UCLOUD

基于cephfs的改进及优化——分布式缓存实践

深信服科技股份有限公司 卢波

UCLOUD

目录

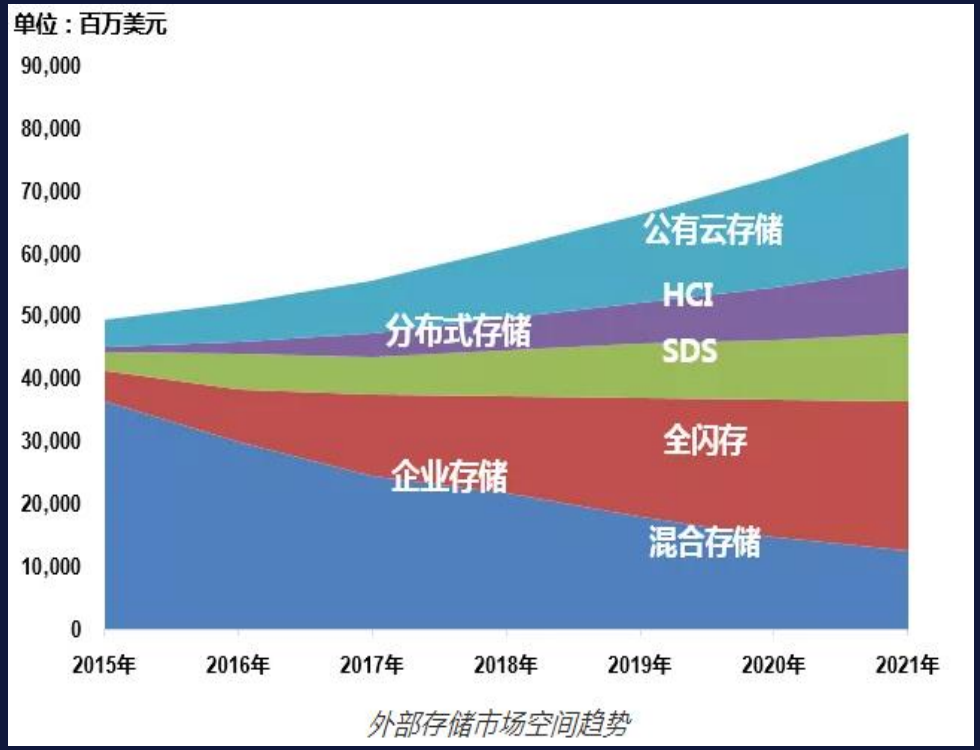
01

背景

背景

UCLOUD

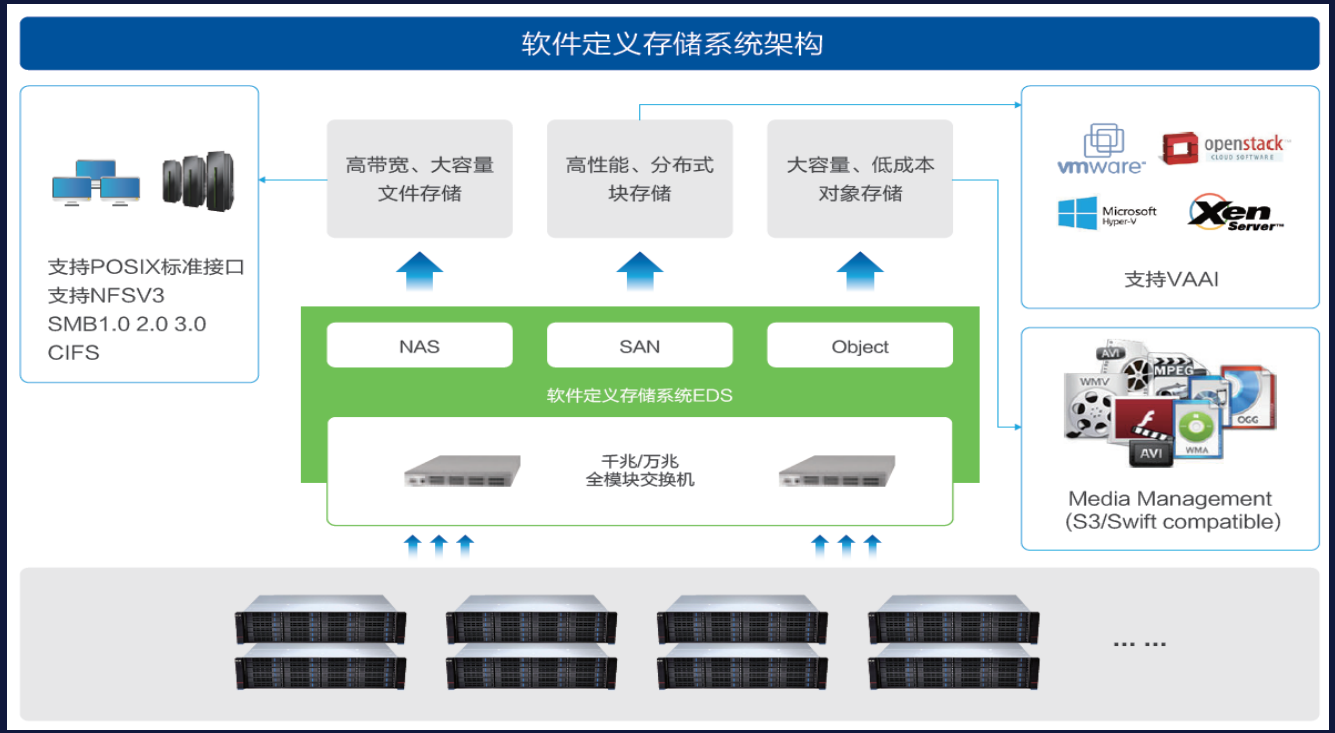
据IDC的调查报告显示，企业中80%的数据都是非结构化数据，这些数据每年都按指数增长60%。
分布式文件存储以其灵活扩展、快速部署等特点越来越受到政府、教育、医疗等行业用户的青睐。



