HGST
a Western Digital company

Panel:
Shingled Disk Drives—File System Vs. Autonomous Block Device

Zvonimir Bandic,
Storage Architecture, HGST Research

Credits

Indirection Systems for Shingled-Recording Disk Drives

Yuval Cassuto*, Marco A. A. Sanvido*, Cyril Guyot*, David R. Hall† and Zvonimir Z. Bandic*
Hitachi Global Storage Technologies
* San Jose Research Center
3403 Yerba Buena Road
San Jose, California 95135 USA
† Advanced Magnetic Recording Laboratory
3605 Hwy 52 N
Rochester, Minnesota 55901 USA
Email: {yuval.cassuto, marco.sanvido, cyril.guyot, david.hall, zvonimir.bandic}@hitachigst.com

Abstract—**Shingled magnetic recording is a promising technology to increase the capacity of hard-disk drives with no significant cost impact. Its main drawback is that random-write access to the disk is restricted due to overlap in the layout of data tracks. For computing and storage systems to enjoy the increased capacity, it is necessary to mitigate these access restrictions, and present a storage device that serves unrestricted read/write requests with adequate performance. This paper proposes**

icant additional capacity gains [1]. Newer magnetic
technologies, such as bit-patterned media (BPM) [2] a
assisted magnetic recording (HAMR) [3], hold the pr
scale beyond conventional media, but neither has yet
to product-level functionality. A different path toward
recording densities is offered by the technology of
recording[1]. Without a dramatic rework of the media,
recording enables the usage of write heads with
fields and higher tolerances, thereby allowing increa

Proceeding MSST '10 Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)

Shingled File System

Host-Side Management of Shingled Magnetic Recording Disks

Damien Le Moal
Hitachi Ltd., Yokohama Laboratory
Yokohama, Japan
damien.lemoal.mb@hitachi.com

Zvonimir Z. Bandic, Cyril Guyot
Hitachi GST, San Jose Research Center
San Jose, CA, USA
{zvonimir.bandic, cyril.guyot}@hitachigst.com

Abstract—**Shingled Magnetic Recording disks overlap tracks to increase storage density, and thus require a special processing to achieve unrestricted random sector updates without data loss. The Shingled File System is a host-side level solution to this problem and allows more efficient optimizations compared to a disk firmware based approach. Experimental results show that no overhead is incurred for large file workloads, making SFS a suitable solution for video processing devices such as personal video recorders and set-top-boxes.**

Keywords- file system, shingled magnetic recording, hard-disk

I. INTRODUCTION

Shingled Magnetic Recording (SMR) allows increasing the recording density of hard disk drives at low cost, by reducing data tracks pitch and tracks width and using write heads with stronger magnetic fields. Storage capacity increases using this technique have been estimated at between 15 % and 40 % [1][2]. However, SMR also has a serious drawback in the form of strong adjacent track interferences resulting in effect in overlapping tracks, as shingles on a house roof overlap. Writing data in a track causes data loss in overlapping tracks.

To overcome such write access restrictions, disk firmware level solutions using indirection systems have been proposed [3], allowing using SMR disks as drop-in replacement in exiting systems. Another possible solution is handling the overlapping of tracks at the host level. This solution requires modifications to the host software and is thus mainly applicable only to consumer electronics devices with a closed environment, such as home NAS devices or set-top-box and personal video recorder (PVR) systems.

II. SHINGLED FILE SYSTEM

The Shingled File System (SFS) implements support for SMR disks without native shingling handling implemented, or enabled, in firmware. SFS operates under the assumption that any write to the disk may cause data loss in sectors in the track following the track being written. In other words, a forward shingling direction is assumed with a write width of 2 tracks.

A. Disk Format

SFS manages the disk storage space using 4 KB blocks grouped into zones of contiguous shingled tracks of a disk platter. Shingled zones are 64 MB in size and separated by gaps (an unused track) to prevent data loss in the first track of a zone due to sector updates in the last track of the preceding zone. SFS furthermore differentiates shingled zones into sequential-write shingle zones and random-write shingle zones. This organization is shown in figure 1.

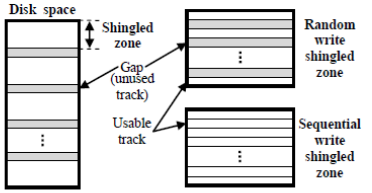
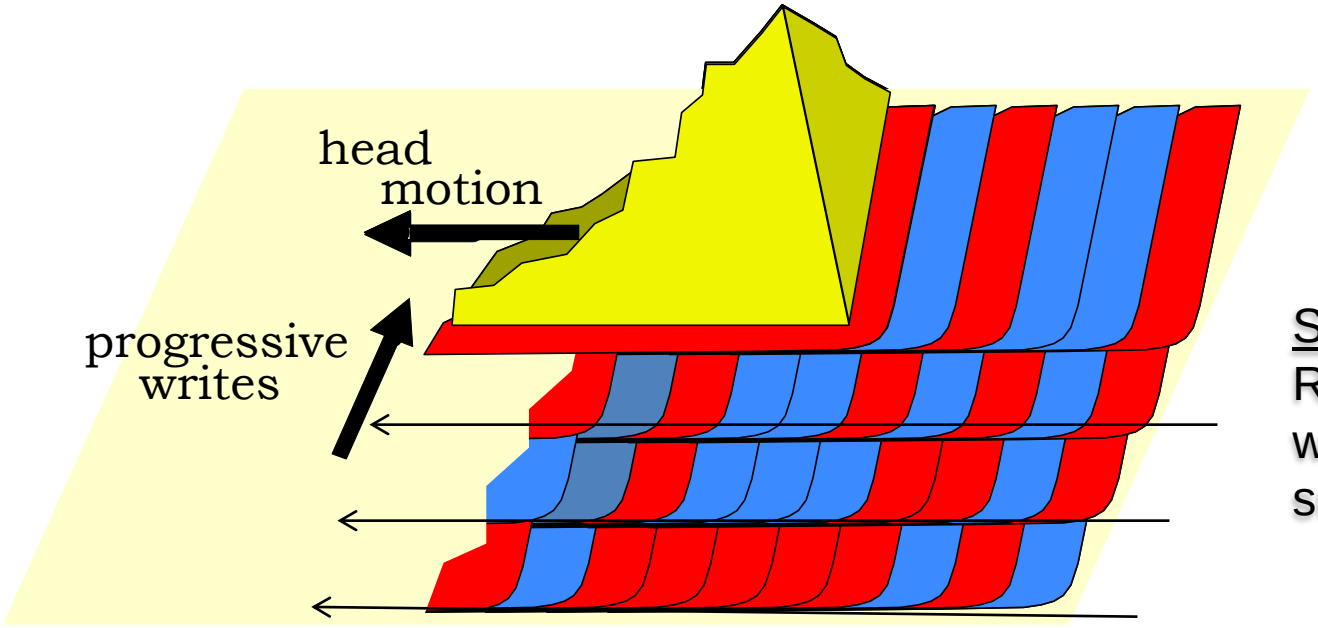


