

# Hbase 集群迁移

方案一: DistCp 迁移(需要停掉集群)

1)旧的集群上先 disable alltable:

disable tablename(单张表 disable)

disable\_all (整个 hbase 所有表 disable)

同理 enable tablename 以及 enable\_all

2)hadoop distcp -Ddfs.checksum.type=CRC32 -skipcrccheck -i -update  
hdfs://src\_cluster:9000/hbase/tablename

hdfs://dest\_cluester:8020/hbase/data/default/tablename

或者使用

hadoop distcp -Ddfs.checksum.type=CRC32 -skipcrccheck -i -update  
hftp://src\_cluster:50070/hbase/tablename

webhdfs://dest\_cluester:50070/hbase/data/default/tablename

或者使用

hadoop distcp -Ddfs.checksum.type=CRC32 -skipcrccheck -i -update  
hftp://src\_cluster:50070/hbase/tablename

hftp://dest\_cluester:50070/hbase/data/default/tablename

使用脚本进行迁移

将所有表名放在一个文件中进行读取:

```
#!/bin/bash
```

```
cat 'xx.txt' | while read line
do
    hadoop distcp -Ddfs.checksum.type=CRC32 -skipcrccheck -i -update
    hftp://src_cluster:50070/hbase/$line webhdfs://dest_cluester:50070/hbase/data/default/$line
```

sleep 5m(针对不同表的数据量, 进行相应的睡眠等待上一张迁移成功再进行下一张的迁移, 睡眠时间自行选择, 看表的数据量自行定义)

done

注: 具体细节需要各自实践进行调整

集体杀死 DistCp 进程:

```
jps | grep DistCp | awk '{print $1}' | xargs kill -9
```

DSTools 修复表操作(针对某平台环境, 提供了修复工具):

sh runDSTools.sh tablename (单张表修复)

sh runDSTools.sh 1>&2 2> /tmp/xx.log.1(不加表名即表示整个集群修复, 此处为脚本工具, 可以参考 hbase 官网数据修复.MATE 修复部分)

DSTools 修复 Hfile 命令(针对某平台环境, 提供了修复工具):

sh runHFileCheck.sh tablename 1>&2 2> /tmp/hfilecheck.log.1

```
sh runHFileCheck.sh -fixReferenceFiles -sidelineCorruptHFiles tablename 1>&2 2>
/tmp/hfilecheck.log.2
sh runHFileCheck.sh tablename 1>&2 2> /tmp/hfilecheck.log.3
```

数据表修复时出现 regionserver 时常死亡时，请修改以下参数(可进行适当的调整)

1)hbase-site.xml 里，配置 hbase.ipc.server.max.callqueue.length 为 10。

```
<property>
  <name>hbase.ipc.server.max.callqueue.length</name>
  <value>10</value>
</property>
```

2)hbase-env.sh 里，修改配置：

```
-XX:CMSInitiatingOccupancyFraction=80 ==> -XX:CMSInitiatingOccupancyFraction=70
-XX:NewRatio=3 ==> -XX:NewRatio=4
```

3)hbase-env.sh 里，regionserver 的 heap size 只允许两种配置：

```
-Xms24000m -Xmx24000m 或者 -Xms30000m -Xmx30000m
```

如果机器内存比较充足，可以选后者，否则选前者。

批量删除 hbase 表命令：

1)先将所有 hbase 表的表名列出存在文件 hbase-tmp1 中

```
for i in `echo 'list' | hbase shell`;do echo $i >> /tmp/hbase-tmp1;done
```

2)打开这个文件删除不相干的信息，只保留表名信息

3)执行批量删除操作

```
for i in `cat /tmp/hbase-tmp1`;do `echo "delete_table '$i'" | hbase shell`;done
```

使用 delete\_table 可以直接删除表，不要事先 disable，单张表删除可以使用 delete\_table 直接删除或者先 disable tablename 再 drop tablename

**方案二：Replication(动态备份)**

这个是动态的备份（可以理解为实时备份）

**步骤：**

1)在 old cluster 将需要迁移的表属性进行修改：

```
disable 'your_table'
alter 'your_table', {NAME => 'family_name', REPLICATION_SCOPE => '1'}
enable 'your_table'
```

