

储宝文件系统（CubaoFS）使用手册



ChubaoFS

ChubaoFS(储宝文件系统)是为大规模容器平台设计的分布式文件系统。



下载手机APP
畅享精彩阅读

目 录

致谢

快速开始

概述

快速开始

设计文档

资源管理子系统

元数据子系统

数据子系统

客户端

用户手册

资源管理节点

元数据节点

数据节点

客户端

性能监控

优化配置FUSE参数

Docker

在Kubernetes中使用Chubaofs

管理手册

资源管理节点

集群管理命令

元数据节点管理命令

数据节点管理命令

卷管理命令

元数据分片管理命令

数据分片管理命令

资源管理命令

元数据节点

元数据分片管理命令

Inode管理命令

Dentry调试命令

运维手册

使用案例

性能评估

常见问题

致谢

当前文档 《储宝文件系统 (CubaoFS) 使用手册》 由 进击的皇虫 使用 书栈网 (BookStack.CN) 进行构建, 生成于 2020-02-01。

书栈网仅提供文档编写、整理、归类等功能, 以及对文档内容的生成和导出工具。

文档内容由网友们编写和整理, 书栈网难以确认文档内容知识点是否错漏。如果您在阅读文档获取知识的时候, 发现文档内容有不恰当的地方, 请向我们反馈, 让我们共同携手, 将知识准确、高效且有效地传递给每一个人。

同时, 如果您在日常工作、生活和学习中遇到有价值有营养的知识文档, 欢迎分享到书栈网, 为知识的传承献上您的一份力量!

如果当前文档生成时间太久, 请到书栈网获取最新的文档, 以跟上知识更新换代的步伐。

内容来源: [CubaoFS](https://github.com/chubaofs/docs-zh) <https://github.com/chubaofs/docs-zh>

文档地址: <http://www.bookstack.cn/books/chubaofs>

书栈官网: <https://www.bookstack.cn>

书栈开源: <https://github.com/TruthHun>

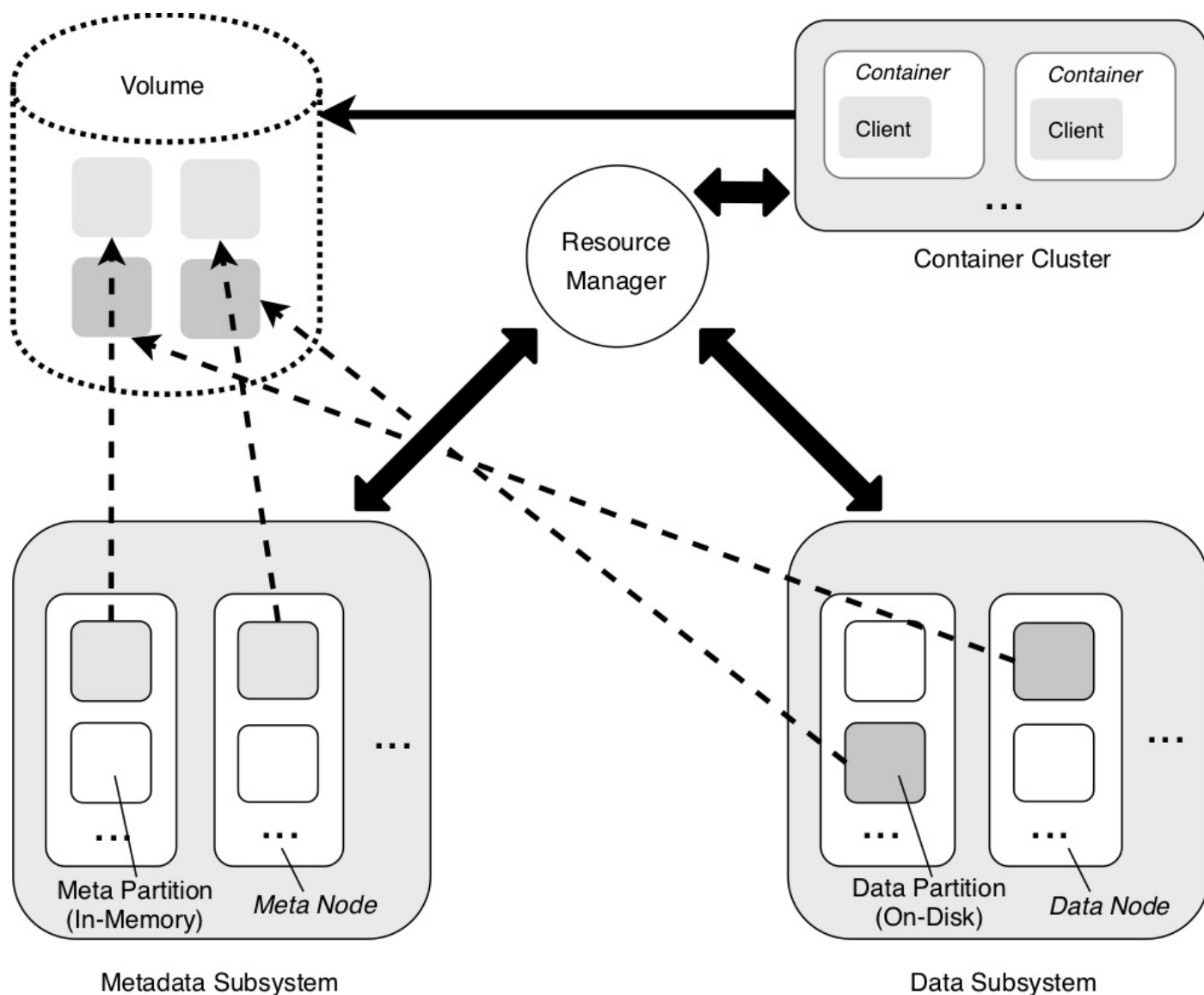
分享, 让知识传承更久远! 感谢知识的创造者, 感谢知识的分享者, 也感谢每一位阅读到此处的读者, 因为我们都将成为知识的传承者。

- [概述](#)
- [快速开始](#)

概述

ChubaoFS(储宝文件系统)是为大规模容器平台设计的分布式文件系统。

整体架构



ChubaoFS由 元数据子系统 ， 数据子系统 和 资源管理节点 组成，可以通过 客户端 访问不同文件系统实例， 卷 。

元数据子系统由元数据节点组成，每个节点可以管理一组 元数据分片 。

数据子系统有数据节点组成，每个节点管理一组 数据分片 。

在ChubaoFS中，卷是一个逻辑概念，由多个元数据和数据分片组成。从客户端的角度看，卷可以被看作是能被容器访问的文件系统实例。一个卷可以在多个容器中挂载，使得文件可以被不同客户端同时访问。一个ChubaoFS集群可以有上百个卷，大小从几GB至几TB不等。

概括来说，资源管理节点定期获取元数据和数据子系统信息，客户端则定期从资源管理器拉取元数据和数据分片信息，并且进行缓存。通常来讲，文件操作由客户端发起，直接与数据和元数据节点通信，无需资源管理节点介入。

系统特性

可扩展元数据管理

元数据操作有时候会成为文件系统的性能瓶颈。在我们的平台中，由于可能会有成百上千的客户端同时访问文件，这个问题变得非常突出。单独节点存储元数据很容易成为性能瓶颈。所以，ChubaoFS中使用了分布式元数据子系统，以便提供高可扩展性。元数据子系统可以认为是内存元数据存储。我们使用了两个B-Tree，inodeTree和dentryTree，来加快索引速度。每个元数据分片根据inode id范围进行划分。

多租户

为了降低存储成本，很多应用和服务都使用了共享的存储基础设施。不同负载交织在一起，文件大小可以从几KB至几百GB，读写模型从顺序到随机。一个成熟的文件系统应该可以高性能地服务于这些负载。ChubaoFS的存储引擎同时提供了对大文件和小文件的随机/顺序读写的支持。

强一致性的复制协议

出于性能考虑，ChubaoFS根据文件写入的方式的不同采用不同的复制协议来保障副本之间的一致性。

POSIX兼容

兼容POSIX接口，可以使得上层应用的开发变得简单，并且大大降低新用户的学习难度。同时，ChubaoFS在实现时放松了对POSIX语义的一致性要求来兼顾文件和元文件操作的性能。

快速开始

编译构建

使用如下命令同时构建server，client及相关的依赖：

```
1. $ git clone http://github.com/chubaofs/chubaofs.git
2. $ cd chubaofs
3. $ make build
```

如果构建成功，将在 `build/bin` 目录中生成可执行文件 `cfs-server` 和 `cfs-client`。

集群部署

启动资源管理节点

```
1. nohup ./cfs-server -c master.json &
```

示例 `master.json` ：注意：master服务最少应该启动3个节点实例

```
1. {
2.   "role": "master",
3.   "ip": "192.168.31.173",
4.   "listen": "80",
5.   "prof": "10088",
6.   "id": "1",
7.   "peers": "1:192.168.31.173:80,2:192.168.31.141:80,3:192.168.30.200:80",
8.   "retainLogs": "20000",
9.   "logDir": "/export/Logs/master",
10.  "logLevel": "info",
11.  "walDir": "/export/Data/master/raft",
12.  "storeDir": "/export/Data/master/rocksdbstore",
13.  "consulAddr": "http://consul.prometheus-cfs.local",
14.  "exporterPort": 9510,
15.  "clusterName": "cfs",
16.  "metaNodeReservedMem": "134217728"
17. }
```

详细配置参数请参考 [资源管理节点](#)。

启动元数据节点

```
1. nohup ./cfs-server -c metanode.json &
```

示例 `meta.json` : 注意: metanode服务最少应该启动3个节点实例

```
1. {
2.     "role": "metanode",
3.     "listen": "9021",
4.     "prof": "9092",
5.     "logLevel": "info",
6.     "metadataDir": "/export/Data/metanode",
7.     "logDir": "/export/Logs/metanode",
8.     "raftDir": "/export/Data/metanode/raft",
9.     "raftHeartbeatPort": "9093",
10.    "raftReplicaPort": "9094",
11.    "totalMem": "17179869184",
12.    "consulAddr": "http://consul.prometheus-cfs.local",
13.    "exporterPort": 9511,
14.    "masterAddr": [
15.        "192.168.31.173:80",
16.        "192.168.31.141:80",
17.        "192.168.30.200:80"
18.    ]
19. }
```

详细配置参数请参考 [元数据节点](#)。

启动数据节点

- 准备数据目录

推荐 使用单独磁盘作为数据目录，配置多块磁盘能够达到更高的性能。

磁盘准备

1.1 查看机器磁盘信息，选择给ChubaoFS使用的磁盘

```
1. fdisk -l
```

1.2 格式化磁盘，建议格式化为XFS

```
1. mkfs.xfs -f /dev/sdx
```

1.3 创建挂载目录


```
1. mkdir /data0
```

1.4 挂载磁盘

```
1. mount /dev/sdx /data0
```

- 启动数据节点

```
1. nohup ./cfs-server -c datanode.json &
```

示例 `datanode.json` :注意: datanode服务最少应该启动4个节点实例

```
1. {
2.   "role": "datanode",
3.   "listen": "6000",
4.   "prof": "6001",
5.   "logDir": "/export/Logs/datanode",
6.   "logLevel": "info",
7.   "raftHeartbeat": "9095",
8.   "raftReplica": "9096",
9.   "raftDir": "export/Logs/datanode/raft",
10.  "consulAddr": "http://consul.prometheus-cfs.local",
11.  "exporterPort": 9512,
12.  "masterAddr": [
13.    "192.168.31.173:80",
14.    "192.168.31.141:80",
15.    "192.168.30.200:80"
16.  ],
17.  "rack": "",
18.  "disks": [
19.    "/data0:21474836480",
20.    "/data1:21474836480"
21.  ]
22. }
```

详细配置参数请参考 [数据节点](#)。

创建Volume卷

```
1. curl -v "http://192.168.31.173/admin/createVol?
   name=test&capacity=100000&owner=cfs"
2.
3. 如果执行性能测试, 请调用相应的API, 创建足够多的数据分片 (data partition), 如果集群中有8块磁
```

盘，那么需要创建80个datapartition

挂载客户端

- 运行 `modprobe fuse` 插入FUSE内核模块。
- 运行 `yum install -y fuse` 安装libfuse。
- 运行 `nohup client -c fuse.json &` 启动客户端。

样例 `fuse.json` ,

```
1. {
2.   "mountPoint": "/mnt/fuse",
3.   "volName": "test",
4.   "owner": "cfs",
5.   "masterAddr": "192.168.31.173:80,192.168.31.141:80,192.168.30.200:80",
6.   "logDir": "/export/Logs/client",
7.   "profPort": "10094",
8.   "logLevel": "info"
9. }
```

详细配置参数请参考 [客户端](#)。

升级注意事项

集群数据节点和元数据节点升级前，请先禁止集群自动为卷扩容数据分片。

- 冻结集群

```
1. curl -v "http://192.168.31.173/cluster/freeze?enable=true"
```

- 升级节点
- 开启自动扩容数据分片

```
1. curl -v "http://192.168.31.173/cluster/freeze?enable=false"
```

- 资源管理子系统
- 元数据子系统
- 数据子系统
- 客户端

资源管理子系统

Master负责异步的处理不同类型的任务，比如 创建/删除/更新/比对副本是否一致等数据分片和元数据分片的操作，管理数据节点和元数据节点的存活状态，创建和维护卷信息。Master有多个节点，它们之间同步raft算法保证元数据一致性，并且把元数据持久化到RocksDB。

基于利用率的分布策略

基于利用率的分布策略放置文件元数据和内容是Master最主要的特征，次分布策略能够更高效的利用集群资源。数据分片和元数据分片的分布策略工作流程：1. 创建卷的时候，master根据剩余磁盘/内存空间加权计算，选择权重最高的数据节点创建数据/元数据分片，写文件时，客户端随机选择数据分片和元数据分片。2. 如果卷的大部分数据分片是只读，只有很少的数据分片是可读写的时候，master会自动创建新的数据分片来分散写请求。

基于利用率的分布策略能带来两点额外的好处：1. 当新节点加入时，不需要重新做数据均衡，避免了因为数据迁移带来的开销。2. 因为使用统一的分布策略，显著降低了产生热点数据的可能性。

副本放置

Master确保一个分片的多个副本都在不同的机器上。

拆分元数据分片

满足下面任意一个条件，元数据分片将会被拆分1. 元数据节点内存使用率达到设置的阈值，比如总内存是64GB，阈值是0.75，如果元数据节点使用的内存达到48GB，该节点上的所有元数据分片都将会被拆分.2. 元数据分片占用的内存达到16GB

只有数据分片ID是卷所有数据分片中ID最大的，才会真正被拆分。假设数据分片A符合拆分条件，其inode范围是 $[0, \text{正无穷})$ ，则拆分后A的范围为 $[0, A.\text{MaxInodeID} + \text{step})$ ，新生成的B分片的范围是 $[A.\text{MaxInodeID} + \text{step} + 1, \text{正无穷})$ ，其中step是步长，默认是2的24方。 MaxInodeID 是由元数据节点汇报。

异常处理

如果数据/元数据分片某个副本不可用（硬盘失败、硬件错误等），该副本上的数据最终会被迁移到新的副本上。

元数据子系统

元数据子系统是一个以内存为中心的分布式数据结构，是由一个或多个元数据分片组成。元数据子系统数据使用multiraft来充分使用服务器资源和保障数据高可用及强一致性，可以很方便的迁移资源，并通过分裂实现横向扩展。

元数据内部设计

每个元数据可以包含成百上千的元数据分片，每个分片由InodeTree (BTree) 和DentryTree (BTree) 组成。每个Inode代表文件系统中的文件或目录，每个dentry代表一个目录项，dentry由parentId和name组成。在DentryTree中，以ParentId和name组成索引，进行存储和检索；在InodeTree中，则以inode id进行索引。使用multiRaft协议保障高可用性和数据一致性复制，且每个节点集合会包含大量的分片组，每个分片组对应一个raft group；每个分片组隶属于某个volume；每个分片组都是某个volume的一段元数据范围（inode id范[100-20000]）；元数据子系统通过分裂来完成动态扩容；当性一分片组的性能（包含如下指标：内存）紧接临近值时，资源管理器服务会预估一个结束点，并通知此组节点设备，只服务到此点之前的数据，同时也会新选出一组节点，并动态加入到当前业务系统中，新节点组其实点刚好是上个节点组的结束点位置。

复制

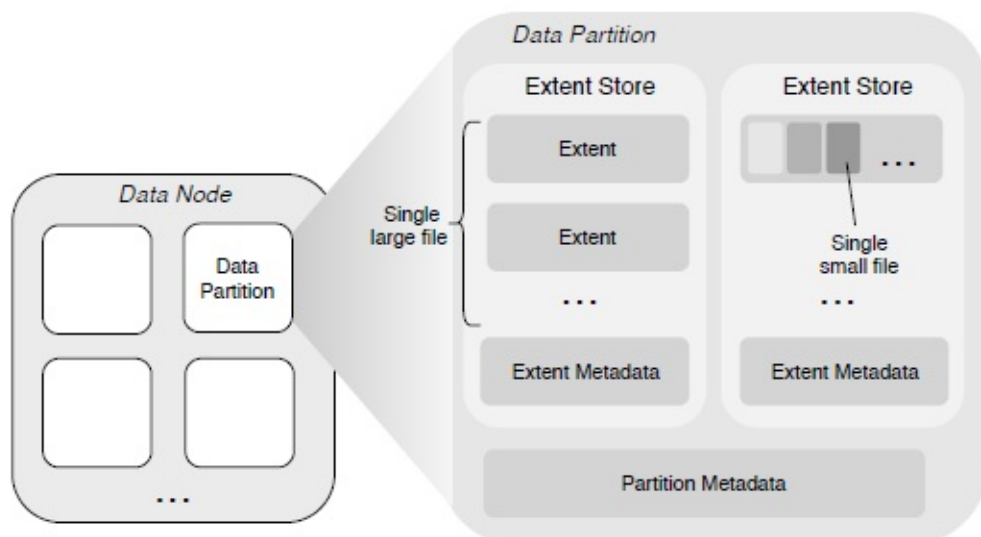
元数据更新的复制是以元数据分片为单位的。复制强一致性是通过Raft改良版本，MultiRaft来实现的。MultiRaft减少了心跳通信的负担。

故障恢复

内存元数据分片通过快照的方式持久化到磁盘以作备份和恢复使用。日志压缩技术被用来减小日志文件大小和恢复时间。值得一提的是，元数据操作有可能导致孤儿inode，即只有inode但是没有对应的dentry。为了减少这种情况的发生，首先，元数据节点通过Raft保证高可用，单点故障后可以迅速恢复；其次，客户端保证在一定时间内进行重试。

数据子系统

数据子系统的设计是为了满足大、小文件支持顺序随机访问的多租户需求。采用两种不同的复制协议，以确保副本之间的强一致性，并在性能和代码可用性上进行一些权衡。



系统特性

- 大文件存储

对于大文件，内容存储为一个或多个扩展数据块的序列，这些扩展数据块可以分布在不同数据节点上的不同数据分区中。将新文件写入扩展数据块存储区始终会导致数据以新扩展数据块的零偏移量写入，这样就不需要在扩展数据块内进行偏移。文件的最后一个范围不需要通过填充来补齐其大小限制（即该范围没有空洞），并且不会存储来自其他文件的数据。

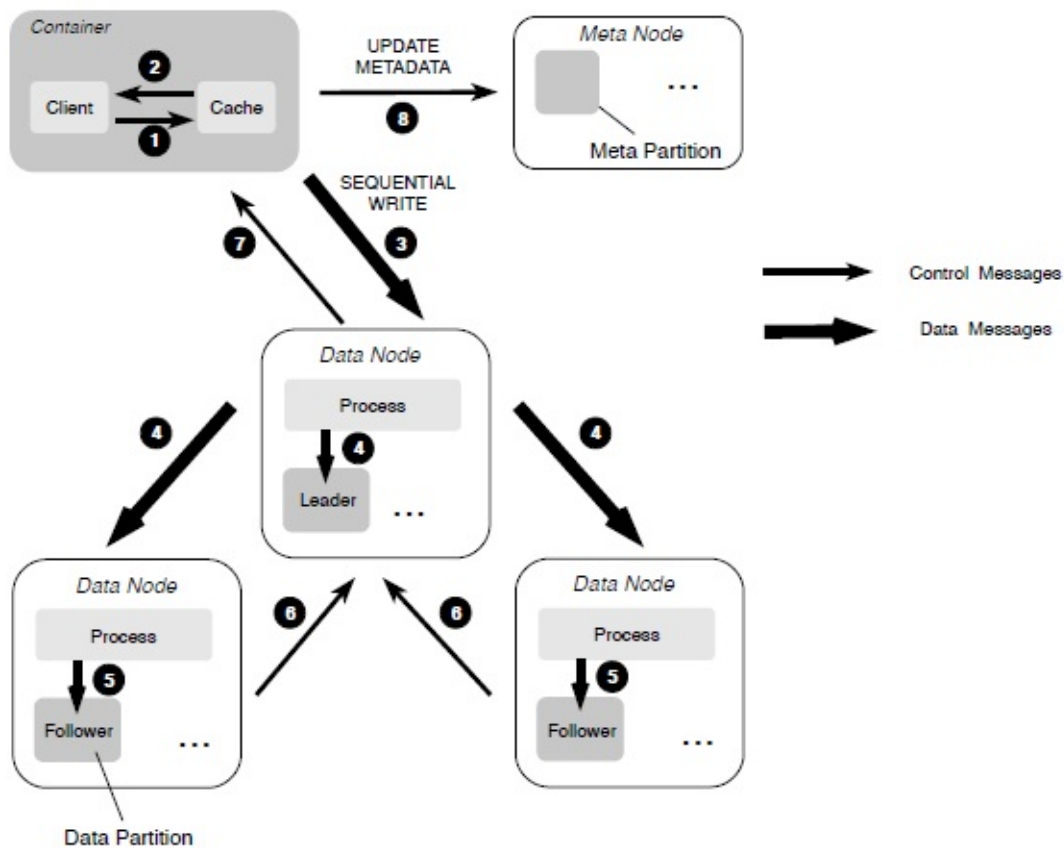
- 小文件存储

将多个小文件的内容聚合存储在一个文件内，并将每个文件内容的物理偏移量记录在相应的元数据中。删除文件内容（释放此文件占用的磁盘空间）是通过底层文件系统提供的文件穿洞接口（`fallocate()`）实现的。这种设计的优点是不需要实现垃圾回收机制，因此在一定程度上避免使用从逻辑偏移到物理偏移的映射。

- 复制

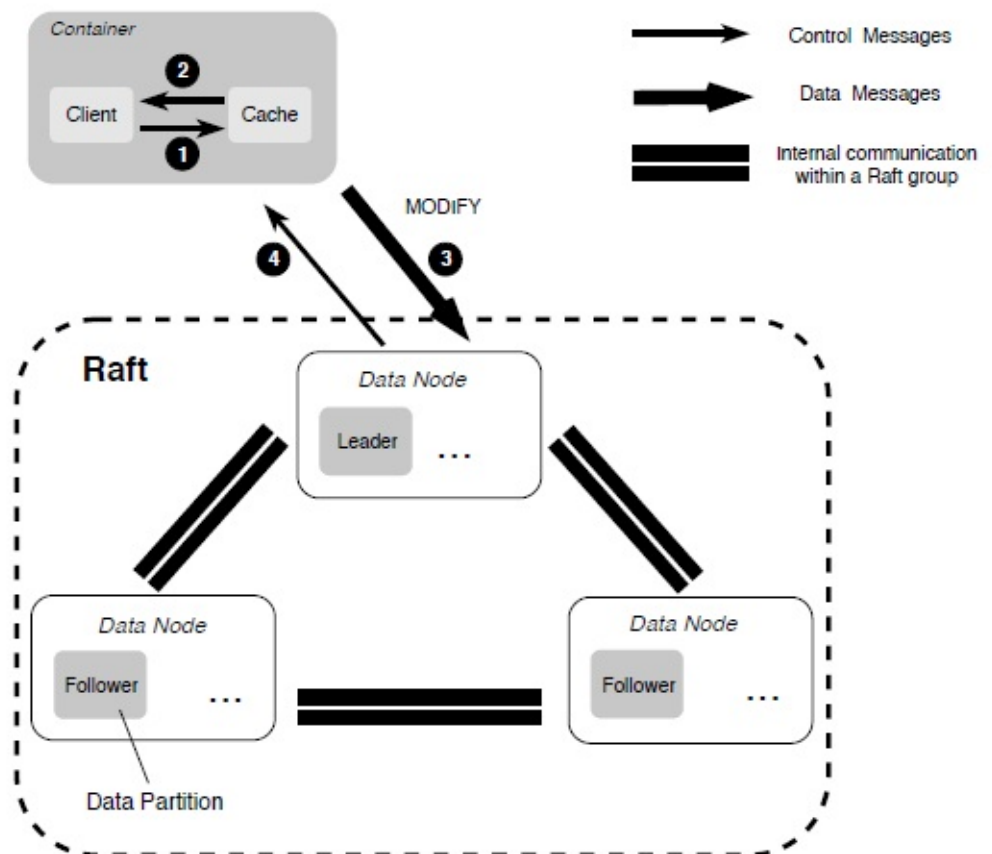
成员间的文件复制，根据文件写入模式，ChubaoFS采用不同的复制策略。

当文件按顺序写入ChubaoFS时，使用主备份复制协议来确保与优化的IO吞吐量的强一致性。



在随机写入时覆盖

现有的文件内容时，我们采用了一种基于Multi-Raft的复制协议，该协议类似于元数据子系统使用



的协议，以确保强一致性。

- 故障恢复

由于存在两种不同的复制协议，当发现复制副本上的故障时，我们首先通过检查每个数据块的长度并使所有数据块对齐，启动基于主备份的复制协议的恢复。一旦这个处理完成，我们就开始在我们的基于Multi-Raft的恢复。

HTTP接口

API	方法	参数	描述
/disks	GET	N/A	获取磁盘的列表和信息。
/partitions	GET	N/A	获取所有数据组的信息。
/partition	GET	partitionId[int]	获取特定数据组的详细信息。
/extent	GET	partitionId[int]&extentId[int]	获取特定数据组里面特定extent文件的信息。
/stats	GET	N/A	获取DATA节点的信息。

客户端

客户端以用户态可执行程序的形式可以运行在容器中，并且通过FUSE将挂载的卷及文件系统接口提供给其它用户态应用。

客户端缓存

客户端进程在以下几种情况下会使用客户端缓存。

客户端为了减少与资源管理节点的通信负担，会在挂载启动时获取该挂载卷中所有元数据和数据节点的地址，并且进行缓存，后续会定期从资源管理节点进行更新。

客户端为了减少与元数据节点的通信，会缓存inode，dentry以及extent元数据信息。通常意义上，读请求应该能够读到之前所有的写入，但是客户端元数据缓存可能会导致多客户端写同一个文件时的一致性问题的。所以，ChubaoFS的设计中，不同客户端，或者说挂载点，可以同时读一个文件，但是不能够同时写一个文件（注意：不同进程可以从同一个挂载点并发写一个文件）。打开文件时，客户端会强制从元数据节点更新文件元数据信息。

