SNIA DSI '16 DATA STORAGE INNOVATION CONFERENCE

Innovation in Storage Products, Services, and Solutions

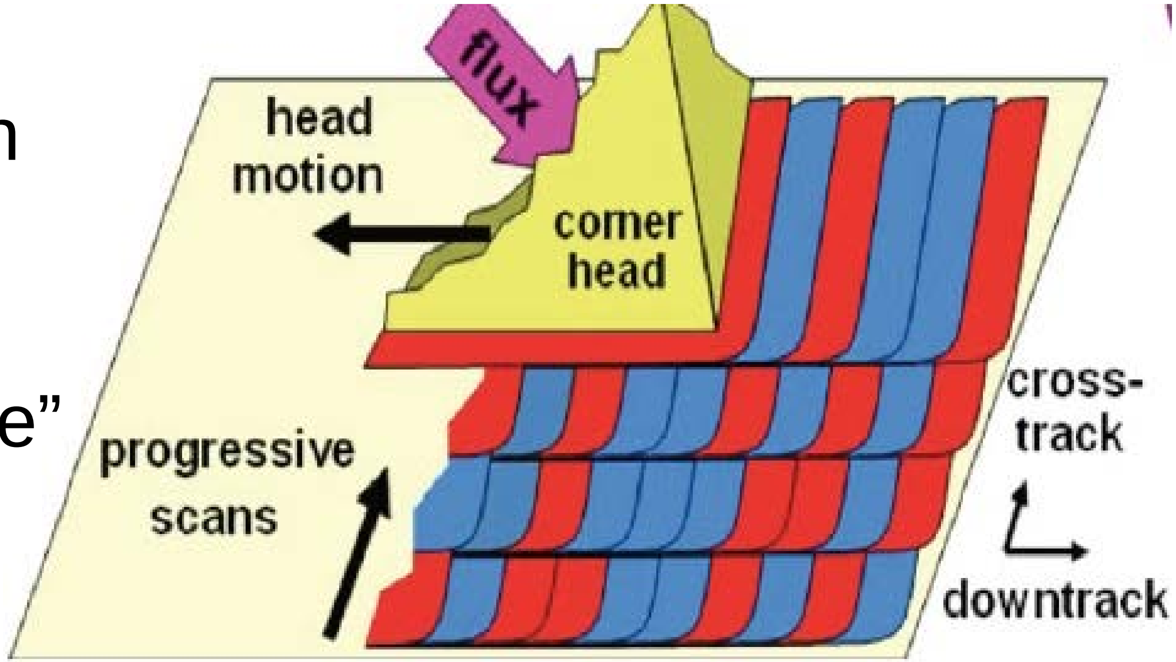
June 13-15, 2016 | Marriott San Mateo | San Mateo, CA

Shingled Magnetic Recording (SMR) Panel: Data Management Techniques Examined

Tom Coughlin
Coughlin Associates

Introduction

- SMR partially overwrites written tracks with new tracks
- Creates an “erase” process in HDDs
- Increase in Areal Density but increase in performance overhead too



