
curvefs chunkid 持久化

背景

1. 将原有的获取chunkid的方法从space迁入mds中，并持久化写入etcd中；
2. 只考虑单 mds 工作的情况；
3. chunkid全局递增。

实现

1. proto/space.proto 中的 message AllocateS3ChunkRequest、message AllocateS3ChunkResponse 复制到 proto/mds.proto；
2. 修改 AllocateS3ChunkResponse 的状态码的类型为 FSStatusCode；
3. proto/space.proto/service SpaceAllocService中的 rpc AllocateS3Chunk 复制到 proto/space.proto/service/MdsService中；
4. curvefs/src/mds/mds_services.h MdsServiceImp类中增加 AllocateS3Chunk 实现；
5. curvefs/src/mds/mds_services.h MdsServiceImp类中增加 ChunkIDGenerator 类对象，方法 AllocateS3Chunk 调用 ChunkIDGenerator对象的GenChunkID方法；
6. ChunkIDGenerator 类
 1. 构造函数
初始化
 2. init 函数：用于初始化或者更改 ChunkIdAllocatorImpl 的一些配置。但是这些配置不会立即生效，而是等到当前 chunkId池枯竭时才会生效。
 3. 析构函数
 4. GenChunkID
 1. 申请的chunkID池是否枯竭？
 1. 是，使用 KVStorageClient 申请新的chunkid Bundle （etcd中记录的已使用chunkID增加一个chunkID Bundle 的大小）；
 1. 申请失败
 2. storeKey_不存在，从现在已使用的 ChunkID 继续申请新的 Bundle
 1. 首次连接 etcd
 2. etcd 环境切换
 3. storeKey_ 被删除
 4. 其他
 3. storeKey_ 在etcd上的Value不可解码
 2. 返回一个新的chunkid

方案

```
class ChunkIDGenerator {  
  
    client_; // etcd  
  
    chunkIdStoreKey_; // chunkid 存储在etcd上的 key  
  
    nextId_; // 下一个可分配的 chunkID  
  
    lastId_; // 当前chunkID bundle 内最后一个可分配的chunkID  
  
    bundleSize_; // chunkId池子的大小  
  
};
```

问题与风险

1. 构造函数内判断 `storeKey_` 不存在时，会从0开始分配，可能会出现覆盖chunkid的情况；
2. chunkID用完情况没有考虑；