HBase 疑难杂症诊治

hbase2.0处理rit状态记录

日期	**版本号**	**类别**	**描述**
2019-07-05	1.0.0	A	排查hbase2.0的rit问题

问题说明

由于使用HDP3.0，HDP3.0使用的是hbase2.0.0版本，hbase的ui页面发现很多表出现了rit，删除表过程中，region的状态卡在了opening。先尝试使用hbck2工具进行修复，发现在hbase2.0的master的rpc方法中没有hbck2中的bypass，assign方法，通过源码发现，hbck2的rpc方法是在hbase2.0.2中才增加的，所以只能尝试手动处理：

一.hdfs上已经没有对应目录，meta里没有对应表状态信息，存在有对应的分区信息

1. 检查表状态

```
get  'hbase:meta','KYLIN_0054K9NLSU','table:state'
```

结果为空

2. 通过源码发现表状态

```
ENABLED,对应meta表里的值\x80\x00
DISABLED,  对应meta表里的值\x80\x01
DISABLING,  对应meta表里的值\x80\x02
ENABLING,  对应meta表里的值\x80\x03
```

3. 查找分区信息

```
scan                                                                                          'hbase:meta',
{FILTER  =>  org.apache.hadoop.hbase.filter.PrefixFilter.new(org.apache.hadoop.hbase.util.Bytes.toBytes('KYLIN_0054K9NLSU'
```

发现存在有分区记录

4. 手动修改表状态或者删除分区信息

```
put  'hbase:meta','KYLIN_0054K9NLSU','table:state','\x80\x01'
```

或者deleteall 表对应的分区信息，修改后重启hbase，发现rit状态消失

二.hdfs上已经有对应目录，meta里有对应表状态信息和分区信息

1. 确认一下表的信息和数据

```
hbase  hbck  -summary  TableName
```

2. 检查表状态

```
get  'hbase:meta','KYLIN_0354K9NLSU','table:state'
```

meta表里的值\x80\x02，表的状态为DISABLING

3. 找出异常的region

```
echo                                                    "scan                                                    'hbase:meta',
{FILTER  =>  org.apache.hadoop.hbase.filter.PrefixFilter.new(org.apache.hadoop.hbase.util.Bytes.toBytes('KYLIN_0354K9NLSU'
n|grep  OPENING
```

找出异常的region

4. 将region信息更新为CLOSED状态

```
put  'hbase:meta','KYLIN_0354K9NLSU,,1561953520536.30b7d24eaa3209c6e5e8de764ad04855.','info:state','CLOSED',1562117738678
```

·········

5. 将表状态更新为disable

```
put  'hbase:meta','KYLIN_0354K9NLSU','table:state',"\x08\x01",1562120793251
```

重启hbase后rit消失

存在问题

- rit是删除表的时候出现，所以表中的数据可以忽略，上述操作也是表中没有数据时操作
- 如果是生成集群，已经存在的数据比较多，不建议直接重启，可以通过切换master的方式
- 可以使用HDP3.1.1，里面hbase版本是2.0.2，可以使用hbck2操作
- 使用hbck2的方法的话，修改meta状态后还会同步改zookeeper状态，能避免状态不一致

HBase2.x之RIT问题解决

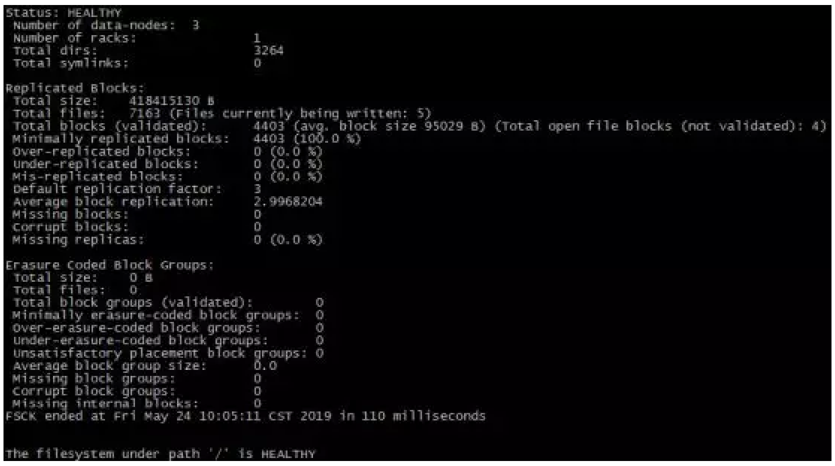
问题描述

Region-In-Trasition机制：从字面意思来看，Region-In-Transition说的是Region变迁机制，实际上是指在一次特定操作行为中Region状态的变迁，例如merge、split、assign、unssign等操作。RIT问题指的是在RIT过程中出现异常情况，然后导致region的状态一直保持在RIT，使得HBase出现异常。

