# Spark on Yarn集群搭建详细过程

由于最近学习大数据开发，spark作为分布式内存计算框架，当前十分火热，因此作为首选学习技术之一。Spark官方提供了三种集群部署方案： Standalone, Mesos, Yarn。其中 Standalone 为spark本身提供的集群模式，搭建过程可以参考官网，本文介绍Spark on Yarn集群部署过程。使用3台普通机器搭建Spark集群，

**软件环境：**

Ubuntu 16.04 LTS

Ubuntu 16.04 LTS

CentOS7

Scala-2.10.6

Hadoop-2.7.2

spark-1.6.1-bin-hadoop2.6

Java-1.8.0\_77

**硬件环境：**

一个Master节点

Intel® Core™ i5-2310 CPU @ 2.90GHz × 4

4G内存

300G硬盘

两个Slave节点

Intel® Core™ i3-2100 CPU @ 3.10GHz × 4

4G内存

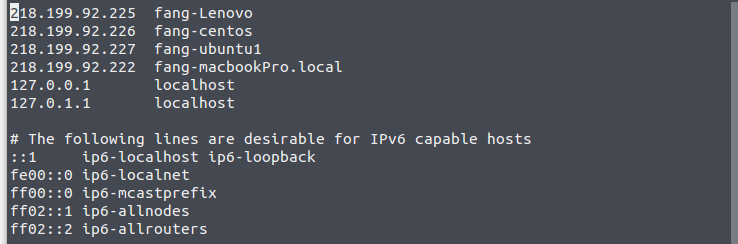
500G硬盘

## 配置/etc/hosts及免密码登录

本文下载安装的软件都放在 home 目录下。

1. 主机hosts文件配置

在每台主机上修改host文件



sudo vim /etc/hosts

218.199.92.227 fang-ubuntu1(Master)

218.199.92.226 fang-centos(Slave)

218.199.92.225 fang-Lenovo(Slave)

127.0.0.1 localhost

127.0.1.1 localhost

注：若此地未配置，或者未配置正确会导致集群启动不正常或者失败

配置之后ping一下各机器名称检查是否生效，例如ssh fang@fang-centos。

1. 配置SSH 免密码登录

如果没有安装ssh，需要安装Openssh server，命令为sudo apt-get install openssh-server。

1. 在所有机器上都生成私钥和公钥

ssh-keygen -t rsa  #一路回车

1. 需要让机器间都能相互访问，就把每个机子上的id\_rsa.pub发给master节点，传输公钥可以用scp来传输。

scp ~/.ssh/id\_rsa.pub fang@fang-ubuntu1:~/.ssh/id\_rsa.pub.slave1

1. 在master上，将所有公钥加到用于认证的公钥文件authorized\_keys中

cat ~/.ssh/id\_rsa.pub\* >> ~/.ssh/authorized\_keys

1. 将公钥文件authorized\_keys分发给每台slave

scp ~/.ssh/authorized\_keys fang@fang-centos:~/.ssh/

1. 在每台机子上验证SSH无密码登录



在终端中输入登录命令，例如：ssh fang@fang-centos 如果直接登录成功而不需要登录密码，则表示设置正确；如果登录不成功，即仍然需要登录密码，则可能需要修改文件authorized\_keys的权限。

注：.ssh 文件夹的权限必须为700，authorized\_keys文件权限必须为600

使用如下命令改变文件夹权限：chmod 600 ~/.ssh/authorized\_keys

## 安装 Java

从官网下载最新版 Java，Spark官方说明 Java 只要是6以上的版本都可以，本文使用的是 jdk-8u91-linux-x64.tar.gz。

在下载目录下直接解压tar -zcvf jdk-8u91-linux-x64.tar.gz并复制文件到/usr/lib/jvm中，命令如下：

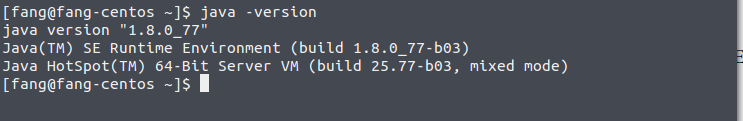
sudo cp -r jdk1.8.0\_77 /usr/lib/jvm(如果没有jvm文件夹，则手动创建一个)，修改环境变量sudo vim /etc/profile，添加下列内容：

export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_77

export JRE\_HOME=$JAVA\_HOME/jre  
 export PATH=$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:$PATH  
 export CLASSPATH=$CLASSPATH:.:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

