Spark RDD写入RMDB(Mysql)方法二

在本博客的《Spark将计算结果写入到Mysql中》文章介绍了如果将Spark计算后的RDD最终写入到Mysql等关系型数据库中，但是这些写操作都是自己实现的，弄起来有点麻烦。不过值得高兴的是，前几天发布的Spark
1.3.0已经内置了读写关系型数据库的方法，我们可以直接在代码里面调用。

Spark 1.3.0中对数据库写操作是通过DataFrame类实现的，这个类也是新增的，是将之前的SchemaRDD重命名之后又定义了一些新方法的类。我们需要通过SQLContext来构造DataFrame对象，在SQLContext类中提供了大量可以构造DataFrame对象的方法，感兴趣的可以去看下。本文是通过SQLContext类中的createDataFrame方法来构造的。函数原型如下：

```
def createDataFrame(rowRDD: RDD[Row], schema: StructType): DataFrame
```

接收的RDD是Row类型的，他代表的是one row of output from a relational operator。第二个参数就是我们需要写入表的结构，包括了表的字段名和对应的类型，完整的代码如下：

```
import org.apache.spark.SparkContext
import org.apache.spark.sql.Row
import org.apache.spark.sql.types.{IntegerType, StringType, StructField, StructType}

/**
 * User: 过往记忆
 * Date: 15-03-17
 * Time: 上午07:24
 * bolg:
 * 本文地址：/archives/1290
 * 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 SparkToJDBC {
  def main(args: Array[String]): Unit = {

    val url = "jdbc:mysql://localhost:3306/spark?user=iteblog&password=iteblog"

    val sc = new SparkContext
    val sqlContext = new org.apache.spark.sql.SQLContext(sc)
    val schema = StructType(
      StructField("name", StringType) ::
        StructField("age", IntegerType)
```

```
      :: Nil)

    val data = sc.parallelize(List(("iteblog", 30), ("iteblog", 29),
       ("com", 40), ("bt", 33), ("www", 23))).
     map(item => Row.apply(item._1, item._2))
    import sqlContext.implicits._

    val df = sqlContext.createDataFrame(data, schema)
    df.createJDBCTable(url, "sparktomysql", false)

    sc.stop
  }
}
```

DataFrame类中提供了很多写数据库的操作，本例中的createJDBCTable就是可以创建表，它的函数原型如下：

```
def createJDBCTable(url: String, table: String, allowExisting: Boolean): Unit
```

table是表的名字，最后一个参数是如果表存在是否删除表的意思，false代表不删除。

DataFrame类中还有insertIntoJDBC方法，调用该函数必须保证表事先存在，它只用于插入数据，函数原型如下：

```
def insertIntoJDBC(url: String, table: String, overwrite: Boolean): Unit
```

前面两个参数和createJDBCTable一致，第三个参数如果设置为true，则在插入数据之前会调用mysql的TRUNCATE TABLE语句先清掉表中的数据。