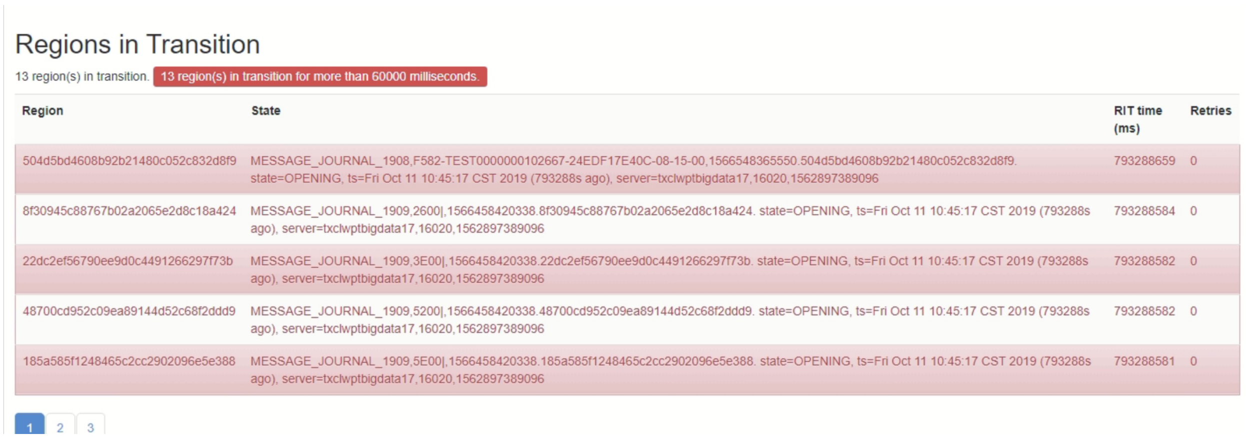
HBase 2.0.0 META 数据修复工具

问题起因

必须先吐槽一下 Cloudera 6.x 和 Hbase 2.0 太坑了!

不久前生产上的一套Hbase集群出现著名的RIT（Regions in Transition）问题。

查看hbase web ui



Regions in Transition

13 region(s) in transition. 13 region(s) in transition for more than 60000 milliseconds.

Region	State	RIT time (ms)	Retries
504d5bd4608b92b21480c052c832d8f9	MESSAGE_JOURNAL_1908,F582-TEST0000000102667-24EDF17E40C-08-15-00,1566548365550.504d5bd4608b92b21480c052c832d8f9. state=OPENING, ts=Fri Oct 11 10:45:17 CST 2019 (793288s ago), server=txclwptbigdata17,16020,1562897389096	793288659	0
8f30945c88767b02a2065e2d8c18a424	MESSAGE_JOURNAL_1909,2600|,1566458420338.8f30945c88767b02a2065e2d8c18a424. state=OPENING, ts=Fri Oct 11 10:45:17 CST 2019 (793288s ago), server=txclwptbigdata17,16020,1562897389096	793288584	0
22dc2ef56790ee9d0c4491266297f73b	MESSAGE_JOURNAL_1909,3E00|,1566458420338.22dc2ef56790ee9d0c4491266297f73b. state=OPENING, ts=Fri Oct 11 10:45:17 CST 2019 (793288s ago), server=txclwptbigdata17,16020,1562897389096	793288582	0
48700cd952c09ea89144d52c68f2ddd9	MESSAGE_JOURNAL_1909,5200|,1566458420338.48700cd952c09ea89144d52c68f2ddd9. state=OPENING, ts=Fri Oct 11 10:45:17 CST 2019 (793288s ago), server=txclwptbigdata17,16020,1562897389096	793288582	0
185a585f1248465c2cc2902096e5e388	MESSAGE_JOURNAL_1909,5E00|,1566458420338.185a585f1248465c2cc2902096e5e388. state=OPENING, ts=Fri Oct 11 10:45:17 CST 2019 (793288s ago), server=txclwptbigdata17,16020,1562897389096	793288581	0