2)打开 new cluster 集群的 replication，修改 hbase-site.xml

```
<property>
  <name>hbase.replication</name>
  <value>true</value>
</property>
```

3)添加 peer，在 new cluster 的 hbase shell 中执行：add\_peer '1','old cluster ip:2181:/hbase'，启动 replication，执行 start\_replication

**说明：**需要两个集群的 hadoop 版本一致，否则出现读写错误

**方案三：CopyTable 迁移(在线，前提在目标集群中比较建立好同名表)**

可以在本集群中拷贝一张表，也可以将表拷贝到其他的集群中。

```
命令：./hbase org.apache.hadoop.hbase.mapreduce.CopyTable --
peer.adr=dest_cluster:2181:/hyperbase1
```

说明: 1)拷贝完成, 不需要重启机器, 在 new cluster 中就可以看到该表;

2)稳定性还需要考虑, 测试过程中遇到一个问题, 拷贝过程中始终都停留在这里  
Lookedup root region location, 查看日志没有什么错误和相关的信息。

方案四: EXPORT/IMPORT 迁移(在线, 前提在目标集群中比较建立好同名表)

步骤:

1)在 old cluster 上执行: `./hbase org.apache.hadoop.hbase.mapreduce.Export test hdfs://new cluster ip:9000/zhuangyang/test`

2)在 new cluster 上执行: `./hbase org.apache.hadoop.hbase.mapreduce.Import test hdfs://new cluster ip:9000/zhuangyang/test`

说明:

1)一定要写全路径, 不能写相对路劲;

2)在 import 前, 需要将表事先在 new cluster 中创建好.

方案五: HBASE REST API 方式迁移

具体细节见 Hbase API 部分

步骤:

1)HBASE 的 REST 全是 GET、POST、PUT 方式处理, 可以在程序代码中转成相应的  
HttpClient 的 GET、POST、PUT 相应请求

2)以 A 表为例处理流程如下(在启动了 REST 方式的前提下):

a)首先根据表名获取该表的 Schema 信息

REST 在 Shell 中实现:

```
curl -vi -X GET \  
-H "Accept: text/xml" \  
"http://example.com:8000/A/schema"
```

在 JAVA 中实现:

```
HttpGet request = new HttpGet(url); // url =  
"http://example.com:8000/A/schema";  
request.setHeader("Content-Type", "text/xml");  
request.setHeader("Accept", "application/json"); // 接收方式可为 text/xml 或  
json 格式, 自己酌情选择
```

```
HttpResponse response = client.execute(request); // 执行请求
```

```
String strResult = EntityUtils.toString(response.getEntity()); // 读取服务器返回  
过来的 json 字符串数据
```

进行相应的解析

b)在新表中根据这 schema 进行建表(该出使用 HBASE JAVA API 方式建表, 传进去表名和列簇名即可)

```
HTableDescriptor tableDesc = new HTableDescriptor(A); // 新建一个表的描述  
tableDesc.addFamily(new HColumnDescriptor(columnFamily)); // 在描述里添  
加列簇信息
```

```
hAdmin.createTable(tableDesc); // 执行表的创建
```

c)在确认目标集群已有表的情况下进行表的数据迁移

首先使用 `http://example.com:8000/A/scanner` 方式扫描表获取 scanner\_id,  
再根据该 id 进行数据的获取

REST 在 Shell 中实现:

```
curl -vi -X PUT \
-H "Accept: text/xml" \
-H "Content-Type: text/xml" \
-d '<Scanner batch="1"/>' \
"http://example.com:8000/users/scanner/"
```

在 JAVA 中实现:

正常请求参数封装在请求头中使用 `addRequestHeader("Accept", "application/json")`

特殊信息使用 `setEntity` 方式

```
putMethod.setRequestEntity(new StringRequestEntity("<Scanner batch=\"" + cnt
+ "\"/>", null, null)); // cnt 为设定的每次返回数据的 cell 数而非 rowkey 的个数
```