由于故障恢复时，raft复制组的主节点有可能发生变化，导致客户端缓存的主节点地址无效。因此，客户端在发送请求收到not leader回复时，会轮询重试该复制组的所有节点。重试成功后识别出新的主节点，客户端会缓存新的主节点地址。

对接FUSE接口

ChubaoFS客户端通过对接FUSE为提供用户态文件系统接口。之前，性能较低被认为是用户态文件系统最大的缺点。但是经过多年的发展，FUSE已经在性能上有了很大提高。后续，ChubaoFS会着手开发内核态文件系统客户端。

目前来看，FUSE的writeback cache特性并未达到预期的性能提升。FUSE默认的写流程走的是directIO接口，使得每次写入长度较小时会有性能问题，因为每次的写请求都会被推送至后端存储。FUSE的解决方案是writeback cache，即小写入写到缓存页即返回，由内核根据回刷策略推送至后端存储。这样，顺序的小请求会被聚合。但是，在实际生产中，我们发现writeback cache特性作用非常有限，原因是走writecache的写操作触发了内核balance dirty page的流程，使得本应该是响应时间非常短的写操作仍然会等较长时间才返回。这个问题在小的写入时尤其明显。

- [资源管理节点](#)
- [元数据节点](#)
- [数据节点](#)
- [客户端](#)
- [性能监控](#)
- [优化配置FUSE参数](#)
- [Docker](#)
- [在Kubernetes中使用Chubaofs](#)

资源管理节点

Master负责管理ChubaoFS整个集群，主要存储5种元数据，包括：数据节点、元数据节点、卷、数据分片、元数据分片。所有的元数据都保存在master的内存中，并且持久化到RocksDB。多个Master之间通过raft协议保证集群元数据的一致性。注意：master的实例最少需要3个

系统特性

- 多租户，资源隔离
- 多个卷共享数据节点和元数据节点, 每个卷独享各自的数据分片和元数据分片
- 与数据节点和元数据节点即有同步交互，也有异步交互，交互方式与任务类型相关。

配置参数

ChubaoFS 使用 **JSON** 作为配置文件的格式。

属性			
配置项	类型	描述	是否必须
role	字符串	进程的角色，值只能是master	是
ip	字符串	主机ip	是
listen	字符串	http服务监听的端口号	是
prof	字符串	golang pprof 端口号	是
id	字符串	区分不同的master节点	是
peers	字符串	raft复制组成员信息	是
logDir	字符串	日志文件存储目录	是
logLevel	字符串	日志级别，默认是error.	否
retainLogs	字符串	保留多少条raft日志.	是
walDir	字符串	raft wal日志存储目录.	是
storeDir	字符串	RocksDB数据存储目录. 此目录必须存在，如果目录不存在，无法启动服务	是
clusterName	字符串	集群名字	是
exporterPort	整型	The prometheus exporter port	否
consulAddr	字符串	consul注册地址，供prometheus exporter使用	否

metaNodeReservedMem	字符串	如果metanode的内存小于这个值，metanode被置为只读状态	
---------------------	-----	------------------------------------	--

Example:

```

1.  {
2.    "role": "master",
3.    "id": "1",
4.    "ip": "192.168.31.173",
5.    "listen": "80",
6.    "prof": "10088",
7.    "peers": "1:192.168.31.173:80,2:192.168.31.141:80,3:192.168.30.200:80",
8.    "retainLogs": "20000",
9.    "logDir": "/export/Logs/master",
10.   "logLevel": "info",
11.   "walDir": "/export/Data/master/raft",
12.   "storeDir": "/export/Data/master/rocksdbstore",
13.   "exporterPort": 9510,
14.   "consulAddr": "http://consul.prometheus-cfs.local",
15.   "clusterName": "test",
16.   "metaNodeReservedMem": "134217728"
17. }

```

启动服务

```
1. nohup ./cfs-server -c master.json > nohup.out &
```

元数据节点

元数据节点是由多个元数据分片(meta partition)和基于multiRaft的对应个数的raft实例组成；每个元数据分片(meta partition)都是一个inode范围，且包含两个内存BTrees： inode BTree和dentry BTree。注意：metanode的实例最少需要3个

Properties				
配置项	类型	描述	是否必须	
role	字符串	进程角色： <i>metanode</i>	是	
listen	字符串	监听和接受请求的端口	是	
prof	字符串	调试和管理员API接口	是	
logLevel	字符串	日志级别，默认： <i>error</i>	否	
metadataDir	字符串	元数据快照存储目录	是	
logDir	字符串	日志存储目录	是	
raftDir	字符串	raft wal日志目录	是	
raftHeartbeatPort	字符串	raft心跳通信端口	是	
raftReplicaPort	字符串	raft数据传输端口	是	
consulAddr	字符串	prometheus注册接口	否	
exporterPort	字符串	prometheus获取监控数据端口	否	
masterAddr	字符串	master服务地址	是	
totalMem	字符串	最大可用内存	是	
localIP	字符串	本机ip地址	否, 如果不填写该选项，则使用和master通信的ip地址	

样例：

```
1. {
2.     "role": "metanode",
3.     "listen": "9021",
4.     "prof": "9092",
5.     "logLevel": "debug",
6.     "localIP": "192.168.31.173",
7.     "metadataDir": "/export/Data/metanode",
```

```
8.     "logDir": "/export/Logs/metanode",
9.     "raftDir": "/export/Data/metanode/raft",
10.    "raftHeartbeatPort": "9093",
11.    "raftReplicaPort": "9094",
12.    "consulAddr": "http://consul.prometheus-cfs.local",
13.    "exporterPort": 9511,
14.    "totalMem": "17179869184",
15.    "masterAddr": [
16.        "192.168.31.173:80",
17.        "192.168.31.141:80",
18.        "192.168.30.200:80"
19.    ]
20. }
```

启动服务

```
1. nohup ./cfs-server -c metanode.json > nohup.out &
```

数据节点

启动数据节点

通过执行ChubaoFS的二进制文件并用“-c”参数指定的配置文件来启动一个DATANODE进程。注意datanode的实例最少需要 4 个

```
1. nohup cfs-server -c datanode.json &
```

配置参数

Properties			
关键字	参数类型	描述	是否必要
role	string	Role必须配置为“datanode”	是
listen	string	数据节点作为服务端启动TCP监听的端口	是
localIP	string	数据节点作为服务端选用的IP	否
prof	string	数据节点提供HTTP接口所用的端口	是
logDir	string	调测日志存放的路径	是
logLevel	string	调测日志的级别。默认是error	否
raftHeartbeat	string	RAFT发送节点间心跳消息所用的端口	是
raftReplica	string	RAFT发送日志消息所用的端口	是
raftDir	string	RAFT调测日志存放的路径。默认在二进制文件启动路径	否
consulAddr	string	监控系统的地址	否
exporterPort	string	监控系统的端口	否
masterAddr	string slice	集群管理器的地址	是
localIP	字符串	本机ip地址	否, 如果不填写该选项, 则使用和master通信的ip地址
rack	string	机架号	否
disks	string slice	格式: 磁盘挂载路径: 预留空间 预留空间配置范围 [20G, 50G]	是

举例：

```
1. {
2.     "role": "datanode",
3.     "listen": "6000",
4.     "prof": "6001",
5.     "localIP": "192.168.31.174",
6.     "logDir": "/export/Logs/datanode",
7.     "logLevel": "info",
8.     "raftHeartbeat": "9095",
9.     "raftReplica": "9096",
10.    "raftDir": "/export/Logs/datanode/raft",
11.    "consulAddr": "http://consul.prometheus-cfs.local",
12.    "exporterPort": 9512,
13.    "masterAddr": [
14.        "10.196.30.200:80",
15.        "10.196.31.141:80",
16.        "10.196.31.173:80"
17.    ],
18.    "rack": "",
19.    "disks": [
20.        "/data0:21474836480",
21.        "/data1:21474836480"
22.    ]
23. }
```


客户端

环境依赖

- 插入内核FUSE模
- 安装libfuse

```
1. modprobe fuse
2. yum install -y fuse
```

配置文件

fuse.json

```
1. {
2.   "mountPoint": "/mnt/fuse",
3.   "volName": "test",
4.   "owner": "cfs",
5.   "masterAddr": "192.168.31.173:80,192.168.31.141:80,192.168.30.200:80",
6.   "logDir": "/export/Logs/client",
7.   "logLevel": "info",
8.   "profPort": "10094"
9. }
```

配置选项

名称	类型	描述	必需
mountPoint	string	挂载点	是
volName	string	集群名称	是
owner	string	所有者	是
masterAddr	string	Master节点地址	是
logDir	string	日志存放路径	否
logLevel	string	日志级别: debug, info, warn, error	否
profPort	string	golang pprof调试端口	否
exporterPort	string	监控模块端口	否
consulAddr	string	监控注册服务器地址	否
lookupValid	string	内核FUSE lookup有效期, 单位: 秒	否
attrValid	string	内核FUSE attribute	否

attrValid	string	有效期，单位：秒	否
icacheTimeout	string	客户端inode cache有效期，单位：秒	否
enSyncWrite	string	使能DirectIO同步写，即DirectIO强制数据节点落盘	否
autoInvalData	string	FUSE挂载使用AutoInvalData选项	否

挂载

执行如下命令挂载客户端：

```
1. nohup ./cfs-client -c fuse.json &
```

如果使用示例的 `fuse.json`，则客户端被挂载到 `/mnt/fuse`。所有针对 `/mnt/fuse` 的操作都将被作用于ChubaoFS。

性能监控

ChubaoFS 集成了prometheus作为性能监控指标采集模块。在各模块配置文件中增加如下配置参数来启用该模块：

```
1. {
2.     "exporterPort": 9510,
3.     "consulAddr": "http://consul.prometheus-cfs.local"
4. }
```

- exporterPort: prometheus指标暴露端口号。设置后，可通过URL([http://\\$hostip:\\$exporterPort/metrics](http://$hostip:$exporterPort/metrics)) 暴露prometheus监控指标。若不设置，prometheus指标监控模块将不会工作。
- consulAddr: consul注册服务器地址。设置后，可配合prometheus的自动发现机制实现chubaofs节点exporter的自动发现服务。若不设置，将不会启用consul自动注册服务。

可使用grafana作为prometheus 监控指标的展示前端，如下图所示：



可以通过prometheus alertmanager组件进行配置，来实现chubaofs系统的监控指标报警通知服务，详细可参考 [alertmanager文档](#)。

相关链接：

- [prometheus 安装配置](#)
- [consul 安装配置](#)
- [grafana 安装配置](#) Grafana 监控面板配置模板如下：

```
1.  {
2.    "__inputs": [
3.      {
4.        "name": "DS_ChubaoFS01",
5.        "label": "cfs01",
6.        "description": "",
7.        "type": "datasource",
8.        "pluginId": "prometheus",
9.        "pluginName": "Prometheus"
10.     }
11.  ],
12.  "__requires": [
13.    {
14.      "type": "grafana",
15.      "id": "grafana",
16.      "name": "Grafana",
17.      "version": "5.2.4"
18.    },
19.    {
20.      "type": "panel",
21.      "id": "graph",
22.      "name": "Graph",
23.      "version": "5.0.0"
24.    },
25.    {
26.      "type": "datasource",
27.      "id": "prometheus",
28.      "name": "Prometheus",
29.      "version": "5.0.0"
30.    },
31.    {
32.      "type": "panel",
33.      "id": "singlestat",
34.      "name": "Singlestat",
35.      "version": "5.0.0"
36.    }
37.  ],
38.  "annotations": {
39.    "list": [
40.      {
41.        "builtIn": 1,
42.        "datasource": "-- Grafana --",
43.        "enable": true,
44.        "hide": true,
45.        "iconColor": "rgba(0, 211, 255, 1)",
46.        "name": "Annotations & Alerts",
47.        "type": "dashboard"
48.      }
49.    ]
50.  }
```

```
49.     ]
50.   },
51.   "editable": true,
52.   "gnetId": null,
53.   "graphTooltip": 0,
54.   "id": null,
55.   "iteration": 1546930136099,
56.   "links": [
57.     {
58.       "icon": "external link",
59.       "tags": [],
60.       "targetBlank": true,
61.       "title": "mdc",
62.       "tooltip": "",
63.       "type": "link",
64.       "url": "http://mdc.jd.com/monitor/chart?ip=$hostip"
65.     }
66.   ],
67.   "panels": [
68.     {
69.       "gridPos": {
70.         "h": 1,
71.         "w": 24,
72.         "x": 0,
73.         "y": 0
74.       },
75.       "id": 85,
76.       "title": "Summary",
77.       "type": "row"
78.     },
79.     {
80.       "cacheTimeout": null,
81.       "colorBackground": false,
82.       "colorValue": false,
83.       "colors": [
84.         "#299c46",
85.         "rgba(237, 129, 40, 0.89)",
86.         "#d44a3a"
87.       ],
88.       "datasource": "${DS_ChubaoFS01}",
89.       "format": "none",
90.       "gauge": {
91.         "maxValue": 100,
92.         "minValue": 0,
93.         "show": false,
94.         "thresholdLabels": false,
95.         "thresholdMarkers": true
96.       },
```

```
97.     "gridPos": {
98.         "h": 4,
99.         "w": 4,
100.        "x": 0,
101.        "y": 1
102.    },
103.    "id": 38,
104.    "interval": null,
105.    "links": [
106.        {
107.            "dashUri": "db/cfs-master",
108.            "dashboard": "cfs-master",
109.            "includeVars": false,
110.            "keepTime": true,
111.            "targetBlank": true,
112.            "title": "cfs-master",
113.            "type": "dashboard"
114.        }
115.    ],
116.    "mappingType": 1,
117.    "mappingTypes": [
118.        {
119.            "name": "value to text",
120.            "value": 1
121.        },
122.        {
123.            "name": "range to text",
124.            "value": 2
125.        }
126.    ],
127.    "maxDataPoints": 100,
128.    "nullPointMode": "connected",
129.    "nullText": null,
130.    "postfix": "",
131.    "postfixFontSize": "50%",
132.    "prefix": "",
133.    "prefixFontSize": "50%",
134.    "rangeMaps": [
135.        {
136.            "from": "null",
137.            "text": "N/A",
138.            "to": "null"
139.        }
140.    ],
141.    "sparkline": {
142.        "fillColor": "rgba(31, 118, 189, 0.18)",
143.        "full": false,
144.        "lineColor": "rgb(31, 120, 193)",
```

```
145.         "show": true
146.     },
147.     "tableColumn": "",
148.     "targets": [
149.         {
150.             "expr": "count(go_info{cluster=~\"$cluster\", app=~\"$app\",
role=~\"master\"})",
151.             "format": "time_series",
152.             "intervalFactor": 1,
153.             "refId": "A"
154.         }
155.     ],
156.     "thresholds": "",
157.     "title": "master_count",
158.     "type": "singlestat",
159.     "valueFontSize": "80%",
160.     "valueMaps": [
161.         {
162.             "op": "=",
163.             "text": "N/A",
164.             "value": "null"
165.         }
166.     ],
167.     "valueName": "current"
168. },
169. {
170.     "cacheTimeout": null,
171.     "colorBackground": false,
172.     "colorValue": false,
173.     "colors": [
174.         "#299c46",
175.         "rgba(237, 129, 40, 0.89)",
176.         "#d44a3a"
177.     ],
178.     "datasource": "${DS_ChubaoFS01}",
179.     "format": "none",
180.     "gauge": {
181.         "maxValue": 100,
182.         "minValue": 0,
183.         "show": false,
184.         "thresholdLabels": false,
185.         "thresholdMarkers": true
186.     },
187.     "gridPos": {
188.         "h": 4,
189.         "w": 4,
190.         "x": 4,
191.         "y": 1
```

```
192.     },
193.     "id": 42,
194.     "interval": null,
195.     "links": [
196.         {
197.             "dashUri": "db/cfs-metanode",
198.             "dashboard": "cfs-metanode",
199.             "includeVars": false,
200.             "keepTime": true,
201.             "targetBlank": true,
202.             "title": "cfs-metanode",
203.             "type": "dashboard"
204.         }
205.     ],
206.     "mappingType": 1,
207.     "mappingTypes": [
208.         {
209.             "name": "value to text",
210.             "value": 1
211.         },
212.         {
213.             "name": "range to text",
214.             "value": 2
215.         }
216.     ],
217.     "maxDataPoints": 100,
218.     "nullPointMode": "connected",
219.     "nullText": null,
220.     "postfix": "",
221.     "postfixFontSize": "50%",
222.     "prefix": "",
223.     "prefixFontSize": "50%",
224.     "rangeMaps": [
225.         {
226.             "from": "null",
227.             "text": "N/A",
228.             "to": "null"
229.         }
230.     ],
231.     "sparkline": {
232.         "fillColor": "rgba(31, 118, 189, 0.18)",
233.         "full": false,
234.         "lineColor": "rgb(31, 120, 193)",
235.         "show": true
236.     },
237.     "tableColumn": "",
238.     "targets": [
239.         {
```



```
240.         "expr": "count(go_info{cluster=~\"$cluster\", app=~\"$app\",
    role=~\"metanode\"})",
241.         "format": "time_series",
242.         "intervalFactor": 1,
243.         "refId": "A"
244.     }
245. ],
246.     "thresholds": "",
247.     "title": "metanode_count",
248.     "type": "singlestat",
249.     "valueFontSize": "80%",
250.     "valueMaps": [
251.     {
252.         "op": "=",
253.         "text": "N/A",
254.         "value": "null"
255.     }
256. ],
257.     "valueName": "current"
258. },
259. {
260.     "cacheTimeout": null,
261.     "colorBackground": false,
262.     "colorValue": false,
263.     "colors": [
264.         "#299c46",
265.         "rgba(237, 129, 40, 0.89)",
266.         "#d44a3a"
267.     ],
268.     "datasource": "${DS_ChubaoFS01}",
269.     "format": "none",
270.     "gauge": {
271.         "maxValue": 100,
272.         "minValue": 0,
273.         "show": false,
274.         "thresholdLabels": false,
275.         "thresholdMarkers": true
276.     },
277.     "gridPos": {
278.         "h": 4,
279.         "w": 4,
280.         "x": 8,
281.         "y": 1
282.     },
283.     "id": 41,
284.     "interval": null,
285.     "links": [
286.     {
```

```
287.         "dashUri": "db/cfs-datanode",
288.         "dashboard": "cfs-datanode",
289.         "includeVars": false,
290.         "keepTime": true,
291.         "targetBlank": true,
292.         "title": "cfs-datanode",
293.         "type": "dashboard"
294.     }
295. ],
296. "mappingType": 1,
297. "mappingTypes": [
298.     {
299.         "name": "value to text",
300.         "value": 1
301.     },
302.     {
303.         "name": "range to text",
304.         "value": 2
305.     }
306. ],
307. "maxDataPoints": 100,
308. "nullPointMode": "connected",
309. "nullText": null,
310. "postfix": "",
311. "postfixFontSize": "50%",
312. "prefix": "",
313. "prefixFontSize": "50%",
314. "rangeMaps": [
315.     {
316.         "from": "null",
317.         "text": "N/A",
318.         "to": "null"
319.     }
320. ],
321. "sparkline": {
322.     "fillColor": "rgba(31, 118, 189, 0.18)",
323.     "full": false,
324.     "lineColor": "rgb(31, 120, 193)",
325.     "show": true
326. },
327. "tableColumn": "",
328. "targets": [
329.     {
330.         "expr": "count(go_info{cluster=~\"$cluster\", app=~\"$app\",
role=~\"dataNode\"})",
331.         "format": "time_series",
332.         "intervalFactor": 1,
333.         "refId": "A"
```

```
334.         }
335.     ],
336.     "thresholds": "",
337.     "title": "datanode_count",
338.     "type": "singlestat",
339.     "valueFontSize": "80%",
340.     "valueMaps": [
341.         {
342.             "op": "=",
343.             "text": "N/A",
344.             "value": "null"
345.         }
346.     ],
347.     "valueName": "current"
348. },
349. {
350.     "gridPos": {
351.         "h": 1,
352.         "w": 24,
353.         "x": 0,
354.         "y": 5
355.     },
356.     "id": 40,
357.     "title": "Cluster",
358.     "type": "row"
359. },
360. {
361.     "aliasColors": {},
362.     "bars": false,
363.     "dashLength": 10,
364.     "dashes": false,
365.     "datasource": "${DS_ChubaoFS01}",
366.     "fill": 1,
367.     "gridPos": {
368.         "h": 6,
369.         "w": 7,
370.         "x": 0,
371.         "y": 6
372.     },
373.     "id": 70,
374.     "legend": {
375.         "avg": false,
376.         "current": false,
377.         "max": false,
378.         "min": false,
379.         "show": true,
380.         "total": false,
381.         "values": false
```

```

382.     },
383.     "lines": true,
384.     "linewidth": 1,
385.     "links": [],
386.     "nullPointMode": "null",
387.     "percentage": false,
388.     "pointradius": 5,
389.     "points": false,
390.     "renderer": "flot",
391.     "seriesOverrides": [],
392.     "spaceLength": 10,
393.     "stack": false,
394.     "steppedLine": false,
395.     "targets": [
396.       {
397.         "expr":
"sum(cfs_metanode_OpCreateMetaPartition{cluster=~\"$cluster\"})",
398.         "format": "time_series",
399.         "intervalFactor": 1,
400.         "legendFormat": "Create",
401.         "refId": "A"
402.       },
403.       {
404.         "expr":
"sum(cfs_metanode_OpLoadMetaPartition{cluster=~\"$cluster\"})",
405.         "format": "time_series",
406.         "intervalFactor": 1,
407.         "legendFormat": "Load",
408.         "refId": "B"
409.       }
410.     ],
411.     "thresholds": [],
412.     "timeFrom": null,
413.     "timeShift": null,
414.     "title": "metanode_OpMetaPartition",
415.     "tooltip": {
416.       "shared": true,
417.       "sort": 0,
418.       "value_type": "individual"
419.     },
420.     "type": "graph",
421.     "xaxis": {
422.       "buckets": null,
423.       "mode": "time",
424.       "name": null,
425.       "show": true,
426.       "values": []
427.     },

```

```
428.     "yaxes": [  
429.         {  
430.             "format": "ns",  
431.             "label": null,  
432.             "logBase": 1,  
433.             "max": null,  
434.             "min": null,  
435.             "show": true  
436.         },  
437.         {  
438.             "format": "short",  
439.             "label": null,  
440.             "logBase": 1,  
441.             "max": null,  
442.             "min": null,  
443.             "show": true  
444.         }  
445.     ],  
446.     "yaxis": {  
447.         "align": false,  
448.         "alignLevel": null  
449.     }  
450. },  
451. {  
452.     "aliasColors": {},  
453.     "bars": false,  
454.     "dashLength": 10,  
455.     "dashes": false,  
456.     "datasource": "${DS_ChubaoFS01}",  
457.     "fill": 1,  
458.     "gridPos": {  
459.         "h": 6,  
460.         "w": 7,  
461.         "x": 7,  
462.         "y": 6  
463.     },  
464.     "id": 71,  
465.     "legend": {  
466.         "avg": false,  
467.         "current": false,  
468.         "max": false,  
469.         "min": false,  
470.         "show": true,  
471.         "total": false,  
472.         "values": false  
473.     },  
474.     "lines": true,  
475.     "linewidth": 1,
```

```

476.     "links": [],
477.     "nullPointMode": "null",
478.     "percentage": false,
479.     "pointradius": 5,
480.     "points": false,
481.     "renderer": "flot",
482.     "seriesOverrides": [],
483.     "spaceLength": 10,
484.     "stack": false,
485.     "steppedLine": false,
486.     "targets": [
487.         {
488.             "expr":
"sum(cfs_metanode_OpMetaBatchInodeGet{cluster=~\"$cluster\"})",
489.             "format": "time_series",
490.             "intervalFactor": 1,
491.             "legendFormat": "BatchGet",
492.             "refId": "A"
493.         },
494.         {
495.             "expr": "sum(cfs_metanode_OpMetaCreateInode{cluster=~\"$cluster\"})",
496.             "format": "time_series",
497.             "intervalFactor": 1,
498.             "legendFormat": "Create",
499.             "refId": "B"
500.         },
501.         {
502.             "expr": "sum(cfs_metanode_OpMetaDeleteInode{cluster=~\"$cluster\"})",
503.             "format": "time_series",
504.             "intervalFactor": 1,
505.             "legendFormat": "Delete",
506.             "refId": "C"
507.         },
508.         {
509.             "expr": "sum(cfs_metanode_OpMetaEvictInode{cluster=~\"$cluster\"})",
510.             "format": "time_series",
511.             "intervalFactor": 1,
512.             "legendFormat": "Evict",
513.             "refId": "D"
514.         },
515.         {
516.             "expr": "sum(cfs_metanode_OpMetaInodeGet{cluster=~\"$cluster\"})",
517.             "format": "time_series",
518.             "intervalFactor": 1,
519.             "legendFormat": "Get",
520.             "refId": "E"
521.         },
522.         {