1 2 3

于是通过hbck命令查看一下集群状态，果然好多inconsistency

```
 ...
 ERROR: Region { meta => XXX,XXX:,1573019231000.ff2aecaf28917792395c341d01e0b8cc.,
hdfs => hdfs://nameservice1/hbase/data/default/XXX/ff2aecaf28917792395c341d01e0b8cc,
deployed => , replicaId => 0 } not deployed on any region server.
 ...
 ERROR: Found inconsistency in table XXX
 ...
 9 inconsistencies detected.
 Status: INCONSISTENT
```

看到错误提示问题明显了，这个Region在hdfs中有数据文件但没有依赖任何Region Server，原因可能region被原来的Region Server unassigned了，但是还没有被assigned到一个新的Region Server上。

那么尝试用[redacted]和[redacted]来修复吧，于是就有了下面的提示，hbase2.0+以后hbck的所有修复功能全都不支持...

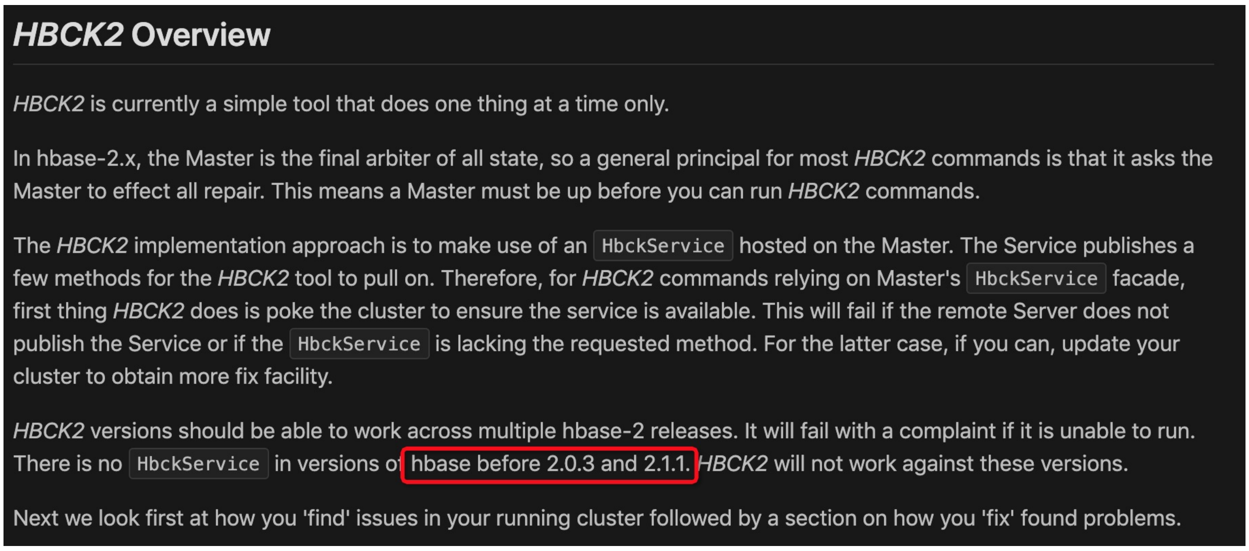
```
-----------------------------------------------------------------------
NOTE: As of HBase version 2.0, the hbck tool is significantly changed.
In general, all Read-Only options are supported and can be be used
safely. Most -fix/ -repair options are NOT supported. Please see usage
below for details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

NOTE: Following options are NOT supported as of HBase version 2.0+.

  UNSUPPORTED Metadata Repair options: (expert features, use with caution!)
```

```
   -fix              Try to fix region assignments.  This is for backwards
compatiblity
   -fixAssignments   Try to fix region assignments.  Replaces the old -fix
   -fixMeta          Try to fix meta problems.  This assumes HDFS region info is good.
   -fixHdfsHoles     Try to fix region holes in hdfs.
   ...
  UNSUPPORTED Metadata Repair shortcuts
   -repair           Shortcut for -fixAssignments -fixMeta -fixHdfsHoles -
fixHdfsOrphans -fixHdfsOverlaps -fixVersionFile -sidelineBigOverlaps -
fixReferenceFiles-fixHFileLinks
   -repairHoles      Shortcut for -fixAssignments -fixMeta -fixHdfsHoles
```
既然hbck不支持，觉得hbase总得有解决方案吧，科学上网后发现hbase2.0+提供了一个叫hbck2工具，不过得自己编译麻烦了点。

克隆下来准备动手编译发现不对，于是仔细看了一下hbck2的介绍，最低支持版本2.0.3和2.1.1


HBCK2 Overview

HBCK2 is currently a simple tool that does one thing at a time only.

