Unlocking Scalable and Efficient Data Storage with Apache Ozone

- An exabyte scale distributed object storage

Uma Maheswara Rao Gangumalla

- Senior Engineering Manager, Cloudera

Agenda

- What is Apache Ozone
- Building Blocks of Ozone
- Apache Ozone Architecture
- Ozone I/O Flows
- Ozone User APIs
- Key Benefits of Ozone
- Apache Ozone Project Community Status
- Roadmap

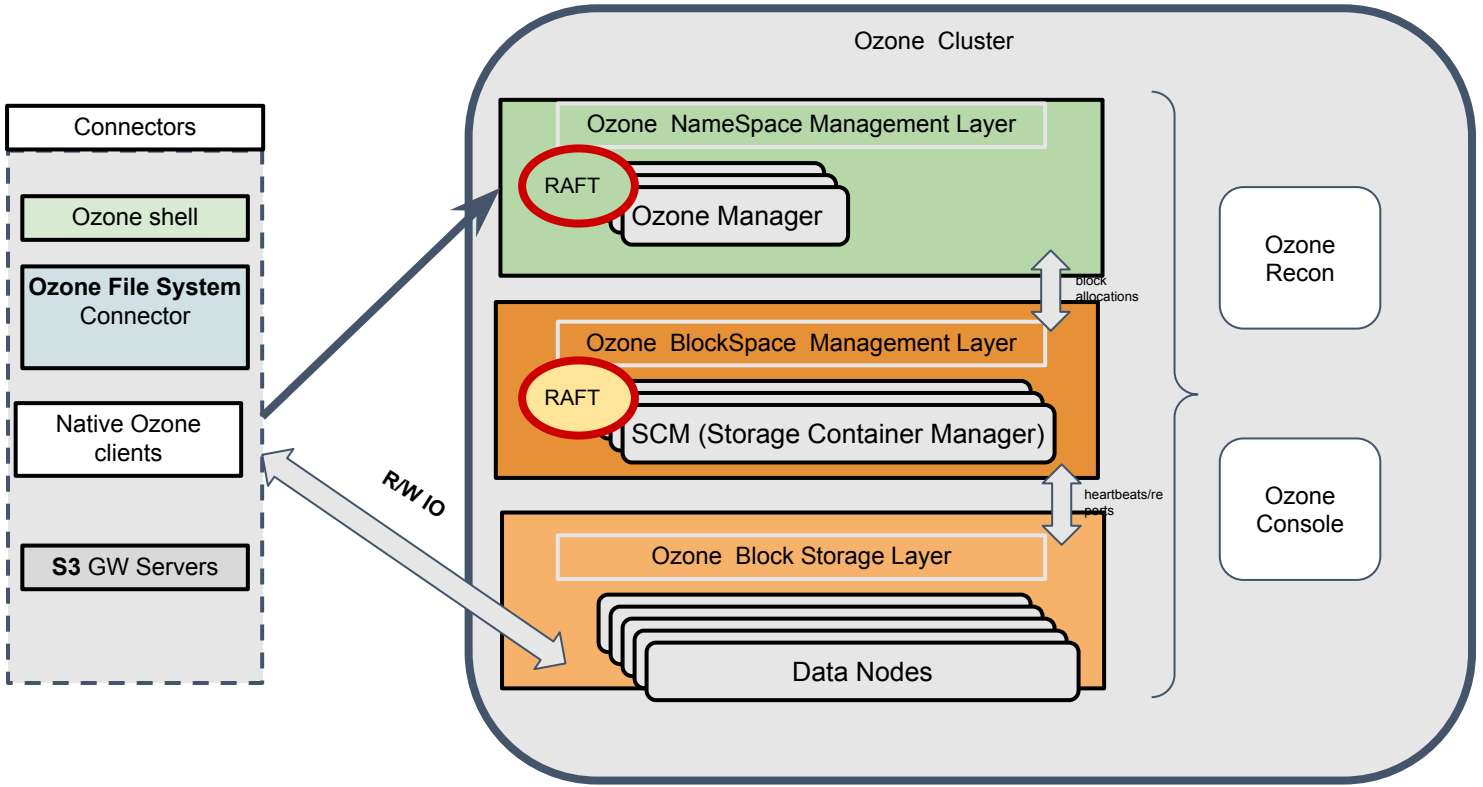
What is Apache Ozone

- A scalable object store for large scale analytical workloads
  - It was designed to fix all HDFS shortcomings and limitations
  - It supports billions of objects (scalable namespace architecture)
  - A single cluster can easily scale to exabytes of data
  - In built HA, RAFT ring for all metadata updates
  - Strongly consistent.
- Apache Ozone was GA in 2020
  - Replication, HA, security, encryption, EC, ACLs
  - Fully integrated with Impala, HIVE, Spark
- A growing open source community
  - Cloudera, G-research, Tencent, DiDi, Shopee, Qihoo ...
  - Ozone is operational at scale

