Spark读取Hbase中的数据

Spark和Flume-ng整合，可以参见本博客：《Spark和Flume-ng整合》
《使用Spark读取HBase中的数据》

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

大家可能都知道很熟悉Spark的两种常见的数据读取方式（存放到RDD中）：（1）、调用parallelize函数直接从集合中获取数据，并存入RDD中；Java版本如下：

```
JavaRDD<Integer> myRDD = sc.parallelize(Arrays.asList(1,2,3));
```

Scala版本如下：

```
val myRDD= sc.parallelize(List(1,2,3))
```

这种方式很简单，很容易就可以将一个集合中的数据变成RDD的初始化值；更常见的是（2）、从文本中读取数据到RDD中，这个文本可以是纯文本文件、可以是sequence文件；可以存放在本地(file://)、可以存放在HDFS（hdfs://）上，还可以存放在S3上。其实对文件来说，Spark支持Hadoop所支持的所有文件类型和文件存放位置。Java版如下：

```
//////////////////////////////////////////////////////////////////////
 User: 过往记忆
 Date: 14-6-29
 Time: 23:59
 bolg:
 本文地址：/archives/1051
```

过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
过往记忆博客微信公共帐号：iteblog_hadoop

```
//////////////////////////////////////////////////////////////////////
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;

SparkConf conf = new SparkConf().setAppName("Simple Application");
JavaSparkContext sc = new JavaSparkContext(conf);
sc.addFile("wyp.data");
JavaRDD<String> lines = sc.textFile(SparkFiles.get("wyp.data"));
```

Scala版本如下：

```
import org.apache.spark.SparkContext
import org.apache.spark.SparkConf

val conf = new SparkConf().setAppName("Simple Application")
val sc = new SparkContext(conf)
sc.addFile("spam.data")
val inFile = sc.textFile(SparkFiles.get("spam.data"))
```

在实际情况下，我们需要的数据可能不是简单的存放在HDFS文本中，我们需要的数据可能就存放在Hbase中，那么我们如何用Spark来读取Hbase中的数据呢？本文的所有测试是基于Hadoop 2.2.0、Hbase 0.98.2、Spark 0.9.1，不同版本可能代码的编写有点不同。本文只是简单地用Spark来读取Hbase中的数据，如果需要对Hbase进行更强的操作，本文可能不能帮你。话不多说，Spark操作Hbase的核心的Java版本代码如下：

```
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.protobuf.ProtobufUtil;
import org.apache.hadoop.hbase.protobuf.generated.ClientProtos;
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;
```

```
//////////////////////////////////////////////////////////////////////
 User: 过往记忆
 Date: 14-6-29
 Time: 23:59
 bolg:
 本文地址：/archives/1051
 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
 过往记忆博客微信公共帐号：iteblog_hadoop
//////////////////////////////////////////////////////////////////////

JavaSparkContext sc = new JavaSparkContext(master, "hbaseTest",
            System.getenv("SPARK_HOME"), System.getenv("JARS"));

Configuration conf = HBaseConfiguration.create();
Scan scan = new Scan();
scan.addFamily(Bytes.toBytes("cf"));
scan.addColumn(Bytes.toBytes("cf"), Bytes.toBytes("airName"));

try {
      String tableName = "flight_wap_order_log";
      conf.set(TableInputFormat.INPUT_TABLE, tableName);
      ClientProtos.Scan proto = ProtobufUtil.toScan(scan);
      String ScanToString = Base64.encodeBytes(proto.toByteArray());
      conf.set(TableInputFormat.SCAN, ScanToString);

      JavaPairRDD<ImmutableBytesWritable, Result> myRDD =
            sc.newAPIHadoopRDD(conf,  TableInputFormat.class,
            ImmutableBytesWritable.class, Result.class);

catch (Exception e) {
        e.printStackTrace();
}
```

这样本段代码段是从Hbase表名为flight_wap_order_log的数据库中读取cf列簇上的airName一列的数据，这样我们就可以对myRDD进行相应的操作：

```
System.out.println(myRDD.count());
```

本段代码需要在pom.xml文件加入以下依赖：

```
<dependency>
        <groupId>org.apache.spark</groupId>
        <artifactId>spark-core_2.10</artifactId>
        <version>0.9.1</version>
</dependency>

<dependency>
        <groupId>org.apache.hbase</groupId>
        <artifactId>hbase</artifactId>
        <version>0.98.2-hadoop2</version>
</dependency>

<dependency>
        <groupId>org.apache.hbase</groupId>
        <artifactId>hbase-client</artifactId>
        <version>0.98.2-hadoop2</version>
</dependency>

<dependency>
        <groupId>org.apache.hbase</groupId>
        <artifactId>hbase-common</artifactId>
        <version>0.98.2-hadoop2</version>
</dependency>

<dependency>
        <groupId>org.apache.hbase</groupId>
        <artifactId>hbase-server</artifactId>
        <version>0.98.2-hadoop2</version>
</dependency>
```

Scala版如下：

```
import org.apache.spark._
import org.apache.spark.rdd.NewHadoopRDD
import org.apache.hadoop.hbase.{HBaseConfiguration, HTableDescriptor}
import org.apache.hadoop.hbase.client.HBaseAdmin
import org.apache.hadoop.hbase.mapreduce.TableInputFormat

////////////////////////////////////////////////////////////////////////
 User: 过往记忆
 Date: 14-6-29
 Time: 23:59
 bolg:
```

本文地址：/archives/1051
过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
过往记忆博客微信公共帐号：iteblog_hadoop
//

```
object HBaseTest {
  def main(args: Array[String]) {
    val sc = new SparkContext(args(0), "HBaseTest",
      System.getenv("SPARK_HOME"), SparkContext.jarOfClass(this.getClass))

    val conf = HBaseConfiguration.create()
    conf.set(TableInputFormat.INPUT_TABLE, args(1))

    val hBaseRDD = sc.newAPIHadoopRDD(conf, classOf[TableInputFormat],
      classOf[org.apache.hadoop.hbase.io.ImmutableBytesWritable],
      classOf[org.apache.hadoop.hbase.client.Result])

    hBaseRDD.count()

    System.exit(0)
  }
}
```

我们需要在加入如下依赖：

```
libraryDependencies ++= Seq(
        "org.apache.spark" % "spark-core_2.10" % "0.9.1",
        "org.apache.hbase" % "hbase" % "0.98.2-hadoop2",
        "org.apache.hbase" % "hbase-client" % "0.98.2-hadoop2",
        "org.apache.hbase" % "hbase-common" % "0.98.2-hadoop2",
        "org.apache.hbase" % "hbase-server" % "0.98.2-hadoop2"
)
```

在测试的时候，需要配置好Hbase、Hadoop环境，否则程序会出现问题，特别是让程序找到Hbase-site.xml配置文件。