a new way of letting the storage manager dynamically, in the field, specify different recording methods for different parts of the media in the device. This article prepares readers for this upcoming disk technology by describing the opportunities to lower the total cost of ownership (TCO) and exploring the new device interface that needs to be defined.

In February 2016, Google's "Disks for Data Centers" white paper [1] proposed options to improve the total cost of ownership, speed, tail latency, and capacity of hard disk drives. One particular capacity improvement was for an implementation in which conventional and shingled recording are mixed in a single, hybrid hard disk drive. The white paper noted that if the outer tracks are conventional magnetic recording (CMR) and the inner tracks are SMR as shown in Figure 1, the outer tracks could be used for short-lived data enjoying the random write performance of CMR, while the inner tracks could hold long-lived data using higher density SMR.

More recently, Google has presented an initial set of requirements [2], and both Seagate and Western Digital have signaled their support in blog posts [3, 4]. In this article, I will refer to the ability to dynamically mix recording methods in a single disk as Flex, Seagate's name for the technology, alternately called Hybrid SMR and Realms by Google and Western Digital, respectively.

Problems and Opportunities

The range of tracks accessed by a workload is known as the *stroke*, referring to the range of motion of the heads. An application that accesses the full logical block address (LBA) space uses 100% of the stroke. If a disk is partitioned into 10 volumes, then each volume uses about 10% of the stroke—more accurately, 6.7% for the partition at the lowest LBAs, 10% in the middle, and 13.3% at the highest LBAs due to the variation in the number of sectors per track, with outer tracks at about twice the capacity of inner tracks. Note that this correspondence of logical addresses to physical radius assumes the conventional logical-to-physical disk mapping in which lower LBAs are on the outer tracks. With this partitioning, if the disk workload is restricted to a single partition, then the workload uses about 10% of the stroke. And since access time is highly sensitive to seek distance, this constrained workload is much faster than a 100% stroke workload.

In practice, accesses to hot data can be sped up by constraining it to a limited range of LBAs, a technique generally known as *short stroking*. This not only increases the I/Os per second (IOPS), but greatly increases the performance density (IOPS per TB) since the denominator of that term gets smaller. Figure 2 shows the relationship between performance density and stroke based on a first-order model of disk performance. Note that performance can increase by 4x just by short-stroking to 33%.

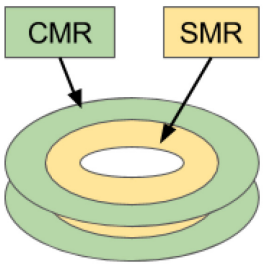

Figure 1: Depiction of a two-platter hybrid hard disk drive with CMR at the outer tracks and SMR at the inner tracks

But if short stroking is the only technique applied, and only 33% of the disk is used, there is unused media in the other 67% of the stroke. Finding a way to use this media while retaining the performance of the hot data is a TCO improvement opportunity.

Cold data, in contrast to hot data, can generally be written sequentially. Using SMR for cold data lowers the cost per byte. But after filling a disk with only cold data, the disk actuator arm will mostly sit idle. Finding a way to keep the disk mechanics busy serving useful I/Os is another TCO improvement opportunity. If a deployment has both hot and cold data, then a solution of segregated tiers not only leaves both TCO improvement opportunities unrealized, but also doubles the logistical complexity of managing two tiers and their unique drive types.

Flex, the ability to dynamically mix recording methods, allows the operating system to configure a single drive to a mix of CMR and SMR. And the mix can change to match a changing mix of hot and cold data. This means that hot data can enjoy the performance benefits of short stroking while cold data makes use of the rest of the media. The disk is then fully subscribed; all of its media and all of its mechanical capability are utilized, and the total cost is minimized.

Flex is not limited to just mixing CMR and SMR. There are other ways to improve TCO, speed, tail latency, and capacity. An idea as simple as using Flash in SLC or MLC mode provides one set of tradeoffs. Heat- or microwave-assisted magnetic recording may be able to record in different track widths by modulating the laser or microwave power and mixing track widths in an interlaced manner, as depicted in Figure 3, which increases the data density and, thus, disk capacity [5].

