5.3 数据读写

5.3.1 本地文件系统的数据读写

5.3.2分布式文件系统HDFS的数据读写

5.3.3 读写MySQL数据库

5.3.1 本地文件系统的数据读写



1.从文件中读取数据创建RDD

```
>>> textFile = sc.textFile("file:///usr/local/spark/mycode/rdd/word.txt")
>>> textFile.first()
'Hadoop is good'
```

因为Spark采用了惰性机制，在执行转换操作的时候，即使输入了错误的语句，spark-shell也不会马上报错（假设word123.txt不存在）

```
>>> textFile = sc.textFile("file:///usr/local/spark/mycode/wordcount/word123.txt")
```

5.3.1 本地文件系统的数据读写



2.把RDD写入到文本文件中

- 可以使用saveAsTextFile()方法把RDD中的数据保存到文本文件中。
- 需要注意的是，saveAsTextFile()中提供的参数，不是文件名称，而是一个目录名称。

- 因为Spark通常是在分布式环境下执行，RDD会存在多个分区，由多个任务对这些分区进行并行计算，每个分区的计算结果都会保存到一个单独的文件中。
- 比如，如果RDD有3个分区，saveAsTextFile()方法就会产生part-00000、part-00001和part-00002，以及一个_SUCCESS文件，其中，part-00000、part-00001和part-00002包含了RDD中的数据，_SUCCESS文件只是用来表示写入操作已经成功执行，该文件里面是空的，可以忽略该文件。

5.3.1 本地文件系统的数据读写



2.把RDD写入到文本文件中

下面把textFile变量中的内容再次写回到另外一个目录writeback中，命令如下：

```
>>> textFile = sc.textFile("file:///usr/local/spark/mycode/rdd/word.txt")
>>> textFile.saveAsTextFile("file:///usr/local/spark/mycode/rdd/writeback")
```

5.3.1 本地文件系统的数据读写



上面语句执行后，请打开一个新的Linux终端，
进入到“/usr/local/spark/mycode/rdd/”目录，
可以看到这个目录下新增了一个名称为“writeback”的子目录。
进入writeback子目录以后，可以看到该目录中生成了3个文件—part-00000、part-00001和_SUCCESS（可以忽略），part-00000和part-00001文件包含了刚才写入的数据。

5.3.2 分布式文件系统HDFS的数据读写



从分布式文件系统HDFS中读取数据，也是采用textFile()方法，
可以为textFile()方法提供一个HDFS文件或目录地址，
如果是一个文件地址，它会加载该文件，
如果是一个目录地址，它会加载该目录下的所有文件的数据

```
>>> textFile = sc.textFile("hdfs://hadoop01:9000/user/hadoop/word.txt")
>>> textFile.first()
```

同样，可以使用saveAsTextFile()方法把RDD中的数据保存到HDFS文件中，
命令如下：

```
>>> textFile = sc.textFile("word.txt")
>>> textFile.saveAsTextFile("writeback")
```

这时，在HDFS中就会生成一个目录“hdfs://hadoop01:9000/user/hadoop/writeback”。

5.3.2 分布式文件系统HDFS的数据读写



从分布式文件系统HDFS中读取数据，也是采用textFile()方法，可以为textFile()方法提供一个HDFS文件或目录地址，如果是一个文件地址，它会加载该文件，如果是一个目录地址，它会加载该目录下的所有文件的数据

```
>>> textFile = sc.textFile("hdfs://hadoop01:9000/user/hadoop/word.txt")
>>> textFile.first()
```

如下三条语句都是等价的：

```
>>> textFile = sc.textFile("hdfs://hadoop01:9000/user/hadoop/word.txt")
>>> textFile = sc.textFile("/user/hadoop/word.txt")
>>> textFile = sc.textFile("word.txt")
```

同样，可以使用saveAsTextFile()方法把RDD中的数据保存到HDFS文件中，命令如下：

```
>>> textFile.saveAsTextFile(“hdfs://hadoop01:9000/user/hadoop/writeback")
```

Hadoop HDFS常用操作命令表格



分类	命令	说明
目录查看	`hdfs dfs -ls /路径`	查看目录内容
	`hdfs dfs -ls -R /路径`	递归查看子目录（包含所有子目录和文件）
	`hdfs dfs -ls -h /路径`	以人性化格式显示文件大小（KB/MB/GB）
目录删除	`hdfs dfs -rm -r /路径`	安全删除（文件会进入回收站）
	`hdfs dfs -rm -r -skipTrash /路径`	永久删除（绕过回收站直接删除）
	`hdfs dfs -rmdir /空目录路径`	仅删除空目录

5.3.3 读写MySQL数据库



1.准备工作

在Linux中启动MySQL数据库，命令如下：

```
$ mysql -u root -p  #屏幕会提示输入密码
```

在MySQL Shell环境中，输入下面SQL语句完成数据库和表的创建：

```
mysql> create database henan;
mysql> use henan;
mysql> create table student (id int(4), name varchar(20), gender varchar(4), age int(4));
mysql> insert into student values(1,'Lee','F',23);
mysql> insert into student values(2,'Wang','M',24);
mysql> select * from student;
```

5.3.3 读写MySQL数据库



2.读取MySQL数据库中的数据

新建一个代码文件SparkReadMySQL.py，内容如下：

```
from pyspark import SparkConf, SparkContext
import pymysql
if __name__ == '__main__':
    conf = SparkConf().setAppName("MySQL RDD Example")
    sc = SparkContext(conf=conf)
    host = "localhost"
    port = 3306
    user = "root"
    password = "123456"
    database = "henan"
    table = "student"
```

5.3.3 读写MySQL数据库



2.读取MySQL数据库中的数据

新建一个代码文件SparkReadMySQL.py，内容如下：

```
# 创建MySQL连接
    conn = pymysql.connect(host=host, port=port, user=user, password=password, database=database)
    cursor = conn.cursor()
    # 执行SQL查询
    sql = "SELECT * FROM {}".format(table)
    cursor.execute(sql)
    # 将结果转换为RDD
    rdd = sc.parallelize(cursor.fetchall())
    rdd.foreach(print)
    sc.stop()
```

5.3.3 读写MySQL数据库



程序的执行结果如下：

```
(1, 'Lee', 'F', 23)
(2, 'Wang', 'M', 24)
```

5.3.3 读写MySQL数据库



3.向MySQL数据库中写入数据

```
from pyspark import SparkConf, SparkContext
import pymysql
if __name__ == '__main__':
    conf = SparkConf().setAppName("MySQL RDD Example")
    sc = SparkContext(conf=conf)
    rddData = sc.parallelize([(3, "Rongcheng", "M", 26), (4, "Guanhua", "M", 27)])
    def insertIntoMySQL(tuple):
        host = "localhost"
        port = 3306
        user = "root"
        password = "123456"
        database = "spark"
        table = "student"
        conn = pymysql.connect(host=host, port=port, user=user, password=password, database=database)
        cursor = conn.cursor()
```

5.3.3 读写MySQL数据库



3.向MySQL数据库中写入数据

```
        sql = "INSERT INTO {} (id, name, gender, age) VALUES (%d, '%s', '%s', %d)".format(table)
        value = (tuple[0], tuple[1], tuple[2], tuple[3])
        cursor.execute(sql % value)
        conn.commit()
        cursor.close()
        conn.close()

    rddData.foreach(insertIntoMySQL)
    sc.stop()
```

执行该程序以后，到MySQL中就可以看到新插入了两条数据。