ZFS on SMR Drives

Enabling Shingled Magnetic Recording (SMR) for Enterprises

ROBERT E. NOVAK | Director of Systems Architecture

JUNE 2014

Agenda

- Data Integrity and Consistency
- Sources of Data Corruption/Loss
- ZFS Integrity
- ZFS Merkel Tree
- Storage Opportunity
- Aspects of Devices
- SSD & SMR Similarities
- Challenges for SMR

Data Integrity & Consistency

Every CIO's nightmare – corrupted/lost data

- Lessons from file systems
  - Unix File System – Learned to make copies of the superblock that was updated every two seconds – long road from dcheck, ncheck, icheck to fsck
  - Ext3/4 – Transaction logging to preserve updates but added RAID to increase performance and reliability
  - Why can't ext4 go beyond 16 TB?
- Disk Drives provide checksums to reduce drive data errors
- Hardware RAID deemed the only "safe" RAID
- Dependence on RAID leads to excessive amounts of storage AND silent corruption AKA bitrot – fueled by volatile write back caches and the "write hole"
- Ceph has added many features for "on the wire" and journal checksums (CRC32)[1]

[1] http://lists.ceph.com/pipermail/ceph-users-ceph.com/2014-January/007540.html

Sources of data corruption/loss

Don’t trust the hardware!

- The connection between the drive and the SAS/SATA buss
- The backplane in the JBOD
- The SAS/SATA cables between the JBOD and RAID/HBA controller
- The PCIe buss in the server

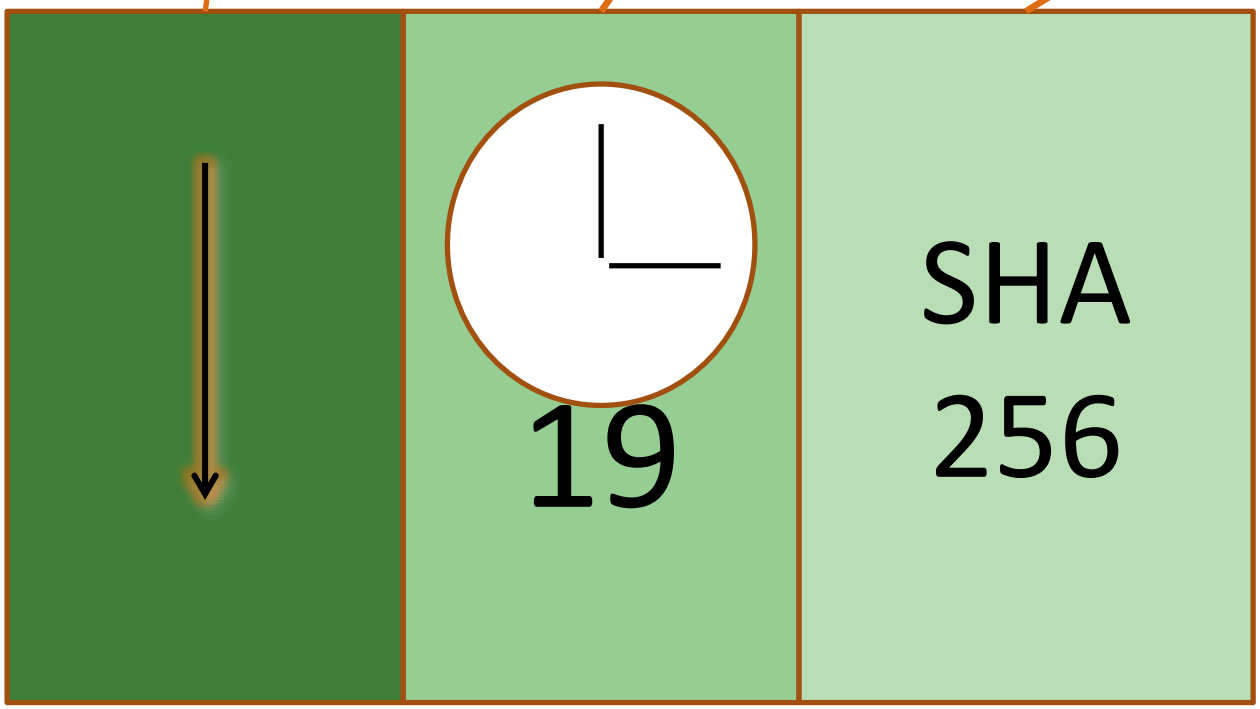
ZFS – Data integrity and consistency are paramount

- ZFS Intention Log
  - Not a write cache
  - Only read on reboot
  - Improves performance
  - Guarantees consistency
- In the presence of system failure, DATA WILL BE LOST
- Prevent data loss from corrupting integrity
- Don’t trust the hardware beyond CPU and Memory
- Use Cryptographic Hash – SHA256 to verify data
- Use mirroring/ZRAID to protect against corrupted data