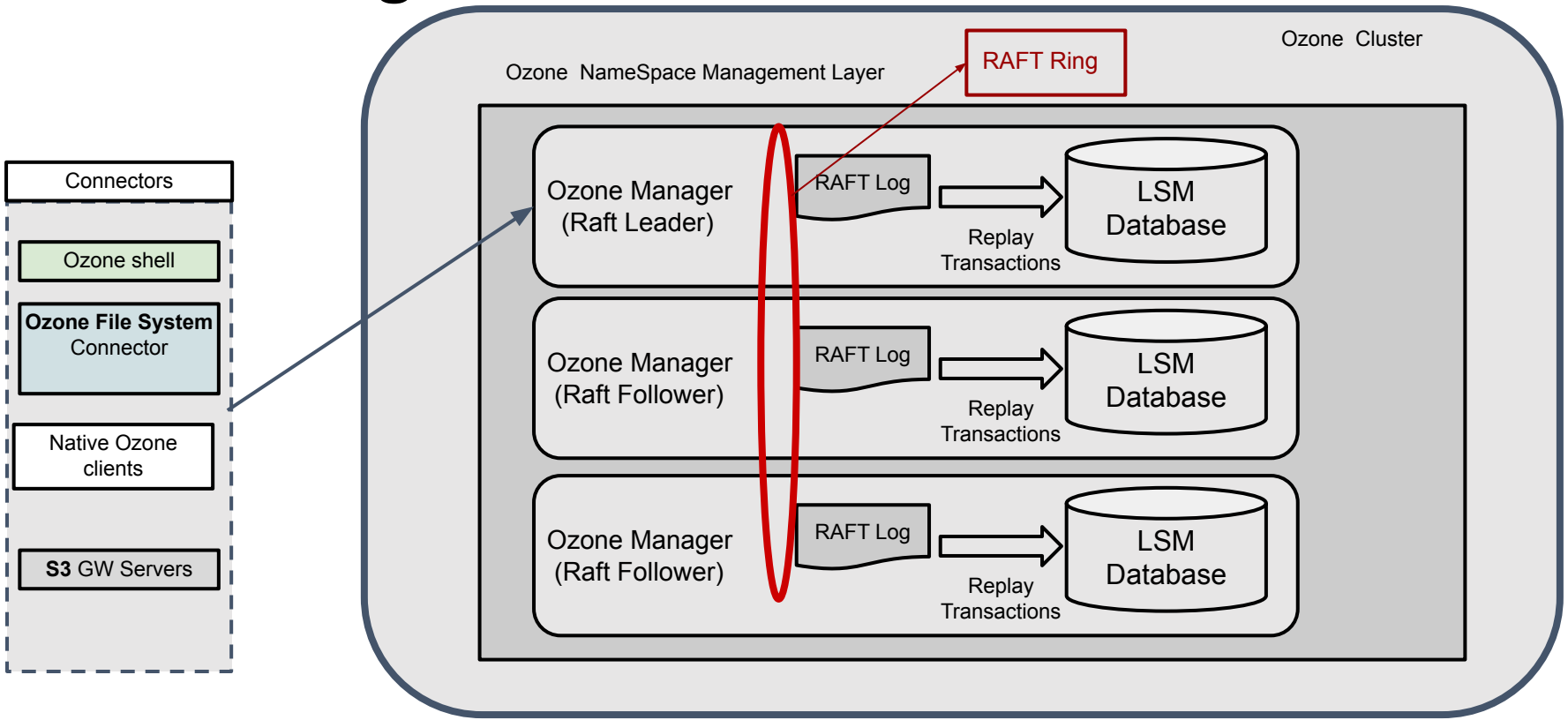
Building Blocks of Ozone

- Ozone separates namespace management and block space management
- Namespace is managed by Ozone Manager (OM)
- Block space is managed by Storage Container Manager (SCM)
- Scales by not tracking individual data blocks. Instead, SCM tracks containers*, which aggregates blocks. By default each container* can be as large as 5 GiB
  - *Not to be confused with Linux or YARN containers :)
- By scaling namespace and block management independently, Ozone can scale to hundreds of billions of files (keys) in a single cluster.

