Improve Distributed Storage System Total Cost of Ownership with Host-Managed SMR HDDs

Albert Chen
KALISTA IO
Introduction

CEO of KALISTA IO. Previously, senior engineering and management roles at WDC, MSFT and various startups. Pioneered industry’s HM-SMR storage solutions.

hselin@kalista.io
https://linkedin.com/in/alberthchen
Preview: enabling HM-SMR everywhere

<table>
<thead>
<tr>
<th>Apache Hadoop®</th>
<th>Gitlab®</th>
</tr>
</thead>
<tbody>
<tr>
<td>NGINX®</td>
<td>Docker® registry</td>
</tr>
<tr>
<td>Ceph®</td>
<td>Media servers</td>
</tr>
<tr>
<td>MongoDB®</td>
<td>Minio®</td>
</tr>
<tr>
<td>Kubernetes® vols</td>
<td>and more...</td>
</tr>
</tbody>
</table>
Preview: without friction

No applications changes
No kernel modifications
Just works
Preview: consistent performance at scale

4.8x

lower latency at 99.99th percentile[^1][^4]

4KB write modifications
600,000 samples
Agenda

Trends

Problems and opportunities

Solutions

Host-Managed SMR

Current implementations and limitations

Improvements

KALISTA Phalanx

Performance and simplicity
Trends
Explosive growth of digital data

Amount of data created globally will increase from 32 zettabytes (ZB) last year to over 100 ZB by 2023[1]
Falling cost ($/GB)[2]
Pushing the limits of device physics

Storage devices are becoming more complex, difficult and costly to use.
New and expected usage models

Buyer beware—that 2TB-6TB “NAS” drive you’ve been eyeing might be SMR

Hard drives were already bad at random access I/O—but SMR disks are worse.

JIM SALTER - 4/17/2020, 3:45 AM
Increasing total capacity & device size[2]
Declining IO density
Limited margin for innovation[2]
“Hard disk is the worst form of storage device, except for all the others.”

Winston Leonard Spencer-Churchill
Demand for agility and optimal TCO

New architectures and usage models are growing increasingly incompatible & adverse for next generation storage technologies
IO Blender

VM 1
App A
OS

VM 2
App B
OS

VM 3
App C
OS

Sequential IO

VMM

Sequential IO

Sequential IO

2020 Storage Developer Conference. © KALISTA IO All Rights Reserved.
Long tail latency
Total cost of ownership
Current Solutions
Host Managed SMR

Higher capacity
Reduced total cost of ownership
Consistent performance
More restrictive usage model
Investment in storage stack
Layers of indirection

- Modified application
- Direct device access
- SMR file system
- Device mapper

Application
File system
HM-SMR
HM-SMR
HM-SMR
Available implementations

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG_IO</td>
<td>Direct access</td>
</tr>
<tr>
<td>libzbc</td>
<td>Direct access library</td>
</tr>
<tr>
<td>f2fs</td>
<td>SMR capable file system</td>
</tr>
<tr>
<td>dm-zoned</td>
<td>Device mapper target</td>
</tr>
</tbody>
</table>
Can we do better?

"Wisdom begins in wonder." — Socrates
Make room for innovation
Improve user experience
Minimize dependency and limitations

- Kernel version
- Modules/drivers
- Hardware configuration
- Protocol support
Leverage existing interfaces

File API
open(), read(), write()...

Object API
GET, PUT, DELETE

Block API
TUR, WRITE, READ
Work for all devices

Conventional device
- HDD
- SSD

Zoned devices
- HM/Hybrid-SMR HDD
- ZNS SSD
Deploy anywhere at anytime

Minimal dependencies
Easy to add & remove capacity
Fits within existing workflows
Works with orchestration fwks
Be device friendly

- Minimize seeks
- Maximize IO transfers
- Prevent hot spots
- Reduce background work
Perform at scale

Reduce contention
Increase IO concurrency
IO prioritization
Trim tail latency
Support new technologies

- Multi-actuator
- Variable capacity
- Large block size
- New usage models
KALISTA IO

Get ready for a storage revolution
PHALANX
STORAGE
SYSTEM
Adding performance and simplicity
## Performance, simplicity and future ready

<table>
<thead>
<tr>
<th>Phalanx</th>
<th>Future ready</th>
<th>Performance at scale</th>
<th>Just works</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Optimized for SMR, ZNS, EAMR</td>
<td>Scale performance with capacity</td>
<td>No application change required</td>
</tr>
<tr>
<td></td>
<td>Device friendly design</td>
<td>Minimize device contention</td>
<td>No kernel change required</td>
</tr>
<tr>
<td></td>
<td>Device aware data placement</td>
<td>Intelligent IO prioritization</td>
<td>Easy to deploy</td>
</tr>
<tr>
<td></td>
<td>Software-defined architecture</td>
<td>Eliminate hot write areas</td>
<td>Turnkey operation</td>
</tr>
</tbody>
</table>

---

2020 Storage Developer Conference. © KALISTA IO All Rights Reserved.
Simplifying data access and device management
Support existing interfaces & device types

User applications and access interfaces

- File interface: open(), read(), write() ...
- Object interface: GET, PUT, DELETE ...
- Block interface: INQUIRY, WRITE(10), READ(10) ...