```

```
523.         "expr": "sum(cfs_metanode_OpMetaLinkInode{cluster=~\"$cluster\"})",
524.         "format": "time_series",
525.         "intervalFactor": 1,
526.         "legendFormat": "Link",
527.         "refId": "F"
528.     }
529. ],
530.     "thresholds": [],
531.     "timeFrom": null,
532.     "timeShift": null,
533.     "title": "metanode_OpMetaInode",
534.     "tooltip": {
535.         "shared": true,
536.         "sort": 0,
537.         "value_type": "individual"
538.     },
539.     "type": "graph",
540.     "xaxis": {
541.         "buckets": null,
542.         "mode": "time",
543.         "name": null,
544.         "show": true,
545.         "values": []
546.     },
547.     "yaxes": [
548.         {
549.             "format": "ns",
550.             "label": null,
551.             "logBase": 1,
552.             "max": null,
553.             "min": null,
554.             "show": true
555.         },
556.         {
557.             "format": "short",
558.             "label": null,
559.             "logBase": 1,
560.             "max": null,
561.             "min": null,
562.             "show": true
563.         }
564.     ],
565.     "yaxis": {
566.         "align": false,
567.         "alignLevel": null
568.     }
569. },
570. {
```

```
571.     "aliasColors": {},
572.     "bars": false,
573.     "dashLength": 10,
574.     "dashes": false,
575.     "datasource": "${DS_ChubaoFS01}",
576.     "fill": 1,
577.     "gridPos": {
578.         "h": 6,
579.         "w": 6,
580.         "x": 14,
581.         "y": 6
582.     },
583.     "id": 45,
584.     "legend": {
585.         "avg": false,
586.         "current": false,
587.         "max": false,
588.         "min": false,
589.         "show": true,
590.         "total": false,
591.         "values": false
592.     },
593.     "lines": true,
594.     "linewidth": 1,
595.     "links": [],
596.     "nullPointMode": "null",
597.     "percentage": false,
598.     "pointradius": 5,
599.     "points": false,
600.     "renderer": "flot",
601.     "seriesOverrides": [],
602.     "spaceLength": 10,
603.     "stack": false,
604.     "steppedLine": false,
605.     "targets": [
606.         {
607.             "expr":
"sum(cfs_metanode_OpMetaCreateDentry{cluster=~\"$cluster\"})",
608.             "format": "time_series",
609.             "intervalFactor": 1,
610.             "legendFormat": "Create",
611.             "refId": "A"
612.         },
613.         {
614.             "expr":
"sum(cfs_metanode_OpMetaDeleteDentry{cluster=~\"$cluster\"})",
615.             "format": "time_series",
616.             "intervalFactor": 1,
```



```

617.         "legendFormat": "Delete",
618.         "refId": "B"
619.     },
620.     {
621.         "expr":
"sum(cfs_metanode_OpMetaUpdateDentry{cluster=~\"$cluster\"})",
622.         "format": "time_series",
623.         "intervalFactor": 1,
624.         "legendFormat": "Update",
625.         "refId": "C"
626.     }
627. ],
628. "thresholds": [],
629. "timeFrom": null,
630. "timeShift": null,
631. "title": "metanode_OpMetaDentry",
632. "tooltip": {
633.     "shared": true,
634.     "sort": 0,
635.     "value_type": "individual"
636. },
637. "type": "graph",
638. "xaxis": {
639.     "buckets": null,
640.     "mode": "time",
641.     "name": null,
642.     "show": true,
643.     "values": []
644. },
645. "yaxes": [
646.     {
647.         "format": "ns",
648.         "label": null,
649.         "logBase": 1,
650.         "max": null,
651.         "min": null,
652.         "show": true
653.     },
654.     {
655.         "format": "short",
656.         "label": null,
657.         "logBase": 1,
658.         "max": null,
659.         "min": null,
660.         "show": true
661.     }
662. ],
663. "yaxis": {

```

```
664.         "align": false,
665.         "alignLevel": null
666.     },
667. },
668. {
669.     "aliasColors": {},
670.     "bars": false,
671.     "dashLength": 10,
672.     "dashes": false,
673.     "datasource": "${DS_ChubaoFS01}",
674.     "fill": 1,
675.     "gridPos": {
676.         "h": 6,
677.         "w": 7,
678.         "x": 0,
679.         "y": 12
680.     },
681.     "id": 79,
682.     "legend": {
683.         "avg": false,
684.         "current": false,
685.         "max": false,
686.         "min": false,
687.         "show": true,
688.         "total": false,
689.         "values": false
690.     },
691.     "lines": true,
692.     "linewidth": 1,
693.     "links": [],
694.     "nullPointMode": "null",
695.     "percentage": false,
696.     "pointradius": 5,
697.     "points": false,
698.     "renderer": "flot",
699.     "seriesOverrides": [],
700.     "spaceLength": 10,
701.     "stack": false,
702.     "steppedLine": false,
703.     "targets": [
704.         {
705.             "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_CreateFile{cluster=~\"$cluster\})",
706.             "format": "time_series",
707.             "intervalFactor": 1,
708.             "legendFormat": "CreateFile",
709.             "refId": "A"
710.         },
```

```

711.         {
712.             "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_MarkDelete{cluster=~\"$cluster\"})",
713.             "format": "time_series",
714.             "intervalFactor": 1,
715.             "legendFormat": "MarkDelete",
716.             "refId": "B"
717.         },
718.         {
719.             "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_Read{cluster=~\"$cluster\"})",
720.             "format": "time_series",
721.             "intervalFactor": 1,
722.             "legendFormat": "Read",
723.             "refId": "C"
724.         },
725.         {
726.             "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_Write{cluster=~\"$cluster\"})",
727.             "format": "time_series",
728.             "intervalFactor": 1,
729.             "legendFormat": "Write",
730.             "refId": "D"
731.         },
732.         {
733.             "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_RandomWrite{cluster=~\"$cluster\"})",
734.             "format": "time_series",
735.             "intervalFactor": 1,
736.             "legendFormat": "RandomWrite",
737.             "refId": "E"
738.         }
739.     ],
740.     "thresholds": [],
741.     "timeFrom": null,
742.     "timeShift": null,
743.     "title": "datanode_CreateFile",
744.     "tooltip": {
745.         "shared": true,
746.         "sort": 0,
747.         "value_type": "individual"
748.     },
749.     "type": "graph",
750.     "xaxis": {
751.         "buckets": null,
752.         "mode": "time",
753.         "name": null,
754.         "show": true,

```

```
755.         "values": []
756.     },
757.     "yaxes": [
758.         {
759.             "format": "ns",
760.             "label": null,
761.             "logBase": 1,
762.             "max": null,
763.             "min": null,
764.             "show": true
765.         },
766.         {
767.             "format": "short",
768.             "label": null,
769.             "logBase": 1,
770.             "max": null,
771.             "min": null,
772.             "show": true
773.         }
774.     ],
775.     "yaxis": {
776.         "align": false,
777.         "alignLevel": null
778.     }
779. },
780. {
781.     "aliasColors": {},
782.     "bars": false,
783.     "dashLength": 10,
784.     "dashes": false,
785.     "datasource": "${DS_ChubaoFS01}",
786.     "fill": 1,
787.     "gridPos": {
788.         "h": 6,
789.         "w": 7,
790.         "x": 7,
791.         "y": 12
792.     },
793.     "id": 75,
794.     "legend": {
795.         "avg": false,
796.         "current": false,
797.         "max": false,
798.         "min": false,
```

```

1.         "show": true,
2.         "total": false,
3.         "values": false
4.     },
5.     "lines": true,
6.     "linewidth": 1,
7.     "links": [],
8.     "nullPointMode": "null",
9.     "percentage": false,
10.    "pointradius": 5,
11.    "points": false,
12.    "renderer": "flot",
13.    "seriesOverrides": [],
14.    "spaceLength": 10,
15.    "stack": false,
16.    "steppedLine": false,
17.    "targets": [
18.        {
19.            "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_OpLoadDataPartition{cluster=~\"$cluster\\
\"})",
20.            "format": "time_series",
21.            "intervalFactor": 1,
22.            "legendFormat": "OpLoadDataPartition",
23.            "refId": "G"
24.        },
25.        {
26.            "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_OpDataNodeHeartbeat{cluster=~\"$cluster\\
\"})",
27.            "format": "time_series",
28.            "intervalFactor": 1,
29.            "legendFormat": "OpDataNodeHeartbeat",
30.            "refId": "F"
31.        },
32.        {
33.            "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_OpGetPartitionSize{cluster=~\"$cluster\\
\"})",
34.            "format": "time_series",
35.            "intervalFactor": 1,
36.            "legendFormat": "OpGetPartitionSize",
37.            "refId": "H"
38.        },
39.        {
40.            "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_OpGetAppliedId{cluster=~\"$cluster\\
\"})",

```

```

41.         "format": "time_series",
42.         "intervalFactor": 1,
43.         "legendFormat": "OpGetAppliedId",
44.         "refId": "I"
45.     },
46.     {
47.         "expr":
48.         "sum(cfs_dataNode_[[cluster]]_datanode_OpCreateDataPartition{cluster=~\"$cluster\"})",
49.         "format": "time_series",
50.         "intervalFactor": 1,
51.         "legendFormat": "OpCreateDataPartition",
52.         "refId": "A"
53.     }
54. ],
55. "thresholds": [],
56. "timeFrom": null,
57. "timeShift": null,
58. "title": "datanode_Op",
59. "tooltip": {
60.     "shared": true,
61.     "sort": 0,
62.     "value_type": "individual"
63. },
64. "type": "graph",
65. "xaxis": {
66.     "buckets": null,
67.     "mode": "time",
68.     "name": null,
69.     "show": true,
70.     "values": []
71. },
72. "yaxes": [
73.     {
74.         "format": "ns",
75.         "label": null,
76.         "logBase": 1,
77.         "max": null,
78.         "min": null,
79.         "show": true
80.     },
81.     {
82.         "format": "short",
83.         "label": null,
84.         "logBase": 1,
85.         "max": null,
86.         "min": null,
87.         "show": true

```

```
87.     }
88.   ],
89.   "yaxis": {
90.     "align": false,
91.     "alignLevel": null
92.   }
93. },
94. {
95.   "aliasColors": {},
96.   "bars": false,
97.   "dashLength": 10,
98.   "dashes": false,
99.   "datasource": "${DS_ChubaoFS01}",
100.  "fill": 1,
101.  "gridPos": {
102.    "h": 6,
103.    "w": 6,
104.    "x": 14,
105.    "y": 12
106.  },
107.  "id": 73,
108.  "legend": {
109.    "avg": false,
110.    "current": false,
111.    "max": false,
112.    "min": false,
113.    "show": true,
114.    "total": false,
115.    "values": false
116.  },
117.  "lines": true,
118.  "linewidth": 1,
119.  "links": [],
120.  "nullPointMode": "null",
121.  "percentage": false,
122.  "pointradius": 5,
123.  "points": false,
124.  "renderer": "flot",
125.  "seriesOverrides": [],
126.  "spaceLength": 10,
127.  "stack": false,
128.  "steppedLine": false,
129.  "targets": [
130.    {
131.      "expr": "sum(cfs_metanode_OpMetaOpen{cluster=~\"$cluster\"})",
132.      "format": "time_series",
133.      "intervalFactor": 1,
134.      "legendFormat": "Open",
```

```

135.         "refId": "A"
136.     },
137.     {
138.         "expr": "sum(cfs_metanode_OpMetaLookup{cluster=~\"$cluster\"})",
139.         "format": "time_series",
140.         "intervalFactor": 1,
141.         "legendFormat": "Lookup",
142.         "refId": "B"
143.     },
144.     {
145.         "expr":
146.         "sum(cfs_metanode_OpMetaNodeHeartbeat{cluster=~\"$cluster\"})",
147.         "format": "time_series",
148.         "intervalFactor": 1,
149.         "legendFormat": "NodeHeartbeat",
150.         "refId": "C"
151.     },
152.     {
153.         "expr": "sum(cfs_metanode_OpMetaReadDir{cluster=~\"$cluster\"})",
154.         "format": "time_series",
155.         "intervalFactor": 1,
156.         "legendFormat": "ReadDir",
157.         "refId": "D"
158.     },
159.     {
160.         "expr": "sum(cfs_metanode_OpMetaReleaseOpen{cluster=~\"$cluster\"})",
161.         "format": "time_series",
162.         "intervalFactor": 1,
163.         "legendFormat": "ReleaseOpen",
164.         "refId": "E"
165.     },
166.     {
167.         "expr": "sum(cfs_metanode_OpMetaSetattr{cluster=~\"$cluster\"})",
168.         "format": "time_series",
169.         "intervalFactor": 1,
170.         "legendFormat": "Setattr",
171.         "refId": "F"
172.     },
173.     {
174.         "expr": "sum(cfs_metanode_OpMetaTruncate{cluster=~\"$cluster\"})",
175.         "format": "time_series",
176.         "intervalFactor": 1,
177.         "legendFormat": "Truncate",
178.         "refId": "G"
179.     }
180. ],
181. "thresholds": [],
182. "timeFrom": null,

```



```
182.     "timeShift": null,
183.     "title": "metanode_OpMeta",
184.     "tooltip": {
185.         "shared": true,
186.         "sort": 0,
187.         "value_type": "individual"
188.     },
189.     "type": "graph",
190.     "xaxis": {
191.         "buckets": null,
192.         "mode": "time",
193.         "name": null,
194.         "show": true,
195.         "values": []
196.     },
197.     "yaxes": [
198.         {
199.             "format": "ns",
200.             "label": null,
201.             "logBase": 1,
202.             "max": null,
203.             "min": null,
204.             "show": true
205.         },
206.         {
207.             "format": "short",
208.             "label": null,
209.             "logBase": 1,
210.             "max": null,
211.             "min": null,
212.             "show": true
213.         }
214.     ],
215.     "yaxis": {
216.         "align": false,
217.         "alignLevel": null
218.     }
219. },
220. {
221.     "aliasColors": {},
222.     "bars": false,
223.     "dashLength": 10,
224.     "dashes": false,
225.     "datasource": "${DS_ChubaoFS01}",
226.     "fill": 1,
227.     "gridPos": {
228.         "h": 7,
229.         "w": 7,
```

```

230.         "x": 0,
231.         "y": 18
232.     },
233.     "id": 80,
234.     "legend": {
235.         "avg": false,
236.         "current": false,
237.         "max": false,
238.         "min": false,
239.         "show": true,
240.         "total": false,
241.         "values": false
242.     },
243.     "lines": true,
244.     "linewidth": 1,
245.     "links": [],
246.     "nullPointMode": "null",
247.     "percentage": false,
248.     "pointradius": 5,
249.     "points": false,
250.     "renderer": "flot",
251.     "seriesOverrides": [],
252.     "spaceLength": 10,
253.     "stack": false,
254.     "steppedLine": false,
255.     "targets": [
256.         {
257.             "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_ExtentRepairRead{cluster=~\"$cluster\"})"
258.         ,
259.             "format": "time_series",
260.             "intervalFactor": 1,
261.             "legendFormat": "ExtentRepairRead",
262.             "refId": "B"
263.         },
264.         {
265.             "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{cluster=~\"$cluster\"})",
266.             "format": "time_series",
267.             "intervalFactor": 1,
268.             "legendFormat": "GetAllExtentWatermark",
269.             "refId": "C"
270.         },
271.         {
272.             "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_NotifyExtentRepair{cluster=~\"$cluster\"
})",

```

```
272.         "format": "time_series",
273.         "intervalFactor": 1,
274.         "legendFormat": "NotifyExtentRepair",
275.         "refId": "D"
276.     }
277. ],
278. "thresholds": [],
279. "timeFrom": null,
280. "timeShift": null,
281. "title": "datanode_Extent",
282. "tooltip": {
283.     "shared": true,
284.     "sort": 0,
285.     "value_type": "individual"
286. },
287. "type": "graph",
288. "xaxis": {
289.     "buckets": null,
290.     "mode": "time",
291.     "name": null,
292.     "show": true,
293.     "values": []
294. },
295. "yaxes": [
296.     {
297.         "format": "ns",
298.         "label": null,
299.         "logBase": 1,
300.         "max": null,
301.         "min": null,
302.         "show": true
303.     },
304.     {
305.         "format": "short",
306.         "label": null,
307.         "logBase": 1,
308.         "max": null,
309.         "min": null,
310.         "show": true
311.     }
312. ],
313. "yaxis": {
314.     "align": false,
315.     "alignLevel": null
316. }
317. },
318. {
319.     "aliasColors": {},
```

```
320.     "bars": false,
321.     "dashLength": 10,
322.     "dashes": false,
323.     "datasource": "${DS_ChubaoFS01}",
324.     "fill": 1,
325.     "gridPos": {
326.         "h": 7,
327.         "w": 7,
328.         "x": 7,
329.         "y": 18
330.     },
331.     "id": 83,
332.     "legend": {
333.         "avg": false,
334.         "current": false,
335.         "max": false,
336.         "min": false,
337.         "show": true,
338.         "total": false,
339.         "values": false
340.     },
341.     "lines": true,
342.     "linewidth": 1,
343.     "links": [],
344.     "nullPointMode": "null",
345.     "percentage": false,
346.     "pointradius": 5,
347.     "points": false,
348.     "renderer": "flot",
349.     "seriesOverrides": [],
350.     "spaceLength": 10,
351.     "stack": false,
352.     "steppedLine": false,
353.     "targets": [
354.         {
355.             "expr":
356.             "sum(cfs_dataNode_[[cluster]]_datanode_streamRead{cluster=~\"$cluster\"})",
357.             "format": "time_series",
358.             "intervalFactor": 1,
359.             "legendFormat": "streamRead",
360.             "refId": "M"
361.         },
362.         {
363.             "expr":
364.             "sum(cfs_dataNode_[[cluster]]_datanode_streamWrite{cluster=~\"$cluster\"})",
365.             "format": "time_series",
366.             "intervalFactor": 1,
367.             "legendFormat": "streamWrite",
```

```

366.         "refId": "N"
367.     },
368.     {
369.         "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_streamCreateFile{cluster=~\"$cluster\\\"})"
,
370.         "format": "time_series",
371.         "intervalFactor": 1,
372.         "legendFormat": "streamCreateFile",
373.         "refId": "A"
374.     },
375.     {
376.         "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_streamExtentRepairRead{cluster=~\"$cluster\\\"})",
377.         "format": "time_series",
378.         "intervalFactor": 1,
379.         "legendFormat": "streamExtentRepairRead",
380.         "refId": "B"
381.     },
382.     {
383.         "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_streamMarkDelete{cluster=~\"$cluster\\\"})"
,
384.         "format": "time_series",
385.         "intervalFactor": 1,
386.         "legendFormat": "streamMarkDelete",
387.         "refId": "C"
388.     },
389.     {
390.         "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_streamOpGetAppliedId{cluster=~\"$cluster\\\"})",
391.         "format": "time_series",
392.         "intervalFactor": 1,
393.         "legendFormat": "streamOpGetAppliedId",
394.         "refId": "D"
395.     },
396.     {
397.         "expr":
"sum(cfs_dataNode_[[cluster]]_datanode_streamOpGetPartitionSize{cluster=~\"$cluster\\\"})",
398.         "format": "time_series",
399.         "intervalFactor": 1,
400.         "legendFormat": "streamOpGetPartitionSize",
401.         "refId": "E"
402.     },
403.     {

```

```
404.         "expr":
405.         "sum(cfs_dataNode_[[cluster]]_datanode_streamNotifyExtentRepair{cluster=~\"$clu
406.         ster\\}\")",
407.         "format": "time_series",
408.         "intervalFactor": 1,
409.         "legendFormat": "streamNotifyExtentRepair",
410.         "refId": "F"
411.     }
412. ],
413. "thresholds": [],
414. "timeFrom": null,
415. "timeShift": null,
416. "title": "datanode_Stream",
417. "tooltip": {
418.     "shared": true,
419.     "sort": 0,
420.     "value_type": "individual"
421. },
422. "type": "graph",
423. "xaxis": {
424.     "buckets": null,
425.     "mode": "time",
426.     "name": null,
427.     "show": true,
428.     "values": []
429. },
430. "yaxes": [
431.     {
432.         "format": "ns",
433.         "label": null,
434.         "logBase": 1,
435.         "max": null,
436.         "min": null,
437.         "show": true
438.     },
439.     {
440.         "format": "short",
441.         "label": null,
442.         "logBase": 1,
443.         "max": null,
444.         "min": null,
445.         "show": true
446.     }
447. ],
448. "yaxis": {
449.     "align": false,
450.     "alignLevel": null
451. }
```

```
450.     },
451.     {
452.         "collapsed": false,
453.         "gridPos": {
454.             "h": 1,
455.             "w": 24,
456.             "x": 0,
457.             "y": 25
458.         },
459.         "id": 60,
460.         "panels": [],
461.         "title": "GoRuntime",
462.         "type": "row"
463.     },
464.     {
465.         "aliasColors": {},
466.         "bars": false,
467.         "dashLength": 10,
468.         "dashes": false,
469.         "datasource": "${DS_ChubaoFS01}",
470.         "fill": 1,
471.         "gridPos": {
472.             "h": 6,
473.             "w": 7,
474.             "x": 0,
475.             "y": 26
476.         },
477.         "id": 61,
478.         "legend": {
479.             "avg": false,
480.             "current": false,
481.             "max": false,
482.             "min": false,
483.             "show": true,
484.             "total": false,
485.             "values": false
486.         },
487.         "lines": true,
488.         "linewidth": 1,
489.         "links": [],
490.         "nullPointMode": "null",
491.         "percentage": false,
492.         "pointradius": 5,
493.         "points": false,
494.         "renderer": "flot",
495.         "seriesOverrides": [],
496.         "spaceLength": 10,
497.         "stack": false,
```

```
498.     "steppedLine": false,
499.     "targets": [
500.         {
501.             "expr": "go_goroutines{instance=~\"$instance\"}",
502.             "format": "time_series",
503.             "intervalFactor": 1,
504.             "legendFormat": "go_goroutines",
505.             "refId": "A"
506.         },
507.         {
508.             "expr": "go_threads{instance=~\"$instance\"}",
509.             "format": "time_series",
510.             "intervalFactor": 1,
511.             "legendFormat": "go_threads",
512.             "refId": "B"
513.         }
514.     ],
515.     "thresholds": [],
516.     "timeFrom": null,
517.     "timeShift": null,
518.     "title": "go info",
519.     "tooltip": {
520.         "shared": true,
521.         "sort": 0,
522.         "value_type": "individual"
523.     },
524.     "type": "graph",
525.     "xaxis": {
526.         "buckets": null,
527.         "mode": "time",
528.         "name": null,
529.         "show": true,
530.         "values": []
531.     },
532.     "yaxes": [
533.         {
534.             "format": "locale",
535.             "label": null,
536.             "logBase": 1,
537.             "max": null,
538.             "min": null,
539.             "show": true
540.         },
541.         {
542.             "format": "short",
543.             "label": null,
544.             "logBase": 1,
545.             "max": null,
```



```
546.         "min": null,
547.         "show": true
548.     }
549. ],
550. "yaxis": {
551.     "align": false,
552.     "alignLevel": null
553. }
554. },
555. {
556.     "aliasColors": {},
557.     "bars": false,
558.     "dashLength": 10,
559.     "dashes": false,
560.     "datasource": "${DS_ChubaoFS01}",
561.     "fill": 1,
562.     "gridPos": {
563.         "h": 6,
564.         "w": 7,
565.         "x": 7,
566.         "y": 26
567.     },
568.     "id": 62,
569.     "legend": {
570.         "avg": false,
571.         "current": false,
572.         "max": false,
573.         "min": false,
574.         "show": true,
575.         "total": false,
576.         "values": false
577.     },
578.     "lines": true,
579.     "linewidth": 1,
580.     "links": [],
581.     "nullPointMode": "null",
582.     "percentage": false,
583.     "pointradius": 5,
584.     "points": false,
585.     "renderer": "flot",
586.     "seriesOverrides": [],
587.     "spaceLength": 10,
588.     "stack": false,
589.     "steppedLine": false,
590.     "targets": [
591.         {
592.             "expr": "go_memstats_alloc_bytes{instance=~\"$instance\"}",
593.             "format": "time_series",
```

```
594.         "intervalFactor": 1,
595.         "legendFormat": "alloc_bytes",
596.         "refId": "A"
597.     },
598.     {
599.         "expr": "go_memstats_alloc_bytes_total{instance=~\"$instance\"}",
600.         "format": "time_series",
601.         "intervalFactor": 1,
602.         "legendFormat": "alloc_bytes_total",
603.         "refId": "B"
604.     },
605.     {
606.         "expr": "go_memstats_heap_alloc_bytes{instance=~\"$instance\"}",
607.         "format": "time_series",
608.         "intervalFactor": 1,
609.         "legendFormat": "heap_alloc_bytes",
610.         "refId": "C"
611.     },
612.     {
613.         "expr": "go_memstats_heap_inuse_bytes{instance=~\"$instance\"}",
614.         "format": "time_series",
615.         "intervalFactor": 1,
616.         "legendFormat": "heap_inuse_bytes",
617.         "refId": "D"
618.     },
619.     {
620.         "expr": "go_memstats_sys_bytes{instance=~\"$instance\"}",
621.         "format": "time_series",
622.         "intervalFactor": 1,
623.         "legendFormat": "sys_bytes",
624.         "refId": "E"
625.     }
626. ],
627. "thresholds": [],
628. "timeFrom": null,
629. "timeShift": null,
630. "title": "go_memstats",
631. "tooltip": {
632.     "shared": true,
633.     "sort": 0,
634.     "value_type": "individual"
635. },
636. "type": "graph",
637. "xaxis": {
638.     "buckets": null,
639.     "mode": "time",
640.     "name": null,
641.     "show": true,
```

```
642.         "values": []
643.     },
644.     "yaxes": [
645.         {
646.             "format": "decbytes",
647.             "label": null,
648.             "logBase": 1,
649.             "max": null,
650.             "min": null,
651.             "show": true
652.         },
653.         {
654.             "format": "short",
655.             "label": null,
656.             "logBase": 1,
657.             "max": null,
658.             "min": null,
659.             "show": true
660.         }
661.     ],
662.     "yaxis": {
663.         "align": false,
664.         "alignLevel": null
665.     }
666. },
667. {
668.     "aliasColors": {},
669.     "bars": false,
670.     "dashLength": 10,
671.     "dashes": false,
672.     "datasource": "${DS_ChubaoFS01}",
673.     "fill": 1,
674.     "gridPos": {
675.         "h": 6,
676.         "w": 7,
677.         "x": 14,
678.         "y": 26
679.     },
680.     "id": 63,
681.     "legend": {
682.         "avg": false,
683.         "current": false,
684.         "max": false,
685.         "min": false,
686.         "show": true,
687.         "total": false,
688.         "values": false
689.     },
```

```

690.     "lines": true,
691.     "linewidth": 1,
692.     "links": [],
693.     "nullPointMode": "null",
694.     "percentage": false,
695.     "pointradius": 5,
696.     "points": false,
697.     "renderer": "flot",
698.     "seriesOverrides": [
699.         {
700.             "alias": "gc_rate",
701.             "yaxis": 2
702.         }
703.     ],
704.     "spaceLength": 10,
705.     "stack": false,
706.     "steppedLine": false,
707.     "targets": [
708.         {
709.             "expr": "go_gc_duration_seconds{instance=~\"$instance\"}",
710.             "format": "time_series",
711.             "intervalFactor": 1,
712.             "legendFormat": "seconds_{quantile}",
713.             "refId": "A"
714.         },
715.         {
716.             "expr": "rate(go_gc_duration_seconds_count{instance=~\"$instance\"}
[1m])",
717.             "format": "time_series",
718.             "intervalFactor": 1,
719.             "legendFormat": "gc_rate",
720.             "refId": "B"
721.         }
722.     ],
723.     "thresholds": [],
724.     "timeFrom": null,
725.     "timeShift": null,
726.     "title": "gc_duration",
727.     "tooltip": {
728.         "shared": true,
729.         "sort": 0,
730.         "value_type": "individual"
731.     },
732.     "type": "graph",
733.     "xaxis": {
734.         "buckets": null,
735.         "mode": "time",
736.         "name": null,