外部存储市场空间趋势

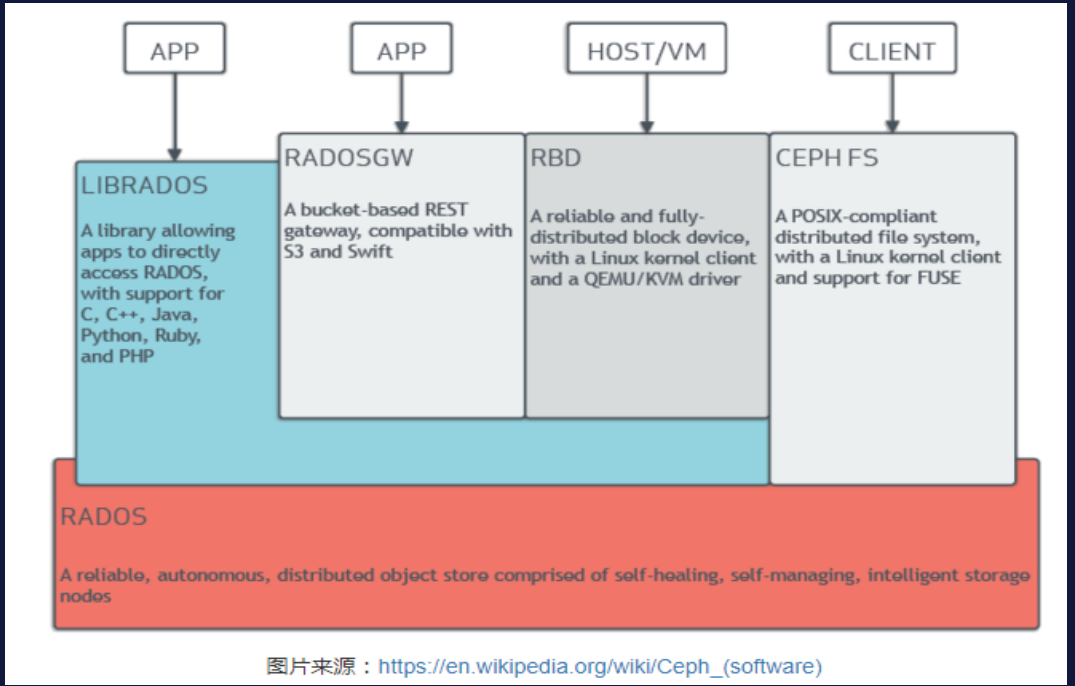
海量非结构化存储

UCLOUD



Ceph 是一个分层的架构，底层是一个基于 CRUSH（哈希）的分布式对象存储---RADOS

上层提供对象存储（RADOSGW）、块存储（RDB）和文件系统（CephFS）三种访问方式



图片来源：https://en.wikipedia.org/wiki/Ceph_(software)

UCLOUD

02

cephfs

单机文件系统

共享：无法同时为分布在多个机器中的应用提供访问，于是有了 NFS 协议
容量：无法提供足够空间来存储数据，数据只好分散在多个隔离的单机文件系统里。
性能：无法满足某些应用需要非常高的读写性能要求，应用只好做逻辑拆分同时读写多个文件系统。
可靠性：受限于单个机器的可靠性，机器故障可能导致数据丢失。
可用性：受限于单个操作系统的可用性，故障或者重启等运维操作会导致不可用。

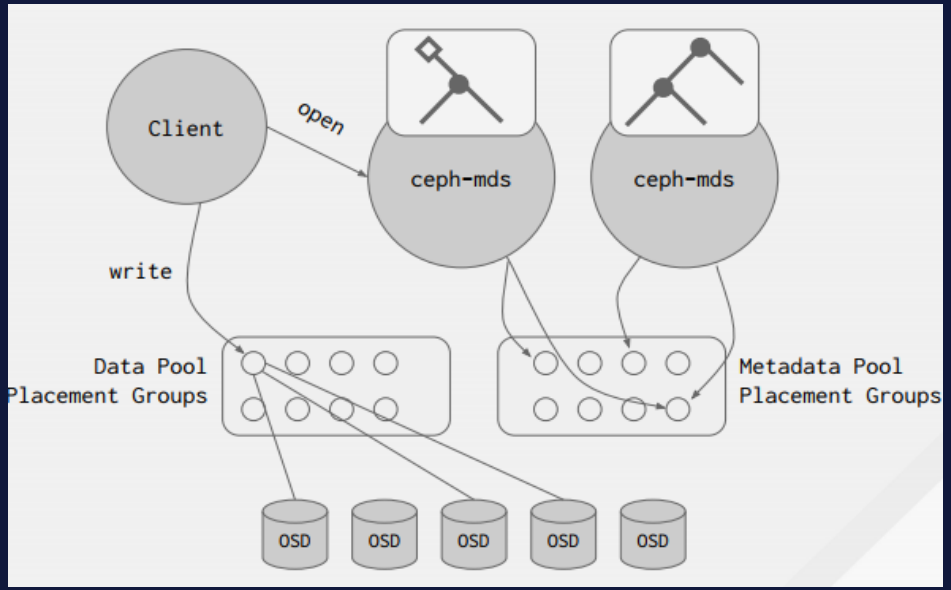
CephFS 始于 Sage Weil 的博士论文研究，目标是实现分布式的元数据管理以支持 EB 级别数据规模。2012年，Sage Weil 成立了 InkTank 继续支持 CephFS 的开发
于 2014年被 Redhat 收购
2016 年，CephFS 发布可用于生产环境的稳定版（单MDS）

cephfs架构

ceph-mds: 缓存文件系统元数据，并将元数据持久化到RADOS中，自身运行在rados之上，本身并不存储数据。
open：客户端从MDS获取文件元数据
write：客户端直接通过RADOS进行数据写入操作