In hbase-2.x, the Master is the final arbiter of all state, so a general principal for most *HBCK2* commands is that it asks the Master to effect all repair. This means a Master must be up before you can run *HBCK2* commands.

The *HBCK2* implementation approach is to make use of an `HbckService` hosted on the Master. The Service publishes a few methods for the *HBCK2* tool to pull on. Therefore, for *HBCK2* commands relying on Master's `HbckService` facade, first thing *HBCK2* does is poke the cluster to ensure the service is available. This will fail if the remote Server does not publish the Service or if the `HbckService` is lacking the requested method. For the latter case, if you can, update your cluster to obtain more fix facility.

HBCK2 versions should be able to work across multiple hbase-2 releases. It will fail with a complaint if it is unable to run. There is no `HbckService` in versions of hbase before 2.0.3 and 2.1.1. *HBCK2* will not work against these versions.

Next we look first at how you 'find' issues in your running cluster followed by a section on how you 'fix' found problems.

WTF......这就是个黑洞啊，还有你就不能把支持的版本号字体放大点吗!

修复方案

吐槽过后，还是得想解决办法啊：

1. 升级Hbase版本
    - 目前这种情况是根本无法升级的，存量数据怎么办，就算数据可以重入，目前使用的hbase是CDH版，Cloudera 6.x版本集成的hbase只有2.0.0和2.1.0版本，还是黑洞。。。此方案行不通。
2. 暴力删除hbase数据
    - 暴力删除数据，格式化hdfs，删除hbasemeta数据，删除zookeeper记录，这和重新部署一套hbase差不多了，但是前提是数据可以重入或者允许清除，那以后怎么办，总不能一遇到问题就删库吧，生产上面的数据一般都比较敏感根本不能删。。。此方案行不通。
3. 写个工具修复hbase
    - 看来只能这样了。。。

修复步骤

回到最初的错误提示，思考一下，如果Region下数据文件在hdfs中存在，那是否可以通过.regioninfo文件（hdfs存储hbase region信息的文件）获取Region信息，同时读

取'hbase:meta'表中的Region信息，进行对比取差集就是要修复的Region，然后将需要修复的Region信息再写入到'hbase:meta'中。

按照这个思路，先验证一下hdfs

检测一下hbase的block是否完整 [redacted]

```
Status: HEALTHY
 Number of data-nodes:  12
 Number of racks:               1
 Total dirs:                    4650
 Total symlinks:                0
...
The filesystem under path '/hbase' is HEALTHY
```

检查一下.regioninfo文件是否完整 [redacted]

```
Found 4 items
-rw-r--r--   3 hbase hbase         65 2019-10-26 18:29
/hbase/data/default/XXX/ff2aecaf28917792395c341d01e0b8cc/.regioninfo
drwxr-xr-x   - hbase hbase          0 2019-11-26 09:37
/hbase/data/default/XXX/ff2aecaf28917792395c341d01e0b8cc/.tmp
drwxr-xr-x   - hbase hbase          0 2019-11-26 13:59
/hbase/data/default/XXX/ff2aecaf28917792395c341d01e0b8cc/0
drwxr-xr-x   - hbase hbase          0 2019-10-26 18:29
/hbase/data/default/XXX/ff2aecaf28917792395c341d01e0b8cc/recovered.edits
```

再看一下'hbase:meta'中的存储结构：

列名	说明
info:state	Region状态
info:sn	Region Server Node，由 server和serverstart 成，如slave1,16020,1557998852385
info:serverstartcode	Region Server启动Code，实质上就是Region 启动的时间戳
info:server	Region Server 地址和端口，如slave1:16020
info:seqnumDuringOpen	表示Region在线时长的一个二进制串
info:regioninfo	Region Info，和.regioninfo内容相同

