大数据课程软件统一说明

一、文档说明

为了统一我们的操作系统与软件环境，我们统一课前基本软件环境，实现全程学习当中的软件版本都是一致的

二、VmWare与linux版本

VmWare版本：

VmWare版本不做要求，使用VmWare10版本以上即可，关于VmWare的安装，直接使用安装包一直下一步安装即可，且安装包当中附带破解秘钥，进行破解即可使用

linux版本

linux统一使用centos

centos统一使用centos7.6 64位版本

种子文件下载地址：http://mirrors.aliyun.com/centos/7.6.1810/isos/x86_64/CentOS-7-x86_64-DVD-1810.torrent

三、使用VmWare来安装linux软件

参见视频操作说明

四、三台linux服务器环境准备

使用三台linux服务器，来做统一的环境准备，通过统一环境，实现所有授课环节环境统一

三台机器IP设置

参见视频设置

三台机器修改ip地址：

```
vi /etc/sysconfig/network-scripts/ifcfg-ens33

BOOTPROTO="static"
IPADDR=192.168.52.100
NETMASK=255.255.255.0
GATEWAY=192.168.52.1
DNS1=8.8.8.8
```

准备三台linux机器，IP地址分别设置成为

第一台机器IP地址：192.168.52.100

第二台机器IP地址：192.168.52.110

第三台机器IP地址：192.168.52.120

三台机器关闭防火墙

三台机器在root用户下执行以下命令关闭防火墙

```
systemctl stop firewalld
systemctl disable firewalld
```

三台机器关闭selinux

三台机器在root用户下执行以下命令关闭selinux

三台机器执行以下命令，关闭selinux

```
vim /etc/selinux/config

SELINUX=disabled
```

三台机器更改主机名

三台机器分别更改主机名

第一台主机名更改为：node01.kaikeba.com

第二台主机名更改为：node02.kaikeba.com

第三台主机名更改为：node03.kaikeba.com

第一台机器执行以下命令修改主机名

```
vim /etc/hostname
node01.kaikeba.com
```

第二台机器执行以下命令修改主机名

```
vim /etc/hostname
node02.kaikeba.com
```

第三台机器执行以下命令修改主机名

```
vim /etc/hostname
node03.kaikeba.com
```

三台机器更改主机名与IP地址映射

三台机器执行以下命令更改主机名与IP地址映射关系

```
vim /etc/hosts

192.168.52.100 node01.kaikeba.com node01
192.168.52.110 node02.kaikeba.com node02
192.168.52.120 node03.kaikeba.com node03
```

三台机器同步时间

三台机器执行以下命令定时同步阿里云服务器时间

```
 yum -y install ntpdate
 crontab -e
 */1 * * * * /usr/sbin/ntpdate time1.aliyun.com


```

三台机器添加普通用户

三台linux服务器统一添加普通用户hadoop，并给以sudo权限，用于以后所有的大数据软件的安装

并统一设置普通用户的密码为 123456

```
 useradd hadoop
 passwd hadoop
```

三台机器为普通用户添加sudo权限

```
visudo

hadoop  ALL=(ALL)       ALL
```

三台定义统一目录

定义三台linux服务器软件压缩包存放目录，以及解压后安装目录，三台机器执行以下命令，创建两个文件夹，一个用于存放软件压缩包目录，一个用于存放解压后目录

```
 mkdir -p /kkb/soft     # 软件压缩包存放目录
 mkdir -p /kkb/install  # 软件解压后存放目录
 chown -R hadoop:hadoop /kkb    # 将文件夹权限更改为hadoop用户
```

三台机器安装jdk

使用hadoop用户来重新连接三台机器，然后使用hadoop用户来安装jdk软件

上传压缩包到第一台服务器的/kkb/soft下面，然后进行解压，配置环境变量即可，三台机器都依次安装即可

```
cd /kkb/soft/

tar -zxf jdk-8u141-linux-x64.tar.gz  -C /kkb/install/
sudo vim /etc/profile


#添加以下配置内容，配置jdk环境变量
export JAVA_HOME=/kkb/install/jdk1.8.0_141
export PATH=:$JAVA_HOME/bin:$PATH
```

hadoop用户免密码登录

三台机器在hadoop用户下执行以下命令生成公钥与私钥

```
ssh-keygen -t rsa
三台机器在hadoop用户下，执行以下命令将公钥拷贝到node01服务器上面去
ssh-copy-id  node01
node01在hadoop用户下，执行以下命令，将authorized_keys拷贝到node02与node03服务器
cd /home/hadoop/.ssh/
scp authorized_keys  node02:$PWD
scp authorized_keys  node03:$PWD
```

