作业 10

钟诚 16307110259

May 20, 2020

1 用 Spark 实现两个矩阵的乘法运算

用 Spark 实现两个矩阵的乘法运算（注意：不能使用 spark 已有的类似库函数实现，如 mllib 和 BlockMatrix 等）
输入文件的每一行为 < 矩阵名　行号　列号　值 > 例如某行为 A 1 2 4，则表示矩阵 A 第一行第二列的值为 4
输出的结果每一行为 < 行号　列号　值 >

1.1 分析

矩阵乘法的法则为，对两个长度分别为 I*K 和 K*J 的矩阵 A, B, 有

$$C_{ij} = \sum_{k=0}^{K} a_{ik} * b_{kj}$$

所以问题可以简化为，寻找 A 矩阵列号和 B 矩阵行号相同的值，再将这个值累加到 A 矩阵的行号和 B 矩阵列号对应的位置上。

因此，我们可以先使用 map 过程将 (A, i, k, A_{ik}) 化为 (k, (i, A_{ik}))，(B, k, j, B_{kj}) 化为 (k, (j, B_{kj}))，而对于对应位置的匹配，我们可以使用 A.join(B) 返回 A, B 中 k 相匹配的值，最后使用 reduce 过程将对应位置的值相加

1.2 程序

相应的程序如下:
Multiplie.py:

```
import pyspark
from pyspark import SparkContext, SparkConf
from operator import add

def multiplied(x):
    _, (A, B) =x
    value =[((row, col), v1*v2) for row, v1 in A for col,v2 in B]
    return value

# 使用SparkContext创建工作环境
conf =SparkConf().setAppName("Homework10").setMaster("local[*]")
sc =SparkContext.getOrCreate(conf)
# 将文本读到rdd
rdd =sc.textFile("file:///D:\Documents\课程介绍\大规模分布式系统\Project10\matrix.txt")
# 转换为列表并构建A, B矩阵
data =rdd.map(lambda x:x.split())
A =data.filter(lambda x:x[0] =='A').map(lambda x:(x[2], [(x[1], int(x[3]))])).reduceByKey(add)

B =data.filter(lambda x:x[0] =='B').map(lambda x:(x[2], [(x[1], int(x[3]))])).reduceByKey(add)
print(A)


# join operater will return an RDD containing all pairs of elements with matching keys
C =A.join(B).flatMap(multiplied).reduceByKey(add).collect()

with open('results.txt','w', encoding ='utf-8') as f:
    for value in C:
        f.writelines( value[0][0]+ ' ' +value[0][1] +' ' +str(value[1]) +'\n')
```

相应的结果储存在随文档提交的 results.txt 中