Figure 1. SFS disk format: the disk space is divided into shingled zones separated by gaps

The 30th IEEE International Conference on Consumer Electronics (ICCE2012)

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

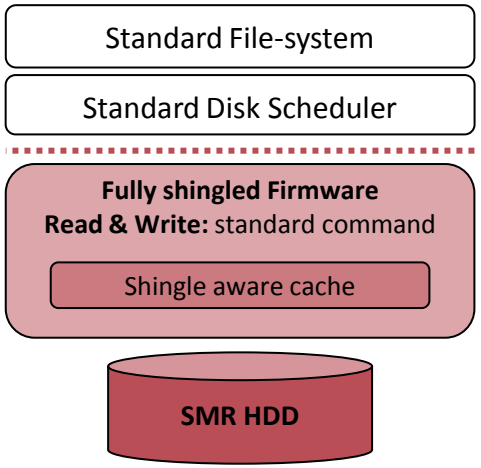
Introduction: Motivation and Goal

HITACHI
Inspire the Next

- **SMR disks require special processing to avoid data loss during write command execution**
  - Basically: read the track following the one to be written, write target track, re-write following tracks
- **Two basic implementation approaches (similar to implementation for Flash memory)**
  - HDD controller level implementation
  - Host side file system level implementation

HDD-side implementation

- Standard HDD command interface
- Firmware ensure respect of shingling constraint

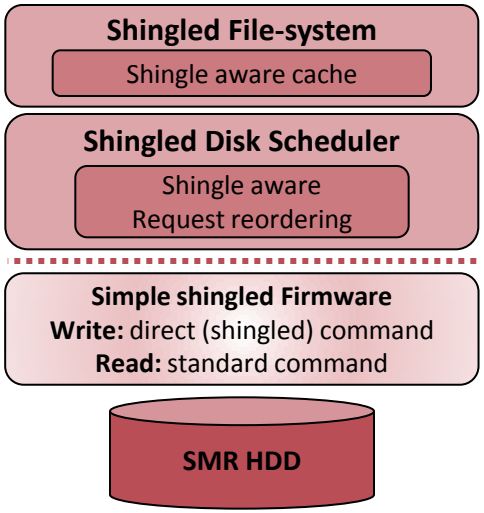


○ **Drop-in replacement possible (Any file system supported)**

❖ **HDD firmware development difficult**

Host-side implementation

- Direct (shingled) write exposed to host
- File-system ensure respect of shingling constraint



○ **More optimizations possible using file system metadata**

○ **Faster development of HDD firmware**

❖ **Can be used only in specific (closed) environments**

Project goal: Using host-side implementation (file system), support shingled HDD with minimal HDD firmware functions for application specific environments (i.e NAS or DVR systems)