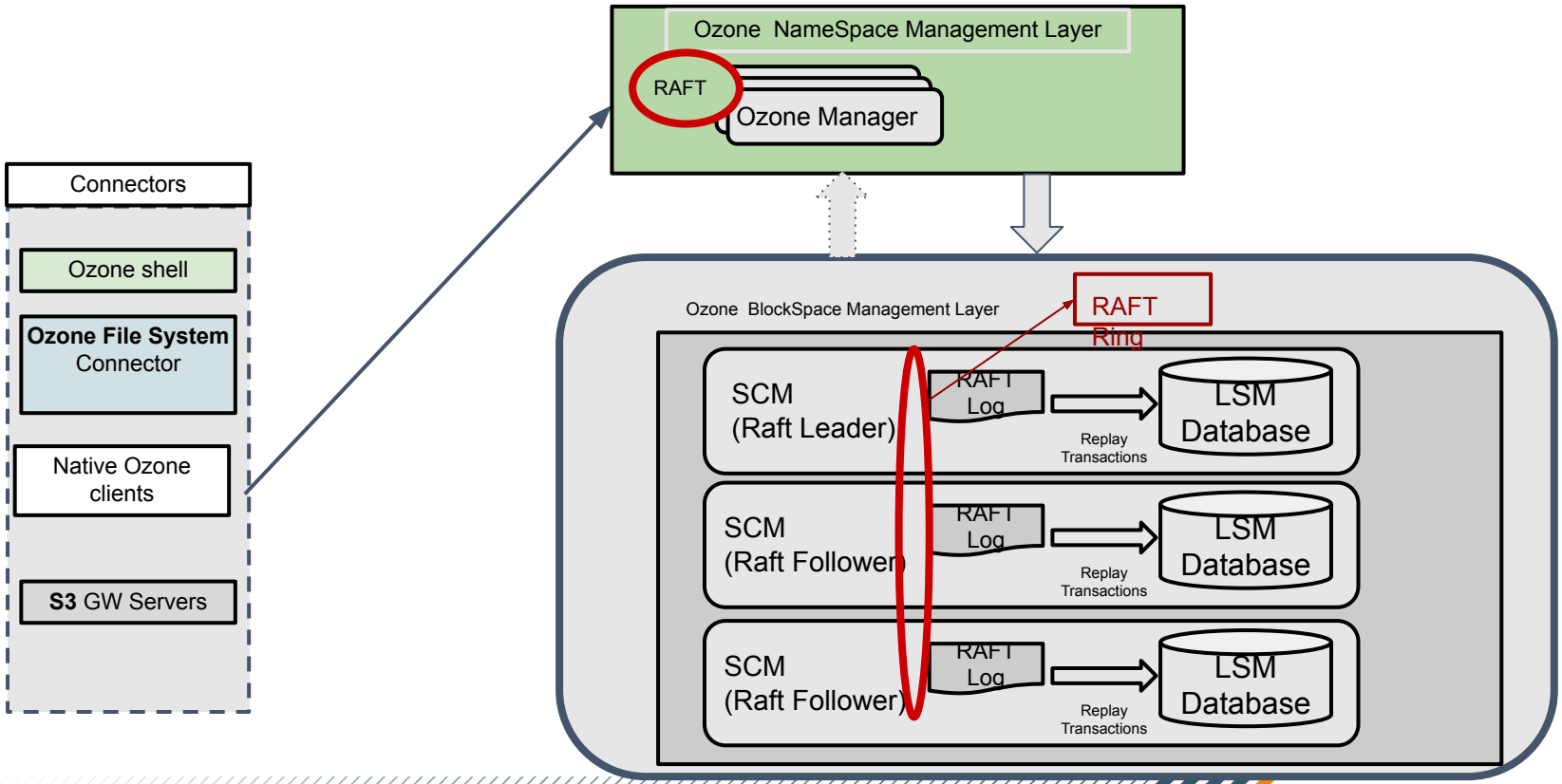
Ozone Architecture



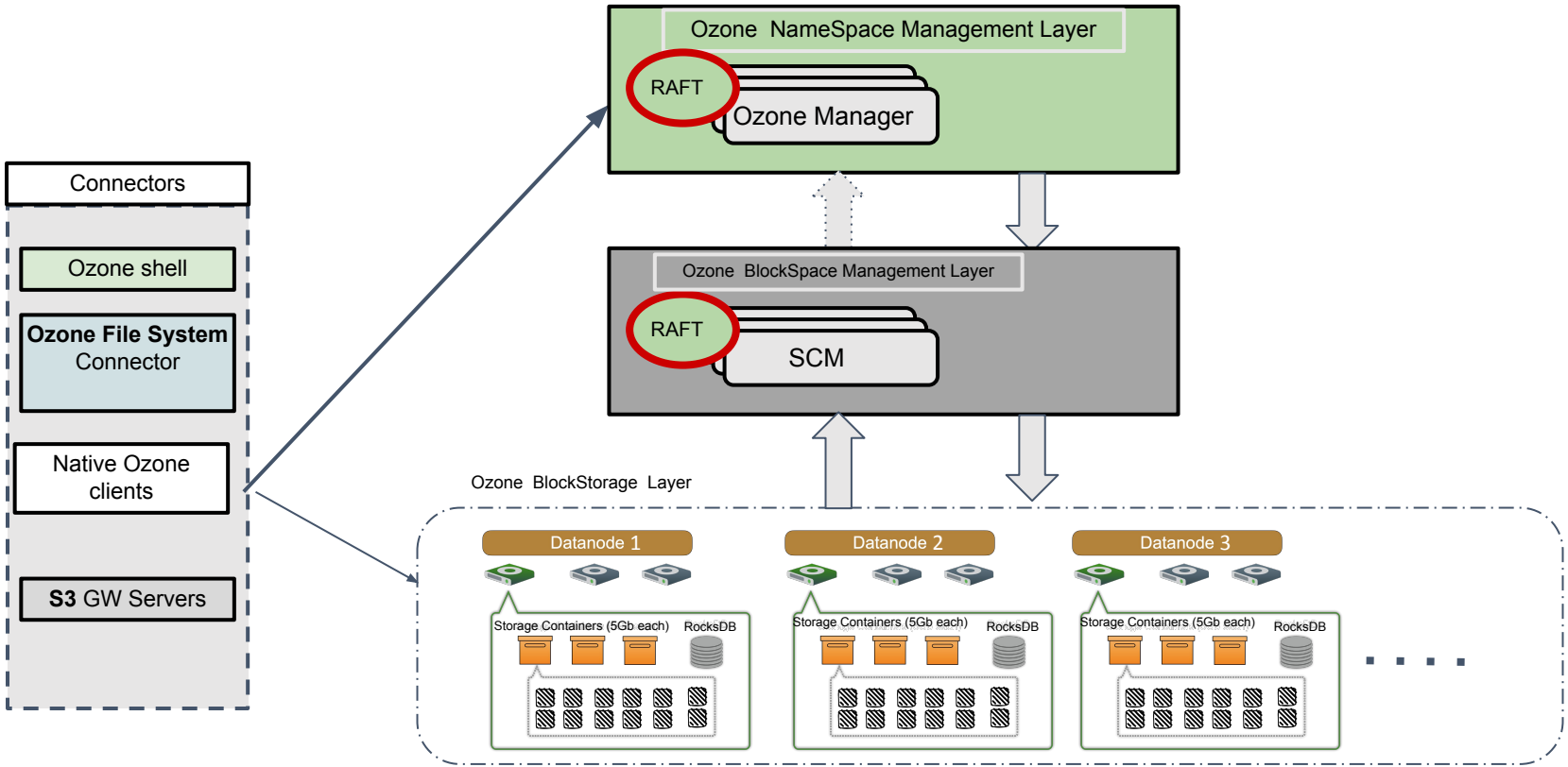
Ozone Manager: more details



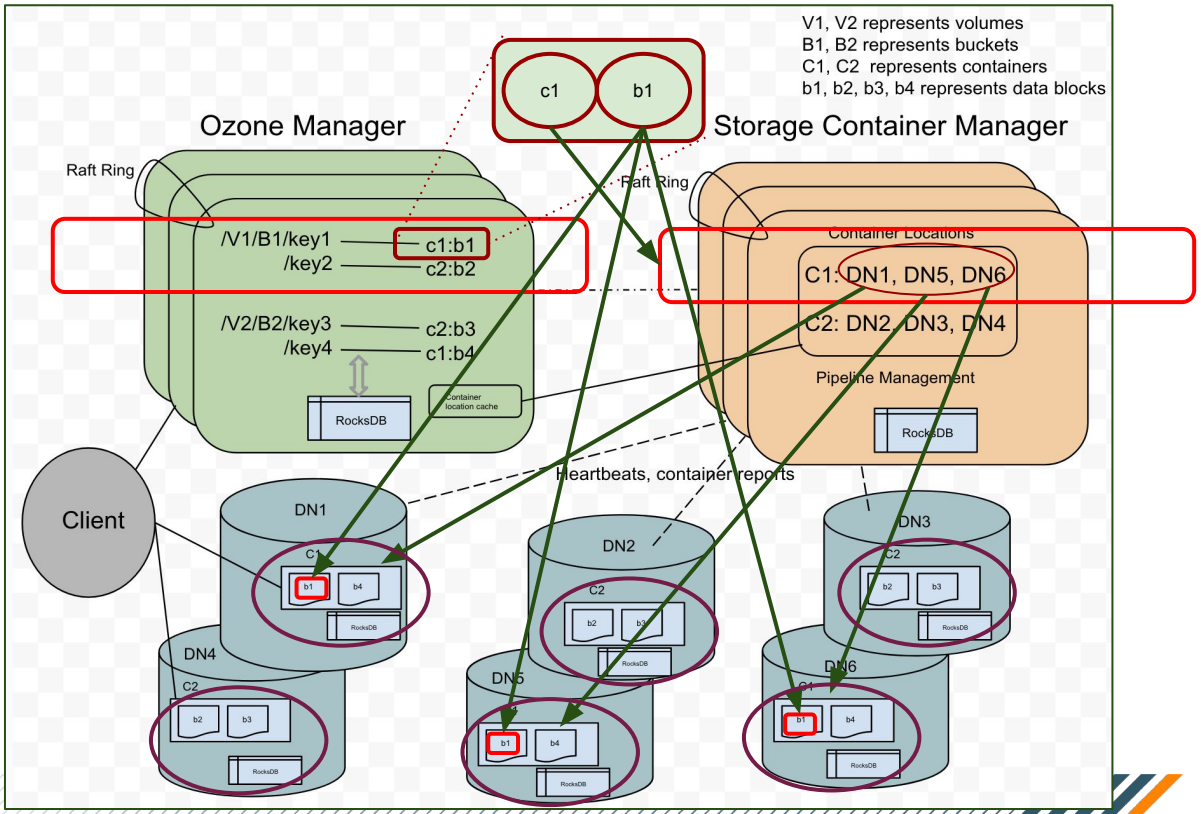
Storage Container Manager: more details



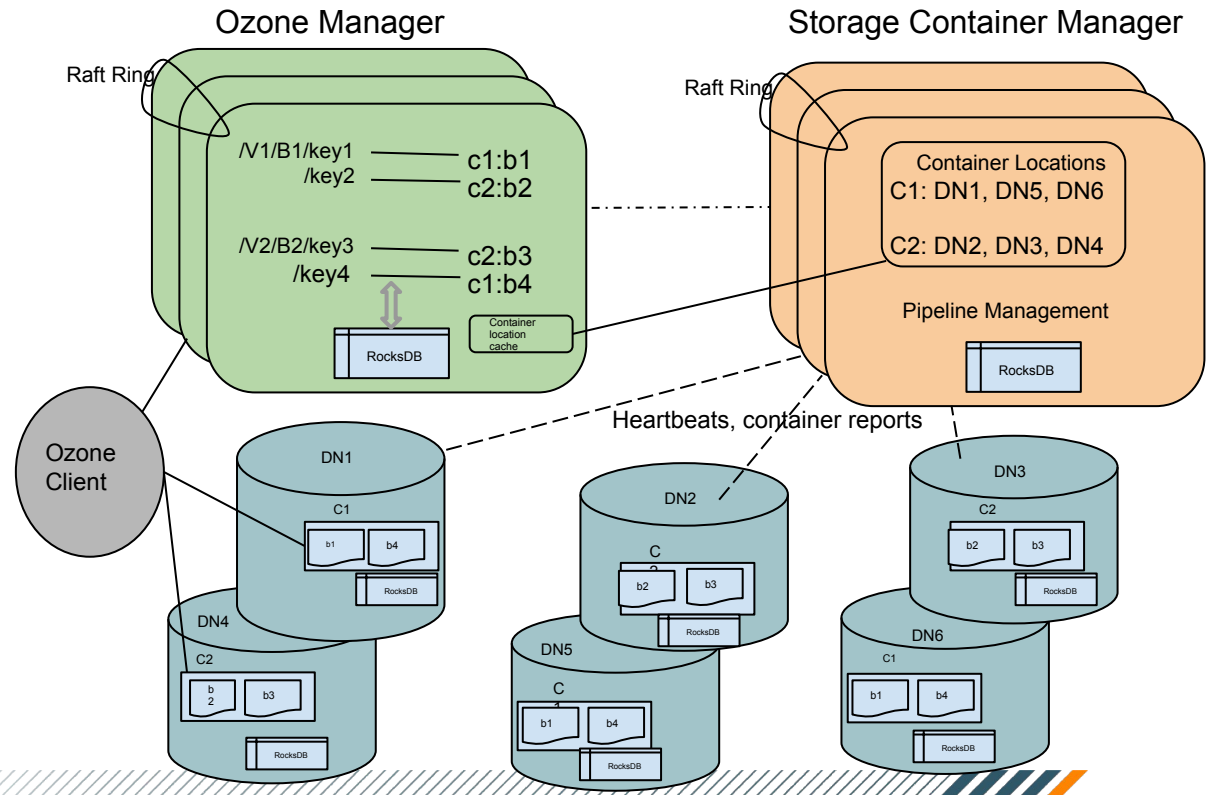
Datanode: more details



Ozone Architecture: Operational view



Ozone Architecture: Operational view



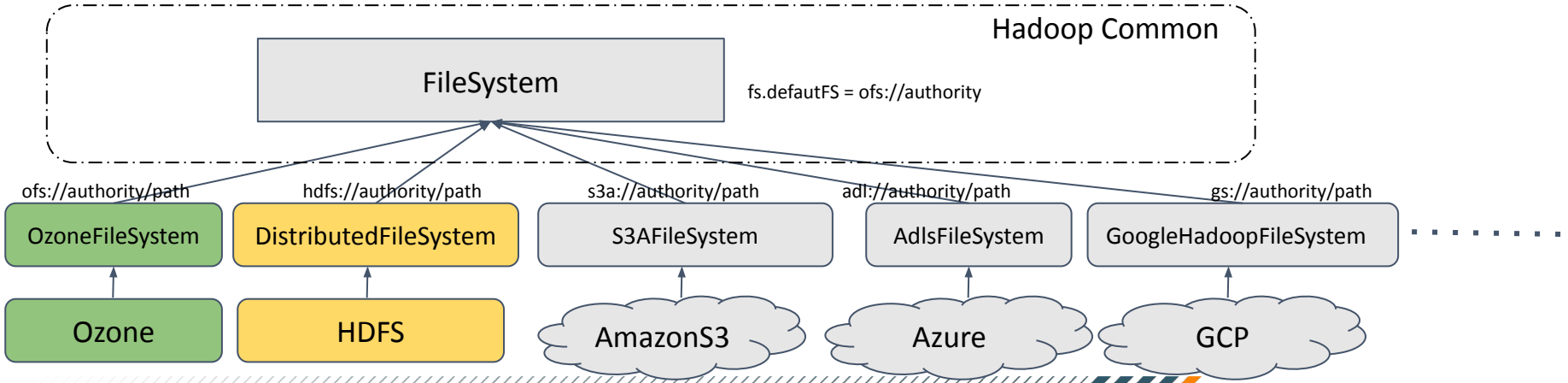
V1, V2 represents volumes
B1, B2 represents buckets
