APACHECON

ASIA 2022

Apache Ozone 的最近进展和实践分享

刘岩 陈怡

2022.07.29

目录

- Apache Hadoop HDFS面临的问题
- Apache Ozone介绍
- Apache Ozone适用场景
- Apache Ozone的最近进展
- Apache Ozone的实践分享

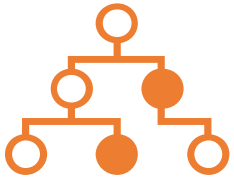
大数据存储的需求

性能

能否提供高并发读取和写入

API 兼容性

是否兼容主流API，如HDFS/S3



扩展性

是否可以扩展至数百PB的存储容量，数千个物理节点以及数十亿个对象

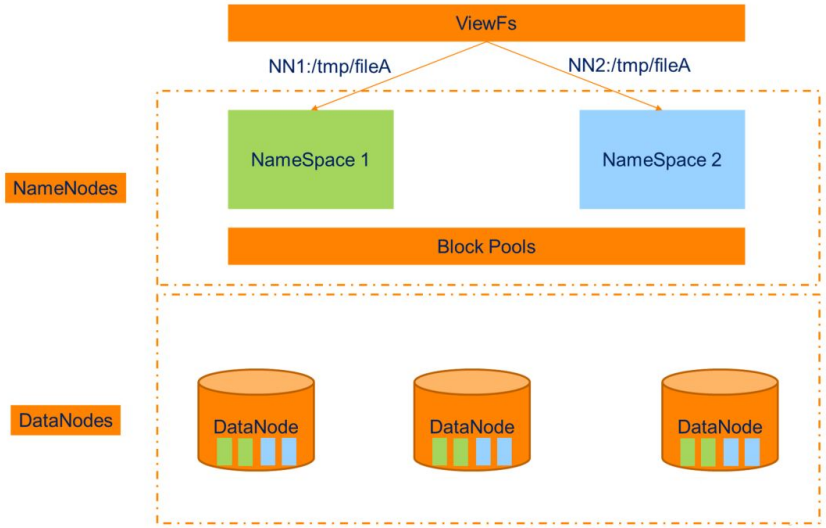
加密

安全

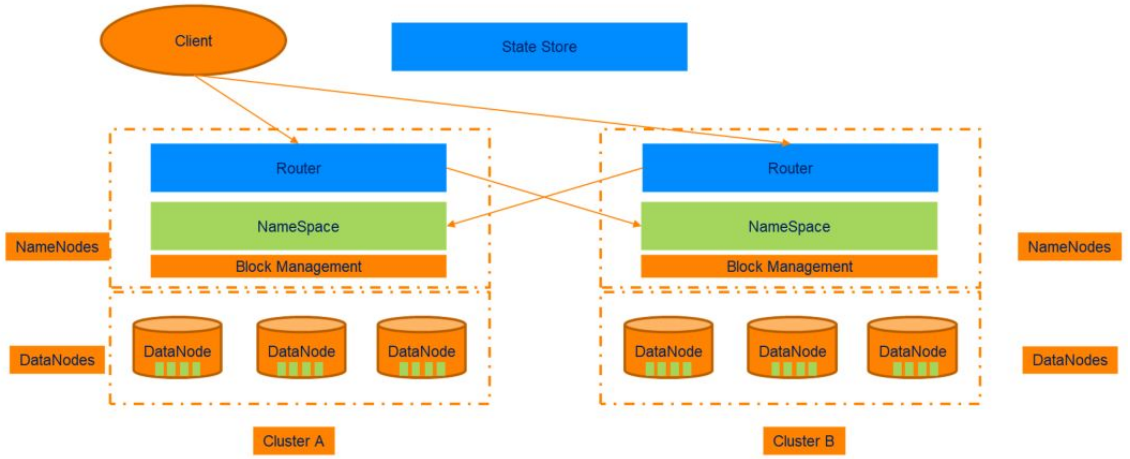
应用对接

是否支持存算分离架构同时也可以兼容存算耦合架构

HDFS现有的一些解决方案



Namenode Federation



Router Based Federation

是否需要一个新的大数据存储?

HDFS的扩展性
达到了上限

无法接受私有化
的数据存储系统

现有的对象存储方案
无法很好的横向扩展

公有云的对象存储服务
无法在线下部署

目录

- Apache Hadoop HDFS面临的问题
- Apache Ozone介绍
- Apache Ozone适用场景
- Apache Ozone的最近进展
- Apache Ozone的实践分享

Apache Ozone

- Ozone是
  - 一个分布式的KV对象存储
  - 可扩展至数十亿个对象，从而对云原生类的应用更友好
  - 强一致性
  - 与HDFS 和 S3 API兼容
  - 可在存储密集型设备中部署进而极大的减少设备开支

Apache Ozone – 数据存储的路径设计

Ozone的存储路径为 **volumes, buckets, 和 keys.**

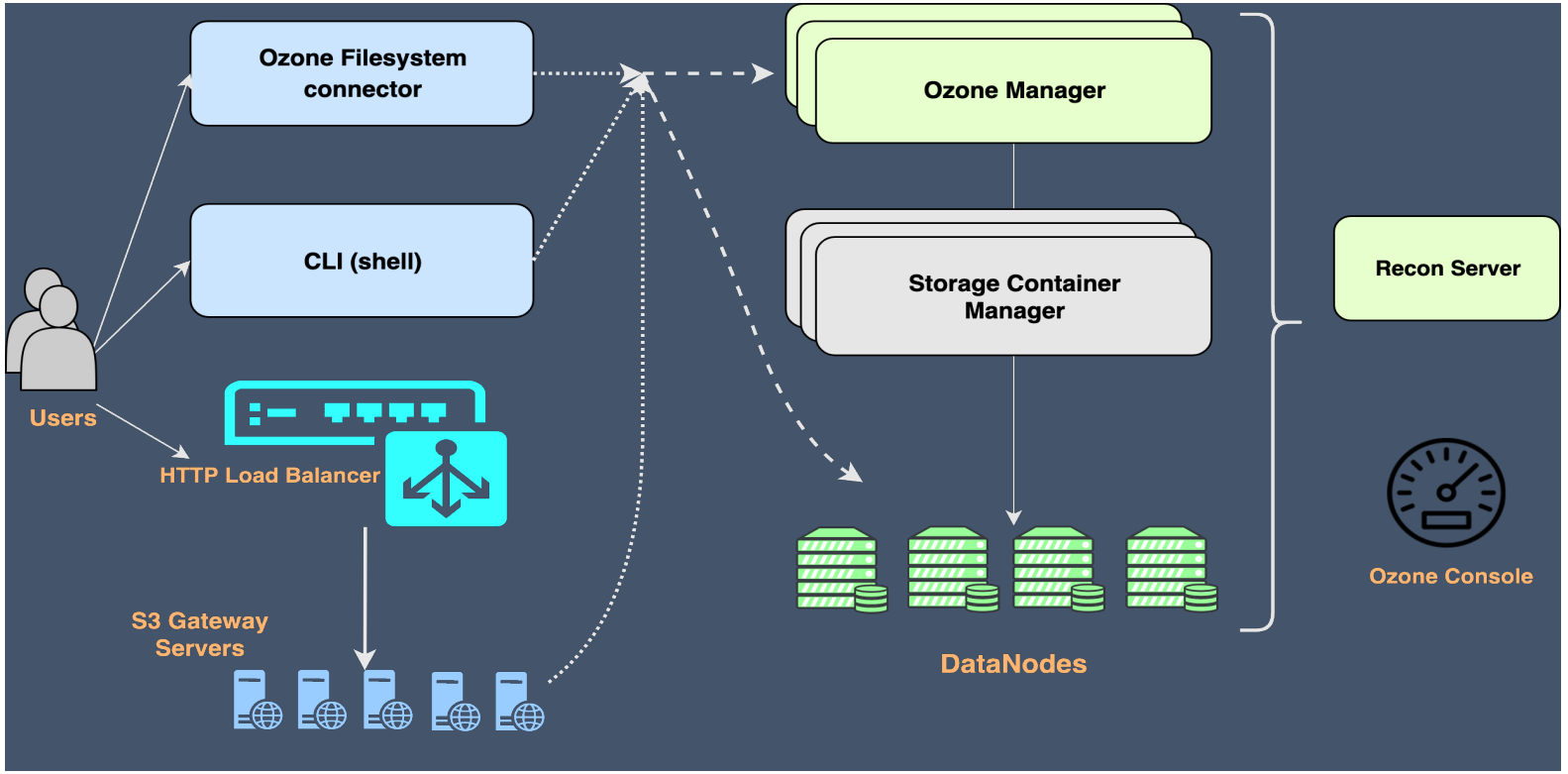
Volumes 类似与用户账号. 只有Admin 可以创建或删除Volumes

Buckets 类似与S3 的 Buckets， 一个Buckets中可以包含任意多个Key，但不能包含其他Buckets

Keys 类似于文件.