Drive vs. Host Indirection

HITACHI
Inspire the Next

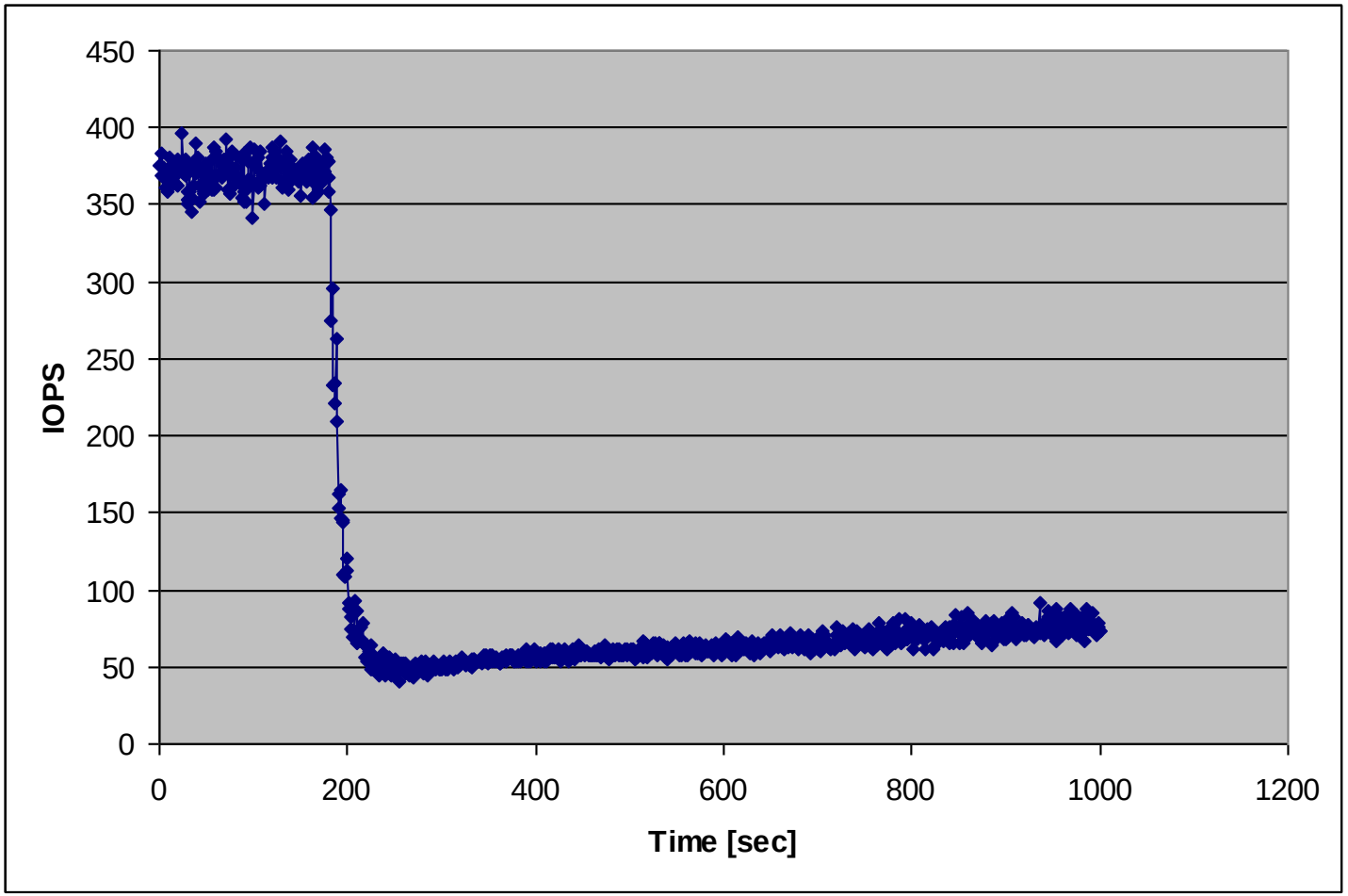
Why on the drive?

- Transparent to Host
- Complete knowledge of physical layout

Why on the host?

- “Shingle aware” access and allocation
- System specific performance optimization

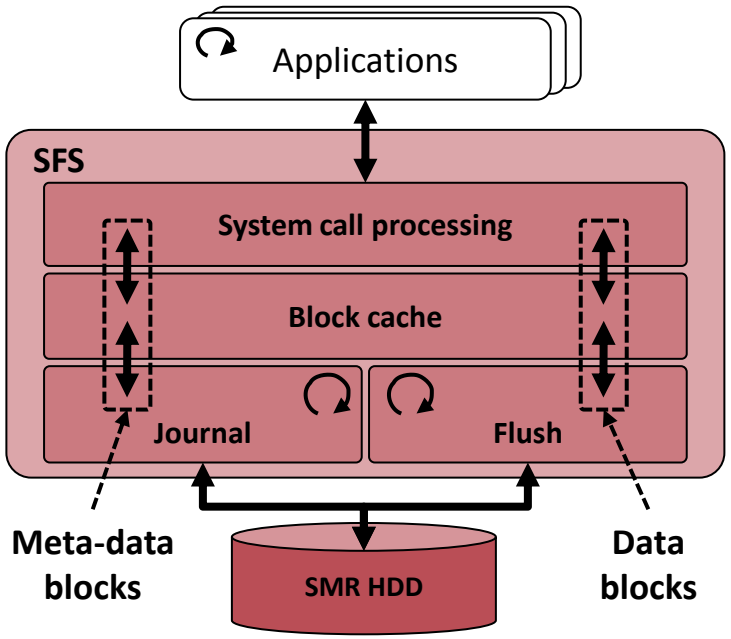
4K Random IOPS



Shingled File System: Overview

HITACHI
Inspire the Next

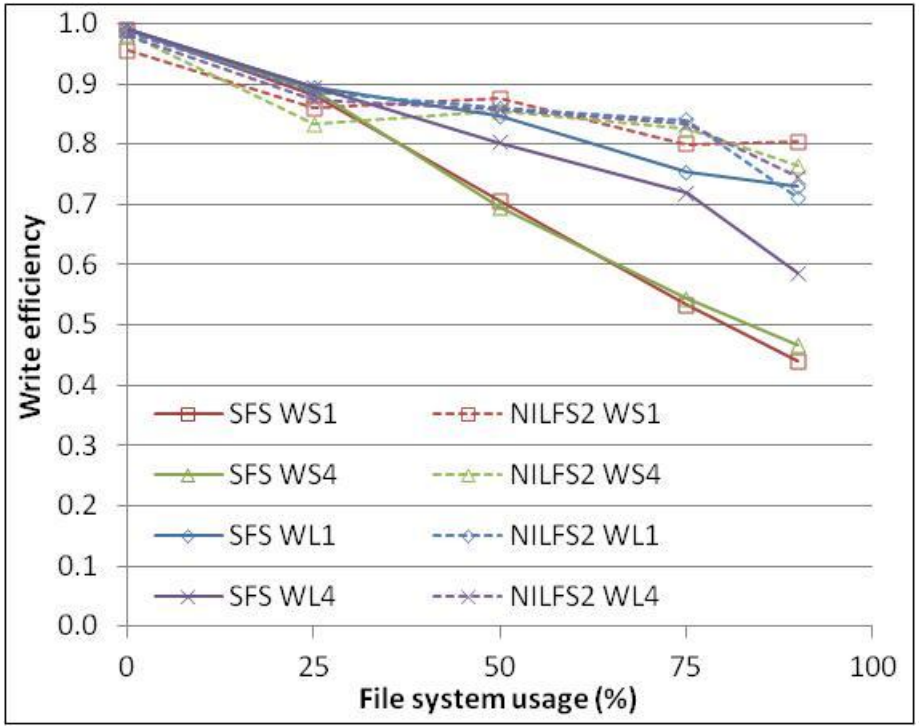
- **The Shingled File System (SFS) is a host-based journaling file system supporting shingled magnetic recording (SMR) disks**
  - Presents a standard API to applications (traditional POSIX set of system calls)
- **Design based on the following assumptions**
  - Track information (track mapping to LBA) is available or can be retrieved from the disk
  - Disk interface is standard: write command exposes directly to the host the shingling constraint (data loss)
- **SFS implements shingling support through a shingle-aware block cache (in host memory) optimized with a shingle-aware block allocation method**
  - All read and write operations by applications, as well as internal disk accesses by the file system (meta-data) are processed through a disk block (page) cache
  - Meta-data write (updates on disk) are always executed through journaling
  - File data block updates on disk are processed through a flush daemon process which avoids loss of data