ZFS Merkle Tree

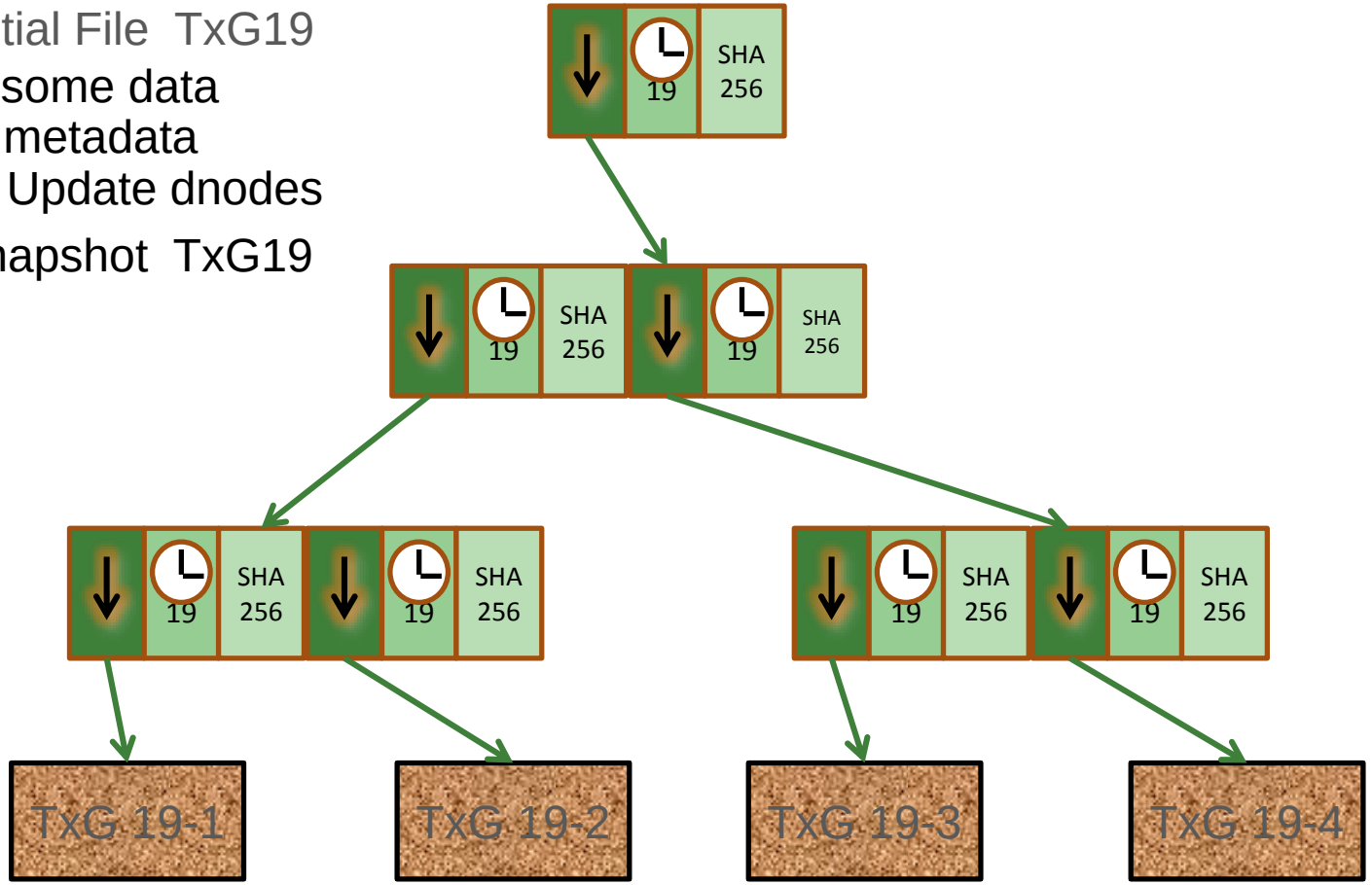
A pointer to a Data Block The Transaction Group Number A SHA256 Digest of the Data Block

