Hadoop 的安装与使用

1、 实验目的

1.1 熟练掌握 Hadoop 的安装及使用方法。

2、 实验环境

2.1 五台计算机，安装 Windows XP 操作系统。

2.2 Virtual PC 2007 虚拟机

2.3 Ubuntu 系统镜像 (ubuntu-10.04-desktop-i386.iso)

2.4 Java 安装包 (jdk-8u77-linux-i586.tar.gz)

2.5 Hadoop 安装包(Hadoop-2.7.2.tar.gz)

3、 实验准备

3.1 安装 Linux 虚拟机。

运行 Virtual PC 2007，单击“new”按钮，选择“Add an existing virtual machine”加载已
经建好的 ubuntu 系统。

3.2 安装 Java。

单击”CD“，选择”Capture ISO Image…“，选择并加载”software.iso“文件。

打开 ubuntu 系统的终端”Applications->Accessories->Terminal“。

拷贝 ISO 文件里的”jdk-8u77-linux-i586.tar.gz”: [cp /media/xxxxxxxx_xxxxxx/jdk-8u77-linux-

i586.tar.gz /home/bigdata/Downloads/] (注：方括号里面的内容为在终端中输入的命令，后
文与此相同，不再赘述。)

解压：[tar –zxvf jdk-8u77-linux-i586.tar.gz]

将 java 安装到目录 /usr/java/jdk1.8.0_77 ： [sudo mkdir /usr/java] [sudo cp –
r ./Downloads/jdk1.8.0_77 /usr/java]

设置环境变量： [sudo gedit /etc/profile]

在文件末尾加入：

set java environment

export JAVA_HOME=/usr/java/jdk1.8.0_77

export CLASSPATH=.:$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib

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

保存并退出编辑器，输入以下命令使配置生效：[source /etc/profile]

验证 java 安装成功：[javac -version] [java -version]

3.3 安装 ssh。

更新源列表：[sudo apt-get update]

安装 ssh：[sudo apt-get install openssh-server]

查看 ssh 是否启动：[sudo ps –e | grep ssh]，有 sshd，说明 ssh 服务已经启动

如果没有启动，输入：[sudo service ssh start]

安装远程数据同步工具 rsync，可通过 LAN/WAN 快速同步多台主机间的文件：[sudo apt-
get install rsync]

3.4 配置 Hadoop-env.sh 文件。

拷贝并解压”Hadoop-2.7.2.tar.gz”。

配置参数：[gedit ~/hadoop-2.7.2/etc/hadoop/hadoop-env.sh]

修改文件： export JAVA_HOME=/usr/java/jdk1.8.0_77，保存退出。

查看 hadoop 命令：[cd ~/hadoop-2.7.2] [bin/hadoop]

至此，我们准备好了 hadoop 的运行环境。

4、 实验步骤

Hadoop 集群支持三种运行模式：单机模式(Local (Standalone) Mode)，伪分布式模
式 (Pseudo-Distributed Mode) ， 完 全 分 布 式 模 式 (Fully-Distributed Mode)。

4.1 单机模式

默认情况下，Hadoop 被配置成一个以非分布式模式运行的独立 Java 进程，适合开始时
做调试工作。

下面运行的实例是，找出配置文件中符合正则表达式的字符

串，输出写入到指定的

output 目录。

[cd ~]

[mkdir input]

[cp ./hadoop-2.7.2/etc/hadoop/*.xml input]

[./hadoop-2.7.2/bin/hadoop jar ./hadoop-2.7.2/share/hadoop/mapreduce/hadoop-mapreduce-example-2.7.2.jar grep input output 'dfs[a-z.]+']

[cat output/*]

4.2 伪分布式模式

Hadoop 可以在单节点上以伪分布式模式运行，用不同的 Java 进程模拟分布式运行中各

类结点。

(1) Hadoop 配置

[gedit ./hadoop-2.7.2/etc/hadoop/core-site.xml]