Phalanx

- Data access
- IO engine
- Device management

Legend

- HM-SMR device
- ZNS device
- Hybrid-SMR device
- Multicloud
- CMR/SSD device
Reducing dependencies and adapting to variations
Engineered to minimize dependency

User space implementation

- No kernel modifications
- No additional modules/drivers
- Generalized for all kernel versions

Hardware

- No zone configuration requirements
- No device and zone size limitations
Know your dependencies

<table>
<thead>
<tr>
<th>Modified applications</th>
<th>Applications</th>
<th>File systems</th>
<th>Device mappers</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG_IO</td>
<td>SMR file systems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Linux kernel releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCSI generic device access /dev/sgx</td>
</tr>
<tr>
<td>Block device access /dev/sdx</td>
</tr>
<tr>
<td>Device mapper support dm-zoned</td>
</tr>
</tbody>
</table>

| 3.18 | 4.10 | 4.13 | 5.8  |
## Declare your independence

### Applications

<table>
<thead>
<tr>
<th>Phalanx</th>
</tr>
</thead>
</table>

### Linux kernel releases

<table>
<thead>
<tr>
<th>SCSI generic device access /dev/sgx</th>
<th>Block device access /dev/sdx</th>
<th>Device mapper support dm-zoned</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.18</td>
<td>4.10</td>
<td>4.13</td>
</tr>
<tr>
<td>4.10</td>
<td></td>
<td>5.8</td>
</tr>
</tbody>
</table>

2020 Storage Developer Conference. © KALISTA IO All Rights Reserved.
Designing for user experience
Deploy anywhere. Run everywhere.
Easy to deploy. Simple to operate.

1. Download image
   
   ```bash
   docker pull kalistaio/phalanx:release
   ```

2. And start container

   ```bash
   docker run \
   ...
   --mount type=bind,src=<mount path>. .\.
   kalistaio/phalanx:release
   ...
   -d <path to HM-SMR devices> \
   ...
   ```
What happens when you remove frictions and barriers to HM-SMR
Distributed systems with HM-SMR

Distributed storage systems

Apache Hadoop (no modification required)
Ceph (no modification required)

Data access IO engine
Device management

Kalista IO Phalanx Storage System

Conventional and host-managed SMR drives
And much more

NGINX®
GitLab®
MongoDB®
OpenStack Swift®
Docker® registry
Kubernetes® volumes
Minio®
Performing at scale
Designed for performance and scalability

Minimize contention
- Data/metadata separation
- Log structured data layout

Maximize IO concurrency
- Support multi-actuator disks
- Distribute workload across devices

Generate device friendly behavior
- Prevent hot spots
- Minimize background work
- Minimize seeks

Scale performance with capacity
- Row and column architecture
Minimize seeks and contention
Distribute workload across devices

- Reads → Writes → Dev A
- Reads → Writes → Dev B
- Reads → Dev A
- Read → Dev B
- Read → Dev A
- Reads → Writes → Dev C
Decrease contention

Decrease contention
Increase read concurrency

Write column
Scale performance with capacity

- Increase concurrency
- Decrease contention
- Increased capacity

Write column
Semantic intelligence

- Prioritization
- Tiering
- Caching
- Predictive optimization
- Quality of service (Qos)
What happens when you enable devices to perform at their best
Write tail latencies with legacy system\textsuperscript{[3]}
Curtailed with Phalanx and HM-SMR\textsuperscript{[4]}
Better percentile latencies (us)

<table>
<thead>
<tr>
<th>Percentile</th>
<th>Phalanx with Ultrastar HC620</th>
<th>Legacy stack with Ultrastar HC530</th>
</tr>
</thead>
<tbody>
<tr>
<td>99%</td>
<td>16,924</td>
<td>28,468</td>
</tr>
<tr>
<td>99.95%</td>
<td>26,211</td>
<td>97,371</td>
</tr>
<tr>
<td>99.99%</td>
<td>41,736</td>
<td>202,227</td>
</tr>
</tbody>
</table>
## Benchmark systems configuration

<table>
<thead>
<tr>
<th>Host-Managed SMR HDD Test System</th>
<th>CMR HDD Test System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark application (e.g. fio/Hadoop/Ceph)</td>
<td>Benchmark application (e.g. fio/Hadoop/Ceph)</td>
</tr>
<tr>
<td>Kalista IO Phalanx storage system</td>
<td>XFS/ext4</td>
</tr>
<tr>
<td>Linux 5.0.0-25-generic kernel</td>
<td>Linux 5.0.0-25-generic kernel</td>
</tr>
<tr>
<td>Western Digital Ultrastar DC HC620 Host-Managed SMR HDD</td>
<td>Western Digital Ultrastar DC HC530 CMR HDD</td>
</tr>
</tbody>
</table>
Benchmark results

