

结果如下:

```
19/09/19 09:53:10 WARN util.HBaseFsock: No HDFS region dir found: { meta =>
XINYAN_TECH:fi_gw_express_order_idcard1,1b1e73ed71329daa72ffb28f42947ee7d40c9cb7872063682cb511fd16dbbfd1201810277ccb
hdfs => null, deployed => , replicald => 0 } meta={ENCODED => 1513a155f24ac2df20b4797077ed6ae0, NAME =>
'XINYAN_TECH:fi_gw_express_order_idcard1,1b1e73ed71329daa72ffb28f42947ee7d40c9cb7872063682cb511fd16dbbfd1201810277cct
STARTKEY =>
'1b1e73ed71329daa72ffb28f42947ee7d40c9cb7872063682cb511fd16dbbfd1201810277ccbfc2577361956656a23f805afb859', ENDKEY
=> '1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e69052f705b07331e32c00b5aae'}
19/09/19 09:53:10 WARN util.HBaseFsock: No HDFS region dir found: { meta =>
XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e
hdfs => null, deployed => , replicald => 0 } meta={ENCODED => d4a172571b229e74109df6c03a959588, NAME =>
'XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0
STARTKEY =>
'1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e69052f705b07331e32c00b5aae', ENDKEY
=> '1b50b073e140ffd3b3bd9eca2bd7a1db27eb757a06dee480f2ebd537ca0f44d201812187ab3f7d880e20053ee071eab7d0d5360'}
2 inconsistencies detected.
Status: INCONSISTENT
```

检测到2个不一致信息。

此外, 在hbase web ui中, 该表的region块存在 **所有数值都为0的异常情况**。(由 @杨坤焱 @丁文权 提供):



可以确认是该表的region数据发生异常, 且日志中异常的region信息如下:

- XINYAN_TECH:fi_gw_express_order_idcard1,1b1e73ed71329daa72ffb28f42947ee7d40c9cb7872063682cb511fd16dbbfd1201810277ccbfc2577361956656a23f805afb859,1549491783878.1513a155f24ac2df20b4797077ed6ae0.
- XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e69052f705b07331e32c00b5aae,1549491783878.d4a172571b229e74109df6c03a959588.

此信息为 hbase:meta 表中的rowkey, 根据meta表rowkey的组成规则, 取rowkey中最后一段的encode编码信息, 到hdfs上验证该region数据目录是否存在:

```
hadoop fs -ls /hbase/data/XINYAN_TECH/fi_gw_express_order_idcard1/1513a155f24ac2df20b4797077ed6ae0
ls: `/hbase/data/XINYAN_TECH/fi_gw_express_order_idcard1/1513a155f24ac2df20b4797077ed6ae0': No such file or directory
```

可以看到, 该region在hdfs上对应的数据目录消失了。

进入 hbase shell 中查询meta表异常rowkey对应的值:

```
get 'hbase:meta','XINYAN_TECH:fi_gw_express_order_idcard1,1b1e73ed71329daa72ffb28f42947ee7d40c9cb7872063682cb511fd16dbbfd1
# 结果如下:
info:regioninfo timestamp=1566891946444, value={ENCODED => 96dd94004c9dd9fce3f4eb80c885ad85, NAME => 'XINYAN_TECH:fi_gw_ex
ENDKEY => '0857
87e9323caa99fc45325a351797fdd4849891167b36bfc6241197b5a58cb1201802142fd6fdc3bf0b5b9cb8faf80d04b71d1a'}
info:seqnumDuringOpen timestamp=1566891946444, value=\x00\x00\x00\x00\x00\x00\x02'
info:server timestamp=1566891946444, value=cdh85-104:16020
info:serverstartcode timestamp=1566891946444, value=1563956428836
info:sn timestamp=1566891945886, value=cdh85-104,16020,1563956428836
info:state timestamp=1566891946444, value=OPEN
```

元数据信息正常, 至此, 可以确认问题原因:

