实验3-3 spark编程

0. scala安装

ubuntu: apt-get install scala 为了后面配置scala的环境变量，这里先找出scala的位置 通过which scala 可以查看到java的执行路径（不同于安装路径）

/usr/bin/scala

执行 ls -lrt /usr/bin/scala

执行 ls -lrt /etc/alternatives/scala，其显示的/usr/share/scala-2.11就是我们需要查询的scala安装目录

```
root@h0:/home/Hadoop/spark-2.4.4/conf# ls -lrt /usr/bin/scala
lrwxrwxrwx 1 root root 23 Dec  3 00:13 /usr/bin/scala -> /etc/altern
atives/scala
root@h0:/home/Hadoop/spark-2.4.4/conf# ls -lrt /etc/alternatives/sca
la
lrwxrwxrwx 1 root root 31 Dec  3 00:13 /etc/alternatives/scala -> /u
sr/share/scala-2.11/bin/scala
```

1. spark安装

版本&环境: Ubuntu 18 + java 1.8 +hadoop-3.2.1 ，使用集群模式，两个节点h0和h1，h0作为master，仅有h1作为worker

spark使用hadoop的resourcemanager分配资源

下载，解压，配置相关文件见https://www.jianshu.com/p/a4a0e7e4e4b7

其中spark-env.sh配置见下图

```
#!/usr/bin/env bash
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export SCALA_HOME=/usr/share/scala-2.11
export HADOOP_CONF_DIR=/home/Hadoop/hadoop-3.2.1/etc/hadoop
export HADOOP_HOME=/home/Hadoop/hadoop-3.2.1
#export HADOOP_HDFS_HOME=/usr/local/hadoop
export SPARK_HOME=/home/Hadoop/spark-2.4.4
export SPARK_MASTER_IP=h0
#export SPARK_MASTER_PORT=7077
export SPARK_MASTER_HOST=h0
#export SPARK_WORKER_PORT=8901
#export SPARK_WORKER_INSTANCES=1
export SPARK_MASTER_WEBUI_PORT=8080
```

start-all.sh

h0

```
root@h0:/home/Hadoop/spark-2.4.4# jps
1248 Jps
355 NameNode
1190 Master
824 ResourceManager
574 SecondaryNameNode
```

h1

```
root@h1:/home/Hadoop/spark-2.4.4# jps
24384 NodeManager
24274 DataNode
24582 Jps
23958 MainGenericRunner
24535 Worker
```

运行样例程序

```
root@h0:/home/Hadoop/spark-2.4.4# bin/run-example SparkPi 2>&1 | gre
p "Pi is"
Pi is roughly 3.1386956934784673
```

webUI

Spark Master at spark://h0:7077

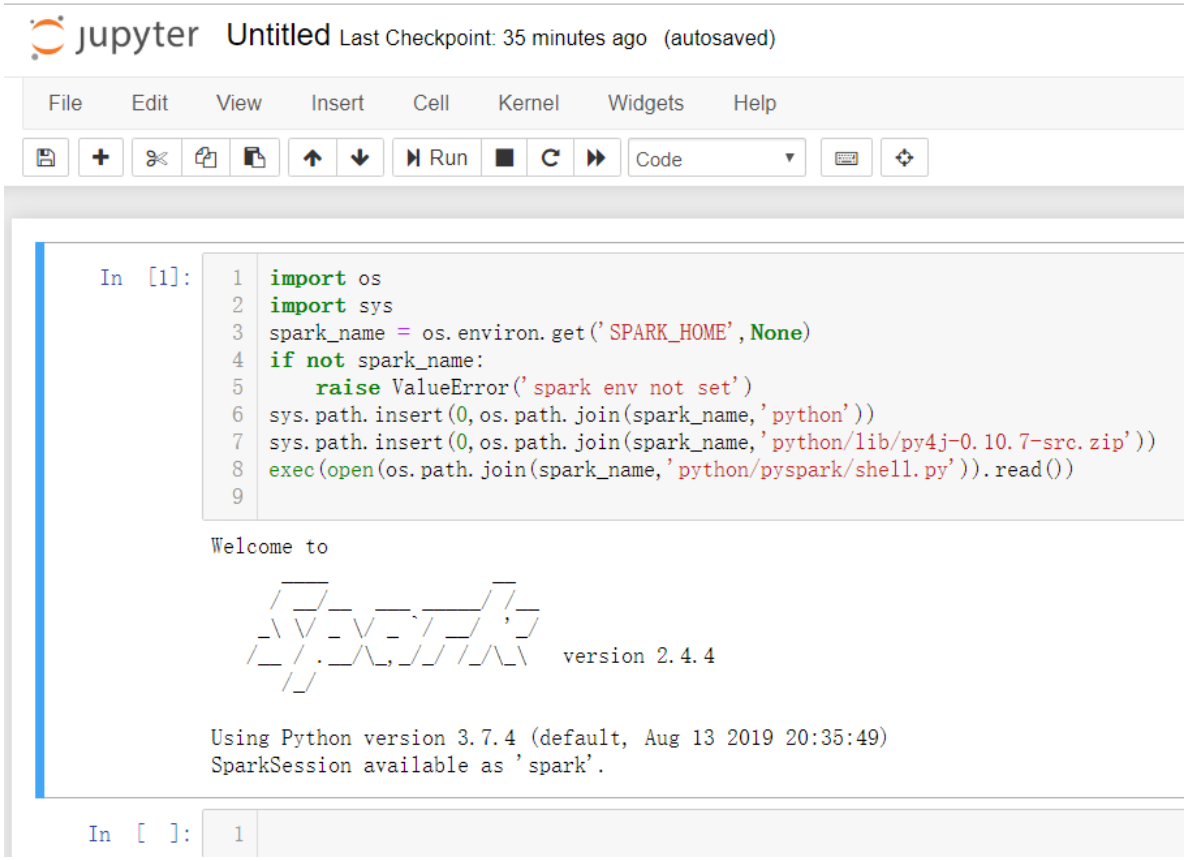
URL: spark://h0:7077
Alive Workers: 1
Cores in use: 1 Total, 0 Used
Memory in use: 1024.0 MB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