文件系统的层级关系是通过扁平的KV路径抽象实现的

Apache Ozone – 数据服务的核心设计



Apache Ozone – 数据服务的核心设计

1. **OM** – 管理Ozone的Namespace， 也使用了RocksDB
2. **SCM** – 管理Ozone集群和数据
3. **Recon Server** – 监控Ozone集群
4. **DataNode** – 负责存储和汇报Storage Containers
5. **Storage Containers** – Ozone的存储单元，内置有RocksDB数据库

Apache Ozone – 数据访问的API

ofs

hdfs dfs -mkdir /volume1/bucket1

o3fs

hdfs dfs -ls o3fs://bucket.volume.om-host.com:5678/key

aws s3

aws s3 ls --endpoint http://localhost:9878 s3://buckettest

ozone cli

ozone sh volume create /vol1

目录

- Apache Hadoop HDFS面临的问题
- Apache Ozone介绍
- Apache Ozone适用场景
- Apache Ozone的最近进展
- Apache Ozone的实践分享

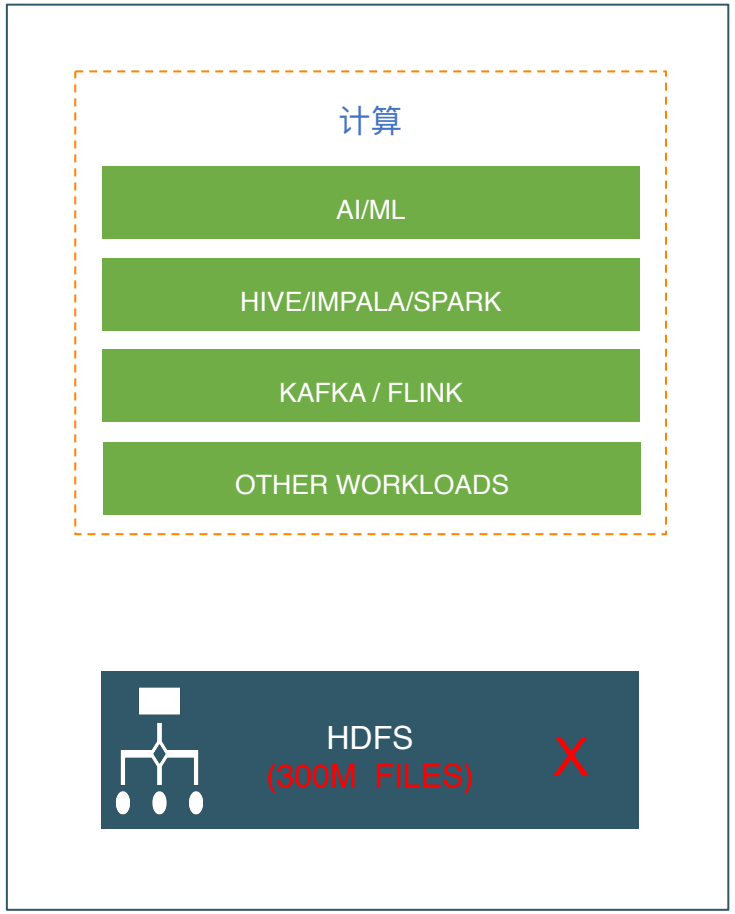
Apache Ozone – 使用场景 #1

业务价值

- 可用于承载实时和批处理的业务
- 扩展性提升
- 无需改变或改造业务应用代码
- 降低控制平面的节点数和服务依赖

运维价值

- 降低大规模集群的运维难度
- 可通过HDFS API和Distcp进行快速迁移
- 降低系统恢复时间
- 尽可能的减少NN Java GC带来的无响应问题



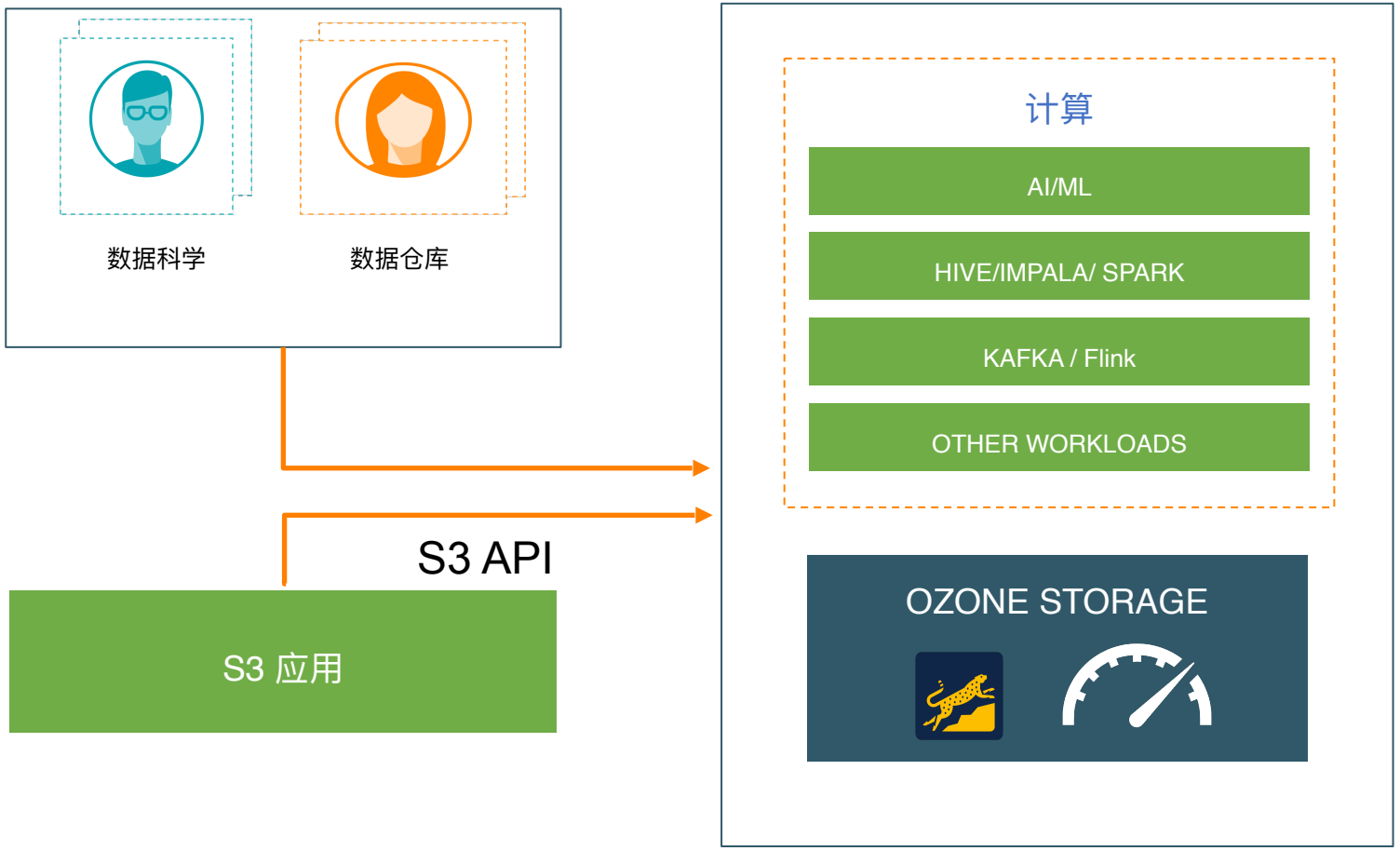
Apache Ozone – 使用场景 #2

业务价值

- 可以快速的对接已适配S3 接口的应用
- 减少数据在多个平台间的迁移
