

1. 版本选择

1. 建议使用最近版本 v2.4.0

1. 增加了一些新功能, 支持volume级别的监控, 删除&修复流程使用多路复用等优化

2. 之前的版本升级到v2.4.0

1. 为了不影响线上升级, 多路复用使用了新的端口进行通信。新的端口是17710和17810, 是在之前meta&data组件监听的端口17210&17310上加500得到的。(根据实际情况进行防火墙配置)

2. 升级的时候依次升级datanode, metanode, master和client, 具体参考

<https://github.com/chubaofs/chubaofs/releases/tag/v2.4.0>

3. 注意: v2.2.2之前的老版本, 升级之前自行参考下升级文档

2. 机器配置

1. master

- 资源管理节点, 要求不高
- 磁盘空间 (100G)&内存(32G)足够即可, 用于存储一些资源节点管理类信息
 - 16cpu+32G内存+100G磁盘+centos7.6
- 3台机器

2. datanode

- 数据节点 (物理机部署, centos 7.6)
- 内存要求不高, cpu数量与磁盘数量相关, 若一个datanode节点挂载磁盘过多, 需要较多的cpu来支持
 - 单机的磁盘存储总量不宜过大, 导致cpu负载高
 - 之前12cpu+36disk的机型, 老是负载高 (可能会是响应变慢)
 - 可以根据业务类型来选择磁盘的类型 (要求高可以使用SSD或者nvme的)
 - 目前还不支持一个集群里面有多种磁盘类型, 后续版本里面将会支持
 - 生产HDD盘 40CPU+12盘 (11T), SSD 40CPU + 8盘 (3.5T)
- datanode部署时, 数据盘的文件系统建议统一使用一种, 如xfs
- 针对读写并发较大的业务, 可以尝试在创建卷的时候, 多创建的dp减缓读写压力
- 4台及以上

3. metanode

- 元数据节点 (物理机部署, centos7.6)
- 虽然所有元数据全量存放在内存, 但内存也不是越大越好, 因为元数据需要定期dump到磁盘, 如果内存过大, dump的时候会消耗cpu, 导致系统负载过高。(125G)
- *cpu 配置一般 (40C), 磁盘需要足够存储本节点元数据信息(100G)
- 注意:
 - 对于小文件较多的业务会占用较多的元数据内存 (要让业务归档调整)

- 由于dentry信息跟父目录放到一起, 不建议有太大的目录 (比如超过千万上亿的)
- 4台及以上

3. 配置文件

配置文件讲解分master&data&meta三个组件进行说明

1. master

配置项	说明
cluserName	集群名字, 会在grafana的图形界面展示
id	master的编号, 必填, 3个节点的话, 就是1,2,3
retainLogs	truncate后保存的log数量, 可通过重放日志恢复数据, 默认是20000
logLevel	客户端不多的情况下, 可以设置成info级别
warnLogDir	异常日志的目录, 可以跟logdir配置到同一个地方, 日志名都是ump开头
logDir	日志路径, 使用home目录即可, 100G足够
walDir	使用home目录
storeDir	home目录, rocksdb 存储的资源管理信息的地方
metaNodeReservedMem	metaNode预留内存大小, 小于该值的话, metanode只读, 一般metanode只读都是通过threold=0.75触发的, 默认2G

2. metanode

配置项	说明
logLevel	建议采用warn
logDir	配置到任意目录下都可 (home)
warnLogDir	异常日志的目录, 可以跟logdir配置到同一个地方, 日志名都是ump开头
totalMem	限制metanode服务使用的内存量, 使用量大于totalMem*0.75的时候, 就会导致metanode只读(100G), 不建议过大
metadataDir	元数据snapshot数据 (home目录)
raftDir	raft日志目录 (home目录)

home目录由于要存储服务日志, raft日志, snap数据, 一本都是100G以上的

3. datanode

配置项	配置说明
raftDir	raft日志相关的目录 (home)
enableSmuxConnPool	最终设置为true, 修复流程走多路复用
logDir&warnLogDir	配置到任意目录下即可 (home)
logLevel	使用warn级别
disks	里面的disk路径配置到数据盘的挂载目录即可, 预留空间稍微多点, 防止并发写导致磁盘写满异常 (150G)

4. 监控告警

1. 工具

1. 监控告警使用prometheus + grafana, 网上有很多教程关于搭建的教程
 1. prometheus&grafana 都有告警的组件, 支持配置告警规则并发送短信和邮件
 2. grafana的配置, 参考chubaofs docker目录下的grafana对应的json文件即可

