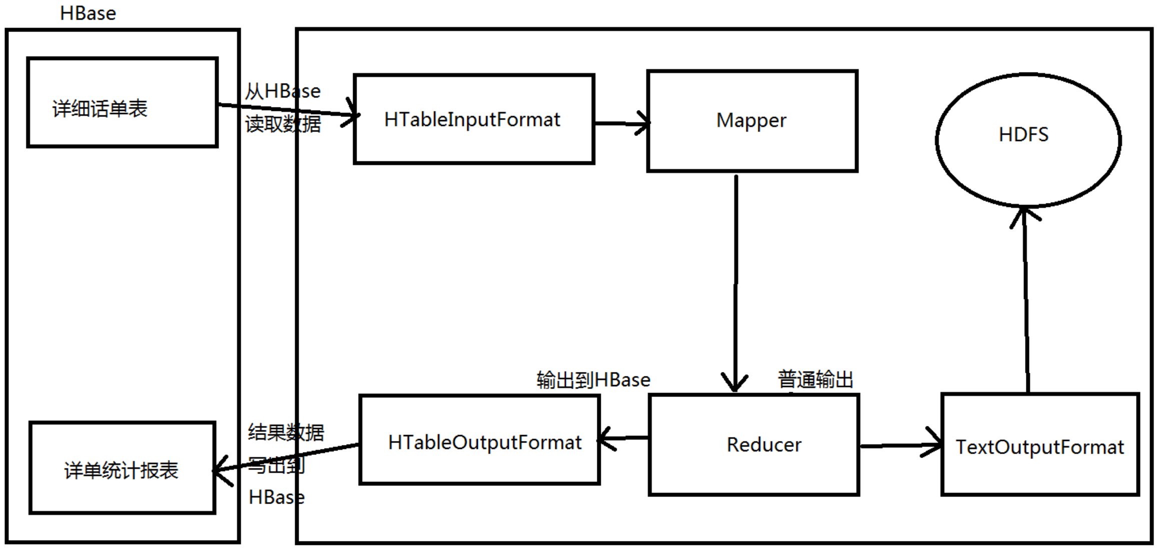
1. HBase结合MapReduce

为什么需要用MapReduce去访问HBase的数据？——加快分析速度和扩展分析能力，MapReduce访问HBase数据作分析一定是在离线分析的场景下应用



1.1. HBaseToHDFS：HBase表数据写入HDFS

从HBase中读取数据，分析之后然后写入HDFS，代码实现：

```java
package com.mazh.hbase.core.nx;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.hbase.Cell;
import org.apache.hadoop.hbase.CellUtil;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.client.Scan;
import org.apache.hadoop.hbase.io.ImmutableBytesWritable;
import org.apache.hadoop.hbase.mapreduce.TableMapReduceUtil;
import org.apache.hadoop.hbase.mapreduce.TableMapper;
import org.apache.hadoop.hbase.util.Bytes;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

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

/**
 * 作者： 马中华   https://blog.csdn.net/zhongqi2513
 * 时间： 2018/10/30 16:04
 * 描述： 编写 mapreduce 程序从 hbase 读取数据，然后存储到 hdfs
 */
public class HBaseDataToHDFSMR {

    public static final String ZK_CONNECT =
"bigdata02:2181,bigdata03:2181,bigdata04:2181";
    public static final String ZK_CONNECT_KEY = "hbase.zookeeper.quorum";

    public static final String HDFS_CONNECT = "hdfs://hadoop277ha/";
    public static final String HDFS_CONNECT_KEY = "fs.defaultFS";

    public static void main(String[] args) throws Exception {

        // 把Hadoop集群的配置文件：core-site.xml 和 hdfs-site.xml 放入 resources 目录
中。
        Configuration conf = HBaseConfiguration.create();
        conf.set(ZK_CONNECT_KEY, ZK_CONNECT);
        conf.set(HDFS_CONNECT_KEY, HDFS_CONNECT);
        System.setProperty("HADOOP_USER_NAME", "bigdata");

        Job job = Job.getInstance(conf);
        job.setJarByClass(HBaseDataToHDFSMR.class);

        // 输入数据来源于hbase的user_info表
        // 当前这句代码中有一个代码指示了：TableInputFormat
        Scan scan = new Scan();
        TableMapReduceUtil.initTableMapperJob("user_info", scan,
HBaseDataToHDFSMRMapper.class, Text.class,
                NullWritable.class, job);

        // RecordReader  --- TableRecordReader
        // InputFormat ----- TableInputFormat

        // 数据输出到hdfs
        FileOutputFormat.setOutputPath(job, new Path("/hbase2hdfs/output2"));

        boolean waitForCompletion = job.waitForCompletion(true);
        System.exit(waitForCompletion ? 0 : 1);
    }

    /**
     * mapper的输入key-value类型是：ImmutableBytesWritable, Result
     * mapper的输出key-value类型就可以由用户自己制定
     */
   static class HBaseDataToHDFSMRMapper extends TableMapper<Text, NullWritable>
{
        /**
         * keyType: LongWritable -- ImmutableBytesWritable:rowkey
         * ValueType: Text --  Result:hbase表中某一个rowkey查询出来的所有的key-value
对
         */
        @Override
        protected void map(ImmutableBytesWritable key, Result value,
Mapper.Context context) throws IOException,
                InterruptedException {

            //  byte[] rowkey = Bytes.copy(key, 0, key.getLength());
            String rowkey = Bytes.toString(key.copyBytes());
            List<Cell> listCells = value.listCells();
            Text text = new Text();

            // 最后输出格式是：rowkey，base_info:name-huangbo，base-info:age-34
            for (Cell cell : listCells) {
                String family = new String(CellUtil.cloneFamily(cell));
                String qualifier = new String(CellUtil.cloneQualifier(cell));
                String v = new String(CellUtil.cloneValue(cell));
                long ts = cell.getTimestamp();

                text.set(rowkey + "\t" + family + "\t" + qualifier + "\t" + v +
"\t" + ts);
                context.write(text, NullWritable.get());
            }
        }
    }
}
```