- 使用单一的API协议来应对混合云架构

运维价值

- 集约化的一套存储来面向不同的业务负载
- 更易于运维的控制面
- 只需要一个运维团队而不是多个



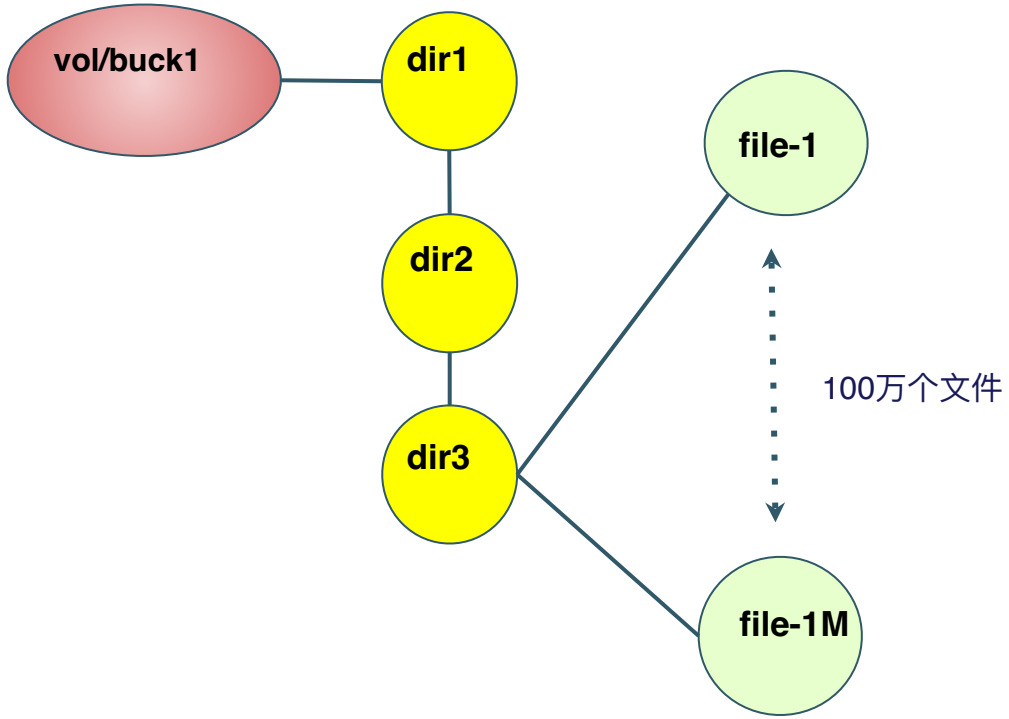
目录

- Apache Hadoop HDFS面临的问题
- Apache Ozone介绍
- Apache Ozone适用场景
- Apache Ozone的最近进展
- Apache Ozone的实践分享

新进展

- 文件系统优化(FSO)
- Ozone Balancer
- 纠删码
- 单数据盘单RocksDB实例

文件系统优化(FSO)



对象存储：采用 **KV** 方式管理对象元数据，无需管理元数据之间的关系

文件系统：额外地，需要采用**树结构**作为索引，管理元数据之间的关系

Ozone Key的存储

Key entry
/vol/buck1/dir1/
/vol/buck1/dir1/dir2/
/vol/buck1/dir1/dir2/dir3/
/vol/buck1/dir1/dir2/dir3/file-1
/vol/buck1/dir1/dir2/dir3/file-2
/vol/buck1/dir1/dir2/dir3/file-3
……..
/vol/buck1/dir1/dir2/dir3/file-n

目录

文件

删除/重命名目录 耗时

文件系统优化

引入Bucket级别 OM Metadata Layout 版本号

- **FILE_SYSTEM_OPTIMIZED (FSO)** ： 支持纯粹的文件语义，有限的 S3 兼容性

  文件的存储Key格式: “<parent unique-id>/<filename>”.

  例如, “1026/file-1”

- **OBJECT_STORE (OBS)** : key-value 存储, 纯粹的S3 对象存储语义

  对象的存储Key格式 : <keyname>

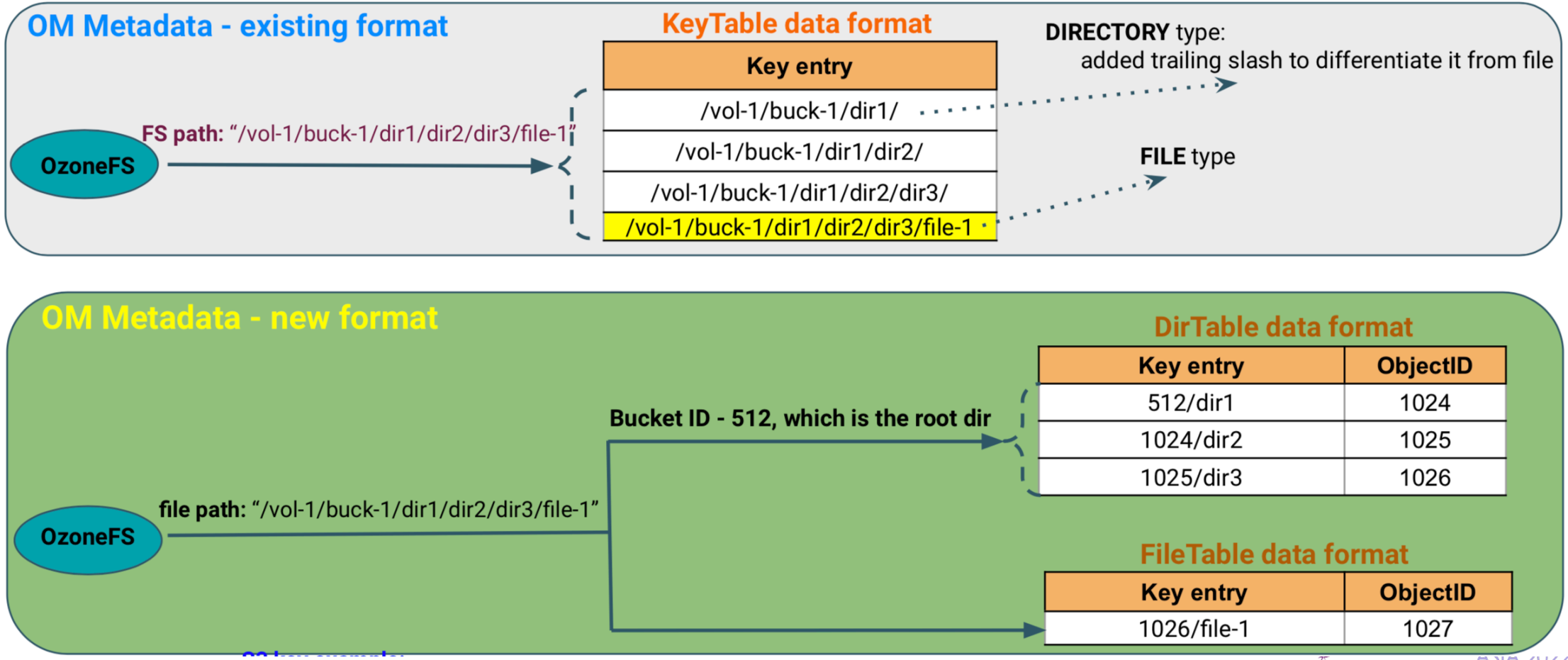
  例如, “/vol-1/buck-1/dir1/dir2/dir3/file-1”

