

Ultrastar[®] Hs¹⁴ Implementation Guide

Introduction

As the industry’s only 2nd-generation enterprise-class SMR 14TB HDD, Ultrastar® Hs¹⁴ addresses the emerging sequential-write segment in the capacity enterprise space. Optimized for highest capacity at the lowest TCO, the Hs¹⁴ provides unprecedented capacity leadership by harnessing two core complementary technologies – HelioSeal® and host-managed shingled magnetic recording (SMR) – to deliver uncompromising quality and reliable performance.

Due to the nature of host-managed SMR design, Ultrastar Hs¹⁴ drives will not work as drop-in replacements for traditional capacity enterprise drives. To take advantage of the capacity and predictable performance of host-managed SMR, customers will need to modify their end application or operating system to interface with new command sets, and data streams must be sequentialized for writing to the drive. The investments made in software changes will also help enable future host-based SMR solutions, and provide a path to future recording technologies.

Getting Started

There are three main approaches that are available for applications to access host-managed SMR HDDs. The direction of approach is dependent on the host system architecture and the ability of customers to modify their applications. See Figure 1.

1. Direct application access: For application software that directly access HDDs, new management commands need to be issued based on the newly defined ZBC/ZAC specifications for host-managed SMR, when the device is accessed through a SCSI generic (SG) node. Applications must also thoroughly control their I/O execution pattern to ensure that write operations are sequential within a zone when the device is accessed through an SG node or as a raw block device. The Linux Kernel since version 3.19 exposes SMR HDDs through a SCSI generic node. The Linux kernel since version 4.10 exposes SMR HDDS as block devices.

2. File system access: For applications needing access to the HDD via a file system, the ZBC/ZAC specifications can be abstracted using a file system designed for host-managed SMR. This approach allows applications to store and retrieve data through a standard POSIX file interface, minimizing the need for application modifications. f2fs, a Linux based File System, includes SMR support from kernel version 4.10 onward.

3. Device mapper abstraction: Host-managed SMR disks constraints can be abstracted within the operating system block device layer and exposed to application as regular disks. Random write operations

can be processed, allowing reusing existing software on top of the abstracted block device, including existing file systems. Such approach can however introduce overhead potentially degrading performance depending on the application workload. The Linux Kernel, since version 4.13, supports the zoned device mapper target, which provides block device driver level abstraction.

Various Approaches to SMR Implementation

- **Direct Access from Application**
 - HGST libzbc
 - sg3utils
- **File Systems**
 - F2FS support in kernel 4.10
 - More will come in future
- **Device Mapper**
 - Included in kernel 4.13
 - Allows using any file system

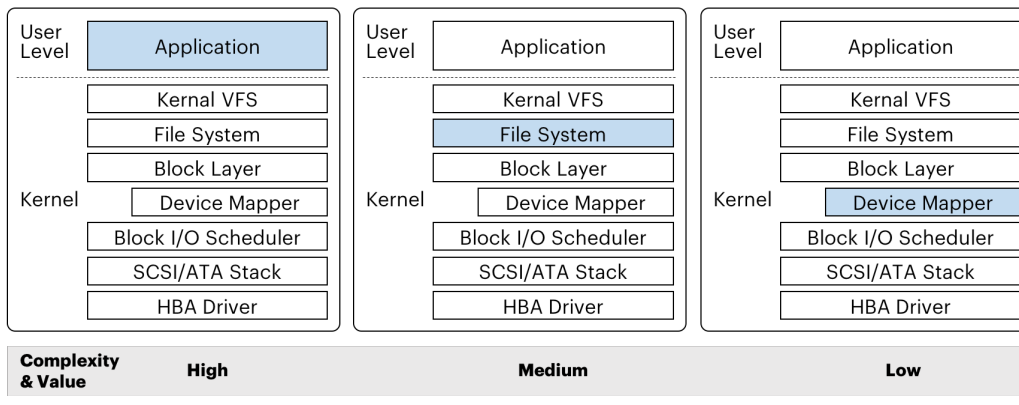


Figure 1. Each option depends on host system architecture and user’s ability to modify their applications

Application Level Support

For an optimal experience with host-managed SMR, modification at the application software level is highly recommended. For Linux-based applications, HGST offers the libzbc open source library allowing applications to implement direct access to host-managed SMR HDDs. The library also supports three different low-level drivers under a single unified interface for ease of implementation.

libzbc also provides a suite of shell command line tools allowing manipulation of a disk zones. Referencing these **libzbc** functions can help in a proprietary application development process.

Finally, **libzbc** also includes a simple example application as well as various tools useful for testing and verifying the correct behavior of applications.

This library is available at:
<https://github.com/hgst/libzbc>.

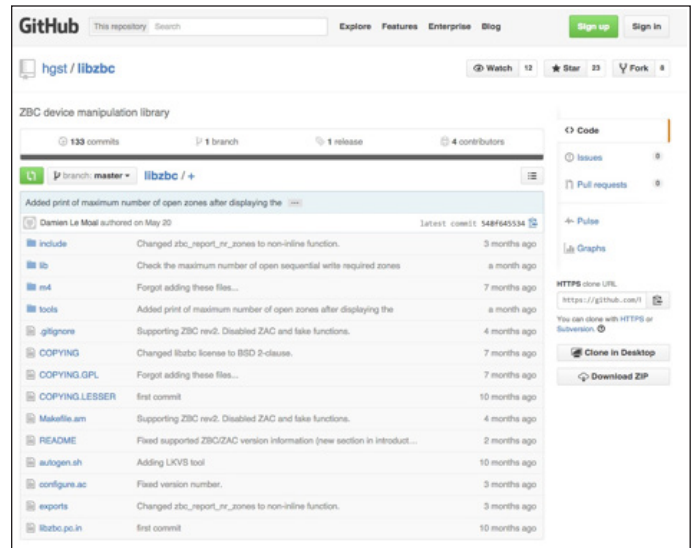


Figure 2. HGST’s libzbc open-source library for host-managed SMR drives

*The ZBC and ZAC standards are available from the INCITS T10 and T13 technical committees. Please refer to T10 Zoned Block Command (ZBC) Draft revision 5 document¹ for SMR SCSI command set specification for Ha10. T13 Zoned-device ATA command set (ZAC) draft revision 5² provide Hs14 SATA command specifications.

¹ <http://www.t10.org/cgi-bin/ac.pl?t=f&f=zbc-r05.pdf>

² http://www.t13.org/Documents/UploadedDocuments/docs2015/di537r05-Zoned_Device_ATA_Command_Set_ZAC.pdf