所有数据的操作时通过客户端直接访问raods的，多客户端的数据访问操作时依靠OSD来控制的

元数据和数据可以在不同的存储池中

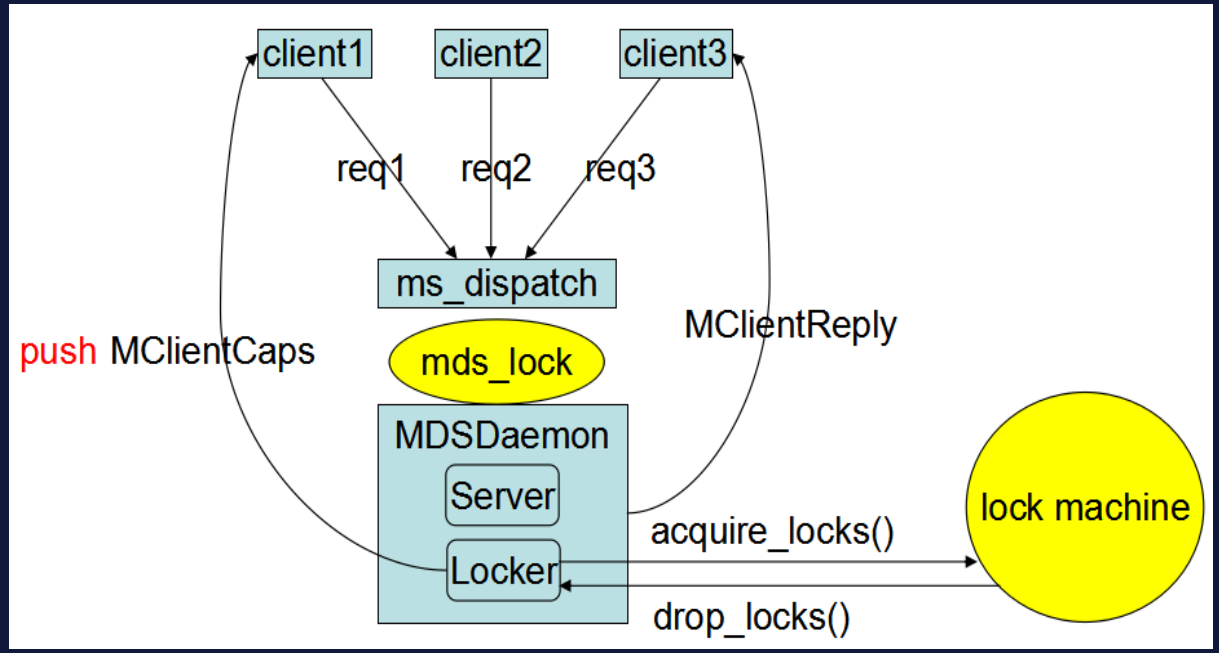


MDS的作用

MDS作为元数据的cache，同时控制元数据的并发处理：

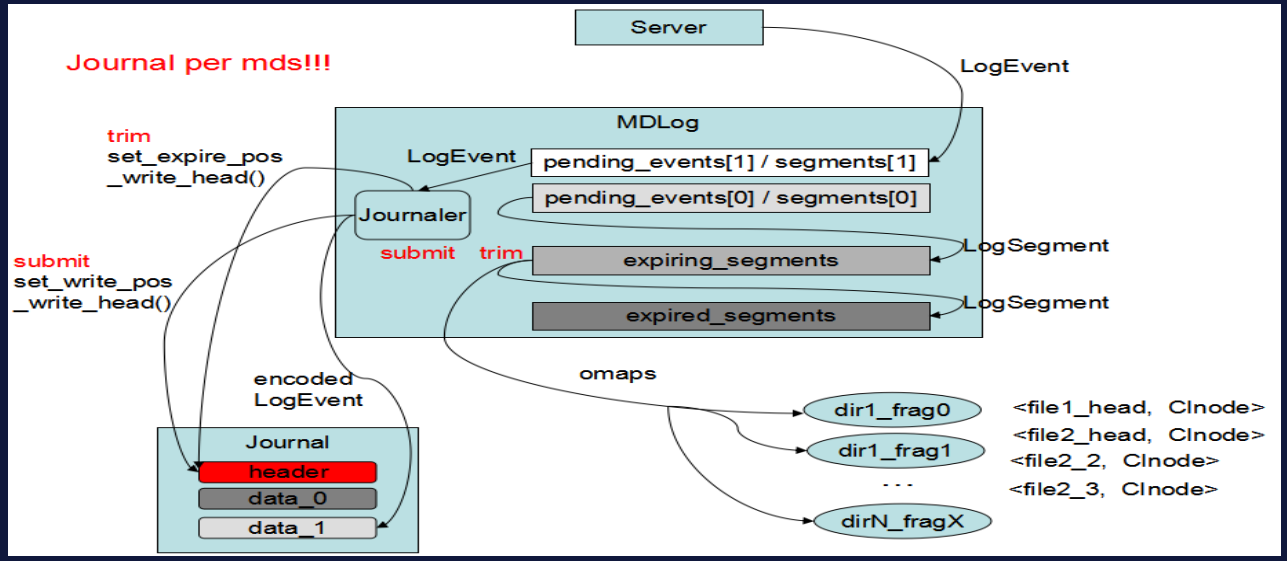
- 提供文件资源
  - 以文件的形式访问数据
  - POSIX兼容
- 文件系统命名空间
  - 提供目录树
- 管理文件元数据
  - 分配inode/dentry
- 持久化文件元数据
  - 存储inode/dentry到osd

MDS并发控制



drop_locks(): 当Server完成op后，释放加在parent_dir和inode上的锁

元数据持久化



1. Server 将 EUpdate() EOpen() 等LogEvent 提交到pending_events队列
2. submit_thread定期从pending_events队列中获取LogEvent, 并使用Journaler进行flush 到 Journal存储区域，把LogEvent中的CInode, CDir, CDentry加入到对应的LogSegment
3. tick线程定期将segments写入到持久存储，当LogSegment所有的LogEvent的所有CInode, CDir持久化完成后，将LogSegment从expiring_segments移动到expired_segments，最后更新Journal头部信息

UCLOUD

03

优化实践

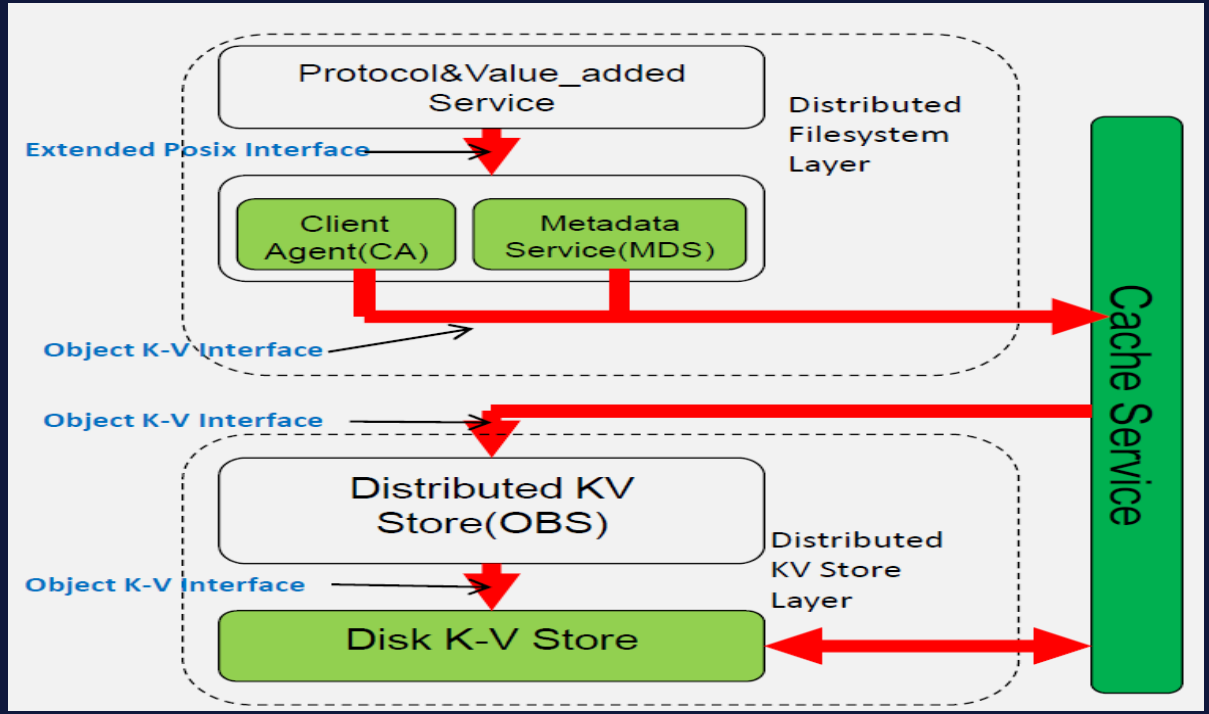
设计目标

- 解决小IO的写放大问题
- 当前Ceph的IO路径过长，延时高
- 提升文件的元数据写操作的性能
- 支持write buffer 写入
- 支持读预取
- 充分利用SSD的性能优势