- **LEGACY**: 所有已存在的桶，升级后变成LEGACY 版本，以支持向后兼容

  存储Key格式基本同**OBS,** 通过配置项区分偏向文件，还是偏向S3对象的支持

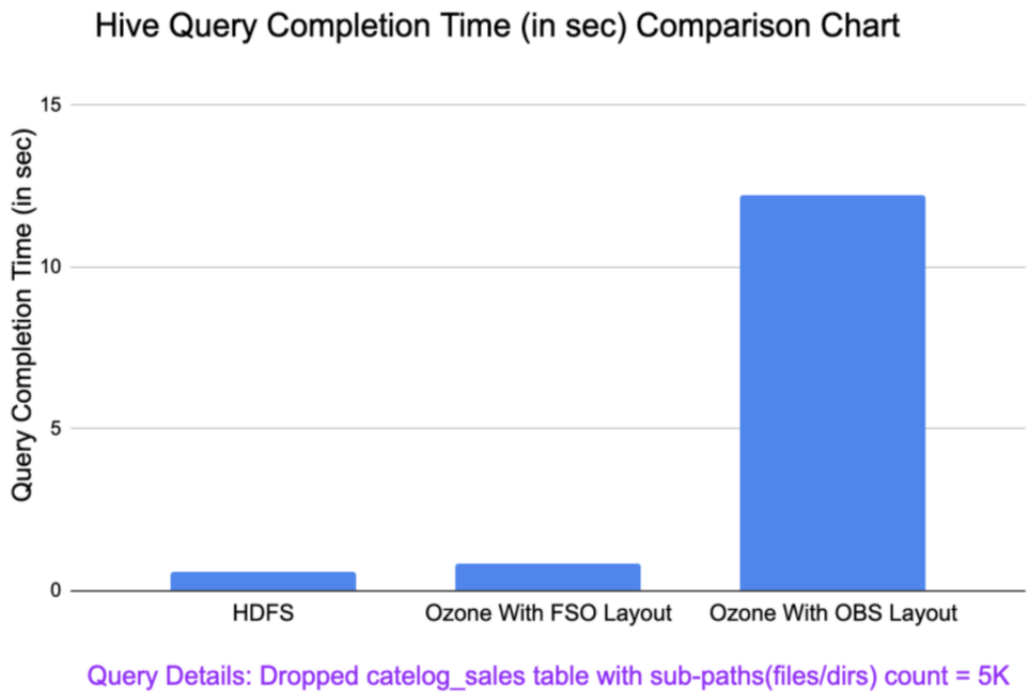
文件系统优化

Proposal : KeyTable ⟶ DirTable & FileTable



文件系统优化效果

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 删除表(Rename操作)

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

举例: *fs.delete(“<prefix_path>/catelog_sales”)*

均衡器Ozone Balancer(HDDS-4656)

时机

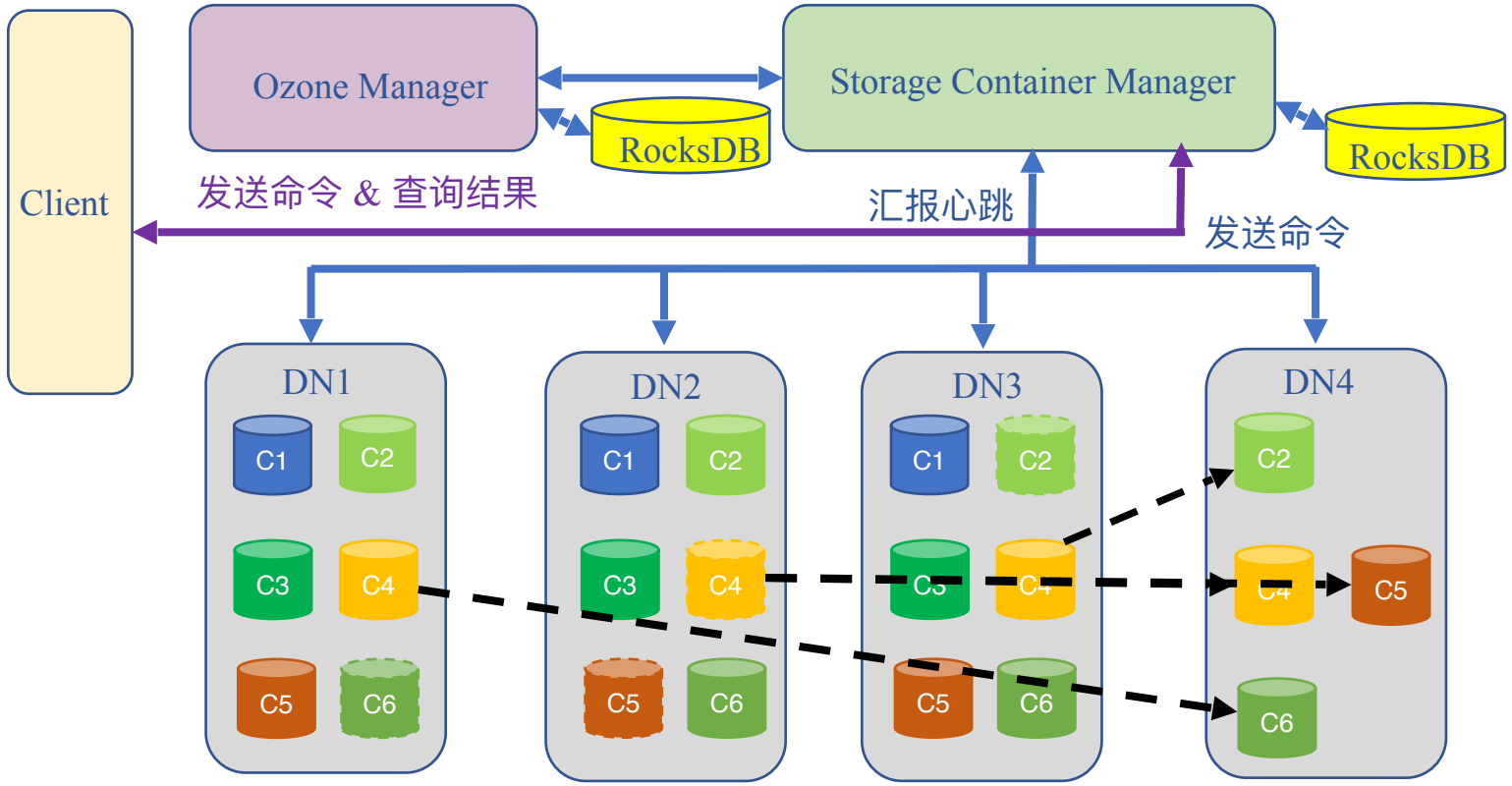
- 新的节点加入Ozone集群
- 删除大量数据后

好处

- 充分利用集群资源
- 均衡集群IO访问

实现

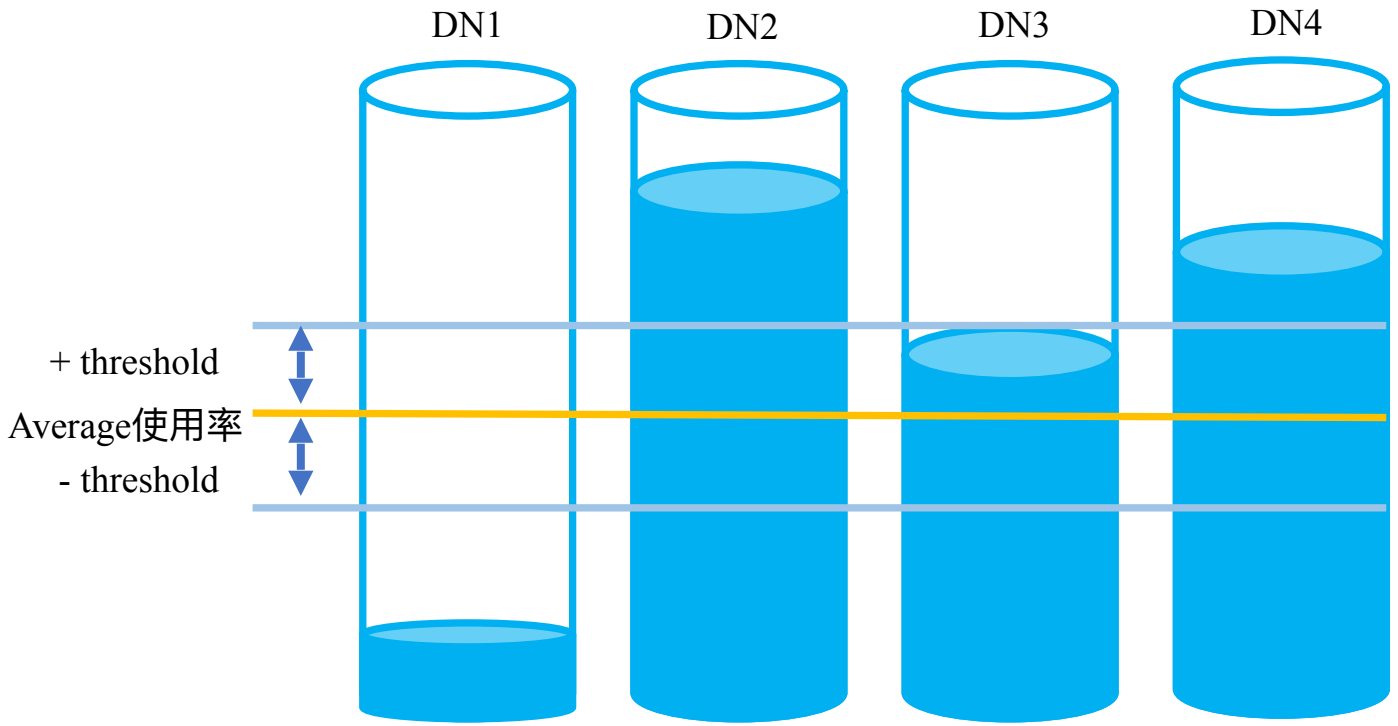
- 均衡器实现为SCM的子功能
- Container是数据迁移的最小单位，只迁移CLOSE状态的Container
- 客户端发送命令给SCM，SCM负责执行和控制流程