OK，觉得这个方案可行，接下来就开始动手coding吧

获取'hbase:mata'中的Region信息

```
    public Set<String> getMetaRegions(Configuration conf, String tableName) throws
Exception {

        Connection conn = ConnectionFactory.createConnection(conf);
        Table table = conn.getTable(TableName.valueOf(TABLE));

        PrefixFilter filter = new PrefixFilter(Bytes.toBytes(tableName + ","));

        Scan scan = new Scan();
        scan.setFilter(filter);

        Set<String> metaRegions = new HashSet<>();

        Iterator<Result> iterator = table.getScanner(scan).iterator();
        while (iterator.hasNext()) {
            Result result = iterator.next();
            metaRegions.add(Bytes.toString(result.getRow()));
        }

        conn.close();

        return metaRegions;
    }
```

读取.regioninfo中的Region信息

```
    public Map<String, RegionInfo> getHdfsRegions(Configuration conf, String
tablePath) throws Exception {

        FileSystem fs = FileSystem.get(conf);
        Path path = new Path(hdfsRootDir + "/data/default/" + tablePath + "/");

        Map<String, RegionInfo> hdfsRegions = new HashMap<>();

        FileStatus[] list = fs.listStatus(path);
        for (FileStatus status : list) {
            if (!status.isDirectory()) {
                continue;
            }

            boolean isRegion = false;
            FileStatus[] regions = fs.listStatus(status.getPath());
            for (FileStatus regionStatus : regions) {
                if (regionStatus.toString().contains(REGION_INFO_FILE)) {
                    isRegion = true;
                    break;
                }
            }

            if (!isRegion) {
                continue;
            }
            RegionInfo hri = HRegionFileSystem.loadRegionInfoFileContent(fs,
status.getPath());
            hdfsRegions.put(hri.getRegionNameAsString(), hri);

        }
        return hdfsRegions;
    }
```

两者进行对比取差集

```
        Set<String> metaRegions = getMetaRegions(configuration, repairTableName);

        Map<String, RegionInfo> hdfsRegions = getHdfsRegions(configuration,
repairTableName);

        Set<String> hdfsRegionNames = hdfsRegions.keySet();

        metaRegions.removeAll(hdfsRegionNames);
```

构造META信息并写入HBase

```
        ServerName[] regionServers = admin.getRegionServers().toArray(new
ServerName[0]);

        int rsLength = regionServers.length;
        int i = 0;
        for (String regionName : hdfsRegionNames) {

            String sn = regionServers[i % rsLength].getServerName();
            String[] snSig = sn.split(",");

            RegionInfo hri = hdfsRegions.get(regionName);
            Put info = MetaTableAccessor.makePutFromRegionInfo(hri,
EnvironmentEdgeManager.currentTime());
            info.addColumn(Bytes.toBytes(FAMILY), Bytes.toBytes(SN),
Bytes.toBytes(sn));
            info.addColumn(Bytes.toBytes(FAMILY), Bytes.toBytes(SERVER),
Bytes.toBytes(snSig[0] + ":" + snSig[1]));
            info.addColumn(Bytes.toBytes(FAMILY), Bytes.toBytes(STATE),
Bytes.toBytes("OPEN"));

            table.put(info);
            i++;
        }
```

重启Region Server 和 Hbase Master，重启之后会自动生成'info:seqnumDuringOpen'以及'info:serverstartcode'