然后使环境变量生效，并验证 Java 是否安装成功

$ source /etc/profile  #生效环境变量  
 $ java -version   #如果打印出如下版本信息，则说明安装成功



## 安装 Scala

Spark官方要求 Scala 版本为 2.10.x，注意不要下错版本，我这里下了 2.10.6同样我们在~/中解压

tar -zcvf scala-2.10.6.tar.gz

再次修改环境变量sudo vim /etc/profile，添加以下内容：

export SCALA\_HOME=/home/fang/scala-2.10.6  
 export PATH=$PATH:$SCALA\_HOME/bin

同样的方法使环境变量生效，并验证 scala 是否安装成功

$ source /etc/profile  #生效环境变量  
 $ scala -version        #如果打印出如下版本信息，则说明安装成功。



## 安装配置 Hadoop YARN

从官网下载 hadoop2.7.2版本,在用户根目录解压tar -zcvf hadoop-2.7.2.tar.gz

再次修改环境变量sudo vim /etc/profile，添加以下内容：

export HADOOP\_HOME=/home/fang/hadoop-2.7.2

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

export YARN\_HOME=/home/fang/hadoop-2.7.2

export YARN\_CONF\_DIR=${YARN\_HOME}/etc/hadoop

同样的方法使环境变量生效

$ source /etc/profile  #生效环境变量

注：有时候修改了/etc/profile文件，执行命令source之后还是不能达到正常的效果，则需要重新机器，看问题是否能解决。

配置 Hadoop，cd ~/hadoop-2.7.2/etc/hadoop进入hadoop配置目录，需要配置有以下7个文件：hadoop-env.sh，yarn-env.sh，slaves，core-site.xml，hdfs-site.xml，maprd-site.xml，yarn-site.xml。

在hadoop-env.sh中配置JAVA\_HOME

# The java implementation to use.  
export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_77

在yarn-env.sh中配置JAVA\_HOME

# some Java parameters  
export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_77

在slaves中配置slave节点的ip或者host，

fang-centos

fang-Lenovo

修改core-site.xml

<configuration>  
    <property>  
        <name>fs.defaultFS</name>  
        <value>hdfs://fang-ubuntu1:9000/</value>  
    </property>  
    <property>  
        <name>hadoop.tmp.dir</name>  
        <value>file:/home/fang//hadoop-2.7.2/tmp</value>  
    </property>  
</configuration>

修改hdfs-site.xml

<configuration>  
    <property>  
        <name>dfs.namenode.secondary.http-address</name>  
        <value>fang-ubuntu1:9001</value>  
    </property>  
    <property>  
        <name>dfs.namenode.name.dir</name>  
        <value>file:/home/fang/hadoop-2.7.2/dfs/name</value>  
    </property>  
    <property>  
        <name>dfs.datanode.data.dir</name>  
        <value>file:/home/fang/hadoop-2.7.2/dfs/data</value>  
    </property>  
    <property>  
        <name>dfs.replication</name>  
        <value>3</value>  
    </property>  
</configuration>

修改mapred-site.xml

<configuration>  
    <property>  
        <name>mapreduce.framework.name</name>  
        <value>yarn</value>  
    </property>  
</configuration>

修改yarn-site.xml

<configuration>  
    <property>  
        <name>yarn.nodemanager.aux-services</name>  
        <value>mapreduce\_shuffle</value>  
    </property>  
    <property>  
        <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>  
        <value>org.apache.hadoop.mapred.ShuffleHandler</value>  
    </property>  
    <property>  
        <name>yarn.resourcemanager.address</name>  
        <value>fang-ubuntu1:8032</value>  
    </property>  
    <property>  
        <name>yarn.resourcemanager.scheduler.address</name>  
        <value>fang-ubuntu1:8030</value>  
    </property>  
    <property>  
        <name>yarn.resourcemanager.resource-tracker.address</name>  
        <value>fang-ubuntu1:8035</value>  
    </property>  
    <property>  
        <name>yarn.resourcemanager.admin.address</name>  
        <value>fang-ubuntu1:8033</value>  
    </property>  
    <property>  
        <name>yarn.resourcemanager.webapp.address</name>  
        <value>fang-ubuntu1:8088</value>  
    </property>  
</configuration>

将配置好的hadoop-2.7.2文件夹分发给所有slaves节点

scp -r ~/hadoop-2.6.0 fang@fang-centos:~/

启动 Hadoop  
在 master节点上执行以下操作，就可以启动 hadoop 了。

cd ~/hadoop-2.7.2    #进入hadoop目录  
bin/hadoop namenode -format    #格式化namenode

注：若格式化之后重新修改了配置文件，重新格式化之前需要删除tmp，dfs，logs文件夹。  
sbin/start-dfs.sh              #启动dfs   
sbin/start-yarn.sh              #启动yarn

