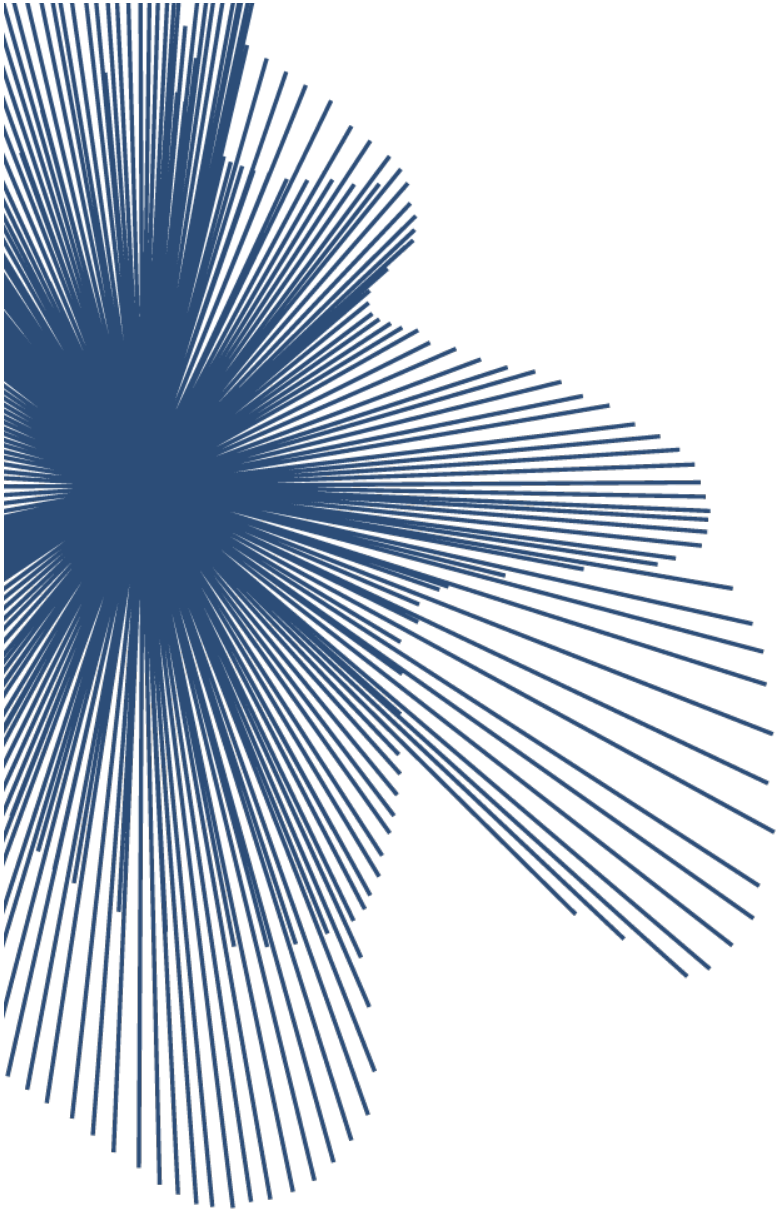


SMR Impact on Linux Storage Subsystem