A quick review of the important parts



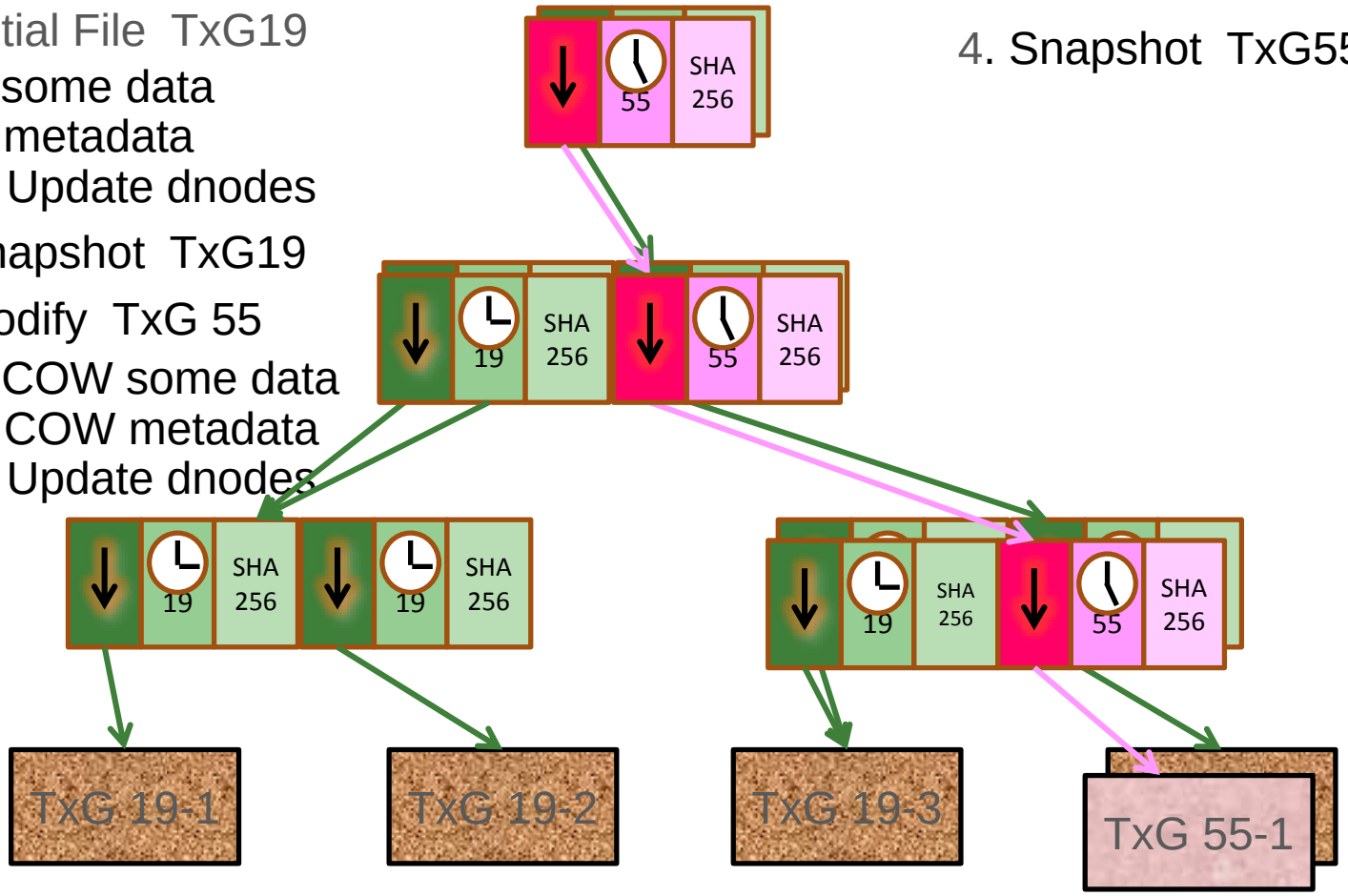
ZFS Copy on Write and Checksumming

1. Initial File TxG19
-- A) some data
-- B) metadata
-- C) Update dnodes
2. Snapshot TxG19



ZFS rebuild tree for new version

1. Initial File TxG19
-- A) some data
-- B) metadata
-- C) Update dnodes
2. Snapshot TxG19
3. Modify TxG 55
-- A) COW some data
-- B) COW metadata
-- C) Update dnodes
4. Snapshot TxG55