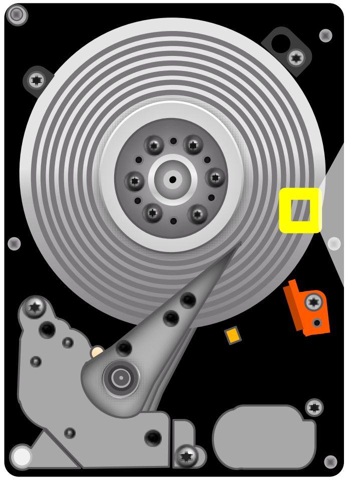
Session Participants

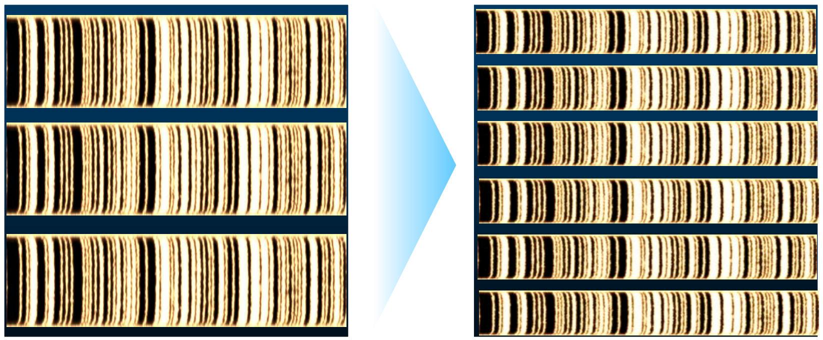
- Jorge Campello,
  - Global Director of Systems and Solutions, Western Digital
- Mark Carlson
  - Principal Engineer, Industry Standards, Toshiba
  - Chair, SNIA Technical Council
- Josh Bingaman
  - Firmware Engineering Manager, Seagate Technology

How SMR is Implemented on HDDs

Jorge Campello
Western Digital

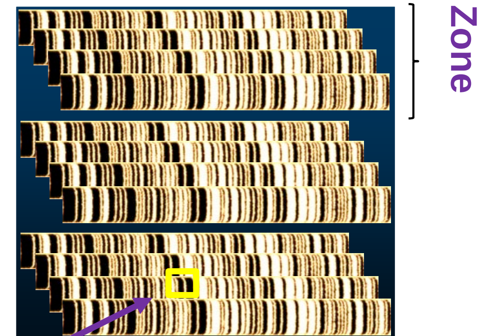
What is Shingled Magnetic Recording?





Conventional PMR HDD

Data in Discrete Tracks



SMR HDD

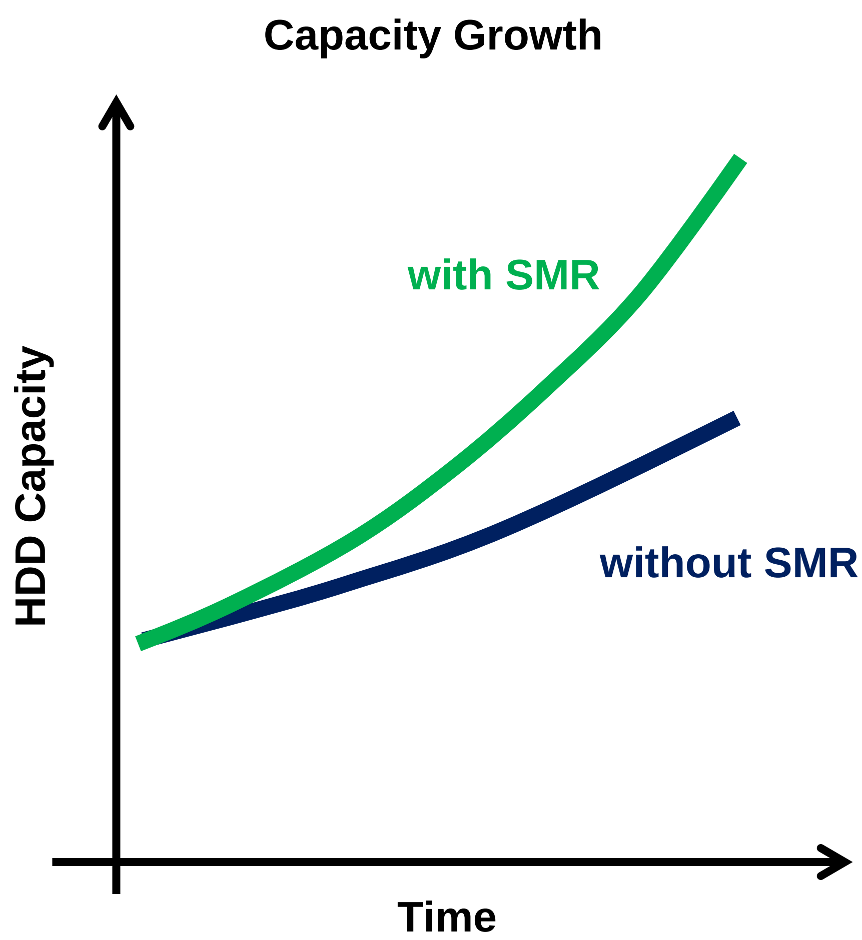
Data in Zones of Overlapped Tracks

While Zones are independent, we can’t change sectors independently within a Zone.

SMR Standards

- T10: ZBC
- T13: ZAC

Why SMR?