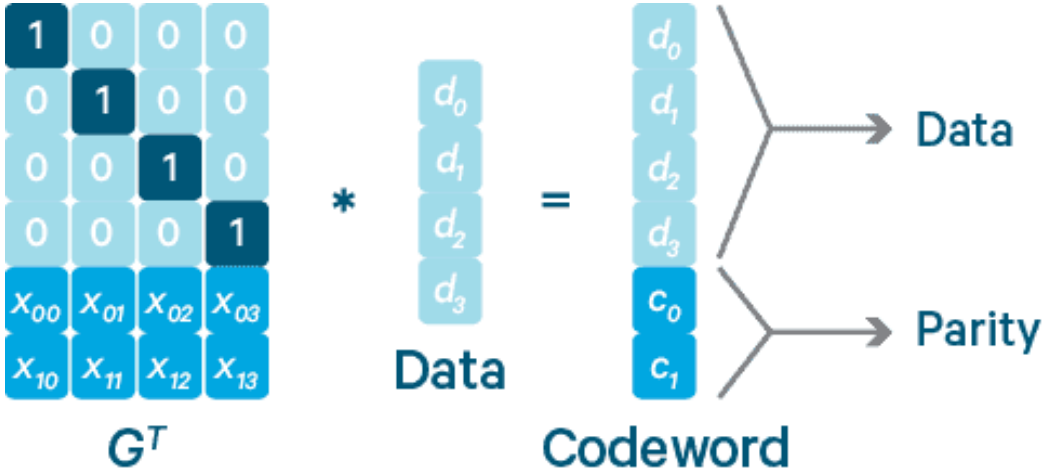
均衡器Ozone Balancer

主要配置项

- 启动服务
- 停止服务
- Threshold配置
- 最多连续迭代运行次数
- 每次迭代最大迁移数据量



纠删码(HDDS-3816)



以计算为代价，在不降低数据可靠性的同时，降低数据存储成本

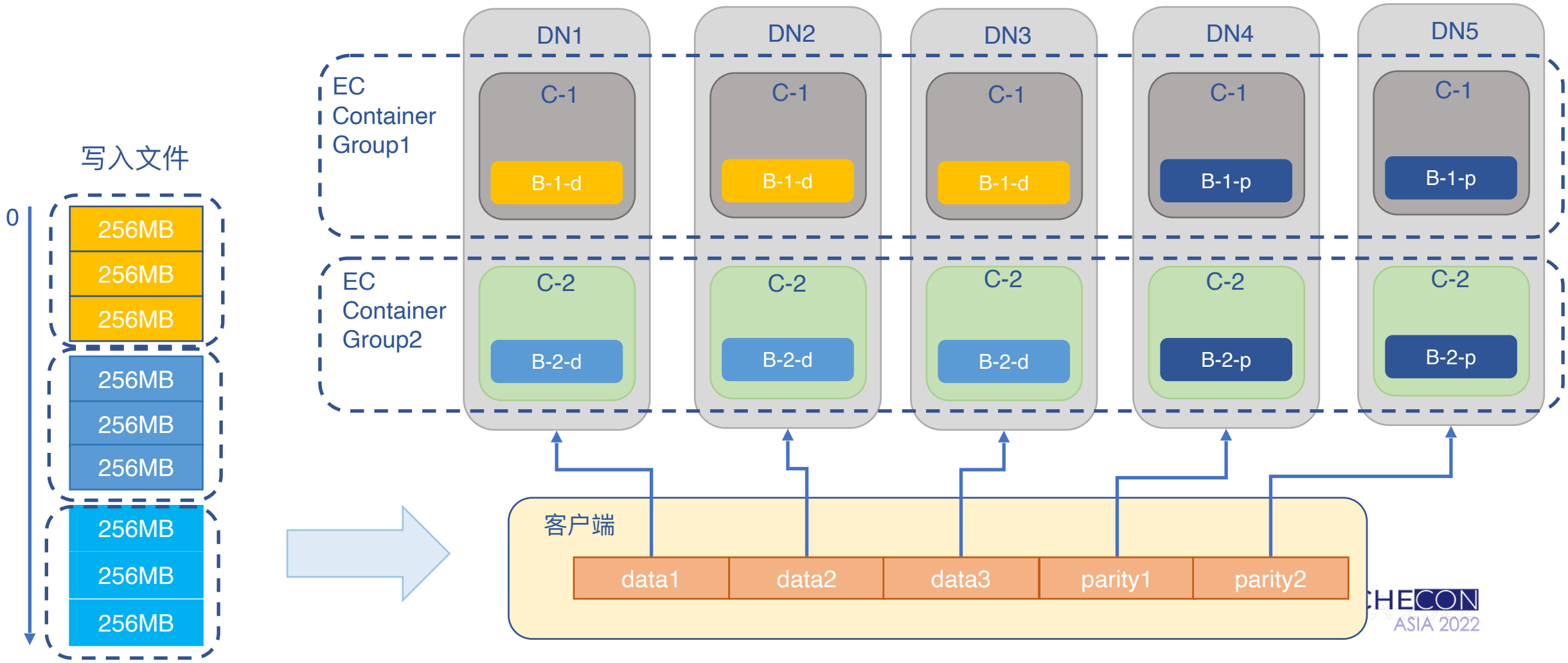
	数据可靠性 (越高越好)	存储效率 (越高越好)
1-replica	0	100%
3-replica	2	33%
EC RS(6,3)	3	67%
EC RS(10, 4)	4	71%
EC RS(3,2)	2	60%

数据可靠性 vs. 存储效率

纠删码策略

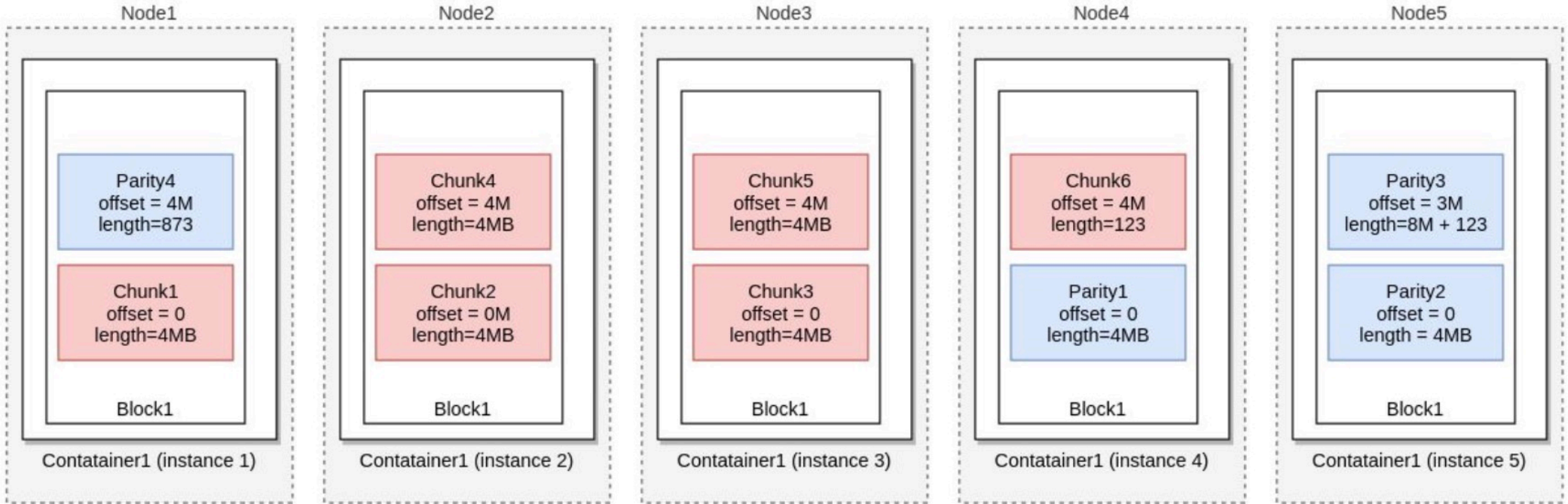
- 内建支持的策略
  - RS-3-2-1024K
  - RS-6-3-1024K
  - XOR-2-1-1024K
- 可定制新的策略
- 策略设置支持
  - 全局策略设置
  - 桶级别策略设置
  - 对象/文件级别策略设置

