Apache Ozone - State of the Union

Dinesh Chitlangia, Manager @ Cloudera

Aravindan Vijayan, Staff Engineer @ Cloudera

September 23, 2021

Speakers

- Dinesh Chitlangia
  - Manager @ Cloudera, Apache Ozone PMC/Committer, Apache Hadoop Committer
  - LinkedIn - Dinesh Chitlangia
  - Github - dineshchitlangia
  - Twitter - dineshneo

- Aravindan Vijayan
  - Staff Engineer @ Cloudera, Apache Ozone, Hadoop, Ambari PMC/Committer, Apache Ratis Committer
  - Github - avijayanhwx

- Brief History and Overview of Apache Ozone
- Current Status of the project
- Highlights and Roadmap

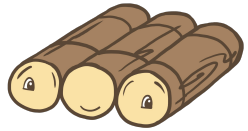
Brief History & Overview of Apache Ozone

- To address scalability limits of HDFS
- Started as a sub-project under HDFS and spun out as a Top Level project after 4 alpha and 1 beta release
- What is Apache Ozone ?
  - Raft based, scalable Object Store
  - Decoupled Namespace and Block Space
  - Metadata stored in high performance RocksDB thus relying on off-heap memory
  - Security is built-in
  - Built by Apache Hadoop community
  - Strengths of HDFS are retained, limitations addressed.

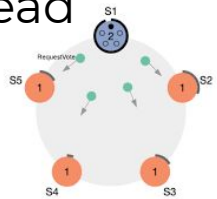
Building Blocks of Ozone

- Ozone separates namespace management and block space management
- Namespace managed by Ozone Manager (OM) daemon.
  - Keeps only the working set in memory.
- The block space is managed by Storage Container Manager (SCM).
  - Scales by not tracking individual data blocks. Instead it tracks *containers** which are aggregations of blocks. Typical container size is 5GB.
- By scaling namespace and block management independently, Ozone can scale to billions of files in a cluster.
- ** No relation to Linux or YARN containers.*