```

```
737.         "show": true,
738.         "values": []
739.     },
740.     "yaxes": [
741.         {
742.             "format": "s",
743.             "label": null,
744.             "logBase": 1,
745.             "max": null,
746.             "min": null,
747.             "show": true
748.         },
749.         {
750.             "format": "locale",
751.             "label": null,
752.             "logBase": 1,
753.             "max": null,
754.             "min": null,
755.             "show": true
756.         }
757.     ],
758.     "yaxis": {
759.         "align": false,
760.         "alignLevel": null
761.     }
762. },
763. {
764.     "collapsed": false,
765.     "gridPos": {
766.         "h": 1,
767.         "w": 24,
768.         "x": 0,
769.         "y": 32
770.     },
771.     "id": 34,
772.     "panels": [],
773.     "title": "Master",
774.     "type": "row"
775. },
776. {
777.     "collapsed": false,
778.     "gridPos": {
779.         "h": 1,
780.         "w": 24,
781.         "x": 0,
782.         "y": 33
783.     },
784.     "id": 36,
```

```
785.     "panels": [],
786.     "title": "Metanode",
787.     "type": "row"
788. },
789. {
790.     "aliasColors": {},
791.     "bars": false,
792.     "dashLength": 10,
793.     "dashes": false,
794.     "datasource": "${DS_ChubaoFS01}",
795.     "fill": 1,
796.     "gridPos": {
797.         "h": 8,
798.         "w": 8,
799.         "x": 0,
800.         "y": 34
801.     },
802.     "id": 58,
803.     "legend": {
804.         "avg": false,
805.         "current": false,
806.         "max": false,
807.         "min": false,
808.         "show": true,
```

```
1.         "total": false,
2.         "values": false
3.     },
4.     "lines": true,
5.     "linewidth": 1,
6.     "links": [],
7.     "nullPointMode": "null",
8.     "percentage": false,
9.     "pointradius": 5,
10.    "points": false,
11.    "renderer": "flot",
12.    "seriesOverrides": [],
13.    "spaceLength": 10,
14.    "stack": false,
15.    "steppedLine": false,
16.    "targets": [
17.        {
18.            "expr":
19.            "cfs_metanode_OpCreateMetaPartition{instance=~\"$instance\"}",
20.            "format": "time_series",
21.            "intervalFactor": 1,
22.            "legendFormat": "Create",
23.            "refId": "A"
24.        },
25.        {
26.            "expr": "cfs_metanode_OpLoadMetaPartition{instance=~\"$instance\"}",
27.            "format": "time_series",
28.            "intervalFactor": 1,
29.            "legendFormat": "Load",
30.            "refId": "B"
31.        }
32.    ],
33.    "thresholds": [],
34.    "timeFrom": null,
35.    "timeShift": null,
36.    "title": "metanode_OpMetaPartition",
37.    "tooltip": {
38.        "shared": true,
39.        "sort": 0,
40.        "value_type": "individual"
41.    },
42.    "type": "graph",
43.    "xaxis": {
44.        "buckets": null,
45.        "mode": "time",
46.        "name": null,
```

```
47.     "values": []
48.   },
49.   "yaxes": [
50.     {
51.       "format": "ns",
52.       "label": null,
53.       "logBase": 1,
54.       "max": null,
55.       "min": null,
56.       "show": true
57.     },
58.     {
59.       "format": "short",
60.       "label": null,
61.       "logBase": 1,
62.       "max": null,
63.       "min": null,
64.       "show": true
65.     }
66.   ],
67.   "yaxis": {
68.     "align": false,
69.     "alignLevel": null
70.   }
71. },
72. {
73.   "aliasColors": {},
74.   "bars": false,
75.   "dashLength": 10,
76.   "dashes": false,
77.   "datasource": "${DS_ChubaoFS01}",
78.   "fill": 1,
79.   "gridPos": {
80.     "h": 8,
81.     "w": 8,
82.     "x": 8,
83.     "y": 34
84.   },
85.   "id": 44,
86.   "legend": {
87.     "avg": false,
88.     "current": false,
89.     "max": false,
90.     "min": false,
91.     "show": true,
92.     "total": false,
93.     "values": false
94.   },
```



```
95.     "lines": true,  
96.     "linewidth": 1,  
97.     "links": [],  
98.     "nullPointMode": "null",  
99.     "percentage": false,  
100.    "pointradius": 5,  
101.    "points": false,  
102.    "renderer": "flot",  
103.    "seriesOverrides": [],  
104.    "spaceLength": 10,  
105.    "stack": false,  
106.    "steppedLine": false,  
107.    "targets": [  
108.        {  
109.            "expr": "cfs_metanode_OpMetaBatchInodeGet{instance=~\"$instance\"}",  
110.            "format": "time_series",  
111.            "intervalFactor": 1,  
112.            "legendFormat": "BatchGet",  
113.            "refId": "A"  
114.        },  
115.        {  
116.            "expr": "cfs_metanode_OpMetaCreateInode{instance=~\"$instance\"}",  
117.            "format": "time_series",  
118.            "intervalFactor": 1,  
119.            "legendFormat": "Create",  
120.            "refId": "B"  
121.        },  
122.        {  
123.            "expr": "cfs_metanode_OpMetaDeleteInode{instance=~\"$instance\"}",  
124.            "format": "time_series",  
125.            "intervalFactor": 1,  
126.            "legendFormat": "Delete",  
127.            "refId": "C"  
128.        },  
129.        {  
130.            "expr": "cfs_metanode_OpMetaEvictInode{instance=~\"$instance\"}",  
131.            "format": "time_series",  
132.            "intervalFactor": 1,  
133.            "legendFormat": "Evict",  
134.            "refId": "D"  
135.        },  
136.        {  
137.            "expr": "cfs_metanode_OpMetaInodeGet{instance=~\"$instance\"}",  
138.            "format": "time_series",  
139.            "intervalFactor": 1,  
140.            "legendFormat": "Get",  
141.            "refId": "E"  
142.        },  
143.    ]  
144. }
```

```
143.         {
144.             "expr": "cfs_metanode_OpMetaLinkInode{instance=~\"$instance\"}",
145.             "format": "time_series",
146.             "intervalFactor": 1,
147.             "legendFormat": "Link",
148.             "refId": "F"
149.         }
150.     ],
151.     "thresholds": [],
152.     "timeFrom": null,
153.     "timeShift": null,
154.     "title": "metanode_OpMetaInode",
155.     "tooltip": {
156.         "shared": true,
157.         "sort": 0,
158.         "value_type": "individual"
159.     },
160.     "type": "graph",
161.     "xaxis": {
162.         "buckets": null,
163.         "mode": "time",
164.         "name": null,
165.         "show": true,
166.         "values": []
167.     },
168.     "yaxes": [
169.         {
170.             "format": "ns",
171.             "label": null,
172.             "logBase": 1,
173.             "max": null,
174.             "min": null,
175.             "show": true
176.         },
177.         {
178.             "format": "short",
179.             "label": null,
180.             "logBase": 1,
181.             "max": null,
182.             "min": null,
183.             "show": true
184.         }
185.     ],
186.     "yaxis": {
187.         "align": false,
188.         "alignLevel": null
189.     }
190. },
```

```
191.     {
192.         "aliasColors": {},
193.         "bars": false,
194.         "dashLength": 10,
195.         "dashes": false,
196.         "datasource": "${DS_ChubaoFS01}",
197.         "fill": 1,
198.         "gridPos": {
199.             "h": 8,
200.             "w": 8,
201.             "x": 16,
202.             "y": 34
203.         },
204.         "id": 72,
205.         "legend": {
206.             "avg": false,
207.             "current": false,
208.             "max": false,
209.             "min": false,
210.             "show": true,
211.             "total": false,
212.             "values": false
213.         },
214.         "lines": true,
215.         "linewidth": 1,
216.         "links": [],
217.         "nullPointMode": "null",
218.         "percentage": false,
219.         "pointradius": 5,
220.         "points": false,
221.         "renderer": "flot",
222.         "seriesOverrides": [],
223.         "spaceLength": 10,
224.         "stack": false,
225.         "steppedLine": false,
226.         "targets": [
227.             {
228.                 "expr": "cfs_metanode_OpMetaCreateDentry{instance=~\"$instance\"}",
229.                 "format": "time_series",
230.                 "intervalFactor": 1,
231.                 "legendFormat": "Create",
232.                 "refId": "A"
233.             },
234.             {
235.                 "expr": "cfs_metanode_OpMetaDeleteDentry{instance=~\"$instance\"}",
236.                 "format": "time_series",
237.                 "intervalFactor": 1,
238.                 "legendFormat": "Delete",
```

```
239.         "refId": "B"
240.     },
241.     {
242.         "expr": "cfs_metanode_OpMetaUpdateDentry{instance=~\"$instance\"}",
243.         "format": "time_series",
244.         "intervalFactor": 1,
245.         "legendFormat": "Update",
246.         "refId": "C"
247.     }
248. ],
249. "thresholds": [],
250. "timeFrom": null,
251. "timeShift": null,
252. "title": "metanode_OpMetaDentry",
253. "tooltip": {
254.     "shared": true,
255.     "sort": 0,
256.     "value_type": "individual"
257. },
258. "type": "graph",
259. "xaxis": {
260.     "buckets": null,
261.     "mode": "time",
262.     "name": null,
263.     "show": true,
264.     "values": []
265. },
266. "yaxes": [
267.     {
268.         "format": "ns",
269.         "label": null,
270.         "logBase": 1,
271.         "max": null,
272.         "min": null,
273.         "show": true
274.     },
275.     {
276.         "format": "short",
277.         "label": null,
278.         "logBase": 1,
279.         "max": null,
280.         "min": null,
281.         "show": true
282.     }
283. ],
284. "yaxis": {
285.     "align": false,
286.     "alignLevel": null
```

```
287.     }
288.   },
289.   {
290.     "aliasColors": {},
291.     "bars": false,
292.     "dashLength": 10,
293.     "dashes": false,
294.     "datasource": "${DS_ChubaoFS01}",
295.     "fill": 1,
296.     "gridPos": {
297.       "h": 8,
298.       "w": 8,
299.       "x": 0,
300.       "y": 42
301.     },
302.     "id": 46,
303.     "legend": {
304.       "avg": false,
305.       "current": false,
306.       "max": false,
307.       "min": false,
308.       "show": true,
309.       "total": false,
310.       "values": false
311.     },
312.     "lines": true,
313.     "linewidth": 1,
314.     "links": [],
315.     "nullPointMode": "null",
316.     "percentage": false,
317.     "pointradius": 5,
318.     "points": false,
319.     "renderer": "flot",
320.     "seriesOverrides": [],
321.     "spaceLength": 10,
322.     "stack": false,
323.     "steppedLine": false,
324.     "targets": [
325.       {
326.         "expr": "cfs_metanode_OpMetaOpen{instance=~\"$instance\"}",
327.         "format": "time_series",
328.         "intervalFactor": 1,
329.         "legendFormat": "Open",
330.         "refId": "A"
331.       },
332.       {
333.         "expr": "cfs_metanode_OpMetaLookup{instance=~\"$instance\"}",
334.         "format": "time_series",
```

```
335.         "intervalFactor": 1,
336.         "legendFormat": "Lookup",
337.         "refId": "B"
338.     },
339.     {
340.         "expr": "cfs_metanode_OpMetaNodeHeartbeat{instance=~\"$instance\"}",
341.         "format": "time_series",
342.         "intervalFactor": 1,
343.         "legendFormat": "NodeHeartbeat",
344.         "refId": "C"
345.     },
346.     {
347.         "expr": "cfs_metanode_OpMetaReadDir{instance=~\"$instance\"}",
348.         "format": "time_series",
349.         "intervalFactor": 1,
350.         "legendFormat": "ReadDir",
351.         "refId": "D"
352.     },
353.     {
354.         "expr": "cfs_metanode_OpMetaReleaseOpen{instance=~\"$instance\"}",
355.         "format": "time_series",
356.         "intervalFactor": 1,
357.         "legendFormat": "ReleaseOpen",
358.         "refId": "E"
359.     },
360.     {
361.         "expr": "cfs_metanode_OpMetaSetattr{instance=~\"$instance\"}",
362.         "format": "time_series",
363.         "intervalFactor": 1,
364.         "legendFormat": "Setattr",
365.         "refId": "F"
366.     },
367.     {
368.         "expr": "cfs_metanode_OpMetaTruncate{instance=~\"$instance\"}",
369.         "format": "time_series",
370.         "intervalFactor": 1,
371.         "legendFormat": "Truncate",
372.         "refId": "G"
373.     }
374. ],
375. "thresholds": [],
376. "timeFrom": null,
377. "timeShift": null,
378. "title": "metanode_OpMeta",
379. "tooltip": {
380.     "shared": true,
381.     "sort": 0,
382.     "value_type": "individual"
```

```
383.     },
384.     "type": "graph",
385.     "xaxis": {
386.         "buckets": null,
387.         "mode": "time",
388.         "name": null,
389.         "show": true,
390.         "values": []
391.     },
392.     "yaxes": [
393.         {
394.             "format": "ns",
395.             "label": null,
396.             "logBase": 1,
397.             "max": null,
398.             "min": null,
399.             "show": true
400.         },
401.         {
402.             "format": "short",
403.             "label": null,
404.             "logBase": 1,
405.             "max": null,
406.             "min": null,
407.             "show": true
408.         }
409.     ],
410.     "yaxis": {
411.         "align": false,
412.         "alignLevel": null
413.     }
414. },
415. {
416.     "aliasColors": {},
417.     "bars": false,
418.     "dashLength": 10,
419.     "dashes": false,
420.     "datasource": "${DS_ChubaoFS01}",
421.     "fill": 1,
422.     "gridPos": {
423.         "h": 8,
424.         "w": 8,
425.         "x": 8,
426.         "y": 42
427.     },
428.     "id": 50,
429.     "legend": {
430.         "avg": false,
```

```
431.         "current": false,
432.         "max": false,
433.         "min": false,
434.         "show": true,
435.         "total": false,
436.         "values": false
437.     },
438.     "lines": true,
439.     "linewidth": 1,
440.     "links": [],
441.     "nullPointMode": "null",
442.     "percentage": false,
443.     "pointradius": 5,
444.     "points": false,
445.     "renderer": "flot",
446.     "seriesOverrides": [],
447.     "spaceLength": 10,
448.     "stack": false,
449.     "steppedLine": false,
450.     "targets": [
451.         {
452.             "expr": "cfs_metanode_OpMetaExtentsAdd{instance=~\"$instance\"}",
453.             "format": "time_series",
454.             "intervalFactor": 1,
455.             "legendFormat": "Add",
456.             "refId": "A"
457.         },
458.         {
459.             "expr": "cfs_metanode_OpMetaExtentsList{instance=~\"$instance\"}",
460.             "format": "time_series",
461.             "intervalFactor": 1,
462.             "legendFormat": "List",
463.             "refId": "B"
464.         }
465.     ],
466.     "thresholds": [],
467.     "timeFrom": null,
468.     "timeShift": null,
469.     "title": "metanode_OpMetaExtents",
470.     "tooltip": {
471.         "shared": true,
472.         "sort": 0,
473.         "value_type": "individual"
474.     },
475.     "type": "graph",
476.     "xaxis": {
477.         "buckets": null,
478.         "mode": "time",
```



```
479.         "name": null,
480.         "show": true,
481.         "values": []
482.     },
483.     "yaxes": [
484.         {
485.             "format": "ns",
486.             "label": null,
487.             "logBase": 1,
488.             "max": null,
489.             "min": null,
490.             "show": true
491.         },
492.         {
493.             "format": "short",
494.             "label": null,
495.             "logBase": 1,
496.             "max": null,
497.             "min": null,
498.             "show": true
499.         }
500.     ],
501.     "yaxis": {
502.         "align": false,
503.         "alignLevel": null
504.     }
505. },
506. {
507.     "collapsed": false,
508.     "gridPos": {
509.         "h": 1,
510.         "w": 24,
511.         "x": 0,
512.         "y": 50
513.     },
514.     "id": 27,
515.     "panels": [],
516.     "title": "Datanode",
517.     "type": "row"
518. },
519. {
520.     "aliasColors": {},
521.     "bars": false,
522.     "dashLength": 10,
523.     "dashes": false,
524.     "datasource": "${DS_ChubaoFS01}",
525.     "fill": 1,
526.     "gridPos": {
```

```
527.         "h": 8,
528.         "w": 8,
529.         "x": 0,
530.         "y": 51
531.     },
532.     "id": 28,
533.     "legend": {
534.         "avg": false,
535.         "current": false,
536.         "max": false,
537.         "min": false,
538.         "show": true,
539.         "total": false,
540.         "values": false
541.     },
542.     "lines": true,
543.     "linewidth": 1,
544.     "links": [],
545.     "nullPointMode": "null",
546.     "percentage": false,
547.     "pointradius": 5,
548.     "points": false,
549.     "renderer": "flot",
550.     "seriesOverrides": [],
551.     "spaceLength": 10,
552.     "stack": false,
553.     "steppedLine": false,
554.     "targets": [
555.         {
556.             "expr":
"cf_dataNode_[[cluster]]_datanode_CreateFile{instance=~\"$instance\"}",
557.             "format": "time_series",
558.             "intervalFactor": 1,
559.             "legendFormat": "CreateFile",
560.             "refId": "A"
561.         },
562.         {
563.             "expr":
"cf_dataNode_[[cluster]]_datanode_MarkDelete{instance=~\"$instance\"}",
564.             "format": "time_series",
565.             "intervalFactor": 1,
566.             "legendFormat": "MarkDelete",
567.             "refId": "B"
568.         }
569.     ],
570.     "thresholds": [],
571.     "timeFrom": null,
572.     "timeShift": null,
```

```
573.     "title": "datanode_CreateFile",
574.     "tooltip": {
575.         "shared": true,
576.         "sort": 0,
577.         "value_type": "individual"
578.     },
579.     "type": "graph",
580.     "xaxis": {
581.         "buckets": null,
582.         "mode": "time",
583.         "name": null,
584.         "show": true,
585.         "values": []
586.     },
587.     "yaxes": [
588.         {
589.             "format": "ns",
590.             "label": null,
591.             "logBase": 1,
592.             "max": null,
593.             "min": null,
594.             "show": true
595.         },
596.         {
597.             "format": "short",
598.             "label": null,
599.             "logBase": 1,
600.             "max": null,
601.             "min": null,
602.             "show": true
603.         }
604.     ],
605.     "yaxis": {
606.         "align": false,
607.         "alignLevel": null
608.     }
609. },
610. {
611.     "aliasColors": {},
612.     "bars": false,
613.     "dashLength": 10,
614.     "dashes": false,
615.     "datasource": "${DS_ChubaoFS01}",
616.     "fill": 1,
617.     "gridPos": {
618.         "h": 8,
619.         "w": 8,
620.         "x": 8,
```

```

621.         "y": 51
622.     },
623.     "id": 74,
624.     "legend": {
625.         "avg": false,
626.         "current": false,
627.         "max": false,
628.         "min": false,
629.         "show": true,
630.         "total": false,
631.         "values": false
632.     },
633.     "lines": true,
634.     "linewidth": 1,
635.     "links": [],
636.     "nullPointMode": "null",
637.     "percentage": false,
638.     "pointradius": 5,
639.     "points": false,
640.     "renderer": "flot",
641.     "seriesOverrides": [],
642.     "spaceLength": 10,
643.     "stack": false,
644.     "steppedLine": false,
645.     "targets": [
646.         {
647.             "expr":
648. "cfs_dataNode_[[cluster]]_datanode_ExtentRepairRead{instance=~\"$instance\"}",
649.             "format": "time_series",
650.             "intervalFactor": 1,
651.             "legendFormat": "ExtentRepairRead",
652.             "refId": "B"
653.         },
654.         {
655.             "expr":
656. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}",
657.             "format": "time_series",
658.             "intervalFactor": 1,
659.             "legendFormat": "GetAllExtentWatermark",
660.             "refId": "C"
661.         },
662.         {
663.             "expr":
664. "cfs_dataNode_[[cluster]]_datanode_NotifyExtentRepair{instance=~\"$instance\"}"
665.         },
666.         {
667.             "expr":
668. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
669.         },
670.         {
671.             "expr":
672. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
673.         },
674.         {
675.             "expr":
676. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
677.         },
678.         {
679.             "expr":
680. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
681.         },
682.         {
683.             "expr":
684. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
685.         },
686.         {
687.             "expr":
688. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
689.         },
690.         {
691.             "expr":
692. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
693.         },
694.         {
695.             "expr":
696. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
697.         },
698.         {
699.             "expr":
700. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
701.         },
702.         {
703.             "expr":
704. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
705.         },
706.         {
707.             "expr":
708. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
709.         },
710.         {
711.             "expr":
712. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
713.         },
714.         {
715.             "expr":
716. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
717.         },
718.         {
719.             "expr":
720. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
721.         },
722.         {
723.             "expr":
724. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
725.         },
726.         {
727.             "expr":
728. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
729.         },
730.         {
731.             "expr":
732. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
733.         },
734.         {
735.             "expr":
736. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
737.         },
738.         {
739.             "expr":
740. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
741.         },
742.         {
743.             "expr":
744. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
745.         },
746.         {
747.             "expr":
748. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
749.         },
750.         {
751.             "expr":
752. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
753.         },
754.         {
755.             "expr":
756. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
757.         },
758.         {
759.             "expr":
760. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
761.         },
762.         {
763.             "expr":
764. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
765.         },
766.         {
767.             "expr":
768. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
769.         },
770.         {
771.             "expr":
772. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
773.         },
774.         {
775.             "expr":
776. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
777.         },
778.         {
779.             "expr":
780. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
781.         },
782.         {
783.             "expr":
784. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
785.         },
786.         {
787.             "expr":
788. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
789.         },
790.         {
791.             "expr":
792. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
793.         },
794.         {
795.             "expr":
796. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
797.         },
798.         {
799.             "expr":
800. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
801.         },
802.         {
803.             "expr":
804. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
805.         },
806.         {
807.             "expr":
808. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
809.         },
810.         {
811.             "expr":
812. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
813.         },
814.         {
815.             "expr":
816. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
817.         },
818.         {
819.             "expr":
820. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
821.         },
822.         {
823.             "expr":
824. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
825.         },
826.         {
827.             "expr":
828. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
829.         },
830.         {
831.             "expr":
832. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
833.         },
834.         {
835.             "expr":
836. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
837.         },
838.         {
839.             "expr":
840. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
841.         },
842.         {
843.             "expr":
844. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
845.         },
846.         {
847.             "expr":
848. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
849.         },
850.         {
851.             "expr":
852. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
853.         },
854.         {
855.             "expr":
856. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
857.         },
858.         {
859.             "expr":
860. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
861.         },
862.         {
863.             "expr":
864. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
865.         },
866.         {
867.             "expr":
868. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
869.         },
870.         {
871.             "expr":
872. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
873.         },
874.         {
875.             "expr":
876. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
877.         },
878.         {
879.             "expr":
880. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
881.         },
882.         {
883.             "expr":
884. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
885.         },
886.         {
887.             "expr":
888. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
889.         },
890.         {
891.             "expr":
892. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
893.         },
894.         {
895.             "expr":
896. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
897.         },
898.         {
899.             "expr":
900. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
901.         },
902.         {
903.             "expr":
904. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
905.         },
906.         {
907.             "expr":
908. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
909.         },
910.         {
911.             "expr":
912. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
913.         },
914.         {
915.             "expr":
916. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
917.         },
918.         {
919.             "expr":
920. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
921.         },
922.         {
923.             "expr":
924. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentRepair{instance=~\"$instance\"}"
925.         },
926.         {
927.             "expr":
928. "cfs_dataNode_[[cluster]]_datanode_GetAllExtentWatermark{instance=~\"$instance\"}"
929.         },
930.         {
931.             "
```

```
664.         "legendFormat": "NotifyExtentRepair",
665.         "refId": "D"
666.     }
667. ],
668.     "thresholds": [],
669.     "timeFrom": null,
670.     "timeShift": null,
671.     "title": "datanode_Extent",
672.     "tooltip": {
673.         "shared": true,
674.         "sort": 0,
675.         "value_type": "individual"
676.     },
677.     "type": "graph",
678.     "xaxis": {
679.         "buckets": null,
680.         "mode": "time",
681.         "name": null,
682.         "show": true,
683.         "values": []
684.     },
685.     "yaxes": [
686.         {
687.             "format": "ns",
688.             "label": null,
689.             "logBase": 1,
690.             "max": null,
691.             "min": null,
692.             "show": true
693.         },
694.         {
695.             "format": "short",
696.             "label": null,
697.             "logBase": 1,
698.             "max": null,
699.             "min": null,
700.             "show": true
701.         }
702.     ],
703.     "yaxis": {
704.         "align": false,
705.         "alignLevel": null
706.     }
707. },
708. {
709.     "aliasColors": {},
710.     "bars": false,
711.     "dashLength": 10,
```

```
712.     "dashes": false,
713.     "datasource": "${DS_ChubaoFS01}",
714.     "fill": 1,
715.     "gridPos": {
716.         "h": 8,
717.         "w": 8,
718.         "x": 16,
719.         "y": 51
720.     },
721.     "id": 81,
722.     "legend": {
723.         "avg": false,
724.         "current": false,
725.         "max": false,
726.         "min": false,
727.         "show": true,
728.         "total": false,
729.         "values": false
730.     },
731.     "lines": true,
732.     "linewidth": 1,
733.     "links": [],
734.     "nullPointMode": "null",
735.     "percentage": false,
736.     "pointradius": 5,
737.     "points": false,
738.     "renderer": "flot",
739.     "seriesOverrides": [],
740.     "spaceLength": 10,
741.     "stack": false,
742.     "steppedLine": false,
743.     "targets": [
744.         {
745.             "expr":
"cfss_dataNode_{{cluster}}_datanode_OpLoadDataPartition{instance=~\"$instance\"}"
,
746.             "format": "time_series",
747.             "intervalFactor": 1,
748.             "legendFormat": "OpLoadDataPartition",
749.             "refId": "G"
750.         },
751.         {
752.             "expr":
"cfss_dataNode_{{cluster}}_datanode_OpDataNodeHeartbeat{instance=~\"$instance\"}"
,
753.             "format": "time_series",
754.             "intervalFactor": 1,
755.             "legendFormat": "OpDataNodeHeartbeat",
```

```

756.         "refId": "F"
757.     },
758.     {
759.         "expr":
760.         "cfs_dataNode_[[cluster]]_datanode_OpGetPartitionSize{instance=~\"$instance\"}"
761.         ,
762.         "format": "time_series",
763.         "intervalFactor": 1,
764.         "legendFormat": "OpGetPartitionSize",
765.         "refId": "H"
766.     },
767.     {
768.         "expr":
769.         "cfs_dataNode_[[cluster]]_datanode_OpGetAppliedId{instance=~\"$instance\"}",
770.         "format": "time_series",
771.         "intervalFactor": 1,
772.         "legendFormat": "OpGetAppliedId",
773.         "refId": "I"
774.     },
775.     {
776.         "expr":
777.         "cfs_dataNode_[[cluster]]_datanode_OpCreateDataPartition{instance=~\"$instance\"}"
778.         "}",
779.         "format": "time_series",
780.         "intervalFactor": 1,
781.         "legendFormat": "OpCreateDataPartition",
782.         "refId": "A"
783.     }
784. ],
785. "thresholds": [],
786. "timeFrom": null,
787. "timeShift": null,
788. "title": "datanode_Op",
789. "tooltip": {
790.     "shared": true,
791.     "sort": 0,
792.     "value_type": "individual"
793. },
794. "type": "graph",
795. "xaxis": {
796.     "buckets": null,
797.     "mode": "time",
798.     "name": null,
799.     "show": true,
800.     "values": []
801. },
802. "yaxes": [
803.     {