数据写入

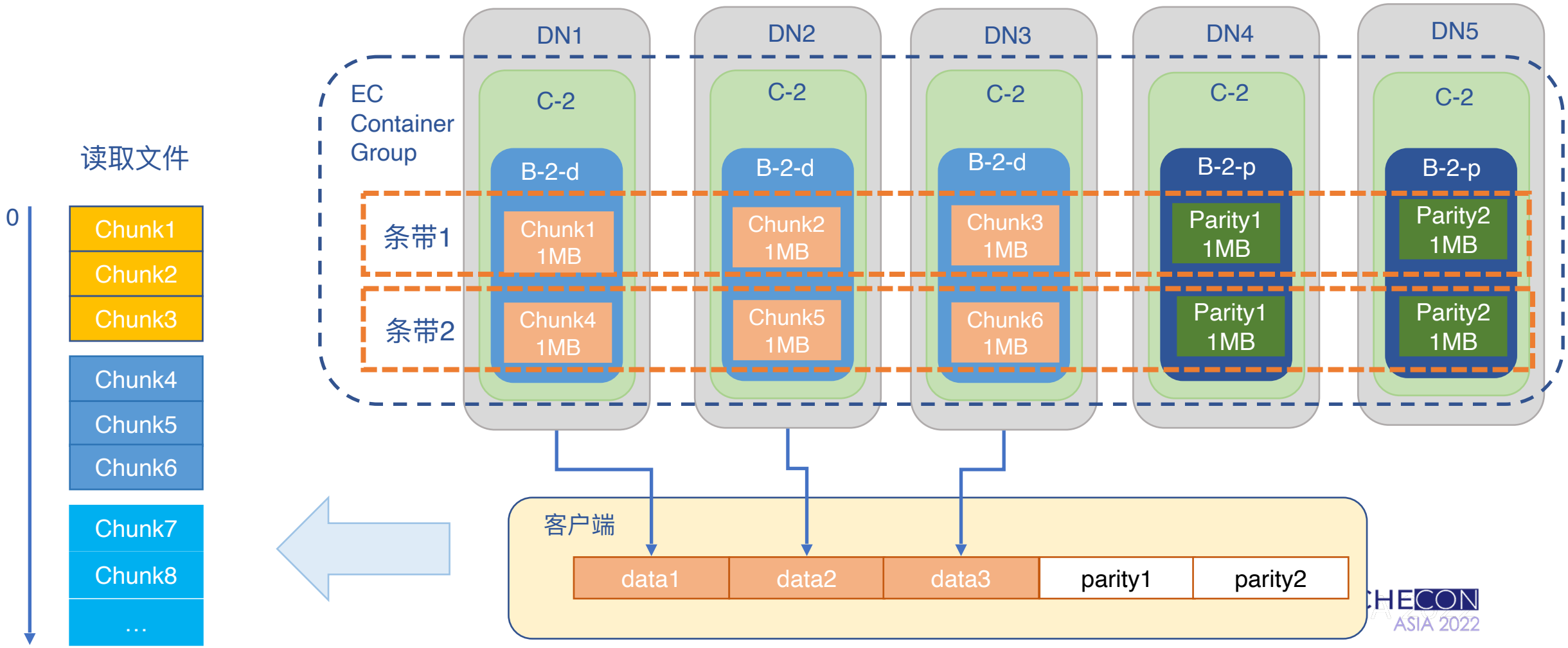


数据写入

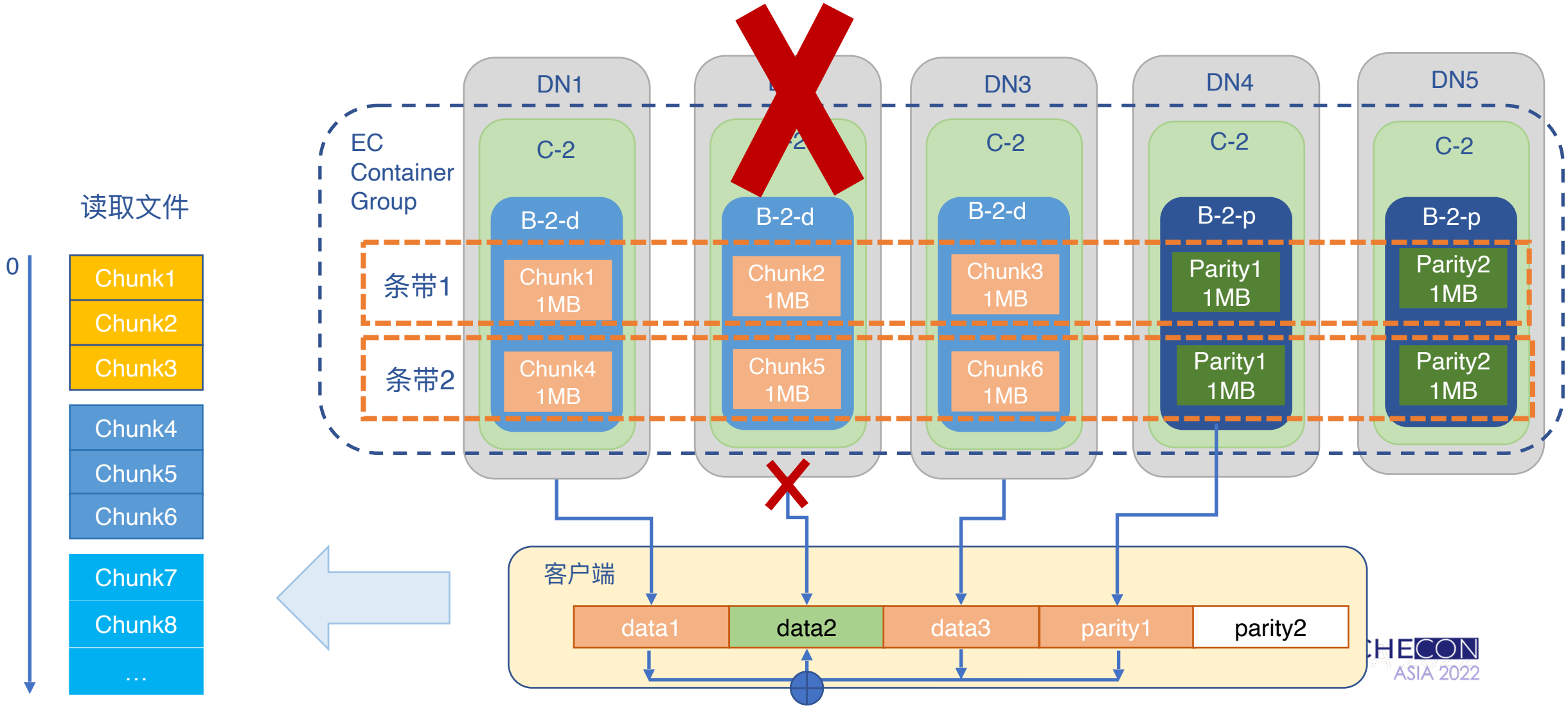
- EC Container Group：给定Container的一组满足EC策略的副本实例
- 物理块：每个DN磁盘上的数据块，默认是256MB
- 逻辑EC块：属于单个条带，满足EC策略的一组数据块。例如EC-3-2，一个逻辑块3*256MB大小
- 条带粒度：条带的粒度默认1MB，可配置



