Performance of Apache Ozone on NVMe

Wei-Chiu Chuang (jojochuang)
Ritesh Shukla (kerneltim)
Agenda

• Overview of how Ozone and how it scales
• Why NVME is important for Ozone for scaling
• Benefits of using NVME
• Impala performance results from NVME clusters
• Write path improvements results from NVME clusters
• Summary
• Questions
Ozone Architecture

- Users
  - HTTP Load Balancer
  - S3 Gateway
    - Ozone Filesystem Connector
    - CLI (shell)
    - Ozone Manager
    - Storage Container Manager
    - DataNodes
    - Ozone Console
    - Recon Server
Why does Ozone Scale? Separation of concerns

- Ozone Client
- Ozone Manager
- Recon
- Datanode
- Storage Container Manager

- Scale out
- No foreground load
- No background load
Why does Ozone Scale? Aggregation via containers

- User
  - Storage Container Manager
  - Blocks
    - Chunks
  - Containers
    - Blocks
  - Volumes
    - Containers
    - Volumes
    - Datanodes

Unit of Recovery

Unit of Failure
Why does foreground Scale?

- No heap limitations, working set can be cached in memory and unused data can be destaged to disk
- OM uses NVME to store RocksDBs
- Future projects such as Snapshots leverage RocksDB to preserve simplicity of IO path.
Ozone scales!
Does background scale up and scale out?

• Datanode count can scale beyond HDFS
  • No memory pressure on OM due to block reports/object counts/heap limitations
• Container abstraction allows scaling of Datanodes and any background processing.
• Much higher density per Datanode than HDFS
Datanode scales out and scale up

- Testbed used:
  - ~400 TB/Datanode
  - Tested with 200k containers per datanode => 1 PB per datanode.

- Cisco UCS M6
  - Capacity node: 256 TB per datanode

- Cisco UCS S3260
  - Extreme Capacity: 384 TB per datanode
## Ozone vs. HDFS

<table>
<thead>
<tr>
<th>Capability</th>
<th>Ozone</th>
<th>HDFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage Density</td>
<td>1000’s of nodes at 600TB per node</td>
<td>1000’s of nodes at 100TB per node</td>
</tr>
<tr>
<td>Scalability</td>
<td>10B Objects</td>
<td>400M Objects</td>
</tr>
<tr>
<td>Recovery</td>
<td>Fast recovery ( &lt; 5 min restart)</td>
<td>Slow startup based on size</td>
</tr>
<tr>
<td>High Availability</td>
<td>Active - Active</td>
<td>Active - Standby</td>
</tr>
<tr>
<td>Protocol Support</td>
<td>Hadoop / S3 API</td>
<td>Hadoop API</td>
</tr>
</tbody>
</table>
Small objects are welcome

Max performance reached at object size around 10-20 MB
Hardware trends

• Cloudera recommends Ozone’s metadata reside on NVME
• Not just metadata increasing number of customers using all NVME clusters for Ozone
• Ozone certified against Cisco all NVME data intelligence platform
• Customers see TCO benefits with all NVME clusters
Why NVME

- Enables destaging of data with minimal impact to performance.
  - Long tail latency is a small percentage of the overall latency
- Vendors increasing shipping configurations with NVME
- Bet in the right direction of hardware trends.
- Low latency metadata can stay on NVME
  - Data at scale can be on spindles.
## Disk characteristics (rule of thumb)

<table>
<thead>
<tr>
<th></th>
<th>HDD</th>
<th>(SATA) SSD</th>
<th>NVMe SSD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transfer rate</strong></td>
<td>Typically 100 MB/s - 200MB/s Up to ~500MB/s</td>
<td>Typically 400 MB/s - 550MB/s Up to 600MB/s</td>
<td>Typically 3,000 MB/s-5,000 MB/s Up to 7,000 MB/s</td>
</tr>
<tr>
<td><strong>Latency (4kb)</strong></td>
<td>~10 ms</td>
<td>~200 us</td>
<td>~60 us</td>
</tr>
<tr>
<td><strong>Size</strong></td>
<td>1TB - 16TB each Up to 20TB</td>
<td>500GB - 4TB each Up to 15TB</td>
<td>500GB - 4TB each Up to 15TB</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td>Low</td>
<td>High</td>
<td>Somewhat same as SATA SSD</td>
</tr>
</tbody>
</table>
## Testbed