验证 Hadoop 是否安装成功，可以通过jps命令查看各个节点启动的进程是否正常。

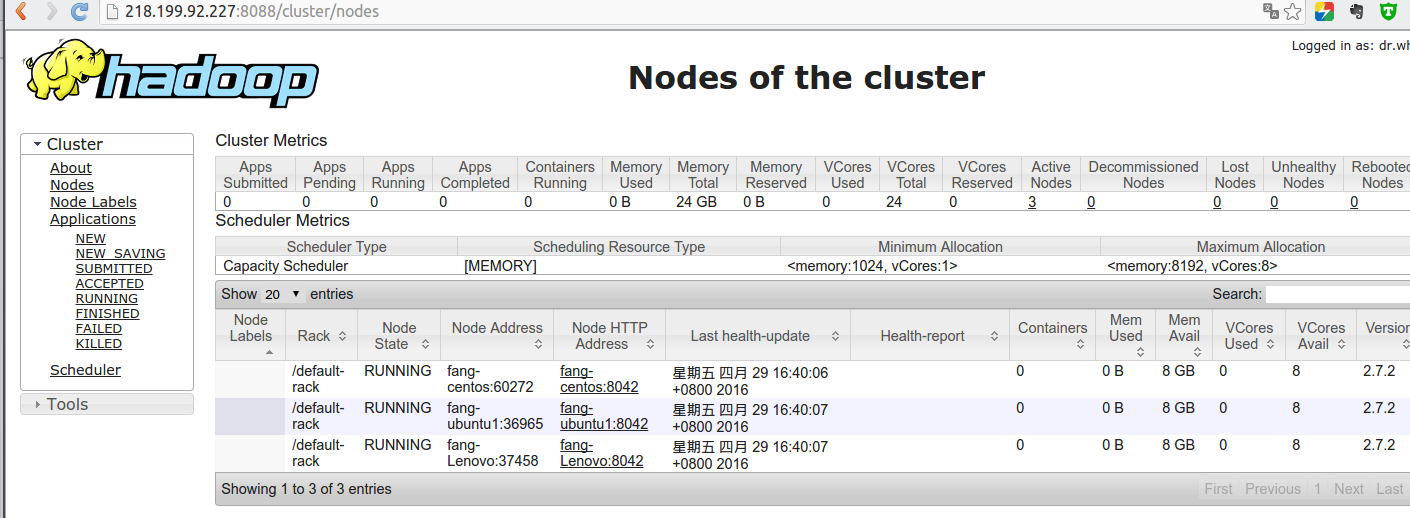
在 master 上应该有以下几个进程：



在每个slave上应该有以下几个进程：



在浏览器中输入 http://fang-ubuntu1:8088 ，可以看到hadoop 的管理界面。



## Spark安装

下载解压，进入[官方下载地址](http://spark.apache.org/downloads.html" \t "/home/fang/文档\\x/_blank)下载最新版 Spark。我下载的是 spark-1.6.1-bin-hadoop2.6.tar.gz。

在~/目录下解压，tar -zcvf spark-1.6.1-bin-hadoop2.6.tar.gz

配置 Spark

cd ~spark-1.6.1-bin-hadoop2.6/conf    #进入spark配置目录

cp spark-env.sh.template spark-env.sh  #从配置模板复制

vim spark-env.sh    #添加配置内容

在spark-env.sh末尾添加以下内容（这是我的配置，你可以自行修改）：

export SPARK\_HOME=/home/fang/spark-1.6.1-bin-hadoop2.6

export SCALA\_HOME=/home/fang/scala-2.10.6

export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_77

export HADOOP\_HOME=/home/fang/hadoop-2.7.2

export PATH=$PATH:$JAVA\_HOME/bin:$HADOOP\_HOME/bin

:$HADOOP\_HOME/sbin:$SCALA\_HOME/bin

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

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

export SPARK\_MASTER\_IP=218.199.92.227

SPARK\_LOCAL\_DIRS=/home/fang/spark-1.6.1-bin-hadoop2.6

SPARK\_DRIVER\_MEMORY=1G

export SPARK\_LIBARY\_PATH=.:$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

