第8课：彻底实战详解使用IDE开发Spark程序--集群模式运行

原创 2016年01月30日 08:41:27 标签：spark 1386

第8课：彻底实战详解使用IDE开发Spark程序--集群模式运行
拷贝WordCount.scala生成WordCountCluster.scala。
1. 将object WordCount改为object WordCountCluster
2. 将conf.setMaster("local")行注释掉。在提交时再配置。
3. 将文件源修改为val lines = sc.textFile("hdfs://192.168.1.121:9000/user/spark/README.md")
4.启动hadoop和spark，并启动spark的history-server.sh
5.导出jar包。在WordCount Project上点击右键->export，选择java下的JAR file，点击Next。选择好导出路径后点击finish。
6.将导出的jar包拷贝到虚拟机中/home/richard/spark-1.6.0/class目录下。
7. 使用spark-submit提交运行jar包。
sprak-submit --class com.dt.spark.WordCountCluster --master spark://slq1:7077 /home/richard/spark-1.6.0/class/WordCount.jar

集群模式代码：

```
package com.dt.spark


import org.apache.spark.SparkConf
import org.apache.spark.SparkContext


/**
 * 使用scala开发集群运行的spark WordCount程序
 * DT大数据梦工厂
 * 新浪微博：http://weibo.com/ilovepains/
 */
object WordCountCluster {
  def main(args: Array[String]){
    /*
     * 第一步：创建spark的配置对象SparkConf，设置Spark程序运行时的配置信息
     * 例如通过setMaster来设置程序要链接的Spark集群的Master的URL，如果设置
     * 为local，则代表Spark程序在本地运行，特别适合于机器配置条件非常差（例如
     * 只有1内存）的初学者
     */

    val conf = new SparkConf()  //创建SparkConf对象。因为是全局唯一的，所以使用new，不用工厂方法模式。
    conf.setAppName("Wow, My First Spark App!")    //设置应用程序的名称，在程序运行的监控界面可以看到名称
//    conf.setMaster("local")  //此时程序在Spark集群。
    /**
     * 第二步：创建SparkContext对象，
     * SparkContext是Spark程序所有功能的唯一入口，无论是采用scala/java/Python/R等都必须有一个SParkContext，而且默认都只有一个。
     * SparkContext核心作用：初始化应用程序运行时所需要的核心组件，包括DAGScheduler,TaskScheduler,Scheduler Backend,
     * 同时还会负责Spark程序往Master注册程序等。SparkContext是整个Spark应用程序中最为重要的一个对象，
     *
     */
    val sc = new SparkContext(conf)     //通过创建SparkContext对象，通过传入SparkConf实例来定制SPark地的具体参数和配置信息。
    /*
     * 第三步：根据具体的数据来源（/HBase/Local FS/DB/S3等）通过SparkContext创建RDD，
     * RDD创建有三种基本方式：1.根据外部数据来源（如HDFS），2.根据Scala集合，3.由其他RDD操作产生
     * 数据会被RDD划分成为一系列的Partitions，分配到每个Partition的数据属于一个Task的处理范畴，
     */
    val lines = sc.textFile("hdfs://192.168.1.121:9000/user/spark/README.md")  //读取HDFS文件，并切分成不同的partitions
    //也可以写成：l lines:RDD[String] = sc.textFile    类型推断
    /**
     * 第4步：对初始RDD进行Transformation级别的处理。例如map/filter等高阶函数等的编程
     * 来进行具体的数据计算。第4.1步：将每一行的字符串拆分成单个的单词。
     */
     val words = lines.flatMap { line => line.split(" ") }   //对每一行的字符串进行单词拆分，map每次循环一行，将每一行的小集合通过flat合并成一个大集合
    /**
     * 第4.2步，在单词拆分的基础上对每个单词实例 进行计数为1，也就是word => (word,1)
     */
    val pairs = words.map { word => (word,1) }
    /**
     * 第4.3步，在每个单词实例计数为1的基础上，统计每个单词在文件中出现的总次数。
     */
    val wordCounts = pairs.reduceByKey(_+_)  //对相同的Key,进行Value的累计（包括Local和Reduce级别同时Reduce）
    wordCounts.collect.foreach(wordNumberPair => println(wordNumberPair._1 + " : " + wordNumberPair._2))
    sc.stop()    //把上下文去掉，释放资源

  }
}
```