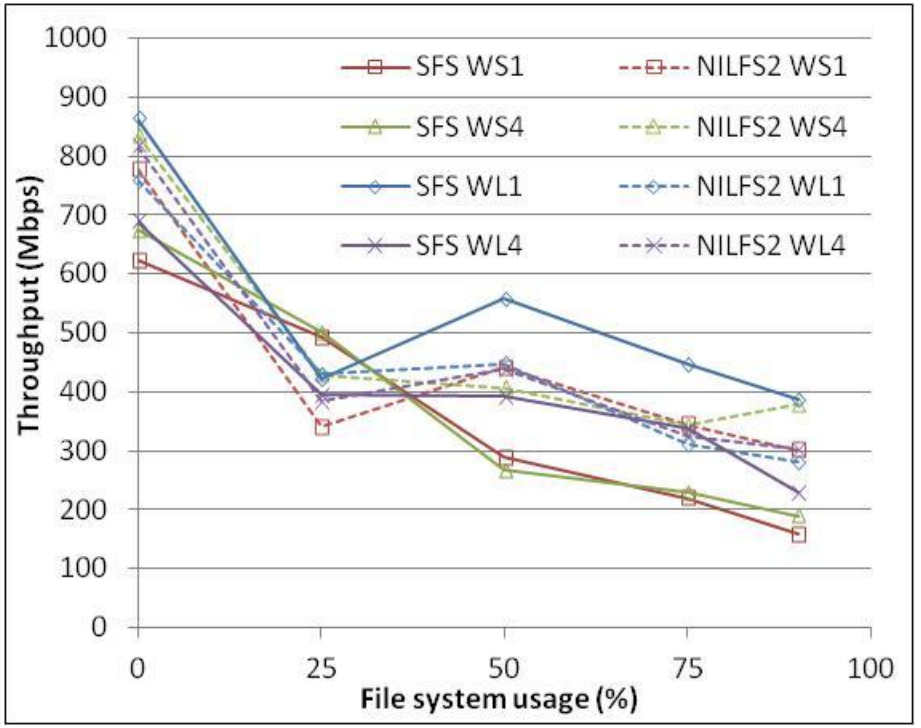


Evaluation Results: Small Files Aged FS

HITACHI
Inspire the Next

- **Write efficiency degrades with disk usage for both file systems**
  - For SFS, higher fragmentation of shingled data regions results in increased block update overhead (read-rewrite use over 50 % of disk throughput in worst case)
  - For NILFS2, fragmentation of log regions forces an increased activity of block reclaiming process, resulting in a higher overhead (up to 29 % in worst case)
- **Disk throughput comparable for both file systems**
  - Lower seek in average for SFS compared to NILFS2 log region defragmentation improves throughput