Interlaced magnetic recording (IMR) does not actually use different physical layers. Instead, "bottom" tracks are simply the wider tracks and "top" tracks are the narrower tracks. Since writing a bottom track can make two top tracks unreadable, IMR

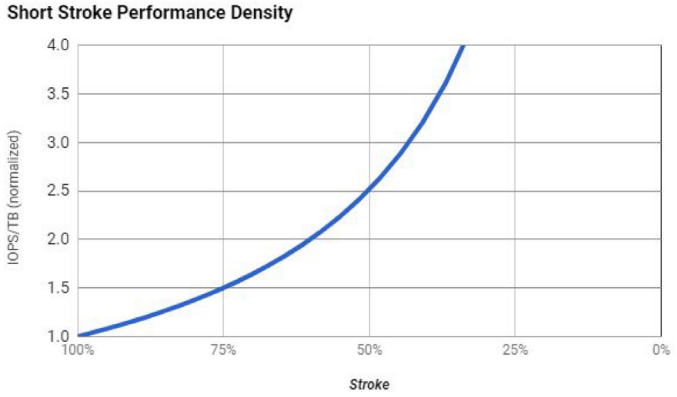

Figure 2: Performance density increase of small, random accesses from short stroking

presents a track write sequence problem similar to SMR, and the solutions invented for SMR can be applied [6]. For instance, 256 MiB worth of interlaced tracks can be mapped as a contiguous set of LBAs; this 256 MiB extent is then a logical zone, and zones can be managed as regions that must be sequentially rewritten. Or other innovative techniques might be used to manage top and bottom tracks. Beyond IMR, there are other ideas in the pipeline not yet in the public domain.

When various techniques can coexist on the same physical device, the fundamental Flex proposition of letting the OS select what recording method to use on a specified set of media is the most flexible solution.

Toward a Flex API

There is no existing API that allows an OS to change the configuration of a block device. A new interface needs to support conversions between the recording methods, and should include API improvements that kernel developers have been requesting for many years.

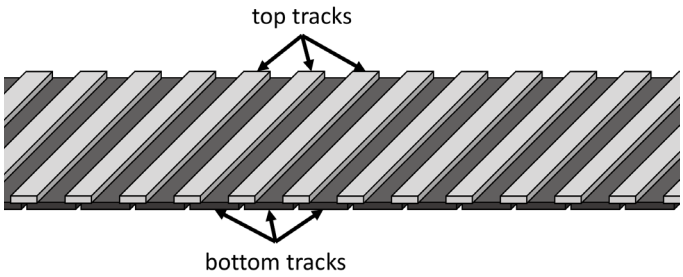

Figure 3: Depiction of interlaced track recording

Here are our goals for a new, superlative interface.

1. Provide backward compatibility
2. Leverage existing protocols
3. Support both ATA and SCSI to enable both SATA and SAS devices
4. Allow diverse configurations with fine-grained assignment of the recording method
5. Have completely discoverable capabilities
6. Enable on-the-fly conversions between recording methods
7. Protect against the loss of locked data or data that is still valid to some application
8. Be extensible to future recording methods

The Open Compute Project, T10 and T13 standards groups, will engage in the work of defining an API that meets these goals in 2018.

Providing Backward Compatibility

Flex devices should typically leave the factory in a configuration that allows existing software that is oblivious about the new capabilities to use the device. This implies that not only is a 100% CMR configuration supported but that this is the initial state. A Flex drive should be able to return to this configuration from any state.

Leveraging Existing Protocols

Cooperatively managed SMR, now codified as the zoned block device model, is the natural starting point. Zones are contiguous sets of LBAs and are always 256 MiB. Zones are either conventional zones without write pointers (for CMR space) or write pointer zones (for SMR space). The zoned block device model even already has an Offline zone state.

There are two observations about conversions that need to be addressed. First, conversions should be fast so that Flex does not introduce commands that take longer than any existing commands. It is important not to break the command timeout model that drivers use to detect dead drives.

Second, there may be valuable data that can be used to initialize space that has just come online as opposed to formatting with fill data only to immediately write the same media with valuable data. A conversion to SMR can finish with all of the space that just came online to be write pointer zones in the Empty state. This allows the device to skip initializing the SMR media with readable fill data, a process that takes about one second per 256-MiB zone. But we also want conversions to CMR space to be just as fast. The obvious extension to the zoned block device model is to define a new zone type for CMR space that also starts off as Empty, but unlike an SMR zone would have no performance penalty for random writes below the write pointer. Both CMR and SMR zones must either fail reads above the write pointer or return zeros, the former catching improper reads and the latter mimicking formatted media.

Supporting ATA and SCSI

Since ATA does not support logical unit numbers (LUNs), the Flex protocol should use separate LBA ranges for the CMR and SMR spaces. This can extend to more than two ranges when the device supports more than two recording methods when Flex is extended beyond just CMR and SMR.