运行时的log：

```
[richard@slq1 bin]$ ./spark-submit --class com.dt.spark.WordCountCluster --master spark://slq1:7077 /home/richard/spark-1.6.0/class/WordCount.jar
16/01/30 08:16:06 INFO spark.SparkContext: Running Spark version 1.6.0
16/01/30 08:16:09 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/01/30 08:16:12 INFO spark.SecurityManager: Changing view acls to: richard
16/01/30 08:16:12 INFO spark.SecurityManager: Changing modify acls to: richard
16/01/30 08:16:12 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(richard); users with modify permissions: Set(richard)
16/01/30 08:16:19 INFO util.Utils: Successfully started service 'sparkDriver' on port 34985.
16/01/30 08:16:23 INFO slf4j.Slf4jLogger: Slf4jLogger started
16/01/30 08:16:24 INFO Remoting: Starting remoting
16/01/30 08:16:26 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://sparkDriverActorSystem@192.168.1.121:33547]
16/01/30 08:16:26 INFO util.Utils: Successfully started service 'sparkDriverActorSystem' on port 33547.
16/01/30 08:16:26 INFO spark.SparkEnv: Registering MapOutputTracker
16/01/30 08:16:27 INFO spark.SparkEnv: Registering BlockManagerMaster
16/01/30 08:16:27 INFO storage.DiskBlockManager: Created local directory at /tmp/blockmgr-a020b1d8-f908-4473-852a-f6a55b545e02
16/01/30 08:16:27 INFO storage.MemoryStore: MemoryStore started with capacity 517.4 MB
16/01/30 08:16:28 INFO spark.SparkEnv: Registering OutputCommitCoordinator
16/01/30 08:16:31 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/01/30 08:16:32 INFO server.AbstractConnector: Started SelectChannelConnector@0.0.0.0:4040
16/01/30 08:16:32 INFO util.Utils: Successfully started service 'SparkUI' on port 4040.
```