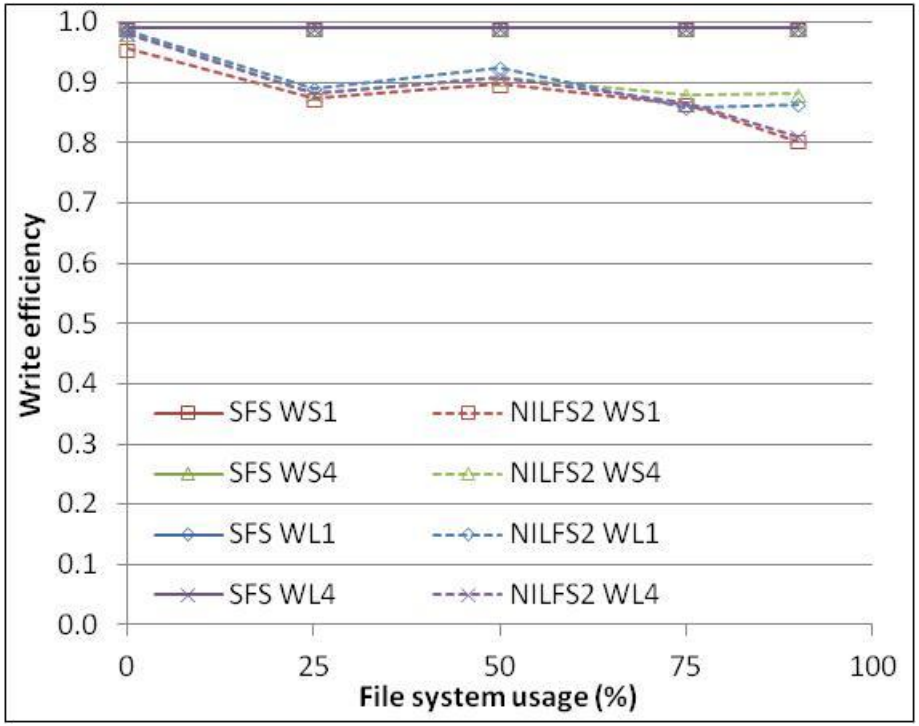


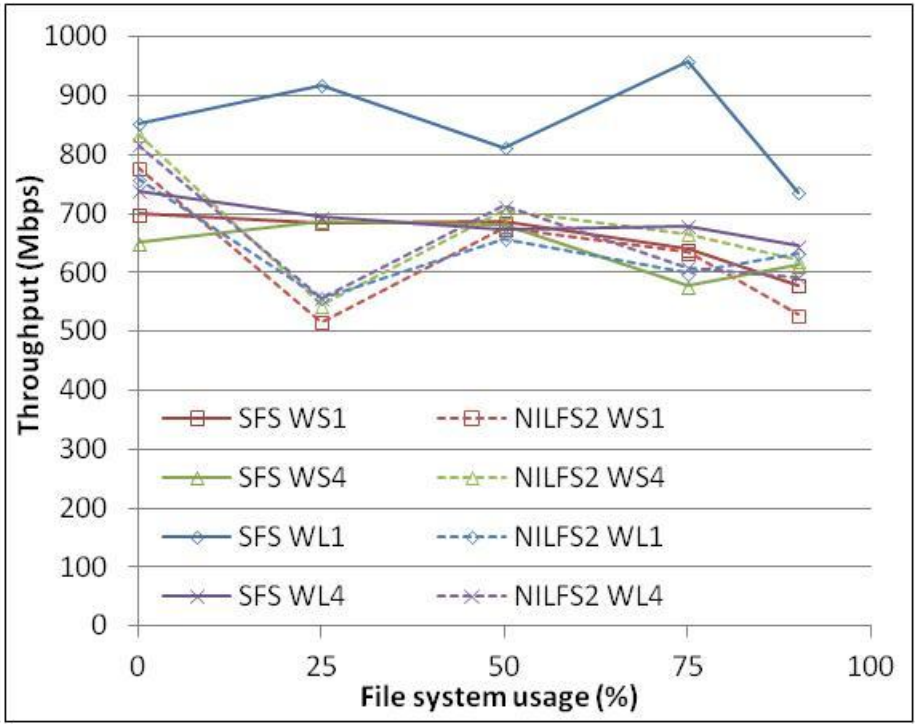


Evaluation Results: Large Files Aged FS

HITACHI
Inspire the Next

- **Write efficiency high and constant for SFS, degradation observed for NILFS2**
  - Almost no read/re-write overhead during data block update for SFS
  - Log region defragmentation still necessary for NILFS2, resulting in higher overhead for high disk usage
- **Disk throughput again comparable for both file systems**
  - Higher throughput achieved compared to small files fragmentation cases (less seek in average)





HGST
a Western Digital company

Standards...

What is it?

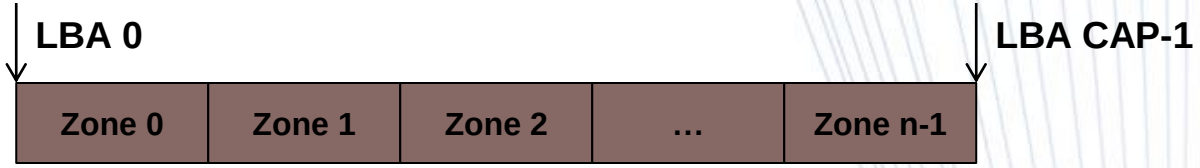
- **New SCSI command set – ZBC (Zoned Block device Command set)**
- **Standardized by T10 (the SCSI technical committee)**
- **Ideal for SMR drives**
- **Mostly SBC(-x) (DASD)**
  - New peripheral device type identifier
  - New profile of mandatory/optional commands
  - 2 new commands
- **Zoned LBA space**
  - LBAs ‘partitioned’ into non-overlapping zones
  - Several types of zones, each with their own characteristics
- **Sequential Write zones**
  - Some zone types must be written sequentially
  - Write pointer specifies LBA for next write

Architecture

- **Zones**
  - LBA space divided into non-overlapping zones
  - Each zone is a contiguous extent of LBAs
  - Each zone has
    - Zone type
    - Zone condition
    - Zone length (number of sectors)
    - Zone start LBA
    - Write pointer LBA (invalid for conventional zones)
- **Three zone types**
  - Conventional
  - Sequential write required
  - Sequential write preferred
- **Three device models**
  - Conventional (e.g. non-ZBC)
  - Host managed zoned block device
  - Host aware zoned block device

Zone Type Models

- **Zones are non-overlapping ranges of LBAs**
  - Zones are accessed using absolute LBAs



- **3 types of zones are defined: conventional zones, sequential write preferred zones and sequential write required zones**
  - Conventional zones do not have a write pointer and operations are performed as described in SBC-4
    - Operation within the zone similar to conventional disk
  - Sequential write preferred zone and sequential write required zones (referred to as write pointer zones) are associated with a write pointer indicating an LBA location within each zone
    - For sequential write preferred zone, the write pointer is a “hint” to indicate the best position for the next write operation
      - » Will function in legacy system
    - For sequential write required zones, writes can only be done at the write pointer position
      - » Will not function in legacy system

Model Characteristics

- **Each device model has various characteristics**
- **Different mix of zone types per model**

Characteristic	Conventional	Host Aware	Host Managed
Command Support	SBC-4	SBC-4	ZBC
PERIPHERAL DEVICE TYPE field value (see SPC-5)	00h	00h	14h
HAW_ZBC bit value (see SBC-4)	0b	1b	0b
Conventional zone	Mandatory	Optional	Optional
Sequential write preferred zone	Not supported	Mandatory	Not supported
Sequential write required zone	Not supported	Not supported	Mandatory
REPORT ZONES command	Not supported	Mandatory	Mandatory
RESET WRITE POINTER command	Not supported	Mandatory	Mandatory