‘Raft’ Consensus

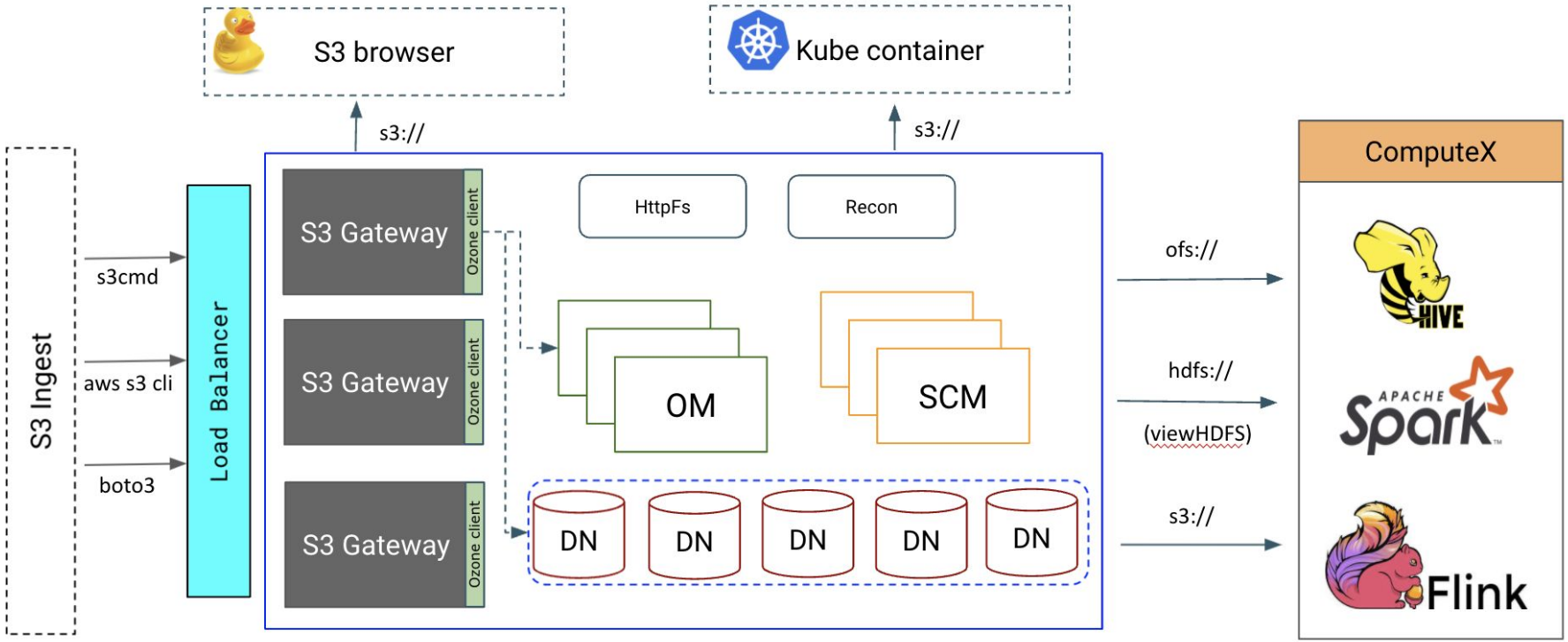


- ‘Raft’ protocol is used in replication flow of data write path (Datanode) as well as in metadata path (OM, SCM)
- Uses Apache Ratis library, a Java based implementation of Raft
  - Includes standard raft asks like leader election, write ahead log replication, snapshots, log compaction etc.
  - In addition, Ratis provides pluggable log, state machine (**SM**) & RPC layer implementations.

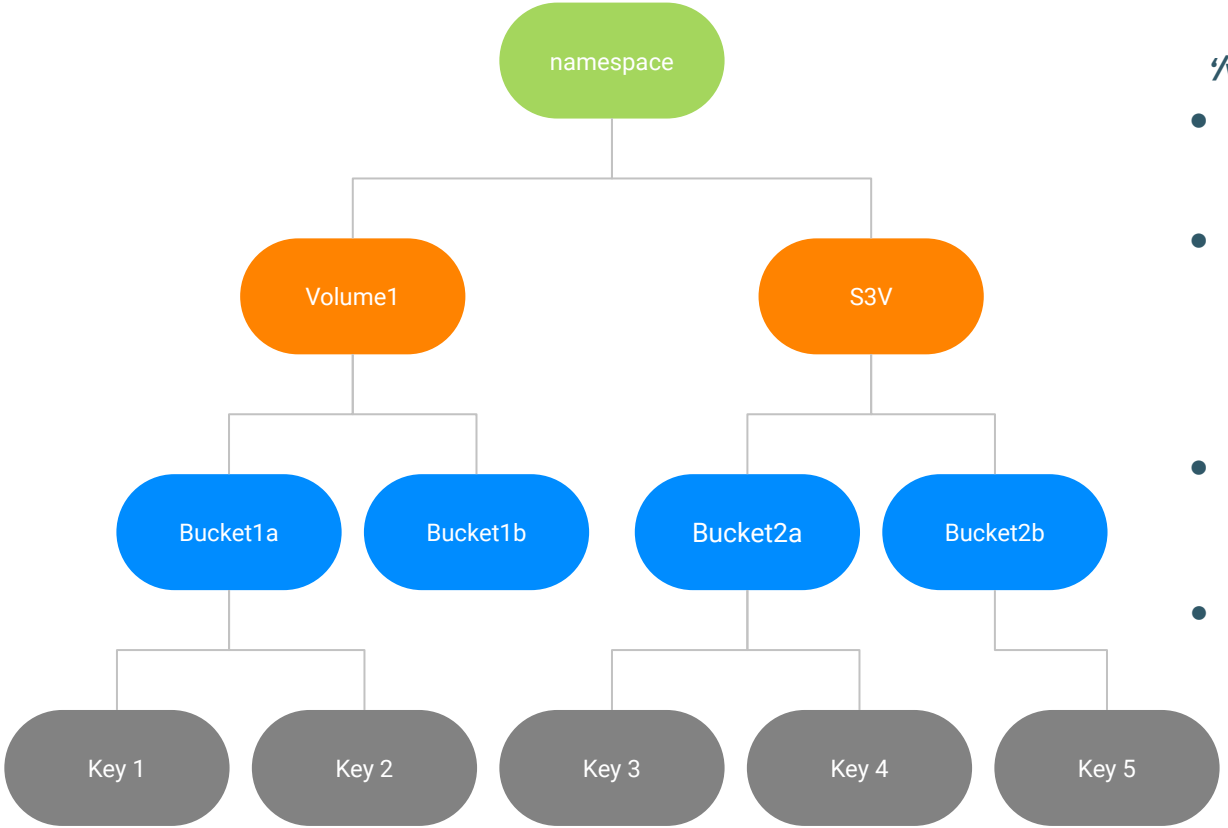


- Every Ozone component uses Ratis to maintain a SM of interest. For eg,
  - Datanode (pipeline) SM tracks the list of containers it has along with blocks, container state etc
  - Ozone Manager SM is the namespace metadata RocksDB whose updates are replicated across the quorum.