梦飞天
+关注

原创	粉丝	喜欢	码云
116	0	0	未开通

他的最新文章 更多文章

第72课：Spark UDF和UDAF解密学习笔记

第71课：Spark SQL窗口函数解密与实战学习笔记

第70课:SparkSQL内置函数解密与实战学习笔记

热门文章

第70课:SparkSQL内置函数解密与实战学习笔记
4404

第69课：SparkSQL通过Hive数据源实战学习笔记
3588

第72课：Spark UDF和UDAF解密学习笔记
3472

第64课：SparkSQL下Parquet的数据切分和压缩内幕详解学习笔记
3411

第68课：SparkSQL JDBC实战详解学习笔记
3183

```
16/01/30 08:16:32 INFO ui.SparkUI: Started SparkUI at http://192.168.1.121:4040
16/01/30 08:16:32 INFO spark.HttpFileServer: HTTP File server directory is /tmp/spark-42b0d08b-938a-45fe-
8dae-81db69953855/httpd-ccd8a47c-1b7f-4ac6-b7e9-510b4f88c4b2
16/01/30 08:16:32 INFO spark.HttpServer: Starting HTTP Server
16/01/30 08:16:32 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/01/30 08:16:32 INFO server.AbstractConnector: Started SocketConnector@0.0.0.0:50966
16/01/30 08:16:32 INFO util.Utils: Successfully started service 'HTTP file server' on port 50966.
16/01/30 08:16:33 INFO spark.SparkContext: Added JAR file:/home/richard/spark-1.6.0/class/WordCount.jar at
http://192.168.1.121:50966/jars/WordCount.jar with timestamp 1454112993094
16/01/30 08:16:34 INFO client.AppClient$ClientEndpoint: Connecting to master spark://slq1:7077...
16/01/30 08:16:37 INFO cluster.SparkDeploySchedulerBackend: Connected to Spark cluster with app ID app-
20160130081636-0001
16/01/30 08:16:37 INFO client.AppClient$ClientEndpoint: Executor added: app-20160130081636-0001/0 on
worker-20160130074034-192.168.1.123-57185 (192.168.1.123:57185) with 1 cores
16/01/30 08:16:37 INFO cluster.SparkDeploySchedulerBackend: Granted executor ID app-20160130081636-
0001/0 on hostPort 192.168.1.123:57185 with 1 cores, 1024.0 MB RAM
16/01/30 08:16:37 INFO client.AppClient$ClientEndpoint: Executor added: app-20160130081636-0001/1 on
worker-20160130074035-192.168.1.122-37406 (192.168.1.122:37406) with 1 cores
16/01/30 08:16:37 INFO cluster.SparkDeploySchedulerBackend: Granted executor ID app-20160130081636-
0001/1 on hostPort 192.168.1.122:37406 with 1 cores, 1024.0 MB RAM
16/01/30 08:16:37 INFO client.AppClient$ClientEndpoint: Executor added: app-20160130081636-0001/2 on
worker-20160130074053-192.168.1.121-45928 (192.168.1.121:45928) with 1 cores
16/01/30 08:16:37 INFO cluster.SparkDeploySchedulerBackend: Granted executor ID app-20160130081636-
0001/2 on hostPort 192.168.1.121:45928 with 1 cores, 1024.0 MB RAM
16/01/30 08:16:37 INFO util.Utils: Successfully started service
'org.apache.spark.network.netty.NettyBlockTransferService' on port 53088.
16/01/30 08:16:37 INFO netty.NettyBlockTransferService: Server created on 53088
16/01/30 08:16:37 INFO storage.BlockManagerMaster: Trying to register BlockManager
16/01/30 08:16:37 INFO storage.BlockManagerMasterEndpoint: Registering block manager 192.168.1.121:53088
with 517.4 MB RAM, BlockManagerId(driver, 192.168.1.121, 53088)
16/01/30 08:16:37 INFO storage.BlockManagerMaster: Registered BlockManager
16/01/30 08:16:38 INFO client.AppClient$ClientEndpoint: Executor updated: app-20160130081636-0001/0 is now
RUNNING
16/01/30 08:16:38 INFO client.AppClient$ClientEndpoint: Executor updated: app-20160130081636-0001/1 is now
RUNNING
16/01/30 08:16:38 INFO client.AppClient$ClientEndpoint: Executor updated: app-20160130081636-0001/2 is now
RUNNING
16/01/30 08:16:43 INFO cluster.SparkDeploySchedulerBackend: SchedulerBackend is ready for scheduling
beginning after reached minRegisteredResourcesRatio: 0.0
16/01/30 08:17:00 INFO storage.MemoryStore: Block broadcast_0 stored as values in memory (estimated size
127.8 KB, free 127.8 KB)
16/01/30 08:17:01 INFO cluster.SparkDeploySchedulerBackend: Registered executor NettyRpcEndpointRef(null)
(slq3:52579) with ID 0
16/01/30 08:17:02 INFO storage.BlockManagerMasterEndpoint: Registering block manager slq3:54589 with 517.4
MB RAM, BlockManagerId(0, slq3, 54589)
16/01/30 08:17:02 INFO storage.MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated
size 14.3 KB, free 142.1 KB)
16/01/30 08:17:02 INFO cluster.SparkDeploySchedulerBackend: Registered executor NettyRpcEndpointRef(null)
(slq2:52779) with ID 1
16/01/30 08:17:02 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on
192.168.1.121:53088 (size: 14.3 KB, free: 517.4 MB)