解决方案

方案一

检查hdfs的健康度，是否有hbase的文件丢失或损坏，运行命令hadoop fsck /，结果如下：

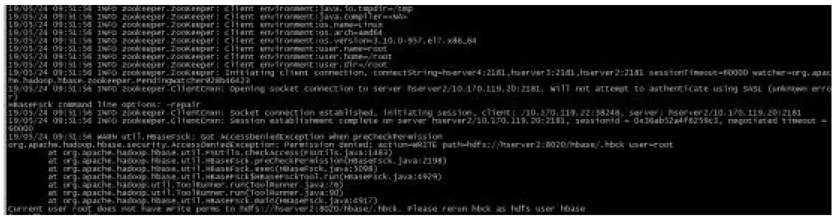


```
Status: HEALTHY
 Number of data-nodes:  3
 Number of racks:               1
 Total dirs:                    3264
 Total symlinks:                0

Replicated Blocks:
 Total size:    418415130 B
 Total files:   7163 (Files currently being written: 5)
 Total blocks (validated):      4403 (avg. block size 95029 B) (Total open file blocks (not validated): 4)
 Minimally replicated blocks:   4403 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    3
 Average block replication:     2.9968204
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)

Erasure Coded Block Groups:
 Total size:    0 B
 Total files:   0
 Total block groups (validated):        0
 Minimally erasure-coded block groups:  0
 Over-erasure-coded block groups:       0
 Under-erasure-coded block groups:      0
 Unsatisfactory placement block groups: 0
 Average block group size:      0.0
 Missing block groups:          0
 Corrupt block groups:          0
 Missing internal blocks:       0
FSCK ended at Fri May 24 10:05:11 CST 2019 in 110 milliseconds

The filesystem under path '/' is HEALTHY
```

排除hdfs丢失block的问题。如果出现hdfs的block损坏或丢失的情况，可以通过hdfs的修复命令进行修复。

方案二

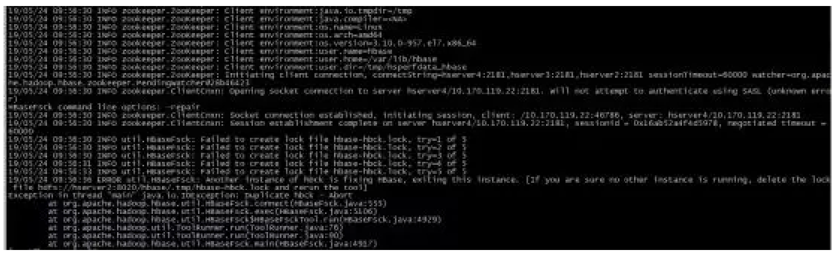
在HBase1.x系列中RIT问题通常可以通过hbase　hbck　－repair操作完成修复。但是在HBase2.x系列中，该命令还没有支持，所以暂时无法通过这种命令完成修复。结果如下：



```
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Client environment:java.io.tmpdir=/tmp
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Client environment:java.compiler=<NA>
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Client environment:os.name=Linux
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Client environment:os.arch=amd64
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Client environment:os.version=3.10.0-957.el7.x86_64
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Client environment:user.name=root
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Client environment:user.home=/root
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Client environment:user.dir=/root
19/05/24 09:51:56 INFO zookeeper.ZooKeeper: Initiating client connection, connectString=hserver4:2181,hserver3:2181,hserver2:2181 sessionTimeout=60000 watcher=org.apac
he.hadoop.hbase.zookeeper.PendingWatcher@28b4623
19/05/24 09:51:56 INFO zookeeper.ClientCnxn: Opening socket connection to server hserver2/10.170.119.20:2181. Will not attempt to authenticate using SASL (unknown erro
r)
HBaseFsck command line options: -repair
19/05/24 09:51:56 INFO zookeeper.ClientCnxn: Socket connection established, initiating session, client: /10.170.119.22:58248, server: hserver2/10.170.119.20:2181
19/05/24 09:51:56 INFO zookeeper.ClientCnxn: Session establishment complete on server hserver2/10.170.119.20:2181, sessionid = 0x36ab52a4f6258c3, negotiated timeout =
60000
19/05/24 09:51:56 WARN util.HBaseFsck: Got AccessDeniedException when preCheckPermission
org.apache.hadoop.hbase.security.AccessDeniedException: Permission denied: action=WRITE path=hdfs://hserver2:8020/hbase/.hbck user=root
        at org.apache.hadoop.hbase.util.FSUtils.checkAccess(FSUtils.java:1483)
        at org.apache.hadoop.hbase.util.HBaseFsck.preCheckPermission(HBaseFsck.java:2198)
        at org.apache.hadoop.hbase.util.HBaseFsck.exec(HBaseFsck.java:5098)
        at org.apache.hadoop.hbase.util.HBaseFsck$HBaseFsckTool.run(HBaseFsck.java:4929)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:90)
        at org.apache.hadoop.hbase.util.HBaseFsck.main(HBaseFsck.java:4917)
Current user root does not have write perms to hdfs://hserver2:8020/hbase/.hbck. Please rerun hbck as hdfs user hbase
```