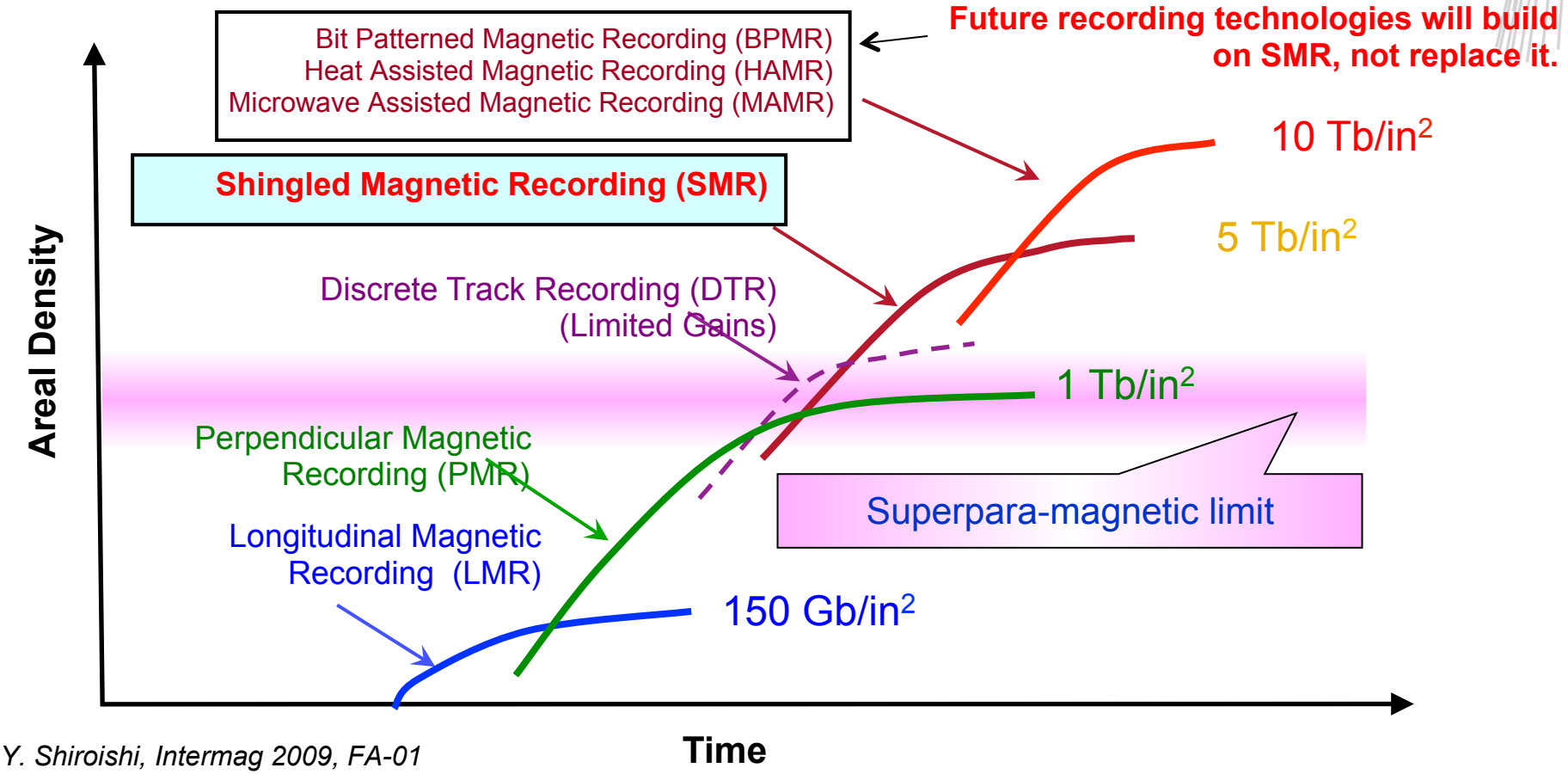
Jorge Campello, Adam Manzanares

HGST, a Western Digital Company.