16/01/30 08:17:03 INFO spark.SparkContext: Created broadcast 0 from textFile at WordCountCluster.scala:36
16/01/30 08:17:03 INFO storage.BlockManagerMasterEndpoint: Registering block manager slq2:38295 with 517.4
MB RAM, BlockManagerId(1, slq2, 38295)
16/01/30 08:17:23 INFO mapred.FileInputFormat: Total input paths to process : 1
16/01/30 08:17:26 INFO spark.SparkContext: Starting job: collect at WordCountCluster.scala:51
16/01/30 08:17:27 INFO scheduler.DAGScheduler: Registering RDD 3 (map at WordCountCluster.scala:46)
16/01/30 08:17:27 INFO scheduler.DAGScheduler: Got job 0 (collect at WordCountCluster.scala:51) with 2 output
partitions
16/01/30 08:17:27 INFO scheduler.DAGScheduler: Final stage: ResultStage 1 (collect at
WordCountCluster.scala:51)
16/01/30 08:17:27 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 0)
16/01/30 08:17:27 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 0)
16/01/30 08:17:27 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 0 (MapPartitionsRDD[3] at map
at WordCountCluster.scala:46), which has no missing parents
16/01/30 08:17:28 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 4.1
KB, free 146.2 KB)
16/01/30 08:17:29 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated
size 2.3 KB, free 148.5 KB)
16/01/30 08:17:29 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on
192.168.1.121:53088 (size: 2.3 KB, free: 517.4 MB)
16/01/30 08:17:29 INFO spark.SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1006
16/01/30 08:17:29 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ShuffleMapStage 0
(MapPartitionsRDD[3] at map at WordCountCluster.scala:46)
16/01/30 08:17:29 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 2 tasks
16/01/30 08:17:30 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, slq3, partition
0,NODE_LOCAL, 2192 bytes)
16/01/30 08:17:30 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, slq2, partition
1,NODE_LOCAL, 2192 bytes)
16/01/30 08:17:34 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on slq2:38295 (size:
2.3 KB, free: 517.4 MB)
16/01/30 08:17:34 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on slq3:54589 (size:
2.3 KB, free: 517.4 MB)
16/01/30 08:17:36 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on slq2:38295 (size:
14.3 KB, free: 517.4 MB)
16/01/30 08:17:37 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on slq3:54589 (size:
14.3 KB, free: 517.4 MB)
16/01/30 08:17:47 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 18118 ms on slq3
(1/2)
16/01/30 08:17:47 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 17770 ms on slq2
(2/2)
16/01/30 08:17:48 INFO scheduler.DAGScheduler: ShuffleMapStage 0 (map at WordCountCluster.scala:46)
finished in 18.399 s
16/01/30 08:17:48 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed,
from pool
16/01/30 08:17:48 INFO scheduler.DAGScheduler: looking for newly runnable stages
16/01/30 08:17:48 INFO scheduler.DAGScheduler: running: Set()
16/01/30 08:17:48 INFO scheduler.DAGScheduler: waiting: Set(ResultStage 1)
16/01/30 08:17:48 INFO scheduler.DAGScheduler: failed: Set()
16/01/30 08:17:48 INFO scheduler.DAGScheduler: Submitting ResultStage 1 (ShuffledRDD[4] at reduceByKey at
WordCountCluster.scala:50), which has no missing parents
16/01/30 08:17:48 INFO storage.MemoryStore: Block broadcast_2 stored as values in memory (estimated size 2.6
KB, free 151.1 KB)
16/01/30 08:17:48 INFO storage.MemoryStore: Block broadcast_2_piece0 stored as bytes in memory (estimated
size 1593.0 B, free 152.6 KB)
16/01/30 08:17:48 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on
192.168.1.121:53088 (size: 1593.0 B, free: 517.4 MB)
16/01/30 08:17:48 INFO spark.SparkContext: Created broadcast 2 from broadcast at DAGScheduler.scala:1006
16/01/30 08:17:48 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 1 (ShuffledRDD[4]
at reduceByKey at WordCountCluster.scala:50)
16/01/30 08:17:48 INFO scheduler.TaskSchedulerImpl: Adding task set 1.0 with 2 tasks
16/01/30 08:17:48 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 1.0 (TID 2, slq3, partition
0,NODE_LOCAL, 1949 bytes)
```

```
16/01/30 08:17:48 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 1.0 (TID 3, slq2, partition