2. spark编程实践

2.1 pyspark配置及使用

实验环境：阿里云服务器，docker spark集群

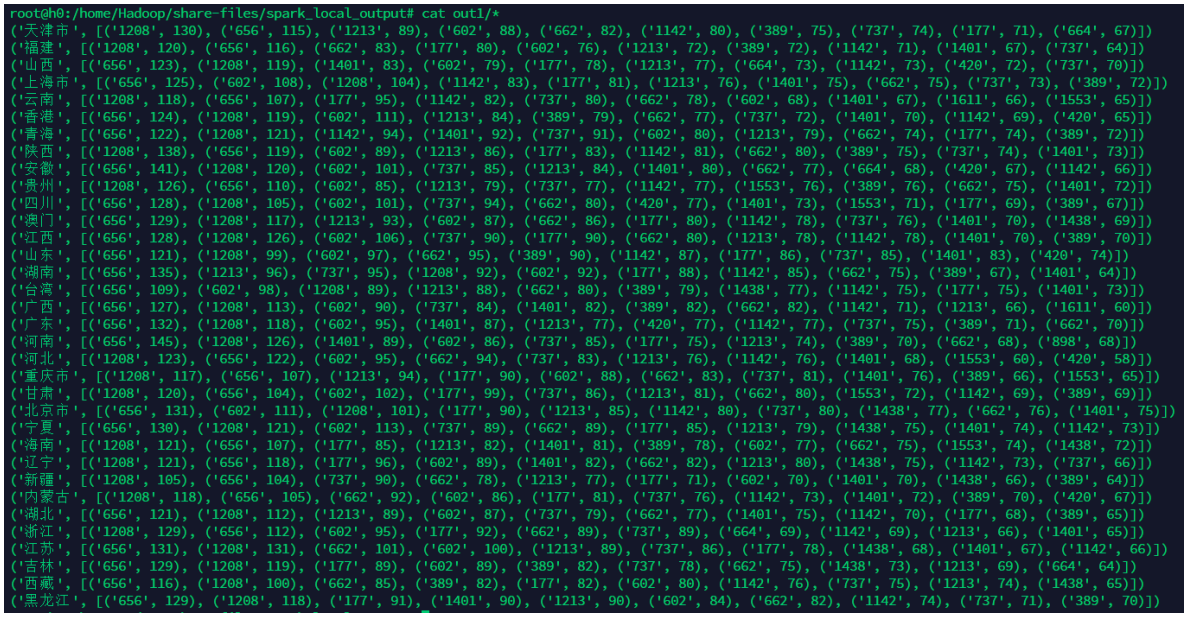
- 安装anaconda，并配置jupyter见教程https://www.jianshu.com/p/670486953d9e（因为是在docker中配置，还需要在创建容器的时候配置端口映射，在阿里云安全组中开放对应的端口，关闭防火墙）
- 配置jupyter的自动补全https://www.jianshu.com/p/0ab80f63af8a
- 运行jupyter，并在本机浏览器访问：xx.xx.xx.x:端口号 （注意使用chrome浏览器，默认edge不行）
- 配置pysparkhttps://blog.csdn.net/dxyna/article/details/79772343