Ozone Interfaces



Metadata Layout



Sample Key ==>
'/volume1/bucket1a/app-1/instance-1/key1'

- *Volumes* are similar to user accounts. Only administrators can create or delete volumes.
- *Buckets* are similar to Amazon S3 buckets. A bucket can contain any number of keys, but buckets cannot contain other buckets.
- Keys are similar to files or 'objects' based on how they are created.
- If created using the Filesystem interface, the intermediate prefixes (app-1 & app-1/instance-1) are created as directories in the Ozone metadata store.

Snapshot of Features (1.1.0)

Broad Strokes

- Multi-Interface support
  - Native object store, S3 and HCFS
- Fully Baked Security
  - Kerberos, Token, TDE, S3 security, GDPR, Audit Logging
- Integration with analytic engines (ofs://) - Yarn, Hive, Spark, Impala, Nifi

Namespace

- High Availability of Ozone Manager
- Volume & Bucket Quota Support
- ACL / Authorizer support
- Bucket Linking

Data Management

- Node Decommissioning
- Network Topology Awareness
- Block Locality
- Data integrity Scanner
- Fault Injection Framework

Monitoring

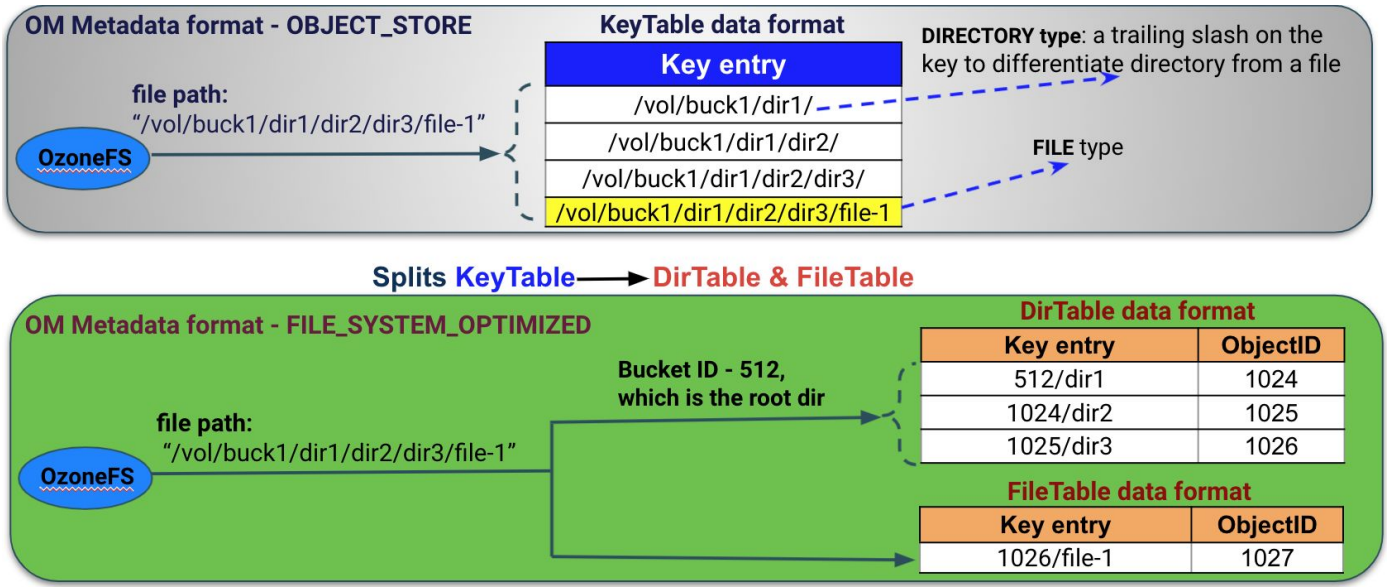
- Management Console (Recon)
- Hadoop Metrics & Prometheus integration
- A host of diagnostic endpoints - REST and CLI

Current Status

- Generally Available since 1.0.0 (Sept 2020 release)
- Latest stable release is 1.1.0
- Upcoming release 1.2.0 is in progress
- 27 PMC members, 51 Committers from Cloudera, Target, Tencent, Infinstor, Oracle, Microsoft, Intel and many more.
- PMC Chair - Sammi Chen, Tencent
- Committers/PMC located in US, Hungary, India, China, Germany and many other countries