HGST
a Western Digital company

Magnetic Recording System Technologies

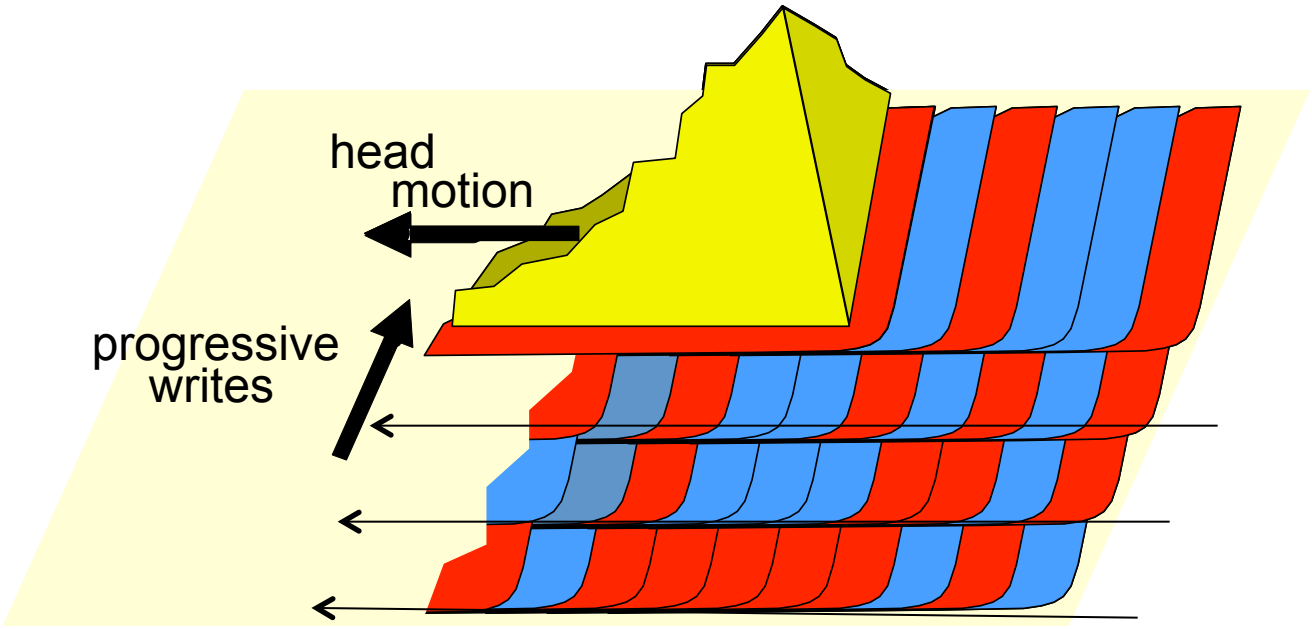
New recording system technologies are needed to keep the HDD industry on its historical track of delivering capacity improvements over time



Y. Shiroishi, Intermag 2009, FA-01

What is Shingled Magnetic Recording (SMR)?

SMR write head geometry extends well beyond the track pitch in order to generate the field necessary for recording. Tracks are written sequentially in an overlapping manner forming a pattern similar to shingles on a roof.



SMR Constraint:
Rewriting a given track will damage one or more subsequent tracks.

Wood, Williams, et al., IEEE TRANSACTIONS ON MAGNETICS, VOL. 45, NO. 2, FEBRUARY 2009

SMR Types

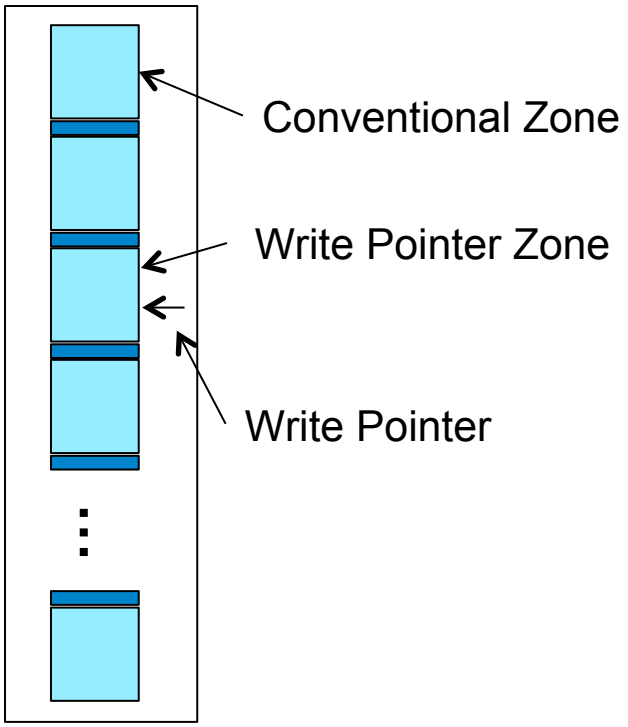
SMR category	Description
Drive managed (Autonomous)	No host changes. SMR device manages all requests. Performance is unpredictable in some workloads. Backward compatible
Host aware	Host uses new commands & information to optimize write behavior. If host sends sub-optimal requests the SMR device accepts the request but performance may become unpredictable. Backward compatible
Host Managed	Host uses new commands & information to optimize write behavior. Performance is predictable. If host sends sub-optimal requests the SMR device rejects the request. Not backward compatible

