20160602--Spark平台搭建

备注：建立文件夹(sparkdir、hadoop、java、scala、spark) ***每台机器均有***

/usr/**sparkdir**

/hadoop

/hadoop-2.7.2

/java

/jdk1.8.0_91

/scala

/scala-2.11.8

/spark

/spark-1.6.1-bin-without-hadoop.tgz

用户组、用户管理：http://www.cnblogs.com/vincedotnet/p/4017574.html

功能：管理组

用法：gpasswd[-a user][-d user][-A user,...][-M user,...][-r][-R]groupname

参数：

-a：添加用户到组

-d：从组删除用户

-A：指定管理员

-M：指定组成员和-A的用途差不多

-r：删除密码

-R：限制用户登入组，只有组中的成员才可以用newgrp加入该组

备注：为了便于权限分配和管理，在

Ubuntu16Master增加用户组**spark**并将**baoling**加入该用户组

Ubuntu16Slave1增加用户组**spark**并将**baoling**加入该用户组

Ubuntu16Slave2增加用户组**spark**并将**baoling**加入该用户组

sudo groupadd spark / sudo usermod -a -G spark baoling

相关命令：

groupdel

gpasswd

资源下载汇总：

hadoop-2.7.2 http://mirrors.hust.edu.cn/apache/hadoop/

jdk-8u91-linux-x64.gz http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

scala-2.11 http://www.scala-lang.org/download/2.11.8.html

spark-1.6.1-bin-hadoop2.6 http://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-1.6.1/spark-1.6.1-bin-hadoop2.6.tgz

整体资源下载：

链接：http://pan.baidu.com/s/1miMtCTi

密码：x3w0

其他隐含明说：

VMavare11

Ubuntu16.04

创建三台虚拟机：Ubuntu16Master、buntu16Slave1、buntu16Slave2

资源解压对应目录：

hadoop-2.7.2.tar.gz -->/usr/sparkdir/hadoop/

jdk-8u91-linux-x64.gz -->/usr/sparkdir/java/

scala-2.11.8.tgz -->/usr/sparkdir/scala/

spark-1.6.1-bin-without-hadoop.tgz -->/usr/sparkdir/spark/

一、安装JDK*(所有机器均需要配置)*

准备工作

JDK资源下载：http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

JDK版本：jdk-8u91-linux-x64.gz

解压jdk-8u91-linux-x64.gz：tar -xzf jdk-8u91-linux-x64.gz /usr/sparkdir/java/

环境变量配置

全局配置文件：/etc/profile ***(此处使用全局)***

局部配置文件：~/.bashrc

配置内容：

JAVA_HOME=**/usr/sparkdir/java/jdk1.8.0_91 #*(加黑部分是存放jdk的绝对路径)***

PATH=$JAVA_HOME/bin:$PATH

CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/jre/lib/dt.jar:$JAVA_HOME/jre/lib/tools.jar

export JAVA_HOME PATH CLASSPATH

备注：在配置文件末尾加入以上内容，最后需要使配置文件有效***(source /etc/profile OR source ~/.bashrc)***

配置成功与否测试：Java -version

```
baoling@ubuntu:/usr/sparkdir$ java -version
java version "1.8.0_91"
Java(TM) SE Runtime Environment (build 1.8.0_91-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)
```

二、安装Scala*(所有机器均需要配置)*

准备工作：

备注：Scala的版本由Spark的版本决定，这里选择spark-1.6.1及其对应Scala版本Scala 2.11

spark-1.6.1下载：http://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-1.6.1/spark-1.6.1-bin-hadoop2.6.tgz

scala-2.11下载：http://www.scala-lang.org/download/2.11.8.html

解压Scala-2.11.8：tar -xzf scala-2.11.8.tgz /usr/sparkdir/scala/

配置内容：

/etc/profile

export SCALA__HOME=**/usr/sparkdir/scala/scala-2.11.8**

export PATH=${SCALA_HOME}/bin:$PATH

三、配置SSH免密码登录

介绍：

在集群管理和配置中有很多工具可以使用。例如，可以采用pssh等Linux工具在集群中分发与复制文件，用户也可以自己书写Shell、Python的脚步分发包。Spark的Master节点向Worker节点命令需要通过ssh进行发送，用户不希望Master每发送一次命令就输入一次密码，因此需要实现Master无密码登陆到所有Worker。Master作为客户端，要实现无密码公钥认证，连接服务端Worker。需要在Master上生成一个秘钥对，包括一个公钥和一个私钥，然后将公钥复制到Worker上。

其它只是补充：

配置成功的关键在于确保各机器的主机名和IP地址之间能正确解析。修改每台机器的/etc/hosts，如果该台机器做NameNode用，则需要在文件中添加集群中所有机器的IP地址及其对应的主机名；如果该台机器仅作DataNode用，则只需要在文件中添加本机和NameNode的IP地址及其对应的主机名。

机器名	IP	作用
Master	192.168.1.114	NameNode and JobTracker
Slave1	192.168.1.118	DataNode and TaskTracker
Slave2	192.168.1.130	DataNode and TaskTracker

备注：这里的Master、Slave1、Slave2等等，指的是机器的机器名（使用命令hostname可以查看本机的机器名），切记，如果不是机器名的话会出问题的，并且集群中所有结点的机器名都应该不一样

配置所有机器/etc/hosts

sudo gedit /etc/hosts

添加如下内容：

192.168.1.114 Master

192.168.1.118 Slave1

192.168.1.130 Slave2

配置主机名/etc/hostname

sudo gedit /etc/hostname

添加如下内容：

Master <-- Ubuntu16Master

Slave1 <-- Ubuntu16Slave1

Slave2 <-- Ubuntu16Slave2

Slave1 和 Slave2节点上：

建立文件夹：~/

.ssh

Master节点上：

1)测试是否能够无密码登录本机

ssh localhost

备注：如果没有安装openssh-server将出现如下提示 *“ssh : connect to host localhost port 22:Connection refused”*，所以安装openssh-server即可

sudo apt-get install openssh-server

2)Master生成秘钥对，Master的公钥id_rsa.pub需要传送给Slave1、Slave2，从而实现Master无密码登录Slave

ssh-keygen -t rsa

生成秘钥过程中会出现提示信息，按Enter即可

```
baoling@Master:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/baoling/.ssh/id_rsa):
Created directory '/home/baoling/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/baoling/.ssh/id_rsa.
Your public key has been saved in /home/baoling/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:06gA6puyIB4XFESe3YMHtdiu9hIWUfeGGn5Q6C65lnA baoling@Master
The key's randomart image is:
+---[RSA 2048]----+
|  o+ .oo.o       |
|  . +.*.+ o      |
|  .+ +oO . o     |
| ... .+.=o.      |
|.  .. ++S..      |
|.  ..E.o..       |
|oo .+o*          |
|= = .=.          |
|o=  . ..         |
+----[SHA256]-----+
baoling@Master:~$ ls .ssh
id_rsa  id_rsa.pub
```

