curvefs s3数据整理(合并碎片、清理冗余)

背景

1. 只考虑单客户端，单metaserver
2. 为了解决的问题：客户端在对一个文件的某个部分多次写入后，同一个chunk会产生很多版本数据；而客户端在读的时候，会需要对这些chunk进行筛选和构建，得到有效的部分，越是散乱的状态，就越需要发送更多次读请求至s3. 最后导致无效旧数据的堆积和读请求性能的下降，所以需要在合适的时候进行重叠元数据和数据的合并
3. 原则是尽力而为，并不能做到完美

方案

基于一下3个基础的数据结构，2层索引

```
s3chuninfolist[index] = [s3chunkinfo(s)]

s3chunkinfo {

        chunkid

        version // write always 0, compact will increase it

        offset

        len

}
```

s3 object命名：chunkid_version_index（index为obj在chunk内的index）

执行步骤

1. 数据整理作为一个后台服务(线程池)，运行于metaserver，遍历metaserver的inode进行数据整理的尝试，入队inodekey，如果是已有inode任务，enqueue直接返回，不入队
2. 任务开始执行，尝试根据inodekey获取inode信息，获取不到就退出；不是s3类型的inode退出
3. 对于每一个s3类型的inode来说，对每一个index内的chunkinfo按照chunkid升序排序.
4. 对于一个chunk来说，chunkinfo数量大于20即进行处理
5. 计算变更
   - 记录整个chunk最大的chunkid
   - 读出一个chunk所有有效的部分（如果是最后一个chunk，需要注意不超出len），compaction+1，chunkid为上一步获取的chunkid，为需要新增的obj
   - 老的obj为全部需要删除的部分
6. 应用变更
   - 先读写新增的s3 objects列表，由于新增了version字段，不会涉及到覆盖老的对象
   - 加锁，增量的更新inode的s3chunkinfolist，保证原子更新，更新失败回退新增数据
   - 等待N秒，保证mds已经告知client缓存失效，需要更新为新的s3chunkinfolist // 需不需要这个步骤@xuchaojie @chenwei确认
   - 删除老的object

问题与风险

1. 在执行变更时，在bcd步挂掉时，会造成s3数据的残留
2. 当同时有多个变更inode元数据(s3chunkinfolist)的动作时，目前的updateinode的实现是直接的覆盖，如果数据整理和client写同时进行了同一个inode的变更，总有一个变更会丢失，

需要进行一个merge的步骤

在做变更时如果有其他op可能会产生的冲突:

读: 在执行变更删除原来的s3 object时, 执行读的客户端的缓存可能还是原有的chunkinfolist, 可能会去读已经删除的object, 这种时候读会失败

可以使用双重保证

1. 读失败的时候retry, 或许可以重拉metadata
2. 整理后, mds在一个时间间隔内主动告知client这个inode元数据缓存失效, 重拉

写: 只是对chunkinfolist做新增, 不影响整理对原有部分的变更

删除: 已标记为删除的inode不进行整理, 已经在整理的任务不会被新的删除标记的请求打断. 如果标记删除到实际删除之间的时间间隔非常短, 并且在标记删除前已经开始了整理任务, 可能会出现边删除边整理的状态(出现概率较小)

1. 可以在实际删除前检查当前整理的inode列表, 如果在列表里就暂时跳过(同步删除)/重新丢进删除队列(异步删除)
2. 或者就不管, 处理一下报错, 让后续的应该会开发的数据清理工具来删除, 因为出现这个冲突的概率比较小

truncate: 只进行元数据里len的改变, 触发一下compact就行, shrink的部分compact会进行处理