T10/T13 ZBC/ZAC

ZBC = Zoned Block Commands

ZAC = Zoned ATA Commands

Zoned Block Device



- **The device is divided into Zones**
- **Two type of Zones**
  - Conventional Zones
  - Write Pointer Zones
- **Conventional Zones**
  - Behave according to the direct access block device type model in SBC-3.
- **Write Pointer Zones**
  - Behave according to the new Zoned Block Device Model, and varies depending on device type
    - There is write pointer (WP), writes should be at WP.
- **Two device types**
  - Host Managed Zoned Block Device
    - Writes not at WP, or spanning Zones are not allowed.
    - Reads not allowed to span Zones, or cross WP.
  - Host Aware Zoned Block Device
    - Non sequential writes allowed, spanning zones allowed for R/W.

Note: These are high-level simplifications. For a more accurate description see the relevant T10/T13 documents.

ZBC/ZAC Device Types – current drafts

	Direct Access	Host Aware	Host Managed
Peripheral Device Type	00h	00h	14h
HAW_ZBC	0b	1b	0b
Conventional zones	n/a	Optional	Optional
Seq'l wr preferred zones	n/a	**Mandatory**	Disallowed
Seq'l wr only zones	n/a	Disallowed	**Mandatory**
Reads and writes crossing seq'l write only zone boundaries	n/a	n/a	Disallowed
REPORT ZONES	Disallowed	Mandatory	Mandatory
RESET WRITE POINTER	Disallowed	Mandatory	Mandatory

