

Spark将计算结果写入到Mysql中

建议用Spark 1.3.0提供的写关系型数据库的方法，参见[《Spark RDD写入RMDB\(Mysql\)方法二》](#)。

在[《Spark与Mysql\(JdbcRDD\)整合开发》](#)

文章中我们介绍了如何通过Spark读取Mysql中的数据，当时写那篇文章的时候，Spark还未提供通过Java来使用JdbcRDD的API，不过目前的Spark提供了Java使用JdbcRDD的API。

今天主要来谈谈如果将Spark计算的结果写入到Mysql或者其他的关系型数据库里面。其实方式也很简单，代码如下：

```
/**
 * User: 过往记忆
 * Date: 15-03-10
 * Time: 上午07:30
 * blog:
 * 本文地址：/archives/1275
 * 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
 */
package scala

import java.sql.{DriverManager, PreparedStatement, Connection}
import org.apache.spark.{SparkContext, SparkConf}

object RDDtoMysql {

  case class Blog(name: String, count: Int)

  def myFun(iterator: Iterator[(String, Int)]): Unit = {
    var conn: Connection = null
    var ps: PreparedStatement = null
    val sql = "insert into blog(name, count) values (?, ?)"
    try {
      conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/spark",
        "root", "123456")
      iterator.foreach(data => {
        ps = conn.prepareStatement(sql)
        ps.setString(1, data._1)
        ps.setInt(2, data._2)
        ps.executeUpdate()
      })
    }
  }
}
```

```
} catch {  
  case e: Exception => println("Mysql Exception")  
} finally {  
  if (ps != null) {  
    ps.close()  
  }  
  if (conn != null) {  
    conn.close()  
  }  
}  
}  
}  
  
def main(args: Array[String]) {  
  val conf = new SparkConf().setAppName("RDDToMysql").setMaster("local")  
  val sc = new SparkContext(conf)  
  val data = sc.parallelize(List(("www", 10), ("iteblog", 20), ("com", 30)))  
  data.foreachPartition(myFun)  
}  
}
```

其实是通过foreachPartition遍历RDD的每个分区，并调用普通的Scala方法来写数据库。在运行程序之前需要确保数据库里面存在blog表，可以通过下面语句创建：

```
CREATE TABLE `blog` (  
  `name` varchar(255) NOT NULL,  
  `count` int(10) unsigned DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf-8
```

然后直接运行上述的代码即可。运行完成你就可以在数据库里面查询结果：

```
SELECT * FROM blog b;  
www      10  
iteblog   20  
com       30
```

需要注意的是：

1、你最好使用foreachPartition

函数来遍历RDD，并且在每台Work上面创建数据库的connection。

- 2、如果你的数据库并发受限，可以通过控制数据的分区来减少并发。
- 3、在插入Mysql的时候最好使用批量插入。
- 4、确保你写入数据库过程能够处理失败，因为你插入数据库的过程可能会经过网络，这可能会导致数据插入到数据库失败。
- 5、不建议将你的RDD数据写入到Mysql等关系型数据库中。

本博客文章除特别声明，全部都是原创！

原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。

本文链接: **【】** ()