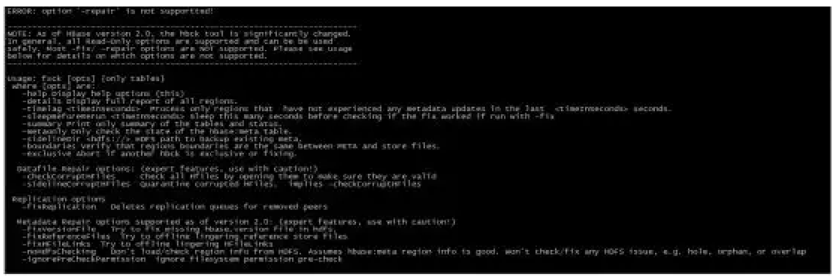
第一次执行发现没有权限，root用户不是hdfs的超级用户，安装提示需要以hbase用户去执行该命令。修改如下：

```
su　hbase　-s　/bin/sh　-c　"hbase　hbck　-repair"
```



```
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Client environment:java.io.tmpdir=/tmp
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Client environment:java.compiler=<NA>
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Client environment:os.name=Linux
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Client environment:os.arch=amd64
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Client environment:os.version=3.10.0-957.el7.x86_64
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Client environment:user.name=hbase
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Client environment:user.home=/var/lib/hbase
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Client environment:user.dir=/tmp/hsperfdata_hbase
19/05/24 09:58:30 INFO zookeeper.ZooKeeper: Initiating client connection, connectString=hserver4:2181,hserver3:2181,hserver2:2181 sessionTimeout=60000 watcher=org.apac
he.hadoop.hbase.zookeeper.PendingWatcher@28b4623
19/05/24 09:58:30 INFO zookeeper.ClientCnxn: Opening socket connection to server hserver4/10.170.119.22:2181. Will not attempt to authenticate using SASL (unknown erro
r)
HBaseFsck command line options: -repair
19/05/24 09:58:30 INFO zookeeper.ClientCnxn: Socket connection established, initiating session, client: /10.170.119.22:46786, server: hserver4/10.170.119.22:2181
19/05/24 09:58:30 INFO zookeeper.ClientCnxn: Session establishment complete on server hserver4/10.170.119.22:2181, sessionid = 0x16ab52a4f4d5978, negotiated timeout =
60000
19/05/24 09:58:30 INFO util.HBaseFsck: Failed to create lock file hbase-hbck.lock, try=1 of 5
19/05/24 09:58:30 INFO util.HBaseFsck: Failed to create lock file hbase-hbck.lock, try=2 of 5
19/05/24 09:58:30 INFO util.HBaseFsck: Failed to create lock file hbase-hbck.lock, try=3 of 5
19/05/24 09:58:31 INFO util.HBaseFsck: Failed to create lock file hbase-hbck.lock, try=4 of 5
19/05/24 09:58:33 INFO util.HBaseFsck: Failed to create lock file hbase-hbck.lock, try=5 of 5
19/05/24 09:58:36 ERROR util.HBaseFsck: Another instance of hbck is fixing HBase, exiting this instance. [If you are sure no other instance is running, delete the lock
 file hdfs://hserver2:8020/hbase/.tmp/hbase-hbck.lock and rerun the tool]
Exception in thread "main" java.io.IOException: Duplicate hbck - Abort
        at org.apache.hadoop.hbase.util.HBaseFsck.connect(HBaseFsck.java:555)
        at org.apache.hadoop.hbase.util.HBaseFsck.exec(HBaseFsck.java:5106)
        at org.apache.hadoop.hbase.util.HBaseFsck$HBaseFsckTool.run(HBaseFsck.java:4929)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:76)
        at org.apache.hadoop.util.ToolRunner.run(ToolRunner.java:90)
        at org.apache.hadoop.hbase.util.HBaseFsck.main(HBaseFsck.java:4917)
```

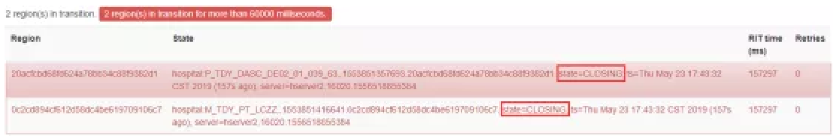
提示为hbase有其他线程正在执行hbck fix命令，但是其实没有这种命令，其实从这里就可以看出HBase2.x对于-repair的支持是不够的。我按照提示删除了hdfs(/hbase/.tmp/)上的hbase-hbck.lock文件，重新运行命令如下：



```
ERROR: option '-repair' is not supported!

-----------------------------------------------------------------------
NOTE: As of HBase version 2.0, the hbck tool is significantly changed.
In general, all Read-Only options are supported and can be be used
safely. Most -fix/ -repair options are NOT supported. Please see usage
below for details on which options are not supported.
-----------------------------------------------------------------------

Usage: fsck [opts] {only tables}
 where [opts] are:
   -help Display help options (this)
   -details Display full report of all regions.
   -timelag <timeInSeconds>  Process only regions that  have not experienced any metadata updates in the last  <timeInSeconds> seconds.
   -sleepBeforeRerun <timeInSeconds> Sleep this many seconds before checking if the fix worked if run with -fix
   -summary Print only summary of the tables and status.
   -metaonly Only check the state of the hbase:meta table.
   -sidelineDir <hdfs://> HDFS path to backup existing meta.
   -boundaries Verify that regions boundaries are the same between META and store files.
   -exclusive Abort if another hbck is exclusive or fixing.

  Datafile Repair options: (expert features, use with caution!)
   -checkCorruptHFiles     Check all Hfiles by opening them to make sure they are valid
   -sidelineCorruptHFiles  Quarantine corrupted HFiles.  implies -checkCorruptHFiles

 Replication options
   -fixReplication   Deletes replication queues for removed peers

  Metadata Repair options supported as of version 2.0: (expert features, use with caution!)
   -fixVersionFile   Try to fix missing hbase.version file in hdfs.
   -fixReferenceFiles  Try to offline lingering reference store files
   -fixHFileLinks  Try to offline lingering HFileLinks
   -noHdfsChecking   Don't load/check region info from HDFS. Assumes hbase:meta region info is good. Won't check/fix any HDFS issue, e.g. hole, orphan, or overlap
   -ignorePreCheckPermission  ignore filesystem permission pre-check
```

方案三

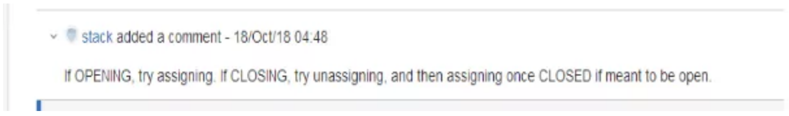
根据RIT状态执行assign或者unassign命令，状态信息如下：



2 region(s) in transition. 2 region(s) in transition for more than 60000 milliseconds.