注意：打 jar 包提交到 Hadoop 集群运行。

1.2. HDFSToHBase：HDFS数据写入HBase

从HDFS从读入数据，处理之后写入HBase，代码实现：

```
package com.mazh.hbase.core.nx;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.mapreduce.TableMapReduceUtil;
import org.apache.hadoop.hbase.mapreduce.TableReducer;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.NullWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

import java.io.IOException;

/**
 * 作者：  马中华   https://blog.csdn.net/zhongqi2513
 * 时间：  2018/10/30 16:04
 * 需求：  读取HDFS上的数据。插入到HBase库中
 *
 * 程序运行之前，要先做两件事：
 * 1、把 students.txt 文件放入：/bigdata/student/input/ 目录中
 * 2、创建好一张 hbase 表： create "student", "info"
 */
public class HDFSDataToHBaseMR extends Configured implements Tool {
    public static void main(String[] args) throws Exception {
        int run = ToolRunner.run(new HDFSDataToHBaseMR(), args);
        System.exit(run);
    }

    @Override
    public int run(String[] arg0) throws Exception {

        Configuration config = HBaseConfiguration.create();
        config.set("hbase.zookeeper.quorum",
"bigdata02:2181,bigdata03:2181,bigdata04:2181");

        System.setProperty("HADOOP_USER_NAME", "bigdata");
        Job job = Job.getInstance(config, "HDFSDataToHBaseMR");

        job.setJarByClass(HDFSDataToHBaseMR.class);

        job.setMapperClass(HDFSDataToHBase_Mapper.class);
        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(NullWritable.class);

        // 设置数据的输出组件
        TableMapReduceUtil.initTableReducerJob("student",
HDFSDataToHBase_Reducer.class, job);

        job.setOutputKeyClass(NullWritable.class);
        job.setOutputValueClass(Put.class);

        FileInputFormat.addInputPath(job, new Path("/bigdata/student/input"));

        boolean isDone = job.waitForCompletion(true);
        return isDone ? 0 : 1;
    }

    static class HDFSDataToHBase_Mapper extends Mapper<LongWritable, Text, Text,
NullWritable> {

        @Override
        protected void map(LongWritable key, Text value, Context context) throws
IOException,
                InterruptedException {
            context.write(value, NullWritable.get());
        }
    }

    static class HDFSDataToHBase_Reducer extends TableReducer<Text,
NullWritable, NullWritable> {

        protected void reduce(Text key, Iterable<NullWritable> values,
Reducer.Context context) throws IOException,
                InterruptedException {
            String[] split = key.toString().split(",");

            Put put = new Put(split[0].getBytes());
            put.addColumn("info".getBytes(), "name".getBytes(),
split[1].getBytes());
            put.addColumn("info".getBytes(), "sex".getBytes(),
split[2].getBytes());
```

```
            put.addColumn("info".getBytes(), "age".getBytes(), 
split[3].getBytes());
            put.addColumn("info".getBytes(), "department".getBytes(), 
split[4].getBytes());

            context.write(NullWritable.get(), put);
        }
    }
}
```

注意：打 jar 包提交到 Hadoop 集群运行。