[gedit ./hadoop-2.7.2/etc/hadoop/hdfs-site.xml]

(2) 免密码 SSH 设置

[ssh-keygen –t dsa –P ' ' –f ~/.ssh/id_dsa]

[cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys]

[chmod 0600 ~/.ssh/authorized_keys]

ssh 连接到本地：[ssh localhost]

断开连接：[exit]

(3) 运行 Hadoop

[cd ~/hadoop-2.7.2]

1) 格式化分布式文件系统。

[bin/hdfs namenode -format]

2) 启动 Hadoop 的 NameNode 和 DataNode 守护进程。

[sbin/start-dfs.sh]

3) 访问 http://localhost:50070/可以查看 NameNode 以及整个分布式文件系统的状

态，浏览分布式文件系统中的文件及日志等。

4) 给 MapReduce 任务创建 HDFS 目录。

[bin/hdfs dfs –mkdir /user]

[bin/hdfs dfs –mkdir /user/bigdata]

5) 将输入数据拷贝到分布式文件系统中。

[bin/hdfs dfs –put ~/input]

6) 运行正则表达式实例。

[bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-

example-2.7.2 jar

grep input output ‘dfs[a-z.]+’]

7) 检查运行结果。

从分布式文件系统中拷贝到本地文件系统来查看：

[bin/hdfs dfs –get output ~/outputFROMdfs]

[cat ~/outputFROMdfs]

或者直接查看分布式文件系统中的结果：

[bin/hdfs dfs –cat output/*]

8) 停止守护进程。

[sbin/stop-dfs.sh]

(附注：如果出现”Java HotSpot(TM) Client VM warning: ……disabled stack guard……

‘execstack –c <libfile>’……”的警告，可按如下方法进行设置)

[gedit ~/.bashrc] 在文件末尾添加，

使配置生效：[source ~/.bashrc]

4.3 完全分布式模式

(1) 集群包括 2 个节点：1 个 Master，1 个 Salve。

机器名称 IP 地址

master 10.13.30.229

slave01 10.13.30.231

(2) 网络配置

注意：此时，虚拟机的网卡选择为实体机上的网卡，而不是 NAT 等其它模式。

(Edit->Settings->Networking->Adapter 1)

在 master 机器中，

查看主机名称：[hostname]

修改主机名称：[sudo gedit /etc/hostname]

修改 IP 地址

配置 hosts 文件：[sudo gedit /etc/hosts]，添加：

10.13.30.229 master

10.13.30.231 slave01

保存并退出。重启系统使上述配置生效。

在 slave01 中，做同样的操作。

(3) SSH 配置

master 和 slave01 上都安装好 ssh 并生成密码对。

把 master 的公钥分发给 slave01 并让 slave01 授权，同样，slave01 的公钥给 master 并

让 master 授权。

master 上：[scp .ssh/id_dsa.pub slave01:~/.ssh/master.pub]

slave01 上：[cat ~/.ssh/master.pub >> ~/.ssh/authorized_keys]

把 slave01 的公钥分发不再赘述。

测试：[ssh slave01] [ssh master]

(4) 配置 Hadoop

创建数据存放的文件夹，hadoop-data/tmp、hadoop-data/hdfs 、hadoop-data/hdfs/data、hadoop-data/hdfs/name

[mkdir ~/hadoop-data]

[mkdir ~/hadoop-data/tmp]

以此类推，建好上述文件夹，并在 slave01 上也创建同样的文件夹。

修改/home/bigdata/hadoop-2.7.2/etc/hadoop 下的配置文件

修改 core-site.xml，加上

```
<configuration>
<property>
        <name>fs.defaultFS</name>
        <value>hdfs://master:9000</value>
    </property>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/home/bigdata/hadoop-data/tmp</value>
```

```
    </property>
    <property>
        <name>io.file.buffer.size</name>
        <value>131702</value>
  </property>
  </configuration>
```

