# Spark安装部署[¶](#Spark安装部署)

## 实验介绍[¶](#实验介绍)

#### 1. 实验内容[¶](#1.-实验内容)

* Spark的安装与部署

## 实验要求[¶](#实验要求)

#### 1. Java （需要安装1.6.x及其以上版本)[¶](#1.-Java-（需要安装1.6.x及其以上版本))

在终端输入 java -version 来查看Java 版本，这里我们使用的是1.7版本；

#### 

#### 2. Hadoop和spark[¶](#2.-Hadoop和spark)

Hadoop的具体安装可以参见我们以前的课程：Hadoop部署及管理。

在这里，我们使用了Hadoop V2.6.0版本以及sqoop-2.0.1版本。

## 实验步骤[¶](#实验步骤)

#### 1.实验前准备[¶](#1.实验前准备)

$ su - hadoop

口令输入：hadoop

$ bash

启动ssh,口令输入：hadoop

hadoop@357987c120a9:~$ sudo service ssh start

[sudo] password for hadoop:

[ ok ] Starting OpenBSD Secure Shell server: sshd.

hadoop@357987c120a9:~$

#### 2.启动hadoop（namenode节点）[¶](#2.启动hadoop（namenode节点）)

启动命令为：

$ start-all.sh

检查是否运行成功

#执行jps命令可以查看到hadoop的几个主要进程:

$ jps

ResourceManager

NodeManager

Jps

NameNode

SecondaryNameNode

DataNode

## 3.安装与配置 Spark[¶](#3.安装与配置-Spark)

#### （1）解压并安装Spark[¶](#（1）解压并安装Spark)

本次实验我们将spark安装在/hadoop/app下，因此我们建立spark的安装目录

$ mkdir /hadoop/app

下载安装包有如下两个方法：

方法一：这里我们将它的压缩包已经下载放到了共享目录/home/ds/data/tools目录下，我们要将其拷贝到/hadoop/app目录下：

$ cp /home/ds/data/tools/spark-2.0.1-bin-hadoop2.7.tgz /hadoop/app

方法二：直接在网上下载

$ wget http://archive.apache.org/dist/spark/spark-2.0.1/spark-2.0.1-bin-hadoop2.7.tgz

解压

$ cd /hadoop/app

$ tar -zxvf spark-2.0.1-bin-hadoop2.7.tgz

删除安装文件

$ rm -r spark-2.0.1-bin-hadoop2.7.tgz

修改文件名称

$ mv spark-2.0.1-bin-hadoop2.7 spark

#### （2）配置 Hadoop 环境变量[¶](#（2）配置-Hadoop-环境变量)

在 Yarn 上运行 Spark 需要配置 HADOOP\_CONF\_DIR、 YARN\_CONF\_DIR 和 HDFS\_CONF\_DIR 环境变量

命令：

$ vi /hadoop/.bash\_profile

在下面添加如下代码：

export SPARK\_HOME=/hadoop/app/spark

export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export HDFS\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

