spark使用java读取hbase数据做分布式计算

由于spark提供的hbaseTest是scala版本，并没有提供java版。我将scala版本改为java版本，并根据数据做了些计算操作。

程序目的：查询出hbase满足条件的用户，统计各个等级个数。

代码如下，西面使用的hbase是0.94注释已经写详细：

kage com.sdyc.ndspark.sys;

import org.apache.commons.logging.Log;

import org.apache.commons.logging.LogFactory;

import org.apache.hadoop.conf.Configuration;

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.TableInputFormat;

import org.apache.hadoop.hbase.util.Base64;

import org.apache.hadoop.hbase.util.Bytes;

import org.apache.spark.api.java.JavaPairRDD;

import org.apache.spark.api.java.JavaSparkContext;

import org.apache.spark.api.java.function.Function2;

import org.apache.spark.api.java.function.PairFunction;

import scala.Tuple2;

import java.io.ByteArrayOutputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.Serializable;

import java.util.List;

/\*\*

 \* <pre>

 \*

 \* spark hbase 测试

 \*

 \*  Created with IntelliJ IDEA.

 \* User: zhangdonghao

 \* Date: 14-1-26

 \* Time: 上午9:24

 \* To change this template use File | Settings | File Templates.

 \* </pre>

 \*

 \* @author zhangdonghao

 \*/

public class HbaseTest implements Serializable {

    public Log log = LogFactory.getLog(HbaseTest.class);

    /\*\*

     \* 将scan编码，该方法copy自 org.apache.hadoop.hbase.mapreduce.TableMapReduceUtil

     \*

     \* @param scan

     \* @return

     \* @throws IOException

     \*/

    static String convertScanToString(Scan scan) throws IOException {

        ByteArrayOutputStream out = new ByteArrayOutputStream();

        DataOutputStream dos = new DataOutputStream(out);

        scan.write(dos);

        return Base64.encodeBytes(out.toByteArray());

    }

    public void start() {

        //初始化sparkContext，这里必须在jars参数里面放上Hbase的jar，

        // 否则会报unread block data异常

        JavaSparkContext sc = new JavaSparkContext("spark://nowledgedata-n3:7077", "hbaseTest",

                "/home/hadoop/software/spark-0.8.1",

                new String[]{"target/ndspark.jar", "target\\dependency\\hbase-0.94.6.jar"});

        //使用HBaseConfiguration.create()生成Configuration

        // 必须在项目classpath下放上hadoop以及hbase的配置文件。

        Configuration conf = HBaseConfiguration.create();

        //设置查询条件，这里值返回用户的等级

        Scan scan = new Scan();

        scan.setStartRow(Bytes.toBytes("195861-1035177490"));

        scan.setStopRow(Bytes.toBytes("195861-1072173147"));

        scan.addFamily(Bytes.toBytes("info"));

        scan.addColumn(Bytes.toBytes("info"), Bytes.toBytes("levelCode"));

        try {

            //需要读取的hbase表名

            String tableName = "usertable";

            conf.set(TableInputFormat.INPUT\_TABLE, tableName);

            conf.set(TableInputFormat.SCAN, convertScanToString(scan));

            //获得hbase查询结果Result

            JavaPairRDD<ImmutableBytesWritable, Result> hBaseRDD = sc.newAPIHadoopRDD(conf,

                    TableInputFormat.class, ImmutableBytesWritable.class,

                    Result.class);

            //从result中取出用户的等级，并且每一个算一次

            JavaPairRDD<Integer, Integer> levels = hBaseRDD.map(

                    new PairFunction<Tuple2<ImmutableBytesWritable, Result>, Integer, Integer>() {

                        @Override

                        public Tuple2<Integer, Integer> call(

                                Tuple2<ImmutableBytesWritable, Result> immutableBytesWritableResultTuple2)

                                throws Exception {

                            byte[] o = immutableBytesWritableResultTuple2.\_2().getValue(

                                    Bytes.toBytes("info"), Bytes.toBytes("levelCode"));

                            if (o != null) {

                                return new Tuple2<Integer, Integer>(Bytes.toInt(o), 1);

                            }

                            return null;

                        }

                    });

            //数据累加

            JavaPairRDD<Integer, Integer> counts = levels.reduceByKey(new Function2<Integer, Integer, Integer>() {

                public Integer call(Integer i1, Integer i2) {

                    return i1 + i2;

                }

            });

            //打印出最终结果

            List<Tuple2<Integer, Integer>> output = counts.collect();

            for (Tuple2 tuple : output) {

                System.out.println(tuple.\_1 + ": " + tuple.\_2);

            }

        } catch (Exception e) {

            log.warn(e);

        }

    }

    /\*\*

     \* spark如果计算没写在main里面,实现的类必须继承Serializable接口，<br>

     \* </>否则会报 Task not serializable: java.io.NotSerializableException 异常

     \*/

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

        new HbaseTest().start();

        System.exit(0);

    }

注意：如果使用的是hbase0.96.1.1-hadoop2

convertScanToString函数需要改为：

/\*\*

 \* 将scan编码，该方法copy自 org.apache.hadoop.hbase.mapreduce.TableMapReduceUtil

 \*

 \* @param scan

 \* @return

 \* @throws IOException

 \*/

static String convertScanToString(Scan scan) throws IOException {

    ClientProtos.Scan proto = ProtobufUtil.toScan(scan);

    return Base64.encodeBytes(proto.toByteArray());

}

运行结果如下：

0: 28528

11: 708

4: 28656

2: 36315

6: 23848

8: 19802

10: 6913

9: 15988

3: 31950

1: 38872

7: 21600

5: 27190

12: 17