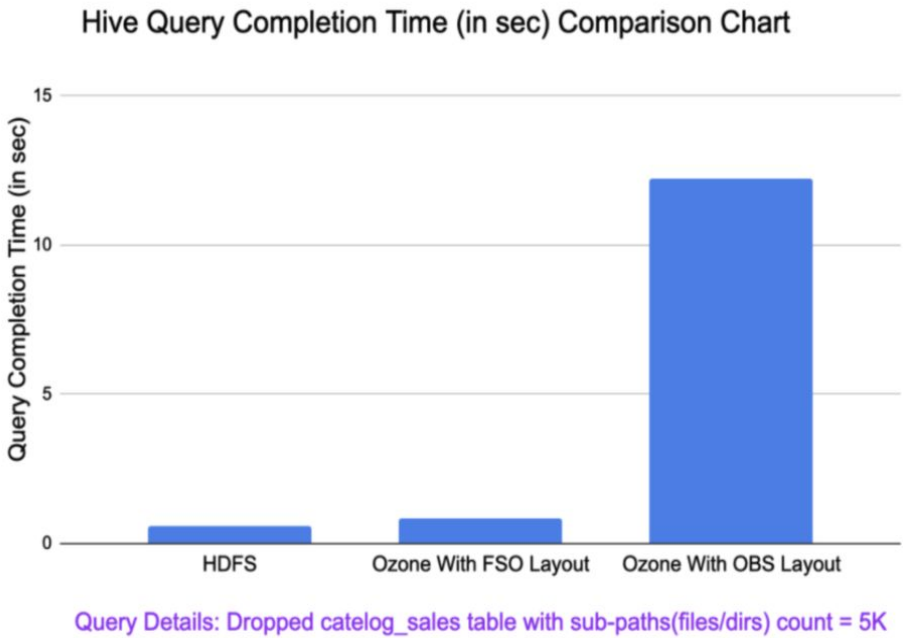
Ozone File System Optimizations (HDDS-2939)

- Atomic guarantees with directory Rename & Delete operations
- Introduces OM Metadata Layout format



Ozone File System Optimizations (HDDS-2939)

Query Details: Dropped "catelog_sales" table with sub-paths(files/dirs) count = 5K	
	Query Completion Time (in sec)
HDFS	0.572
Ozone With FSO Layout	0.854
Ozone With OBS Layout	12.219



Hive drop table query(Rename Operation)

- FileSystem delete on table directory path
- Moves table data to trash

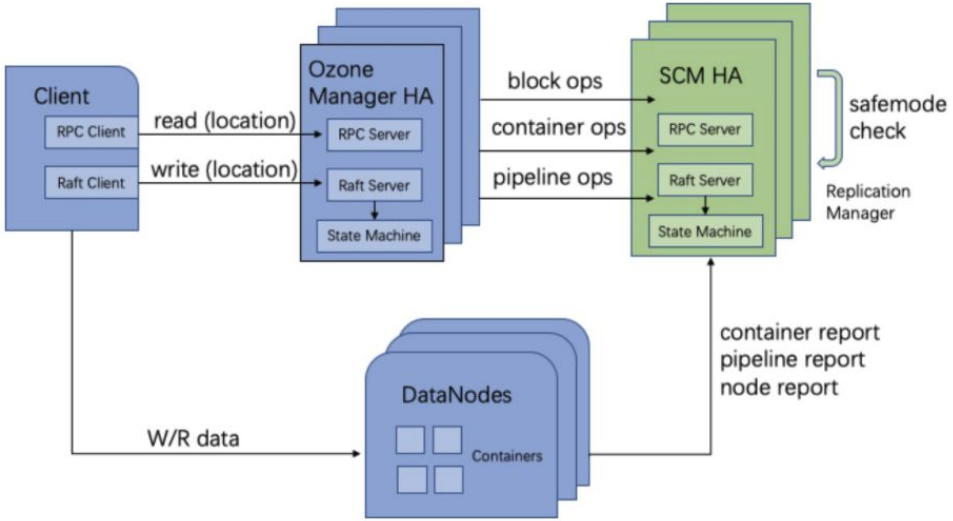
For example:

fs.delete("<prefix_path>/catelog_sales")

- 10-15x improvement on a Hive Drop Table query, ~5% penalty on the write path. More Information here.