C1, C2 represents containers
b1, b2, b3, b4 represents data blocks

Apache Ozone - User APIs

Ozone Client : File System and S3

File System:

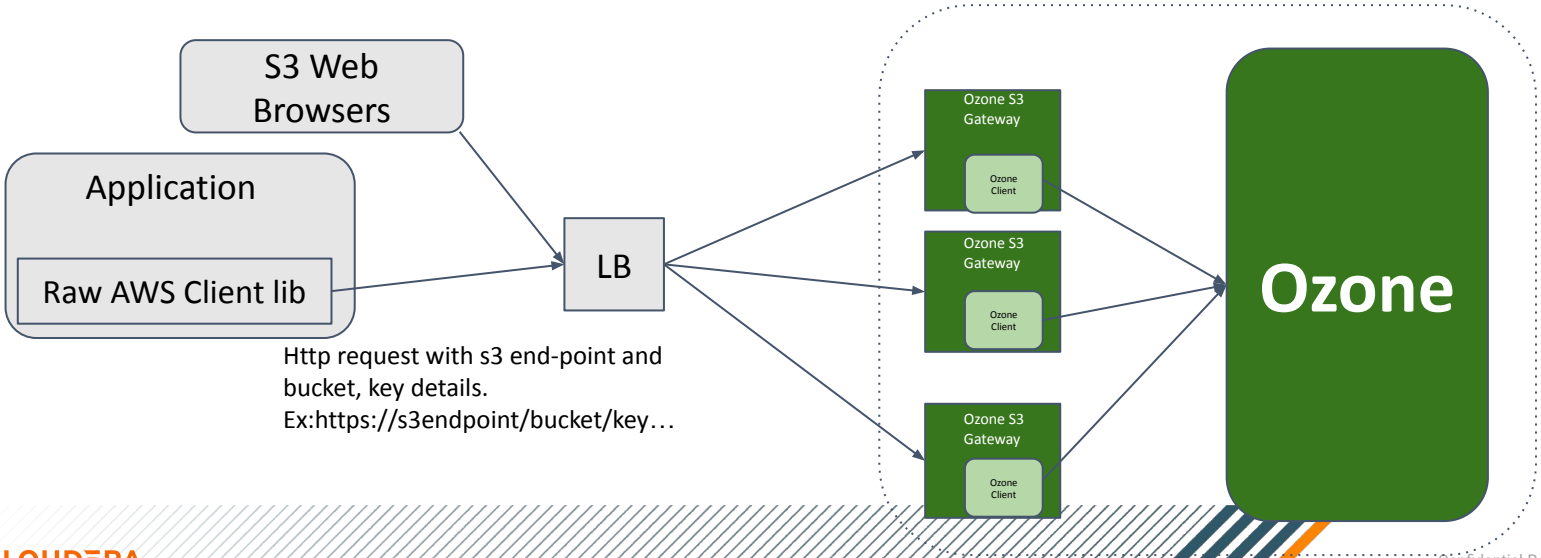
- An implementation of Hadoop Compatible FileSystem Interface
- Ozone File System protocol scheme is “ofs”
- Hadoop FileSystem Interface abstraction layer is intelligent enough to find the implementation classes based on scheme in the path. Example: ofs://testpath



