

Hadoop安装部署



東北大學
Northeastern University



HORIZON
昊宸科技



环境组成与注意事项

本环境由两台虚拟机组成，一台为主节点，一台为数据节点。

IP	主机名	作用
192.168.88.90	master	主节点
192.168.88.91	datanode	数据节点

根据以上信息修改IP与计算机名

1、修改IP并关闭防火墙

```
service iptables stop
```

```
chkconfig iptables off
```

2、关闭Selinux

```
vi /etc/sysconfig/selinux
```

```
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUX can take one of these two values:
```

2、修改hostname

在/etc/sysconfig/network中修改hostname项，即执行
vi /etc/sysconfig/network命令进行修改，修改完后重启生效。

```
[root@localhost ~]# vi /etc/sysconfig/network

NETWORKING=yes
HOSTNAME=master
```

3、修改host文件

修改机器的/etc/hosts文件，将每个机器的hostname和ip对应，
将修改后/etc/hosts文件复制到集群中的每台机器，操作如下：

```
[root@localhost ~]# vi /etc/hosts

127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.88.90 master
192.168.88.91 datanode
```

4、执行scp /etc/hosts 192.168.88.91:/etc/

此命令用于将hosts文件分发至其他机器对应目录下。

5、全部分发完毕以后重启机器（reboot）

设置ssh无密码登录

(1) 输入命令, `ssh-keygen -t rsa`, 生成key, 都不输入密码, 一直回车, /root就会生成.ssh文件夹, 每台服务器都要设置,

(2) 合并公钥到authorized_keys文件, 在Master服务器, 进入/root/.ssh目录, 通过SSH命令合并,

```
cat id_rsa.pub >> authorized_keys
```

```
ssh 192.168.88.91 cat ~/.ssh/id_rsa.pub >> authorized_keys
```

```
scp ~/.ssh/authorized_keys 192.168.88.90:/root
```

```
cd /
```

```
cat authorized_keys >> ~/.ssh/authorized_keys
```

(3) 把Master服务器的authorized_keys、known_hosts复制到Slave服务器的/root/.ssh目录

(4) 完成, `ssh root@192.168.88.230`、`ssh root@192.168.88.231`就不需要输入密码了

JDK安装



本次安装使用的jdk版本为jdk-7u75-linux-x64版本。新建一个jdk的安装目录， /usr/local/下新建了Java目录，命令:mkdir /usr/local/java ，再把 jdk-7u75-linux-x64.tar.gz压缩包通过Xftp放入/usr/local/java文件夹内并通过tar命令解压

(1) 通过tar -vxf命令将jdk文件解压到java文件夹

```
-rw-r--r--. 1 root root 142245547 3月 20 10:58 jdk-7u75-linux-x64.tar.gz  
[root@master java]# tar -vxf jdk-7u75-linux-x64.tar.gz
```

(2) 配置环境变量

在"/etc/profile"文件的尾部添加以下内容:

```
# set java environment
```

```
export JAVA_HOME=/usr/local/java/jdk1.7.0_75
```

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

```
export PATH=$PATH:$JAVA_HOME/bin
```

通过source /etc/profile 使配置文件立即生效

(3) 判断是否安装成功

输入java -version 判断是否安装成功

注意: 每个部署节点都必须安装jdk

如虚拟机上有自带的OpenJdk必须卸载后安装 (通过yum -y remove [包名]的方式卸载 rpm -qa |grep jdk 查询)

Hadoop安装



➤ Standalone模式 (Local 模式)

这种模式，仅1个节点运行1个java进程，主要用于调试。

➤ Pseudo-Distributed模式

这种模式是，1个节点上运行，HDFS daemon的 NameNode 和 DataNode、YARN daemon的 ResourceManger 和 NodeManager，分别启动单独的java进程，主要用于调试。

➤ Cluster模式

完全分布式。

Hadoop安装



本文档使用的hadoop版本为2.7.1版本

1、只在Master服务器解压，再复制到Slave服务器

(1) 将“hadoop-2.7.1.tar.gz”，放到/usr/local/目录下，并将解压以后的文件夹改为/hadoop

(2) 在/usr/local/hadoop目录下创建数据存放的文件夹，tmp、hdfs、hdfs/data、hdfs/name

2、修改Hadoop 配置文件（/usr/local/hadoop/etc/Hadoop/）

1) hadoop-env.sh 文件

在文件末尾添加 `export JAVA_HOME=/usr/local/java/jdk1.7.0_75`

2) yarn-env.sh 文件

在文件末尾添加 `export JAVA_HOME=/usr/local/java/jdk1.7.0_75`

3、配置hadoop环境变量

在“/etc/profile”文件的尾部添加以下内容：

```
# set hadoop environment
```

```
export HADOOP_HOME=/usr/local/hadoop
```

```
:$HADOOP_HOME/bin:$HADOOP_HOME/sbin（在path后面加）
```