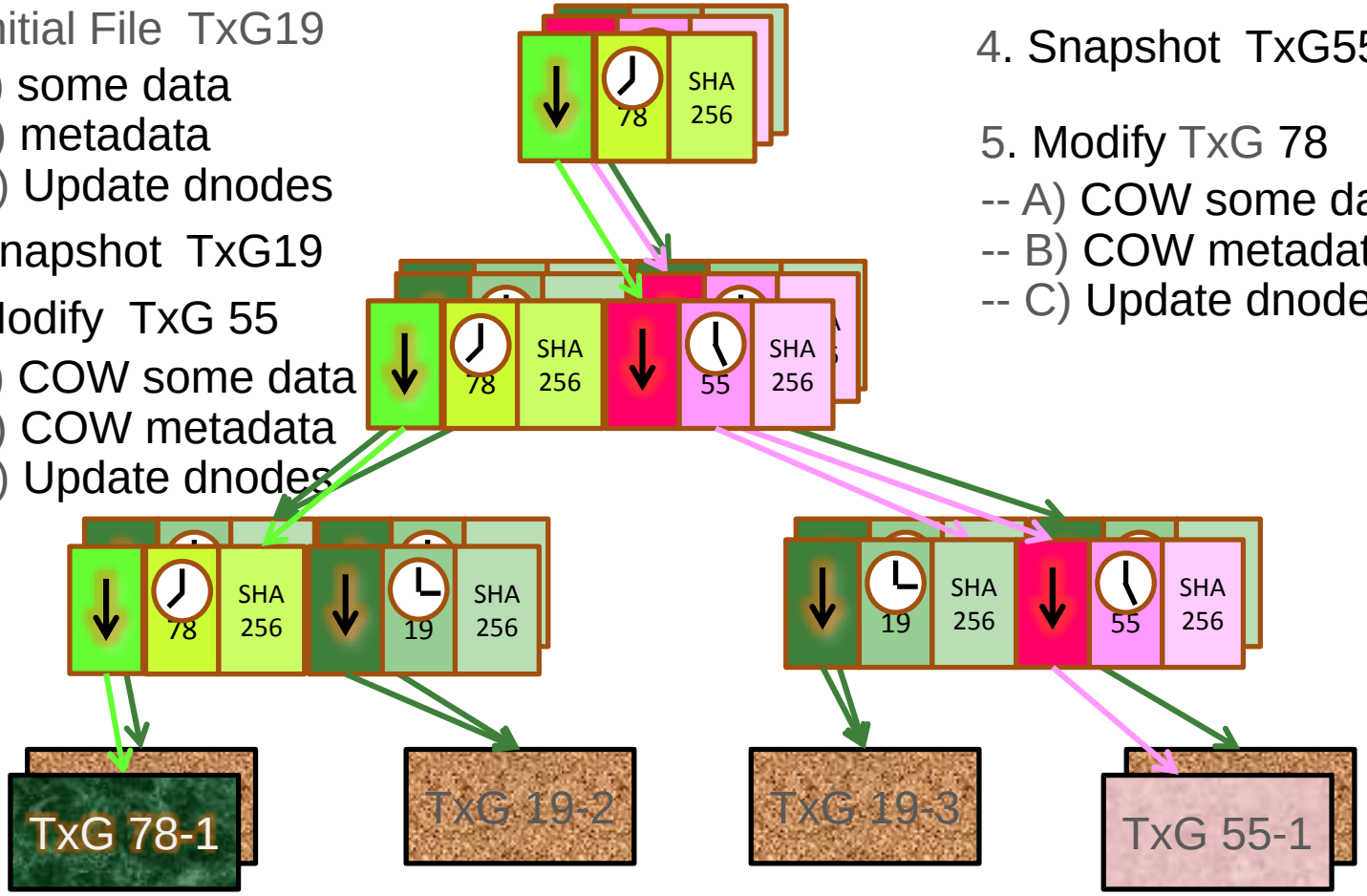


ZFS inherent deduplication in COW

1. Initial File TxG19
-- A) some data
-- B) metadata
-- C) Update dnodes

2. Snapshot TxG19

3. Modify TxG 55
-- A) COW some data
-- B) COW metadata
-- C) Update dnodes



4. Snapshot TxG55

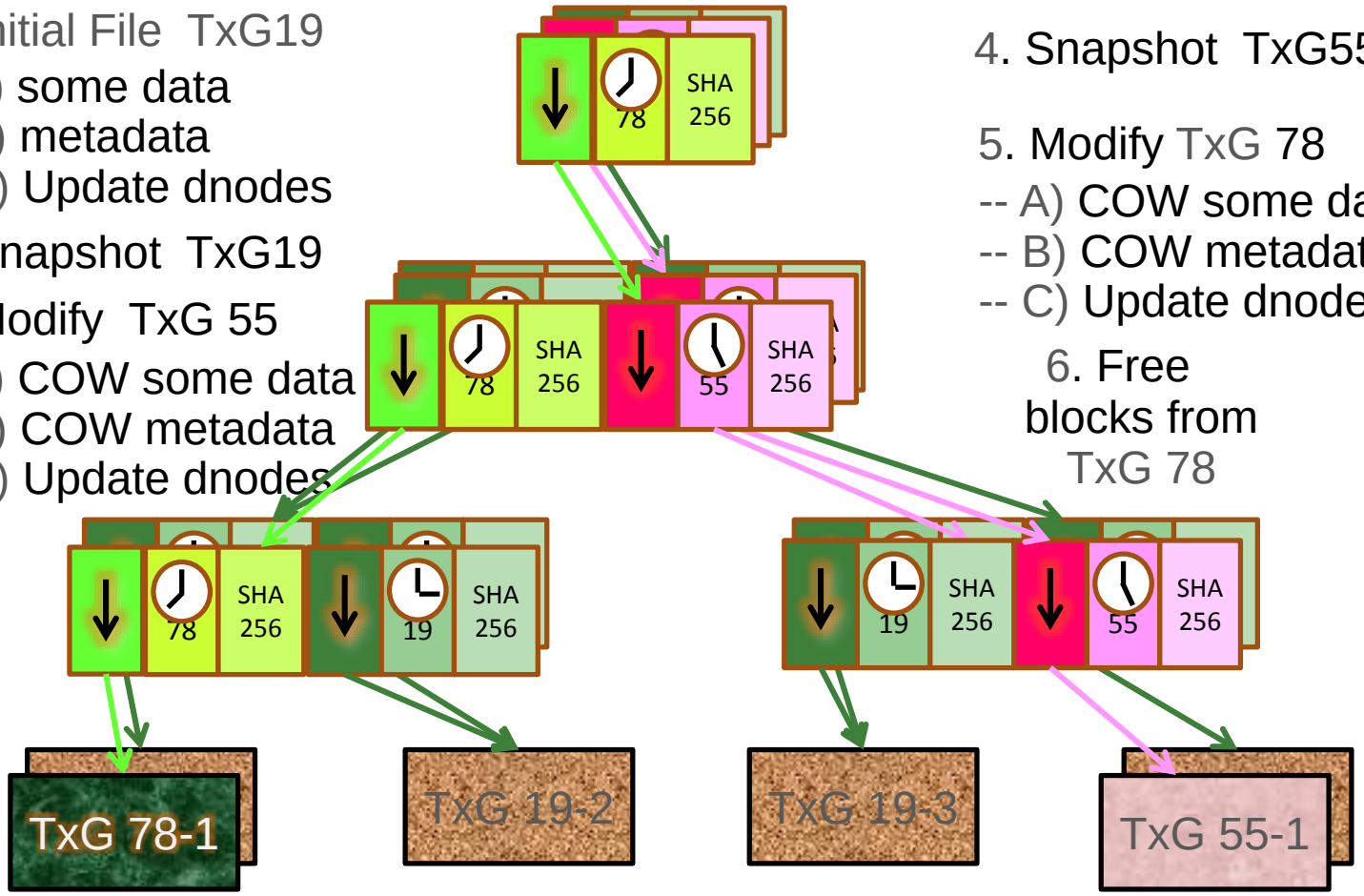
5. Modify TxG 78
-- A) COW some data
-- B) COW metadata
-- C) Update dnodes

ZFS Freedom to rollback versions/snapshots

1. Initial File TxG19
-- A) some data
-- B) metadata
-- C) Update dnodes

2. Snapshot TxG19