16x
more IOPS
with fio random write\textsuperscript{[5]}

19%
faster throughput
with Hadoop TestDFSIO read\textsuperscript{[6]}

58%
higher IOPS
with Ceph Rados write bench\textsuperscript{[7]}

10x
better performance consistency
with Ceph Rados write bench\textsuperscript{[7]}
Thank you!
Contact

http://www.kalista.io
@kalista.io
hselin@kalista.io

"There is nothing impossible to him who will try." – Alexander
References


2. Source: Seagate Technology LLC and Western Digital Corp quarterly reports

3. Testing conducted by Kalista IO in July 2020 using XFS file system with Linux kernel 5.4.0-42-generic, and Intel® Core™ i7-4771 CPU 3.50GHz with 16GiB DDR3 Synchronous 2400 MHz RAM, and Western Digital Ultrastar DC HC530 CMR drive connected through SATA 3.2, 6.0 Gb/s interface. Write bench created a single 1GB file and executed 600,000 write commands each overwriting the first 64KB region of the file to capture latency values.

4. Testing conducted by Kalista IO in July 2020 using preproduction Olympus (Phalanx) software with Linux kernel 5.4.0-42-generic, and Intel® Core™ i7-4771 CPU 3.50GHz with 16GiB DDR3 Synchronous 2400 MHz RAM, and Western Digital Ultrastar DC HC620 host managed SMR drives connected through SATA 3.2, 6.0 Gb/s interface. Write bench created a single 1GB file and executed 600,000 write commands each overwriting the first 64KB region of the file capture latency values.
References

5. Testing conducted by Kalista IO in August 2019 using preproduction Phalanx software with Linux kernel 4.18.0-25-generic, and Intel Core i7-4771 CPU 3.50GHz with 16GiB DDR3 Synchronous 2400 MHz RAM, and Western Digital Ultrastar DC HC620 host managed SMR and Ultrastar DC HC530 CMR drives connected through SATA 3.2, 6.0 Gb/s interface. Tested with Flexible I/O tester (fio) version 3.14-11-g308a. Random write bench ran for 1800 seconds with 4KB block and 200GB file size, 64 concurrent threads each with queue depth of 1. Executed 3 times to capture average and standard deviation IOPS values.

6. Testing conducted by Kalista IO in August 2019 using preproduction Phalanx software with Linux kernel 5.0.0-25-generic, and Intel® Core™ i7-4771 CPU 3.50GHz with 16GiB DDR3 Synchronous 2400 MHz RAM, and Western Digital Ultrastar DC HC620 host managed SMR and Ultrastar DC HC530 CMR drives connected through SATA 3.2, 6.0 Gb/s interface. Tested with Apache Hadoop version 3.2.0 in single node pseudodistributed mode with single block replica, and TestDFSIO version 1.8 on OpenJDK version 1.8.0_222. TestDFSIO read benchmark ran with 32 files, 16GB each for a 512GB dataset. Executed 3 times to capture average and standard deviation throughput values.
References

7. Testing conducted by Kalista IO in August 2019 using preproduction Phalanx software with Linux kernel 5.0.0-25-generic, and Intel Core i7-4771 CPU 3.50GHz with 16GiB DDR3 Synchronous 2400 MHz RAM, and Western Digital Ultrastar DC HC620 host managed SMR and Ultrastar DC HC530 CMR drives connected through SATA 3.2, 6.0 Gb/s interface. Tested with Ceph version 13.2.6 Mimic in single node mode with single object replica. Rados write bench ran with 4MB object and block (op) size with 16 concurrent operations for 1800 seconds to capture average and standard deviation IOPS values.
Additional information
Additional information

1. Western Digital Ultrastar DC HC600 SMR Series HDD

2. KALISTA IO and Western Digital joint solution brief:
   Distributed Storage System with Host-Managed SMR HDDs

3. Addressing Shingled Magnetic Recording drives with Linear Tape File System
   https://www.snia.org/sites/default/files/files2/files2/SDC2013/presentations/Hardware/AlbertChenMalina_Addressing_Shingled_Magnetic_Recording.pdf

4. Host Managed SMR
   https://www.snia.org/sites/default/files/SDC15_presentations/smr/AlbertChen_JimMalina_Host_Managed_SMR_revision5.pdf
Additional information

5. Linux SCSI Generic (sg) driver

6. libzbc
   https://github.com/hgst/libzbc

7. dm-zoned

8. Flash-Friendly File System (F2FS)
   https://www.kernel.org/doc/Documentation/filesystems/f2fs.txt

9. Zoned storage
   https://zonedstorage.io

10. Linux kernel changes
    https://kernelnewbies.org/LinuxVersions
Additional information

11. Another Layer of Indirection
https://www.linkedin.com/pulse/another-layer-indirection-albert-chen/


13. Phalanx Flexible I/O tester (fio) benchmarks

14. Phalanx Hadoop TestDFSIO benchmarks

15. Phalanx Ceph OSD and Rados benchmarks
Attributions
Attributions

1. Icons from Font Awesome. License available at https://fontawesome.com/license
   No modifications made.
Please take a moment to rate this session.

Your feedback matters to us.