DeleteStorageGroup流程修改

1 目的

当前的DeleteStorageGroup流程中，需要经过三个步骤。

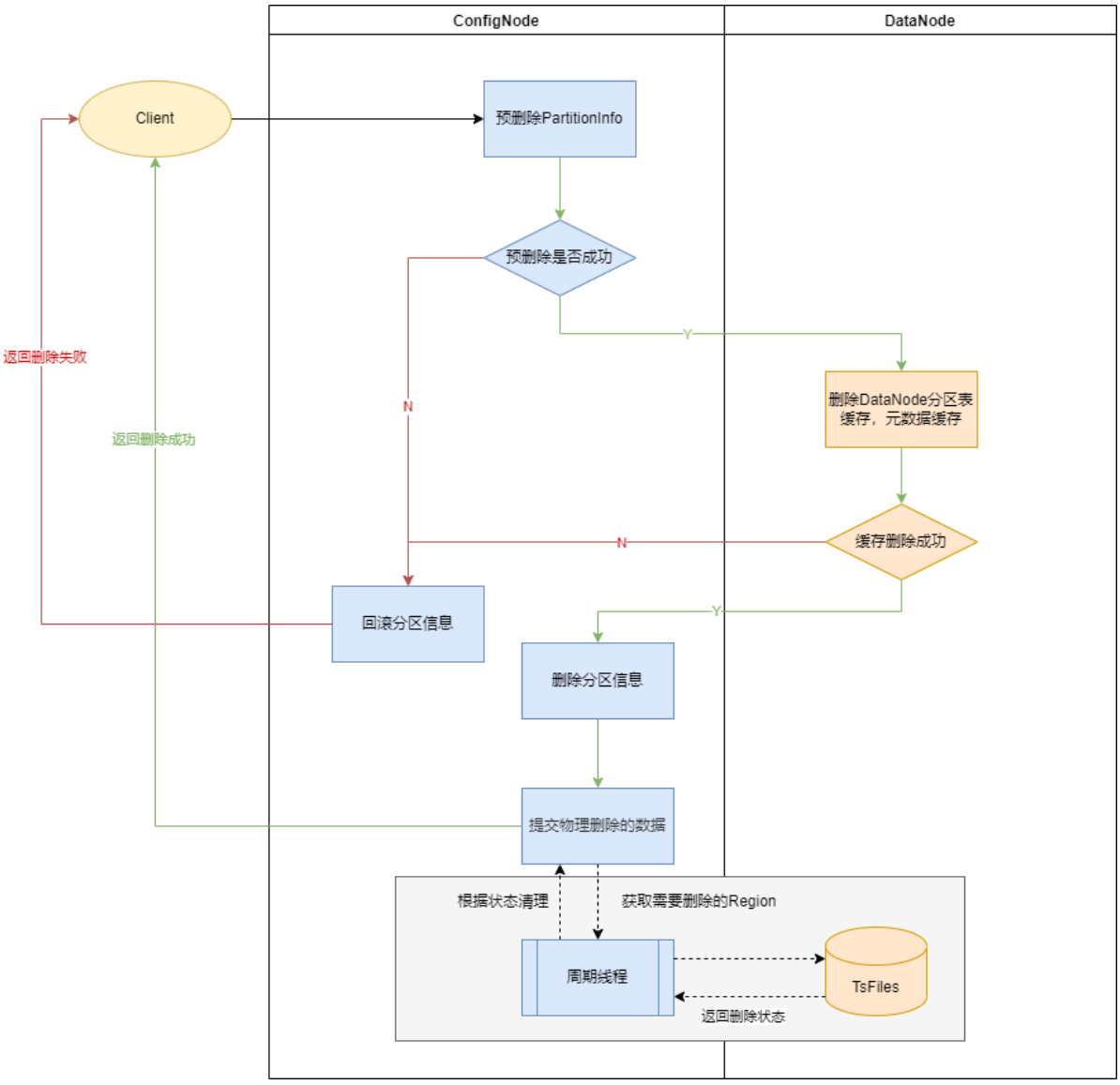
1. 删除PartitionRegion上的分区信息
2. 根据分区信息，删除SchemaRegion上的元数据信息，并清理缓存
3. 根据分区信息，删除DataRegion上的数据文件与目录

步骤3删除文件与目录，耗时长且容易发生异常，由此容易发生删除失败的现象。

这里讨论一种逻辑删除的方案，替代直接物理删除。将删除分为逻辑删除与物理删除，逻辑删除屏蔽存储组的读写，物理删除即删除文件的过程转为异步执行。简化删除RPC的操作，降低超时与异常现象发生的概率。

2 流程

当由于数据的读写，需要通过分区表进行寻址。当分区表信息删除，分区缓存被清理之后，同名存储组的读写，将在新的region下进行。所以，只需要对旧的region，进行读写屏蔽即可实现逻辑上的删除。



PartitionInfo上，新增两个数据结构

Map<String，RegionReplicaSet> preDeletedRegionMap，用于进行预删除， 内容sg->region映射

Map<RegionId, DataNodeLocations> deletedRegionMap 用于存放逻辑删除成功，需要进行物理删除的region

一个周期线程

ScheduledThread deleteRegionThread，轮询deletedRegionMap ，发送物理删除rpc，删除成功则删除deletedRegionMap对应entry

逻辑删除阶段

阶段一
在partitionRegion上对存储组关联的分区信息进行逻辑删除，即新增至preDeletedRegionMap中。拉取RPC接口中，对preDeletedRegionMap中的存储组进行过滤。

阶段二
清理datanode上partition缓存，schema缓存，对于阶段一中预删除的存储组，不再拉取缓存。

阶段三
如果阶段二成功，将partitionRegion上的storageGroup相关的分区信息删除（regionMap，partitionTableMap，mTree），将preDeletedRegionMap转移到deletedRegionMap ，返回删除成功。
如果阶段二失败，执行回滚操作，清空预删除preDeletedRegionMap，返回删除失败。

物理删除阶段

deleteRegionThread定时地，遍历deletedRegionMap向dataNode，提交删除rpc，进行物理删除，删除成功，则移除对应entry。

3 对读写影响

DDL：

由于confignode上缓存mtree中仍然存在该confignode，在做setStorageGroup的时候，会报存储组已存在。

对读写的影响

写入：

删除阶段二完成之后，在校验元数据的时候，需要重新拉取分区表与元数据，可以拒绝写入。

查询：

重新拉取分区表后，MPP在生成执行计划时，不会分配到预删除的dataRegion。

4 故障恢复

1. 逻辑删除过程中，ConfigNode发生Leader切换

   当前客户端将返回超时，通过Procedure恢复机制，从磁盘中恢复Procedure，重新执行。

   后续可以优化为轮询查询Procedure状态

2. 物理删除过程中，DataNode发生故障

   返回超时，等待下一次物理删除

5 优缺点

优点：

删除存储组不再涉及磁盘操作，RPC出现超时或I/O异常的可能性降低。

异步删除的机制，相对更安全。对于误删场景，可以提供一些补救措施。

缺点：

没有即时释放空间。

需要对读写接口进行修改。