SMR Introduction Models

User Space

Kernel

Hardware

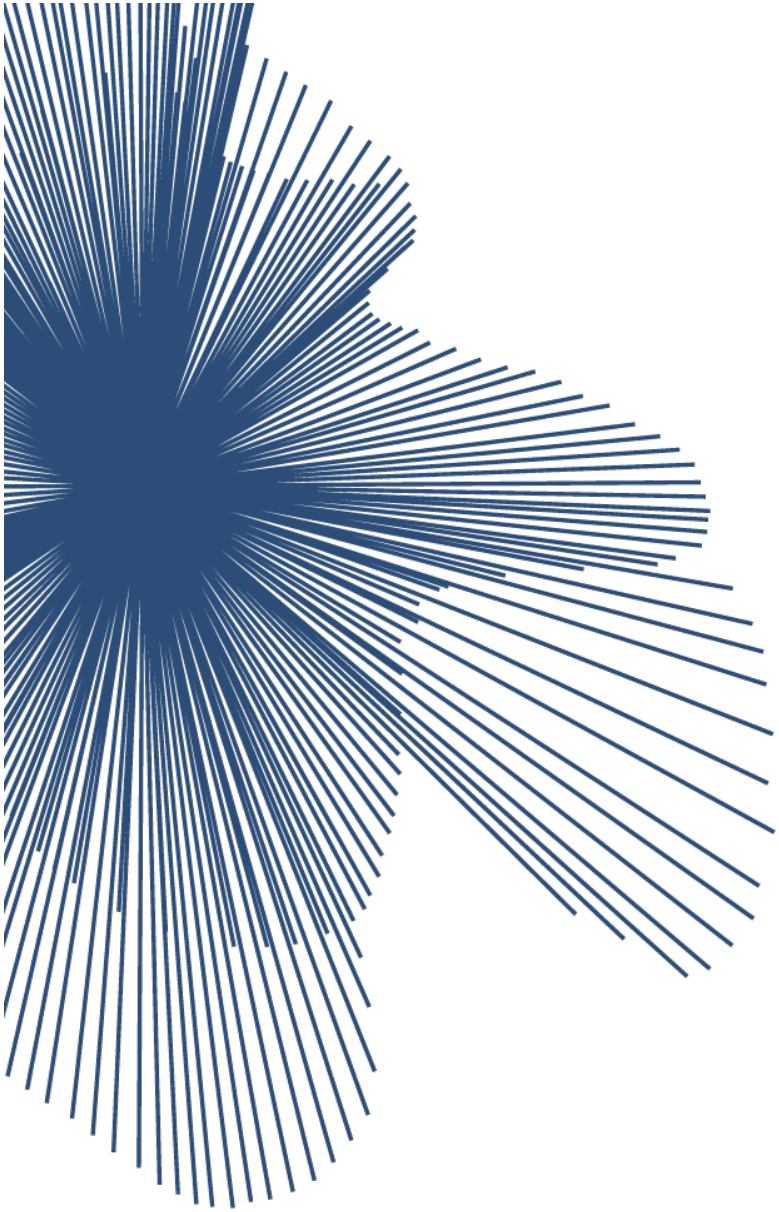
ZBC/ZAC
Host Aware

ZBC/ZAC

Host Aware
Host Managed