3)把Master上的~/.ssh/id_rsa.pub文件追加到Slave1和Slave2的~/.ssh/authorized_keys

首先将master公钥id_rsa.pub传送到Slave1和Slave2的~/.ssh/

sudo scp id_rsa.pub baoling@Slave1:/home/baoling/.ssh/

sudo scp id_rsa.pub baoling@Slave2:/home/baoling/.ssh/

然后分别在Slave1和Slave2上将~/.ssh/id_rsa.pub追加或是复制到~/.ssh/authorized_keys

cp id_rsa.pub authorized_keys

或

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

备注：配置完毕，如果Master仍然不能访问Worker，可以修改Worker的authorized_keys文件的权限，命令为 chmod 600 ~/.ssh/authorized_keys

四、安装Hadoop

准备工作：

hadoop-2.7.2下载：http://mirrors.hust.edu.cn/apache/hadoop/

解压hadoop-2.7.2.tar.gz：tar -zxf hadoop-2.7.2.tar.gz /usr/sparkdir/hadoop/

备注：sudo chown -R baoling:spark /usr/sparkdir/hadoop

配置环境变量：

/etc/profile

```
export HADOOP_HOME=/usr/sparkdir/hadoop/hadoop-2.7.2

export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

export HADOOP_COMMON_HOME=$HADOOP_HOME

export HADOOP_HDFS_HOME=$HADOOP_HOME

export HADOOP_MAPRED_HOME=$HADOOP_HOME

export HADOOP_YARN_HOME=$HADOOP_HOME

#export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop

#export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native

#export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#export JAVA_HOME=/usr/sparkdir/hadoop/hadoop-2.7.2
```

编辑配置文件

进入/usr/sparkdir/hadoop/hadoop-2.7.2/etc/hadoop/进行如下配置(涉及文件包括：hadoop-env.sh、core-site.xml、yarn-site.xml、mapred-site.xml)

1)配置hadoop-env.sh文件

export JAVA_HOME=**/usr/sparkdir/java/jdk1.8.0_91**

2)配置core-site.xml文件

```
<configuration>
  /*这里的值指的是默认的HDFS路径*/
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://Master:9000</value>
  </property>

  /*缓冲区大小: io.file.buffer.size默认是4KB*/
  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>

  /*临时文件夹路径*/
  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:/usr/sparkdir/temp</value>
    <description> Abase for other temporary directories </description>
  </property>

  <property>
    <name>hadoop.proxyuser.hduser.hosts</name>
    <value>*</value>
  </property>

  <property>
    <name>hadoop.proxyuser.hduser.groups</name>
    <value>*</value>
  </property>
```

```
</configuration>
```

3)配置yarn-site.xml文件

```
<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>

    /*resourceManager的地址*/
    <property>
        <name>yarn.resourcemanager.address</name>
        <value>Master:18040</value>
    </property>

    /*调度器的端口*/
    <property>
        <name>yarn.resourcemanager.scheduler.address</name>
        <value>Master:18030</value>
    </property>

    /*resource-tracker端口*/
    <property>
        <name>yarn.resourcemanager.resource-tracker.address</name>
        <value>Master:8031</value>
    </property>

    /*resourcemanager管理器端口*/
    <property>
```

<name>yarn.resourcemanager.admin.address</name>

<value>Master:8033</value>

</property>

/*ResourceManager 的Web端口、监控 job 的资源调度*/

<property>

<name>yarn.resourcemanager.webapp.address</name>

<value>Master:8088</value>

</property>

</configuration>

4)配置mapred-site.xml文件

备注：看到的是mapred-site.xml.template，因此需要复制一份 --> sudo cp mapred-site.xml.template mapred-site.xml

^-_-^发现只有读的权限，因此需要--> sudo chown -R baoling:spark /usr/sparkdir/hadoop/hadoop-2.7.2/etc/hadoop/mapred-site.xml

<configuration>

/*hadoop对map-reduce运行矿建一共提供了3种实现，在mapred-site.xml中通过“mapreduce.framework.name”这个属性来设置为"classic"."yarn"或者“local”*/

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

/*MapReduce JobHistory Server地址*/

<property>

<name>mapreduce.jobhistory.address</name>

<value>Master:10020</value>

</property>

/*MapReduce JobHistory Server web UI 地址*/

<property>

<name>mapreduce.jobhistory.webapp.address</name>

<value>Master:19888</value>

</property>

```
    <property>
      <name>mapred.job.tracker</name>
      <value>Master:9001</value>
    </property>

</configuration>
```

创建namenode和datanode目录，并配置其相应路径

1)创建namenode和datanode目录

```
/usr/spark
        /hdfs
            /namenode
            /datanode
cd /usr/sparkdir
sudo mkdir /hdfs
cd ./hdfs
sudo mkdir namenode
sudo mkdir datanode
```

2)执行命令后，再次回到/usr/sparkdir/hadoop/hadoop-2.7.2，配置hdfs-site.xml文件

```
<configuration>

  /*配置主节点名和端口*/

  <property>

    <name>dfs.namenode.secondary.http-address</name>

    <value>Master:9001</value>

  </property>


  /*配置从节点和端口号*/

  <property>

    <name>dfs.namenode.name.dir</name>

    <value>file:/usr/sparkdir/hdfs/namenode</value>

  </property>


  /*配置datanode的数据存储目录*/

  <property>

    <name>dfs.datanode.data.dir</name>

    <value>file:/usr/sparkdir/hdfs/datanode</value>

  </property>


  /*配置副本数*/

  <property>

    <name>dfs.replication</name>
```

```
    <value>3</value>
  </property>
```