3. Modify TxG 55
-- A) COW some data
-- B) COW metadata
-- C) Update dnodes



4. Snapshot TxG55

5. Modify TxG 78
-- A) COW some data
-- B) COW metadata
-- C) Update dnodes

6. Free blocks from TxG 78

The Storage Opportunity

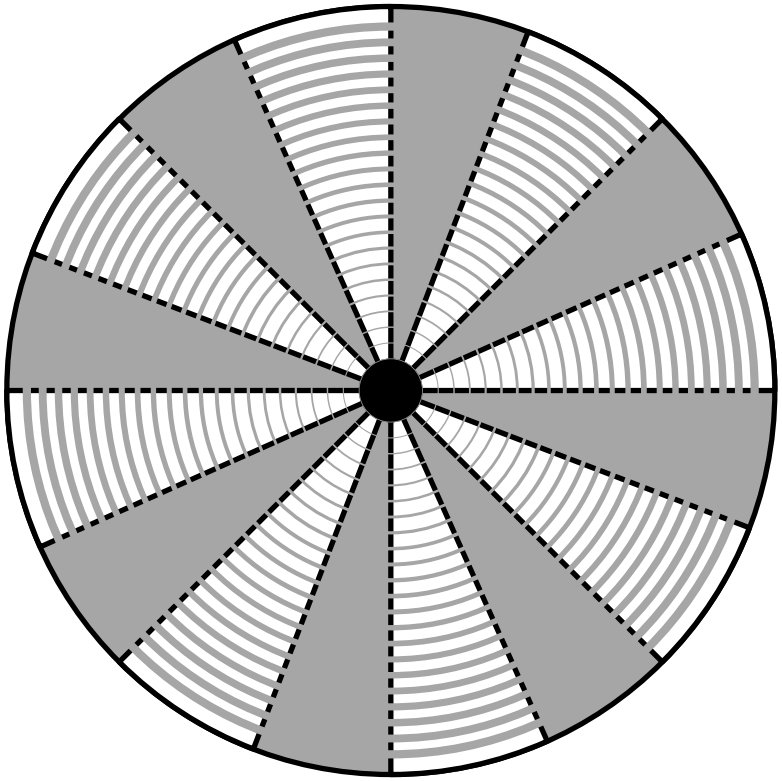
- Storage is growing at 2.7 zettabytes/year[1]
- HDD Shipments growth 30%/year[1]
- HDD Areal Density growth 20%/year[1]
- SMR improves areal density by 25%[1]
- **ZFS may be the only file system that can use SMR drives with NO LOSS in performance!**