1,NODE_LOCAL, 1949 bytes)
16/01/30 08:17:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on slq2:38295 (size:
1593.0 B, free: 517.4 MB)
16/01/30 08:17:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on slq3:54589 (size:
1593.0 B, free: 517.4 MB)
16/01/30 08:17:49 INFO spark.MapOutputTrackerMasterEndpoint: Asked to send map output locations for
shuffle 0 to slq2:52779
16/01/30 08:17:49 INFO spark.MapOutputTrackerMaster: Size of output statuses for shuffle 0 is 151 bytes
16/01/30 08:17:49 INFO spark.MapOutputTrackerMasterEndpoint: Asked to send map output locations for
shuffle 0 to slq3:52579
16/01/30 08:17:51 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 1.0 (TID 3) in 2404 ms on slq2 (1/2)
16/01/30 08:17:51 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 1.0 (TID 2) in 2531 ms on slq3 (2/2)
16/01/30 08:17:51 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed,
from pool
16/01/30 08:17:51 INFO scheduler.DAGScheduler: ResultStage 1 (collect at WordCountCluster.scala:51) finished in
2.538 s
16/01/30 08:17:51 INFO scheduler.DAGScheduler: Job 0 finished: collect at WordCountCluster.scala:51, took
24.676312 s
package : 1
this : 1
Version"](http://spark.apache.org/docs/latest/building-spark.html#specifying-the-hadoop-version) : 1
Because : 1
Python : 2
cluster. : 1
its : 1
[run : 1
general : 2
have : 1
pre-built : 1
locally. : 1
locally : 2
changed : 1
sc.parallelize(1 : 1
only : 1
several : 1
This : 2
basic : 1
Configuration : 1
learning, : 1
documentation : 3
YARN, : 1
graph : 1
Hive : 2
first : 1
["Specifying : 1
"yarn-client" : 1
page](http://spark.apache.org/documentation.html) : 1
[params]`. : 1
application : 1
[project : 2
prefer : 1
SparkPi : 2
<http://spark.apache.org/> : 1
engine : 1
version : 1
file : 1
documentation, : 1
MASTER : 1
example : 3
distribution. : 1
are : 1
params : 1
scala> : 1
systems. : 1
provides : 1
refer : 2
configure : 1
Interactive : 2
can : 6
build : 3
when : 1
easiest : 1
Apache : 1
Distributions"](http://spark.apache.org/docs/latest/hadoop-third-party-distributions.html) : 1
works : 1
how : 2
package. : 1
1000).count() : 1
Note : 1
Data. : 1
>>> : 1
Scala : 2
Alternatively, : 1
variable : 1
submit : 1
Testing : 1
Streaming : 1
thread, : 1
rich : 1
them, : 1
detailed : 2
stream : 1
GraphX : 1
tests](https://cwiki.apache.org/confluence/display/SPARK/Contributing+to+Spark#ContributingtoSpark-
AutomatedTesting). : 1
distribution : 1
Please : 3
return : 2
is : 6
Thriftserver : 1
["Third : 1
same : 1
start : 1
built : 1
one : 2
with : 4
Party : 1
Spark](#building-spark). : 1
Spark"](http://spark.apache.org/docs/latest/building-spark.html). : 1
data : 2
wiki](https://cwiki.apache.org/confluence/display/SPARK). : 1
using : 2
```

```
automated : 1
talk : 1
Shell : 2
class : 2
README : 1
computing : 1
Python, : 2
example: : 1
## : 8
from : 1
set : 2
building : 3
N : 1
Hadoop-supported : 1
other : 1
Example : 1
analysis. : 1
runs. : 1
Building : 1
higher-level : 1
need : 1
Big : 1
fast : 1
guide, : 1
Java, : 1
<class> : 1
uses : 1
SQL : 2
will : 1
guidance : 3
requires : 1
 : 66
Documentation : 1
web : 1
cluster : 2
using: : 1
MLlib : 1
shell: : 2
Scala, : 1
supports : 2
built, : 1
./dev/run-tests : 1
sample : 1
For : 2
Spark : 14
particular : 3
Programs : 1
The : 1
processing. : 1
APIs : 1
computation : 1
Try : 1
[Configuration : 1
library : 1
A : 1
through : 1
# : 1
./bin/pyspark : 1
following : 2
"yarn-cluster" : 1
More : 1
which : 2
See : 1
also : 5
storage : 1
for : 11
should : 2
To : 2
Once : 1
mesos:// : 1
setup : 1
Maven](http://maven.apache.org/). : 1
your : 1
latest : 1
processing, : 2
the : 21
not : 1
different : 1
guide](http://spark.apache.org/docs/latest/configuration.html) : 1
distributions. : 1
given. : 1
About : 1
if : 4
instructions. : 1
be : 2
Tests : 1
do : 2
no : 1
all : 1
./bin/run-example : 2
programs, : 1
including : 3
`./bin/run-example : 1
Spark. : 1
Versions : 1
HDFS : 1
spark:// : 1
It : 2
an : 3
programming : 1
machine : 1
environment : 1
run: : 1
clean : 1
1000: : 2
And : 1
run : 7
URL, : 1
./bin/spark-shell : 1
threads. : 1
"local" : 1
MASTER=spark://host:7077 : 1
on : 6
```