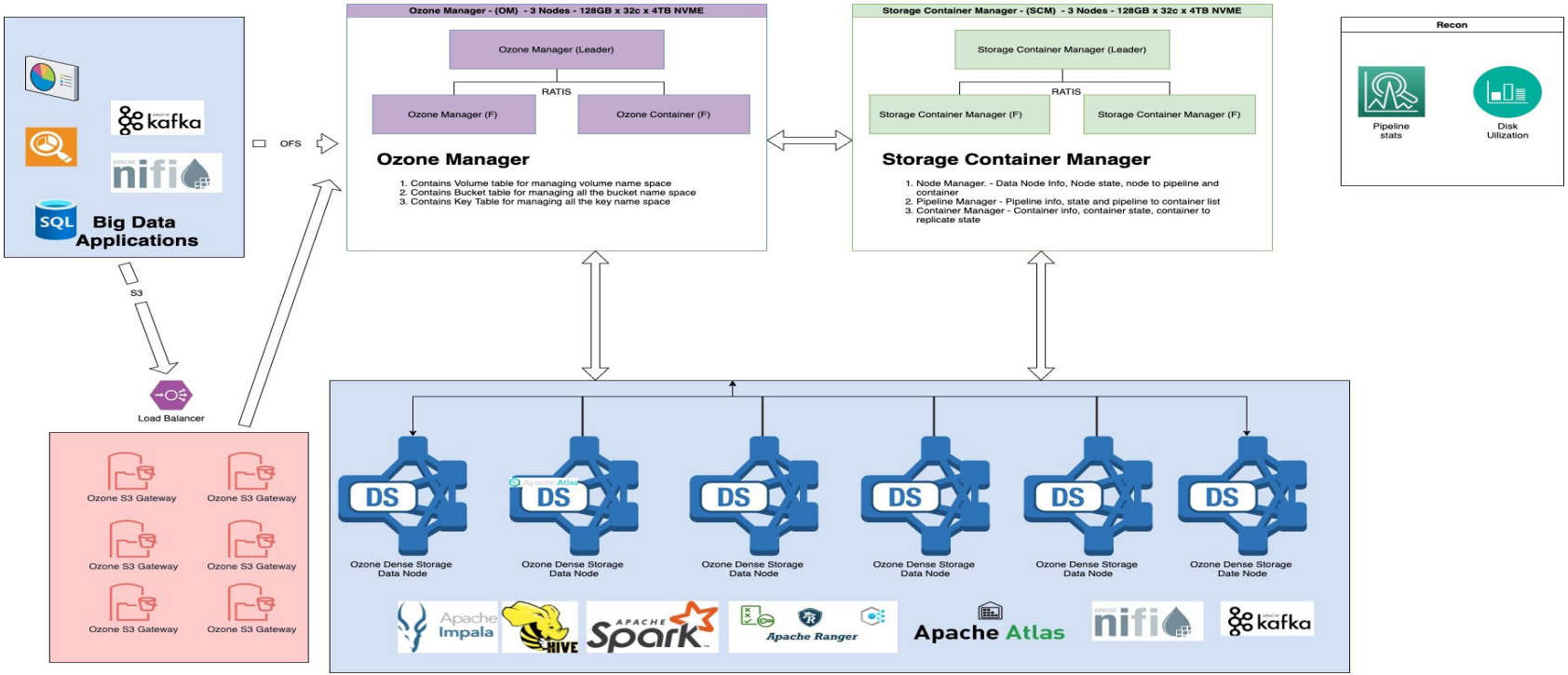
Ozone Client : File System and S3

S3 API:

- Amazon S3 API compatibility
- Http calls with amazon-s3 client based calls(http) will land on Ozone S3 Gateway servers
- S3 Gateway servers redirect them to