[1] http://www.storagenewsletter.com/news/disk/seagate-shingled-magnetic-recording

Some Aspects of Devices

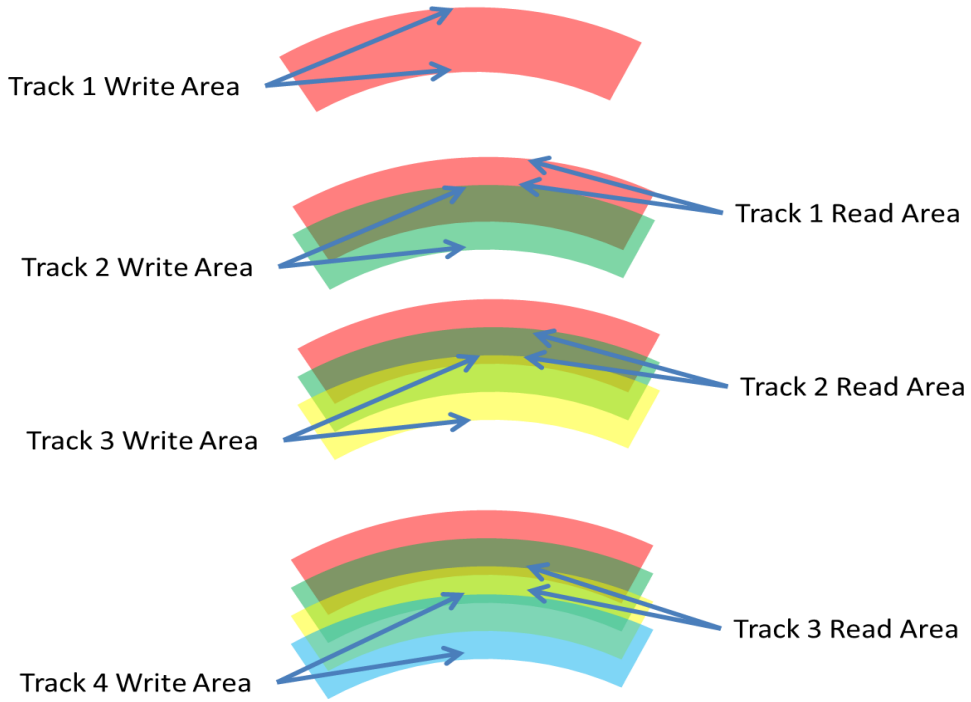
SMR DEVICES & FLASH DEVICES
HOW ARE THEY THE SAME?

Traditional Drives



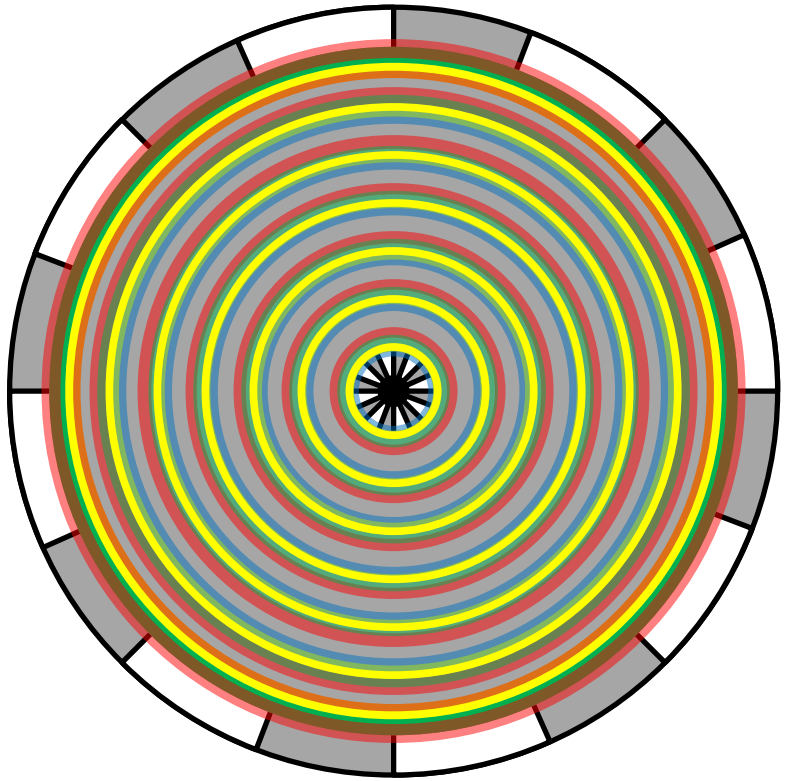
- Data areas – White Sectors
- Guarded by gaps – Gray
- Can Randomly Read/Write any data area

Laying Down Shingles (like on a roof)