数据读取

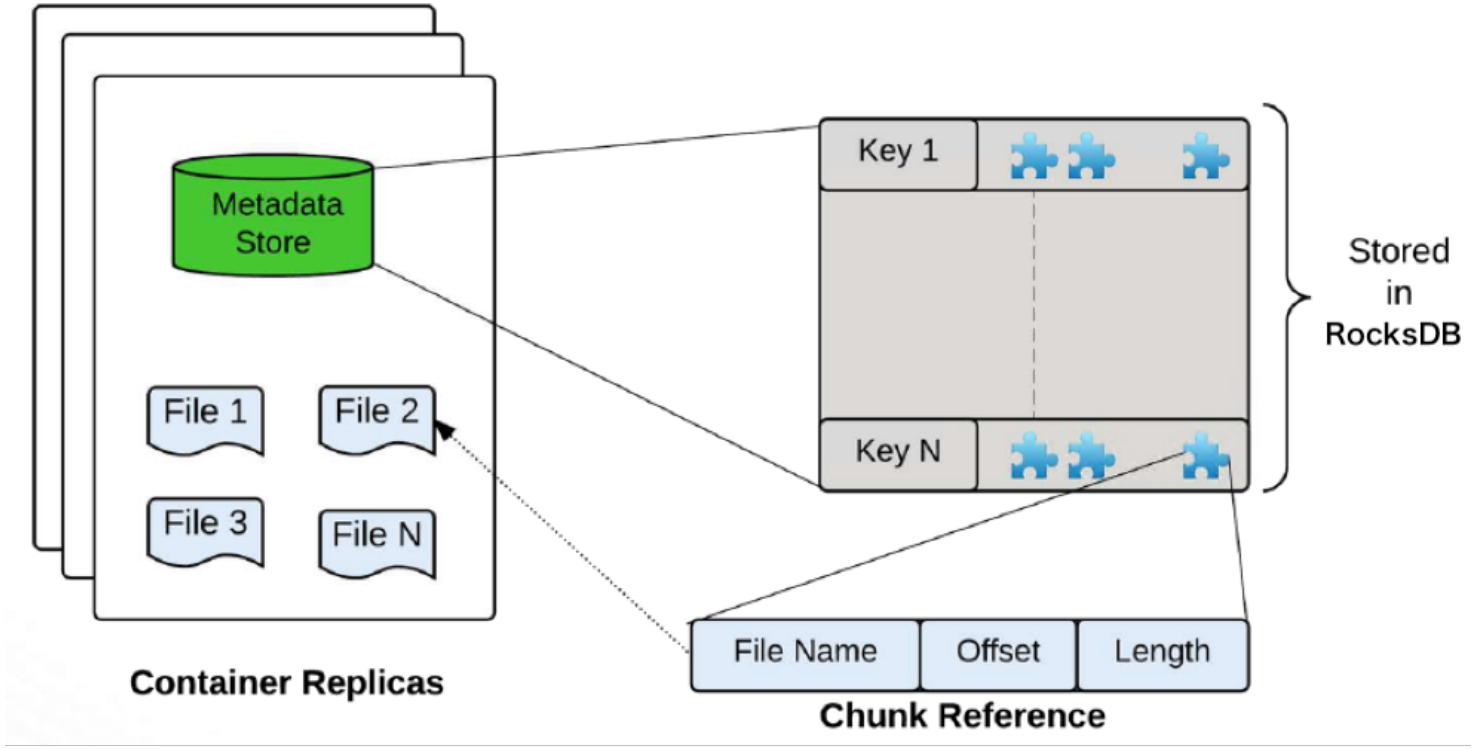


数据读取在线恢复



单盘单RocksDB实例(HDDS-3630)

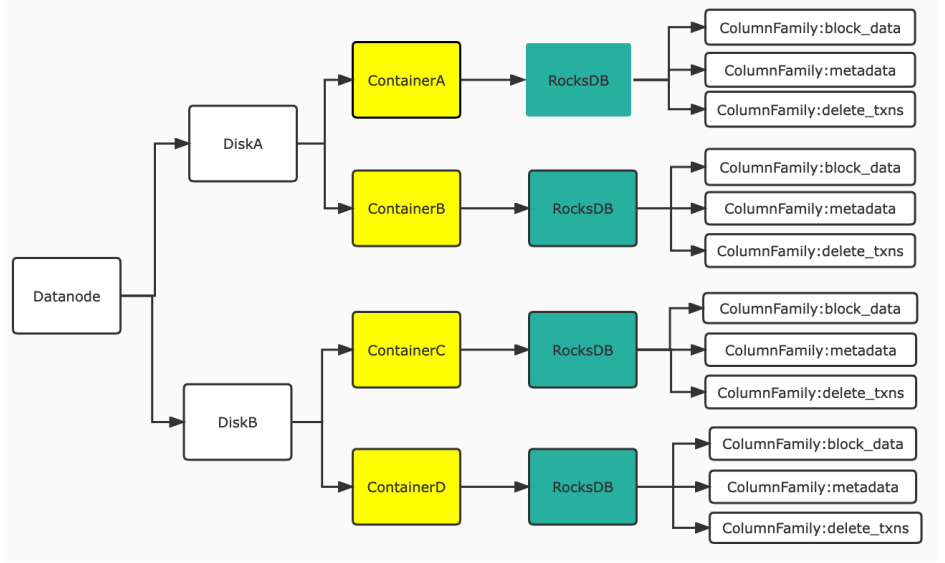
当前 - 每个Container的元数据保存在独立RocksDB实例中

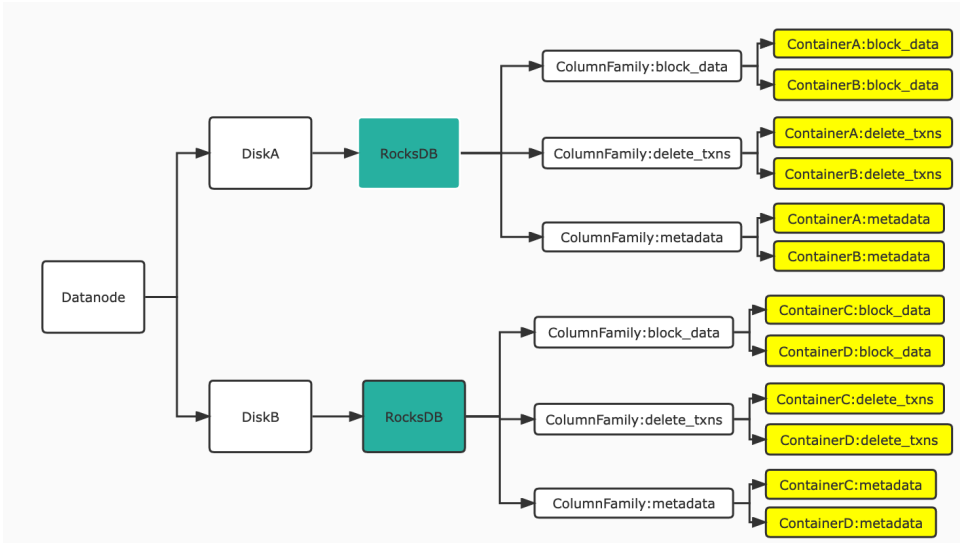


单盘单RocksDB实例

问题

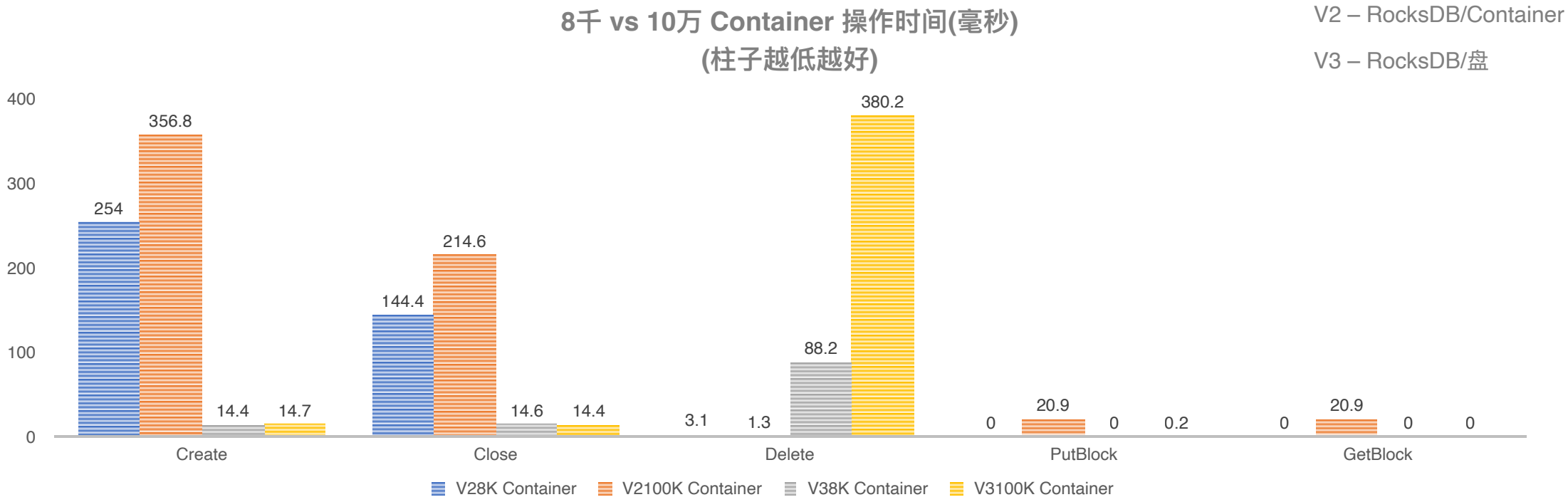
- 大容量磁盘，系统中有上万个Container和RocksDB实例
- 内存开销大，需保留众多RocksDB实例
- 性能影响，频繁open/close实例
- 磁盘使用量，不可精准预测
- 稳定性，频繁open/close非RocksDB的推荐用法，容易触发潜在问题