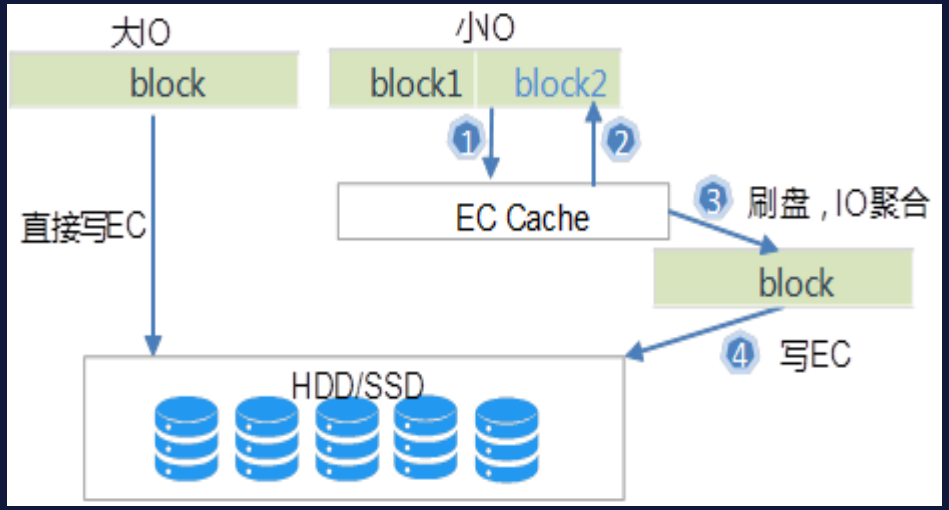
OceanStor 9000

UCLOUD

缓存是整系统全局共享的，即只要缓存在任意一个节点上的文件分条数据，其它任意节点再次收到该数据的访问请求后都可以从一级缓存中命中该数据。
通过全局缓存，实现数据合并，利用K-V存储的高吞吐，从而大大提高系统整体性能。



FusionStorage



FusionStorage块存储根据业务不同的IO大小，智能地对不同大小的IO采取不同的处理方式。如下图所示，对于小块IO，FusionStorage块存储采用多副本的方式写入分布式EC Cache中，并在Cache中做条带聚合；而对于大块IO，则绕过分布式EC Cache，直接提交EC写入后端硬盘。由于大块IO直接下盘，系统可以释放原来大块IO占用的宝贵的Cache资源，缓存更多的随机小块I/O，间接的提高了随机小块I/O的Cache命中率，提升系统随机小IO的性能。而HDD在写入大块顺序IO时，写性能差距相比SSD并没有那么明显，加上多块HDD并发处理，在大块顺序IO的场景下系统也能获得很好的写入带宽，兼顾了系统的整体性能。

Isilon

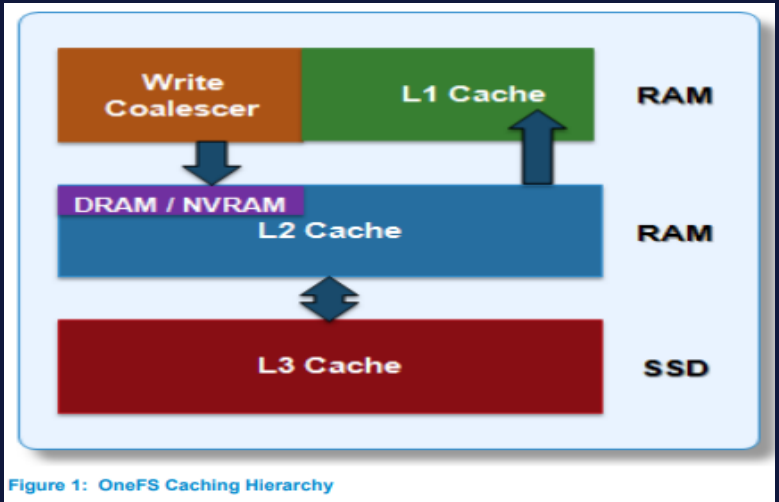


Figure 1: OneFS Caching Hierarchy

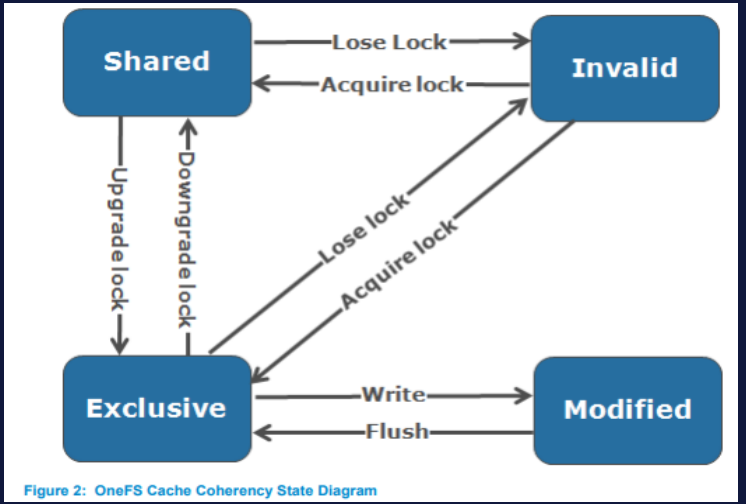
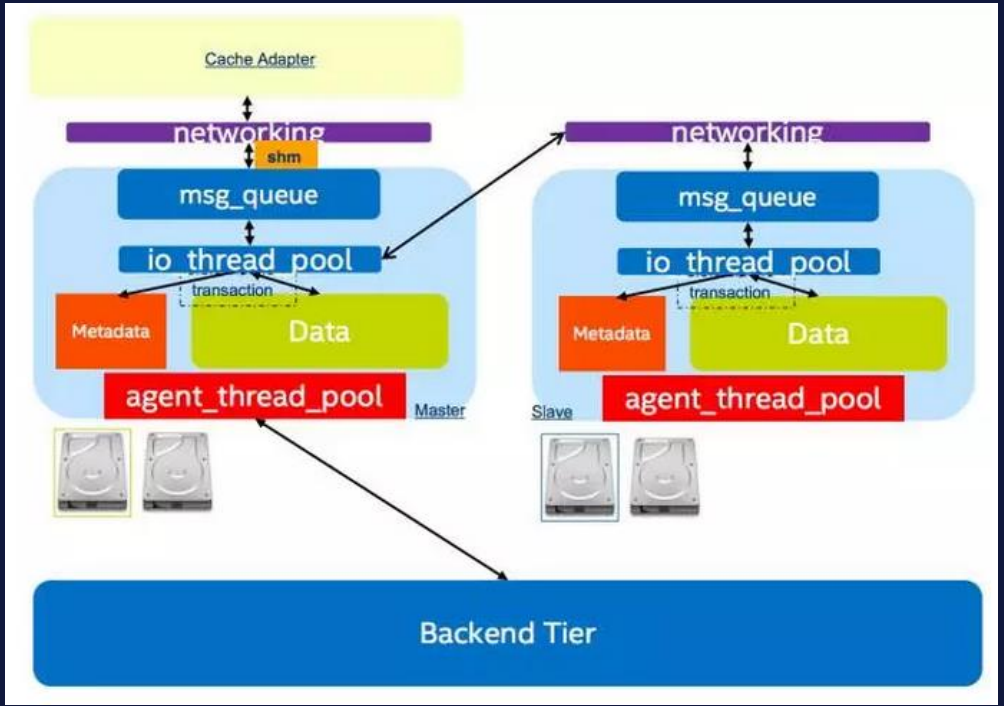