Region	State	RIT time (ms)	Retries
20acfcbd68fd624a78bb34c88f9382d1	hospital:P_TDY_DASC_DE02_01_039_63,,1553851357693.20acfcbd68fd624a78bb34c88f9382d1. state=CLOSING, ts=Thu May 23 17:43:32 CST 2019 (157s ago), server=hserver2,16020,1556518655384	157297	0
0c2cd894cf612d58dc4be619709106c7	hospital:M_TDY_PT_LCZZ,,1553851416641.0c2cd894cf612d58dc4be619709106c7. state=CLOSING, ts=Thu May 23 17:43:32 CST 2019 (157s ago), server=hserver2,16020,1556518655384	157297	0

经过查询资料，解决方案如下：



stack added a comment - 18/Oct/18 04:48

If OPENING, try assigning. If CLOSING, try unassigning, and then assigning once CLOSED if meant to be open.

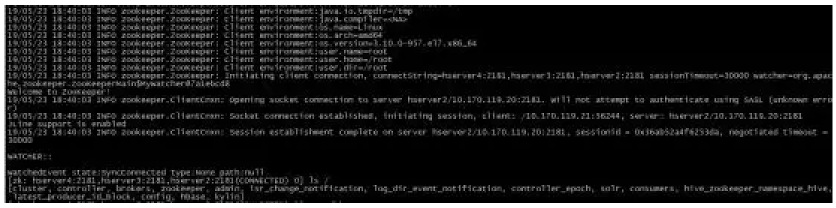
hbase shell多次执行unassign '20acfcbd68fd624a78bb34c88f9382d1'和unassign '20acfcbd68fd624a78bb34c88f9382d1', true都超时结束，通过修改rpc和zk的超时时间都无法完成(正常超时时间为60000ms，修改为600000ms)。

方案四

经过多次试验，最终都无法使得HBase回复正常，最终决定删除进行测试。

Zookeeper节点删除：

通过hbase zkcli命令进入Zookeeper命令行界面：



```
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Client environment:java.io.tmpdir=/tmp
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Client environment:java.compiler=<NA>
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Client environment:os.name=Linux
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Client environment:os.arch=amd64
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Client environment:os.version=3.10.0-957.el7.x86_64
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Client environment:user.name=root
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Client environment:user.home=/root
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Client environment:user.dir=/root
19/05/23 18:40:03 INFO zookeeper.ZooKeeper: Initiating client connection, connectString=hserver4:2181,hserver3:2181,hserver2:2181 sessionTimeout=30000 watcher=org.apac
he.zookeeper.ZooKeeperMain$MyWatcher@7a1ebcd8
Welcome to ZooKeeper!
19/05/23 18:40:03 INFO zookeeper.ClientCnxn: Opening socket connection to server hserver2/10.170.119.20:2181. Will not attempt to authenticate using SASL (unknown erro
r)
19/05/23 18:40:03 INFO zookeeper.ClientCnxn: Socket connection established, initiating session, client: /10.170.119.21:56244, server: hserver2/10.170.119.20:2181
JLine support is enabled
19/05/23 18:40:03 INFO zookeeper.ClientCnxn: Session establishment complete on server hserver2/10.170.119.20:2181, sessionid = 0x16ab52a4f6233da, negotiated timeout =
30000

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
[zk: hserver4:2181,hserver3:2181,hserver2:2181(CONNECTED) 0] ls /
[cluster, controller, brokers, zookeeper, admin, isr_change_notification, log_dir_event_notification, controller_epoch, solr, consumers, hive_zookeeper_namespace_hive,
 latest_producer_id_block, config, hbase, kylin]
```

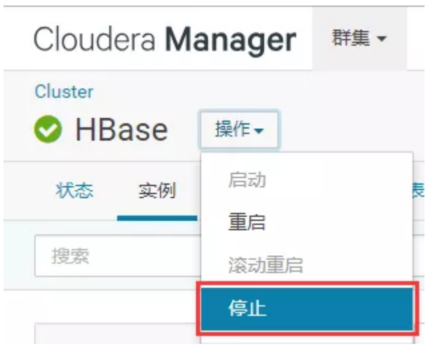
我想删除节点 /hbase-unsecure/region-in-transition，但是发现并没有该节点，经过资料查询了解到HBase2.x的RIT状态并不像HBase1.x系列存储在Zookeeper上。经过测试删除/hbase节点重启hbase并不能解决RIT问题。

HBase表删除：

```
hbase  shell>disable  M_TDY_PT_LCZZ
```

disable失败，所以正常删除表操作无法执行。需要进行暴力删除，暴力删除指的是从元数据进行删除。

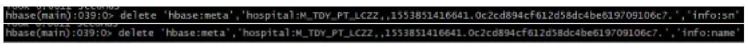
先停掉HBase



删除hdfs表目录(记得先备份，等下恢复用)

```
hdfs  dfs  -cp  /hbase/data/hospital/P_TDY_DASC_DE02_01_039_63  /
hdfs  dfs  -cp  /hbase/data/hospital/M_TDY_PT_LCZZ  /
hdfs  dfs  -rm  -r  -skipTrash  /hbase/data/hospital/P_TDY_DASC_DE02_01_039_63
hdfs  dfs  -rm  -r  -skipTrash  /hbase/data/hospital/  M_TDY_PT_LCZZ
delete  'hbase:meta','rowkey',  'column'
```