修改 hdfs-site.xml，加上

```
  <configuration>
  <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/home/bigdata/hadoop-
data/hdfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/home/bigdata/hadoop-
data/hdfs/data</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>2</value>
```

```
        </property>
        <property>
            <name>dfs.namenode.secondary.http-
address</name>
            <value>master:9001</value>
        </property>
        <property>
  <name>dfs.webhdfs.enabled</name>
  <value>true</value>
        </property>
 </configuration>
```

修改 mapred-site.xml，加上

```
  <configuration>
  <property>
            <name>mapreduce.framework.name</name>
            <value>yarn</value>
        </property>
      <property>
          <name>mapreduce.jobhistory.address</name>
          <value>master:10020</value>
      </property>
      <property>
```

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

<value>master:19888</value>

</property>

</configuration>

修改 yarn-site.xml，加上

<configuration>

<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>master:8032</value>

</property>

```
        <property>

<name>yarn.resourcemanager.scheduler.address</name>
            <value>master:8030</value>
        </property>
        <property>
            <name>yarn.resourcemanager.resource-
tracker.address</name>
            <value>master:8031</value>
        </property>
        <property>
            <name>yarn.resourcemanager.admin.address</name>
            <value>master:8033</value>
        </property>
        <property>

<name>yarn.resourcemanager.webapp.address</name>
            <value>master:8088</value>
        </property>
        <property>
            <name>yarn.nodemanager.resource.memory-
mb</name>
```

<value>768</value>

</property>

配置/home/bigdata/hadoop-2.7.2/etc/hadoop 目录下 hadoop-env.sh、yarn-env.sh 的

JAVA_HOME,

Export JAVA_HOME=/usr/java/jdk1.8.0_77

配置/home/bigdata/hadoop-2.7.2/etc/hadoop 目录下 salves，这里只有一个 slave01。

然后把/home/bigdata/hadoop-2.7.2 复制到 slave01 上，

[scp –r ~/hadoop-2.7.2 slave01:~/]

(5) 启动 hadoop 集群

[hdfs namenode -format]

[start-dfs.sh]

[start-yarn.sh]

NameNode http://master:50070/

ResourceManager http://master:8088/

MapReduce JobHistory Server http://master:19888/

(6) 停止 hadoop 集群

[stop-dfs.sh]

[stop-yarn.sh]

(7) 基本的对分布式文件系统的操作

1.上传文件到 hdfs

首先为要上传的文件在 HDFS 上创建文件夹 hdfs dfs –mkdir [指定目录和文件夹名]

如：hdfs dfs –mkdir /test

把本地文件夹上传到 hdfs 上 hdfs dfs –put [本地文件] [新创建的 hdfs 上文件]

如：hdfs dfs –put ~/test/pagerankTest.in /test/

2.查看 hdfs 上的文件信息

hadoop fs –ls / (查看根目录下所有已经上传的文件)或者进入 http://localhost:50070，

Utilities->Browse the file system 中查看。查看具体内容如下：

3.从 hdfs 上下载文件到本地

从 HDFS 文件系统中下载文件到本地，有 2 种方法，一种可以直接进入 WebUI 下载，一种是

通过命令获取。

进入 http://localhost:50070 后， Utilities->Browse the file system 可以点击 name 下载。

在本地某个目录下创建一个新文件夹，用于接收从 hdfs 下载下来的文件，mkdir newfile

获取文件存放到本地指定路径 hadoop fs –get [hdfs 上的文件] [本地文件路径]

如：hadoop fs –get /test /home/sj/newfile

然后查看文件内容：hadoop fs –cat example.txt

4.删除 hdfs 上指定文件

hadoop fs –rm [指定文件]，如果是文件夹里包含多个子文件的话，可以使用 hadoop fs –

rm –r [指定文件]

再查看 hdfs 上文件是否该文件已不存在了，表示删除成功。

5.查看 HDFS 块的信息

使用命令 hdfs fsck [指定文件] –blosks

如：hdfs fsck /test –blocks