Ozone Dense Data and Full Deployment Architecture

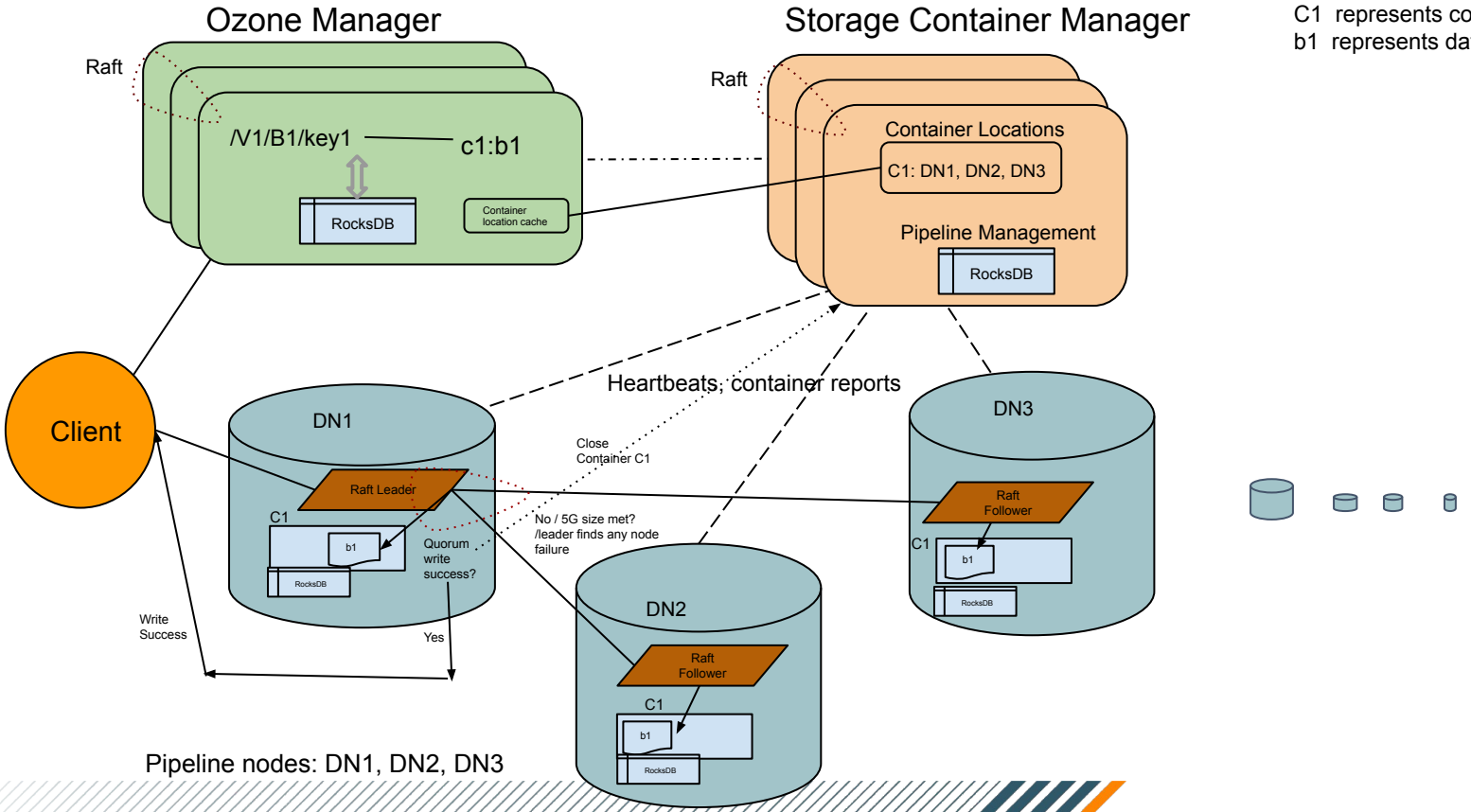


Ozone I/O Flows Overview

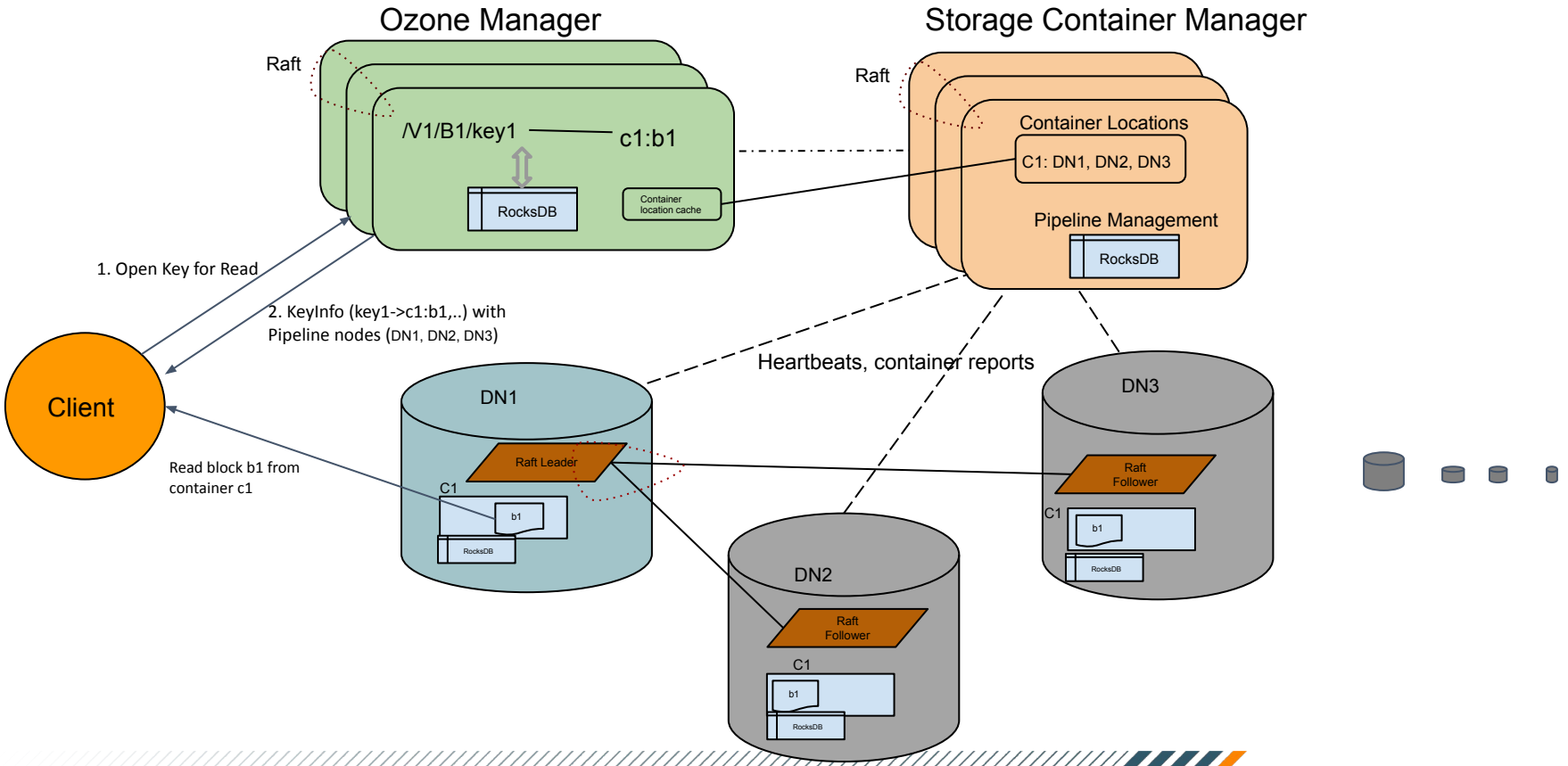
Replicated I/O Path

Ozone Replication Writes

V1 represents volumes
B1 represents buckets
C1 represents containers
b1 represents data blocks



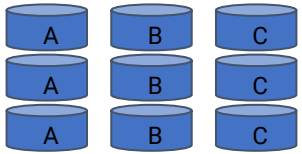
Ozone Reads


Ozone Manager
Raft
/V1/B1/key1
c1:b1
RocksDB
Container location cache
Storage Container Manager
Raft
Container Locations
C1: DN1, DN2, DN3
Pipeline Management
RocksDB
1. Open Key for Read
2. KeyInfo (key1->c1:b1,..) with Pipeline nodes (DN1, DN2, DN3)
Client
Read block b1 from container c1
Heartbeats, container reports
DN1
Raft Leader
C1
b1
RocksDB
DN2
Raft Follower
C1
b1
RocksDB
DN3
Raft Follower
C1
b1
RocksDB