Hadoop1 : namenode/datanode/resourceManager/nodeManager

Hadoop2 : datanode/secondary namenode/nodeManager

Hadoop3 : datanode/nodeManager

3) core-site.xml 文件

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://192.168.88.90:9000</value>
  </property>
  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>file:/usr/local/hadoop/tmp</value>
    <description>Abase for other temporary directories.</description>
  </property>
  <property>
    <name>hadoop.proxyuser.root.hosts</name>
    <value>*</value>
  </property>
  <property>
    <name>hadoop.proxyuser.root.groups</name>
    <value>*</value>
  </property>
</configuration>
```

这里有一个地方需要注意，最后设置hadoop.proxyuser时，后面跟的是用户名，我是用root用户登录的。所以填的是root。

4) hdfs-site.xml 文件

```
<configuration>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/usr/local/hadoop/hdfs/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/usr/local/hadoop/hdfs/data</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>192.168.88.90:9001</value>
  </property>
  <property>
    <name>dfs.webhdfs.enabled</name>
    <value>true</value>
  </property>
</configuration>
```

5) mapred-site.xml 文件

```
<configuration>
```

```
  <property>
```

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

```
    <value>yarn</value>
```

```
  </property>
```

```
</configuration>
```

6) yarn-site.xml 文件

```
<configuration>
<!-- Site specific YARN configuration properties -->
<property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
</property>
<property>
    <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
<property>
    <name>yarn.resourcemanager.address</name>
    <value>192.168.88.90:8032</value>
</property>
<property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>192.168.88.90:8030</value>
</property>
<property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>192.168.88.90:8031</value>
</property>
<property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>192.168.88.230:8033</value>
</property>
<property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>192.168.88.230:8088</value>
</property>
</configuration>
```



7) slaves 文件

192.168.88.91

Hadoop启动与测试



先格式化namenode，然后启动hadoop。

```
bin/hdfs namenode -format
```

```
sbin/start-all.sh
```

或者使用

```
sbin/start-dfs.sh ; sbin/start-yarn.sh:
```

yarn监控界面: <http://192.168.88.90:8088>

hadoop监控界面: <http://192.168.88.90:50070>

```
[root@hdp1 ~]# jps
25645 ResourceManager
25225 NameNode
25473 SecondaryNameNode
25313 DataNode
27127 Jps
25737 NodeManager
```



http://192.168.88.93:50070

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

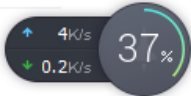
Datanode Information

In operation

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
hdp1:50010 (192.168.88.93:50010)	0	In Service	46.89 GB	28 KB	6.23 GB	40.66 GB	0	28 KB (0%)	0	2.7.3

Decomissioning

Node	Last contact	Under replicated blocks	Blocks with no live replicas	Under Replicated Blocks In files under construction
------	--------------	-------------------------	------------------------------	--



Hadoop, 2016.

Hadoop启动与测试



http://192.168.88.93:8088

Logged in as: dr.whc

All Applications

Cluster

[About](#)
[Nodes](#)
[Node Labels](#)
[Applications](#)
[NEW](#)
[NEW SAVING](#)
[SUBMITTED](#)
[ACCEPTED](#)
[RUNNING](#)
[FINISHED](#)
[FAILED](#)
[KILLED](#)
[Scheduler](#)

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCoresh Used	VCoresh Total	VCoresh Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>

Show 20 entries

Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes
No data available in table											

Showing 0 to 0 of 0 entries

First Previous Next Last

Tools

2.9K/s

12.9K/s

37%

Hadoop启动与测试



执行wordcount

1) 创建本地示例文件

首先在"/usr/local/hadoop"目录下创建文件夹"localfile"。

```
[root@master1 local]# cd hadoop/  
[root@master1 hadoop]# mkdir localfile
```

接着创建两个文本文件file1.txt和file2.txt, 使file1.txt内容为"Hello World", 而file2.txt的内容为"Hello Hadoop"。

2) 在HDFS上创建输入文件夹

```
[root@master1 localfile]# hadoop fs -mkdir input  
[root@master1 localfile]# hadoop fs -ls
```

3) 上传本地file中文件到集群的input目录下

```
[root@master1 localfile]# hadoop fs -put /usr/local/hadoop/localfile/file*.txt input
```

4) MapReduce执行过程显示信息

```
[root@master1 mapreduce]# hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount input output
```

5) 查看结果输出文件内容

```
[root@master1 mapreduce]# hadoop fs -ls output  
Found 2 items  
-rw-r--r--  3 root supergroup      0 2016-09-26 22:40 output/_SUCCESS  
-rw-r--r--  3 root supergroup    24 2016-09-26 22:40 output/part-r-00000
```


THANKS

