编译、执行Spark程序

来自xt_wiki
在spark下有两种方式运行spark程序，一种是通过spark-shell方式进行交互式的编程，另一种是编写driver程序，编译，打成jar包，用spark-submit提交运行程序。Spark提供scala、java、python等多种语言api，方便用户选择自己熟悉的语言进行编程。

spark-shell

spark-shell使用很简单，当spark以standalone模式运行后，使用$SPARK_HOME/spark-shell进入shell即可，在Spark-shell中SparkContext已经创建好了，默认为sc，可直接使用。

示例：

```
scala> val textFile = sc.textFile("hdfs://w-namenode.qss.zzbc2.qihoo.net:9000/home/xitong/ouyangwen/test")
 textFile: spark.RDD[String] = spark.MappedRDD@2ee9b6e3
scala> textFile.count() // Number of items in this RDD
 res0: Long = 126
scala> textFile.first() // First item in this RDD
 res1: String = # Apache Spark
scala> val linesWithSpark = textFile.filter(line => line.contains("Spark"))
 linesWithSpark: spark.RDD[String] = spark.FilteredRDD@7dd4af09
scala> textFile.filter(line => line.contains("Spark")).count() // How many lines contain "Spark"?
 res3: Long = 15
```

spark-submit

使用spark-submit提交运行程序时，主要分为三步：1）编写自己的driver程序；2）编译、打成jar包；3）spark-submit提交程序执行。

以WordCount程序为例：

1. 编写driver程序

```
import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._
import org.apache.spark.SparkConf

object WordCount {
    def main(args: Array[String]) {
        val inputFile = " hdfs://w-namenode.qss.zzbc2.qihoo.net:9000/home/xitong/ouyangwen/test "
        val outputPath = "hdfs://w-namenode.qss.zzbc2.qihoo.net:9000/home/xitong/ouyangwen/result"
        val conf = new SparkConf().setAppName("WordCount")
        val sc = new SparkContext(conf)
        val textFile = sc.textFile(inputFile)
        //对RDD做各种transformation操作
        val result = textFile.flatMap(line => line.split(“\\s+”)).map(word => (word,1)).reduceByKey(_+_)
        //对RDD做action操作，只有action才能触发job执行
        result.saveAsTextFile(outputPath)
        sc.stop()
    }
}
```

在driver程序中，需要创建SparkConf、SparkContext。

2. 编译、打包程序

2.1 Scala编译（编译时请选择最新版的spark-assembly *.jar包，路径为/usr/bin/hadoop/software/spark/lib/）

1) 新建build、src文件夹，在build下面新建classes文件夹。把所有的Scala源文件放到src文件夹中。然后运行scalac进行编译

```
scalac -d build/classes -cp /usr/bin/hadoop/software/spark/lib/spark-assembly-1.4.1-hadoop0.20.2.1U15.jar src/*.scala
```

2) 编译完成之后，用jar打包,XXX.jar为指定jar包的名字。在build路径下会出现jar包

```
jar cf build/XXX.jar -C build/classes .
```

2.2 SBT编译

1) 按照下面的路径建立文件夹：

```
.
./build.sbt
```

```
./src
./src/main
./src/main/scala
./src/main/scala/*.scala
```

2) 编辑build.sbt文件

```
name := "XXX"
version := "1.0"
scalaVersion := "2.10.4"
libraryDependencies ++= Seq(
"org.apache.spark" %% "spark-core" % "1.3.1",
"org.apache.spark" %% "spark-mllib" % "1.3.1"
)
```

3) 用SBT编译和打包。

```
在./target/scala-2.10/下会看到生成的jar包。
sbt package
```

3. spark-submit 提交程序执行

YYY表示Scala源代码中的主类名。XXX.jar表示上面得到jar包的路径。

```
spark-submit --class YYY XXX.jar
```

配置参数

在使用spark-shell和spark-submit运行程序时，可以根据实际情况，自行修改配置参数。配置参数可通过spark-shell --help 或 spark-submit --help 查找，常见配置参数如下： 常见参数如下:

参数	含义
--driver-memory	申请driver内存，默认512M
--driver-cores	指定driver的core数量，在YARN的cluster模式，默认为1
--executor-memory	申请每个executor多少内存，默认2G