Figure 2: OneFS Cache Coherency State Diagram

Intel

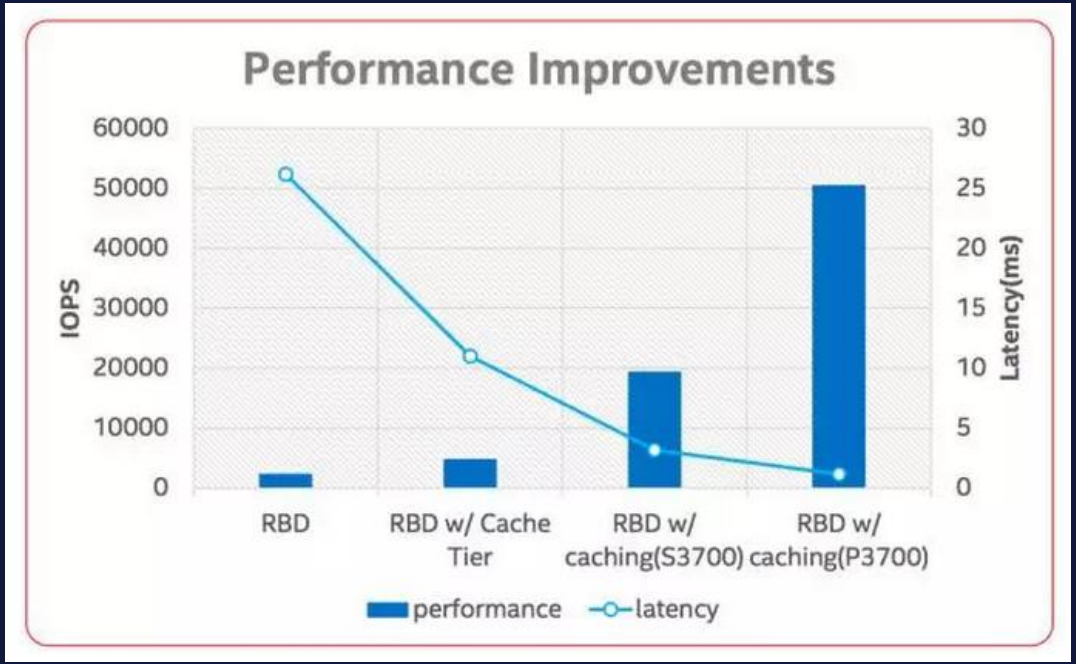
Intel 提出的 Hyper Converged Cache 是不错的亮点，主要思路是在 Client Side 有 SSD 来进行读写加速，将后端 Ceph 作为一个慢存储来使用。在 RBD，RGW 场景中，作为读缓存问题不大，主要是期望能不能作为写缓存提供，这就涉及到写缓存的冗余问题。因此在设计中也提到了类似 DRBD 的双拷贝架构



