## 1.编译hadoop2.2源码

由于在apache官网上下载的hadoop二进制包中native函数库还是在32位，在我们一般的使用的机器系统都是64位的，所以要编译安装hadoop2.2

\* Unix System

\* JDK 1.6+

\* Maven 3.0 or later

\* Findbugs 1.3.9 (if running findbugs)

\* ProtocolBuffer 2.5.0

\* CMake 2.6 or newer (if compiling native code)

\* Internet connection for first build (to fetch all Maven and Hadoop dependencies)

上面就是下载hadoop2.2源码中README中安装的需求，按照上面的要求安装：

**JDK1.6+ （jdk1.8太新好像不行）**

**Maven （编译项目，这是一个很好的项目管理工具）**

**Findbugs （可以不装）**

**ProtocolBuffer （必须要装，因为编译的时候依赖这个函数库）**

**Cmake**

**联网（maven 编译时要联网下载一些软件包）**

除了这些之外还要安装**libssl-dev，**以上的软件下载对应的源码包安装就行，注意检测安装是否成功。

由于hadoop2.2本身就存在一个bug，

问题描述：<https://issues.apache.org/jira/browse/HADOOP-10110>

解决方法：<https://issues.apache.org/jira/secure/attachment/12614482/HADOOP-10110.patch>

安装ssh和配置ssh无密码登录

$ sudo apt-get install openssh-server

$ cd ~/.ssh

$ ssh-keygen -t rsa #一直按回车就行

$ cat ./id\_rsa.pub >> ./authorized\_keys # 加入授权

测试一下：

$ ssh localhost # 是不是没输密码就行

$ exit #退出

编译命令：

在编译之后会生产一个二进制的压缩包，直接部署到环境中就行

## 2. 配置hadoop2.2单机和伪分布式

将压缩文件解压到/etc/local/hadoop

配置环境变量 将/etc/local/hadoop/bin（常用的hadoop命令） /etc/local/hadoop/sbin（常用的hadoop脚本） 加入到PATH 中

测试 ： $ hadoop version

**单机配置：**

单机式安装就是在安装完成之后就行了，hadoop默认的就是单机配置。

运行例子：

$./bin/hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-\*.jar grep ./input ./output 'dfs[a-z.]+'

**注意：**hadoop默认不会覆盖输出结果，所以要是还是输出到相同的地方，在执行命令之前要把文件夹删除，不然可能报错。

查看结果：

$ car ./output/\*

**伪分布式配置**

Hadoop可以在单节点上以伪分布式的方式运行，节点既是namenode也是DataNode，同时读取hdfs的内容。

伪分布式需要修改两个配置文件（存在/usr/local/hadoop/etc/hadoop）core-site.xml，hdfs-site.xml

Core-site.xml修改：



Hdfs-site.xml设置：



配置完成之后，执行namenode的格式化

$ hdfs namenode -format

会看到 “successfully formatted” 和 “Exitting with status 0” 的提示，若为 “Exitting with status 1” 则是出错。

在/etc/local/hadoop/etc/hadoop/hadoop-env.sh 中找到JAVA\_HOME,将其改成对应的jdk的安装路径。

开启namenode和datanode的守护进程。

$ start-dfs.sh

这些脚本存放在/usr/local/hadoop/sbin 中，在前面设置环境变量（PATH）后，打上前面几个字母TAB就有提示。

启动完成后，可以通过命令 jps 来判断是否成功启动，若成功启动则会列出如下进程: “NameNode”、”DataNode” 和 “SecondaryNameNode”（如果 SecondaryNameNode 没有启动，请运行 sbin/stop-dfs.sh 关闭进程，然后再次尝试启动尝试）。如果没有 NameNode 或 DataNode ，那就是配置不成功，请仔细检查之前步骤，或通过查看启动日志排查原因。

## 3.运行hadoop伪分布式实例

上面单机模式读取的是本地input文件夹的数据，伪分布式读取的是hdfs上的数据，要使用HDFS就要要创建用户目录

$ hdfs dfs -mkdir -p /user/xtx

目录名自己起就行。

将hadoop配置文件\*.xml都复制都input中

$ hdfs dfs -mkdir input

$ hdfs dfs -put ./etc/hadoop/\*.xml input

复制完成之后查看 input文件夹中内容：

$ hdfs dfs -ls input

执行命令：

$ hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-\*.jar grep input output 'dfs[a-z.]+'