下面就可以愉快的写代码啦

2.2 编程实践

- 统计各省销售最好的产品类别前十（每个省的数据中，第一个元素为商品类别，第二个是统计购买量）



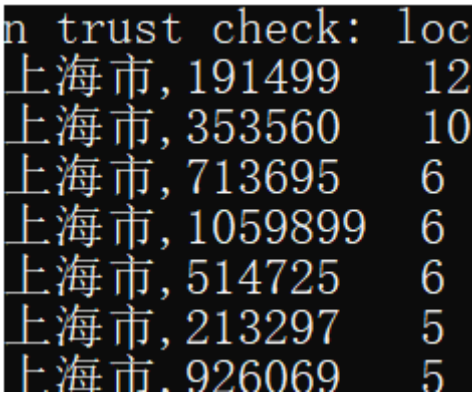

```
data=sc.textFile('file:///home/Hadoop/share-
files/million_user_log.csv')#transform 类操作
#Spark每次作行动操作时，都是从最初的转化操作开始计算；如果不想从头开始计算，想保存中间结果表，
就应该把数据载入缓存
data.cache()#相当于 .persist(MEMORY_ONLY) 将数据缓存存内存中
print(data.count())#count()为action类操作立即执行
print(data.first())
data=data.map(lambda x: list(x.split(',')))
```

```
data=data.filter(lambda x: x[7]=='2')
print(data.first())
data=data.map(lambda x: (x[10],x[2]))
print(data.first())
data=data.groupByKey()# 返回[(key,pyspark.resultiteratable),(,)]
data=data.mapValues(list)
#print(data.first())
def cat_10 (x):
    dct={}
    for key in x:
        dct[key]=dct.get(key,0)+1
    lst=sorted(dct.items(),key=lambda y:y[1],reverse=True)
    return lst[:10]
data=data.mapValues(cat_10)
print(data.first())
data.saveAsTextFile('file:///home/Hadoop/share-files/spark_local_output/out2')
#这里结果保存在本地，也可以选择saveAsHadoopData等保存在hdfs上
```

- 统计各省的双十一前十热门销售产品（购买最多前10的产品）-- 和MapReduce作业对比结果


```
代码同上一题，只需把(x[10],x[2]) 替换为(x[10],x[1])即可
```

与mapreduce对比



可以发现结果相同，但是用python 编写的spark程序简单非常非常多

- 查询双11那天浏览次数前十的品牌 -- 和Hive作业对比结果

    1. 使用RDD编程

```
data=sc.textFile('file:///home/Hadoop/share-
files/million_user_log.csv')
data.cache()
 data=data.map(lambda x: list(x.split(',')))
data=data.filter(lambda x: x[7]=='0')
 data=data.map(lambda x:x[4])
lst=sorted(data.countByValue().items(), key= lambda x:
x[1],reverse=True)
print(lst[:10])
```

结果

```
[('1360', 49151), ('3738', 10130), ('82', 9719), ('1446', 9426), ('6215', 8568), ('1214', 8470), ('5376', 828
2), ('2276', 7990), ('1662', 7808), ('8235', 7661)]
```

之前使用Hive的结果：

```
OK
1360    49151
3738    10130
82      9719
1446    9426
6215    8568
1214    8470
5376    8282
2276    7990
1662    7808
8235    7661
```

2. 此外还可以用dataframe进行查询（pyspark不支持Dataset，因为python本身不是一种类型安全的语言）

```
 #使用Dataframe+sparkSQL
from pyspark.sql.types import *
 #如果使用默认反射推断会将全部数据推断为String，并且淘宝数据没有header，这里用编程
指定类型
schema = StructType([
 StructField("user_id", StringType()),
 StructField("item_id",StringType()),
 StructField("cat_id", StringType()),
 StructField("merchant_id", StringType()),
 StructField("brand_id", StringType()),
 StructField("month", StringType()),
 StructField("day", StringType()),
 StructField("action", StringType()),
 StructField("age_range", StringType()),
 StructField("gender", StringType()),
 StructField("province", StringType())])

data1=spark.read.csv('file:///home/Hadoop/share-
files/million_user_log.csv',header=False,schema=schema)
data1.createOrReplaceTempView('taobao')
data1.show(5)
```

```
+-------+-------+------+-----------+--------+-----+---+------+---------+------+--------+
|user_id|item_id|cat_id|merchant_id|brand_id|month|day|action|age_range|gender|province|
+-------+-------+------+-----------+--------+-----+---+------+---------+------+--------+
| 328862| 406349|  1280|       2700|    5476|   11| 11|     0|        0|     1|    四川|
| 328862| 406349|  1280|       2700|    5476|   11| 11|     0|        7|     1|  重庆市|
| 328862| 807126|  1181|       1963|    6109|   11| 11|     0|        1|     0|  上海市|
| 328862| 406349|  1280|       2700|    5476|   11| 11|     2|        6|     0|    台湾|
| 328862| 406349|  1280|       2700|    5476|   11| 11|     0|        6|     2|    甘肃|
+-------+-------+------+-----------+--------+-----+---+------+---------+------+--------+
only showing top 5 rows
```

```
data2=spark.sql('select brand_id, count ( action ) actions from taobao
where action = 0 group by brand_id order by actions desc limit 10')
data2.show()
```

```
+--------+-------+
|brand_id|actions|
+--------+-------+
|    1360|  49151|
|    3738|  10130|
|      82|   9719|
|    1446|   9426|
|    6215|   8568|
|    1214|   8470|
|    5376|   8282|
|    2276|   7990|
|    1662|   7808|
|    8235|   7661|
+--------+-------+
```

参考教程：https://www.jianshu.com/p/cb0fec7a4f6d

3. 此外还可以是加载hiveContext然后运行SQL语句进行查询

```
from pyspark.sql import HiveContext
hive_context=HiveContext(sc)
hive_context.sql('select brand_id, count ( action ) actions from tUser
where action = 0 group by brand_id order by actions desc limit
10').show()
```