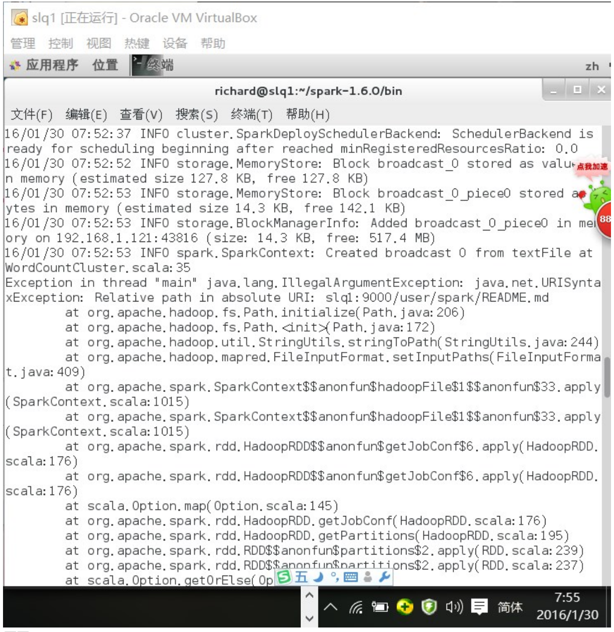
```
You : 3
against : 1
help : 1
print : 1
tests : 1
[Apache : 1
examples : 2
at : 2
in : 5
-DskipTests : 1
usage : 1
downloaded : 1
versions : 1
online : 1
graphs : 1
optimized : 1
abbreviated : 1
comes : 1
directory. : 1
overview : 1
[building : 1
`examples` : 2
Many : 1
Running : 1
way : 1
use : 3
Online : 1
site, : 1
running : 1
find : 1
sc.parallelize(range(1000)).count() : 1
contains : 1
you : 4
project : 1
name : 1
protocols : 1
that : 3
a : 9
or : 3
high-level : 1
Pi : 1
Hadoop, : 2
to : 14
available : 1
core : 1
(You : 1
instance: : 1
see : 1
of : 5
tools : 1
"local[N]" : 1
programs : 2
structured : 1
package.) : 1
["Building : 1
must : 1
and : 10
command, : 2
system : 1
mvn : 1
Hadoop : 4
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/metrics/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/stage/kill,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/api,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/static,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/threadDump/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/threadDump,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/executors,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/environment/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/environment,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/rdd/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/rdd,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/storage,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/pool/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/pool,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/stage/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/stage,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/stages,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/job/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/job,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs/json,null}
16/01/30 08:17:52 INFO handler.ContextHandler: stopped o.s.j.s.ServletContextHandler{/jobs,null}
16/01/30 08:17:52 INFO ui.SparkUI: Stopped Spark web UI at http://192.168.1.121:4040
16/01/30 08:17:52 INFO cluster.SparkDeploySchedulerBackend: Shutting down all executors
16/01/30 08:17:52 INFO cluster.SparkDeploySchedulerBackend: Asking each executor to shut down
16/01/30 08:17:52 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
16/01/30 08:17:53 INFO storage.MemoryStore: MemoryStore cleared
16/01/30 08:17:53 INFO storage.BlockManager: BlockManager stopped
16/01/30 08:17:53 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
16/01/30 08:17:53 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/01/30 08:17:53 INFO spark.SparkContext: Successfully stopped SparkContext
16/01/30 08:17:53 INFO remote.RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
16/01/30 08:17:53 INFO remote.RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
16/01/30 08:17:54 INFO util.ShutdownHookManager: Shutdown hook called
16/01/30 08:17:54 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-42b0d08b-938a-45fe-8dae-81db69953855/httpd-ccd8a47c-1b7f-4ac6-b7e9-510b4f88c4b2
16/01/30 08:17:54 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-42b0d08b-938a-45fe-8dae-81db69953855
16/01/30 08:17:55 INFO remote.RemoteActorRefProvider$RemotingTerminator: Remoting shut down.
[richard@slq1 bin]$
```

