# Spark-on-YARN

## 官方文档

<http://spark.apache.org/docs/latest/running-on-yarn.html>

## 配置安装

1. **安装hadoop：**需要安装HDFS模块和YARN模块，HDFS必须安装，spark运行时要把jar包存放到HDFS上。
2. **安装Spark：**解压Spark安装程序到一台服务器上，修改spark-env.sh配置文件，spark程序将作为YARN的客户端用于提交任务

export JAVA\_HOME=/usr/local/jdk1.8.0\_131

export HADOOP\_CONF\_DIR=/usr/local/hadoop-2.7.3/etc/Hadoop

//spark-env.sh配置文件有：export SPARK\_MASTER\_HOST=master

Tips：或者把HADOOP\_CONF\_DIR在/etc/profile里配置

1. **启动HDFS和YARN**

## 运行模式（cluster模式和client模式）

1. **cluster模式**

./bin/spark-submit --class org.apache.spark.examples.SparkPi \

--master yarn \

--deploy-mode cluster \

--driver-memory 1g \

--executor-memory 1g \

--executor-cores 2 \

--queue default \ //默认资源调度器

lib/spark-examples\*.jar \

10 //参数

---------------------------------------------------------------------------------------------------------------------------------

./bin/spark-submit --class cn.edu360.spark.day1.WordCount \

--master yarn \

--deploy-mode cluster \

--driver-memory 1g \

--executor-memory 1g \

--executor-cores 2 \

--queue default \

/home/bigdata/hello-spark-1.0.jar \

hdfs://node-1.edu360.cn:9000/wc hdfs://node-1.edu360.cn:9000/out-yarn-1

1. **client模式**

./bin/spark-submit --class org.apache.spark.examples.SparkPi \

--master yarn \

--deploy-mode client \

--driver-memory 1g \

--executor-memory 1g \

--executor-cores 2 \

--queue default \

lib/spark-examples\*.jar \

10

spark-shell必须使用client模式

./bin/spark-shell --master yarn --deploy-mode client

Tips：原启动spark-shell的方式

spark-shell \

–master spark://master:7077,slave1:7077,slave2:7077 \

--executor-memory 5g \

--total-executor-cores 12

如果不加—master启用的是一个本地进程（Local模式）

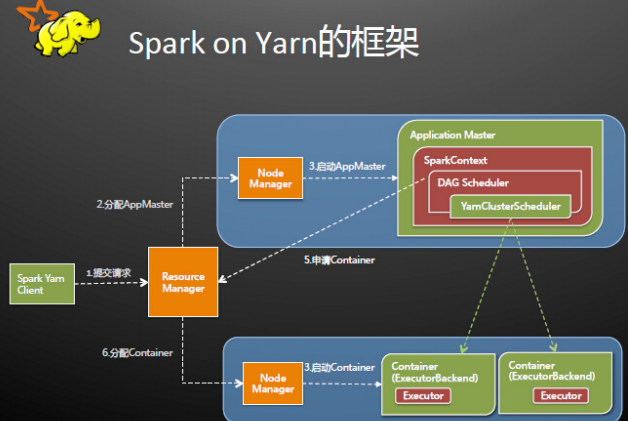
1. **两种模式的区别**

cluster模式：Driver程序在YARN中运行，应用的运行结果不能在客户端显示，所以最好运行那些将结果最终保存在外部存储介质（如HDFS、Redis、Mysql）而非stdout输出的应用程序，客户端的终端显示的仅是作为YARN的job的简单运行状况。

client模式：Driver运行在Client上，应用程序运行结果会在客户端显示，所有适合运行结果有输出的应用程序（如spark-shell）

1. **原理**

cluster模式：



Spark Driver首先作为一个ApplicationMaster在YARN集群中启动，客户端提交给ResourceManager的每一个job都会在集群的NodeManager节点上分配一个唯一的ApplicationMaster，由该ApplicationMaster管理全生命周期的应用。具体过程：

1. 由client向ResourceManager提交请求，并上传jar到HDFS上

这期间包括四个步骤：

a).连接到RM

b).从RM的ASM（ApplicationsManager ）中获得metric、queue和resource等信息。

c). upload app jar and spark-assembly jar

d).设置运行环境和container上下文（launch-container.sh等脚本)

2. ResouceManager向NodeManager申请资源，创建Spark ApplicationMaster（每个SparkContext都有一个ApplicationMaster）

3. NodeManager启动ApplicationMaster，并向ResourceManager AsM注册

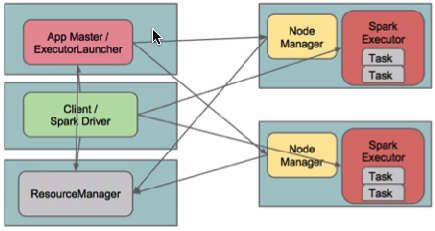
4. ApplicationMaster从HDFS中找到jar文件，启动SparkContext、DAGscheduler和YARN Cluster Scheduler

5. ResourceManager向ResourceManager AsM注册申请container资源

6. ResourceManager通知NodeManager分配Container，这时可以收到来自ASM关于container的报告。（每个container对应一个executor）

7. Spark ApplicationMaster直接和container（executor）进行交互，完成这个分布式任务。

client模式：



在client模式下，Driver运行在Client上，通过ApplicationMaster向RM获取资源。本地Driver负责与所有的executor container进行交互，并将最后的结果汇总。结束掉终端，相当于kill掉这个spark应用。一般来说，如果运行的结果仅仅返回到terminal上时需要配置这个。

客户端的Driver将应用提交给Yarn后，Yarn会先后启动ApplicationMaster和executor，另外ApplicationMaster和executor都 是装载在container里运行，container默认的内存是1G，ApplicationMaster分配的内存是driver- memory，executor分配的内存是executor-memory。同时，因为Driver在客户端，所以程序的运行结果可以在客户端显 示，Driver以进程名为SparkSubmit的形式存在。

如果使用spark on yarn 提交任务，一般情况，都使用cluster模式，该模式，Driver运行在集群中，其实就是运行在ApplicattionMaster这个进程成，如果该进程出现问题，yarn会重启ApplicattionMaster（Driver），SparkSubmit的功能就是为了提交任务。

如果使用交换式的命令行，必须用Client模式，该模式，Driver是运行在SparkSubmit进程中，因为收集的结果，必须返回到命令行（即启动命令的那台机器上），该模式，一般测试，或者运行spark-shell、spark-sql这个交互式命令行是使用

注意：如果你配置spark-on-yarn的client模式，其实会报错（内存检测机制）。

修改$HADOOP\_HOME /etc/hadoop所有yarn节点的yarn-site.xml，在该文件中添加如下配置

<property>

<name>yarn.nodemanager.pmem-check-enabled</name>

<value>false</value>

</property>

<property>

<name>yarn.nodemanager.vmem-check-enabled</name>

<value>false</value>

</property>