/*将dfs.webhdfs.enabled属性设置为true,否则就不能使用webhdfs的LISTSTATUS，LISTFILESTATUS等需要列出文件，文件夹状态的命令，因为这些信息都是由namenode保存的*/

```
  <property>
    <name>dfs.webhdfs.enabled</name>
    <value>true</value>
  </property>
</configuration>
```

备注：以上参数配置仅供Hadoop平台搭建学习之用，或有许多纰漏，请读者自行参见**“Apache Hadoop-2.7.2官方使用文档”** http://hadoop.apache.org/docs/current/index.html

配置Master和Slave文件

1)master文件负责配置主节点的主机名

备注：最开始没有master文件，需创建 --> sudo gedit master

```
#Master为主节点主机名
Master
```

2)配置slaves文件添加从节点主机名

```
#Slave为从节点主机名
Slave1
Slave2
```

Hadoop文件复制

不安全的做法

Master:

```
sudo chown -R baoling:spark /usr/sparkdir/
sudo chmod -R 777 /usr/sparkdir/
```

Slava1:

```
sudo chown -R baoling:spark /usr/sparkdir/
sudo chmod -R 777 /usr/sparkdir/
```

Slava2:

```
sudo chown -R baoling:spark /usr/sparkdir/
sudo chmod -R 777 /usr/sparkdir/
```

Master:

```
sudo scp -r /usr/sparkdir  baoling@Slave1:/usr/
sudo scp -r /usr/sparkdir  baoling@Slave2:/usr/
```

备注：当然也可以Hadoop所有文件通过pssh发送到各个节点（暂时没有去配置）

Think Time（在Slave1和Slave2节点还遗漏了什么？）

在Slave1和Slaves的/etc/profile文件里没有对jdk、scala、hadoop进行配置

```
sudo gedit /etc/profile
```

文件末尾增加如下内容：

```
#JDK Configuration
JAVA_HOME=/usr/sparkdir/java/jdk1.8.0_91 #Please modify JAVA_HOME
PATH=$JAVA_HOME/bin:$PATH
CLASSPATH=.:$JAVA_HOME/jre/lib/rt.jar:$JAVA_HOME/jre/lib/dt.jar:$JAVA_HOME/jre/lib/tools.jar
export JAVA_HOME PATH CLASSPATH

#SCALA Configuration
export SCALA__HOME=/usr/sparkdir/scala/scala-2.11.8 #Please modify SCALA_HOME
export PATH=${SCALA_HOME}/bin:$PATH

#HADOOP Configuration
export HADOOP_HOME=/usr/sparkdir/hadoop/hadoop-2.7.2
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop

#综合起来配置
export JAVA_HOME=/usr/sparkdir/java/jdk1.8.0_91
export JRE_HOME=${JAVA_HOME}/jre
export SCALA_HOME=/usr/sparkdir/scala/scala-2.11.8
export HAD00P_HOME=/usr/sparkdir/hadoop/hadoop-2.7.2
export CLASS_PATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export
PATH=${JAVA_HOME}/bin:${SCALA_HOME}/bin:${HAD00P_HOME}/bin:${HAD00P_HOME}/sbin:$PATH
```

格式化Namenode（/usr/sparkdir/hadoop/hadoop-2.7.2/目录下）

./bin/hadoop namenode -format

OR hadoop/namenode-format

启动Hadoop（/usr/sparkdir/hadoop/hadoop-2.7.2/目录下）

./sbin/start-all.sh

```
baoling@Master:/usr/sparkdir/hadoop/hadoop-2.7.2$ ./sbin/start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [Master]
Master: starting namenode, logging to /usr/sparkdir/hadoop/hadoop-2.7.2/logs/hadoop-baoling-namenode-Master.out
Slave2: starting datanode, logging to /usr/sparkdir/hadoop/hadoop-2.7.2/logs/hadoop-baoling-datanode-Slave2.out
Slave1: starting datanode, logging to /usr/sparkdir/hadoop/hadoop-2.7.2/logs/hadoop-baoling-datanode-Slave1.out
Starting secondary namenodes [Master]
Master: starting secondarynamenode, logging to /usr/sparkdir/hadoop/hadoop-2.7.2/logs/hadoop-baoling-secondarynamenode-Master.out
starting yarn daemons
starting resourcemanager, logging to /usr/sparkdir/hadoop/hadoop-2.7.2/logs/yarn-baoling-resourcemanager-Master.out
Slave2: starting nodemanager, logging to /usr/sparkdir/hadoop/hadoop-2.7.2/logs/yarn-baoling-nodemanager-Slave2.out
Slave1: starting nodemanager, logging to /usr/sparkdir/hadoop/hadoop-2.7.2/logs/yarn-baoling-nodemanager-Slave1.out
```

备注：

查看日志/usr/sparkdir/hadoop/hadoop-2.7.2/logs/

hadoop-baoling-namenode-Master.log

yarn-baoling-resourcemanager-Master.log

yarn-baoling-resourcemanager-Master.log(**通过查询发现如下三个启动问题**)

ERROR org.apache.hadoop.yarn.server.resourcemanager.ResourceManager: Returning, interrupted : java.lang.InterruptedException

ERROR org.apache.hadoop.security.token.delegation.AbstractDelegationTokenSecretManager: ExpiredTokenRemover received java.lang.InterruptedException: sleep interrupted

FATAL org.apache.hadoop.yarn.server.resourcemanager.ResourceManager: **Error** starting ResourceManager

org.apache.hadoop.yarn.exceptions.YarnRuntimeException:

java.net.BindException: Problem binding to [Master:8030] java.net.BindException: Address already in use;
For more details see: http://wiki.apache.org/hadoop/BindException

解决：

第一第二个错误改变/tmp权限-->**sudo chmod -R 777 /tmp/**

第三个错误改变yarn-site.xml的yarn.resourcemanager.address属性以及yarn.resourcemanager.scheduler.address属性，此处分别为Master:18040和Master:18030

yarn-baoling-resourcemanager-Master.log

ERROR org.apache.hadoop.hdfs.server.namenode.NameNode: RECEIVED SIGNAL 15: SIGTERM

解决：改变/tmp权限-->**sudo chmod -R 777 /tmp/**

hadoop-baoling-datanode-Slave1.log(**通过查询发现如下三个启动问题**)