--executor-cores	指定每个executor的core的数量，在YARN模式，默认为1
--num-executors	指定executors的数量，在YARN模式，默认值为10
--master	yarn-client 、yarn-cluster或者local模式（仅用于测试）
--name	指定Application 名称
--py-files	执行python任务过程中所需的.zip .egg或者.py文件，如有多个，以逗号分隔
--files	在executor执行任务过程中所需要的文件，如有多个，以逗号分隔
--jars	执行任务过程中所需要的jar包，如有多个，以逗号分隔
--conf	指定Spark configuration property，比如 --conf spark.shuffle.memoryFraction=0.3

除此之外，也可对spark属性、环境等配置参数进行调整，常见的spark配置参数如下：

参数	含义
--conf spark.shuffle.memoryFraction	默认为0.2，如果spark.shuffle.spill为“true”，shuffle中聚合和合并组操作使用的java堆内存占总内存的比重。在任何时候，shuffles使用的所有内存内maps的集合大小都受这个限制的约束。超过这个限制，spilling数据将会保存到磁盘上。 如果spilling太过频繁，考虑增大这个值。
--conf spark.storage.memoryFraction	executor内存中有多少可以用于RDD Cache，默认为0.6，即内存的60%，建议这个比值不要超过JVM Old Gen区域的比值，因为RDD Cache数据通常都是长期驻留内存的，理论上也就是说最终会被转移到Old Gen区域（如果该RDD还没有被删除的话），如果这部分数据允许的尺寸太大，势必把Old Gen区域占满，造成频繁的FULL GC
--conf spark.kryoserializer.buffer.max.mb	默认为64，Kryo序列化缓存允许的最大值。这个值必须大于你尝试序列化的对象

```
示例：
  spark-shell --driver-memory 2g  -- executor-memory 50g  --total-executor-cores 50  --conf spark.storage.memoryFraction=0.4
  spark-submit  - class YYY XXX.jar  --driver-memory 5g  -- executor-memory 50g  --total-executor-cores 100  --conf
spark.shuffle.memoryFraction=0.4
```

pyspark pypy解析器使用方法

cluster模式

采用spark-submit提交作业时，添加如下参数：

参数	内容
--conf spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON	设置PYSPARK_DRIVER_PYTHON路径，如./pypy/bin/pypy
--conf spark.yarn.appMasterEnv.PYSPARK_PYTHON	设置PYSPARK_PYTHON路径，如./pypy/bin/pypy
--archives	pypy压缩包所在路径+"#解压后文件夹名称"，如：hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/pypy.tgz#pypy

```
 示例：
$SPARK_HOME/bin/spark-submit \
  --master yarn-cluster \
  --conf spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON=./pypy/bin/pypy \
  --conf spark.yarn.appMasterEnv.PYSPARK_PYTHON=./pypy/bin/pypy \
  --archives hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/pypy.tgz#pypy \
  pypytest.py
```

client模式

1）下载pypy.tgz至客户端本地并解压到$PYPY_HOME，提交作业时设置

```
PYSPARK_DRIVER_PYTHON=$PYPY_HOME/bin/pypy
PYSPARK_PYTHON=./pypy/bin/pypy
```

2）在所提交的作业代码中，添加：

```
import os
os.environ['PYSPARK_PYTHON'] = './pypy/bin/pypy'
```

3）采用spark-submit提交作业时，添加如下参数：

参数	内容	
--archives	pypy压缩包所在路径+"#解压后文件夹名称"，如：hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/pypy.tgz	

```
示例：
   export PYSPARK_DRIVER_PYTHON=$PYPY_HOME/bin/pypy
  export PYSPARK_PYTHON=./pypy/bin/pypy
  $SPARK_HOME/bin/spark-submit \
    --master yarn-client \
    --archives hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/pypy.tgz#pypy \
    pypytest.py
```

注意：

1）建议使用cluster模式提交作业；

2）目前，pypy压缩包路径为hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/pypy.tgz，装有numpy，可直接使用，或下载至本地，利用$PYPY_HOME/bin/pip来安装第三方库，再次打包为.tgz格式文件提交；