```

```
799.         "format": "ns",
800.         "label": null,
801.         "logBase": 1,
802.         "max": null,
803.         "min": null,
804.         "show": true
805.     },
806.     {
807.         "format": "short",
808.         "label": null,
```



```
1.         "logBase": 1,
2.         "max": null,
3.         "min": null,
4.         "show": true
5.     }
6. ],
7.     "yaxis": {
8.         "align": false,
9.         "alignLevel": null
10.    }
11. },
12. {
13.     "aliasColors": {},
14.     "bars": false,
15.     "dashLength": 10,
16.     "dashes": false,
17.     "datasource": "${DS_ChubaoFS01}",
18.     "fill": 1,
19.     "gridPos": {
20.         "h": 8,
21.         "w": 8,
22.         "x": 0,
23.         "y": 59
24.     },
25.     "id": 76,
26.     "legend": {
27.         "avg": false,
28.         "current": false,
29.         "max": false,
30.         "min": false,
31.         "show": true,
32.         "total": false,
33.         "values": false
34.     },
35.     "lines": true,
36.     "linewidth": 1,
37.     "links": [],
38.     "nullPointMode": "null",
39.     "percentage": false,
40.     "pointradius": 5,
41.     "points": false,
42.     "renderer": "flot",
43.     "seriesOverrides": [],
44.     "spaceLength": 10,
45.     "stack": false,
46.     "steppedLine": false,
47.     "targets": [
```

```
48.     {
49.         "expr":
"cfss_dataNode_[[cluster]]_datanode_Read{instance=~\"$instance\"}",
50.         "format": "time_series",
51.         "intervalFactor": 1,
52.         "legendFormat": "Read",
53.         "refId": "J"
54.     },
55.     {
56.         "expr":
"cfss_dataNode_[[cluster]]_datanode_Write{instance=~\"$instance\"}",
57.         "format": "time_series",
58.         "intervalFactor": 1,
59.         "legendFormat": "Write",
60.         "refId": "K"
61.     },
62.     {
63.         "expr":
"cfss_dataNode_[[cluster]]_datanode_RandomWrite{instance=~\"$instance\"}",
64.         "format": "time_series",
65.         "intervalFactor": 1,
66.         "legendFormat": "RandomWrite",
67.         "refId": "L"
68.     }
69. ],
70. "thresholds": [],
71. "timeFrom": null,
72. "timeShift": null,
73. "title": "datanode_IO",
74. "tooltip": {
75.     "shared": true,
76.     "sort": 0,
77.     "value_type": "individual"
78. },
79. "type": "graph",
80. "xaxis": {
81.     "buckets": null,
82.     "mode": "time",
83.     "name": null,
84.     "show": true,
85.     "values": []
86. },
87. "yaxes": [
88.     {
89.         "format": "ns",
90.         "label": null,
91.         "logBase": 1,
92.         "max": null,
```

```
93.         "min": null,
94.         "show": true
95.     },
96.     {
97.         "format": "short",
98.         "label": null,
99.         "logBase": 1,
100.        "max": null,
101.        "min": null,
102.        "show": true
103.    }
104. ],
105. "yaxis": {
106.     "align": false,
107.     "alignLevel": null
108. }
109. },
110. {
111.     "aliasColors": {},
112.     "bars": false,
113.     "dashLength": 10,
114.     "dashes": false,
115.     "datasource": "${DS_ChubaoFS01}",
116.     "fill": 1,
117.     "gridPos": {
118.         "h": 8,
119.         "w": 8,
120.         "x": 8,
121.         "y": 59
122.     },
123.     "id": 77,
124.     "legend": {
125.         "avg": false,
126.         "current": false,
127.         "max": false,
128.         "min": false,
129.         "show": true,
130.         "total": false,
131.         "values": false
132.     },
133.     "lines": true,
134.     "linewidth": 1,
135.     "links": [],
136.     "nullPointMode": "null",
137.     "percentage": false,
138.     "pointradius": 5,
139.     "points": false,
140.     "renderer": "flot",
```

```

141.     "seriesOverrides": [],
142.     "spaceLength": 10,
143.     "stack": false,
144.     "steppedLine": false,
145.     "targets": [
146.         {
147.             "expr":
148. "cfs_dataNode_{{cluster}}_datanode_streamRead{instance=~\"$instance\"}",
149.             "format": "time_series",
150.             "intervalFactor": 1,
151.             "legendFormat": "streamRead",
152.             "refId": "M"
153.         },
154.         {
155.             "expr":
156. "cfs_dataNode_{{cluster}}_datanode_streamWrite{instance=~\"$instance\"}",
157.             "format": "time_series",
158.             "intervalFactor": 1,
159.             "legendFormat": "streamWrite",
160.             "refId": "N"
161.         },
162.         {
163.             "expr":
164. "cfs_dataNode_{{cluster}}_datanode_streamCreateFile{instance=~\"$instance\"}",
165.             "format": "time_series",
166.             "intervalFactor": 1,
167.             "legendFormat": "streamCreateFile",
168.             "refId": "A"
169.         },
170.         {
171.             "expr":
172. "cfs_dataNode_{{cluster}}_datanode_streamExtentRepairRead{instance=~\"$instance\"}",
173.             "format": "time_series",
174.             "intervalFactor": 1,
175.             "legendFormat": "streamExtentRepairRead",
176.             "refId": "B"
177.         },
178.         {
179.             "expr":
180. "cfs_dataNode_{{cluster}}_datanode_streamMarkDelete{instance=~\"$instance\"}",
181.             "format": "time_series",
182.             "intervalFactor": 1,
183.             "legendFormat": "streamMarkDelete",
184.             "refId": "C"
185.         }
186.     ]
187. }

```

```

    "cfs_dataNode_[[cluster]]_datanode_streamOpGetAppliedId{instance=~\"$instance\"
  }",
183.         "format": "time_series",
184.         "intervalFactor": 1,
185.         "legendFormat": "streamOpGetAppliedId",
186.         "refId": "D"
187.     },
188.     {
189.         "expr":
    "cfs_dataNode_[[cluster]]_datanode_streamOpGetPartitionSize{instance=~\"$instan
ce\"}",
190.         "format": "time_series",
191.         "intervalFactor": 1,
192.         "legendFormat": "streamOpGetPartitionSize",
193.         "refId": "E"
194.     },
195.     {
196.         "expr":
    "cfs_dataNode_[[cluster]]_datanode_streamNotifyExtentRepair{instance=~\"$instan
ce\"}",
197.         "format": "time_series",
198.         "intervalFactor": 1,
199.         "legendFormat": "streamNotifyExtentRepair",
200.         "refId": "F"
201.     }
202. ],
203. "thresholds": [],
204. "timeFrom": null,
205. "timeShift": null,
206. "title": "datanode_Stream",
207. "tooltip": {
208.     "shared": true,
209.     "sort": 0,
210.     "value_type": "individual"
211. },
212. "type": "graph",
213. "xaxis": {
214.     "buckets": null,
215.     "mode": "time",
216.     "name": null,
217.     "show": true,
218.     "values": []
219. },
220. "yaxes": [
221.     {
222.         "format": "ns",
223.         "label": null,
224.         "logBase": 1,

```

```
225.         "max": null,
226.         "min": null,
227.         "show": true
228.     },
229.     {
230.         "format": "short",
231.         "label": null,
232.         "logBase": 1,
233.         "max": null,
234.         "min": null,
235.         "show": true
236.     }
237. ],
238. "yaxis": {
239.     "align": false,
240.     "alignLevel": null
241. }
242. },
243. {
244.     "collapsed": true,
245.     "gridPos": {
246.         "h": 1,
247.         "w": 24,
248.         "x": 0,
249.         "y": 67
250.     },
251.     "id": 66,
252.     "panels": [
253.         {
254.             "aliasColors": {},
255.             "bars": false,
256.             "dashLength": 10,
257.             "dashes": false,
258.             "datasource": "${DS_ChubaoFS01}",
259.             "fill": 1,
260.             "gridPos": {
261.                 "h": 6,
262.                 "w": 7,
263.                 "x": 0,
264.                 "y": 85
265.             },
266.             "id": 64,
267.             "legend": {
268.                 "avg": false,
269.                 "current": false,
270.                 "max": false,
271.                 "min": false,
272.                 "show": true,
```

```

273.         "total": false,
274.         "values": false
275.     },
276.     "lines": true,
277.     "linewidth": 1,
278.     "links": [],
279.     "nullPointMode": "null",
280.     "percentage": false,
281.     "pointradius": 5,
282.     "points": false,
283.     "renderer": "flot",
284.     "seriesOverrides": [],
285.     "spaceLength": 10,
286.     "stack": false,
287.     "steppedLine": false,
288.     "targets": [
289.     {
290.         "expr": "cfs_fuseclient_OpMetaOpen{instance=~\"$instance\"}",
291.         "format": "time_series",
292.         "intervalFactor": 1,
293.         "legendFormat": "Open",
294.         "refId": "B"
295.     },
296.     {
297.         "expr":
298.         "cfs_fuseclient_OpMetaExtentsAdd{instance=~\"$instance\"}",
299.         "format": "time_series",
300.         "intervalFactor": 1,
301.         "legendFormat": "ExtentsAdd",
302.         "refId": "H"
303.     },
304.     {
305.         "expr":
306.         "cfs_fuseclient_OpMetaExtentsList{instance=~\"$instance\"}",
307.         "format": "time_series",
308.         "intervalFactor": 1,
309.         "legendFormat": "ExtentsList",
310.         "refId": "I"
311.     },
312.     {
313.         "expr": "cfs_fuseclient_OpMetaReadDir{instance=~\"$instance\"}",
314.         "format": "time_series",
315.         "intervalFactor": 1,
316.         "legendFormat": "ReadDir",
317.         "refId": "K"
318.     },
319.     {
320.         "expr": "cfs_fuseclient_OpMetaSetattr{instance=~\"$instance\"}",

```

```
319.         "format": "time_series",
320.         "intervalFactor": 1,
321.         "legendFormat": "Setattr",
322.         "refId": "L"
323.     }
324. ],
325. "thresholds": [],
326. "timeFrom": null,
327. "timeShift": null,
328. "title": "fuseclient_OpMeta",
329. "tooltip": {
330.     "shared": true,
331.     "sort": 0,
332.     "value_type": "individual"
333. },
334. "type": "graph",
335. "xaxis": {
336.     "buckets": null,
337.     "mode": "time",
338.     "name": null,
339.     "show": true,
340.     "values": []
341. },
342. "yaxes": [
343.     {
344.         "format": "ns",
345.         "label": null,
346.         "logBase": 1,
347.         "max": null,
348.         "min": null,
349.         "show": true
350.     },
351.     {
352.         "format": "short",
353.         "label": null,
354.         "logBase": 1,
355.         "max": null,
356.         "min": null,
357.         "show": true
358.     }
359. ],
360. "yaxis": {
361.     "align": false,
362.     "alignLevel": null
363. }
364. },
365. {
366.     "aliasColors": {},
```



```
367.         "bars": false,
368.         "dashLength": 10,
369.         "dashes": false,
370.         "datasource": "${DS_ChubaoFS01}",
371.         "fill": 1,
372.         "gridPos": {
373.             "h": 6,
374.             "w": 7,
375.             "x": 7,
376.             "y": 85
377.         },
378.         "id": 67,
379.         "legend": {
380.             "avg": false,
381.             "current": false,
382.             "max": false,
383.             "min": false,
384.             "show": true,
385.             "total": false,
386.             "values": false
387.         },
388.         "lines": true,
389.         "linewidth": 1,
390.         "links": [],
391.         "nullPointMode": "null",
392.         "percentage": false,
393.         "pointradius": 5,
394.         "points": false,
395.         "renderer": "flot",
396.         "seriesOverrides": [],
397.         "spaceLength": 10,
398.         "stack": false,
399.         "steppedLine": false,
400.         "targets": [
401.             {
402.                 "expr":
403.                 "cfs_fuseclient_OpMetaBatchInodeGet{instance=~\"$instance\"}",
404.                 "format": "time_series",
405.                 "intervalFactor": 1,
406.                 "legendFormat": "BatchInodeGet",
407.                 "refId": "A"
408.             },
409.             {
410.                 "expr":
411.                 "cfs_fuseclient_OpMetaCreateInode{instance=~\"$instance\"}",
412.                 "format": "time_series",
413.                 "intervalFactor": 1,
414.                 "legendFormat": "CreateInode",
```

```
413.         "refId": "D"
414.     },
415.     {
416.         "expr":
417.         "cfs_fuseclient_OpMetaDeleteInode{instance=~\"$instance\"}",
418.         "format": "time_series",
419.         "intervalFactor": 1,
420.         "legendFormat": "DeleteInode",
421.         "refId": "F"
422.     },
423.     {
424.         "expr":
425.         "cfs_fuseclient_OpMetaEvictInode{instance=~\"$instance\"}",
426.         "format": "time_series",
427.         "intervalFactor": 1,
428.         "legendFormat": "EvictInode",
429.         "refId": "G"
430.     },
431.     {
432.         "expr": "cfs_fuseclient_OpMetaInodeGet{instance=~\"$instance\"}",
433.         "format": "time_series",
434.         "intervalFactor": 1,
435.         "legendFormat": "InodeGet",
436.         "refId": "J"
437.     }
438. ],
439. "thresholds": [],
440. "timeFrom": null,
441. "timeShift": null,
442. "title": "fuseclient_OpMetaInode",
443. "tooltip": {
444.     "shared": true,
445.     "sort": 0,
446.     "value_type": "individual"
447. },
448. "type": "graph",
449. "xaxis": {
450.     "buckets": null,
451.     "mode": "time",
452.     "name": null,
453.     "show": true,
454.     "values": []
455. },
456. "yaxes": [
457.     {
458.         "format": "ns",
459.         "label": null,
460.         "logBase": 1,
```

```
459.         "max": null,
460.         "min": null,
461.         "show": true
462.     },
463.     {
464.         "format": "short",
465.         "label": null,
466.         "logBase": 1,
467.         "max": null,
468.         "min": null,
469.         "show": true
470.     }
471. ],
472. "yaxis": {
473.     "align": false,
474.     "alignLevel": null
475. }
476. },
477. {
478.     "aliasColors": {},
479.     "bars": false,
480.     "dashLength": 10,
481.     "dashes": false,
482.     "datasource": "${DS_ChubaoFS01}",
483.     "fill": 1,
484.     "gridPos": {
485.         "h": 6,
486.         "w": 7,
487.         "x": 14,
488.         "y": 85
489.     },
490.     "id": 68,
491.     "legend": {
492.         "avg": false,
493.         "current": false,
494.         "max": false,
495.         "min": false,
496.         "show": true,
497.         "total": false,
498.         "values": false
499.     },
500.     "lines": true,
501.     "linewidth": 1,
502.     "links": [],
503.     "nullPointMode": "null",
504.     "percentage": false,
505.     "pointradius": 5,
506.     "points": false,
```

```
507.         "renderer": "flot",
508.         "seriesOverrides": [],
509.         "spaceLength": 10,
510.         "stack": false,
511.         "steppedLine": false,
512.         "targets": [
513.             {
514.                 "expr":
515.                 "cfs_fuseclient_OpMetaCreateDentry{instance=~\"$instance\"}",
516.                 "format": "time_series",
517.                 "intervalFactor": 1,
518.                 "legendFormat": "Create",
519.                 "refId": "C"
520.             },
521.             {
522.                 "expr":
523.                 "cfs_fuseclient_OpMetaDeleteDentry{instance=~\"$instance\"}",
524.                 "format": "time_series",
525.                 "intervalFactor": 1,
526.                 "legendFormat": "Delete",
527.                 "refId": "E"
528.             }
529.         ],
530.         "thresholds": [],
531.         "timeFrom": null,
532.         "timeShift": null,
533.         "title": "fuseclient_OpMetaDentry",
534.         "tooltip": {
535.             "shared": true,
536.             "sort": 0,
537.             "value_type": "individual"
538.         },
539.         "type": "graph",
540.         "xaxis": {
541.             "buckets": null,
542.             "mode": "time",
543.             "name": null,
544.             "show": true,
545.             "values": []
546.         },
547.         "yaxes": [
548.             {
549.                 "format": "ns",
550.                 "label": null,
551.                 "logBase": 1,
552.                 "max": null,
```

```
553.         },
554.         {
555.             "format": "short",
556.             "label": null,
557.             "logBase": 1,
558.             "max": null,
559.             "min": null,
560.             "show": true
561.         }
562.     ],
563.     "yaxis": {
564.         "align": false,
565.         "alignLevel": null
566.     }
567. },
568. {
569.     "aliasColors": {},
570.     "bars": false,
571.     "dashLength": 10,
572.     "dashes": false,
573.     "datasource": "${DS_ChubaoFS01}",
574.     "fill": 1,
575.     "gridPos": {
576.         "h": 6,
577.         "w": 7,
578.         "x": 0,
579.         "y": 91
580.     },
581.     "id": 69,
582.     "legend": {
583.         "avg": false,
584.         "current": false,
585.         "max": false,
586.         "min": false,
587.         "show": true,
588.         "total": false,
589.         "values": false
590.     },
591.     "lines": true,
592.     "linewidth": 1,
593.     "links": [],
594.     "nullPointMode": "null",
595.     "percentage": false,
596.     "pointradius": 5,
597.     "points": false,
598.     "renderer": "flot",
599.     "seriesOverrides": [],
600.     "spaceLength": 10,
```

```
601.         "stack": false,
602.         "steppedLine": false,
603.         "targets": [
604.             {
605.                 "expr":
606.                 "cfs_fuseclient_OpMetaExtentsAdd{instance=~\"$instance\"}",
607.                 "format": "time_series",
608.                 "intervalFactor": 1,
609.                 "legendFormat": "ExtentsAdd",
610.                 "refId": "H"
611.             },
612.             {
613.                 "expr":
614.                 "cfs_fuseclient_OpMetaExtentsList{instance=~\"$instance\"}",
615.                 "format": "time_series",
616.                 "intervalFactor": 1,
617.                 "legendFormat": "ExtentsList",
618.                 "refId": "I"
619.             }
620.         ],
621.         "thresholds": [],
622.         "timeFrom": null,
623.         "timeShift": null,
624.         "title": "fuseclient_OpMetaExtent",
625.         "tooltip": {
626.             "shared": true,
627.             "sort": 0,
628.             "value_type": "individual"
629.         },
630.         "type": "graph",
631.         "xaxis": {
632.             "buckets": null,
633.             "mode": "time",
634.             "name": null,
635.             "show": true,
636.             "values": []
637.         },
638.         "yaxes": [
639.             {
640.                 "format": "ns",
641.                 "label": null,
642.                 "logBase": 1,
643.                 "max": null,
644.                 "min": null,
645.                 "show": true
646.             },
647.             {
648.                 "format": "short",
```

```
647.         "label": null,
648.         "logBase": 1,
649.         "max": null,
650.         "min": null,
651.         "show": true
652.     }
653. ],
654.     "yaxis": {
655.         "align": false,
656.         "alignLevel": null
657.     }
658. }
659. ],
660.     "title": "FuseClient",
661.     "type": "row"
662. }
663. ],
664.     "refresh": false,
665.     "schemaVersion": 16,
666.     "style": "dark",
667.     "tags": [],
668.     "templating": {
669.         "list": [
670.             {
671.                 "allValue": null,
672.                 "current": {
673.                     "selected": true,
674.                     "text": "cfs",
675.                     "value": "cfs"
676.                 },
677.                 "hide": 2,
678.                 "includeAll": false,
679.                 "label": "App",
680.                 "multi": false,
681.                 "name": "app",
682.                 "options": [
683.                     {
684.                         "selected": true,
685.                         "text": "cfs",
686.                         "value": "cfs"
687.                     }
688.                 ],
689.                 "query": "cfs",
690.                 "type": "custom"
691.             },
692.             {
693.                 "allValue": null,
694.                 "current": {},
```

```

695.     "datasource": "${DS_ChubaoFS01}",
696.     "hide": 0,
697.     "includeAll": false,
698.     "label": "Cluster",
699.     "multi": false,
700.     "name": "cluster",
701.     "options": [],
702.     "query": "label_values(go_info{app=~\"$app\"}, cluster)",
703.     "refresh": 1,
704.     "regex": "",
705.     "sort": 0,
706.     "tagValuesQuery": "",
707.     "tags": [],
708.     "tagsQuery": "",
709.     "type": "query",
710.     "useTags": false
711. },
712. {
713.     "allValue": null,
714.     "current": {},
715.     "datasource": "${DS_ChubaoFS01}",
716.     "hide": 0,
717.     "includeAll": false,
718.     "label": "Role",
719.     "multi": false,
720.     "name": "role",
721.     "options": [],
722.     "query": "label_values(go_info{app=~\"$app\", cluster=~\"$cluster\"},
role)",
723.     "refresh": 1,
724.     "regex": "",
725.     "sort": 0,
726.     "tagValuesQuery": "",
727.     "tags": [],
728.     "tagsQuery": "",
729.     "type": "query",
730.     "useTags": false
731. },
732. {
733.     "allValue": null,
734.     "current": {},
735.     "datasource": "${DS_ChubaoFS01}",
736.     "hide": 0,
737.     "includeAll": false,
738.     "label": "Instance",
739.     "multi": false,
740.     "name": "instance",
741.     "options": [],

```



```

742.         "query": "label_values(go_info{app=~\"$app\", role=~\"$role\",
cluster=~\"$cluster\"}, instance)",
743.         "refresh": 1,
744.         "regex": "",
745.         "sort": 0,
746.         "tagValuesQuery": "",
747.         "tags": [],
748.         "tagsQuery": "",
749.         "type": "query",
750.         "useTags": false
751.     },
752.     {
753.         "allValue": null,
754.         "current": {},
755.         "datasource": "${DS_ChubaoFS01}",
756.         "hide": 2,
757.         "includeAll": false,
758.         "label": "Host",
759.         "multi": false,
760.         "name": "hostip",
761.         "options": [],
762.         "query": "label_values(go_info{instance=~\"$instance\",
cluster=~\"$cluster\"}, instance)",
763.         "refresh": 1,
764.         "regex": "/([^\:]+):.*/",
765.         "sort": 0,
766.         "tagValuesQuery": "",
767.         "tags": [],
768.         "tagsQuery": "",
769.         "type": "query",
770.         "useTags": false
771.     }
772. ]
773. },
774. "time": {
775.     "from": "now-1h",
776.     "to": "now"
777. },
778. "timepicker": {
779.     "refresh_intervals": [
780.         "5s",
781.         "10s",
782.         "30s",
783.         "1m",
784.         "5m",
785.         "15m",
786.         "30m",
787.         "1h",

```

```
788.         "2h",
789.         "1d"
790.     ],
791.     "time_options": [
792.         "5m",
793.         "15m",
794.         "1h",
795.         "6h",
796.         "12h",
797.         "24h",
798.         "2d",
799.         "7d",
800.         "30d"
801.     ]
802. },
803.     "timezone": "",
804.     "title": "cfs-cluster",
805.     "uid": "tu_qnbsmk",
806.     "version": 140
807. }
```

优化配置FUSE参数

适当调整内核FUSE参数，能够在顺序写及高并发情况下获得更好的性能。具体可参考如下步骤：

- 获取Linux内核源码

下载对应的Linux内核源码包并且安装源码，源码安装目录为 `~/rpmbuild/BUILD/`

```
1. rpm -i kernel-3.10.0-327.28.3.el7.src.rpm 2>&1 | grep -v exist
2. cd ~/rpmbuild/SPECS
3. rpmbuild -bp --target=$(uname -m) kernel.spec
```

- 优化Linux FUSE内核模块参数

为了达到最优的性能，可以修改内核FUSE的参数 `FUSE_MAX_PAGES_PER_REQ` 和 `FUSE_DEFAULT_MAX_BACKGROUND`，优化后的参考值如下：

```
1. /* fs/fuse/fuse_i.h */
2. #define FUSE_MAX_PAGES_PER_REQ 256
3. /* fs/fuse/inode.c */
4. #define FUSE_DEFAULT_MAX_BACKGROUND 32
```

- 编译对应版本Linux内核模块

```
1. yum install kernel-devel-3.10.0-327.28.3.el7.x86_64
2. cd ~/rpmbuild/BUILD/kernel-3.10.0-327.28.3.el7/linux-3.10.0-327.28.3.el7.x86_64/fs/fuse
3. make -C /lib/modules/$(uname -r)/build M=$PWD
```

- 插入内核模块

```
1. cp fuse.ko /lib/modules/$(uname -r)/kernel/fs/fuse
2. rmmod fuse
3. depmod -a
4. modprobe fuse
```

Docker

在docker目录下，run_docker.sh工具用来方便运行ChubaoFS docker-compose试用集群。

执行下面的命令，可完全重新创建一个最小的ChubaoFS集群。注意的是/data/disk是数据根目录，至少需要30GB大小空闲空间。

```
1. $ docker/run_docker.sh -r -d /data/disk
```

客户端启动成功后，在客户端docker容器中使用 `mount` 命令检查目录挂载状态：

```
1. $ mount | grep chubaofs
```

在浏览器中打开<http://127.0.0.1:3000>，使用`admin/123456`登录，可查看chubaofs的grafana监控指标界面。

或者使用下面的命令分步运行：

```
1. $ docker/run_docker.sh -b
2. $ docker/run_docker.sh -s -d /data/disk
3. $ docker/run_docker.sh -c
4. $ docker/run_docker.sh -m
```

更多命令：

```
1. $ docker/run_docker.sh -h
```

监控的Prometheus和Grafana相关配置位于 `docker/monitor` 目录下。

在Kubernetes中使用Chubaofs

Chubaofs基于Container Storage Interface (CSI) (<https://kubernetes-csi.github.io/docs/>) 接口规范开发了cfscsi插件，以支持在Kubernetes集群中使用云存储。

cfscsi	kubernetes
v0.3.0	v1.12
v1.0.0	v1.15

Kubernetes v1.12

在Kubernetes v1.12集群中使用Chubaofs。

Kubernetes配置要求

为了在kubernetes集群中部署cfscsi插件，kubernetes集群需要满足以下配置。

kube-apiserver启动参数：

1. `--feature-gates=CSIPersistentVolume=true,MountPropagation=true`
2. `--runtime-config=api/all`

kube-controller-manager启动参数：

1. `--feature-gates=CSIPersistentVolume=true`

kubelet启动参数：

1. `--feature-gates=CSIPersistentVolume=true,MountPropagation=true,KubeletPluginsWatcher=true`
2. `--enable-controller-attach-detach=true`

获取插件源码及脚本

1. `$ git clone -b csi-spec-v0.3.0 https://github.com/chubaofs/chubaofs-csi.git`
2. `$ cd chubaofs-csi`

拉取官方CSI镜像

```
1. docker pull quay.io/k8scsi/csi-attacher:v0.3.0
2. docker pull quay.io/k8scsi/driver-registrar:v0.3.0
3. docker pull quay.io/k8scsi/csi-provisioner:v0.3.0
```

获取cfscsi镜像

有两种方式可以实现。

- 从docker.io拉取镜像

```
1. docker pull docker.io/chubaofs/cfscsi:v0.3.0
```

- 根据源码编译镜像

```
1. make cfs-image
```

创建kubeconfig

```
1. kubectl create configmap kubecfg --from-file=pkg/cfs/deploy/kubernetes/kubecfg
```

创建RBAC和StorageClass

```
1. kubectl apply -f pkg/cfs/deploy/dynamic_provision/cfs-rbac.yaml
2. kubectl apply -f pkg/cfs/deploy/dynamic_provision/cfs-sc.yaml
```

部署cfscsi插件

- 方式一：将cfscsi ControllerServer和NodeServer绑定在同一个sidecar容器

修改 `pkg/cfs/deploy/dynamic_provision/sidecar/cfs-sidecar.yaml` 文件，将环境变量 `MASTER_ADDRESS` 设置为Chubaofs的实际Master地址，将 `<NodeServer IP>` 设置为kubernetes集群任意IP（如果被调度到该IP的pod需要动态挂载Chubaofs网盘，则必须为该IP部署cfscsi sidecar容器）。

```
1. kubectl apply -f pkg/cfs/deploy/dynamic_provision/sidecar/cfs-sidecar.yaml
```

- 方式二：将cfscsi插件ControllerServer和NodeServer分别部署为statefulset和daemonset（推荐此种）

修改 `pkg/cfs/deploy/dynamic_provision/independent` 文件夹下 `csi-controller-statefulset.yaml` 和 `csi-node-daemonset.yaml` 文件，将环境变量 `MASTER_ADDRESS` 设置为Chubaofs的实际Master地址，将 `<ControllerServer IP>` 设置为kubernetes集群中任意节点IP。

为Kubernetes集群中的节点添加标签，拥有 `csi-role=controller` 标签的节点为ControllerServer。拥有 `csi-role=node` 标签的节点为NodeServer，也可以删除 `csi-node-daemonset.yaml` 文件中的 `nodeSelector`，这样kubernetes集群所有节点均为NodeServer。

1. `kubectl label nodes <ControllerServer IP> csi-role=controller`
2. `kubectl label nodes <NodeServer IP1> csi-role=node`
3. `kubectl label nodes <NodeServer IP2> csi-role=node`
4. ...