2. 监控及处理

1. 机器负载&cpu&内存相关的基础告警
2. 坏盘监控: `cfs_master_disk_error` 指标获取坏盘路径和ip, 然后磁盘下线的接口把会坏盘下线
3. 监控datanode总容量的使用率, 大于75%的时候, 需要及时扩容


```
sum(cfs_master_dataNodes_used_GB{ })by(cluster)/sum(cfs_master_dataNodes_total_GB{ })by(cluster)
```
4. 监控metanode使用总量, 大于75%的时候, 及时扩容 `sum(cfs_master_metaNodes_used_GB{ })by(cluster) / sum(cfs_master_metaNodes_total_GB{ })by(cluster) * 100`
5. 监控每个vol的容量使用率, 大于70%的时候, 及时调用接口进行扩容卷的容量, 否则卷写满的时候, 将会变成只读
6. 日志监控 `ump_master_business.log` 也就是关注`master_warn_count`这个指标不为0时的情况, 根据告警日志信息进行处理, 下面详细描述
7. 日志监控datanode上的 `raft_debug.log`, 根据 `snapshot concurrency exceed the limit` 过滤告警字段
 1. 出现大量该类型告警后 `Transport] 837 send snapshot to 76 failed error is: snapshot concurrency exceed the limit 10`, 可能是因为较多dp同时在迁移导致, 如果长时间未恢复的话, 可能是太多dp同时发送堵塞了, 这时可以重启下该告警机器上的 datanode服务

5. 运维操作

- 检测到坏盘后处理流程
 - 调用接口下线磁盘 -> 不停服换盘 (磁盘支持热插拔) -> 重启datanode节点
 - 若换盘需要停服:

- 若换盘动作很迅速：调用接口下线磁盘 -> 停止datanode服务 -> 换盘 -> 启动datanode服务
 - 若换盘比较慢：调用接口下线整个机器--> 停止datanode服务 --> 换盘 -> 启动datanode服务
- 机器宕机处理流程
 - 若能很快恢复，不用处理
 - 短期无法恢复：调用接口下线整个机器 --> 停止datanode服务
- 卷的创建
 - 创建卷的时候，以为owner权限较大，最好不要把卷的owner信息给到客户端，提高安全性
 - 流程如下：
 - 创建卷vol
 - 创建user
 - 将vol授权给用户
 - 把用户的ak&sk&userId和volume的名字进行挂载
- 子目录挂载
 - 如果卷里有个目录"/a/b/c"，新的客户端挂载到子目录/a/b的话，挂载上去后只能看到目录c，无法看到目录a
 - 只需在客户端配置 "subdir":"/a/b"
- 集群扩容
 - 直接启动meta&data服务即可，会自动注册到集群当中

6. 常见问题

- 单机部署
 - 建议使用docker在单机上进行集群部署
- volume下有哪些client机器
 - 这个目前上报的监控数据里没有，下个版本会加上
 - 目前可以通过解析master的info日志获取到，因为所有client都会定时的请求master接口
 - URL[/admin/getVol?name=bot-cfs],remoteAddr[ip:60490],response ok
- 大盘指标关注点
 - 集群空间是否足够
 - master_warn_count 是否为0
 - 是否有节点not active
 - 关注读写耗时是否异常
- 出现no leader
 - no leader 是因为重选进行选举leader, 一般都会在几秒内选举成功
 - 若长期选举不成功 (raft问题), 重启下dp或者mp副本集对应的节点
- kill 无法杀死程序
 - 比如meta暂用内存较多的时候，原因正在排查，使用kill -9杀死服务
- 遇到LackPartitions异常导致服务无法正常重启
 - 我们一般是哪个partition缺失，就把那个partition从对应的节点下线就能恢复了
 - 或者把缺失的副本从其他机器拷贝过来也行，相对麻烦点
 - 由于一个dp或&mp缺失两个副本后无法恢复，因此线上的运维操作一定要一台台机器的操作，确保恢复后，再操作下一台，避免出现两副本同时不可用的情况，一次建议大家升级的时候，不同节点重启也要间隔一段时间

- master的 `ump_master_business.log` 中出现 `checkFileCrcTaskErr` 错误，对比该文件的三个副本后
 - 如果三副本不同，可能是数据真的损坏，看下系统日志进行确认，可通过下线改副本所在dp进行修复
 - 如果副本相同，可能是覆盖写与校验并发导致，重启下异常的副本即可恢复
 - 通常是随机写频率较高或者有节点在根据raft-log不停commit时会遇到，下个版本会解决这个并发的问题