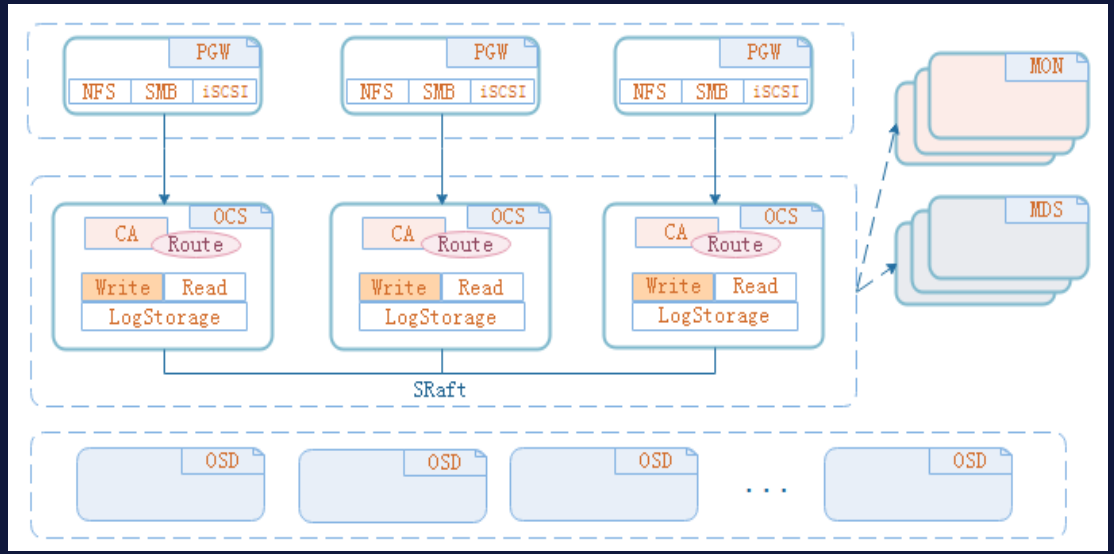
性能对比

Datrium公司开发的基于写缓存架构的产品性能提升明显，但是该架构会带来的问题：

- 整个分布式存储弱化后端持久化集群的空间
- 同时加大缓存层的设计，意味着缓存层要再实现一次众多分布式管理逻辑

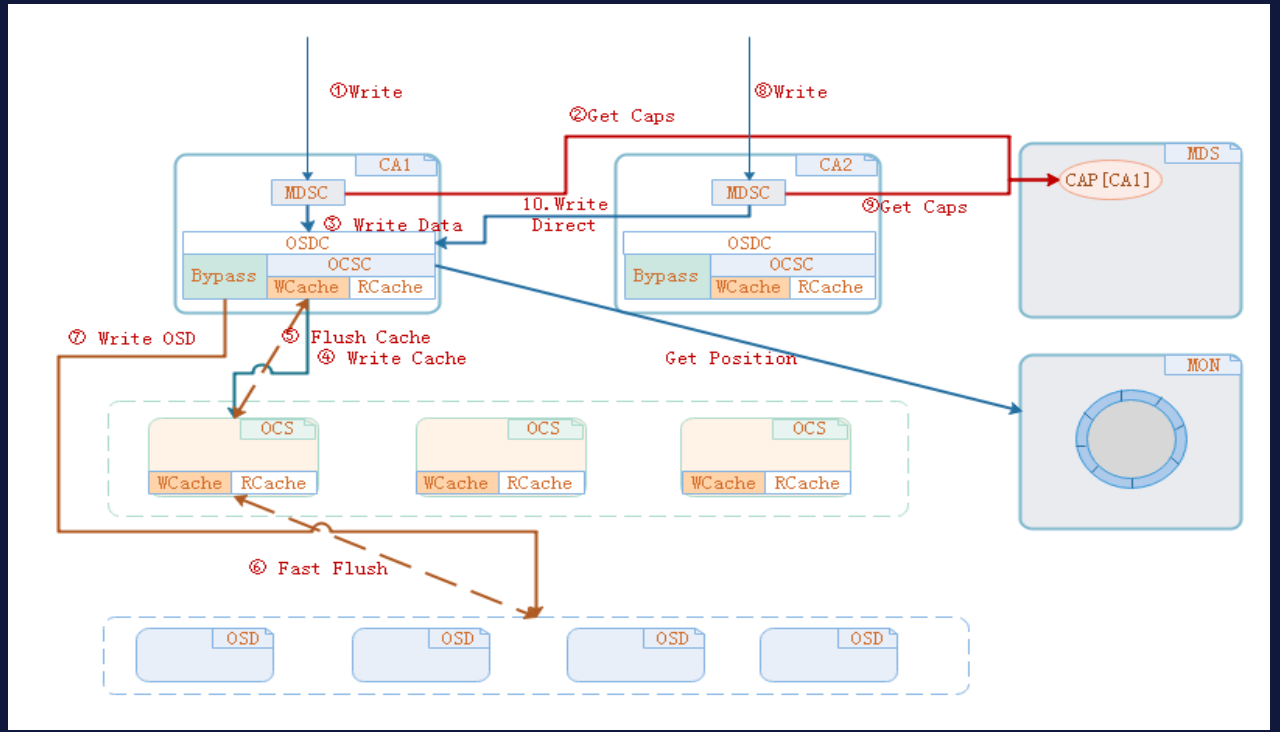


系统架构

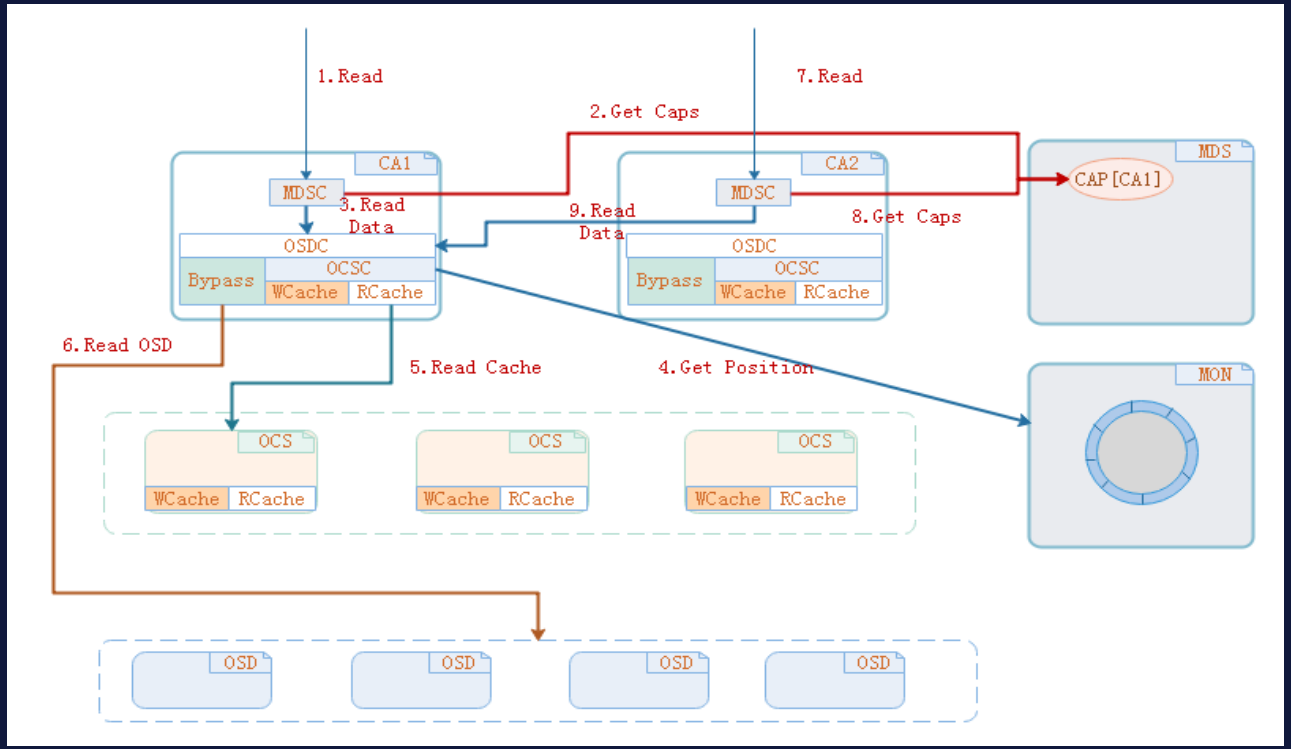


PGW（Protocol Gateway）是客户端访问的协议转换层，将外部的标准访问协议转换为EDS的内部访问协议，PGW也是整个EDS的前端，负责数据的IO接入。
MON（Monitor）EDS的协调控制器，监控和调度整个EDS
MDS（Metadata Server）文件系统的元数据访问层，同时也是分布式缓存的分布式锁和路由控制层
OSD（Object Storage Driver）对象存储设备，主要构建分布式对象存储服务
OCS（Object Cache Server）对象存储缓存服务，主要负责分布式缓存的逻辑处理