解析相应信息, 写入目标集群吧(目标集群需开启 REST 方式)

REST 在 Shell 中实现:

```
curl -vi -X PUT \
-H "Accept: text/xml" \
-H "Content-Type: text/xml" \
-d '<?xml version="1.0" encoding="UTF-8" standalone="yes"?><CellSet><Row
key="cm93NQo="><Cell column="Y2Y6ZQo=">dmFsdWU1Cg==</Cell></Row></CellSet>' \
"http://example.com:8000/users/fakerow"
```

在 JAVA 中实现:

```
String result = httpUtils.httpPutToTDH(new_url, dataResult); // new_url:
http://example.com:8000/users/fakerow; dataResult:之前扫描回来的数据
```

为使用持续扫描数据持续写入新集群表, 需期间只是用一个 `scanner_id`, 依次扫描依次写入即可

在写入时需判定最后是否还有数据, 无数据了即停止写入(此处本人使用的是拿扫描时指定的 `cell` 个数 `cnt` 和查到的数据的 `cell` 总数做比较, =则继续, <则不再扫描)

**说明:**

以上则为使用 JAVA 代码实现 HBASE 集群间数据和表的迁移方式(主要技能: HttpClient、Hbase REST API、Hbase Java PAI)

**方案六:** 快照迁移(HBASE 0.94.6+才支持该功能, 默认是不开启的)

1)如版本支持快照却未开启, 请修改以下参数:

```
<property>
  <name>hbase.snapshot.enabled</name>
  <value>true</value>
</property>
```

2)生成快照:

```
$ ./bin/hbase shell
hbase> snapshot 'myTable', 'myTableSnapshot-122112'
```

3)用快照克隆表(这个克隆操作不涉及数据拷贝, 在新集群中操作):

```
$ hbase shell
hbase> clone_snapshot 'myTableSnapshot-122112', 'myNewTestTable'
```

4)导出到另一集群:

要拷贝一张叫 MySnapshot 的快照到 HBase 集群  
srv2(hdfs://dest\_cluster:8020/hbase)下, 使用 16 个 map task

```
$ sudo -u hbase hbase org.apache.hadoop.hbase.snapshot.ExportSnapshot -  
snapshot MySnapshot
```

```
-copy-to hdfs://dest_cluster:8020/hbase -mappers 16
```

说明: 这个 ExportSnapshot 工具拷贝了整个快照相关的所有的数据到另一个集群上

方案七: 将 HDFS 中 hbase 表文件落地再迁移(集群不通)

把某个表(table1)从集群 1 迁移到集群 2 (两个集群互相看不见), 步骤如下

1)拷贝集群 1 的表文件到本地磁盘:

```
/app/cloud/Hadoop/bin/hadoop fs -copyToLocal /hbase/table1 /data0/hbase-  
backup/table1
```

2)对于文件操作, 很简单吧, 随便你怎么去拷贝来拷贝去

3)如果集群 2 中也有对应的表文件, 那么删除掉, 然后拷贝

```
/app/cloud/hadoop/bin/hadoop fs -rmr /hbase/tablename  
/app/cloud/hadoop/bin/hadoop fs -copyFromLocal /data0/hbase-backup/tablename  
/hbase/tablename
```

4)重置该表在.META.表中的分区信息

```
bin/hbase org.jruby.Main bin/add_table.rb /hbase/tablename
```

另外:

1)如果表的数据量过大呢? 那么按照该表在 HDFS 中的文件夹数据, 分批拷贝。

2)如果两个集群可以互相通信呢? 采用第一种方案即可

附: <http://wiki.pentaho.com/display/BAD/Loading+Data+into+HBase> (本人没大看明白, 感兴趣的可以看看)

说明:

以上即为所接触的各种迁移 HBASE 到新集群的各种方法, 如需操作, 请针对自身情况酌情选择, 本人由于新旧集群的版本问题, 采用的是方案五(JAVA 代码方式实现), 本人联系方式: (58123441 《、扎啤》), 有问题欢迎沟通