小插曲：

第一次运行时报错：

```
Exception in thread "main" java.lang.IllegalArgumentException: java.net.URISyntaxException: Relative path in
```

absolute URI: slq1:9000/user/spark/README.md。


slq1 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
应用程序 位置 终端
zh
richard@slq1:~/spark-1.6.0/bin
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
点我加速
88
简体
7:55
2016/1/30

```
16/01/30 07:52:37 INFO cluster.SparkDeploySchedulerBackend: SchedulerBackend is
ready for scheduling beginning after reached minRegisteredResourcesRatio: 0.0
16/01/30 07:52:52 INFO storage.MemoryStore: Block broadcast_0 stored as valu
n memory (estimated size 127.8 KB, free 127.8 KB)
16/01/30 07:52:53 INFO storage.MemoryStore: Block broadcast_0_piece0 stored a
ytes in memory (estimated size 14.3 KB, free 142.1 KB)
16/01/30 07:52:53 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in me
ory on 192.168.1.121:43816 (size: 14.3 KB, free: 517.4 MB)
16/01/30 07:52:53 INFO spark.SparkContext: Created broadcast 0 from textFile at
WordCountCluster.scala:35
Exception in thread "main" java.lang.IllegalArgumentException: java.net.URISynta
xException: Relative path in absolute URI: slq1:9000/user/spark/README.md
        at org.apache.hadoop.fs.Path.initialize(Path.java:206)
        at org.apache.hadoop.fs.Path.<init>(Path.java:172)
        at org.apache.hadoop.util.StringUtils.stringToPath(StringUtils.java:244)
        at org.apache.hadoop.mapred.FileInputFormat.setInputPaths(FileInputForma
t.java:409)
        at org.apache.spark.SparkContext$$anonfun$hadoopFile$1$$anonfun$33.apply
(SparkContext.scala:1015)
        at org.apache.spark.SparkContext$$anonfun$hadoopFile$1$$anonfun$33.apply
(SparkContext.scala:1015)
        at org.apache.spark.rdd.HadoopRDD$$anonfun$getJobConf$6.apply(HadoopRDD.
scala:176)
        at org.apache.spark.rdd.HadoopRDD$$anonfun$getJobConf$6.apply(HadoopRDD.
scala:176)
        at scala.Option.map(Option.scala:145)
        at org.apache.spark.rdd.HadoopRDD.getJobConf(HadoopRDD.scala:176)
        at org.apache.spark.rdd.HadoopRDD.getPartitions(HadoopRDD.scala:195)
        at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:239)
        at org.apache.spark.rdd.RDD$$anonfun$partitions$2.apply(RDD.scala:237)
        at scala.Option.getOrElse(Op
```

原因：

val lines = sc.textFile("hdfs://192.168.1.121:9000/user/spark/README.md") 原来写成相对路径：/user/spark/README.md，导致程序找不到文件。

以上内容是王家林老师DT大数据梦工厂《 IMF传奇行动》第8课的学习笔记。

王家林：Spark、Flink、Docker、Android技术中国区布道师。Spark亚太研究院院长和首席专家，DT大数据梦工厂创始人，Android软硬整合源码级专家，英语发音魔术师，健身狂热爱好者。

微信公众账号：DT_Spark

联系邮箱18610086859@126.com

电话：18610086859

QQ:1740415547

微信号：18610086859

新浪微博：ilovepains

王家林老师的第一个中国梦：免费为全社会培养100万名优秀的大数据从业人员！

目前您尚未登录，请 登录 或 注册 后进行评论

相关文章推荐

spark-submit

[jifeng@feng03 spark-1.4.0-bin-hadoop2.6]$ ./bin/spark-submit --class "SimpleApp" --master spark://f...

wind520 2015年08月20日 23:28 2179

Flume环境部署和配置详解及案例大全

flume是一个分布式、可靠、和高可用的海量日志采集、聚合和传输的系统。支持在日志系统中定制各类数据发送方，用于收集数据;同时，Flume提供对数据进行简单处理，并写到各种数据接受方(比如文本、HDF...

moonpure 2015年07月22日 13:36 460

Spark 入门实战之最好的实例

转载：https://www.ibm.com/developerworks/cn/opensource/os-cn-spark-practice1/ 搭建开发环境 安装 Scala I...

sysmedia 2017年04月06日 17:18 372

第8课：彻底实战详解使用IDE开发Spark程序

本地安装好java和scala并创建工程WordCount

Diamonds_ys 2016年08月25日 16:35 258

第8课：彻底实战详解使用IDE开发Spark程序

第8课：彻底实战详解使用IDE开发Spark程序 1.下载安装windows下的scala-2.10.4. 2.打开eclipse，新建scala project： WordCount 3.修改依赖的...

slq1023 2016年01月24日 00:13 2523

3000门徒内部训练绝密视频（泄密版）第8课：彻底实战详解使用IDE开发Spark程序

彻底实战详解使用IDE开发Spark程序使用IDE开发Spark分析 使用IDE开发Spark实战 使用IDE开发Spark的Local和Cluster开发两种选择：IDEA、Eclipse下载S...