Storage Efficient IO Path - Ozone Erasure Coding

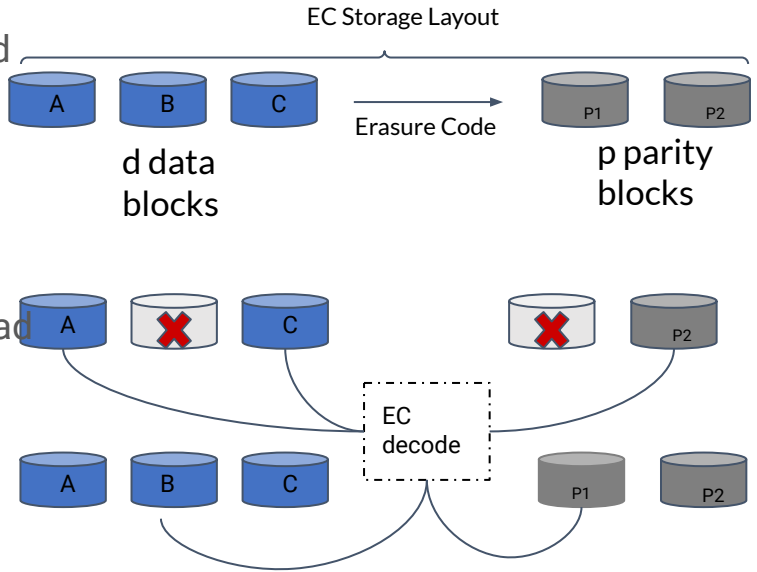
Quick EC Introduction

Traditional 3-Way Storage Layout

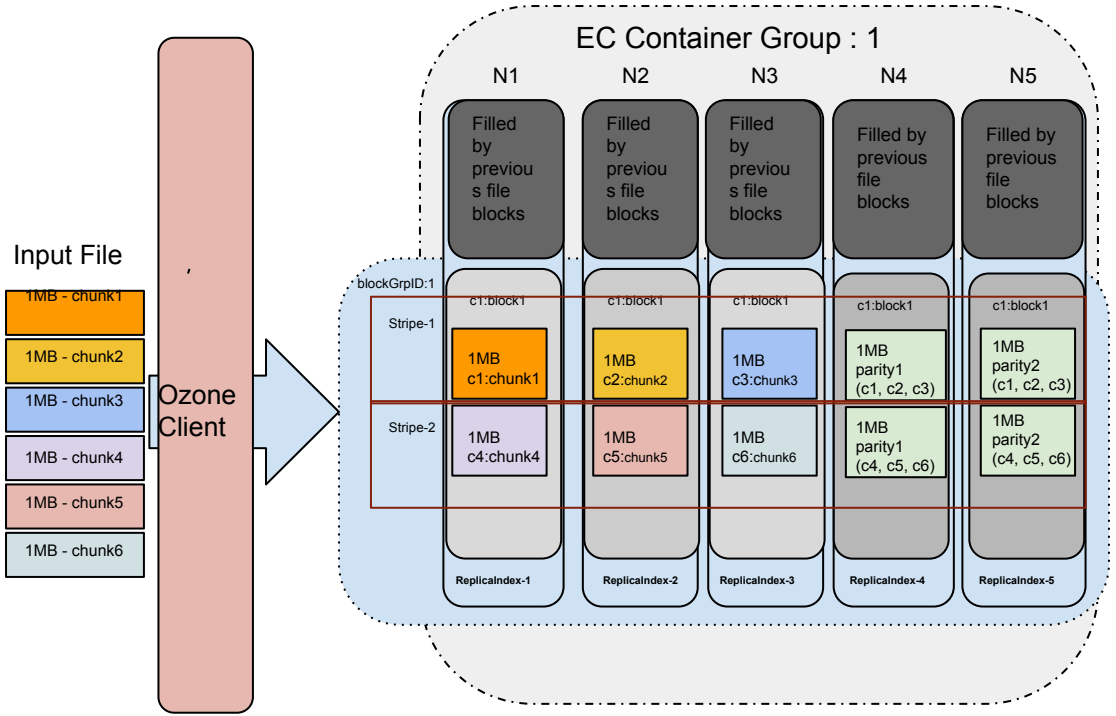


- 3-Way Replication needs additional storage space(3x) to store it's actual and duplicate copies. It's costly.

- Upto 50% storage space savings compared to replicated layout
- Erasure Coding(EC) is a method of data protection.
- It generates additional protection data pieces called parity.
- We don't need to store duplicate copies any more, instead we just store the original data and parity data.
- In EC, we can choose data(d) and parity(p) numbers.
- Ozone supports 3(d):2(p), 6:3, 10:4 schemas