SMR accelerates areal density growth

Some Architectural Constructs

- Caching
  - Stage writes to sequentialize the IOs.
  - This can be done both on the media or on Solid State Storage.
- Indirection system
  - Not a fixed mapping from LBA to physical location
- Over provisioning
  - Need extra space for internal bookkeeping
- Garbage Collection
  - Need background process to fix up the data-structures.
- Indirection system storage
  - Need special mechanism to maintain the indirection system.
- Solid State NV Storage
  - Emergency storage for indirection system

Drive Managed Model

- Sequential Read
  - Similar to PMR

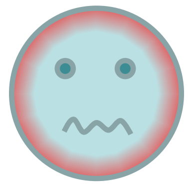
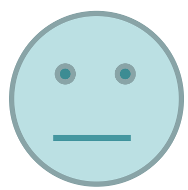
- Random Read
  - Similar to PMR

- Sequential Write
  - Similar to PMR

- Random Write
  - YMMV

?

Drive Managed Model: Random Write

	Small Block	Large Block	Huge Block
High Duty Cycle	Performance dominated by seek time. Caching writes on media and moving later has good performance. 	Seek time no longer dominates. Writing twice has overhead. High duty cycle fills up cache quickly and doesn't allow time for recovery. 	Behaves close to sequential writes. 
Low Duty Cycle	Performance dominated by seek time. Caching writes on media and moving later has good performance. 	Seek time no longer dominates. Writing twice has overhead. Low duty cycle allows drive to hide overhead. 	Behaves close to sequential writes. 

SNIA DSI '16 DATA STORAGE INNOVATION CONFERENCE

Innovation in Storage Products, Services, and Solutions

June 13-15, 2016 | Marriott San Mateo | San Mateo, CA

SMR Management Models and Standards

Mark Carlson
Toshiba
SNIA Technical Council

SMR Management Models

- Hide the complexity of SMR from host software
  - Drive Managed model – performance impact
- Allow the host software to manage the SMR complexity
  - Host Managed Model – best performance, but all new software
- Something in between
  - Host Aware Model

SMR Standards

- For Serial ATA (SATA)
  - ZAC – Zoned ATA Commands
- For Serial Attached SCSI (SAS)
  - ZBC – Zoned Block Commands
- Two primary commands
  - **Report Zones** – discover zone configuration and write pointers
  - **Reset Write Pointer** – reset the write pointer the the beginning of zone (destructive to zone contents)

Writing to a Zoned Device

- Additional commands:
  - **Open Zone** – nail down resources for a zone
  - **Close Zone** – free up those resources
  - **Finish Zone** – fill out the remaining space
- Proposed simplification
  - Allow **Report Zones** even in a Drive Managed model

ZAC/ZBC Host Interactions

Josh Bingaman
Seagate Technology

Primary Host Issue: Non-Sequential Writes

- File modifications via appends are a primary example (write in place) - does not conform to ZAC/ZBC
- Host workloads would need to become copy on write for modifications and discard/trim old data
- This ensures writes are written at the write pointer – friendly both for Host Aware as well as Host Managed
- Multiple ways to solve this problem…

Full Stack Solution

- This design would require modifying file systems and relevant parts of the I/O stack as appropriate to conform to ZAC/ZBC specifications
- Many cases likely require extensive modifications – complicated!
- Provides optimal performance as all layers of the stack are aware of ZAC/ZBC with no accounting overhead on system resources

Emulation / Shim: Sequentializer (STL)

- Translation layer akin to FTL
- Maintains LBA remap and requires metadata storage and searches as well as garbage collection
- Possible workload dependent performance implications, but majority of the I/O stack does not need to change
- Open Source prototype example for Linux
  - https://github.com/Seagate/ZDM-Device-Mapper

Emulation / Shim: Caching

- Use part of the drive (conventional space) for a “random” cache to clean later
- Garbage collection and metadata tracking/searching required, similar to sequentializer
- Different performance tradeoffs than sequentializer

Impact of SMR on the Storage Marketplace

- SMR increases HDD areal density
  - But increased performance overhead with "erase" cycle
  - SMR may be best for archive or write seldom applications
- SMR could be path to two-dimensional magnetic recording (TDMR) which could increase AD further
- SMR with He could be basis of future cold storage near-line HDDs—largest growing HDD segment