FATAL org.apache.hadoop.hdfs.server.datanode.DataNode: Initialization failed for Block pool <registering> (Datanode Uuid unassigned) service to Master/192.168.1.116:9000. Exiting.

yarn-baoling-nodemanager-Slave1.log

解决：http://www.cnblogs.com/kinglau/p/3796274.html

hadoop-baoling-datanode-Slave2.log(**通过查询发现如下三个启动问题**)

FATAL org.apache.hadoop.hdfs.server.datanode.DataNode: Initialization failed for Block pool <registering> (Datanode Uuid unassigned) service to Master/192.168.1.116:9000. Exiting.

yarn-baoling-nodemanager-Slave2.log

解决：http://www.cnblogs.com/kinglau/p/3796274.html

其它错误：**-- process information unavailable**

进入**/tmp**，删除名称为**hsperfdata_{username}-->(此处为hsperfdata_baoling)**的文件夹，然后重新启动Hadoop

查看Hadoop启动情况

jps

Master

```
baoling@Master:/usr/sparkdir/hadoop/hadoop-2.7.2$ jps
5012 NameNode
5224 SecondaryNameNode
5628 Jps
5375 ResourceManager
```

Slave1

利用ssh登录Slave1并利用jps命令查看

```
baoling@Master:/usr/sparkdir/hadoop/hadoop-2.7.2$ ssh Slave1
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

145 packages can be updated.
0 updates are security updates.

Last login: Sun Jun  5 14:22:45 2016 from 192.168.1.116
baoling@Slave1:~$ jps
3092 DataNode
3222 NodeManager
3405 Jps
```

Slave2

利用ssh登录Slave1并利用jps命令查看

```
baoling@Slave1:~$ exit
logout
Connection to slave1 closed.
baoling@Master:/usr/sparkdir/hadoop/hadoop-2.7.2$ ssh Slave2
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

145 packages can be updated.
0 updates are security updates.

Last login: Sun Jun  5 14:23:00 2016 from 192.168.1.116
baoling@Slave2:~$ jps
3312 NodeManager
3186 DataNode
3502 Jps
```

启动Hadoop（/usr/sparkdir/hadoop/hadoop-2.7.2/目录下）

./sbin/stop-all.sh

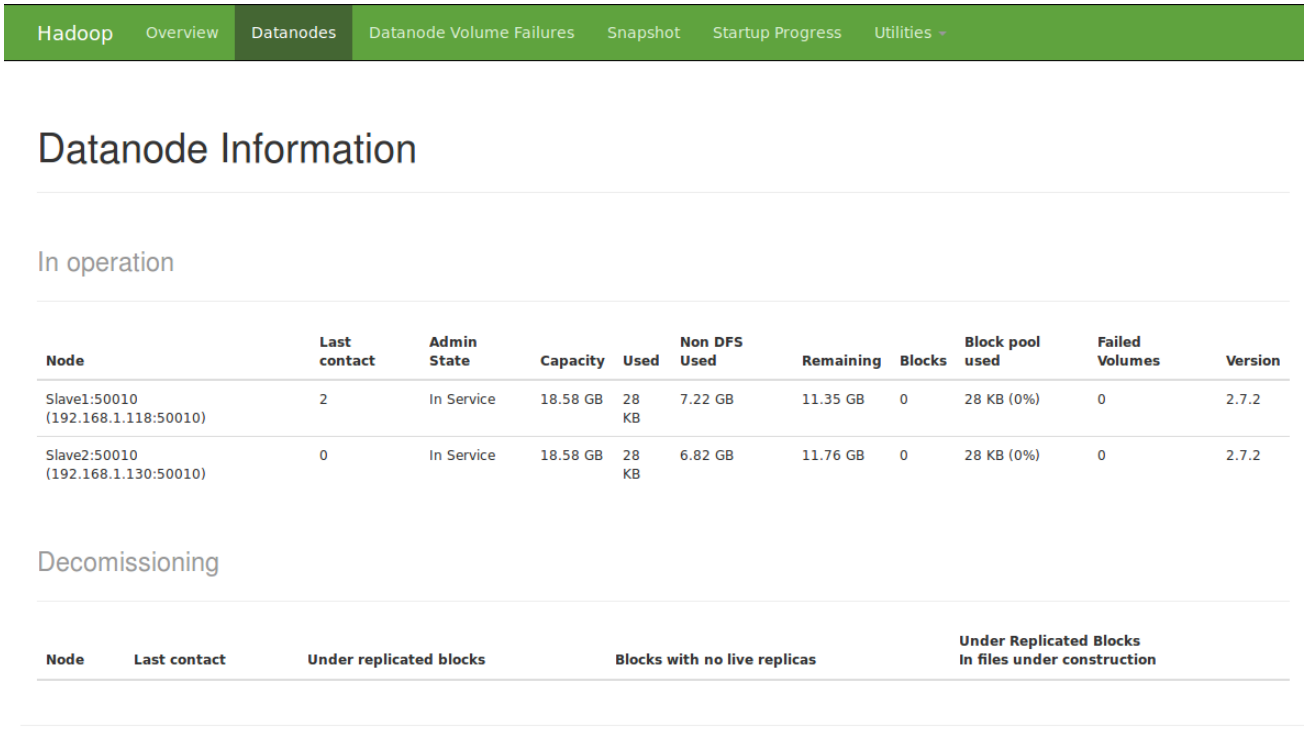
```
baoling@Master:/usr/sparkdir/hadoop/hadoop-2.7.2$ ./sbin/stop-all.sh
This script is Deprecated. Instead use stop-dfs.sh and stop-yarn.sh
Stopping namenodes on [Master]
Master: stopping namenode
Slave2: stopping datanode
Slave1: stopping datanode
Stopping secondary namenodes [Master]
Master: stopping secondarynamenode
stopping yarn daemons
stopping resourcemanager
Slave2: stopping nodemanager
Slave1: stopping nodemanager
no proxyserver to stop
baoling@Master:/usr/sparkdir/hadoop/hadoop-2.7.2$ jps
6168 Jps
```