元数据显示该region在正常提供服务中, 客户端到具体节点上检索数据时发现该region的数据目录不存在, 抛出异常。

附: meta表结构

hbase:meta表结构如下:

列	释义
rowkey	\$(表名),\$(起始键),\$(region时间戳).\$(encode编码).
info:state	Region状态, 正常情况下为 OPEN

info:serverstartcode	RegionServer启动的13位时间戳
info:server	所在RegionServer 地址和端口, 如cdh85-47:16020
info:sn	server和serverstartcode组成, 如cdh85-47:16020,1549491783878
info:sequenceDurationDurableOpen	Region在线时长的二进制串
info:regioninfo	region的详细信息, 如: ENCODED、NAME、STARTKEY、ENDKEY等

其中, regioninfo是重要信息:

- ENCODED: 基于\${表名},\${起始键},\${region时间戳}生成的32位md5字符串, region数据存储在hdfs上时使用的唯一编号, 可以从meta表中根据该值定位到hdfs中的具体路径。rowkey中最后的\${encode编码}就是 ENCODED 的值, 其是rowkey组成的一部分。
- NAME: 与ROWKEY值相同
- STARTKEY: 该region的起始键
- ENDKEY: 该region的结束键

ENCODED 生成方式验证:

```
// rowkey:
XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905
import com.xinyan.bigdata.security._
// 截取rowkey中 ${表名},${起始键},${region时间戳} 的部分
println(MD5Util.encrypt32("XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905"))
// 结果为: d4a172571b229e74109df6c03a959588, rowkey中的最后一部分
```

修复过程

使用元数据修复工具

尝试直接使用命令修复:

```
hbase hbck -repair XINYAN_TECH:fi_gw_express_order_idcard1
hbase hbck -fixMeta
```

执行失败: HBase 2.0 中已经不支持元数据修复工具, 只支持「查看」类型的操作, 不支持「读入」类型的操作。

使用hdfs工具检查是否有文件块异常:

```
hdfs fsck /hbase
# 结果正常
```

手动修复元数据

备份原有的region信息:

```
get 'hbase:meta', 'XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905'
# 备份数据
# 删除该region信息
delete 'hbase:meta', 'XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905'
delete 'hbase:meta', 'XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905'
delete 'hbase:meta', 'XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905'
delete 'hbase:meta', 'XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905'
delete 'hbase:meta', 'XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905'
delete 'hbase:meta', 'XINYAN_TECH:fi_gw_express_order_idcard1,1b37fe2afa75581bf5f7574336f1cdfbf0f902966db54bf38aa505665828929020181207edad0e6905'
gOpen'
```

刷新hbase web ui页面, 发现 d4a172571b229e74109df6c03a959588 region 已经消失。

执行hbck -summary 发现 不一致的部分由2变成了1。

尝试:

- hbase shell中手动scan
- 通过接口重新查询该值

可以成功获得结果。

继续删除 1513a155f24ac2df20b4797077ed6ae0 region块对应的meta数据后, 重新测试数据样本, 可得到正确的结果。

问题分析

- d4a172571b229e74109df6c03a959588
- 1513a155f24ac2df20b4797077ed6ae0

以上两个reigon在split时，子region的数据块已经并在meta表中更新上线提供正常服务，父region的数据块已删除，但是 **meta表中没有更新对应的元数据信息**（原因仍待排查）。

导致对应的数据查询时，仍然通过父region检索数据，但是父region的数据已被删除，故无法成功检索。

可以通过以下命令列出该表在meta表中所有的region信息，**分析排查是否有相关的region范围 覆盖了有问题的region数据**：

```
echo "scan 'hbase:meta',{FILTER => org.apache.hadoop.hbase.filter.PrefixFilter.new(org.apache.hadoop.hbase.util.Bytes.toBy  
| sort | uniq
```

来判断消失的region块是否已经由其他region托管服务。

hbase:meta 表是检查和修复问题的关键。