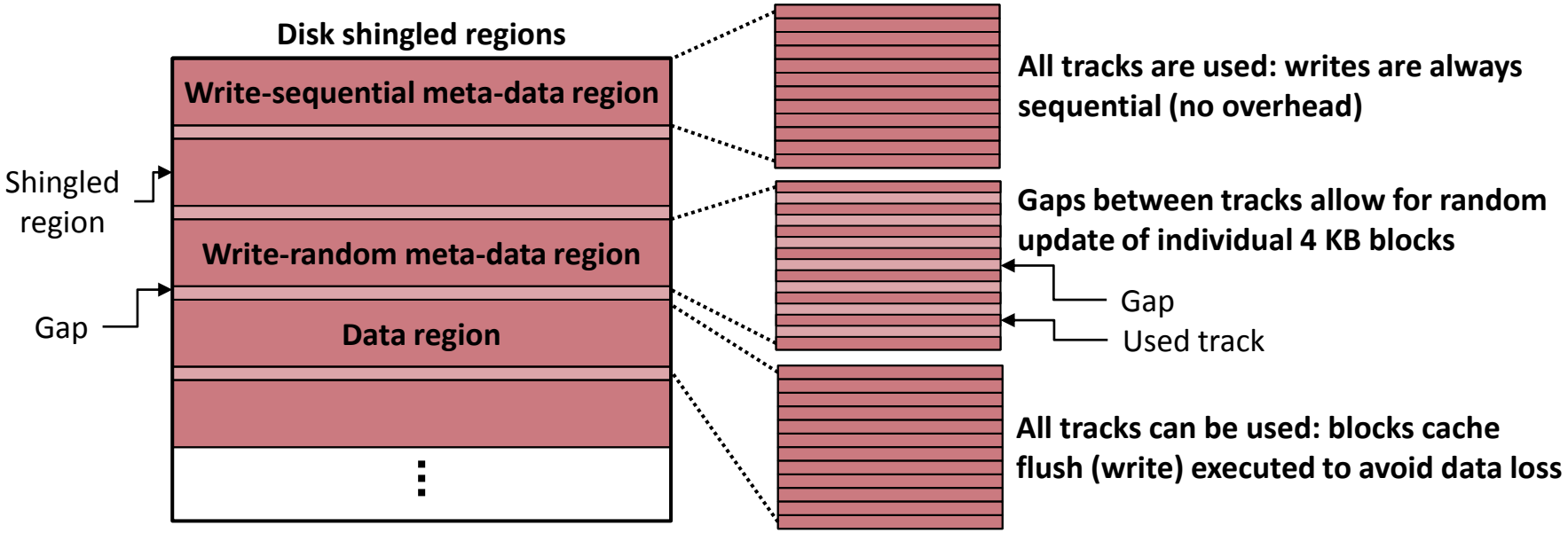
HGST
a Western Digital company

BACKUP

Shingled File System: Block Management

HITACHI
Inspire the Next

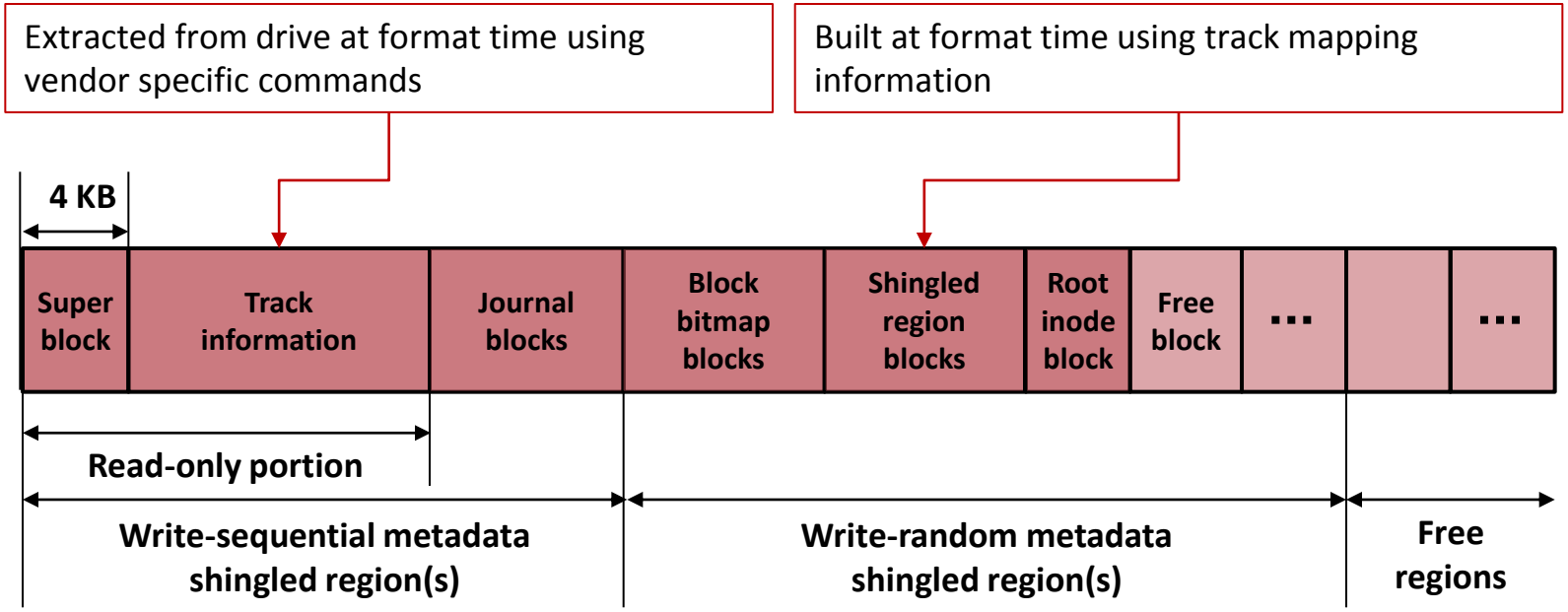
- **Disk blocks are managed through two different abstraction levels**
  - First level based on division of the disk into shingled regions separated by gaps (unused tracks)
  - Second level manages 4 KB blocks within shingled regions
- **Shingled region are assigned a type (dynamically) and used differently**
  - **Write-sequential meta-data region**: Store file system meta-data that can be written sequentially (i.e. read-only meta-data and journal blocks)
  - **Write-random meta-data region**: Store file system meta-data requiring random update
  - **Data region**: Store file data blocks