三台机器关机重启

三台机器在root用户下执行以下命令，实现关机重启

```
reoot -h now
```

五、三台机器安装zookeeper集群

注意事项：三台机器一定要保证时钟同步

第一步：下载zookeeeper的压缩包，下载网址如下

http://archive.cloudera.com/cdh5/cdh/5/

我们在这个网址下载我们使用的zk版本为zookeeper-3.4.5-cdh5.14.2.tar.gz

下载完成之后，上传到我们的node01的/kkb/soft路径下准备进行安装

第二步：解压

node01执行以下命令解压zookeeper的压缩包到node01服务器的/kkb/install路径下去，然后准备进行安装

```
cd /kkb/soft

tar -zxvf zookeeper-3.4.5-cdh5.14.2.tar.gz  -C /kkb/install/
```

第三步：修改配置文件

第一台机器修改配置文件

```
cd /kkb/install/zookeeper-3.4.5-cdh5.14.2/conf

cp zoo_sample.cfg zoo.cfg

mkdir -p /kkb/install/zookeeper-3.4.5-cdh5.14.2/zkdatas

vim  zoo.cfg

dataDir=/kkb/install/zookeeper-3.4.5-cdh5.14.2/zkdatas

autopurge.snapRetainCount=3

autopurge.purgeInterval=1

server.1=node01:2888:3888
server.2=node02:2888:3888
server.3=node03:2888:3888
```

第四步：添加myid配置

在第一台机器的/kkb/install/zookeeper-3.4.5-cdh5.14.2/zkdatas/

这个路径下创建一个文件，文件名为myid ,文件内容为1

```
echo 1 > /kkb/install/zookeeper-3.4.5-cdh5.14.2/zkdatas/myid
```

第五步：安装包分发并修改myid的值

安装包分发到其他机器

```
第一台机器上面执行以下两个命令

scp -r /kkb/install/zookeeper-3.4.5-cdh5.14.2/ node02:/kkb/install/

scp -r /kkb/install/zookeeper-3.4.5-cdh5.14.2/ node03:/kkb/install/

第二台机器上修改myid的值为2

直接在第二台机器任意路径执行以下命令

echo 2 > /kkb/install/zookeeper-3.4.5-cdh5.14.2/myid



第三台机器上修改myid的值为3
```

```
16
17  直接在第三台机器任意路径执行以下命令
18
19  echo 3 > /kkb/install/zookeeper-3.4.5-cdh5.14.2/myid
```

第六步：三台机器启动zookeeper服务

三台机器启动zookeeper服务

这个命令三台机器都要执行

```
1  /kkb/install/zookeeper-3.4.5-cdh5.14.2/bin/zkServer.sh start
2
3  查看启动状态
4
5  /kkb/install/zookeeper-3.4.5-cdh5.14.2/bin/zkServer.sh status
```

六，hadoop环境安装

1、CDH软甲版本重新进行编译

1、为什么要编译hadoop

由于CDH的所有安装包版本都给出了对应的软件版本，一般情况下是不需要自己进行编译的，但是由于cdh给出的hadoop的安装包没有提供带C程序访问的接口，所以我们在使用本地库（本地库可以用来做压缩，以及支持C程序等等）的时候就会出问题，好了废话不多说，接下来看如何编译