部署：

1. `kubectl apply -f pkg/cfs/deploy/dynamic_provision/independent/csi-controller-statefulset.yaml`
2. `kubectl apply -f pkg/cfs/deploy/dynamic_provision/independent/csi-node-daemonset.yaml`

创建PVC

1. `kubectl apply -f pkg/cfs/deploy/dynamic_provision/cfs-pvc.yaml`

nginx动态挂载Chubaofs示例

1. `docker pull nginx`
2. `kubectl apply -f pkg/cfs/deploy/dynamic_provision/pv-pod.yaml`

Kubernetes v1.15

在Kubernetes v1.15集群中使用Chubaofs。

Kubernetes配置要求

为了在kubernetes集群中部署cfscsi插件，kubernetes集群需要满足以下配置。

kube-apiserver启动参数：

1. `--feature-gates=CSIPersistentVolume=true,MountPropagation=true`
2. `--runtime-config=api/all`

kube-controller-manager启动参数：

1. `--feature-gates=CSIPersistentVolume=true`

kubelet启动参数：

1. `--feature-gates=CSIPersistentVolume=true,MountPropagation=true,KubeletPluginsWatcher=true`
2. `--enable-controller-attach-detach=true`

获取插件源码及脚本

1. `$ git clone https://github.com/chubaofs/chubaofs-csi.git`
2. `$ cd chubaofs-csi`

拉取官方CSI镜像

1. `docker pull quay.io/k8scsi/csi-attacher:v1.0.0`
2. `docker pull quay.io/k8scsi/csi-node-driver-registrar:v1.0.2`
3. `docker pull quay.io/k8scsi/csi-provisioner:v1.0.0`

获取cfscsi镜像

有两种方式可以实现。

- 方式一：从docker.io拉取镜像

1. `docker pull docker.io/chubaofs/cfscsi:v1.0.0`

- 方式二：根据源码编译镜像

1. `make cfs-image`

创建kubeconfig

1. `kubectl create configmap kubecfg --from-file=pkg/chubaofs/deploy/kubernetes/kubecfg`

创建RBAC和StorageClass

1. `kubectl apply -f pkg/chubaofs/deploy/dynamic_provision/cfs-rbac.yaml`
2. `kubectl apply -f pkg/chubaofs/deploy/dynamic_provision/cfs-sc.yaml`

部署cfscsi插件

- 方式一：将cfscsi ControllerServer和NodeServer绑定在同一个sidecar容器

修改 `pkg/chubaofs/deploy/dynamic_provision/sidecar/cfs-sidecar.yaml` 文件，将环境变量 `MASTER_ADDRESS` 设置为Chubaofs的实际Master地址，将 `<NodeServer IP>` 设置为kubernetes集群任意IP（如果被调度到该IP的pod需要动态挂载Chubaofs网盘，则必须为该IP部署cfscsi sidecar容器）。

```
1. kubectl apply -f pkg/chubaofs/deploy/dynamic_provision/sidecar/cfs-sidecar.yaml
```

- 方式二：将cfscsi插件ControllerServer和NodeServer分别部署为statefulset和daemonset（推荐此种）

修改 `pkg/chubaofs/deploy/dynamic_provision/independent` 文件夹下 `csi-controller-statefulset.yaml` 和 `csi-node-daemonset.yaml` 文件，将环境变量 `MASTER_ADDRESS` 设置为Chubaofs的实际Master地址，将 `<ControllerServer IP>` 设置为kubernetes集群中任意节点IP。

为Kubernetes集群中的节点添加标签，拥有 `csi-role=controller` 标签的节点为ControllerServer。拥有 `csi-role=node` 标签的节点为NodeServer，也可以删除 `csi-node-daemonset.yaml` 文件中的 `nodeSelector`，这样kubernetes集群所有节点均为NodeServer。

```
1. kubectl label nodes <ControllerServer IP> csi-role=controller
2. kubectl label nodes <NodeServer IP1> csi-role=node
3. kubectl label nodes <NodeServer IP2> csi-role=node
4. ...
```

部署：

```
1. kubectl apply -f pkg/chubaofs/deploy/dynamic_provision/independent/csi-controller-statefulset.yaml
2. kubectl apply -f pkg/chubaofs/deploy/dynamic_provision/independent/csi-node-daemonset.yaml
```

创建PVC

```
1. kubectl apply -f pkg/chubaofs/deploy/dynamic_provision/cfs-pvc.yaml
```

nginx动态挂载Chubaofs示例

```
1. docker pull nginx
2. kubectl apply -f pkg/chubaofs/deploy/dynamic_provision/pv-pod.yaml
```

- [资源管理节点](#)
- [元数据节点](#)

资源管理节点

Master

- [集群管理命令](#)
 - [概述](#)
 - [冻结集群](#)
- [元数据节点管理命令](#)
 - [查询](#)
 - [下线节点](#)
 - [设置阈值](#)
- [数据节点管理命令](#)
 - [查询](#)
 - [下线节点](#)
- [卷管理命令](#)
 - [创建](#)
 - [删除](#)
 - [查询](#)
 - [统计](#)
 - [更新](#)
- [元数据分片管理命令](#)
 - [创建](#)
 - [查询](#)
 - [下线副本](#)
 - [比对副本](#)
- [数据分片管理命令](#)
 - [创建](#)
 - [查询](#)
 - [下线副本](#)
 - [比对副本文件](#)
 - [磁盘下线](#)
- [资源管理命令](#)
 - [增加](#)
 - [删除](#)

元数据节点

Metanode

- 元数据分片管理命令
 - 获取当前设备上所有分片信息
 - 获取指定分片ID的当前状态信息
- Inode管理命令
 - 获取指定Inode基本信息
 - 获取指定Inode的数据存储信息
 - 获取指定元数据分片的全部inode信息
- Dentry调试命令
 - 获取Dentry信息
 - 获取指定目录下全部文件
 - 获取指定分片的全部目录信息

集群管理命令

概述

```
1. curl -v "http://127.0.0.1/admin/getCluster" | python -m json.tool
```

展示集群基本信息，比如集群包含哪些数据节点和元数据节点，卷等。

响应示例

```
1. {
2.     "Name": "test",
3.     "LeaderAddr": "127.0.0.1:80",
4.     "DisableAutoAlloc": false,
5.     "Applied": 225,
6.     "MaxDataPartitionID": 100,
7.     "MaxMetaNodeID": 3,
8.     "MaxMetaPartitionID": 1,
9.     "DataNodeStat": {},
10.    "MetaNodeStat": {},
11.    "VolStat": {},
12.    "MetaNodes": {},
13.    "DataNodes": {}
14. }
```

冻结集群

```
1. curl -v "http://127.0.0.1/cluster/freeze?enable=true"
```

如果启用了冻结集群功能，卷就不在自动的创建数据分片

参数列表		
参数	类型	描述
enable	bool	如果设置为true，则集群被冻结

元数据节点管理命令

查询

```
1. curl -v "http://127.0.0.1/metaNode/get?addr=127.0.0.1:9021" | python -m json.tool
```

展示元数据节点的详细信息, 包括地址, 总的内存大小, 已使用内存大小等等.

参数列表		
参数	类型	描述
addr	string	元数据节点和master的交互地址

响应示例

```
1. {
2.     "ID": 3,
3.     "Addr": "127.0.0.1:9021",
4.     "IsActive": true,
5.     "Sender": {
6.         "TaskMap": {}
7.     },
8.     "Rack": "",
9.     "MaxMemAvailWeight": 66556215048,
10.    "TotalWeight": 67132641280,
11.    "UsedWeight": 576426232,
12.    "Ratio": 0.008586377967698518,
13.    "SelectCount": 0,
14.    "Carry": 0.6645600532184904,
15.    "Threshold": 0.75,
16.    "ReportTime": "2018-12-05T17:26:28.29309577+08:00",
17.    "MetaPartitionCount": 1
18. }
```

下线节点

```
1. curl -v "http://127.0.0.1/metaNode/decommission?addr=127.0.0.1:9021"
```

从集群中下线某个元数据节点, 该节点上的所有元数据分片都会被异步的迁移到集群中其它可用的元数据节点

参数列表		
------	--	--

参数	类型	描述
addr	string	元数据节点和master的交互地址

设置阈值

```
1. curl -v "http://127.0.0.1/threshold/set?threshold=0.75"
```

如果某元数据节点内存使用率达到这个阈值, 则该节点上所有的元数据分片都会被设置为只读.

参数列表

参数	类型	描述
threshold	float64	元数据节点能使用本机内存的最大比率

数据节点管理命令

查询

```
1. curl -v "http://127.0.0.1/dataNode/get?addr=127.0.0.1:5000" | python -m json.tool
```

显示数据节点的详情, 包括数据节点的地址, 总的容量, 已使用空间等等.

参数列表		
参数	类型	描述
addr	string	数据节点和master的交互地址

响应示例

```
1. {
2.     "MaxDiskAvailWeight": 3708923232256,
3.     "CreatedVolWeights": 2705829396480,
4.     "RemainWeightsForCreateVol": 36960383303680,
5.     "TotalWeight": 39666212700160,
6.     "UsedWeight": 2438143586304,
7.     "Available": 37228069113856,
8.     "Rack": "rack1",
9.     "Addr": "10.196.30.231:6000",
10.    "ReportTime": "2018-12-06T10:56:38.881784447+08:00",
11.    "Ratio": 0.06146650815226848,
12.    "SelectCount": 5,
13.    "Carry": 1.0655859145960367,
14.    "Sender": {
15.        "TaskMap": {}
16.    },
17.    "DataPartitionCount": 21
18. }
```

下线节点

```
1. curl -v "http://127.0.0.1/dataNode/decommission?addr=127.0.0.1:5000"
```

从集群中下线某个数据节点, 该数据节点上的所有数据分片都会被异步的迁移到集群中其它可用的数据节点

参数列表		
------	--	--

参数	类型	描述
addr	string	数据节点和master的交互地址

卷管理命令

创建

```
1. curl -v "http://127.0.0.1/admin/createVol?
    name=test&capacity=100&owner=cfs&mpCount=3"
```

为用户创建卷，并分配一组数据分片和元数据分片。在创建新卷时，默认分配10个数据分片和3个元数据分片。

参数列表		
参数	类型	描述
name	string	
capacity	int	卷的配额, 单位是GB
owner	string	vol的所有者
mpCount	int	初始化metaPartition个数

删除

```
1. curl -v "http://127.0.0.1/vol/delete?name=test&authKey=md5(owner)"
```

首先把卷标记为逻辑删除，然后通过周期性任务删除所有数据分片和元数据分片，最终从持久化存储中删除。

参数列表		
参数	类型	描述
name	string	
authKey	string	计算vol的所有者字段的MD5值作为认证信息

查询

```
1. curl -v "http://127.0.0.1/client/vol?name=test&authKey=md5(owner)" | python -m
    json.tool
```

展示卷的基本信息，包括卷的名字，所有的数据分片和元数据分片信息等。

参数列表		
参数	类型	描述

name	string	
authKey	string	计算vol的所有者字段的MD5值作为认证信息

响应示例

```
1. {
2.     "Name": "test",
3.     "VolType": "extent",
4.     "MetaPartitions": {},
5.     "DataPartitions": {}
6. }
```

统计

```
1. curl -v http://127.0.0.1/client/volStat?name=test
```

展示卷的总空间大小和已使用空间大小信息

参数列表		
参数	类型	描述
name	string	

响应示例

```
1. {
2.     "Name": "test",
3.     "TotalSize": 3221225472000000000,
4.     "UsedSize": 15551511283278
5. }
```

更新

```
1. curl -v "http://127.0.0.1/vol/update?name=test&capacity=100&authKey=md5(owner)"
```

增加卷的配额

参数列表		
参数	类型	描述
name	string	
capacity	int	卷的配额, 单位是GB
authKey	string	计算vol的所有者字段的MD5值作为认证信息

元数据分片管理命令

创建

```
1. curl -v "http://127.0.0.1/metaPartition/create?name=test&start=10000"
```

手动切分元数据分片, 如果卷的最大的元数据分片inode的范围是[0, end), end 大于start参数, 原来最大的元数据分片的inode范围变为[0, start], 新创建的元数据分片的范围是[start+1, end)

参数列表		
参数	类型	描述
name	string	卷的名字
start	uint64	根据此值切分元数据分片

查询

```
1. curl -v "http://127.0.0.1/client/metaPartition?id=1" | python -m json.tool
```

展示元数据分片的详细信息, 包括分片ID, 分片的起始范围等等.

参数列表		
参数	类型	描述
id	uint64	元数据分片ID

响应示例

```
1. {
2.     "PartitionID": 1,
3.     "Start": 0,
4.     "End": 9223372036854776000,
5.     "MaxNodeID": 1,
6.     "Replicas": {},
7.     "ReplicaNum": 3,
8.     "Status": 2,
9.     "PersistenceHosts": {},
10.    "Peers": {},
11.    "MissNodes": {}
12. }
```

下线副本

```
1. curl -v "http://127.0.0.1/metaPartition/decommission?id=13&addr=127.0.0.1:9021"
```

下线元数据分片的某个副本, 并且创建一个新的副本

参数列表		
参数	类型	描述
id	uint64	元数据分片ID
addr	string	要下线副本的地址

比对副本

```
1. curl -v "http://127.0.0.1/metaPartition/load?id=1"
```

发送比对副本任务到各个副本, 然后检查各个副本的Crc是否一致

参数列表		
参数	类型	描述
id	uint64	元数据分片ID

数据分片管理命令

创建

```
1. curl -v "http://127.0.0.1/dataPartition/create?count=400&name=test"
```

创建指定数量的数据分片

参数列表		
参数	类型	描述
count	int	创建多少个数据分片
name	string	卷的名字

查询

```
1. curl -v "http://127.0.0.1/dataPartition/get?id=100" | python -m json.tool
```

参数列表		
参数	类型	描述
id	uint64	数据分片的ID

响应示例

```
1. {
2.     "PartitionID": 100,
3.     "LastLoadTime": 1544082851,
4.     "ReplicaNum": 3,
5.     "Status": 2,
6.     "Replicas": {},
7.     "PartitionType": "extent",
8.     "PersistenceHosts": {},
9.     "Peers": {},
10.    "MissNodes": {},
11.    "VolName": "test",
12.    "RandomWrite": true,
13.    "FileInCoreMap": {}
14. }
```

下线副本

```
1. curl -v "http://127.0.0.1/dataPartition/decommission?id=13&addr=127.0.0.1:5000"
```

移除数据分片的某个副本，并且创建一个新的副本

参数列表

参数	类型	描述
id	uint64	数据分片的ID
addr	string	要下线的副本的地址

比对副本文件

```
1. curl -v "http://127.0.0.1/dataPartition/load?id=1"
```

给数据分片的每个副本都发送比对副本文件的任务，然后异步的检查每个副本上的文件crc是否一致

参数列表

参数	类型	描述
id	uint64	数据分片的ID

磁盘下线

```
1. curl -v "http://127.0.0.1/disk/decommission?addr=127.0.0.1:5000&disk=/cfs1"
```

同步下线磁盘上的所有数据分片，并且为每一个数据分配在集群内创建一个新的副本

参数列表

参数	类型	描述
addr	string	要下线的副本的地址
disk	string	故障磁盘

资源管理命令

增加

```
1. curl -v "http://127.0.0.1/raftNode/add?addr=127.0.0.1:80&id=3"
```

增加新的master节点到raft复制组

参数列表

参数	类型	描述
addr	string	master的ip地址, 格式为ip:port
id	uint64	master的节点标识

删除

```
1. curl -v "http://127.0.0.1/raftNode/remove?addr=127.0.0.1:80&id=3"
```

从raft复制组总移除某个节点

参数列表

参数	类型	描述
addr	string	master的ip地址, 格式为ip:port
id	uint64	master的节点标识

- [元数据分片管理命令](#)
- [Inode管理命令](#)
- [Dentry调试命令](#)

元数据分片管理命令

获取当前设备上所有分片信息

```
1. curl -v http://127.0.0.1:9092/getPartitions
```

获取指定分片ID的当前状态信息

```
1. curl -v http://127.0.0.1:9092/getPartitionById?pid=100
```

获取指定分片id的当前状态信息，包含当前分片组的raft leader地址，raft组成员，inode分配游标等信息

请求参数：		
参数	类型	描述
pid	整型	元数据分片的ID

Inode管理命令

获取指定Inode基本信息

```
1. curl -v http://127.0.0.1:9092/getInode?pid=100&ino=1024
```

请求参数说明:

参数	类型	描述
pid	整型	分片id
ino	整型	inode的id

获取指定Inode的数据存储信息

```
1. curl -v http://127.0.0.1:9092/getExtentsByInode?pid=100&ino=1024
```

请求参数:

参数	类型	描述
pid	整型	分片id
ino	整型	inode id

获取指定元数据分片的全部inode信息

```
1. curl -v http://127.0.0.1:9092/getAllInodes?pid=100
```

请求参数:

参数	类型	描述
pid	整型	分片 id

Dentry调试命令

获取Dentry信息

```
1. curl -v 'http://127.0.0.1:9092/getDentry?pid=100&name="aa.txt"&parentIno=1024'
```

Parameters		
Parameter	Type	Description
pid	integer	meta partition id
name	string	directory or file name
parentIno	integer	parent directory inode id

获取指定目录下全部文件

```
1. curl -v 'http://127.0.0.1:9092/getDirectory?pid=100&parentIno=1024'
```

Parameters		
Parameter	Type	Description
pid	integer	partition id
ino	integer	inode id

获取指定分片的全部目录信息

```
1. curl -v 'http://127.0.0.1:9092/getAllDentry?pid=100'
```

Parameters		
Parameter	Type	Description
pid	integer	partition id

运维手册

编译构建

编译服务端

使用如下命令同时构建server，client及相关的依赖：

```
1. make build
```

如果构建成功，将在 `build/bin` 目录中生成可执行文件 `cfs-server` 和 `cfs-client`。

集群部署

启动资源管理节点

```
1. nohup ./cfs-server -c master.json &
```

示例 `master.json` ：注意：master服务最少应该启动3个节点实例

```
1. {
2.   "role": "master",
3.   "ip": "192.168.31.173",
4.   "port": "80",
5.   "prof": "10088",
6.   "id": "1",
7.   "peers": "1:192.168.31.173:80,2:192.168.31.141:80,3:192.168.30.200:80",
8.   "retainLogs": "20000",
9.   "logDir": "/export/Logs/master",
10.  "logLevel": "info",
11.  "walDir": "/export/Data/master/raft",
12.  "storeDir": "/export/Data/master/rocksdbstore",
13.  "warnLogDir": "/export/home/tomcat/UMP-Monitor/logs/",
14.  "consulAddr": "http://consul.prometheus-cfs.local",
15.  "exporterPort": 9510,
16.  "clusterName": "cfs"
17. }
```

详细配置参数请参考 [资源管理节点](#)。

启动元数据节点

```
1. nohup ./cfs-server -c metanode.json &
```

示例 `meta.json` ：注意：metanode服务最少应该启动3个节点实例

```
1. {
2.     "role": "metanode",
3.     "listen": "9021",
4.     "prof": "9092",
5.     "logLevel": "info",
6.     "metadataDir": "/export/Data/metanode",
7.     "logDir": "/export/Logs/metanode",
8.     "raftDir": "/export/Data/metanode/raft",
9.     "raftHeartbeatPort": "9093",
10.    "raftReplicaPort": "9094",
11.    "totalMem": "17179869184",
12.    "consulAddr": "http://consul.prometheus-cfs.local",
13.    "warnLogDir": "/export/home/tomcat/UMP-Monitor/logs/",
14.    "exporterPort": 9511,
15.    "masterAddrs": [
16.        "192.168.31.173:80",
17.        "192.168.31.141:80",
18.        "192.168.30.200:80"
19.    ]
20. }
```

详细配置参数请参考 [元数据节点](#)。

启动数据节点

- 准备数据目录

推荐 使用单独磁盘作为数据目录，配置多块磁盘能够达到更高的性能。

磁盘准备

1.1 查看机器磁盘信息，选择给ChubaoFS使用的磁盘

```
1. fdisk -l
```

1.2 格式化磁盘，建议格式化为XFS

```
1. mkfs.xfs -f /dev/sdx
```

1.3 创建挂载目录

```
1. mkdir /data0
```

1.4 挂载磁盘

```
1. mount /dev/sdx /data0
```

- 启动数据节点

```
1. nohup ./cfs-server -c datanode.json &
```

示例 `datanode.json` :注意: datanode服务最少应该启动4个节点实例

```
1. {
2.   "role": "datanode",
3.   "port": "6000",
4.   "prof": "6001",
5.   "logDir": "/export/Logs/datanode",
6.   "logLevel": "info",
7.   "raftHeartbeat": "9095",
8.   "raftReplica": "9096",
9.   "consulAddr": "http://consul.prometheus-cfs.local",
10.  "warnLogDir": "/export/home/tomcat/UMP-Monitor/logs/",
11.  "exporterPort": 9512,
12.  "masterAddr": [
13.    "192.168.31.173:80",
14.    "192.168.31.141:80",
15.    "192.168.30.200:80"
16.  ],
17.  "rack": "",
18.  "disks": [
19.    "/data0:21474836480",
20.    "/data1:21474836480"
21.  ]
22. }
```

详细配置参数请参考 [数据节点](#)。

创建Volume卷

1. `curl -v "http://192.168.31.173/admin/createVol?name=test&capacity=10000&owner=cfs"`
- 2.
3. 如果执行性能测试，请调用相应的API，创建足够多的数据分片（data partition），如果集群中有8块磁盘，那么需要创建80个datapartition

挂载客户端

- 运行 `modprobe fuse` 插入FUSE内核模块。
- 运行 `yum install -y fuse` 安装libfuse。
- 运行 `nohup client -c fuse.json &` 启动客户端。

样例 `fuse.json` ,

```

1. {
2.   "mountPoint": "/mnt/fuse",
3.   "volName": "test",
4.   "owner": "cfs",
5.   "masterAddr": "192.168.31.173:80,192.168.31.141:80,192.168.30.200:80",
6.   "logDir": "/export/Logs/client",
7.   "warnLogDir": "/export/home/tomcat/UMP-Monitor/logs/",
8.   "profPort": "10094",
9.   "logLevel": "info"
10. }
```

详细配置参数请参考 [客户端](#)。

升级注意事项

集群数据节点和元数据节点升级前，请先禁止集群自动为卷扩容数据分片。

- 冻结集群

```
1. curl -v "http://192.168.31.173/cluster/freeze?enable=true"
```

- 升级节点
- 开启自动扩容数据分片

```
1. curl -v "http://192.168.31.173/cluster/freeze?enable=false"
```

集群管理

资源管理节点

- 动态添加新节点假设原集群资源管理节点有三台，节点标识分别是1、2、3，现在要加入节点4。

机器列表

ip:端口号	节点标识id
192.168.31.173:80	1
192.168.31.174:80	2
192.168.31.175:80	3
192.168.31.176:80	4

操作步骤如下

1.1 执行资源管理添加节点的命令

```
1. curl -v "http://192.168.31.173/raftNode/add?addr=192.168.31.176:80&id=4&#34;
```

1.2 启动资源管理节点4

```
1. nohup ./cfs-server -c master.json &
```

- 动态删除节点4

1.1 执行资源管理添加节点的命令

```
1. curl -v "http://192.168.31.173/raftNode/remove?addr=192.168.31.176:80&id=4&#34;
```

1.2 关闭资源管理节点4

```
1. pkill cfs-server
```

- 静态添加或删除节点直接修改所有节点的配置文件，peers项只保留想要的节点，重启所有资源管理节点即可

```
1. {  
2.   "peers": "1:192.168.31.173:80,2:192.168.31.174:80,3:192.168.31.175:80",  
3. }
```

元数据节点

- 扩容新节点在新节点上编辑好配置文件，执行启动命令，详情请参考启动元数据节点章节
- 下线节点为了保证有足够的副本冗余，如果出现机器故障，需要及时下线故障机器。执行元数据节点下线的命令，命令详情请参考 [元数据节点管理命令](#)。

```
1. curl -v "http://127.0.0.1/metaNode/decommission?addr=127.0.0.1:9021"
```