安装成功验证

查看机器集群状态

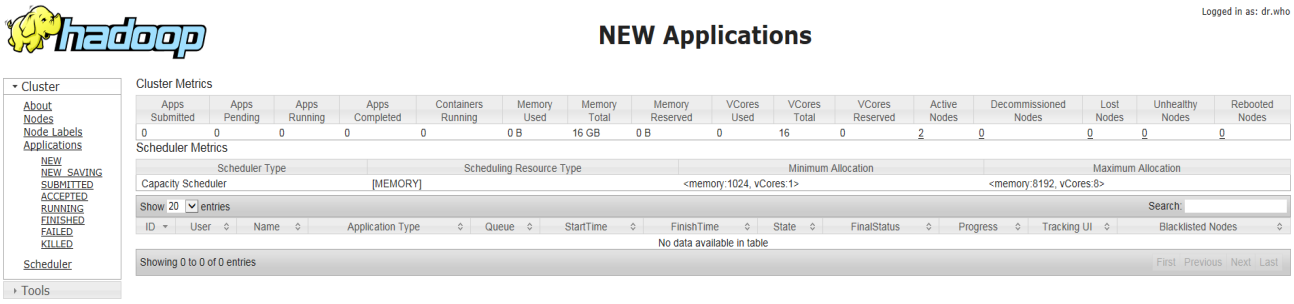
http://master:50070 OR http://192.168.1.114:5007

可以看到当前的live nodes有slave1和slave2两个节点信息



备注：hadoop2.x取消jobtraker，因此也没有http://master:50030管理页面。出现Live Nodes为0，请参见http://www.linuxidc.com/Linux/2012-03/57749.htm

http://master:8088 OR http://192.168.1.114:8088

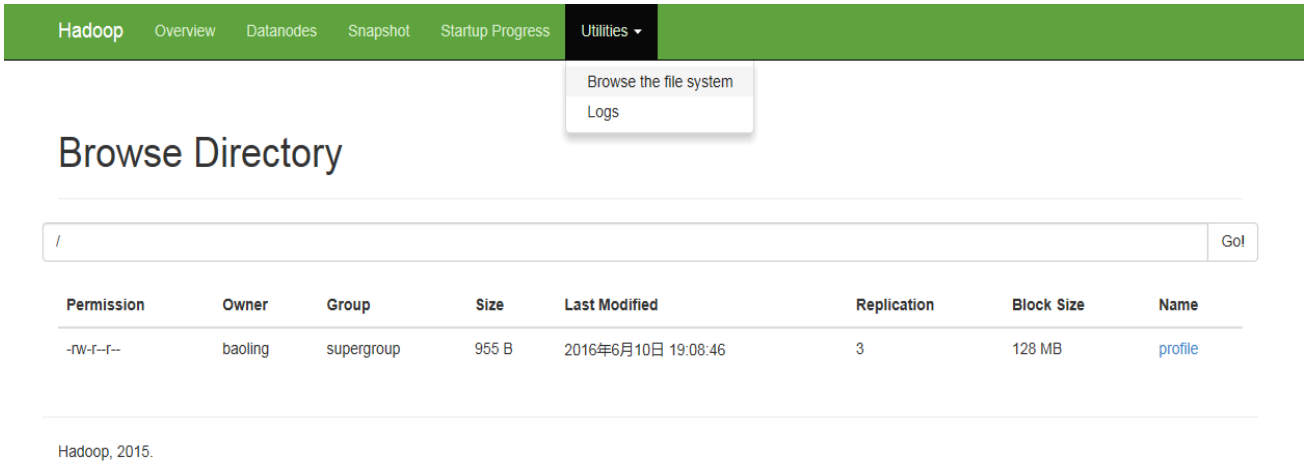


输入以下命令上传文件到hadoop(**新建HDFS文件夹/testing　hadoop fs -mkdir /testing**) -->

hadoop fs 命令可参看http://www.blogjava.net/changedi/archive/2013/08/12/402696.html

hadood fs -put /etc/profile /testing

查看50070页面的显示结果



至此，Hadoop开发平台就搭建完毕了

参考：http://blog.csdn.net/lovehuangjiaju/article/details/46883973

五、安装Spark（以Spark Standalone为例）

准备工作：

下载spark-1.6.1-bin-hadoop2.6.tgz：http://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-1.6.1/spark-1.6.1-bin-hadoop2.6.tgz

解压spark-1.6.1-bin-hadoop2.6.tgz：tar -zxf spark-1.6.1-bin-hadoop2.6.tgz -C /usr/sparkdir/spark/

配置工作：

修改/etc/profile文件

```
export SPARK_HOME=/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6
export PATH=${SPARK_HOME}/bin:${SPARK_HOME}/sbin:$PATH

#综合起来配置
export JAVA_HOME=/usr/sparkdir/java/jdk1.8.0_91
export JRE_HOME=${JAVA_HOME}/jre
export SCALA_HOME=/usr/sparkdir/scala/scala-2.11.8
export HAD00P_HOME=/usr/sparkdir/hadoop/hadoop-2.7.2
export SPARK_HOME=/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6
export CLASS_PATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:${SCALA_HOME}/bin:${HAD00P_HOME}/bin:${HAD00P_HOME}/sbin:${SPARK_HOME}/bin:${SPARK_HOME}/sbin:$PATH
```

备注：进入/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6/conf

配置conf/spark-defaults.conf文件：（**spark-defaults.conf.template 复制一份 spark-default.sh --> cp spark-defaults.conf.template spark-defaults.conf**）

```
spark.master=spark://Master:7077
spark.eventLog.enabled=true
#hdfs://Master:9000是前面core-site.xml中定义的hs.defaultFS属性值
spark.eventLog.dir=hdfs://Master:9000/testing
```

备注：新建HDFS文件夹/testing　hadoop fs -mkdir /testing

配置conf/spark-env.sh文件：（**spark-env.sh.template 复制一份 spark-env.sh --> cp spark-env.sh.template spark-env.sh**）

export JAVA_HOME=/usr/sparkdir/java/jdk1.8.0_91
export HADOOP_CONF_DIR=/usr/sparkdir/hadoop/hadoop-2.7.2/etc/hadoop
SPARK_DRIVER_MEMORY=1000M

配置conf/slaves文件：(**slaves.template 复制一份 spark-env.sh --> cp slaves.template slaves**)