ZBC/ZAC

Host Aware
Host Managed



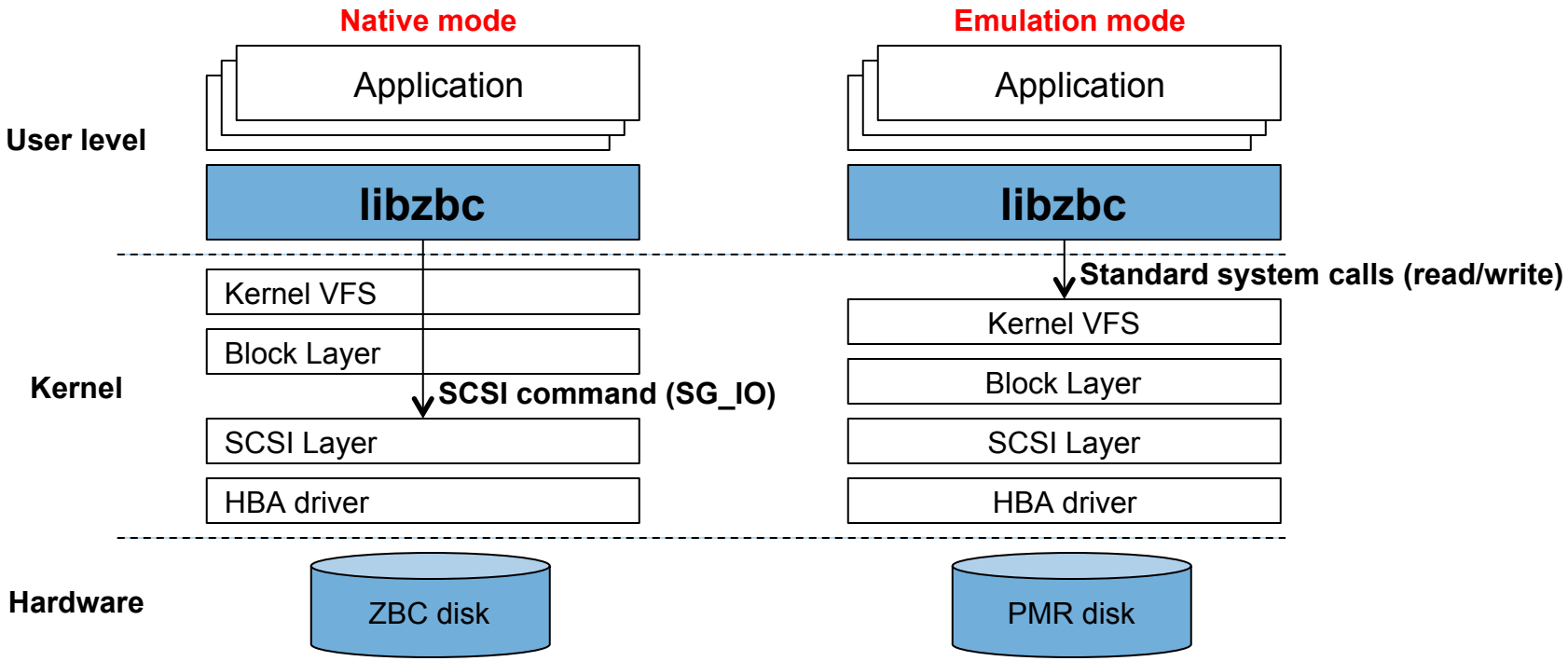
libzbc and lkvs:
Linux ZBC library and Linear Key Value Store Application