查看结果：

$ hdfs dfs -cat output/\*

可以将hdfs上的数据存到本地上

$ hdfs dfs -get output ./output

关闭hadoop： $ stop-dfs.sh

## 4.启动yarn

YARN是从MAPREDUCE中分离出来的，负责资源管理和任务调度，

修改配置文件 mapred-site.xml

$ mv ./etc/hadoop/mapred-site.xml.template ./etc/hadoop/mapred-site.xmls

编辑内容：



修改yarn-site.xml



启动yarn（前提执行过start-dfs.sh）

$ start-yarn.sh

$ mr-jobhistory-daemon.sh start historyserver # 开启历史服务器，才能在Web中查看任务运行情况

开启通过jps查看，多了NodeManager，ResourceManager两个后台进程。

启动yarn的一个好处就是可以在web‘查看任务的运行情况：

<http://localhost:8088/cluster>’

## 5.hadoop集群安装配置

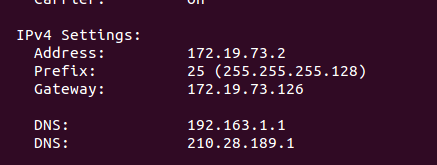
### （1）网络配置

因为是使用虚拟机的原因，网络配置必须选择桥接。

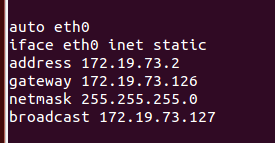
因为slave是master复制过去的，所以slave机器的mac地址要更新

在虚拟机开关机之后会改变IP，所以设置静态IP。

1. 使用nm-tool这个命令查看ip 网关 dns



1. 编辑/etc/network/interfaces



1. 设置dns，编辑/etc/resolv.conf



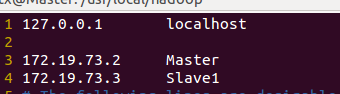
1. 别重启网卡了还是重启电脑最管用

### （2）更改主机名和hosts文件

$ sudo vim /etc/hostname

更改成Master 和 Slave1

$ sudo vim /etc/hosts



这样当你尝试去ping slave1节点就不能输那么长的IP了

$ ping Slave -c 3

### （3）ssh 无密码登录节点

操作原理：将master生成的公钥拷贝到所有节点上，加入授权。

$ cd ~/.ssh # 如果没有该目录，先执行一次ssh localhost

$ rm ./id\_rsa\* # 删除之前生成的公匙（如果有）

$ ssh-keygen -t rsa # 一直按回车就可以

无密码登录master本机：

$ cat ./id\_rsa.pub >> ./authorized\_keys

将公钥拷贝到slave1节点的xtx用户的家目录

$ scp ~/.ssh/id\_rsa.pub xtx@Slave1:/home/xtx/

Scp 是（secure copy）的缩写，用于linux下远程拷贝文件。

在Slave1节点上加入授权

$ mkdir ~/.ssh # 如果不存在该文件夹需先创建，若已存在则忽略

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

$ rm ~/id\_rsa.pub # 用完就可以删掉了

Master上 $ ssh Slave1 尝试。

### （4） 配置分布式环境

修改五个配置文件，更多设置请参考官方文档：

Core-site.xml:<http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-common/core-default.xml>

Hdfs-site.xml:<http://hadoop.apache.org/docs/r2.6.0/hadoop-project-dist/hadoop-hdfs/hdfs-default.xml>

Mapred.xml:<http://hadoop.apache.org/docs/r2.6.0/hadoop-mapreduce-client/hadoop-mapreduce-client-core/mapred-default.xml>

Yarn-site.xml:<http://hadoop.apache.org/docs/r2.6.0/hadoop-yarn/hadoop-yarn-common/yarn-default.xml>

1. 文件slaves 将作为datanode的主机名写入这里，所以我们这边写入Slave1
2. Core-site.xml

属性 fs.defaultFS 设置默认的文件系统，url指定主机名和namenode工作的rpc端口。

属性 hadoop.tmp.dir 设置hdfs的存放路径



1. Hdfs-site.xml

属性 dfs.namenode.name.dir 以逗号分隔的目录名称，namenode 存储永久性的元数据的目录列表，namenode在列表上的各个目录中均存放着相同的元数据。

属性 dfs.namenode.data.dir 以逗号分隔的目录名称 ，datanode存放数据块的目录，各个数据块分别存放在某一个目录下。



1. Mapred-site.xml

在YARN运行MapReduce是，mapred.xml 仍被用于记录通用MapReduce属性，只是与jobtracker，tasktracker相关的属性已经被废除了。



1. Yarn-site.xml

在yarn上用户不再运行jobtracker或者tasktracker。

Yarn的start-yarn.sh脚本会在脚本运行的机器上启动**资源管理器**，还在Slaves文件中列举的每台机器上启动一个**节点管理器**。

YARN 还有一个作业历史服务器和一个web应用程序代理服务器。前者可以显示历史作业的执行细节。后者可以使用户安全的访问YARN应用的用户界面。默认情况下，web应用程序代理服务器和资源管理器在同一个进程运行，但是也可以将其配置成各一个单独运行的守护进程。