#加入数据节点主机名
Slave1
Slave2

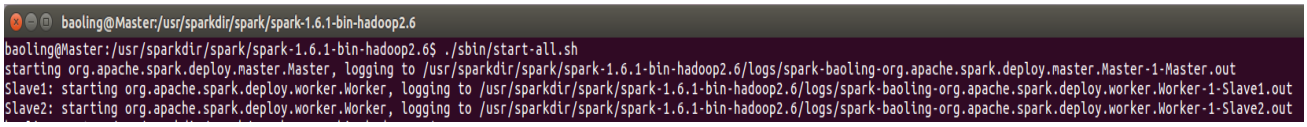
复制spark-1.6.1-bin-hadoop2.6到Slave1、Slave2的/usr/sparkdir/spark/

scp -r ./spark-1.6.1-bin-hadoop2.6/ baoling@Slave1:/usr/sparkdir/spark/
scp -r ./spark-1.6.1-bin-hadoop2.6/ baoling@Slave2:/usr/sparkdir/spark/

Spark启动与关闭：

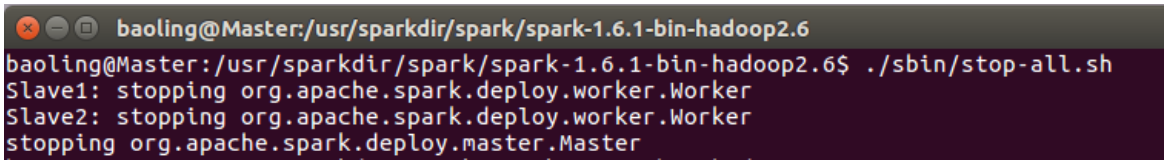
1)在Spark根目录启动Spark

./sbin/start-all.sh



2)关闭Spark

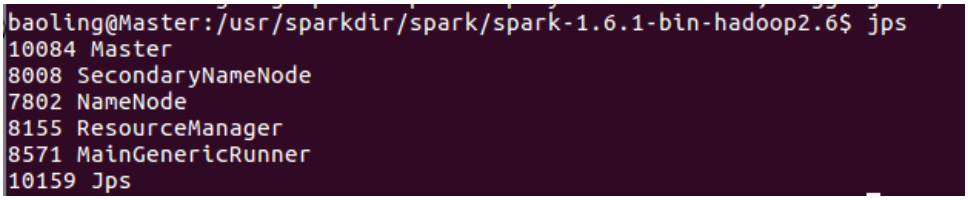
./sbin/stop-all.sh



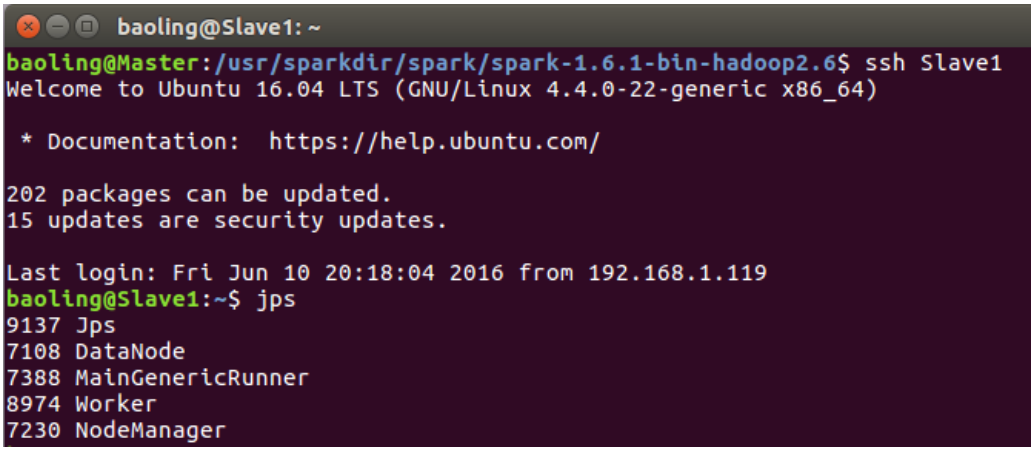
安装成功验证

jps

Master



Slave1



Slave2

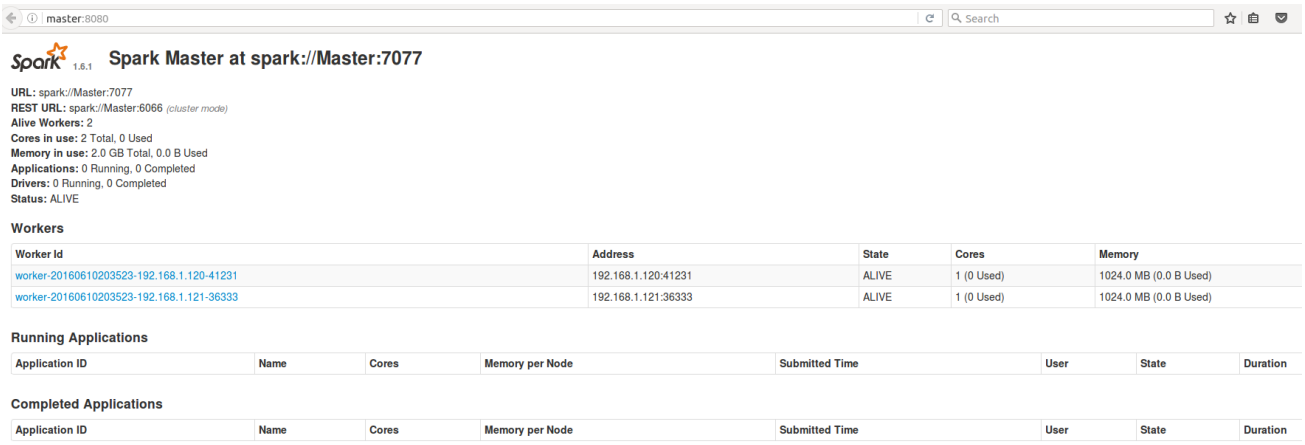
```
baoling@Slave2: ~
baoling@Master:/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6$ ssh Slave2
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

187 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Fri Jun 10 20:19:29 2016 from 192.168.1.119
baoling@Slave2:~$ jps
21728 Worker
21895 Jps
20204 NodeManager
20077 DataNode
```

