

TiKV 掉电数据恢复指南



TiKV 数据丢失风险

- 没开 sync log, 掉电
 - 最近的写入没落盘
 - 掉电机器数量超过副本数的一半
 - 部分 region 最近(或全部)的写入会丢失
- 2.1.12 之前的版本, 掉电(无论是否开启 sync-log)
 - ingest 的 snapshot 可能损坏, TiKV 无法启动
 - 出问题的机器数量超过副本数的一半
 - 部分 region 最近(或全部)的写入会丢失
- “最近” 如何定义
 - 理论上: 最落后的副本落后于 leader 多少
- 影响评估
 - 从 region 映射到 表、业务

OnCall 数据恢复步骤

- 需要记录的东西
 - 集群版本、sync-log 配置、失败 TiKV 数量
 - 过半副本在这些失败的 TiKV 上的 region
 - `pd-ctl -u <endpoint> -d region --jq=".regions[] | {id: .id, peer_stores: [.peers[] | select(length as $total | map(if .==(1,4,5) then . else empty end) | length>=$total-length) }"`
 - 无法启动的 TiKV 的数量及其日志
- 需要与用户沟通的内容
 - 服务恢复时间限制
 - 有数据丢失风险的情况, 询问是否能够容忍
 - 丢失部分最近的写入
 - 丢失部分 region 的全部数据
 - 是否可以重导丢失的部分数据
- 需要与 TiKV oncall 同学沟通的内容
 - 建立 jira, 收集足够信息
 - 整理恢复步骤, 确定备份方案, 并互相 review

把无法启动的 TiKV 拉起来

- Rocksdb 数据损坏
 - 2.1.12 前的版本 ingest 之后 snapshot 文件因为掉电损坏
 - 实验复现、演练
 - 启动一个 tikv, 插入少量数据
 - truncate /path/to/tikv-data/db/some.sst --size 0

```
[2019/06/20 17:00:52.320 +08:00] [FATAL] [server.rs:176] ["failed to create kv engine: RocksDb Corruption: Sst file size mismatch: /home/qupeng/var-lib/local/tikv1/db/000012.sst. Size recorded in manifest 931, actual size 0\n"]
```

- ldb 修复 [工具](#) 及命令
 - `ldb --db=db remove_sst_file 000012.sst`
 - 修复之后, 需要 tombstone sst 文件对应的 region
- 预期效果
 - TiKV 可以正常启动, 但是可能丢失部分数据
 - 遗留问题: 如何看 sst 文件对应哪个 region?

把无法启动的 TiKV 拉起来

- Raft 元数据损坏
 - 最新 tikv-ctl [下载地址](#)
 - 找出 Raft 损坏的所有 region
 - `tikv-ctl --db /path/to/tikv-data/db bad-regions`
 - 不超过副本数一半的 TiKV 无法启动
 - `pd-ctl remove-peer <region-id> <store-id>`
 - `tikv-ctl --db /path/to/tikv-data/db tombstone -r <region-id> --pd <endpoint>`
 - 超过副本数一半的 TiKV 无法启动, 并且知道是哪些 region
 - `tikv-ctl --db /path/to/tikv-data/db tombstone -r <region-id> --force`
 - 超过副本数一半的 TiKV 无法启动, 并且**不知道**是哪些 region
 - 放弃抢救这些 TiKV
 - 记录哪些 region 有过半副本在这些 TiKV 上
- 操作预期结果
 - 所有 TiKV 可以正常启动

处理不可服务的 region

- 过半副本不可用, 选不出 leader, 不可用可能的原因
 - 过半副本坏掉了, 且在上一步放弃了所在的整个 store
 - 过半副本坏掉了, 且在上一步 tombstone 掉了
- 处理办法: 在剩余所有 store 上执行 `unsafe-recover remove-fail-stores`
 - 对个别 region
 - `tikv-ctl --db /path/to/tikv-data/db unsafe-recover remove-fail-stores -s <s1,s2> -r <r1,r2,r3>`
 - 对放弃整个 store 的情况
 - `tikv-ctl --db /path/to/tikv-data/db unsafe-recover remove-fail-stores -s <s1,s2> --all-regions`
- 操作预期结果
 - 所有操作的 region 都可以选出 leader
 - 这些 region 可能丢失部分最近的写入

处理完全丢失的 region

- 所有副本都不可用, 数据 彻底丢失, 影响
 - 部分查询可能报 Region Unavailable
 - TiDB 可能无法服务
- 根据 region 信息查看属于哪部分数据
 - `curl http://{TiDBIP}:10080/regions/{regionID}`
 - 一般业务表的记录或索引
 - TiDB 重要元数据
- 重建空 Region 解决 Unavailable 的报错
 - `tikv-ctl --db /path/to/tikv-data/db recreate-region --pd <endpoint> -r <region_id>`
- 操作预期结果
 - TiDB 可以启动并服务, 没有 Region Unavailable 错误
 - 涉及的 region 全部数据丢失, 需要考虑补数据

处理数据丢失

- 需要处理的情况
 - 过半副本掉过电 Region (没开 sync log)
 - 过半副本掉过电且不可用了的 Region (开了 sync log)
- 根据 region ID 查询对应的表
- 对于索引: 检查数据索引一致性
 - `select count(*) from table`
 - `select count(*) from table force index `idx_name``
- 对于记录: 重新导入这部分数据

TODO

- 在 schrodinger 上创建有问题的数据进行演练
- TiDB 重要元数据备份
- 从 SST 文件找对应 region

Thank You !