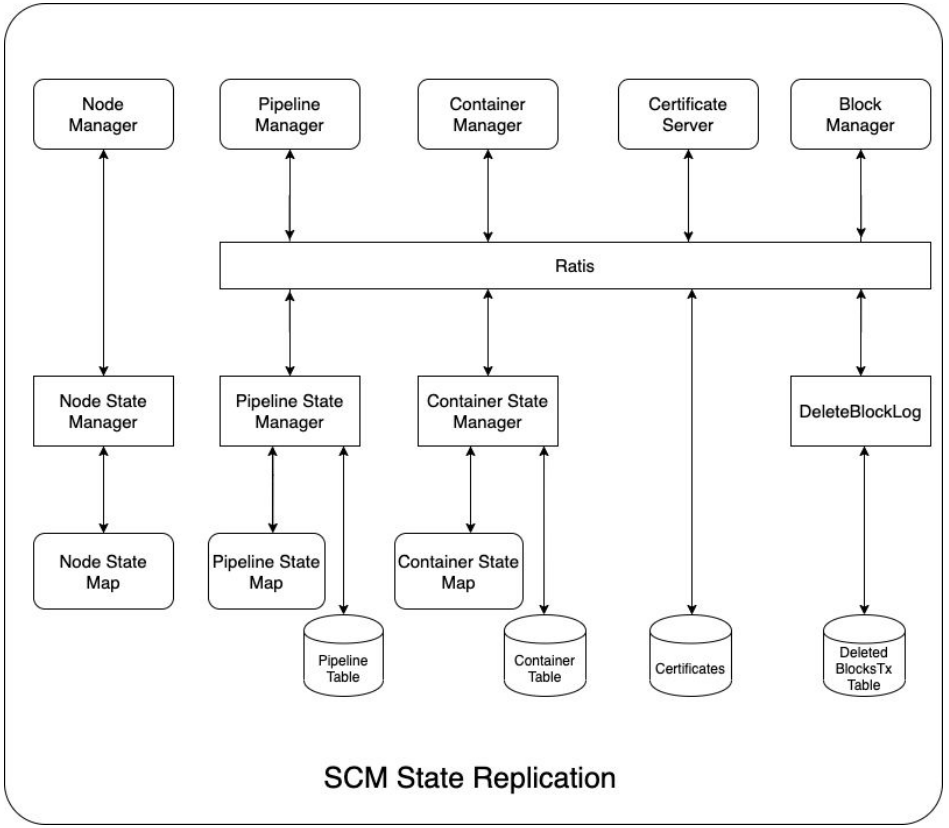
SCM High Availability (HDDS-2823)

- Ratis based SCM HA implementation (2N + 1 instances, Leader & Followers)
- SCM Ratis State Machine → RocksDB with container, pipeline metadata.
- Datanodes will heartbeat and send reports to all the SCMs, but process commands only from current leader SCM.
- All SCM container, block & pipeline state mutations go through replication. Node states do not.



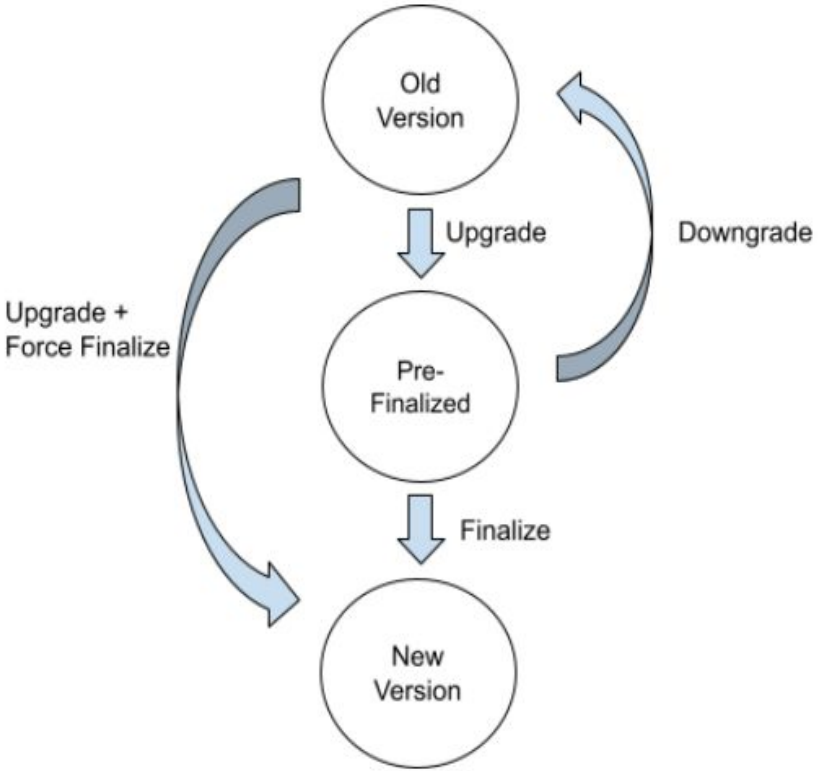
SCM High Availability (HDDS-2823)