:$HADOOP\_HOME/lib/native

注：在设置Worker进程的CPU个数和内存大小，要注意机器的实际硬件条件，如果配置的超过当前Worker节点的硬件条件，Worker进程会启动失败。

vim slaves在slaves文件下填上slave主机名：

slave1  
 slave2

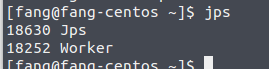
将配置好的spark-1.6.1-bin-hadoop2.6文件夹分发给所有slaves吧

scp -r ~/spark-1.6.1-bin-hadoop2.6 fang@fang-cenos:~/

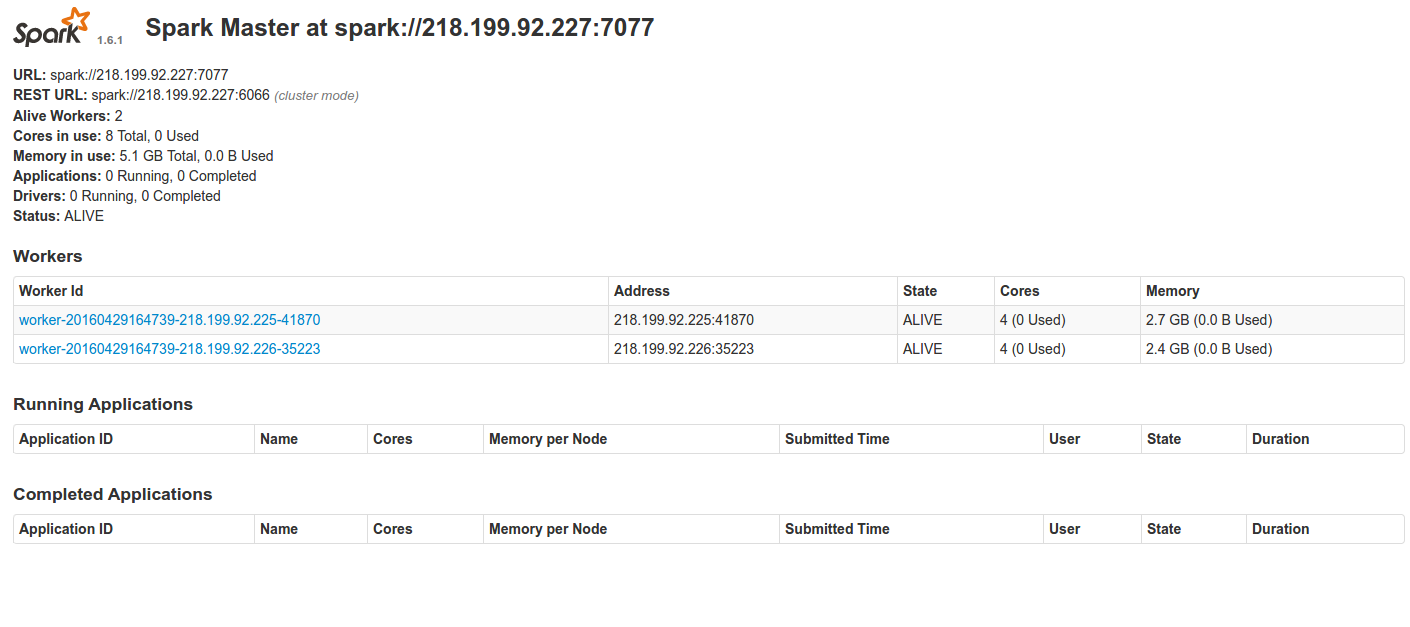
启动Spark ,sbin/start-all.sh

验证 Spark 是否安装成功  
主节点上启动了Master进程：

在 slave 上启动了Worker进程：



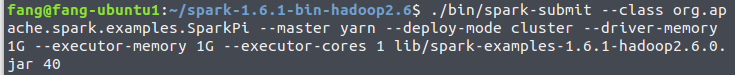
进入Spark的Web管理页面：http://fang-ubuntu1:8080