数据节点

- 扩容新节点在新节点上编辑好配置文件，执行启动命令，详情请参考启动数据节点章节
- 下线节点为了保证有足够的副本冗余，如果出现机器故障，需要及时下线故障机器。执行数据节点下线的命令，命令详情请参考 [数据节点管理命令](#)。

```
1. curl -v "http://127.0.0.1/dataNode/decommission?addr=127.0.0.1:9021"
```

磁盘损坏处理

磁盘损坏事件经常发生，当数据节点出现磁盘损坏时，要及时下线有问题的磁盘。具体操作步骤如下

1. 执行磁盘下线命令

```
1. curl -v "http://127.0.0.1/disk/decommission?addr=127.0.0.1:5000&disk=/cfs1&#34;
```

2. 修改数据节点配置文件，重启数据节点

2.1 从配置文件中删除故障磁盘，详细配置参数请参考 [数据节点](#)。

2.2 关闭并重启数据节点

```
1. pkill cfs-servernohup ./cfs-server -c datanode.json &
```

使用案例

ChubaoFS是一个分布式文件系统，兼容绝大部分POSIX文件系统语义，挂载后可以像使用本地文件系统一样简单。基本上可以用在任何需要文件系统的场合，替换本地文件系统，实现可无限扩展的、无物理边际的存储。已经应用在多种场景，下面是摘取的部分场景

HBase

使用hdfs作为后端存储数据会经常遇到下面的问题

- 对小文件不友好，小文件个数太多，比较占用namenode内存，文件系统所能容纳的文件数目是由Namenode 的内存大小来决定
- 运维复杂
- Namenode管理元数据，不易扩容

使用chubaofs作为后端存储的优势

- 对小文件友好，文件个数不受限制
- 运维简单
- Metanode管理文件元数据，支持水平扩容
- 不同的volume文件元数据分布在不同的metanode上，彼此隔离，互不影响

ElasticSearch

使用本地磁盘存储数据会经常遇到下面的问题：

- 磁盘使用率不均匀，磁盘IO无法充分利用
- 本地磁盘空间大小受限制

使用chubaofs作为后端存储的优势

- 磁盘空间不受限制，易扩容，根据磁盘使用百分比自动扩展磁盘容量，能够实现存储系统按需扩容，极大的节省存储成本
- 磁盘IO使用率均匀，磁盘IO得到充分利用
- 保证数据高可靠，不用担心丢失数据

Nginx日志存储

你是不是经常为本地磁盘空间已满问题而发愁，使用chubaofs，可以将数据存储到分布式文件系统中，不用再担心磁盘空间不够用问题。

使用本地磁盘存储日志的问题

- docker本地磁盘空间小
- docker容器故障，日志丢失且不可恢复
- 物理机和docker机器混合部署，难以管理，运维成本高

使用chubaofs存储Nginx日志优势

- 磁盘空间不受限制，易扩容，根据磁盘使用百分比自动扩展磁盘容量，能够实现存储系统按需扩容，极大的节省存储成本
- 保证数据高可靠，不用担心丢失数据
- 多副本，可解决磁盘级和datanode节点故障导致日志无法写入问题
- 兼容posix文件系统接口，应用程序无需任何改动
- chubaofs运维简单，一个人就可以轻松的管理上万台机器的集群

Spark

在大数据集场景下，你是否为存储Spark中间计算结果需要精心计算每个task的数据量而发愁，可以将shuffle结果存储到cfs，不用再担心磁盘没有可用空间而导致任务失败问题，实现存储和计算分离。

使用本地磁盘存储shuffle中间结果的痛点

- 磁盘空间不足
- 临时目录文件过多，无法创建新文件

使用chubaofs优势

- 磁盘空间不受限制，易扩容，根据磁盘使用百分比自动扩展磁盘容量，能够实现存储系统按需扩容，极大的节省存储成本
- Metanode管理文件元数据，可以水平扩容，文件个数不受限制

MySQL数据库备份

使用云存储备份MySQL数据库的缺点

- 需要利用云存储SDK或者RESTful API开发备份程序，增加运维难度。
- 备份文件失败，排查问题比较困难
- 备份文件到云存储后，不方便查看文件是否上传成功
- 备份文件经过多层服务处理，影响性能

使用chubaofs备份MySQL数据库的优点

- 简单易用，兼容POSIX文件接口，可以当作本地文件系统使用

- 完整且详尽的操作记录存储在本地文件系统中，排查问题简单方便
- 只需执行ls命令，即可验证文件是否上传成功
- 支持PageCache和WriteCache，与云存储相比，文件读写性能显著提升

机器学习

使用本地磁盘存储训练数据集的缺点

- 本地磁盘空间小，有多个模型，每个模型的训练数据集到达TB级别，使用本地磁盘存储训练数据集，需要缩减训练数据集大小
- 训练数据集需要经常更新，需要更多的磁盘空间
- 如果机器故障，存在训练数据集丢失风险

使用chubaofs存储点击流日志优势

- 磁盘空间不受限制，易扩容，根据磁盘使用百分比自动扩展磁盘容量，能够实现存储系统按需扩容，极大的节省存储成本
- 数据有多个副本，保证数据高可靠，不用担心丢失数据
- 兼容posix文件系统接口，应用程序无需任何改动

性能评估

环境准备

集群信息

集群信息						
节点类型	节点数	CPU	内存	存储	网络	备注
管理节点	3	32	32 GB	120 GB SSD	10 Gb/s	
元数据节点	10	32	32 GB	16 x 1TB SSD	10 Gb/s	混合部署
数据节点	10	32	32 GB	16 x 1TB SSD	10 Gb/s	混合部署

卷设置

```
1. #!/bin/bash
2. # create volume
3. curl "http://${MASTER}/admin/createVol?
    name=intest&owner=cfs&capacity=3000000000&mpCount=10"
4. # increase 1500 data partitions
5. curl "http://${MASTER}/dataPartition/create?name=intest&count=1500"
```

- 初始元数据分区数： 10
- 元数据副本数： 3
- 初始数据分区数： 1510
- 数据副本数： 3

小文件性能评估

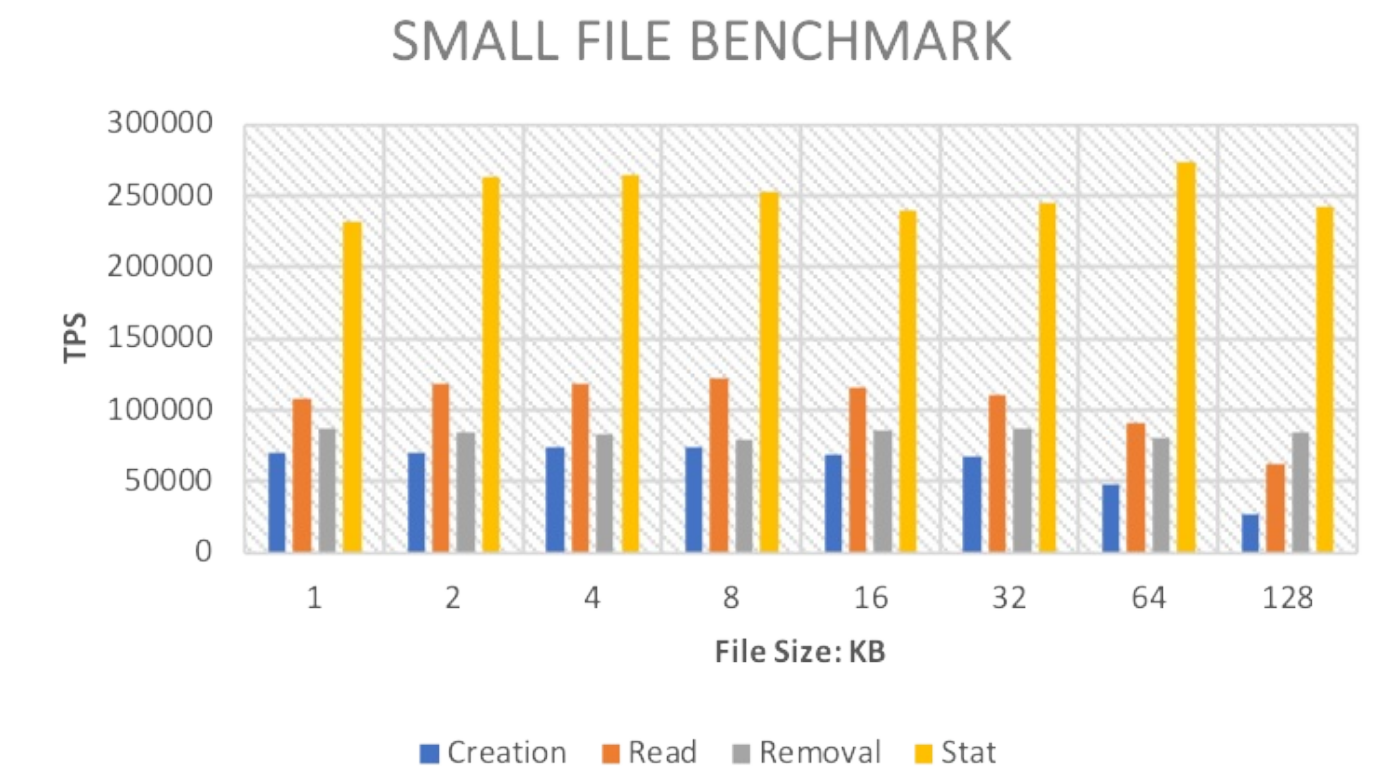
通过 mdtest 进行小文件性能测试的结果如下：

配置

```
1. #!/bin/bash
2. set -e
3. TARGET_PATH="/mnt/test/mdtest" # mount point of ChubaoFS volume
4. for FILE_SIZE in 1024 2048 4096 8192 16384 32768 65536 131072 # file size
5. do
6. mpirun --allow-run-as-root -np 512 --hostfile hfile64 mdtest -n 1000 -w $i -e
    $FILE_SIZE -y -u -i 3 -N 1 -F -R -d $TARGET_PATH;
```

7. [done](#)

测试结果



小文件性能评估结果								
文件大小 (KB)	1	2	4	8	16	32	64	128
创建操作 (TPS)	70383	70383	73738	74617	69479	67435	47540	27147
读取操作 (TPS)	108600	118193	118346	122975	116374	110795	90462	62082
删除操作 (TPS)	87648	84651	83532	79279	85498	86523	80946	84441
信息查看 (TPS)	231961	263270	264207	252309	240244	244906	273576	242930

IO性能评估

通过 [fio](#) 进行IO性能测试的结果如下：

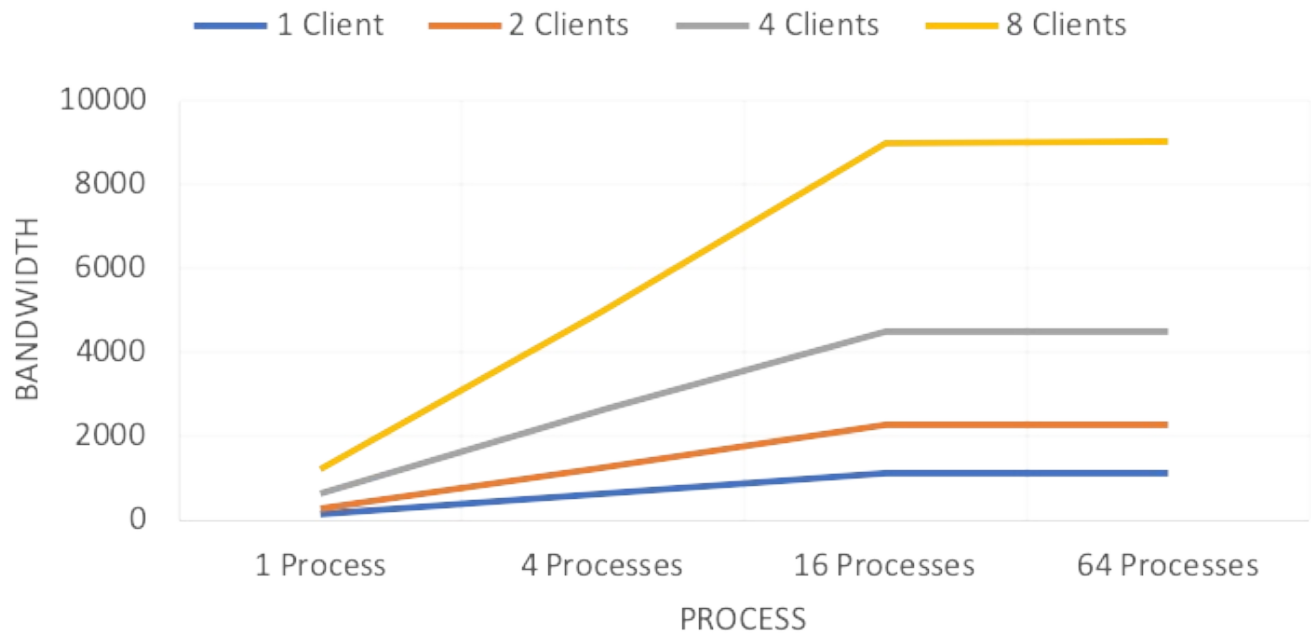
1. 顺序读

工具设置

```
1. #!/bin/bash
2. fio -directory={} \
3.     -ioengine=psync \
4.     -rw=read \ # sequential read
5.     -bs=128k \ # block size
6.     -direct=1 \ # enable direct IO
7.     -group_reporting=1 \
8.     -fallocate=none \
9.     -time_based=1 \
10.    -runtime=120 \
11.    -name=test_file_c{} \
12.    -numjobs={} \
13.    -nrfiles=1 \
14.    -size=10G
```

带宽

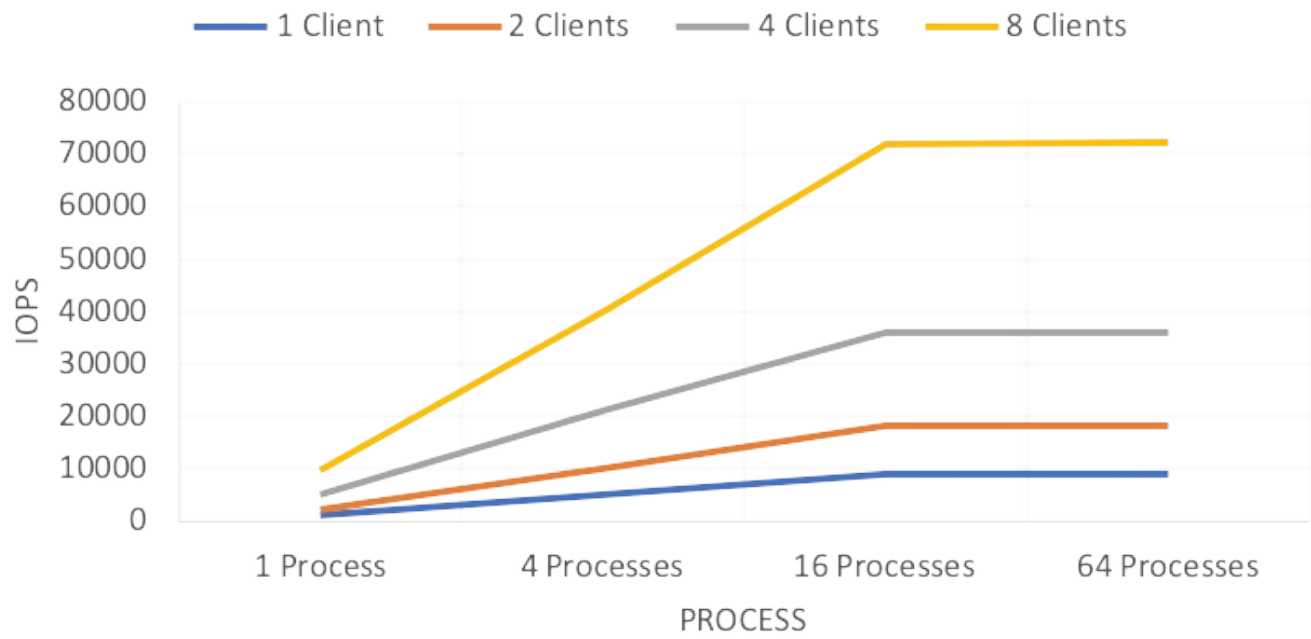
SEQUENTIAL READ BANDWIDTH



	顺序读带宽（MB/s）			
	1 进程	4 进程	16 进程	64 进程
1 客户端	148.000	626.000	1129.000	1130.000
2 客户端	284.000	1241.000	2258.000	2260.000
4 客户端	619.000	2640.000	4517.000	4515.000
8 客户端	1193.000	4994.000	9006.000	9034.000

IOPS

SEQUENTIAL READ IOPS



顺序读IOPS				
	1 进程	4 进程	16 进程	64 进程
1 客户端	1180.000	5007.000	9031.000	9040.000
2 客户端	2275.000	9924.000	18062.000	18081.000
4 客户端	4954.000	21117.000	36129.000	36112.000
8 客户端	9531.000	39954.000	72048.000	72264.000

延迟

SEQUENTIAL READ LATENCY



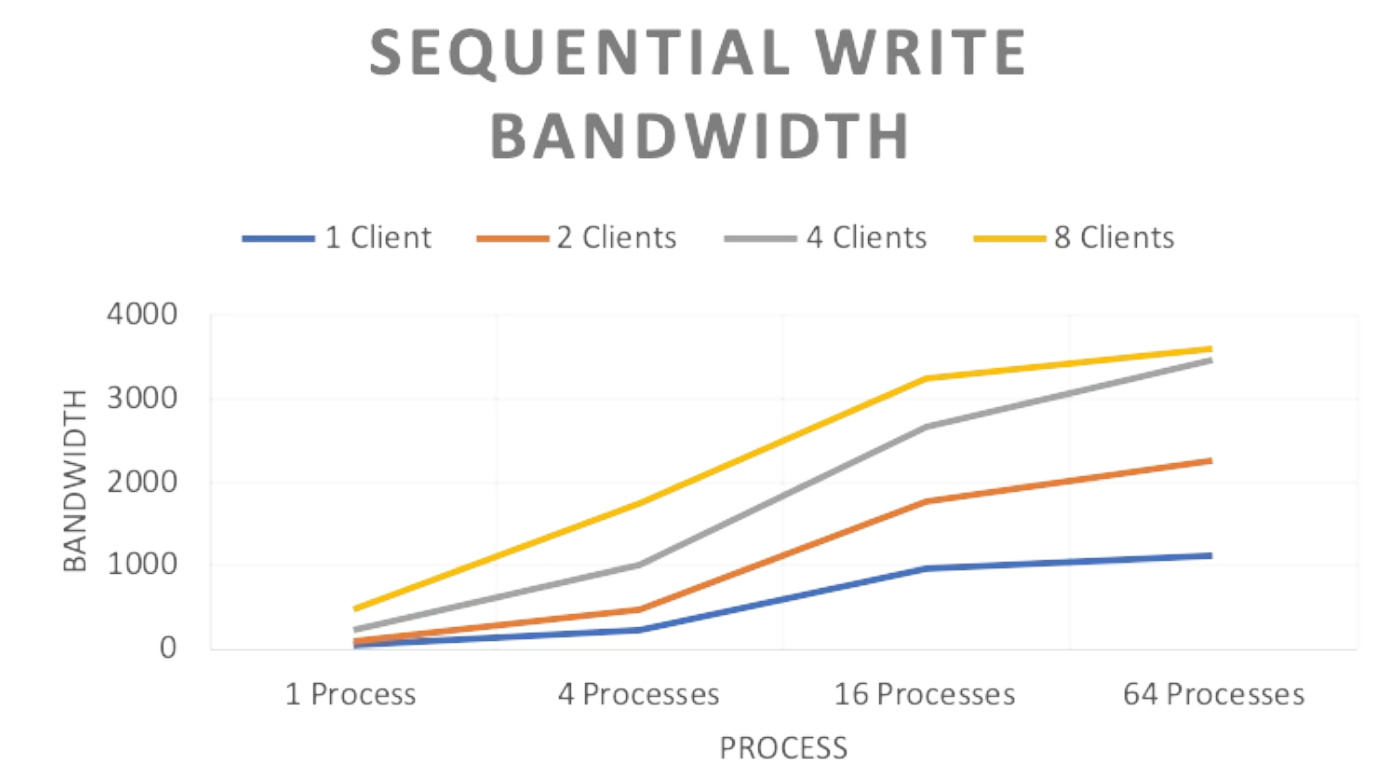
	顺序读延迟（微秒）			
	1 进程	4 进程	16 进程	64 进程
1 客户端	842.200	794.340	1767.310	7074.550
2 客户端	874.255	801.690	1767.370	7071.715
4 客户端	812.363	760.702	1767.710	7077.065
8 客户端	837.707	799.851	1772.620	7076.967

2. 顺序写

工具设置

```
1. #!/bin/bash
2. fio -directory={} \
3.     -ioengine=psync \
4.     -rw=write \ # sequential write
5.     -bs=128k \ # block size
6.     -direct=1 \ # enable direct IO
7.     -group_reporting=1 \
8.     -fallocate=none \
9.     -name=test_file_c{} \
10.    -numjobs={} \
11.    -nrfiles=1 \
12.    -size=10G
```

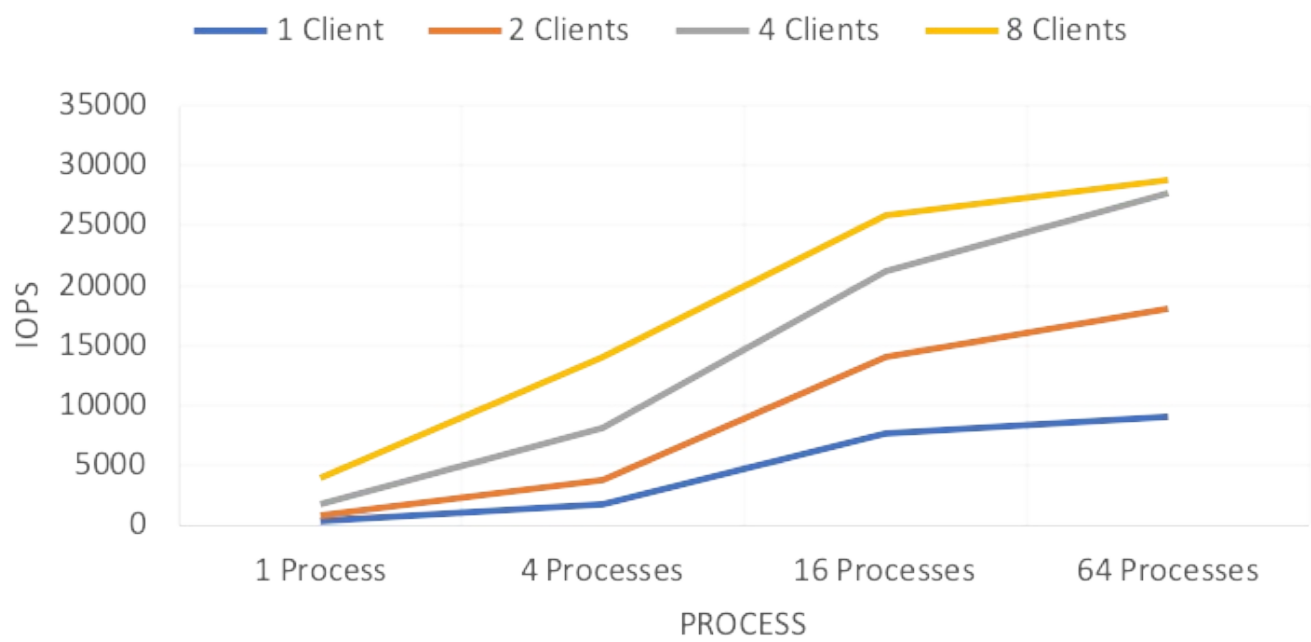
带宽



顺序写带宽 （MB/s）				
	1 进程	4 进程	16 进程	64 进程
1 客户端	52.200	226.000	956.000	1126.000
2 客户端	104.500	473.000	1763.000	2252.000
4 客户端	225.300	1015.000	2652.000	3472.000
8 客户端	480.600	1753.000	3235.000	3608.000

IOPS

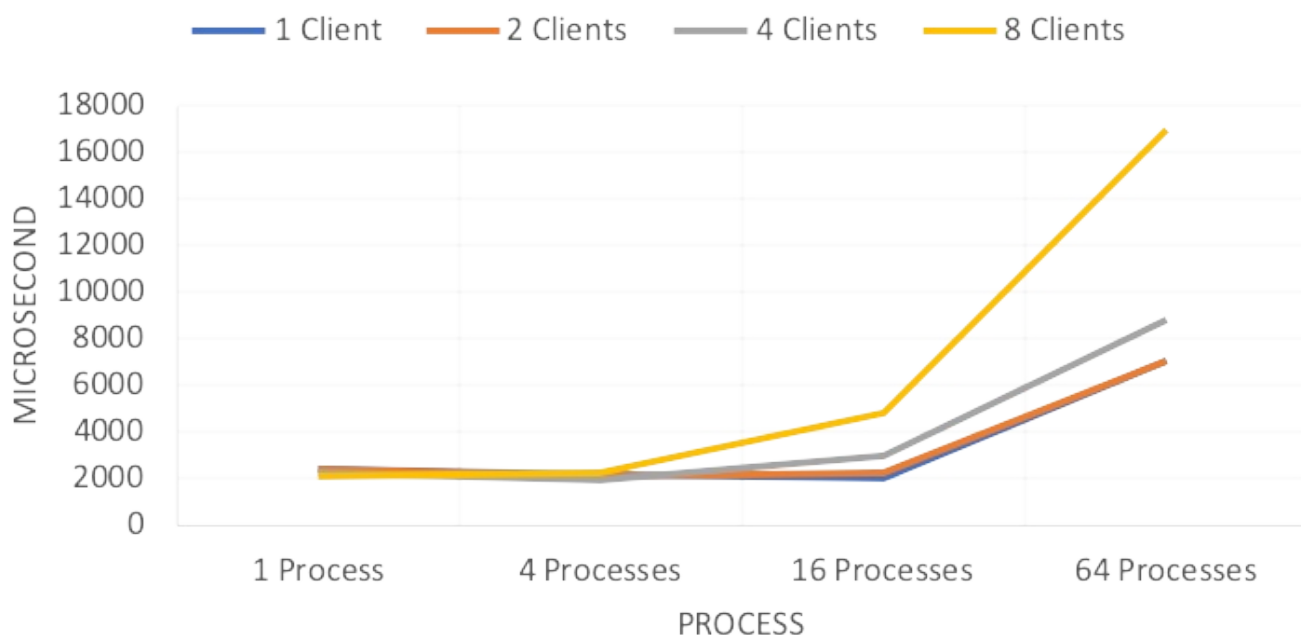
SEQUENTIAL WRITE IOPS



顺序写IOPS				
	1 进程	4 进程	16 进程	64 进程
1 客户端	417	1805	7651	9004
2 客户端	835	3779	14103	18014
4 客户端	1801	8127	21216	27777
8 客户端	3841	14016	25890	28860