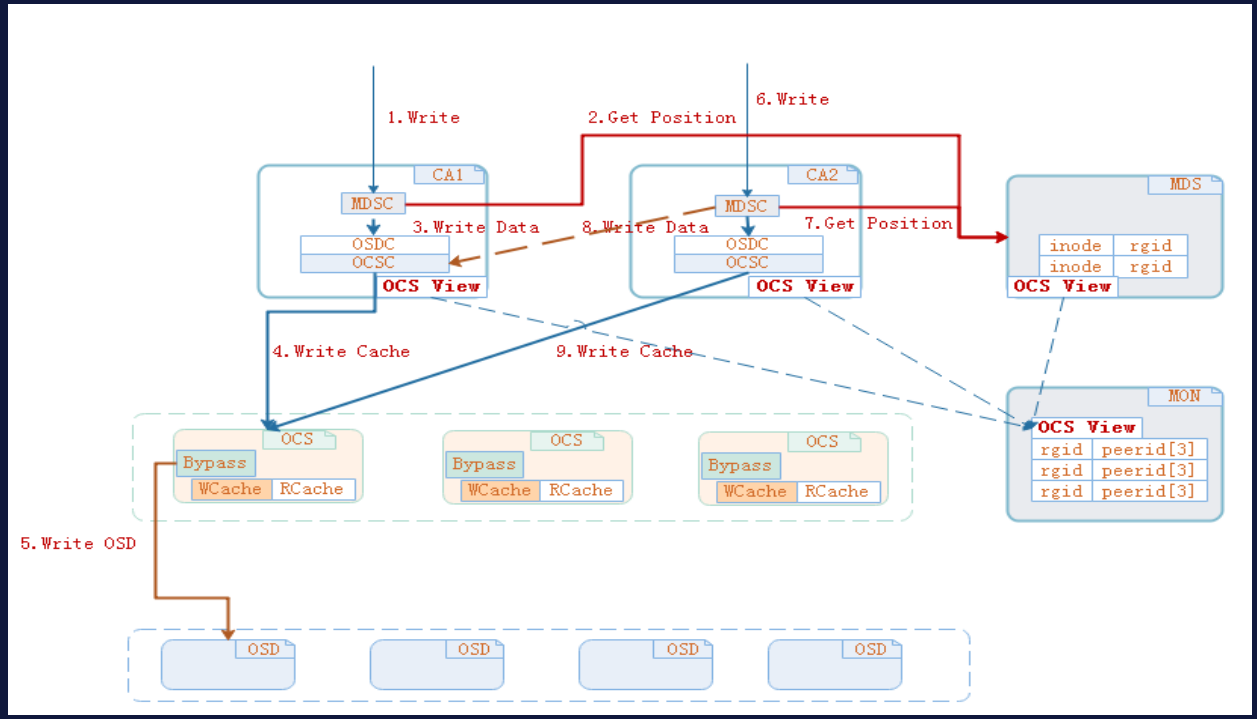
写流程--DHT


①Write
②Get Caps
⑧Write
CA1
MDSC
③ Write Data
10.Write Direct
CA2
MDSC
⑨Get Caps
MDS
CAP[CA1]
OSDC
Bypass
OCSC
WCache
RCache
OSDC
Bypass
OCSC
WCache
RCache
⑦ Write OSD
⑤ Flush Cache
④ Write Cache
Get Position
MON
OCS
WCache
RCache
OCS
WCache
RCache
OCS
WCache
RCache
⑥ Fast Flush
OSD
OSD
OSD
OSD

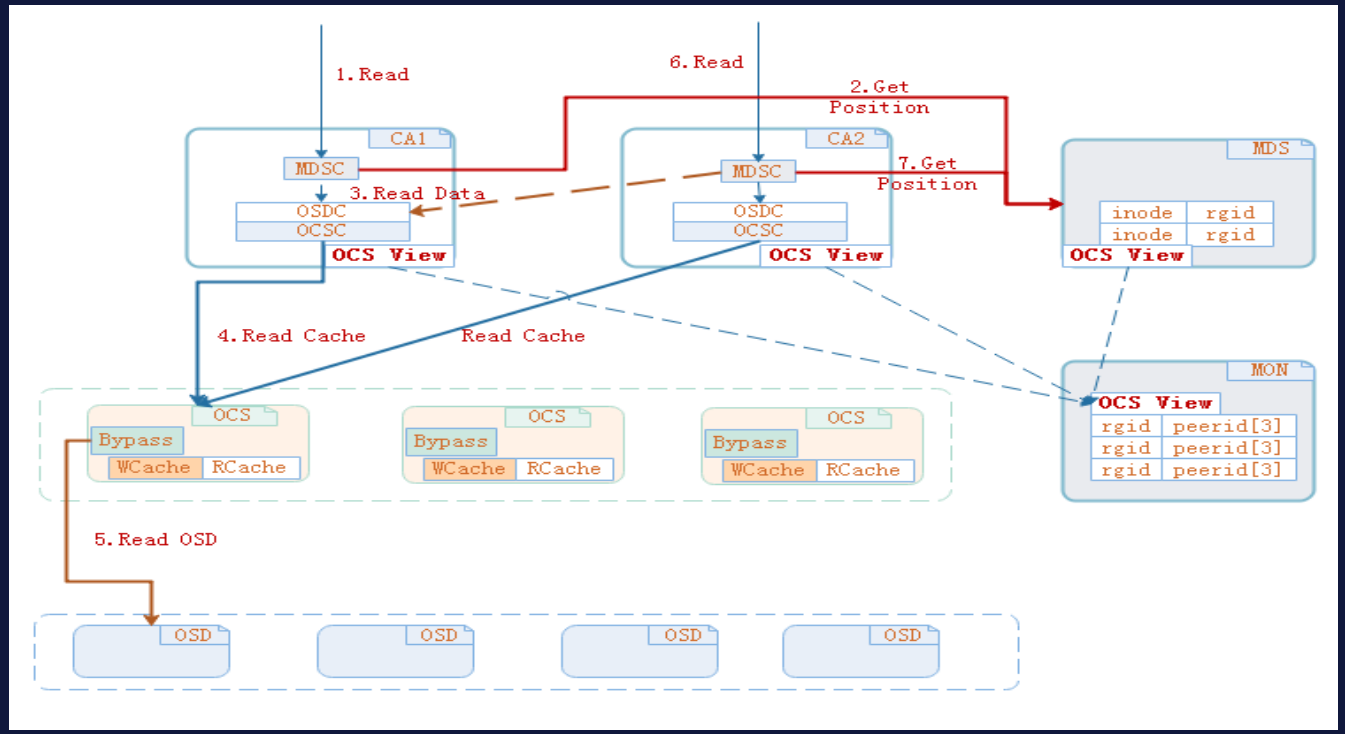
读流程--DHT


1.Read
7.Read
2.Get Caps
CA1
MDSC
3.Read Data
OSDC
Bypass
OCSC
WCache
RCache
CA2
MDSC
9.Read Data
8.Get Caps
OSDC
Bypass
OCSC
WCache
RCache
MDS
CAP[CA1]
6.Read OSD
5.Read Cache
4.Get Position
MON
OCS
WCache
RCache
OCS
WCache
RCache
OCS
WCache
RCache
OSD
OSD
OSD
OSD

写流程--查表


1.Write
2.Get Position
6.Write
CA1
MDSC
OSDC
OCSC
OCS View
CA2
MDSC
OSDC
OCSC
OCS View
MDS
inode rgid
inode rgid
OCS View
3.Write Data
8.Write Data
7.Get Position
4.Write Cache
9.Write Cache
OCS
Bypass
WCache RCache
OCS
Bypass
WCache RCache
OCS
Bypass
WCache RCache
MON
OCS View
rgid peerid[3]
rgid peerid[3]
rgid peerid[3]
5.Write OSD
OSD
OSD
OSD
OSD

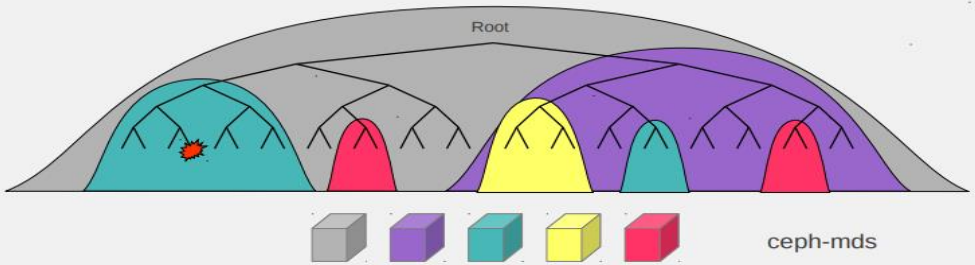
读流程--查表


1.Read
6.Read
2.Get Position
CA1
MDSC
OSDC
OCSC
OCS View
3.Read Data
CA2
MDSC
OSDC
OCSC
OCS View
7.Get Position
MDS
inode rgid
inode rgid
OCS View
4.Read Cache
Read Cache
MON
OCS View
rgid peerid[3]
rgid peerid[3]
rgid peerid[3]
OCS
Bypass
WCache RCache
OCS
Bypass
WCache RCache
OCS
Bypass
WCache RCache
5.Read OSD
OSD
OSD
OSD
OSD