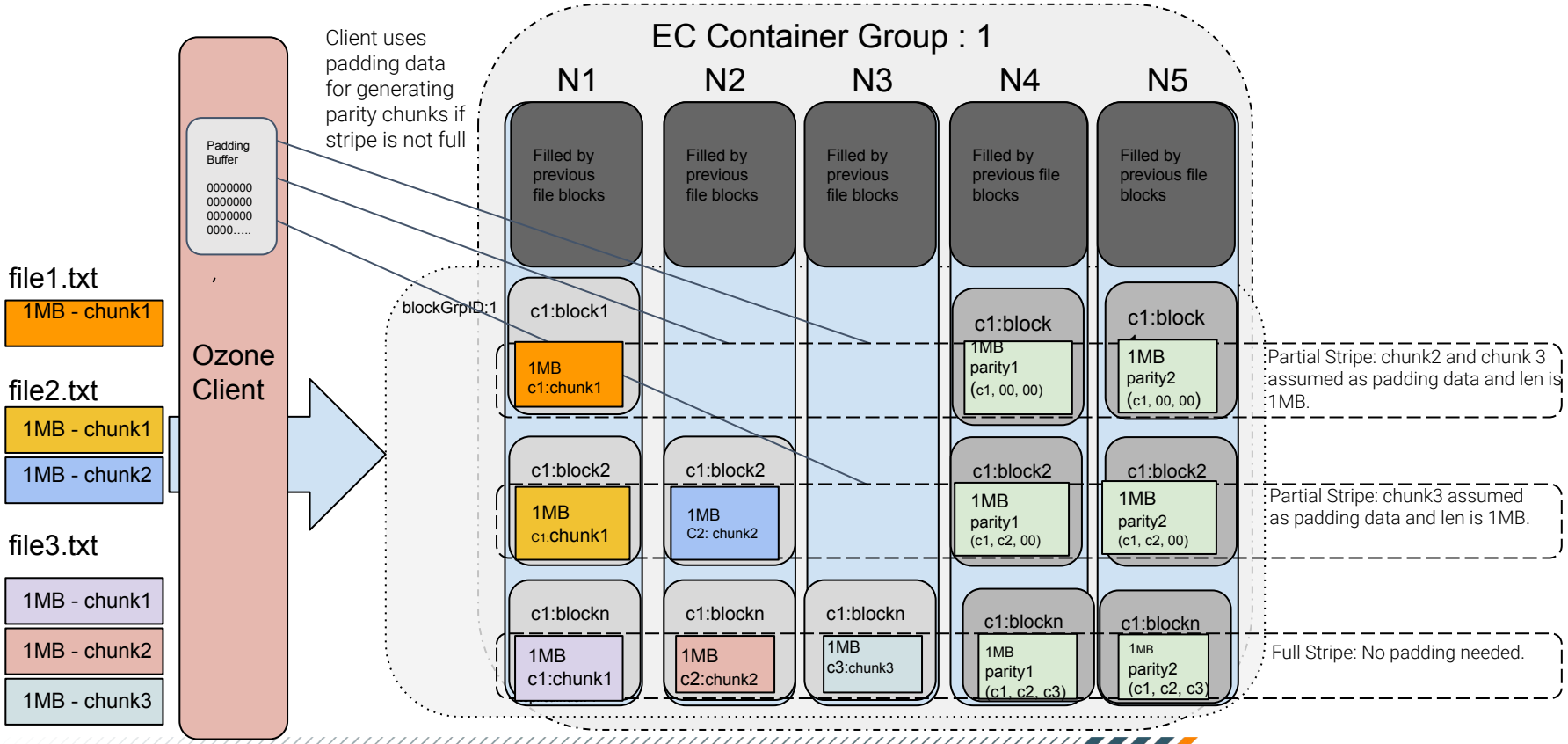


Ozone EC Write: Striping

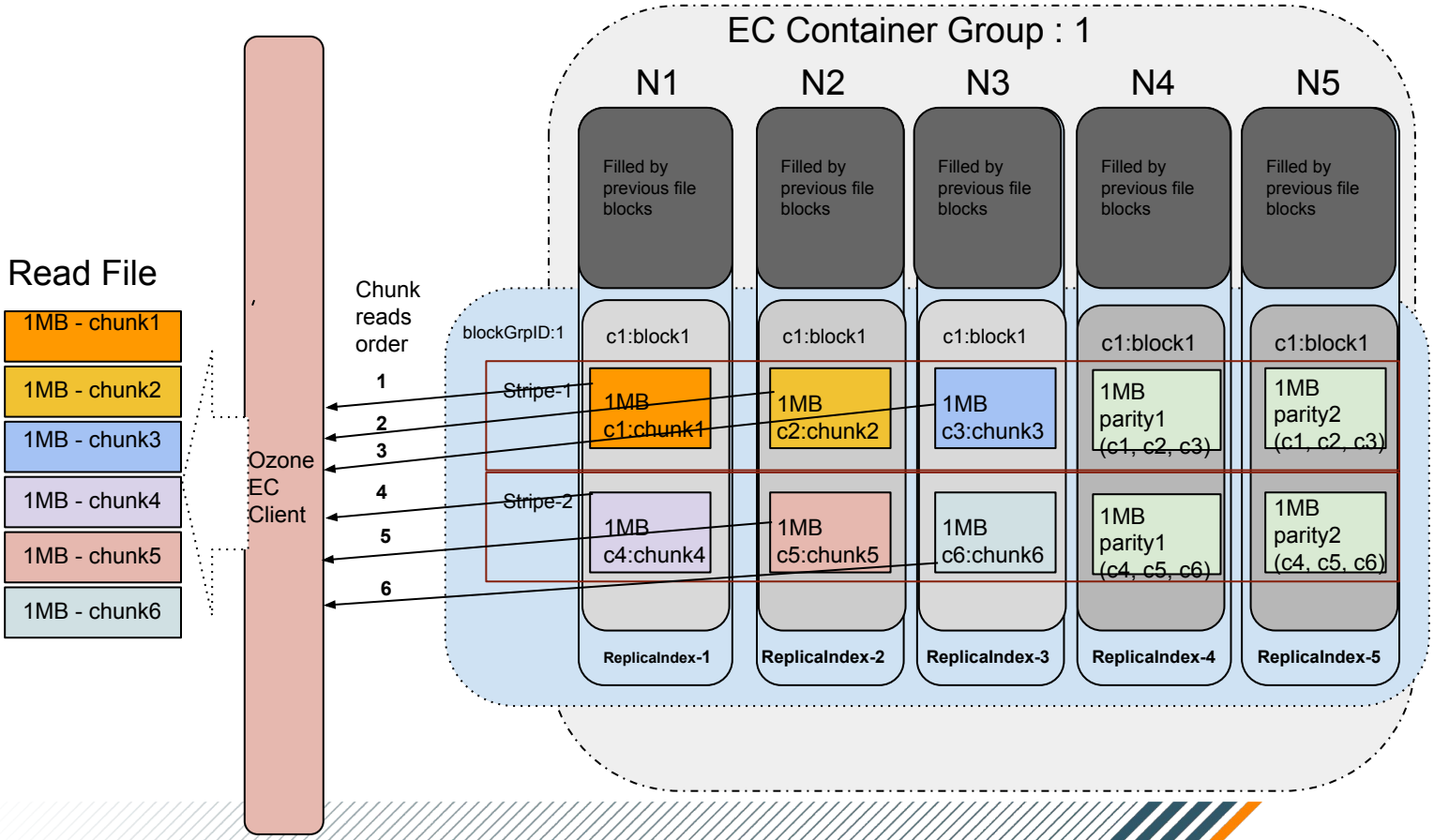


- Chunks would be written in round robin fashion to data nodes.
- Parity Generation: After every d number of chunks written, p parity will be generated and send to remaining nodes in group.
- ReplicaIndex: It will represent the position of chunk with respective to ec input buffers order. In other words, EC Chunk position in full stripe, in the order of 1 to (data + parity)
- If stripe write fails, the current block group will be closed and rewrite the failed stripe to new block group.
- Client keeps track of bytes written and check for failures.

Ozone EC Write: Partial Stripe with Padding



Ozone EC Reads



Key Benefits of Ozone

- Flexible API - FS and S3
- Resilient to hardware failures
- Effient storage layout support
- Seamless integration with big data applications like spark, iceberg etc
- Dense storage - Storage nodes can have large capacity (tested up to 400TB in a single storage node in cluster)
- Beyond exabyte scale capacity with high performance
- Healthy Apache Ozone Community - One of the most active projects in Apache
- Feature set - Key features like snapshots, multitenancy etc.

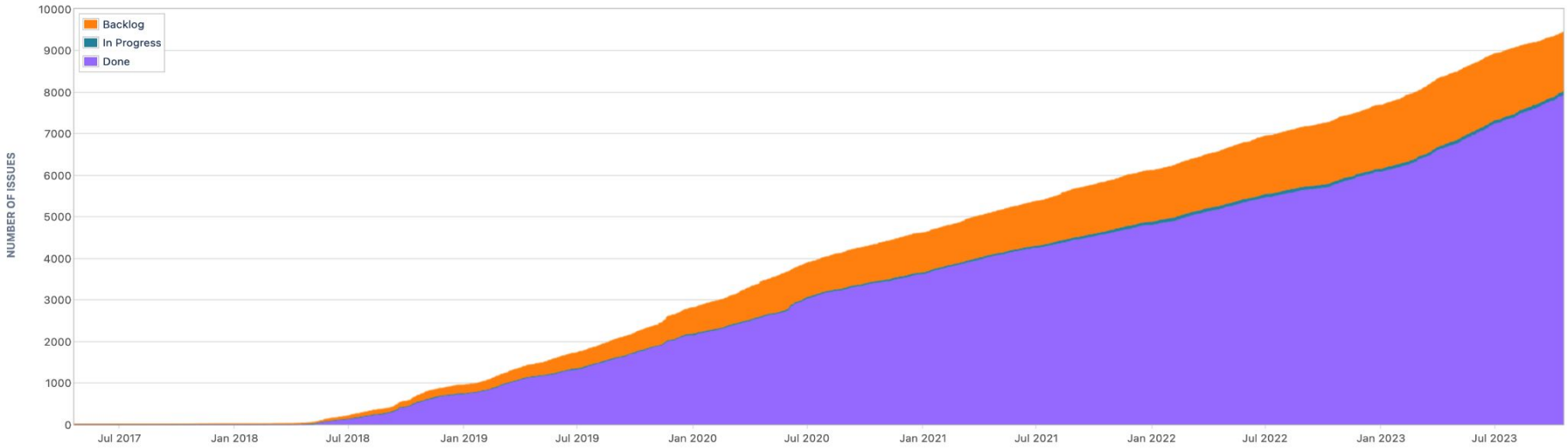
Ozone Project Status

Apache Ozone Committee and Community

- Ozone PMC Chair: Sammi Chen
- 36 PMC members, 74 Committers
  - Committers / PMC members located in US, Hungary, India, China, Germany, …
  - from Cloudera, Target, Tencent, Infinstor, G-Research, Shopee, Qihoo, Didi …
- 233 contributors (who has at least one PR merged), 120+ active contributors in the past two years.
- 6536 commits in total on the main branch, 2437 merged in the past two years.

Apache Ozone JIRA

- 9,500+ JIRAs opened under Apache Ozone (HDDS) project and counting
- 3,673 JIRAs opened, 2,822 of them resolved in the past 2 years



Roadmap

- ❑ HBase Ozone support
- ❑ Recon UI/UX improvements and new features
- ❑ Storage tiering
- ❑ CSI Support
- ❑ Rolling upgrades
- ❑ Deduplication
- ❑ Solr support
- ❑ and more ….

For more details

- Ozone homepage: https://ozone.apache.org
- Ozone repo: https://github.com/apache/ozone
- Ozone dev wiki: https://cwiki.apache.org/confluence/display/OZONE
- Developer mailing list: dev@ozone.apache.org

Q & A

umamahesh@apache.org

@UmamaheswaraG

OSA CON 23