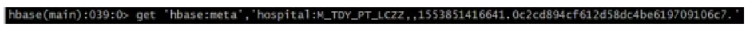
Rowkey信息可以通过hbase的UI看到：



```
hbase(main):039:0> delete 'hbase:meta','hospital:M_TDY_PT_LCZZ,,1553851416641.0c2cd894cf612d58dc4be619709106c7.','info:sn'
hbase(main):039:0> delete 'hbase:meta','hospital:M_TDY_PT_LCZZ,,1553851416641.0c2cd894cf612d58dc4be619709106c7.','info:name'
```

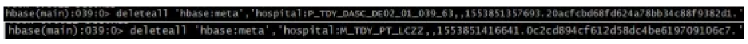
然后重启HBase，但是发现问题没有解决。

hbase shell查询数据看到hbase的meta删除失败了，原本的meta信息还在：



```
hbase(main):039:0> get 'hbase:meta','hospital:M_TDY_PT_LCZZ,,1553851416641.0c2cd894cf612d58dc4be619709106c7.'
```

然后重新删除(delete命令换成deleteall命令)：



```
hbase(main):039:0> deleteall 'hbase:meta','hospital:P_TDY_DASC_DE02_01_039_63,,1553851357693.20acfcbd68fd624a78bb34c88f9382d1.'
hbase(main):039:0> deleteall 'hbase:meta','hospital:M_TDY_PT_LCZZ,,1553851416641.0c2cd894cf612d58dc4be619709106c7.'
```

再删除Zookeeper中的/hbase节点，重启HBase发现RIT问题已经解决了。

后续就是重建表，然后恢复数据。

Phoenix故障处理笔记

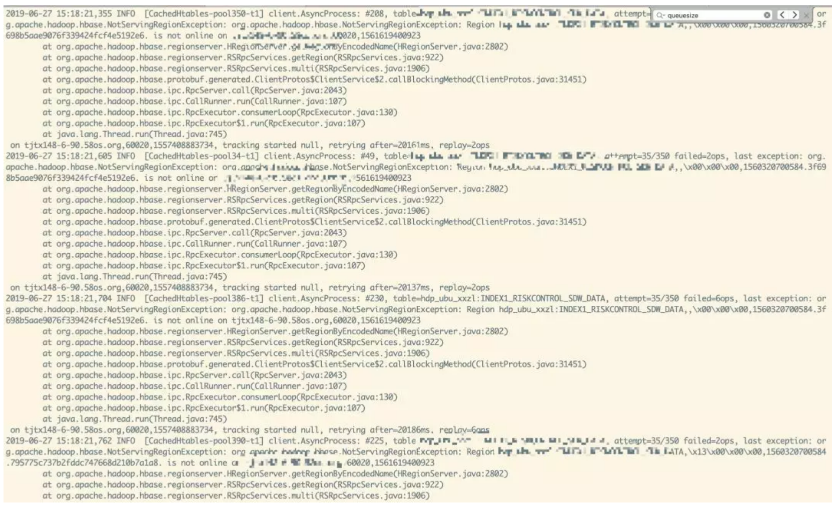
1. Timeline

- 06-26 16:00 Phoenix使用方反馈慢；
- 06-26 16:02 同事通过监控看到Phoenix HBase集群一个对应的RegionServer，QueueSize过高，此bug基本是Butch Put Lock在高并发写入的问题，我们已在下个版本中增加信息日志定位此问题；
- 06-26 16:05 同事重启该队列过高的RegionServer；
- 06-26 16:10 同事跟我说，好多Phoenix的Region处于RIT状态；
- 06-26 17:00 暂停该Phoenix集群所有的写入；
- 06-26 20:00 跟业务沟通，可能会正常影响一段时间，经同意。至此各种hbck，各种重启RegionServer&Master不怎么管用，RIT数量升至550个；
- 06-27 12:00 尝试修复；
- 06-27 15:00 问题修复。