export YARN\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop

保存关闭后，执行以下命令使得环境变量生效：

$ source /hadoop/.bash\_profile

#### （2）修改配置文件[¶](#（2）修改配置文件)

$ cd /hadoop/app/spark/conf/

$ cp spark-env.sh.template spark-env.sh

$ vi spark-env.sh

在第一行“#!/usr/bin/env bash”下，写入以下内容

export SPARK\_MASTER\_HOST=127.0.0.1

export SPARK\_MASTER\_PORT=7077

export SPARK\_WORKER\_CORES=1

export SPARK\_WORKER\_MEMORY=512M

#### 4. Spark的 启动[¶](#4.-Spark的-启动)

#### （1）进入spark-shell[¶](#（1）进入spark-shell)

进入 Spark 安装主目录,

$ cd /hadoop/app/spark

进入spark的shell界面

$ ./bin/spark-shell



* 使用“CTRL+C”键退出spark-shell界面。

#### （2）启动spark[¶](#（2）启动spark)

①首先启动master

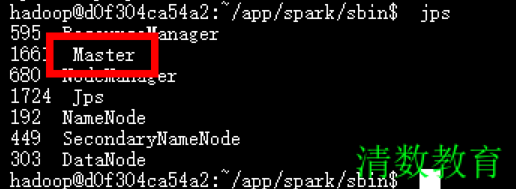
$ cd /hadoop/app/spark/sbin/

$ ./start-master.sh



查看Master进程是否启动

$ jps



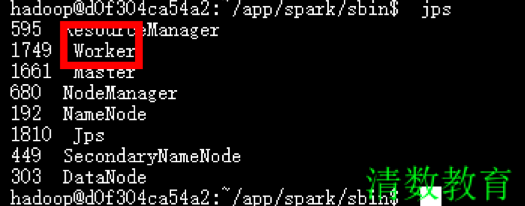
②启动slave

$ ./start-slave.sh spark://127.0.0.1:7077



查看Woker进程是否启动

$ jps



#### 5. 验证Spark[¶](#5.-验证Spark)

#### 运行pi（π）的实例[¶](#运行pi（π）的实例)

$ cd /hadoop/app/spark/

* ①执行pi实例

$ ./bin/spark-submit --class org.apache.spark.examples.SparkPi --master spark://127.0.0.1:7077 --driver-memory 512M --executor-memory 512M --executor-cores 1 ./examples/jars/spark-examples\*.jar

说明：spark-submit 可以提交任务到 spark 集群执行，也可以提交到 hadoop 的 yarn 集群执行。

参数的含义：

--class 应用程序的主类，仅针对 java 或 scala 应用。这里我们使用的是spark自带的计算pi的类。

--master master 的地址，提交任务到哪里执行。

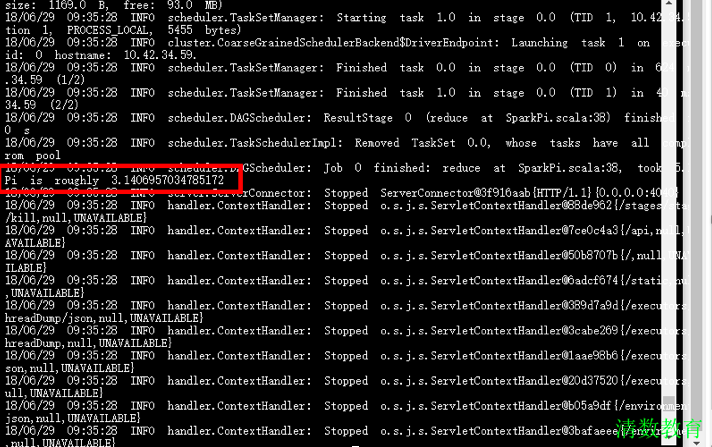
--driver-memory Driver内存，默认 1G。

--executor-memory 每个 executor 的内存，默认是1G。

--driver-cores Driver 的核数，默认是1。在 yarn 或者 standalone 下使用。

./examples/jars/spark-examples\*.jar 指的是/hadoop/app/spark/examples/jars下的spark-examples\*.jar包，运行pi的类就写在这些jar包里。

结果在执行过程中的其中一行，需要大家仔细查看，如下图：



* ②执行时会输出非常多的运行信息，输出结果不容易找到，可以通过 grep 命令进行过滤（命令中的 2>&1 可以将所有的信息都输出到 stdout 中，否则由于输出日志的性质，还是会输出到屏幕中）：

$ ./bin/spark-submit --class org.apache.spark.examples.SparkPi --master spark://127.0.0.1:7077 --driver-memory 512M --executor-memory 512M --executor-cores 1 ./examples/jars/spark-examples\*.jar 2>&1 | grep "Pi is roughly"

结果如下图（结果可能会有微小差别）：