04

展望

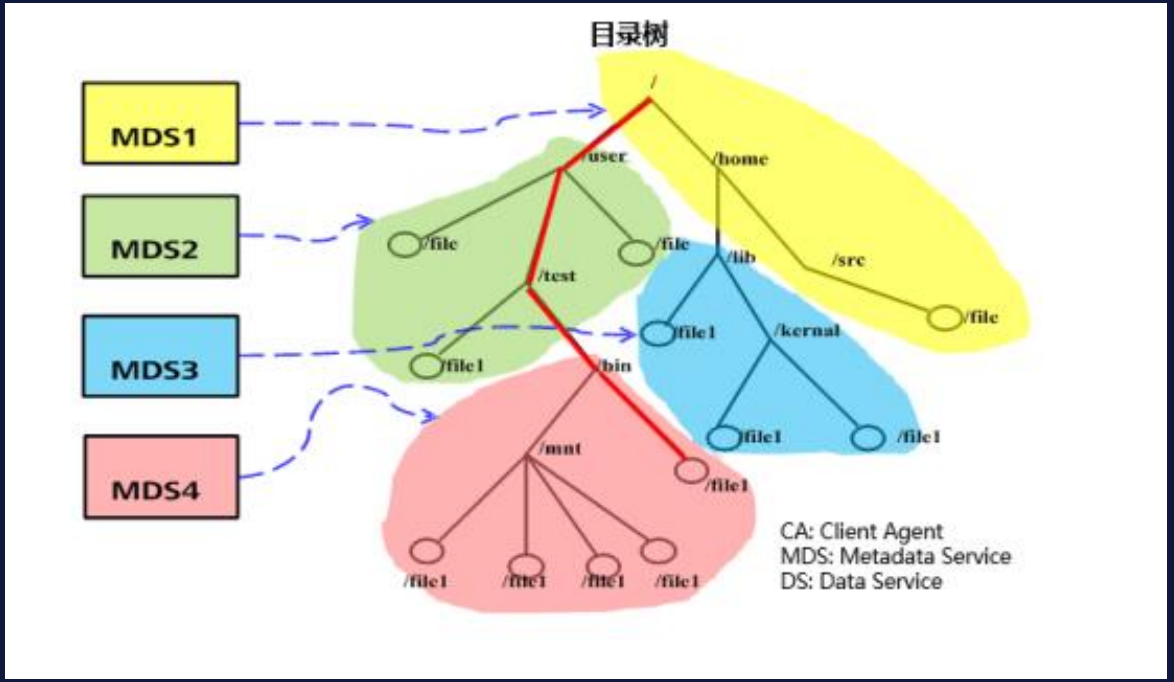
多MDS

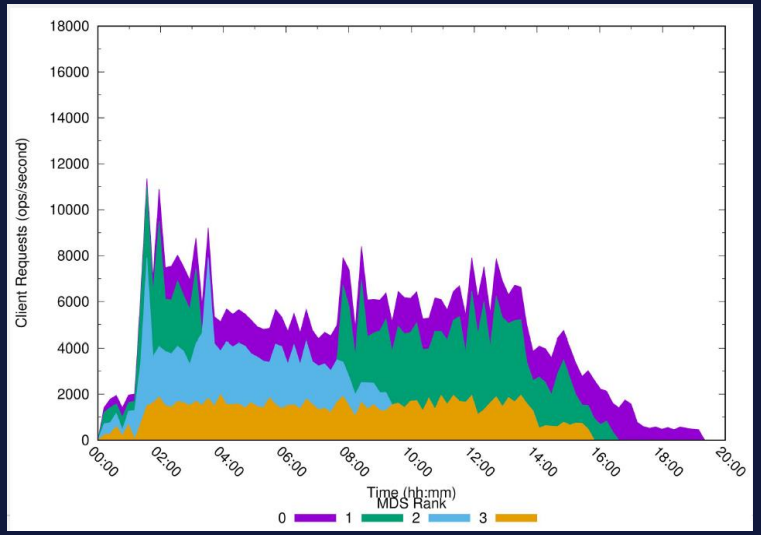


- efficient
  - hierarchical partition preserve locality
  - single mds for any piece of metadata
- adaptive
  - move work from busy to idle servers
  - hot metadata gets replicated
- scalable
  - arbitrarily partition metadata
  - coarse when possible, fine when necessary
- dynamic
  - daemons can join/leave
  - take over for failed nodes

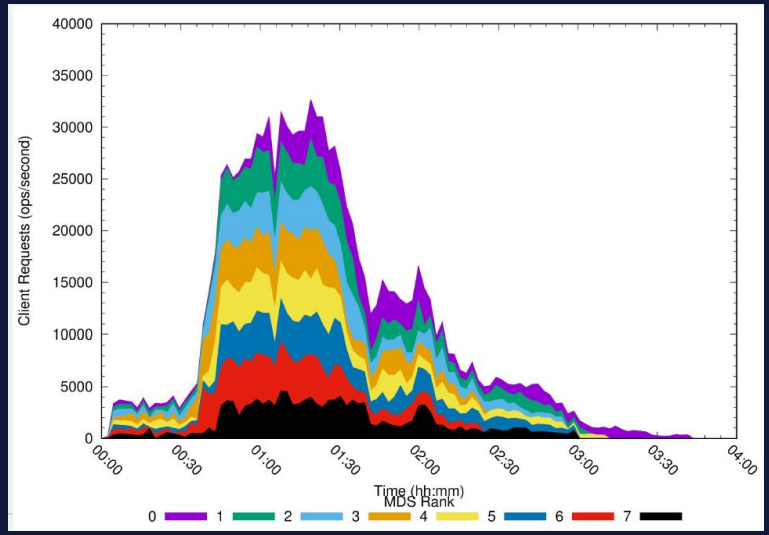
多MDS

UCLOUD


目录树
MDS1
MDS2
MDS3
MDS4
/
/user
/home
/file
/file
/test
/lib
/src
/file
file1
/kernal
/bin
file1
/file1
/mnt
/file1
/file1
/file1
/file1
/file1
CA: Client Agent
MDS: Metadata Service
DS: Data Service



4 MDS 4 Client



8 MDS 8 Client

参考

UCLOUD

- http://events.linuxfoundation.org/events/vault/program/slides
- https://www.dellemc.com/resources/en-us/asset/white-papers/products/storage/h13249-isilon-onefs-smartflash-wp.pdf
- https://www.snia.org/events/storage-developer/presentations15 Cache Service In Distributed FileSystem
- Vault 2017: Large-scale Stability and Performance of the Ceph File System

UCLOUD

THANKS

2018.11.10 武汉