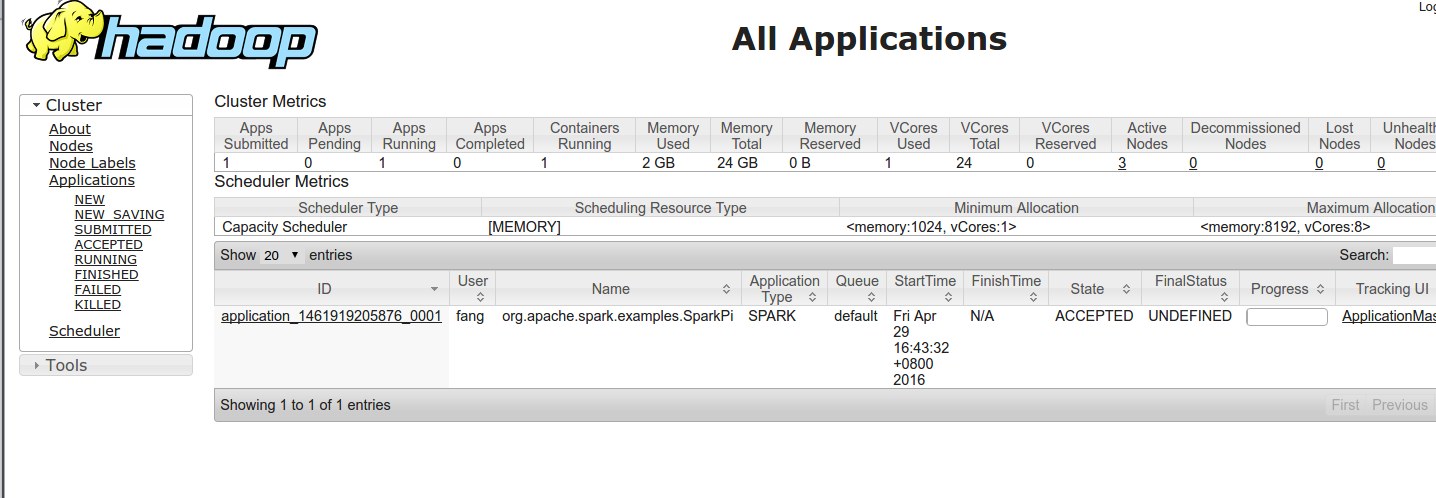
## 运行示例

本例以集群模式运行SparkPi实例程序(deploy-mode 设置为cluster)

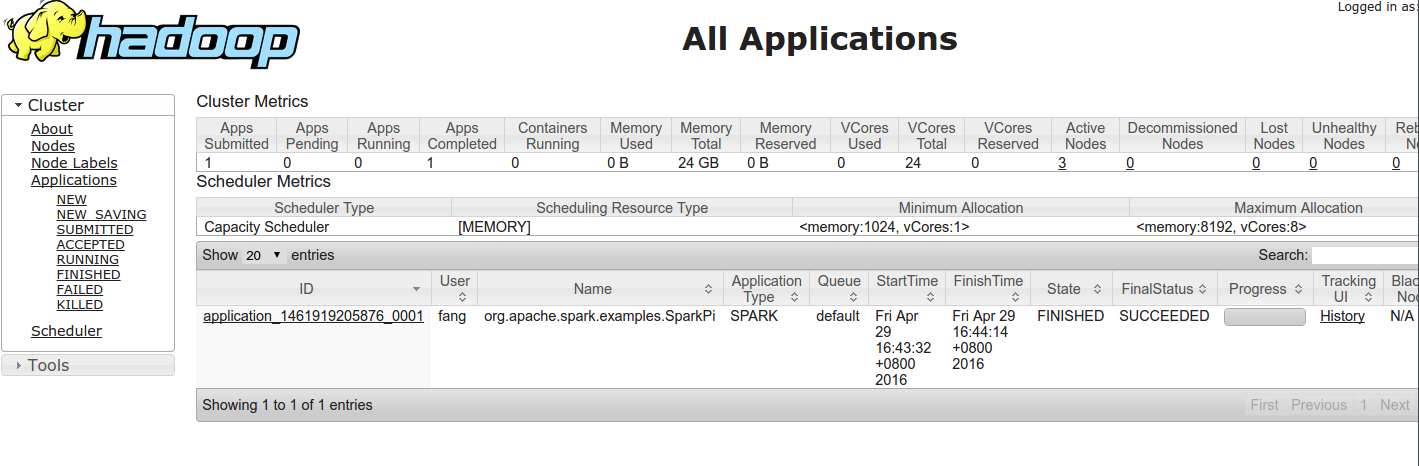
./bin/spark-submit --class org.apache.spark.examples.SparkPi --master yarn --deploy-mode cluster --driver-memory 1G --executor-memory 1G --executor-cores 1 lib/spark-examples-1.6.1-hadoop2.6.0.jar 40



任务提交时web界面



作业运行完成web界面



注意 Spark on YARN 支持两种运行模式，分别为yarn-cluster和yarn-client，yarn-cluster适用于生产环境；而yarn-client适用于交互和调试，因为能在客户端终端看到程序输出。客户端模式实例和上面集群模式运行过程类似，在此不在赘述。