解决办法 - 单盘单RocksDB实例

单盘单RocksDB实例

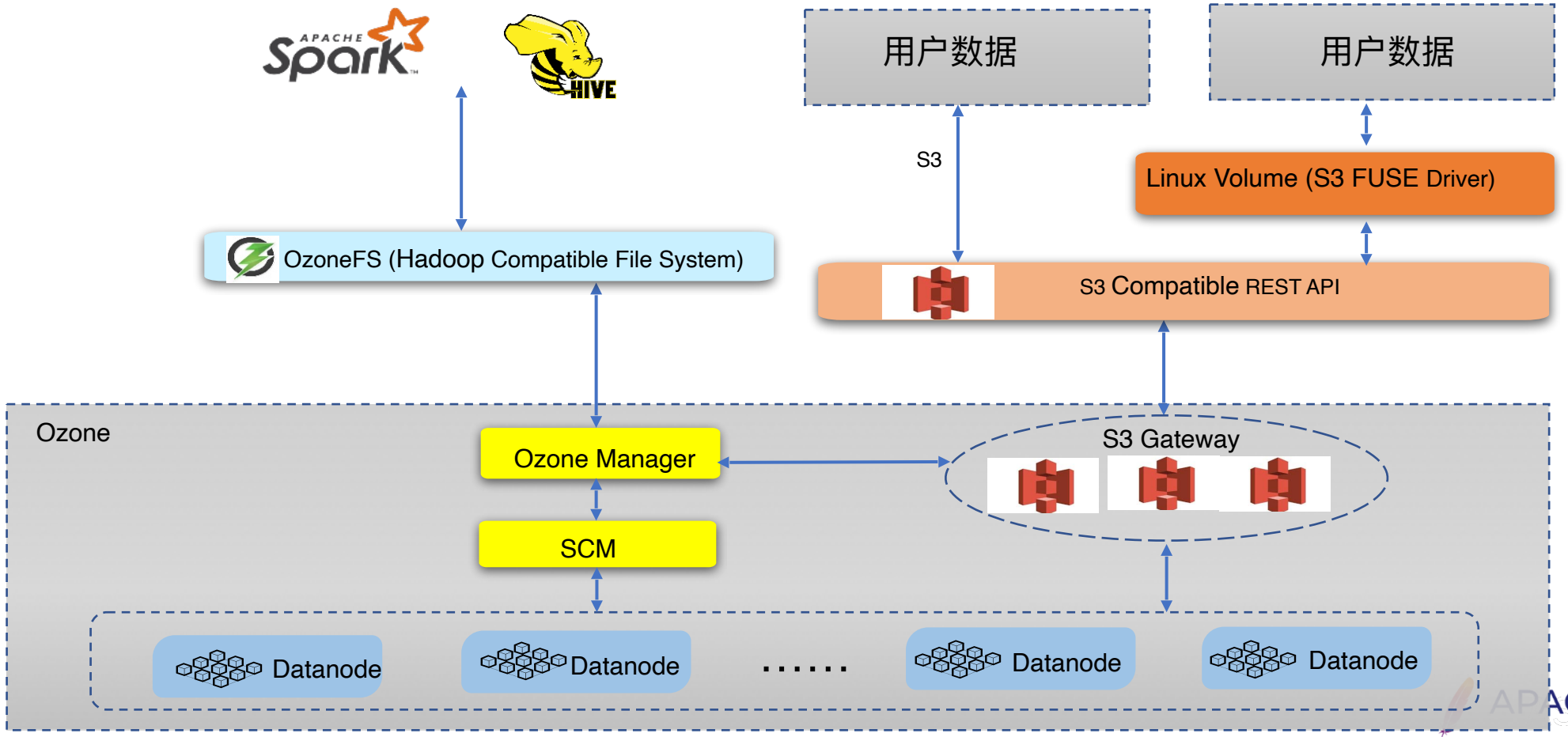


- 除了Container的删除，其他操作V3都要比V2有数量级的提升
- 随着单盘Container数量的增多，V2的各操作性能出现下降，而V3性能基本没有变化

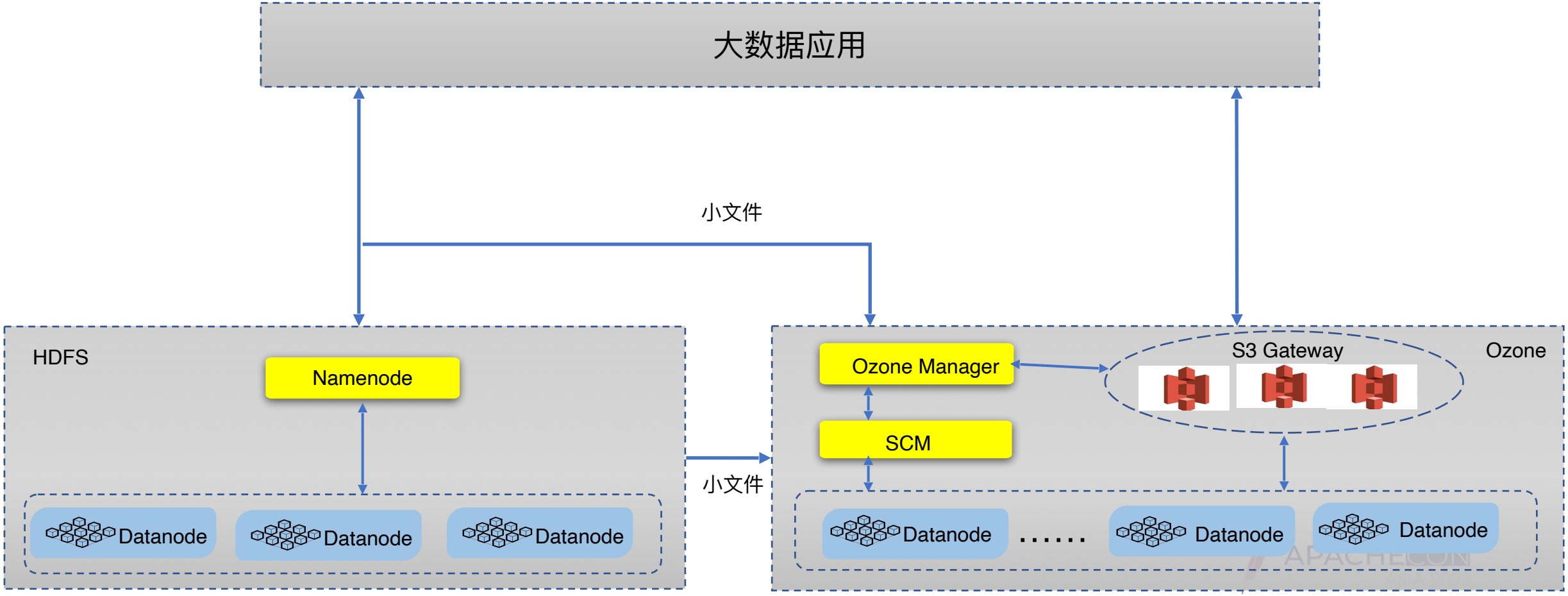
目录

- Apache Hadoop HDFS面临的问题
- Apache Ozone介绍
- Apache Ozone适用场景
- Apache Ozone的最近进展
- Apache Ozone的实践分享

实践一



实践二


大数据应用
小文件
HDFS
Namenode
Datanode
Datanode
Datanode
小文件
Ozone Manager
SCM
S3 Gateway
Ozone
Datanode
.....
Datanode
Datanode

Thank You