tom_8899_li 2016年08月11日 21:55 178

Scala-IDE Eclipse（Windows)中开发Spark应用程序，在Ubuntu Spark集群上运行

Scala-IDE EclipseWindows)中开发Spark应用程序，在Ubuntu Spark集群上运行在进行实际的Spark应用程序开发时，常常会利用Window环境进行程序开发，开发测试好...

lovehuangjiaju 2015年07月11日 18:43 5167

大数据Spark “蘑菇云”行动前传第3课：在IDE下开发第一个Scala程序透彻解析及Scala控制结构详解实战

大数据Spark “蘑菇云”行动前传第3课：在IDE下开发第一个Scala程序透彻解析及Scala控制结构详解实战 1 通过IDE 开发第一个scala程序彻

底详解 2 scala基础语法和控制结构实...

duan_zhihua 2016年07月20日 20:53 568

Local模式下开发第一个Spark程序并运行于集群环境

第一阶段（1-3月）：会从浅入深，基于大量案例实战，深度剖析和讲解Spark，并且会包含完全从企业真实复杂业务需求中抽取出的案例实战。课程会涵盖Scala编程详解、Spark核心编程、Spark SQ...

kxr0502 2016年01月10日 08:09 1268

Spark on yarn Intellij ide 安装，编译，打包，集群运行 详解

一intellij 安装centos65系统 步骤一 步骤二 步骤三本地运行 打包集群 yarn 运行说明：已经安装好hadoop2.2.0 完全分布，scala，spark已安装好，环境配置完毕；主...

ZHAOLEI5911 2017年03月28日 14:26 388

第8课：Spark Streaming源码解读之RDD生成全生命周期彻底研究和思考

本期内容 1. DStream与RDD关系彻底研究 2. Streaming中RDD的生成彻底研究

chinsun_1 2016-06-01 18:34 272

第7课在自己搭建的spark集群下运行程序实例

第一阶段：Spark streaming、spark sql、kafka、spark内核原理（必须有一个大型项目经验）；第二阶段：spark运行的各种环境，各种故障的解决，性能优化（精通spark...

lhui798 2016-01-10 19:53 288

第8课：Spark Streaming源码解读之RDD生成全生命周期彻底研究和思考

本期内容 Dstream与rdd关系彻底研究 Dstream中rdd生成彻底研究从整个sparkstreaming角度来考虑，rdd分为三个方面的内容： a.怎么生成的，依靠什么生成的； b....

MichaelLi916 2016-05-29 00:59 1487

【Spark亚太研究院系列丛书】Spark实战高手之路-第一章 构建Spark集群-配置Hadoop-伪分布模式并运...

执行文件拷贝操作

wwttz1974 2014-08-27 11:17 657

第8课：零基础实战Scala最常用数据结构Map和Tuple及Spark源码鉴赏

零基础实战Scala最常用数据结构Map和Tuple及Spark源码鉴赏

w517424787 2016-07-30 21:34 892

第10课：底实战详解使用Java开发Spark程序学习笔记（二）

Maven下的Spark配置： http://maven.outofmemory.cn/org.apache.spark，这个网站提供了Spark core、Spark Streaming使用Mave...

slq1023 2016-02-28 10:15 1564

第3课：在IDE下开发第一个Scala程序纯傻瓜式彻底透彻解析

Scala零基础实战详解！

w517424787 2016-07-30 09:26 1673

第10课：底实战详解使用Java开发Spark程序学习笔记

第10课：底实战详解使用Java开发Spark程序学习笔记 本期内容： 1. 为什么要使用Java？ 2. 使用Java开发Spark实战 3. 使用Java开发Spark的Local和Cluster...

slq1023 2016-02-27 00:00 1056

第95讲:使用Scala开发集群运行的Spark来实现在线热搜索词获取

package com.dt.streaming import org.apache.spark.SparkConf import org.apache.spark.streaming.{Secon...

qq_21234493 2016-05-15 16:37 564

spark mllib 应用程序开发及提交到spark集群运行--入门

一、程序开发 Dependencies MLlib uses the linear algebra package Breeze, which depends on netlib-java f...

hualizhuanshen2014 2016-03-24 13:16 544