Shingled File System: Disk Format

HITACHI
Inspire the Next

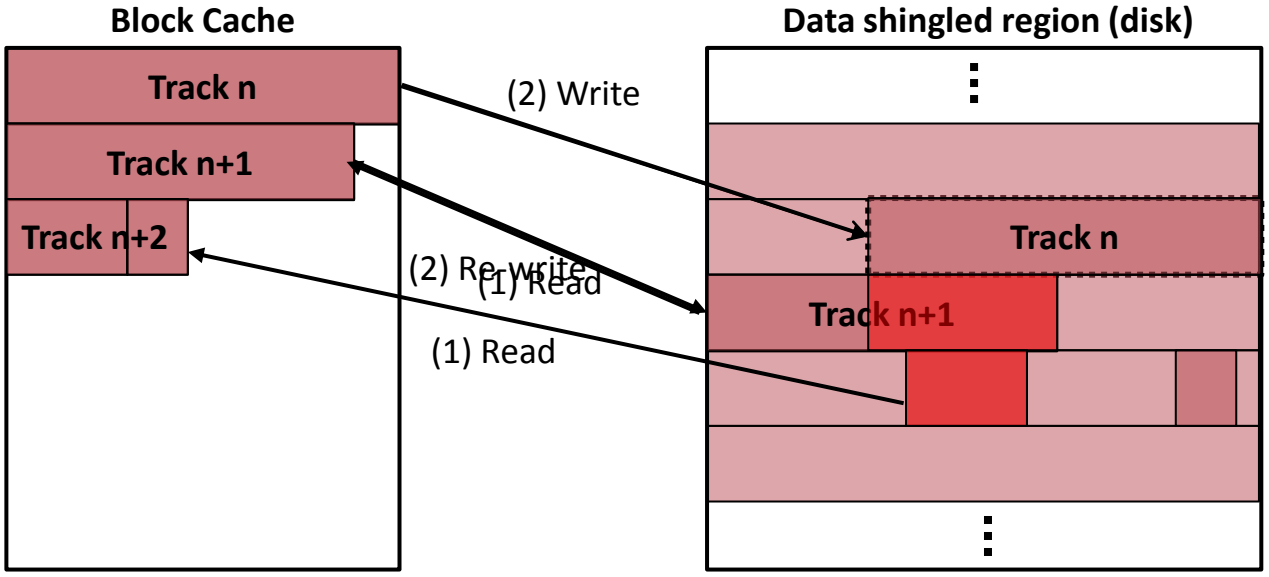
- **Format itself can be done only with sequential writes**
  - Can support “pure” shingled drives lacking a compatibility mode (internal processing of random write with respect to the shingle constraint)



Shingled File System: On-Disk Blocks Updates

HITACHI
Inspire the Next

- **Meta-data blocks**
  - Journal: Sequential write into write-sequential shingled region (no overhead)
  - In-place updates: Random writes into random-write shingled region (no overhead)
- **Data blocks**
  - (1) Starting from the first dirty block of a data region, read all blocks of the following track **IF** that track is allocated. Mark all blocks read as dirty (i.e. requiring on-disk update)
  - (2) Write back current track
  - (3) Loop until no more dirty blocks or last track of current region processed

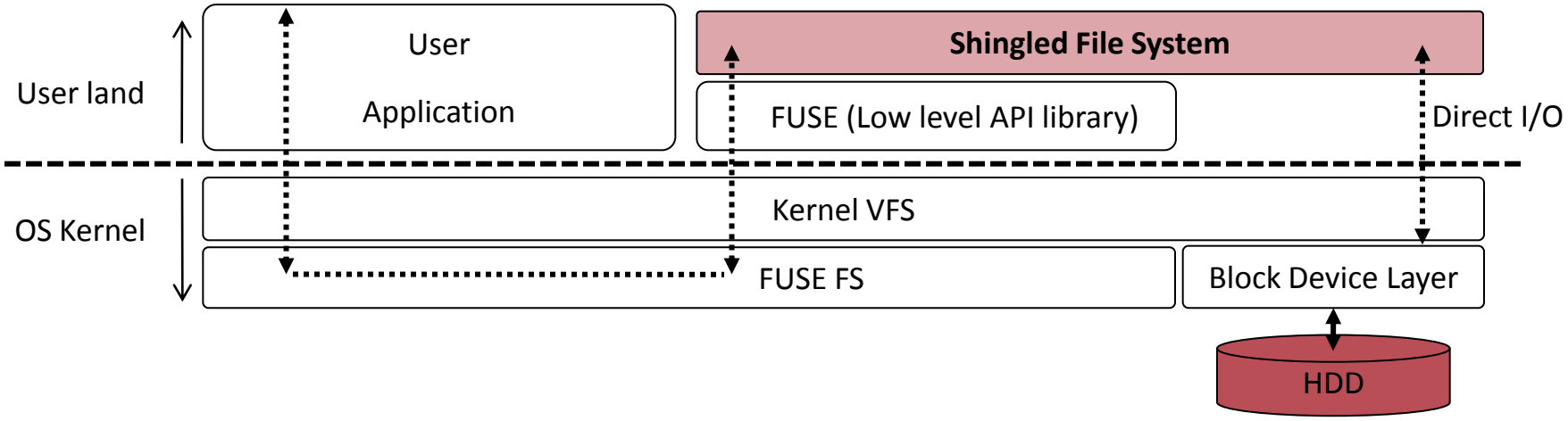


Overhead for data blocks update dependent on the allocation state (fragmentation) of data regions

Evaluation Results: Test Environment

HITACHI
Inspire the Next

- **Used prototype implementation of SFS based on Linux FUSE (File-system in User SpacE)**
  - Using low-level FUSE API and disk direct I/O operations to bypass all kernel level meta-data and data caching
  - Block cache size limited to 128 MB