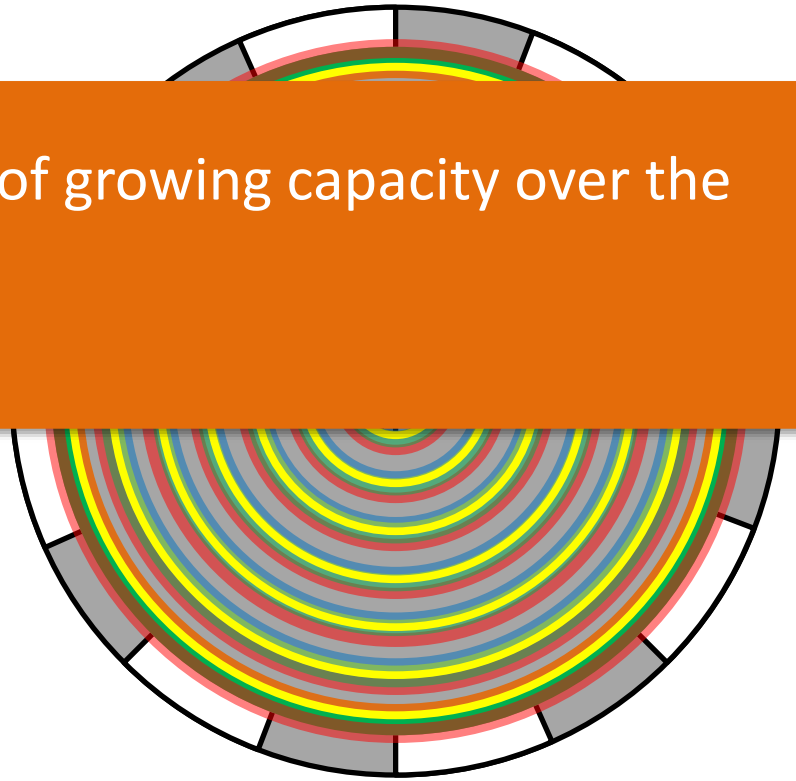
Shingled Magnetic Recording

- Shingles are grouped by Zone
- Copy from one Zone to another to perform consolidation and/or deletion
- Random I/O in outer Zone (OD) and inner Zone (ID)
- Perfect organization for ZFS Uberblocks



Shingled Magnetic Recording

- Shingles are grouped by Zone
- SMR will be the dominant method of growing capacity over the next 1-3 years.
- 20+TB drives are on the way.
- (OD) and inner Zone (ID)
- Perfect organization for ZFS Uberblocks



SSD vs. SMR

- The Flash Translation Layer hides the differences between Flash and HDDs
- FTL provides a remapping of failed cells in SSDs (overprovisioning)
- Erasure Blocks in SSDs are typically 128KB
  - Valid Data in an Erasure Block must be relocated to an alternate location before Erase
  - Uses the overprovisioning remapping mechanism
- These operations are transparent to Operating Systems and File Systems

Challenges for SMR

UNPREDICTABLE EMULATION OF RANDOM I/O
A NEW GARBAGE COLLECTOR
NEED FOR BP-REWRITE

Challenges of SMR Drives

- Writes must be in monotonically increasing order
- Large writes (~1 MB) give best performance
- Redirect on Write† (e.g. Tegile RoW) requires rewriting Volume Metadata
- Existing file systems (e.g., ext3, ext4, ntfs, etc.) don't work well
- WD has some success with LTFS* (tape emulation on disk) and NILFS‡
- Drive manufacturers have released consumer backup drives only

†http://publib.boulder.ibm.com/infocenter/ibmxiv/r2/index.jsp?topic=%2Fcom.ibm.help.xiv.doc%2FGen2%2Fxiv_po_ch04_05.html

*http://en.wikipedia.org/wiki/Linear_Tape_File_System
‡ http://nilfs.sourceforge.net/en/about_nilfs.html

Emulating Random I/O on SMR

- Emulation of Random I/O on SMR
  - Performance is unpredictable
  - Not ready for Enterprise
  - Firmware is complicated and difficult to do
- There are ANSI committees defining command sets for SMR (T10 & T13)

Three Versions of SMR Firmware

- Drive Managed Firmware
  - Drive will emulate a random I/O drive
  - Response times will be unpredictable
  - Challenging to develop & QA from a drive manufacturer perspective
- Cooperative Host/Drive Firmware
  - Drive presents accidental overwrite
  - Drive performs limited random I/O remapping
  - Less challenging than full random I/O
- Host Managed Firmware
  - All aspects of reads/writes are managed by Host
  - Random blocks available on I.D. and O.D. of drive
  - Drive prevents accidental overwrite
  - Simplest firmware

The Technical Opportunity

- ZFS Copy on Write technology theoretically performs all writes of user and metadata in locations that are monotonically increasing the disk address point.
- The only exceptions are the Random Update to the Uberblocks with the dnode data for the most recently updated transaction groups.