3）pypy在python与numpy对象转换场景下性能较差，若存在此场景，请测试评估；

各集群pypy.tgz路径如下： safe.lycc集群：hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/pypy.tgz qss.zzzc集群：hdfs://namenodefd1v.qss.zzzc.qihoo.net:9000/home/spark/yarn/pypy.tgz dfs.shbt集群：hdfs://namenode.dfs.shbt.qihoo.net:9000/home/spark/yarn/pypy.tgz dfs.shgt集群：hdfs://w-namenode.dfs.shgt.qihoo.net:9000/home/spark/yarn/pypy.tgz

pyspark 自定义python2.7环境使用方法

cluster模式

采用spark-submit提交作业时，添加如下参数：

参数	内容
--conf spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON	设置PYSPARK_DRIVER_PYTHON路径，如./python27/bin/python
--conf spark.yarn.appMasterEnv.PYSPARK_PYTHON	设置PYSPARK_PYTHON路径，如./python27/bin/python
--archives	python27.tgz压缩包所在路径+"#解压后文件夹名称"，如：hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/python27.tgz#python27

```
示例：
$SPARK_HOME/bin/spark-submit \
  --master yarn-cluster \
  --conf spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON=./python27/bin/python \
  --conf spark.yarn.appMasterEnv.PYSPARK_PYTHON=./python27/bin/python \
  --archives hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/python27.tgz#python27 \
  test.py
```

client模式

0）准备工作：需要安装python2.7所需要的依赖库，如openssl-devel等。

1）下载python27.tgz至客户端本地并解压到./python27，提交作业时设置

```
PYSPARK_DRIVER_PYTHON=./python27/bin/python
PYSPARK_PYTHON=./python27/bin/python
```

2）在所提交的作业代码中，添加：

```
import os
os.environ['PYSPARK_PYTHON'] = './python27/bin/python'
```

3）采用spark-submit提交作业时，添加如下参数：

参数	内容
--archives	python27.tgz压缩包所在路径+"#解压后文件夹名称"，如： hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/python27.tgz#python27

示例：

```
export PYSPARK_DRIVER_PYTHON=./python27/bin/python
export PYSPARK_PYTHON=./python27/bin/python
$SPARK_HOME/bin/spark-submit \
  --master yarn-client \
  --archives hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/python27.tgz#python27 \
  test.py
```

注意：

1）因client模式提交作业时，python2.7依赖于本地环境，故建议使用cluster模式提交作业；

2）目前，集群提供更有基础python2.7压缩包路径为hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/python27.tgz，装有numpy、six、web、scipy、sklearn，可直接使用，或下载至本地，利用./bin/pip来安装第三方库，再次打包为.tgz格式文件提交，也可自行安装编译python2.7包进行自定义环境配置；

所提供基础版本python27.tgz在各集群路径如下： safe.lycc集群：hdfs://namenode.safe.lycc.qihoo.net:9000/home/spark/yarn/python27.tgz qss.zzzc集群：hdfs://namenodefd1v.qss.zzzc.qihoo.net:9000/home/spark/yarn/python27.tgz dfs.shbt集群：hdfs://namenode.dfs.shbt.qihoo.net:9000/home/spark/yarn/python27.tgz

python2.7自定义环境包设置

1、python2.7安装
1）安装python2.7环境依赖；
2）下载python2.7安装包；
3）解压，并进入安装包目录，执行./configure --prefix=/home/xxx/python2.7。
4）执行 make & make install
5）export PATH=/home/xxx/python2.7/bin:$PATH
2、setuptools安装
1）下载setuptools安装包；
2）执行安装；
3、pip安装
1）下载pip安装包；
2）解压进入安装包目录；
3）以自定义安装python命令，执行 python setup.py install
4、上述步骤无误后，则可用自定义的python及pip命令安装所需模块；
5、spark作业使用自定义安装环境打包：在/home/xxx/python2.7/目录下，进行打包压缩，上传至自定义hdfs路径，按之前介绍进行作业提交。

scala编译器包下载地址

wget http://360stat:stat360@c05.dfs.shgt.qihoo.net:8360/publish/spark/release/scala-2.10.5.tgz