Spark SQL整合PostgreSQL

本博客的《Spark与Mysql(JdbcRDD)整合开发》和《Spark RDD写入RMDB(Mysql)方法二》文章中介绍了如何通过Spark读写Mysql中的数据。

在生产环境下，很多公司都会使用PostgreSQL数据库，这篇文章将介绍如何通过Spark获取PostgreSQL中的数据。我将使用Spark 1.3中的DataFrame（也就是之前的SchemaRDD），我们可以通过SQLContext加载数据库中的数据，并转成DataFrame，我们可以使用SQLContext的load方法：

```
def load(source: String, options: Map[String, String]): DataFrame = {
   read.options(options).format(source).load()
}
```

其中，options可以传入的参数包括：url、dbtable、driver、partitionColumn、lowerBound、upperBound与numPartitions。

不过在Spark 1.4版本中，这个方法已经被标记为deprecated，我们得调用read.format(source).options(options).load()来替代。

那么我们的代码可以这么写：

```
/**
 * User: 过往记忆
 * Date: 2015-05-23
 * Time: 下午23:23
 * bolg:
 * 本文地址：/archives/1369
 * 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 IteblogApp {
  def main(args: Array[String]): Unit = {
    val sparkConf = new SparkConf().setAppName("IteblogpostgreSQL")
    val sc = new SparkContext(sparkConf)
    val sqlContext = new SQLContext(sc)

    val url = "jdbc:postgresql://www.iteblog.com:1234/test?user=iteblog&password=123456"
```

```
    val testDataFrame= sqlContext.load("jdbc", Map(
      "url" -> url,
      "driver" -> "org.postgresql.Driver",
      "dbtable" -> "SELECT * FROM iteblog"
    ))

    testDataFrame.foreach(println)
  }
}
```

在上面的使用中，我们是直接将SQL语句直接传入到dbtable中，但是很多情况下这还不符合我们的需求，不过，我们还可以通过调用registerTempTable()方法来注册临时表，并调用sql()方法执行查询：

```
val testDataFrame= sqlContext.load("jdbc", Map(
      "url" -> url,
      "driver" -> "org.postgresql.Driver",
      "dbtable" -> "iteblog"
))

testDataFrame.registerTempTable("iteblog")
sqlContext.sql("select * from iteblog").foreach(println)
```

最后，如果你使用的是SBT来管理项目，那么你需要在你的build.sbt文件中添加相关的依赖，如下：

```
libraryDependencies ++= {
    "org.apache.spark" % "spark-core_2.10"  % "1.3.1",
    "org.apache.spark" % "spark-sql_2.10"   % "1.3.1",
    "org.postgresql"   %  "postgresql"  % "9.4-1201-jdbc41"
}
```

如果你使用的是Maven，请在你的pom.xml文件里面加入以下依赖：

```
<dependencys>
 <dependency>
  <groupId>org.apache.spark</groupId>
```

```
 <artifactId>spark-core_2.10</artifactId>
 <version>1.3.1</version>
</dependency>

<dependency>
 <groupId>org.apache.spark</groupId>
 <artifactId>spark-sql_2.10</artifactId>
 <version>1.3.1</version>
</dependency>

<dependency>
 <groupId>org.postgresql</groupId>
 <artifactId>postgresql</artifactId>
 <version>9.4-1201-jdbc41</version>
</dependency>
</dependencys>
```