2. 处理流程

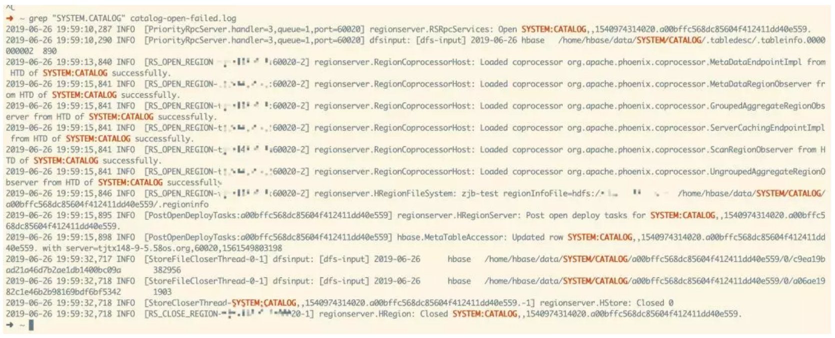
2.1 异常现象

1. 大量Region无法上线(NotServingRegionException)



```
2019-06-27 15:18:21,355 INFO  [CachedHtables-pool350-t1] client.AsyncProcess: #208, table[redacted] attempt= or
g.apache.hadoop.hbase.NotServingRegionException: org.apache.hadoop.hbase.NotServingRegionException: Region [redacted]A,,\x00\x00\x00,1560320700584.3f
698b5aae9076f339424fcf4e5192e6. is not online on [redacted]0020,1561619400923
        at org.apache.hadoop.hbase.regionserver.HRegionServer.[redacted]ByEncodedName(HRegionServer.java:2802)
        at org.apache.hadoop.hbase.regionserver.RSRpcServices.getRegion(RSRpcServices.java:922)
        at org.apache.hadoop.hbase.regionserver.RSRpcServices.multi(RSRpcServices.java:1906)
        at org.apache.hadoop.hbase.protobuf.generated.ClientProtos$ClientService$2.callBlockingMethod(ClientProtos.java:31451)
        at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:2043)
        at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:107)
        at org.apache.hadoop.hbase.ipc.RpcExecutor.consumerLoop(RpcExecutor.java:130)
        at org.apache.hadoop.hbase.ipc.RpcExecutor$1.run(RpcExecutor.java:107)
        at java.lang.Thread.run(Thread.java:745)
 on tjtx148-6-90.58os.org,60020,1557408883734, tracking started null, retrying after=20161ms, replay=2ops
2019-06-27 15:18:21,605 INFO  [CachedHtables-pool34-t1] client.AsyncProcess: #49, table[redacted] attempt=35/350 failed=2ops, last exception: org.
apache.hadoop.hbase.NotServingRegionException: org.[redacted]hbase.NotServingRegionException: [redacted],,\x00\x00\x00,1560320700584.3f69
8b5aae9076f339424fcf4e5192e6. is not online or [redacted]561619400923
        at org.apache.hadoop.hbase.regionserver.HRegionServer.getRegionByEncodedName(HRegionServer.java:2802)
        at org.apache.hadoop.hbase.regionserver.RSRpcServices.getRegion(RSRpcServices.java:922)
        at org.apache.hadoop.hbase.regionserver.RSRpcServices.multi(RSRpcServices.java:1906)
        at org.apache.hadoop.hbase.protobuf.generated.ClientProtos$ClientService$2.callBlockingMethod(ClientProtos.java:31451)
        at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:2043)
        at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:107)
        at org.apache.hadoop.hbase.ipc.RpcExecutor.consumerLoop(RpcExecutor.java:130)
        at org.apache.hadoop.hbase.ipc.RpcExecutor$1.run(RpcExecutor.java:107)
        at java.lang.Thread.run(Thread.java:745)
 on tjtx148-6-90.58os.org,60020,1557408883734, tracking started null, retrying after=20137ms, replay=2ops
2019-06-27 15:18:21,704 INFO  [CachedHtables-pool386-t1] client.AsyncProcess: #230, table=hdp_ubu_xxzl:INDEX1_RISKCONTROL_SDW_DATA, attempt=35/350 failed=6ops, last exception: or
g.apache.hadoop.hbase.NotServingRegionException: org.apache.hadoop.hbase.NotServingRegionException: Region hdp_ubu_xxzl:INDEX1_RISKCONTROL_SDW_DATA,,\x00\x00\x00,1560320700584.3f
698b5aae9076f339424fcf4e5192e6. is not online on tjtx148-6-90.58os.org,60020,1561619400923
        at org.apache.hadoop.hbase.regionserver.HRegionServer.getRegionByEncodedName(HRegionServer.java:2802)
        at org.apache.hadoop.hbase.regionserver.RSRpcServices.getRegion(RSRpcServices.java:922)
        at org.apache.hadoop.hbase.regionserver.RSRpcServices.multi(RSRpcServices.java:1906)
        at org.apache.hadoop.hbase.protobuf.generated.ClientProtos$ClientService$2.callBlockingMethod(ClientProtos.java:31451)
        at org.apache.hadoop.hbase.ipc.RpcServer.call(RpcServer.java:2043)
        at org.apache.hadoop.hbase.ipc.CallRunner.run(CallRunner.java:107)
        at org.apache.hadoop.hbase.ipc.RpcExecutor.consumerLoop(RpcExecutor.java:130)
        at org.apache.hadoop.hbase.ipc.RpcExecutor$1.run(RpcExecutor.java:107)
        at java.lang.Thread.run(Thread.java:745)
 on tjtx148-6-90.58os.org,60020,1557408883734, tracking started null, retrying after=20186ms, replay=6ops
2019-06-27 15:18:21,762 INFO  [CachedHtables-pool390-t1] client.AsyncProcess: #225, table [redacted], attempt=35/350 failed=2ops, last exception: or
g.apache.hadoop.hbase.NotServingRegionException: org[redacted]hbase.NotServingRegionException: Region [redacted]ATA,\x13\x00\x00\x00,1560320700584
.795775c737b2fddc747668d210b7a1a8. is not online o[redacted]60020,1561619400923
        at org.apache.hadoop.hbase.regionserver.HRegionServer.getRegionByEncodedName(HRegionServer.java:2802)
        at org.apache.hadoop.hbase.regionserver.RSRpcServices.getRegion(RSRpcServices.java:922)
        at org.apache.hadoop.hbase.regionserver.RSRpcServices.multi(RSRpcServices.java:1906)
```