浏览器中输入http://master:8080 OR http://192.168.1.114:8080/



运行示例

./bin/run-example SparkPi 10

备注：出现如下错误

```
baoling@Master: /usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6
16/06/10 20:54:46 INFO storage.BlockManager: BlockManager stopped
16/06/10 20:54:46 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
16/06/10 20:54:46 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/06/10 20:54:46 INFO spark.SparkContext: Successfully stopped SparkContext
16/06/10 20:54:46 INFO remote.RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
Exception in thread "main" 16/06/10 20:54:46 INFO remote.RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
java.net.ConnectException: Call From Master/192.168.1.119 to Master:9000 failed on connection exception: java.net.ConnectException: Connection refused; For more details see:  http://wiki.
apache.org/hadoop/ConnectionRefused
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
        at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
        at org.apache.hadoop.net.NetUtils.wrapWithMessage(NetUtils.java:791)
        at org.apache.hadoop.net.NetUtils.wrapException(NetUtils.java:731)
        at org.apache.hadoop.ipc.Client.call(Client.java:1472)
        at org.apache.hadoop.ipc.Client.call(Client.java:1399)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEngine.java:232)
        at com.sun.proxy.$Proxy12.getFileInfo(Unknown Source)
```

根据提示进入网站：http://wiki.apache.org/hadoop/ConnectionRefused

但是这个问题没有给出答案，问题出在用户的配置（暂时这样，后面解决）

成功运行后截图

```
16/06/11 11:19:51 INFO spark.MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
16/06/11 11:19:51 INFO storage.MemoryStore: MemoryStore cleared
16/06/11 11:19:51 INFO storage.BlockManager: BlockManager stopped
16/06/11 11:19:51 INFO storage.BlockManagerMaster: BlockManagerMaster stopped
16/06/11 11:19:51 INFO scheduler.OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
16/06/11 11:19:52 INFO spark.SparkContext: Successfully stopped SparkContext
16/06/11 11:19:52 INFO remote.RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
16/06/11 11:19:52 INFO remote.RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.
16/06/11 11:19:52 INFO util.ShutdownHookManager: Shutdown hook called
16/06/11 11:19:52 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-7a492251-010d-4c24-9978-1103021eccb5/httpd-427053d8-1a18-4cfd-be39-ea04d3cd09bd
16/06/11 11:19:52 INFO util.ShutdownHookManager: Deleting directory /tmp/spark-7a492251-010d-4c24-9978-1103021eccb5
baoling@Master:/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6$
```

集群程序运行测试

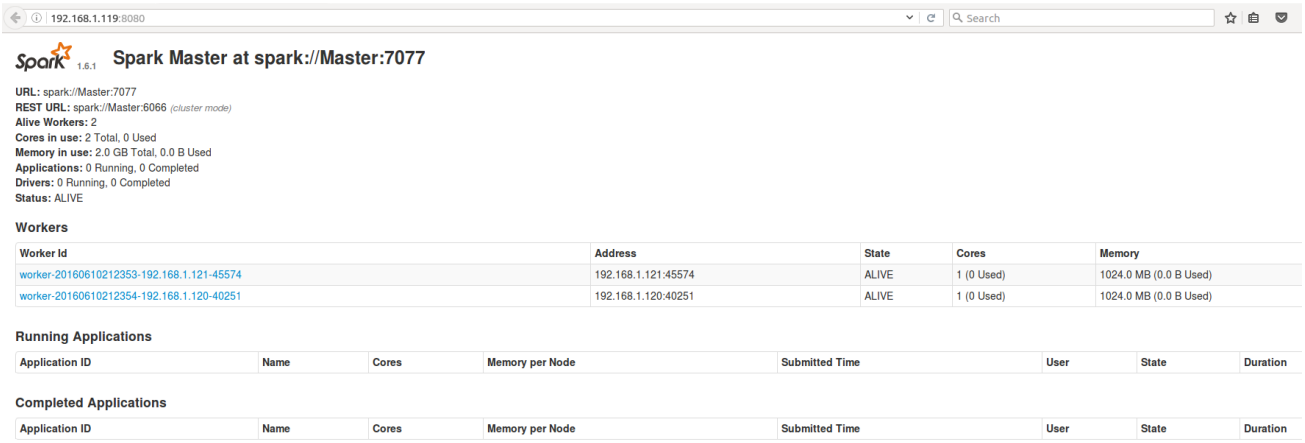
上传 README.md文件到hdfs /目录

hadoop fs -put /usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6/README.md /testing

顺便查看文件/testing hadoop fs -ls /testing

```
baoling@Master:/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6$ hadoop fs -ls /testing
Found 5 items
-rw-r--r--   3 baoling supergroup       3359 2016-06-10 21:24 /testing/README.md
-rwxrwx---   3 baoling supergroup       5340 2016-06-11 10:34 /testing/app-20160611102844-0001
-rwxrwx---   3 baoling supergroup        234 2016-06-11 10:38 /testing/app-20160611103848-0002.inprogress
-rwxrwx---   3 baoling supergroup      19486 2016-06-11 11:19 /testing/local-1465615163275
-rw-r--r--   3 baoling supergroup        955 2016-06-10 19:08 /testing/profile
```

在Master节点，进入 /usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6目录，执行./bin/spark-shell，刷新http://sparkmaster:8080后可以看到以下内容：



执行spark-shell截图

```
baoling@Master:/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6$ ./bin/spark-shell
16/06/11 11:27:21 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
classes where applicable
16/06/11 11:27:21 INFO spark.SecurityManager: Changing view acls to: baoling
16/06/11 11:27:21 INFO spark.SecurityManager: Changing modify acls to: baoling
16/06/11 11:27:21 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view p
ermissions: Set(baoling); users with modify permissions: Set(baoling)
16/06/11 11:27:22 INFO spark.HttpServer: Starting HTTP Server
16/06/11 11:27:22 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/06/11 11:27:22 INFO server.AbstractConnector: Started SocketConnector@0.0.0.0:43609
16/06/11 11:27:22 INFO util.Utils: Successfully started service 'HTTP class server' on port 43609.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.6.1
      /_/
```

备注：执行的时候出现如下问题

```
baoling@Master: /usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6
baoling@Master:/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6$ ./bin/spark-shell
Java HotSpot(TM) 64-Bit Server VM warning: INFO: os::commit_memory(0x00000000d5550000, 715849728, 0) failed; error='Cannot allocate memory' (errno=12)
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Native memory allocation (mmap) failed to map 715849728 bytes for committing reserved memory.
# An error report file with more information is saved as:
# /usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6/hs_err_pid5070.log
```