The Challenges

- ZFS *may* hiccup and perform some writes out of order.
- ZFS needs good sized random I/O areas at the beginning and the end of the drive (outermost diameter – O.D. and the innermost – I.D. tracks.
- ZFS may do other unscheduled writes in the middle of the drive.
- Garbage collection of unused blocks will require copying the “in use” blocks to other Zones of shingles. Then the “old” Zone can be free to use for new write operations

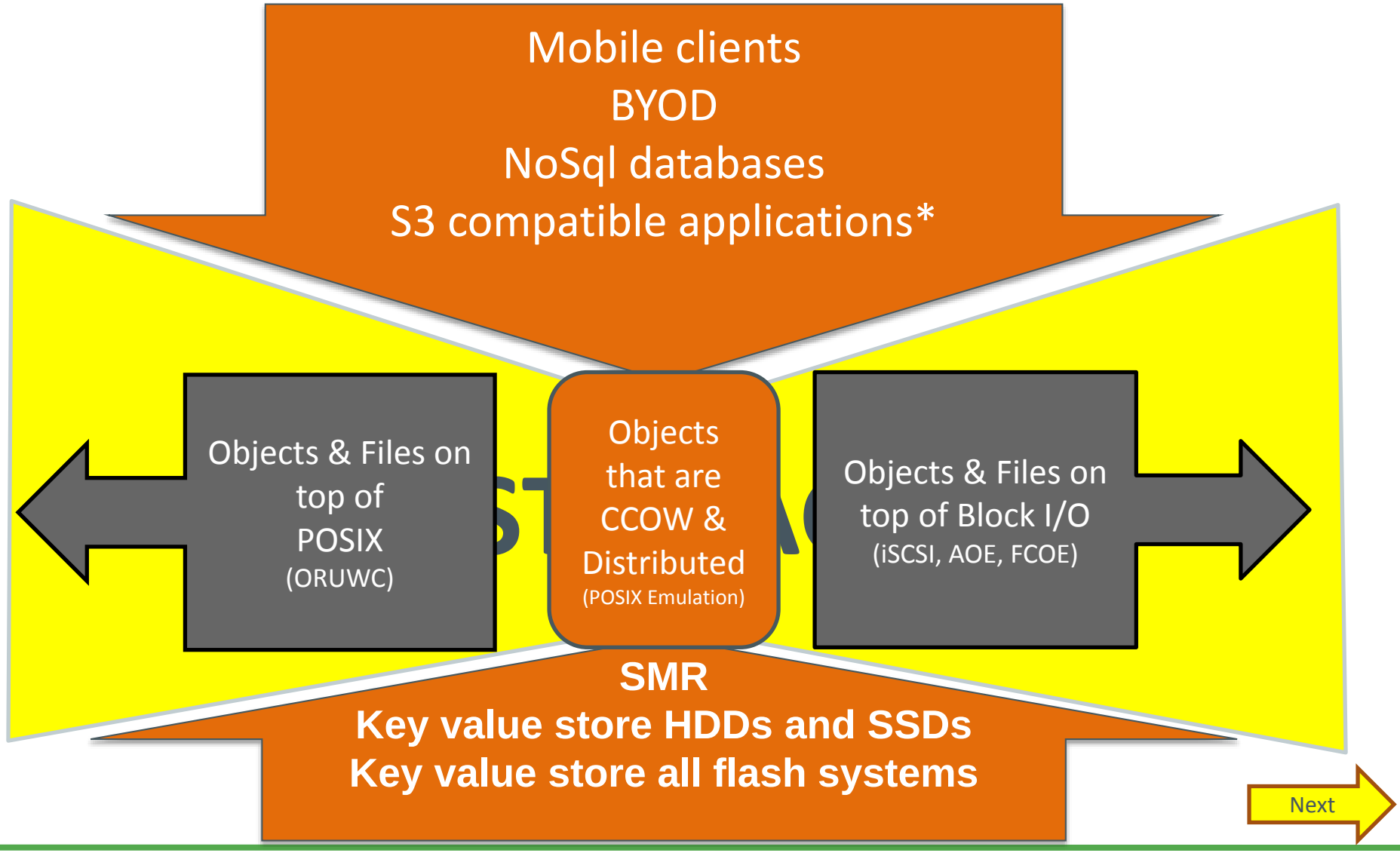
More Challenges

- Requires a new version of Garbage Collection
  - Blocks move to “Candidate to Free” on deletion/free
  - Zones with the highest “Candidate to Free” count are done first
  - “Live” (AKA in-use) blocks in these zones have to be copied/moved with BP-rewrite ← The Achilles heel!
  - After all of the live data is moved off a Zone, the entire Zone is “Free”
- Requires implementation of BP-Rewrite to ***remap blocks*** moving out of a Zone or shingle to other Zones on the drive
- Without BP-Rewrite:
  - No Snapshots
  - No Deduplication

Legacy Storage is Getting Squeezed

Next

Megatrend to Object Storage



Next