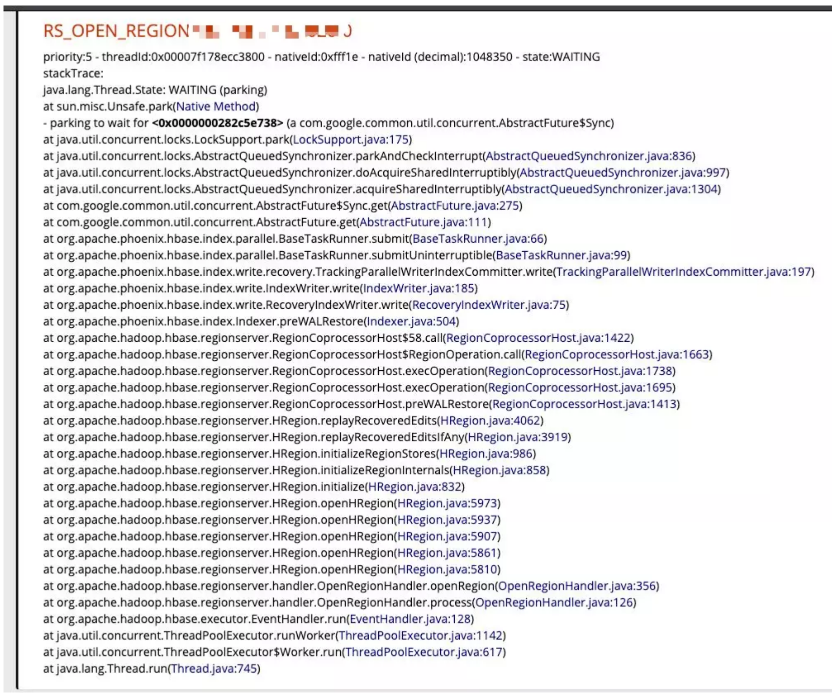
2. Phoenix的SYSTEM.CATALOG系统表也无法上线。



```
^C
➜ ~ grep "SYSTEM.CATALOG" catalog-open-failed.log
2019-06-26 19:59:10,287 INFO  [PriorityRpcServer.handler=3,queue=1,port=60020] regionserver.RSRpcServices: Open SYSTEM:CATALOG,,1540974314020.a00bffc568dc85604f412411dd40e559.
2019-06-26 19:59:10,290 INFO  [PriorityRpcServer.handler=3,queue=1,port=60020] dfsinput: [dfs-input] 2019-06-26 hbase   /home/hbase/data/SYSTEM/CATALOG/.tabledesc/.tableinfo.0000
000002  890
2019-06-26 19:59:13,840 INFO  [RS_OPEN_REGION[redacted]:60020-2] regionserver.RegionCoprocessorHost: Loaded coprocessor org.apache.phoenix.coprocessor.MetaDataEndpointImpl from
 HTD of SYSTEM:CATALOG successfully.
2019-06-26 19:59:15,841 INFO  [RS_OPEN_REGION-[redacted]:60020-2] regionserver.RegionCoprocessorHost: Loaded coprocessor org.apache.phoenix.coprocessor.MetaDataRegionObserver fr
om HTD of SYSTEM:CATALOG successfully.
2019-06-26 19:59:15,841 INFO  [RS_OPEN_REGION-[redacted]:60020-2] regionserver.RegionCoprocessorHost: Loaded coprocessor org.apache.phoenix.coprocessor.GroupedAggregateRegionObs
erver from HTD of SYSTEM:CATALOG successfully.
2019-06-26 19:59:15,841 INFO  [RS_OPEN_REGION-t[redacted]:60020-2] regionserver.RegionCoprocessorHost: Loaded coprocessor org.apache.phoenix.coprocessor.ServerCachingEndpointImpl
 from HTD of SYSTEM:CATALOG successfully.
2019-06-26 19:59:15,841 INFO  [RS_OPEN_REGION-t[redacted]:60020-2] regionserver.RegionCoprocessorHost: Loaded coprocessor org.apache.phoenix.coprocessor.ScanRegionObserver from H
TD of SYSTEM:CATALOG successfully.
2019-06-26 19:59:15,841 INFO  [RS_OPEN_REGION-t[redacted]:60020-2] regionserver.RegionCoprocessorHost: Loaded coprocessor org.apache.phoenix.coprocessor.UngroupedAggregateRegionO
bserver from HTD of SYSTEM:CATALOG successfully.
2019-06-26 19:59:15,846 INFO  [RS_OPEN_REGION-[redacted]:60020-2] regionserver.HRegionFileSystem: zjb-test regionInfoFile=hdfs:/[redacted]  /home/hbase/data/SYSTEM/CATALOG/
a00bffc568dc85604f412411dd40e559/.regioninfo
2019-06-26 19:59:15,895 INFO  [PostOpenDeployTasks:a00bffc568dc85604f412411dd40e559] regionserver.HRegionServer: Post open deploy tasks for SYSTEM:CATALOG,,1540974314020.a00bffc5
68dc85604f412411dd40e559.
2019-06-26 19:59:15,898 INFO  [PostOpenDeployTasks:a00bffc568dc85604f412411dd40e559] hbase.MetaTableAccessor: Updated row SYSTEM:CATALOG,,1540974314020.a00bffc568dc85604f412411dd
40e559. with server=tjtx148-9-5.58os.org,60020,1561549803198
2019-06-26 19:59:32,717 INFO  [StoreFileCloserThread-0-1] dfsinput: [dfs-input] 2019-06-26      hbase   /home/hbase/data/SYSTEM/CATALOG/a00bffc568dc85604f412411dd40e559/0/c9ea19b
ad21a46d7b2ae1db1400bc09a        382956
2019-06-26 19:59:32,718 INFO  [StoreFileCloserThread-0-1] dfsinput: [dfs-input] 2019-06-26      hbase   /home/hbase/data/SYSTEM/CATALOG/a00bffc568dc85604f412411dd40e559/0/a06ae19
82c1e46b2b98169bdf6bf5342        1903
2019-06-26 19:59:32,718 INFO  [StoreCloserThread-SYSTEM:CATALOG,,1540974314020.a00bffc568dc85604f412411dd40e559.-1] regionserver.HStore: Closed 0
2019-06-26 19:59:32,718 INFO  [RS_CLOSE_REGION-[redacted]20-1] regionserver.HRegion: Closed SYSTEM:CATALOG,,1540974314020.a00bffc568dc85604f412411dd40e559.
➜ ~ ▌
```