For maximum flexibility, both queued and non-queued commands should be defined. And by co-developing ATA and SCSI, we can end up with a straight-wire SCSI to ATA translation (SAT) layer.

Allowing Diverse Configurations

To allow each storage stack to pursue its own optimal design point, conversions should be fine-grained. As part of embracing the zoned block device model, we want the zone to be the unit of conversion and the minimum allocation unit of the top-level allocator; that is, each zone is either online or offline, and a conversion can target any contiguous extent of zones.

For maximum short stroking benefit, all of the CMR space should be contiguous. But there may be other configurations needed. For instance, a 10-TB drive chopped into 10 one-TB pieces for 10 different tenants may want each tenant to have a CMR space and an SMR space. Thus, multiple "seams" between differing recording methods should be allowed, albeit with a small efficiency loss that averages one-half zone at each seam.

Discovering Capabilities

Device discovery includes detecting the device type through its signature and Identify Device data. Due to the backward-compatibility goal, Flex devices should identify as conventional disks. They also need to report the 100% CMR capacity in the existing capacity reporting fields.

Capabilities discovery then allows a host to learn what features a device supports. Simple additions to ATA logs and SCSI vital product data pages can serve to alert a stack that is cognizant of Flex to find out whether a Flex device is present. From there, existing zoned block device mechanisms, including Report Zones, can expose the SMR space in addition to the CMR space.

Enabling On-the-Fly Conversion

Availability is critical. Conversions need to be allowed as part of the normal workflow and not be restricted to system integration or an offline mode. Conversion commands need reordering constraints if they overlap reads or writes to the same LBAs, but

the rest of the LBA space that is not participating in a conversion needs not only to retain its data, but concurrent reads and writes must be allowed.

Protecting Valid and Locked Data

Before a conversion that takes space offline, any data in that space that is still valid for some application needs to be copied, either to space on this disk that will stay online or elsewhere. Since a conversion operation has a side effect of making previously written data unreadable, a conversion that gets ahead of the valid data copy process will lose data.

The existing SMR zone types support an operation, Reset Write Pointer, for the host to move a zone's write pointer back to the start of the zone. Since reads to LBAs above a write pointer either fail or return zeros, this also declares that the previously written data are discarded. Extending the Reset Write Pointer operation to the CMR zones allows a strong, firm handshake in the protocol: requiring that a zone's write pointer is reset before it is allowed to be converted to Offline, the conversion itself has no data retention side effects.

Similarly, enforcing that zones must be unlocked for a conversion allows a security management layer to know that locked data cannot be lost through execution of new Flex commands.

Being Extensible

New zone types can be defined as needed to support techniques like interlaced tracks. Other innovations might pack data more densely in other ways, but the tradeoffs often break legacy requirements. Simply getting all of the media provisioned to user-addressable space has been boxed in by ingrained assumptions about static configurations.

While the first generation of Flex will address the hybrid mix of CMR and SMR, the protocol needs to be extensible. This means that capabilities reporting and conversion commands need to be open to more than just two recording methods.

Adding Value

Flex Dynamic Recording recognizes that a single hardware configuration can be deployed in various ways, all the way down to physical recording methods on media. The philosophy of Flex is that allowing the owner of a device to configure what recording method is best for them adds value to the whole system. So rather than locking down the method at the factory, Flex moves the decision to the field.

References

[1] E. Brewer, L. Ying, L. Greenfield, R. Cypher, and T. Ts'o, "Disks for Data Centers": https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/44830.pdf.

[2] T. Ts'o, "Hybrid-SMR Product Requirements Proposal for OCP": http://files.opencompute.org/oc/public.php?service=files&t=50192ac3fff6f7d96c314dc39cd92f26.

[3] Seagate, "New Flex Dynamic Recording Method Redefines the Data Center Hard Drive": https://blog.seagate.com/intelligent/new-flex-dynamic-recording-method-redefines-data-center-hard-drive/.

[4] Western Digital, Dynamic Hybrid SMR: https://itblog.sandisk.com/dynamic-hybrid-smr/.

[5] E. Hwang, J. Park, R. Rauschmayer, and B. Wilson, "Interlaced Magnetic Recording (IMR)," *Journal of Transactions on Magnetics*, vol. 53, no. 4 (April 2017): http://ieeexplore.ieee.org/document/7781604/.

[6] T. Feldman and G. Gibson, "Shingled Magnetic Recording: Areal Density Increase Requires New Data Management," *;login:*, vol. 38, no. 3 (USENIX, June 2013): https://goo.gl/wj5Doi.