

HBase 中加盐 (Salting) 之后的表如何读取 : MapReduce 篇

前两篇文章, [《HBase 中加盐 \(Salting\) 之后的表如何读取 : 协处理器篇》](#) 和 [《HBase 中加盐 \(Salting\) 之后的表如何读取 : Spark 篇》](#) 分别介绍了两种方法读取加盐之后的 HBase 表。本文将介绍如何在 MapReduce 读取加盐之后的表。

在 MapReduce 中也可以使用 [《HBase 中加盐 \(Salting\) 之后的表如何读取 : Spark 篇》](#) 文章里面的 SaltRangeTableInputFormat。剩余的就要求我们编写一个 Mapper, 来解析查询出来的数据, 代码如下 :

```
package com.iteblog.data.hadoop;

import org.apache.hadoop.hbase.Cell;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.io.ImmutableBytesWritable;
import org.apache.hadoop.hbase.mapreduce.TableMapper;
import org.apache.hadoop.hbase.util.Bytes;
import org.apache.hadoop.io.Text;

import java.io.IOException;
import java.util.List;
import java.util.StringJoiner;

public class HBaseMapper extends TableMapper<Text, Text> {
    @Override
    protected void map(ImmutableBytesWritable key,
                      Result value,
                      Context context) throws IOException, InterruptedException {

        String rowKey = Bytes.toString(value.getRow());
        List<Cell> cells = value.listCells();
        StringJoiner stringJoiner = new StringJoiner("\n");
        for (Cell cell : cells) {
            String family = Bytes.toString(cell.getFamilyArray(),
                                           cell.getFamilyOffset(), cell.getFamilyLength());

            String qualifier = Bytes.toString(cell.getQualifierArray(),
                                             cell.getQualifierOffset(), cell.getQualifierLength());

            String v = Bytes.toString(cell.getValueArray(),
                                     cell.getValueOffset(), cell.getValueLength());
```

```
stringJoiner.add("column=" + family + ":" + qualifier
    + ", timestamp=" + cell.getTimestamp() + ", value=" + v);
}

context.write(new Text(rowKey), new Text(stringJoiner.toString()));
}
}
```

这个 Mapper 程序将查询出来的数据进行解析，其代码和 [《HBase 中加盐 \(Salting\) 之后的表如何读取：协处理器篇》](#)

里面的协处理器服务端的数据解析很类似。因为表有多个列，为了显示方便，我这里使用 \n 分隔符来分割每列的数据。

在这个场景下，HBaseMapper 类已经将数据解析好了，所以我们不需要编写 Reducer，我们直接把数据存储到 HDFS 上，所以我们的驱动程序实现就很简单了，如下：

```
package com.iteblog.data.hadoop;

import com.iteblog.data.spark.SaltRangeTableInputFormat;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.client.Scan;
import org.apache.hadoop.hbase.mapreduce.TableInputFormat;
import org.apache.hadoop.hbase.mapreduce.TableMapReduceUtil;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.LazyOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

import java.io.IOException;

public class Hadoop {
    public static void main(String[] args)
        throws IOException, ClassNotFoundException, InterruptedException {
        Configuration conf = HBaseConfiguration.create();
        conf.set("hbase.zookeeper.quorum", "https://www.iteblog.com:2181");
        conf.set(TableInputFormat.SCAN_ROW_START, "1000");
        conf.set(TableInputFormat.SCAN_ROW_STOP, "1001");

        Job job = Job.getInstance(conf);
        job.setJobName("iteblog_HBase");
    }
}
```

```
job.setJarByClass(Hadoop.class);

job.setNumReduceTasks(1);
TableMapReduceUtil.initTableMapperJob("iteblog",
    new Scan(),
    HBaseMapper.class,
    Text.class,
    Text.class,
    job, true, SaltRangeTableInputFormat.class);

FileOutputFormat.setOutputPath(job, new Path("hdfs://www.iteblog.com:8020/result/"));
LazyOutputFormat.setOutputFormatClass(job, TextOutputFormat.class);

int result = job.waitForCompletion(true) ? 0 : 1;
}
}
```

最后我们编译打包上面的类，然后使用下面命令运行这个 MapReduce 程序

```
hadoop jar ~/hbase-1.0-SNAPSHOT.jar com.iteblog.data.hadoop.Hadoop
```

我们可以看到一共

Job Overview

Job Name: iteblog_HBase
State: RUNNING
Uberized: false
Started: Sat Feb 02 22:49:50 CST 2019
Elapsed: 4mins, 15sec

ApplicationMaster

Attempt Number	Start Time	Node	Logs
1	Sat Feb 02 22:49:39 CST 2019	https://www.iteblog.com:8042	logs

Task Type	Progress	Total	Pending	Running	Complete
Map		27	0	4	23
Reduce		1	0	1	0
Attempt Type	New	Running	Failed	Killed	Successful
Maps	0	4	0	1	23
Reduces	0	1	0	0	0

如果想及时了

解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

程序运行完之后，会在 `hdfs://www.iteblog.com:8020/result/` 目录下生产最终的结果，如下：

```
[root@master hadoop-2.7.7]# hadoop fs -ls /result
```

```
Found 2 items
```

```
-rw-r--r-- 1 iteblog supergroup      0 2019-02-02 22:54 /result/_SUCCESS
-rw-r--r-- 1 iteblog supergroup 14442 2019-02-02 22:54 /result/part-r-00000
```

```
A-1000-1550572395399 column=f:age, timestamp=1549091990253, value=54
column=f:uuid, timestamp=1549091990253, value=e9b10a9f-1218-43fd-bd01
A-1000-1550572413799 column=f:age, timestamp=1549092008575, value=4
column=f:uuid, timestamp=1549092008575, value=181aa91e-5f1d-454c-959c
A-1000-1550572414761 column=f:age, timestamp=1549092009531, value=33
column=f:uuid, timestamp=1549092009531, value=19aad8d3-621a-473c-8f9f
B-1000-1550572388491 column=f:age, timestamp=1549091983276, value=1
column=f:uuid, timestamp=1549091983276, value=cf720efe-2ad2-48d6-81b8
B-1000-1550572392922 column=f:age, timestamp=1549091987701, value=7
column=f:uuid, timestamp=1549091987701, value=8a047118-e130-48cb-adfe
B-1000-1550572424681 column=f:age, timestamp=1549092019451, value=57
column=f:uuid, timestamp=1549092019451, value=4217ab00-7cb9-4a81-bf29
C-1000-1550572390493 column=f:age, timestamp=1549091985284, value=89
column=f:uuid, timestamp=1549091985284, value=414d7df1-1925-4aaa-8298
```

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: **【】**（**）**