查看对应日志发现是因为：**# Out of Memory Error (os_linux.cpp:2627), pid=5070, tid=140020683282176 (** *机器物理内存不足* free -m **)**

http://blog.csdn.net/pengych_321/article/details/51252911

输入如下语句：

```
val textCount = sc.textFile("/testing/README.md").filter(line => line.contains("Spark")).count()
```

```
scala> val textCount = sc.textFile("/testing/README.md").filter(line => line.contains("Spark")).count()
16/06/11 13:30:07 INFO storage.MemoryStore: Block broadcast_0 stored as values in memory (estimated size 211.6 KB, free 211.6 KB)
16/06/11 13:30:07 INFO storage.MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated size 19.7 KB, free 231.3 KB)
16/06/11 13:30:08 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on 192.168.1.119:36084 (size: 19.7 KB, free: 500.0 MB)
16/06/11 13:30:08 INFO spark.SparkContext: Created broadcast 0 from textFile at <console>:27
16/06/11 13:30:08 INFO mapred.FileInputFormat: Total input paths to process : 1
16/06/11 13:30:09 INFO spark.SparkContext: Starting job: count at <console>:27
16/06/11 13:30:09 INFO scheduler.DAGScheduler: Got job 0 (count at <console>:27) with 2 output partitions
16/06/11 13:30:09 INFO scheduler.DAGScheduler: Final stage: ResultStage 0 (count at <console>:27)
16/06/11 13:30:09 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/06/11 13:30:09 INFO scheduler.DAGScheduler: Missing parents: List()
16/06/11 13:30:09 INFO scheduler.DAGScheduler: Submitting ResultStage 0 (MapPartitionsRDD[2] at filter at <console>:27), which has no missing parents
16/06/11 13:30:10 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 3.1 KB, free 234.5 KB)
16/06/11 13:30:10 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 1883.0 B, free 236.3 KB)
16/06/11 13:30:10 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on 192.168.1.119:36084 (size: 1883.0 B, free: 500.0 MB)
16/06/11 13:30:10 INFO spark.SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1006
16/06/11 13:30:10 INFO scheduler.DAGScheduler: Submitting 2 missing tasks from ResultStage 0 (MapPartitionsRDD[2] at filter at <console>:27)
16/06/11 13:30:10 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 2 tasks
16/06/11 13:30:10 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, Slave2, partition 0,NODE_LOCAL, 2138 bytes)
16/06/11 13:30:10 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, Slave1, partition 1,NODE_LOCAL, 2138 bytes)
16/06/11 13:30:14 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on Slave1:42877 (size: 1883.0 B, free: 517.4 MB)
16/06/11 13:30:14 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on Slave2:33264 (size: 1883.0 B, free: 517.4 MB)
16/06/11 13:30:17 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on Slave2:33264 (size: 19.7 KB, free: 517.4 MB)
16/06/11 13:30:19 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on Slave1:42877 (size: 19.7 KB, free: 517.4 MB)
16/06/11 13:30:26 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 16113 ms on Slave1 (1/2)
16/06/11 13:30:26 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 16259 ms on Slave2 (2/2)
16/06/11 13:30:26 INFO scheduler.DAGScheduler: ResultStage 0 (count at <console>:27) finished in 16.309 s
16/06/11 13:30:26 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
16/06/11 13:30:26 INFO scheduler.DAGScheduler: Job 0 finished: count at <console>:27, took 17.312491 s
textCount: Long = 17
```

基于如上配置，快速搭建spark-1.6.1-bin-without-hadoop

下载部署包sparkdir.tar.gz

链接　http://pan.baidu.com/s/1c1PH45a

密码　2jc9

一、创建用户及用户组、配置hosts及hostname、配置环境变量、配置ssh、新建文件夹/usr/sparkdir

1、创建用户及用户组(所有机器)

sudo adduser baoling　#创建用户baoling

sudo groupadd spark　#创建用户组spark

sudo usermod -a -G spark baoling　#将用户baoling加入用户组spark

相关命令：

groupdel　#删除用户组

gpasswd　#将用户从用户组中删除

备注：区分adduser、deluser、useradd、userdel

2、配置hosts和hostname(所有机器)

sudo gedit /etc/hosts

添加：

192.168.1.114　Master

192.168.1.118　Slave1

192.168.1.130　Slave2

备注：这里的ip是对应主机的ip，可以通过ifconfig查到，此处也不做永久ip配置

sudo gedit /etc/hostname

分别添加：

Master --> Master
Slave1 --> Slave1
Slave2 --> Slave2

3、配置环境变量(所有机器)

sudo gedit /etc/profile

```
export JAVA_HOME=/usr/sparkdir/java/jdk1.8.0_91
export JRE_HOME=${JAVA_HOME}/jre
export SCALA_HOME=/usr/sparkdir/scala/scala-2.11.8
export HAD00P_HOME=/usr/sparkdir/hadoop/hadoop-2.7.2
export SPARK_HOME=/usr/sparkdir/spark/spark-1.6.1-bin-hadoop2.6
export CLASS_PATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export
PATH=${JAVA_HOME}/bin:${SCALA_HOME}/bin:${HAD00P_HOME}/bin:${HAD00P_HOME}/sbin:${SPARK_HOME}/bin:${SPARK_HOME}/sbin:$PATH
```

4、配置ssh(所有机器)

在Master生成秘钥

ssh-keygen -t rsa

在各个Slave创建文件夹~/.ssh

mkdir ~/.ssh

在Master将秘钥传输到各个Slave(此时需要输入密码)

sudo scp id_rsa.pub baoling@Slave1:/home/baoling/.ssh/

sudo scp id_rsa.pub baoling@Slave2:/home/baoling/.ssh/

在所有机器执行如下命令

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

5、新建文件夹/usr/sparkdir(所有机器)

sudo mkdir /usr/sparkdir

sudo chown -R baoling:spark /usr/sparkdir

sudo chmod -R 777 /usr/sparkdir

二、在Master解压sparkdir.tar.gz到/usr/、传送到Slave

1、解压(Master)

sudo tar -zxf ./sparkdir.tar.gz -C /usr

2、传送到Slave(在Master)

scp -r /usr/sprkdir baoling@Slave1:/usr/

scp -r /usr/sprkdir baoling@Slave2:/usr/

三、测试

略