工具开发完成后，找了个环境验证了一下，没出什么问题，接下来就部署到生产上试试了，反正hbase已经这个样子，死马当司马懿吧。

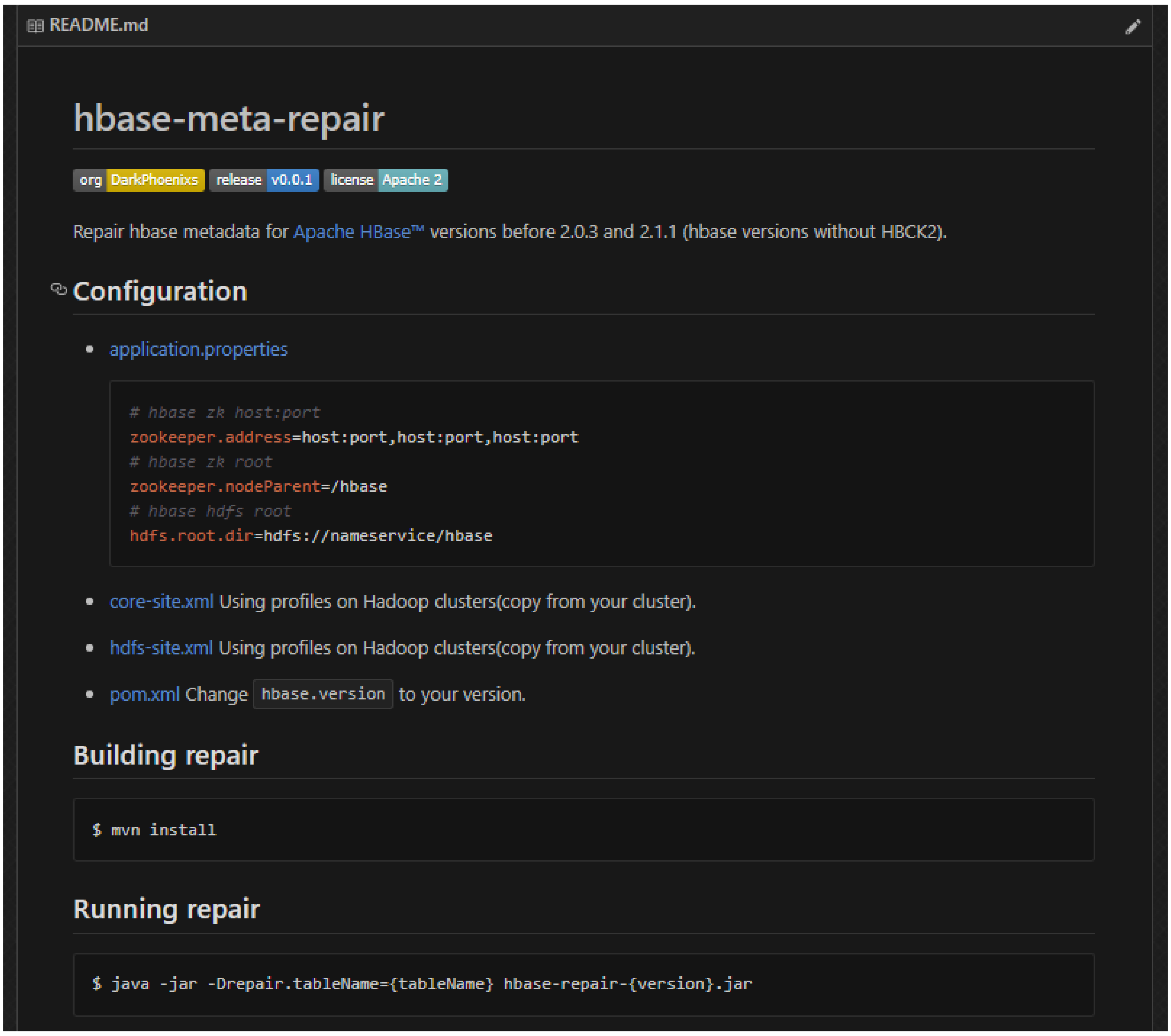
先用了个region不多的表试验，发现可以呀，然后陆续把所有错误的表都修复一遍，重启hbase，接下来就是见证BUG的时刻：

```
...
0 inconsistencies detected.
Status: OK
```

hbase修复完成 此处有掌声

修复工具

本着开源精神，工具已上传GitHub ： hbase-meta-repair



README.md

hbase-meta-repair

org DarkPhoenixs release v0.0.1 license Apache 2

Repair hbase metadata for Apache HBase™ versions before 2.0.3 and 2.1.1 (hbase versions without HBCK2).

Configuration

- application.properties

```
# hbase zk host:port
zookeeper.address=host:port,host:port,host:port
# hbase zk root
zookeeper.nodeParent=/hbase
# hbase hdfs root
hdfs.root.dir=hdfs://nameservice/hbase
```

- core-site.xml Using profiles on Hadoop clusters(copy from your cluster).
- hdfs-site.xml Using profiles on Hadoop clusters(copy from your cluster).
- pom.xml Change `hbase.version` to your version.

Building repair

```
$ mvn install
```

Running repair

```
$ java -jar -Drepair.tableName={tableName} hbase-repair-{version}.jar
```