- Replicated State in SCM ⇒
- Primordial SCM
  - A designated ‘first’ SCM that is used to init the SCM HA setup
  - Acts as a root CA. In addition all 3 SCMs are also a sub CA.
  - Other SCMs are “bootstrapped” from this SCM.
  - Not to be confused with Ratis leadership. Any SCM can be a leader based on its StateMachine being up to date.



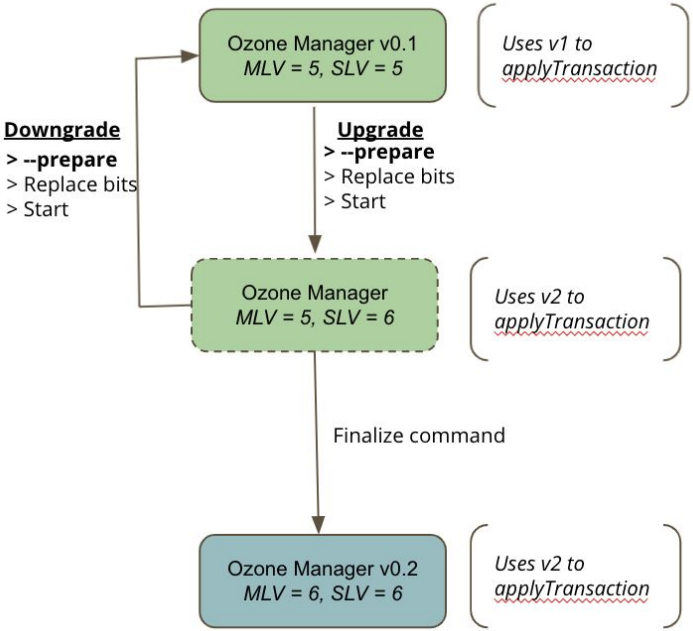
Non Rolling Upgrades (HDDS-3698)

- Borrowed concepts from HDFS
  - Finalization
  - Layout Feature
- Separated OM and SCM version hierarchy & finalization
- Each component writes down layout version on disk.
- Datanodes are finalized through SCM heartbeats.
- Framework provides @Annotation based registering of actions for running on specific phase (On_Prefinalize, On_Finalize, On_Upgraded_First_Start).
- Aspect Oriented programming for separating out functional & upgrade code.



Non Rolling Upgrades (HDDS-3698)

- OM preparation
- Need to prepare OM
  - Ensures that the same version of the software ‘applies’ txn to the OM RocksDB
- Working
  - For every operational OM (at least a quorum of OMs should be operational), all unapplied transactions are applied.
  - For an OM that is not operational during the prepare step, it should get a Ratis snapshot (entire OM RocksDB) to get up to speed with the rest of the OMs after the upgrade.
- Provides utility to prepare & cancel prepare from command line.



Roadmap...

- Erasure Coding (HDDS-3816)
- Persistent connections between s3g and Ozone Manager (HDDS-4440)
- Streaming write pipeline(through Ratis Streaming) (HDDS-4454)
- Container Balancer (HDDS-4656)
- Multi Tenancy support in the S3 Interface (HDDS-4944)
- Namespace Summaries in Recon (HDDS-5305)
- Bucket level layout - Object store / Filesystem (HDDS-5672)

Thank you!

More Ozone talks in ApacheCon today

- **18:00 UTC** - Balancing data in Apache Ozone
- **18:50 UTC** - Performance at billions' scale
- **19:40 UTC** - Secure Apache Ozone with High Availability

Contributions are welcome!

- File/Fix issues at https://issues.apache.org/jira/projects/HDDS/issues
- Questions at **dev@ozone.apache.org**

External References

- https://ozone.apache.org/
- https://blog.cloudera.com/?s=Ozone
- https://blog.csdn.net/Androidlushangderen/article/details/103997315

APACHECON 2021