Storage Architecture (SJRC & JRL)
HGST Research

HGST
a Western Digital company

libzbc

- Allows Linux applications access to ZBC host-managed disks
  - Access to disk zone information and read/write operations in zones through direct SCSI command execution (SG_IO)
  - ZAC drives will be supported by libzbc as well
- Additionally, provide a ZBC emulation layer for operation on top of standard SAS/SATA block devices
  - Zone configuration of the disk is emulated within the library



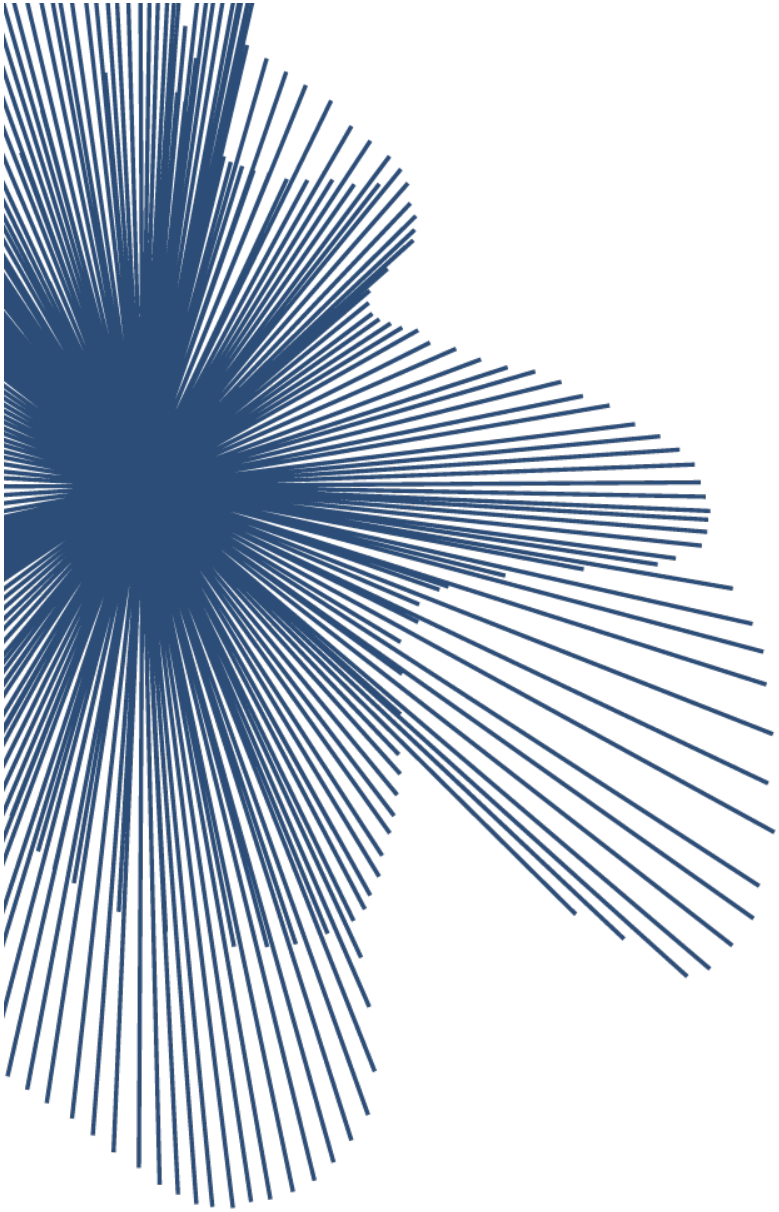
libzbc Interface

Functions	Description	Input	Output	SCSI command (native mode)
zbc_open	Open a device	Device file path	Device handle	INQUIRY, READ CAPACITY 16
zbc_close	Close an open device	Device handle	None	None
zbc_get_device_info	Get a device information (size, sector size, ...)	Device handle	Device information	None
zbc_report_zones	Get information on zones following a specified LBA	Device handle, zone start LBA, zone filter	Zone information	REPORT ZONES
zbc_reset_write_pointer	Reset the write pointer of an open or full zone	Device handle, zone start LBA	None	RESET WRITE POINTER
zbc_pread	Read data from a zone	Device handle, Zone to read, LBA offset in the zone, number of sectors to read, data buffer	Amount of sectors read and data	READ 16
zbc_pwrite	Write data to a zone	Device handle, Zone to write, LBA offset in the zone, number of sectors to write, data buffer	Amount of sectors written	WRITE 16

libzbc Interface (Emulation Mode)

- These functions are used to initialize an emulated ZBC device
  - Write pointer persistency is also emulated
    - » Zone configuration and current write pointer values are saved to the disk on execution of the zbc_close function

Functions	Description	Input	Output	SCSI command (native mode)
zbc_set_zones	Configure the zones of an emulated device	Device handle, size of conventional zone, size of sequential write zones	None	None*
zbc_set_write_pointer	Change a zone write pointer LBA value	Device handle, zone start LBA, write pointer value	None	None*



Linear Key Value Store (lkvs) Application

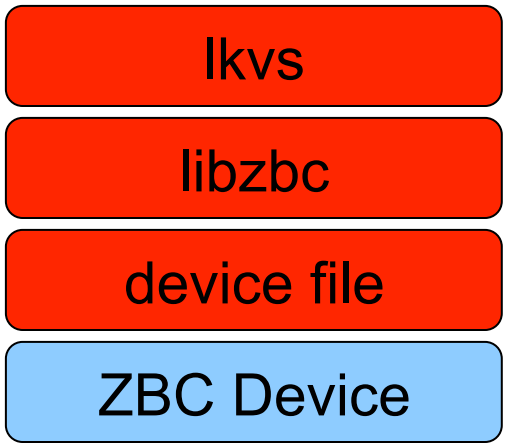
HGST
a Western Digital company

Linear Key Value Store Architecture

- lkvs
  - Implements a simple append only KVS as an example use of libzbc
  - Queries drive info (write pointer, zone information) through libzbc
  - Read/write executed through libzbc
- libzbc
  - Provides zone information, write pointers, to lkvs

Applications link with libzbc

lkvs gets ZBC device information and read/write operations are perfromed through libzbc



lkvs Interface

Functions	Description	Input	Output
openDev	Open a device	Device file path, format flags	Bool success
Put	Insert key/value pair into the store	Key string, value buffer, size	Bool success
Get	Get key/value pair form the store	Key string, value buffer, size	Bool success
List	List key/value pairs on the device (Not Finalized)	TBD	TBD