3 x master nodes, 16 x DataNodes

### Master nodes

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2 x Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz/20 cores</td>
</tr>
<tr>
<td>memory</td>
<td>384GB ( 12 x 32GB DDR4 @ 2933MHz)</td>
</tr>
<tr>
<td>OS Boot</td>
<td>Cisco Boot optimized M.2 Raid controller with 2 x 240GB SATA SSD</td>
</tr>
<tr>
<td>SSD</td>
<td>3.8TB SATA SSD Enterprise Value</td>
</tr>
<tr>
<td>Storage Controller</td>
<td>Cisco 12G Modular Raid Controller with 2GB cache</td>
</tr>
<tr>
<td>Network Adapter</td>
<td>Cisco UCS VIC 1387 2 x 40Gbps ports x8 PCIe Gen3</td>
</tr>
</tbody>
</table>

### Data Nodes

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2 x Intel(R) Xeon(R) Gold 6262V CPU @ 1.90GHz/24 cores</td>
</tr>
<tr>
<td>memory</td>
<td>384GB ( 12 x 32GB DDR4 @ 2933MHz)</td>
</tr>
<tr>
<td>OS Boot</td>
<td>Cisco Boot optimized M.2 Raid controller with 2 x 240GB SATA SSD</td>
</tr>
<tr>
<td>NVMe</td>
<td>10 x 8TB Intel P4510 U.2 High Performance Value</td>
</tr>
<tr>
<td>Storage Controller</td>
<td>NA</td>
</tr>
<tr>
<td>Network Adapter</td>
<td>Cisco UCS VIC 1387 2 x 40Gbps ports x8 PCIe Gen3</td>
</tr>
</tbody>
</table>
Tests conducted

• Freon read load post hard restart (minimal caching)
• Warp test to measure network saturation when using S3
• Impala TPCDS benchmark
• Ratis streaming performance tests
How much does disk read cost with NVME?
Why Impala and Ozone?

- Data Warehouse is the most common use case. ($$$)
- Impala historically optimized on HDFS -> what will it do on Ozone
Software under test

CDP Private Cloud Base 7.1.8 +

- IMPALA-11457 Fix regression with unknown disk id
- HDDS-4970 Significant overhead when DataNode is over-subscribed
- HDDS-7135 ofs file input stream should support StreamCapabilities interface
- HDDS-7136 Memory leak due to ChunkInputStream.close() not releasing buffer
- HDDS-7161 Make Checksum.int2ByteString() zero-copy

All fixes are upstreamed in Apache Ozone 1.3.0 + Apache Impala 4.1.1
Ozone has a small overhead compared to HDFS (13% more than HDFS, and 12% more than remote HDFS).

SCR = short-circuit read
remote = REPLICA_PREFERENCE=REMOTE

Impala TPC-DS 3TB execution time

- HDFS (w/ SCR)
- HDFS (w/o SCR)
- HDFS (remote)
- Ozone
Ozone has a small overhead compared to HDFS (5% more than HDFS).

SCR = short-circuit read
remote = REPLICA_PREFERENCE=REMOTE
Lesson Learned
Too many rocksdb instances is bad

One RocksDB to manage the metadata of a 5GB container
But a DataNode can be up to a few hundred TB → 100k rocksdb instances.

Very slow to load (HDDS-3892, HDDS-4427, HDDS-4488, HDDS-5785)

Error prone (HDDS-5756/rocksdb issue:8617)

→ HDDS-3630 (Merge rocksdb in datanode)
  - One rocksdb instance to manage the containers of a disk
Write path improvements in the pipeline with Ratis Streaming API (RATIS-979)

- The Leader does not get more traffic
  - It is no longer the performance bottleneck.

- Better network topology awareness
  - Client writes to the closest datanode instead of the Leader

- Netty zero buffer copy
  - No gRPC buffer problem
Ratis streaming
Benchmark – Observation

V1 (Async API) vs V2 (Streaming API)

- **V2 Streaming** multiple-client cases can be ~3x of V1 Async
  - Streaming can use the full power of all **three** datanodes.
Performance roadmap ahead

1. Ratis streaming merge
2. OM Performance improvements
3. DN saturation of network
4. Better leveraging benefits of NVME
   a. Squeezing every bit of latency from each request processing
   b. Better caching architectures from computation down to disk to leverage HW.
Conclusion

- Ozone architecturally addresses scale issues
- Hardware trends in the right direction for Ozone architecture.
  - NVME for Ozone Manager
  - High density datanodes with higher node counts
- Tests validate the architecture and direction for Ozone.
Acknowledgement

Cisco

Apache Ozone and Ratis communities
The unexpected: JDK performance problems

JDK lock contention [JDK-7092821](https://bugs.openjdk.java.net/browse/JDK-7092821) (resolved in JDK 8u241 and 11u07)

Token verification (SHA256withRSA) [HDDS-7256](https://issues.apache.org/jira/browse/HDDS-7256)
Contributions welcome!

github.com/apache/ozone/

Questions?