- **Hardware**
  - Fast PC (8 CPU cores, 8 GB of RAM) to mitigate FUSE overhead (context switches and data copy)
  - 2 TB SATA 3.5” disk (2 platters, 4 heads, 7200 rpm, 32 MB buffer)
  - Disk shingling “assumed” with a shingle width of 2 tracks (writing one track overwrites the next track)

Evaluation Results: Protocol

HITACHI
Inspire the Next

- **Measurement performed for SFS prototype and NILFS2 log-structured file system**
  - NILFS2 is arguably the only file system that would allow using a SMR disk with very few modifications
- **To observed performance and efficiency of the file systems with different state, the file systems are first aged and measurements performed at different usage rate**
  - Aging creates and randomly deletes files repeatedly
  - Aging done in 2 cases: with small files (1 MB to 4 MB random size) and large files (10 GB to 20 GB random size)
  - Measurements done at 0 % (FS empty), 25 %, 50 %, 75 % and 90 % use of the FS capacity
  - For each measurement point, the following workloads are applied
    - WS1: 1 process writing small files (1 MB to 4 MB random size)
    - WS4: 4 processes writing small files (1 MB to 4 MB random size)
    - WL1: 1 process writing large files (10 GB to 20 GB random size)
    - WL4: 4 processes writing large files (10 GB to 20 GB random size)
- **For each measurement point, the write efficiency and application write throughput of the configurations are measured for each workload**
  - Write efficiency is defined as the ratio of the application data write throughput to the total disk throughput
  - A write efficiency of 1.0 thus means that no read/write overhead is observed (i.e. only application data is written to the disk).

T10 Administrivia

- **Draft Standard available – zbc-r01a**
  - Recently drafted (authorized at T10 plenary May 2014)
  - Combination of several proposals that have been discussed over the last year
  - http://www.t10.org/cgi-bin/ac.pl?t=f&f=zbc-r01a.pdf
- **Schedule**
  - Feature cutoff 2014 Jun
  - Letter Ballot 2015 Feb
  - For INCITS approval 2016 Jan
- **Development**
  - Main development done in T10
  - Sister effort in T13 – ZAC (Zoned ATA device Command set)
  - T10 and T13 meetings – monthly
  - Weekly telecons

ZBC Commands

- **Mandatory SPC/SBC commands**

Command	Description
INQUIRY	SPC-4
LOG SENSE	SPC-4
MODE SELECT (10)	SPC-4
MODE SENSE (10)	SPC-4
READ (16)	SBC-3
READ CAPACITY (16)	SBC-3
REPORT LUNS	SPC-4
REPORT SUPPORTED OPCODES	SPC-4
REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS	SPC-4
REQUEST SENSE	SPC-4
START STOP UNIT	SBC-3
SYNCHRONIZE CACHE (16)	SBC-3
TEST UNIT READY	SPC-4
WRITE (16)	SBC-3
WRITE SAME (16)	SBC-3

ZBC Commands

- **Mandatory ZBC defined commands**

Command	Description	Data
REPORT ZONES	Report the zone structure of the device (can specify subset of zones)	Zone descriptor for each zone (see below)
RESET WRITE POINTER	Move the write pointer to the start LBA of a write pointer zone	Zone start LBA

Zone descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved				ZONE TYPE			
1	ZONE CONDITION				Reserved			RESET
2	Reserved							
•••								
7								
8	(MSB)							
•••	ZONE LENGTH							
15								(LSB)
16	(MSB)							
•••	ZONE START LBA							
23								(LSB)
24	(MSB)							
•••	WRITE POINTER LBA							
31								(LSB)
32	Reserved							
•••								
63								

Zone Type Models

- **Allowed read/write operations depend on the target zone type**
  - Conditions not listed in the table below result in error
  - FORMAT UNIT operation operates as specified by SBC-4 but also resets the write pointer location of write pointer zones

Characteristic	Conventional Zone	Sequential Write Preferred Zone	Sequential Write Required Zone
Write pointer	None	Mandatory	
Write operation starting LBA	Anywhere within the zone	Anywhere, but preferably at write pointer	At write pointer LBA location only
Write operation ending LBA	Need not end in the same zone (write can span zones)		Within the zone
Write pointer location after write operation	N/A	Vendor specific	Write operation ending LBA + 1
Read operation starting LBA	Anywhere within the zone	Anywhere within the zone*	Before write pointer location
Read operation ending LBA	Need not end in the same zone (write can span zones)		Before write pointer location

*** The read operation returns the data written since the last zone reset operation, or an initialization pattern data if the accessed sectors have never been written since the last zone reset operation**

Backup

- **Additional detail suitable for more in-depth discussion**

Zone condition

Code	Name	Applies to zone type (see table 14)	Description
0h			Reserved
1h	EMPTY	sequential write preferred and sequential write required	The device server has not performed a write operation to this writer pointer zone since the write pointer was set to the lowest LBA of this zone. This zone is available to perform read operations and write operations.
2h	OPEN	sequential write preferred and sequential write required	The device server has attempted a write operation to this writer pointer zone since the write pointer was set to the lowest LBA of this zone and the zone condition is not FULL. This zone is available to perform read operations and write operations.
3h to Ch			Reserved
Dh	READ ONLY	all	Only read operations are allowed in this zone. The WRITE POINTER LBA field is invalid. The device server shall terminate any command that attempts a write operation in this zone with CHECK CONDITION status with the sense key set to DATA PROTECT and additional sense code set to ZONE IS READ ONLY.
Eh	FULL	sequential write preferred and sequential write required	All logical blocks in this writer pointer zone contain logical block data. The WRITE POINTER LBA field is invalid.
Fh	OFFLINE	all	Read commands and write commands shall be terminated as described in 4.4.3. The WRITE POINTER LBA field is invalid.