延迟

SEQUENTIAL WRITE LATENCY



顺序写延迟（微秒）

	1 进程	4 进程	16 进程	64 进程
1 客户端	2385.400	2190.210	2052.360	7081.320
2 客户端	2383.610	2081.850	2233.790	7079.450
4 客户端	2216.305	1947.688	2946.017	8842.903
8 客户端	2073.921	2256.120	4787.496	17002.425

3. 随机读

工具设置

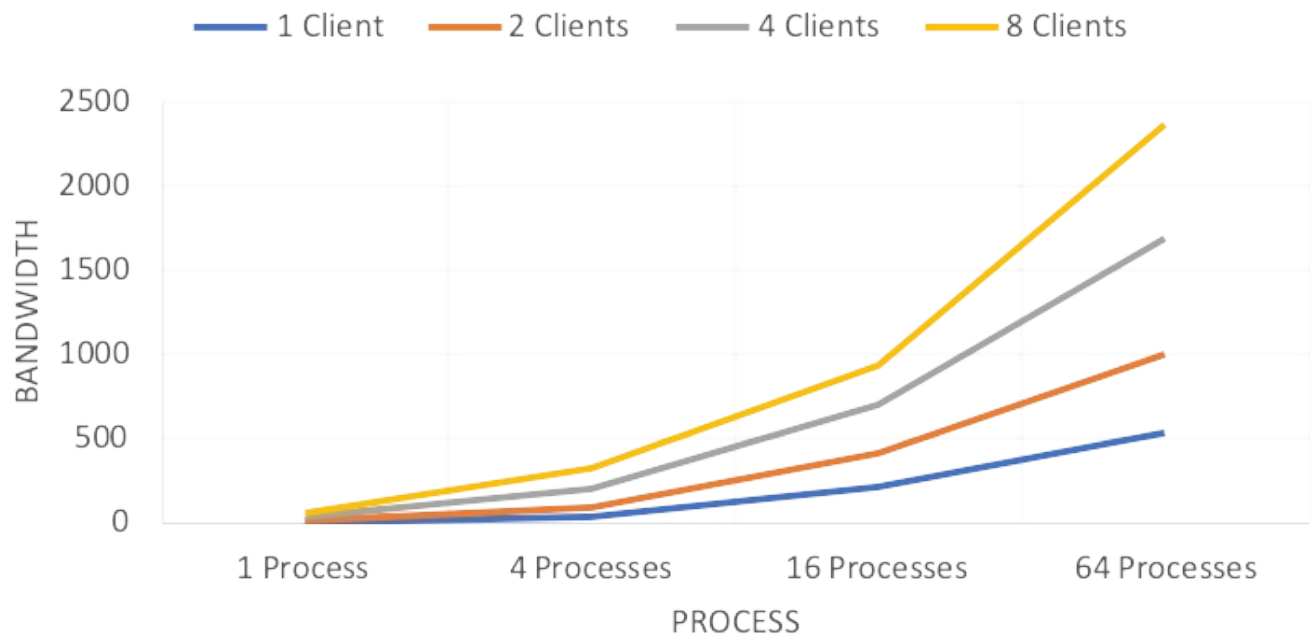
```

1. #!/bin/bash
2. fio -directory={} \
3.     -ioengine=psync \
4.     -rw=randread \ # random read
5.     -bs=4k \       # block size
6.     -direct=1 \     # enable direct IO
7.     -group_reporting=1 \
8.     -fallocate=none \
9.     -time_based=1 \
10.    -runtime=120 \
11.    -name=test_file_c{} \
12.    -numjobs={} \
13.    -nrfiles=1 \
14.    -size=10G

```

带宽

RANDOM READ BANDWIDTH



	随机读带宽（MB/s）			
	1 进程	4 进程	16 进程	64 进程
1 客户端	6.412	39.100	216.000	534.000
2 客户端	14.525	88.100	409.000	1002.000
4 客户端	33.242	200.200	705.000	1693.000
8 客户端	59.480	328.300	940.000	2369.000

IOPS

RANDOM READ IOPS



随机读IOPS				
	1 进程	4 进程	16 进程	64 进程
1 客户端	1641	10240	56524.800	140288
2 客户端	3718	23142.4	107212.8	263168
4 客户端	8508	52428.8	184627.2	443392
8 客户端	15222	85072.8	246681.6	621056

延迟

RANDOM READ LATENCY



随机读延迟（微秒）

	1 进程	4 进程	16 进程	64 进程
1 客户端	603.580	395.420	287.510	466.320
2 客户端	532.840	351.815	303.460	497.100
4 客户端	469.025	317.140	355.105	588.847
8 客户端	524.709	382.862	530.811	841.985

4. 随机写

工具设置

```

1. #!/bin/bash
2. fio -directory={} \
3.     -ioengine=psync \
4.     -rw=randwrite \ # random write
5.     -bs=4k \        # block size
6.     -direct=1 \     # enable direct IO
7.     -group_reporting=1 \
8.     -fallocate=none \
9.     -time_based=1 \
10.    -runtime=120 \
11.    -name=test_file_c{} \
12.    -numjobs={} \
13.    -nrfiles=1 \
14.    -size=10G

```

带宽

RANDOM WRITE BANDWIDTH



	随机写带宽（MB/s）			
	1 进程	4 进程	16 进程	64 进程
1 客户端	3.620	17.500	118.000	318.000
2 客户端	7.540	44.800	230.000	476.000
4 客户端	16.245	107.700	397.900	636.000
8 客户端	39.274	208.100	487.100	787.100

IOPS

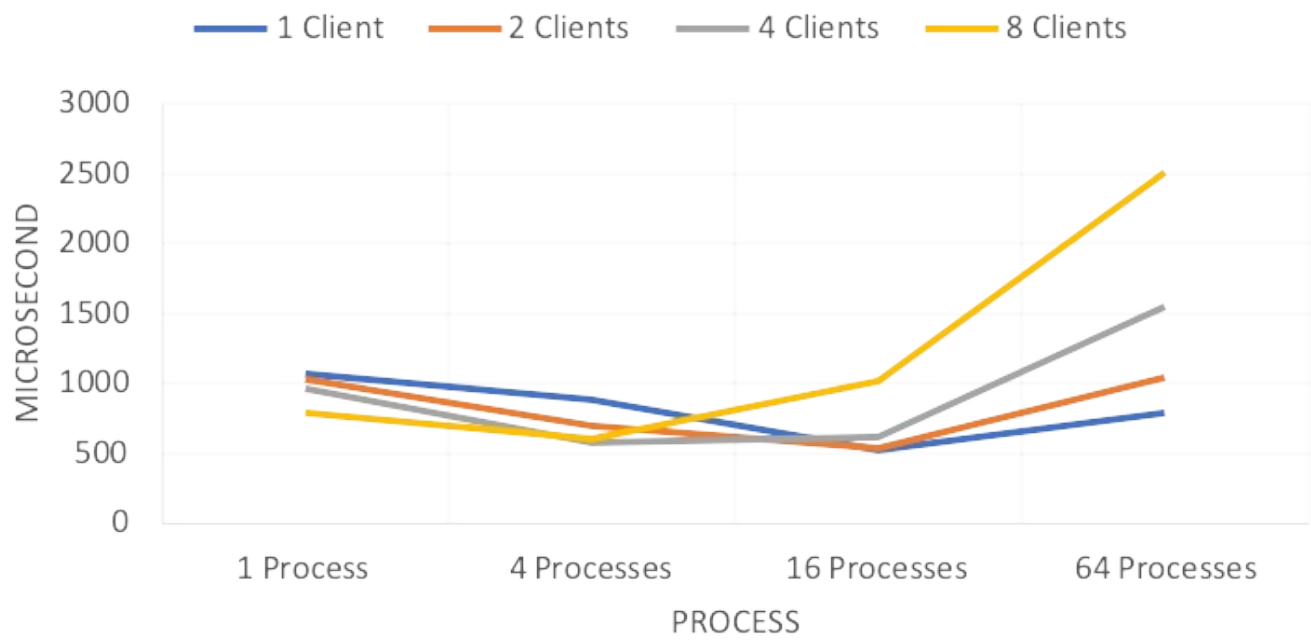
RANDOM WRITE IOPS



随机写IOPS				
	1 进程	4 进程	16 进程	64 进程
1 客户端	926.000	4476.000	31027.200	83251.200
2 客户端	1929.000	11473.000	60313.600	124620.800
4 客户端	4156.000	27800.000	104243.200	167014.400
8 客户端	10050.000	53250.000	127692.800	206745.600

延迟

RANDOM WRITE LATENCY



	随机写延迟（微秒）			
	1 进程	4 进程	16 进程	64 进程
1 客户端	1073.150	887.570	523.820	784.030
2 客户端	1030.010	691.530	539.525	1042.685
4 客户端	955.972	575.183	618.445	1552.205
8 客户端	789.883	598.393	1016.185	2506.424

元数据性能评估

通过 `mdtest` 进行元数据性能测试的结果如下：

工具设置

```
1. #!/bin/bash
2. TEST_PATH=/mnt/cfs/mdtest # mount point of ChubaoFS volume
3. for CLIENTS in 1 2 4 8 # number of clients
4. do
5. mpirun --allow-run-as-root -np $CLIENTS --hostfile hfile01 mdtest -n 5000 -u -z
   2 -i 3 -d $TEST_PATH;
6. done
```

目录创建

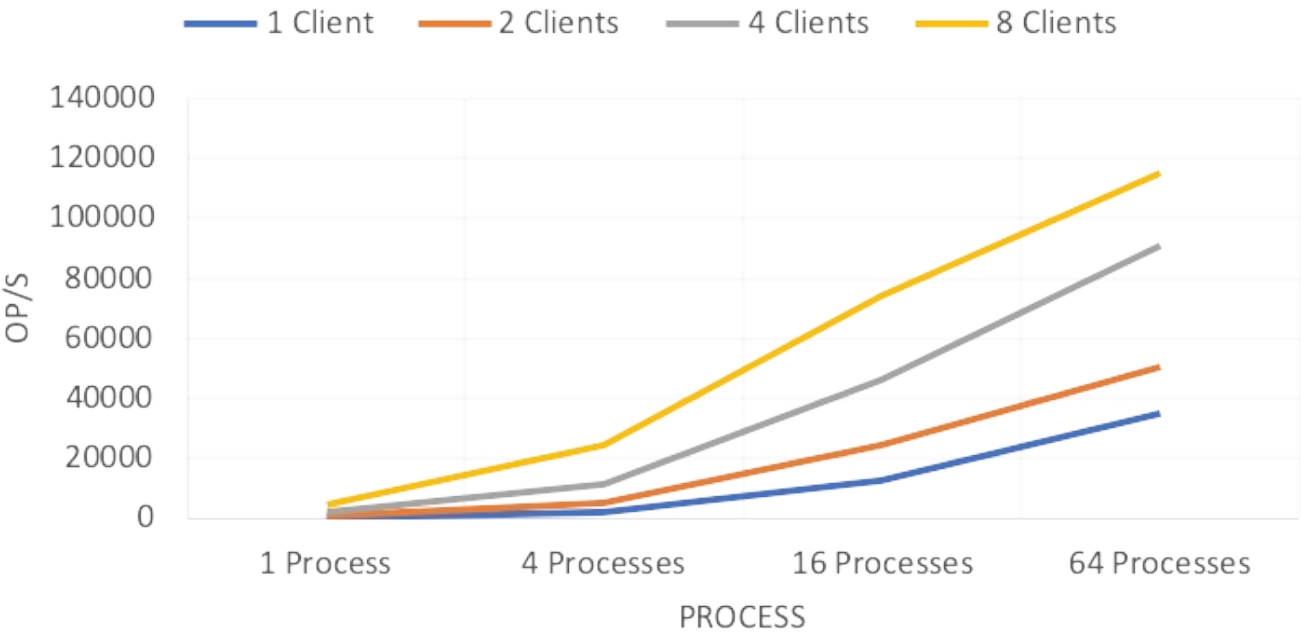
DIR CREATION



目录创建评估结果				
	1 进程	4 进程	16 进程	64 进程
1 客户端	448.618	2421.001	14597.97	43055.15
2 客户端	956.947	5917.576	28930.431	72388.765
4 客户端	2027.02	13213.403	54449.056	104771.356
8 客户端	4643.755	27416.904	89641.301	119542.62

目录删除

DIR REMOVAL

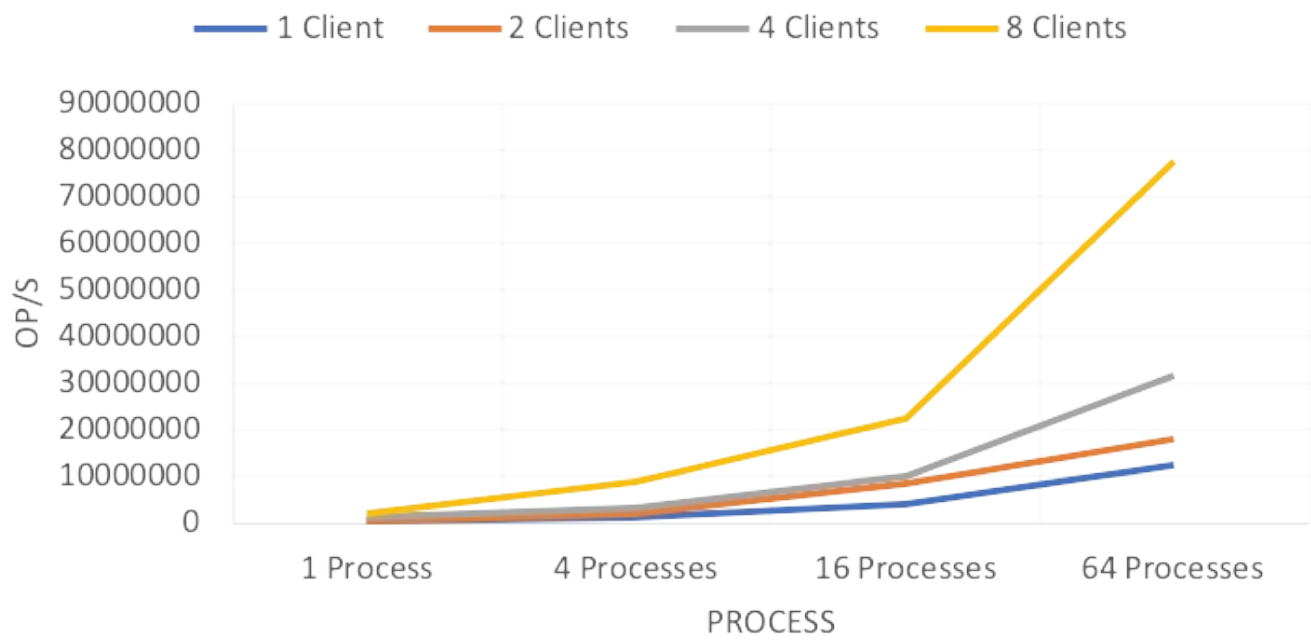


目录删除评估结果

	1 进程	4 进程	16 进程	64 进程
1 客户端	399.779	2118.005	12351.635	34903.672
2 客户端	833.353	5176.812	24471.674	50242.973
4 客户端	1853.617	11462.927	46413.313	91128.059
8 客户端	4441.435	24133.617	74401.336	115013.557

目录状态查看

DIR STAT

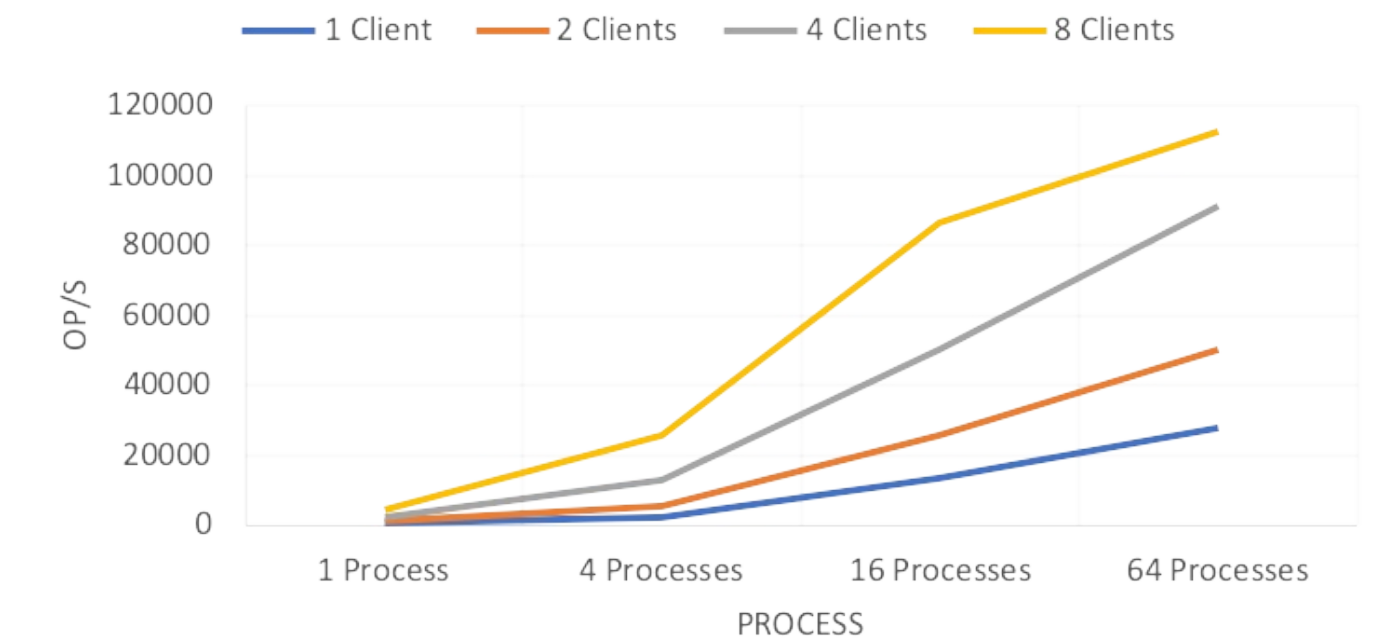


目录状态查看评估结果

	1 进程	4 进程	16 进程	64 进程
1 客户端	283232.761	1215309.524	4231088.104	12579177.02
2 客户端	572834.143	2169669.058	8362749.217	18120970.71
4 客户端	1263474.549	3333746.786	10160929.29	31874265.88
8 客户端	2258670.069	8715752.83	22524794.98	77533648.04

文件创建

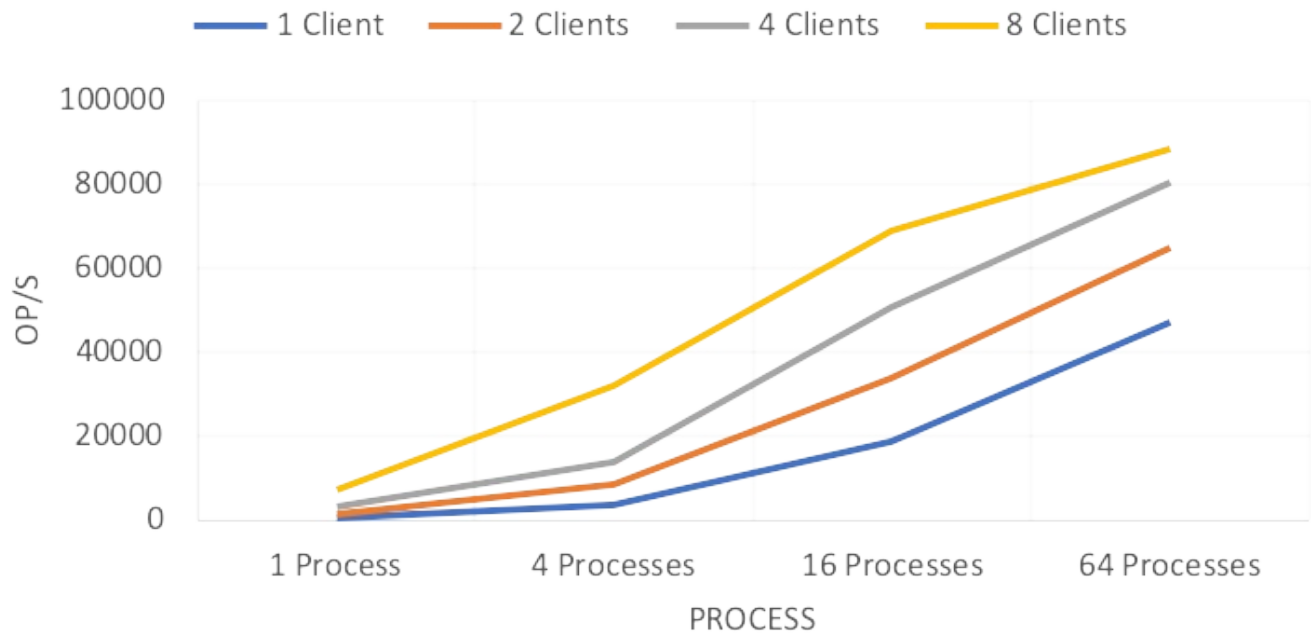
FILE CREATION



文件创建评估结果				
	1 进程	4 进程	16 进程	64 进程
1 客户端	448.888	2400.803	13638.072	27785.947
2 客户端	925.68	5664.166	25889.163	50434.484
4 客户端	2001.137	12986.968	50330.952	91387.825
8 客户端	4479.831	25933.437	86667.966	112746.199

文件删除

FILE REMOVAL

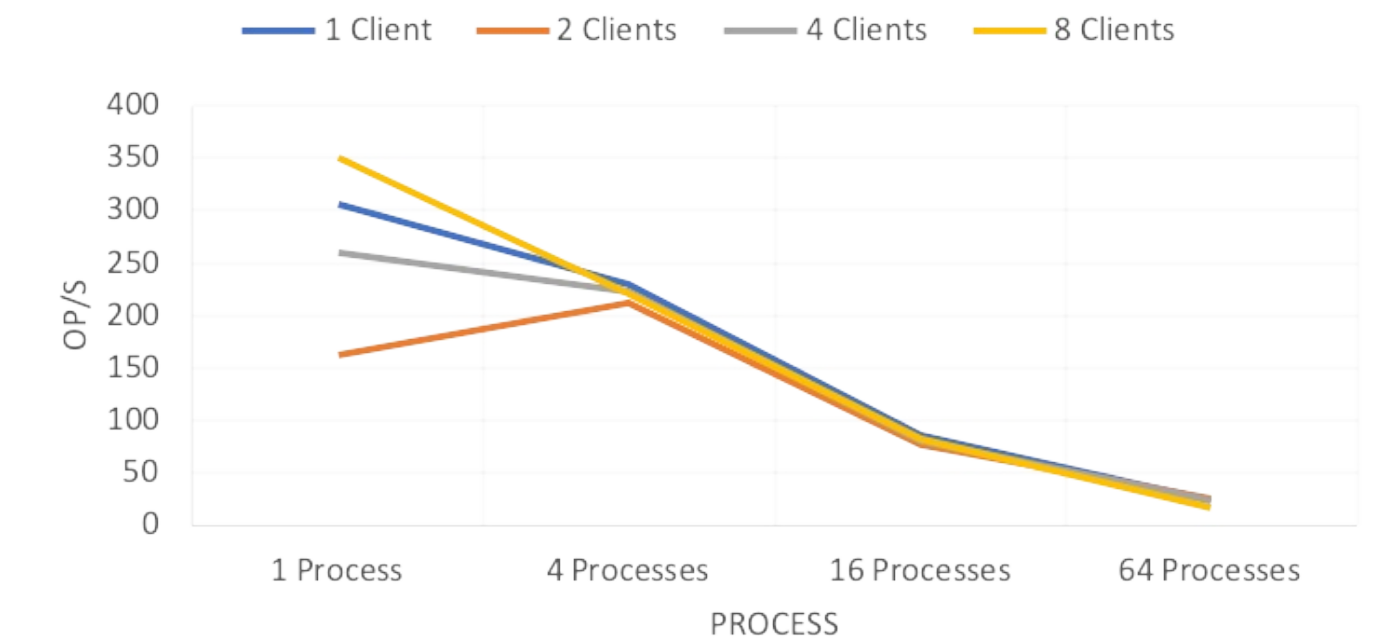


文件删除评估结果

	1 进程	4 进程	16 进程	64 进程
1 客户端	605.143	3678.138	18631.342	47035.912
2 客户端	1301.151	8365.667	34005.25	64860.041
4 客户端	3032.683	14017.426	50938.926	80692.761
8 客户端	7170.386	32056.959	68761.908	88357.563

Tree创建

TREE CREATION



Tree创建评估结果				
	1 进程	4 进程	16 进程	64 进程
1 客户端	305.778	229.562	86.299	23.917
2 客户端	161.31	211.119	76.301	24.439
4 客户端	260.562	223.153	81.209	23.867
8 客户端	350.038	220.744	81.621	17.144

Tree删除

TREE REMOVAL



Tree删除评估结果				
	1 进程	4 进程	16 进程	64 进程
1 客户端	137.462	70.881	31.235	7.057
2 客户端	217.026	113.36	23.971	7.128
4 客户端	231.622	113.539	30.626	7.299
8 客户端	185.156	83.923	20.607	5.515

功能完整性评估

- Linux Test Project / fs

多种负载评估

- Database backup
- Java application logs
- Code git repo
- Database systems

MyRocks, MySQL Innodb, HBase,

可扩展性评估

- 卷扩展性：单集群可以支持百万级别的cfs卷
- 元数据扩展性：单卷可以支持十亿级别文件或者目录

常见问题

- undefined reference to 'ZSTD_versionNumber' 类似问题

可以使用下面两种方式解决

1. CGO_LDFLAGS添加指定库即可编译，

例如： `CGO_LDFLAGS="-L/usr/local/lib -lrocksdb -lzstd"` 这种方式，要求其他部署机器上也要安装 `zstd` 库

2. 删除自动探测是否安装zstd库的脚本

文件位置示例： `rocksdb-5.9.2/build_tools/build_detect_platform`

删除的内容如下

```
1. # Test whether zstd library is installed    $CXX $CFLAGS $COMMON_FLAGS -x c++ - -o
/dev/null 2>/dev/null <<EOF      #include <zstd.h>      int main() {}EOF      if [ "$?" =
0 ]; then          COMMON_FLAGS="$COMMON_FLAGS -DZSTD"
PLATFORM_LDFLAGS="$PLATFORM_LDFLAGS -lzstd"          JAVA_LDFLAGS="$JAVA_LDFLAGS -lzstd"
fi
```

- 本机编译ChubaoFS，部署到其它机器上无法启动

首先请确认使用 `PORTABLE=1 make static_lib` 命令编译rocksdb，然后使用ldd命令查看依赖的库，在机器上是否安装，安装缺少的库后，执行 `ldconfig` 命令