2.2 处理流程

1. 手动assign SYSTEM.CATALOG 表的Region上线，并且跟踪Master&对应RegionServer的日志。整个offline&open流程都正常。但是中间会由于各种其他的表不在线failover后close掉；
2. 打印jstack, 感觉这几个线程有问题，都在waiting；



```
RS_OPEN_REGION[redacted]
priority:5 - threadId:0x00007f178ecc3800 - nativeId:0xfff1e - nativeId (decimal):1048350 - state:WAITING
stackTrace:
java.lang.Thread.State: WAITING (parking)
at sun.misc.Unsafe.park(Native Method)
- parking to wait for <0x0000000282c5e738> (a com.google.common.util.concurrent.AbstractFuture$Sync)
at java.util.concurrent.locks.LockSupport.park(LockSupport.java:175)
at java.util.concurrent.locks.AbstractQueuedSynchronizer.parkAndCheckInterrupt(AbstractQueuedSynchronizer.java:836)
at java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireSharedInterruptibly(AbstractQueuedSynchronizer.java:997)
at java.util.concurrent.locks.AbstractQueuedSynchronizer.acquireSharedInterruptibly(AbstractQueuedSynchronizer.java:1304)
at com.google.common.util.concurrent.AbstractFuture$Sync.get(AbstractFuture.java:275)
at com.google.common.util.concurrent.AbstractFuture.get(AbstractFuture.java:111)
at org.apache.phoenix.hbase.index.parallel.BaseTaskRunner.submit(BaseTaskRunner.java:66)
at org.apache.phoenix.hbase.index.parallel.BaseTaskRunner.submitUninterruptible(BaseTaskRunner.java:99)
at org.apache.phoenix.hbase.index.write.recovery.TrackingParallelWriterIndexCommitter.write(TrackingParallelWriterIndexCommitter.java:197)
at org.apache.phoenix.hbase.index.write.IndexWriter.write(IndexWriter.java:185)
at org.apache.phoenix.hbase.index.write.RecoveryIndexWriter.write(RecoveryIndexWriter.java:75)
at org.apache.phoenix.hbase.index.Indexer.preWALRestore(Indexer.java:504)
at org.apache.hadoop.hbase.regionserver.RegionCoprocessorHost$58.call(RegionCoprocessorHost.java:1422)
at org.apache.hadoop.hbase.regionserver.RegionCoprocessorHost$RegionOperation.call(RegionCoprocessorHost.java:1663)
at org.apache.hadoop.hbase.regionserver.RegionCoprocessorHost.execOperation(RegionCoprocessorHost.java:1738)
at org.apache.hadoop.hbase.regionserver.RegionCoprocessorHost.execOperation(RegionCoprocessorHost.java:1695)
at org.apache.hadoop.hbase.regionserver.RegionCoprocessorHost.preWALRestore(RegionCoprocessorHost.java:1413)
at org.apache.hadoop.hbase.regionserver.HRegion.replayRecoveredEdits(HRegion.java:4062)
at org.apache.hadoop.hbase.regionserver.HRegion.replayRecoveredEditsIfAny(HRegion.java:3919)
at org.apache.hadoop.hbase.regionserver.HRegion.initializeRegionStores(HRegion.java:986)
at org.apache.hadoop.hbase.regionserver.HRegion.initializeRegionInternals(HRegion.java:858)
at org.apache.hadoop.hbase.regionserver.HRegion.initialize(HRegion.java:832)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:5973)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:5937)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:5907)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:5861)
at org.apache.hadoop.hbase.regionserver.HRegion.openHRegion(HRegion.java:5810)
at org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler.openRegion(OpenRegionHandler.java:356)
at org.apache.hadoop.hbase.regionserver.handler.OpenRegionHandler.process(OpenRegionHandler.java:126)
at org.apache.hadoop.hbase.executor.EventHandler.run(EventHandler.java:128)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
```

- 通过上面的信息看，open region确实有问题。查看Phoenix Indexer Observer源码就会知道是在根据Recover WAL Entry构建索引；
- 修改hbase.regionserver.executor.openregion.threads数,此配置是负责open region的handler数：

```
<property>
        <name>hbase.regionserver.executor.openregion.threads</name>
        <value>50</value>
</property>
```

默认 3， 我们这边的hbase版本(1.0.0-cdh5.4.4)

- **重启RegionServer;**
- assign SYSTEM.CATAOG 表的Region，上线成功;
- 修修补补，fixMeta fixAssignments就ok了。

3. 原理分析

1. 重启RegionServer， 会造成该RegionServer上面的Region下线，并且被重新Balance到新的RegionServer中。

2. Region在新的RegionServer中open过程会找到该Region下的recover.edits 文件，进行replay;

3.Phoenix表使用HBase的协处理类型之Observer，具体使用查看示例

org.apache.phoenix.hbase.index.Indexer，此用作根据WAL构建索引的，具体参考Phoenix的相关材料。

4. 在SYSTEM.CATALOG 的打开过程中，会查询其他的里面表，其他的表也处于RIT未恢复。然而其他的表Region在open的过程也需要构建Index，尚且有一部分在openregion的队列里面。最终SYSTEM.CATALOG无法上线(此处不准确，纯属囫囵吞枣似的查看源码推测)。

5. 增加open region handler数之后，重启RegionServer后，需要进行一些hbck -fixMeta -fixAssginment 将一些未上线的Region上线， 就ok了。

6. 如果出现个别的Region还是上线失败，那就手动解决吧！个人认为比hbck -repair暴力修复靠谱。