属性 yarn.resourcemanager.hostname YARN的资源管理器

属性 yarn.nodemanager.aux-services 由于YARN是一个通用目的服务，MapReduce的shuffle句柄还需要被显式的启动，



其他配置：

1. 修改hadoop日志的生成位置，

因为默认是在hadoop安装目录下的log文件夹，但是在hadoop升级之后会有影响，在hadoop-env.sh中声明HADOOP\_LOG\_DIR=/var/log/hadoop

注意： 配置之后启动hadoop，会报错没有权限创建这个文件夹和更改文件的属主，所以要自己命令行创建这个文件夹并且更换属主。

1. 使用resync工具同步 master hadoop配置文件到 Slave hadoop上。

在hadoop1.中在hadoop-env.sh设置HADOOP\_MASTER项就行

在hadoop2.X中怎么做？？？？？？？

如何确保每个工作节点都在hadoop-env.sh中设置好HADOOP\_MASTER呢？

对于小型集群，编写一个脚本就行，对于大型集群，可以采用dsh之类的工具复制文件。

考虑到已启动远程同步功能的大型集群，在集群启动是，所以工作几乎同时产生，并向主节点发出rsync请求，很可能导致主节点瘫痪。为了防止这样的事情发生，需要将HADOOP\_SLAVE\_SLEEP 设置为一小段时间。

配置完成之后将master的hadoop文件夹拷贝到所有节点上，在此之前删除所有的临时文件：

$ cd /usr/local

$ sudo rm -r ./hadoop/tmp # 删除 Hadoop 临时文件

$ sudo rm -r ./hadoop/logs/\* # 删除日志文件

$ tar -zcf ~/hadoop.master.tar.gz ./hadoop # 先压缩再复制

$ cd ~

$ scp ./hadoop.master.tar.gz Slave1:/home/xtx # 复制到节点上

在Slave1节点上：

$ sudo rm -r /usr/local/hadoop # 删掉旧的（如果存在）

$ sudo tar -zxf ~/hadoop.master.tar.gz -C /usr/local

$ sudo chown -R xtx /usr/local/hadoop

### (5 ) 启动

启动前要在master节点对namenode格式化

$ hdfs namenode -format

在master启动hadoop

$ start-dfs.sh

$ start-yarn.sh

$ mr-jobhistory-daemon.sh start historyserver

通过命令 jps 可以查看各个节点所启动的进程。正确的话，**在 Master 节点**上可以看到 NameNode、ResourceManager、SecondrryNameNode、JobHistoryServer 进程

**在 Slave 节**点可以看到 DataNode 和 NodeManager 进程

在Master节点上使用 **hdfs dfsadmin -report** 查看datanode是否启动

### （6） 执行分布式实例

**执行分布式实例和执行伪分布式实例一样**

HDFS就要要创建用户目录

$ hdfs dfs -mkdir -p /user/xtx

目录名自己起就行。

将hadoop配置文件\*.xml都复制都input中

$ hdfs dfs -mkdir input

$ hdfs dfs -put ./etc/hadoop/\*.xml input

复制完成之后查看 input文件夹中内容：

$ hdfs dfs -ls input

运行MapReduce作业：

$ hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-\*.jar grep input output 'dfs[a-z.]+'

关闭hadoop集群不在赘述

和伪分布式一样可以，可以不启动yarn，记得改掉mapred.xml 文件名 ，改回之前的带template后缀。

## 使用命令行编译打包MapReduce程序

Hadoop2.x中版本的依赖的jar包，不再是集中到一个jar中，而分成了多个jar包；

实际上通过 hadoop classpath 命令，我们可以得到运行hadoop程序所需要所有的classpath信息。

1. 编译打包 hadoop MapReduce程序

将hadoop classpath信息加入环境变量 CLASSPATH中

export CLASSPATH=$($HADOOP\_HOME/bin/hadoop classpath):$CLASSPATH

1. 使用javac命令编译 WordCount.java
2. 将生成的class文件打包

jar -cvf WordCount.jar ./WordCount\*.class

1. 启动hadoop，（我启动的是分布式的）

所以使用的是hdfs上的input 和 output（记得使用前要删除它）

运行下面命令

hadoop jar WordCount.jar org.apache.hadoop.examples.WordCount input output

## 7.搭建eclipse环境

rm: Cannot delete /user/xtx/output. Name node is in safe mode.

$hdfs dfsadmin -safemode leave