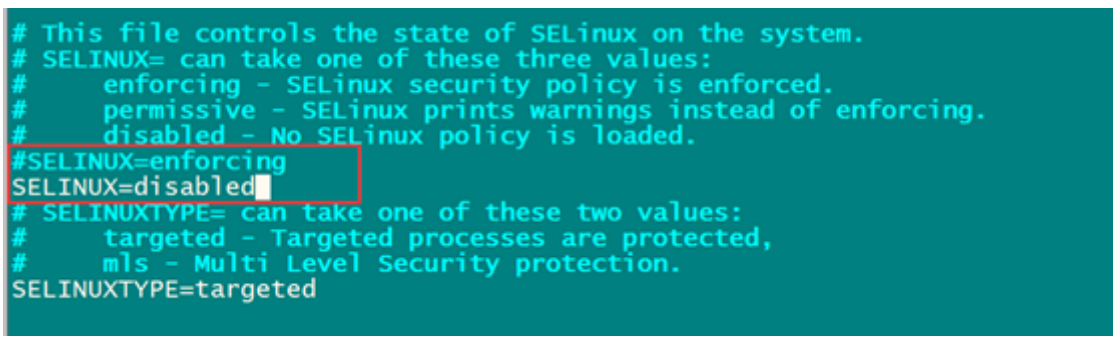
2、编译环境的准备

2.1：准备linux环境

准备一台linux环境，内存4G或以上，硬盘40G或以上，我这里使用的是Centos6.9 64位的操作系统（注意：一定要使用64位的操作系统）

2.2：虚拟机联网，关闭防火墙，关闭selinux

```
1  关闭防火墙命令：
2
3  service  iptables   stop
4  chkconfig   iptables  off
5
6  关闭selinux
7  vim /etc/selinux/config
8
```



2.3：安装jdk1.7

注意：亲测证明hadoop-2.6.0-cdh5.14.2 这个版本的编译，只能使用jdk1.7，如果使用jdk1.8那么就会报错

注意：这里一定不要使用jdk1.8，亲测jdk1.8会出现错误

将我们jdk的安装包上传到/kkb/soft（我这里使用的是jdk1.7.0_71这个版本）

解压我们的jdk压缩包

统一两个路径

```
mkdir -p /kkb/soft
mkdir -p /kkb/install
cd /kkb/soft
tar -zxvf jdk-7u71-linux-x64.tar.gz -C ../servers/
```

配置环境变量

```
vim /etc/profile

export JAVA_HOME=/kkb/install/jdk1.7.0_71

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

```
让修改立即生效

source /etc/profile
```

2.4：安装maven

这里使用maven3.x以上的版本应该都可以，不建议使用太高的版本，强烈建议使用3.0.5的版本即可

将maven的安装包上传到/kkb/soft

然后解压maven的安装包到/kkb/install

```
cd /kkb/soft/

tar -zxvf apache-maven-3.0.5-bin.tar.gz -C ../servers/
```

配置maven的环境变量

```
vim /etc/profile

export MAVEN_HOME=/kkb/install/apache-maven-3.0.5

export MAVEN_OPTS="-Xms4096m -Xmx4096m"

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

```
让修改立即生效

source /etc/profile
```

2.5：安装findbugs

下载findbugs

```
cd  /kkb/soft

wget --no-check-certificate https://sourceforge.net/projects/findbugs/files/findbugs/1.3.9/findbugs-1.3.9.tar.gz/download -O findbugs-1.3.9.tar.gz



解压findbugs

tar -zxvf findbugs-1.3.9.tar.gz -C ../install/



配置findbugs的环境变量

vim /etc/profile

export JAVA_HOME=/kkb/install/jdk1.7.0_75

export PATH=:$JAVA_HOME/bin:$PATH



export MAVEN_HOME=/kkb/install/apache-maven-3.0.5

export PATH=:$MAVEN_HOME/bin:$PATH


export FINDBUGS_HOME=/kkb/install/findbugs-1.3.9
export PATH=:$FINDBUGS_HOME/bin:$PATH
```

```
让修改立即生效

 source  /etc/profile
```

2.6：在线安装一些依赖包

```
yum install autoconf automake libtool cmake

yum install ncurses-devel

yum install openssl-devel

yum install lzo-devel zlib-devel gcc gcc-c++



bzip2压缩需要的依赖包

yum install -y  bzip2-devel
```

2.7：安装protobuf

protobuf下载百度网盘地址

https://pan.baidu.com/s/1pJlZubT

下载之后上传到 /kkb/soft

解压protobuf并进行编译

```
cd  /kkb/soft

tar -zxvf protobuf-2.5.0.tar.gz -C ../servers/

cd   /kkb/install/protobuf-2.5.0

./configure

make && make install
```

2.8、安装snappy

snappy下载地址：

http://code.google.com/p/snappy/

```
cd /kkb/soft/

tar -zxf snappy-1.1.1.tar.gz  -C ../servers/

cd ../servers/snappy-1.1.1/

./configure

make && make install
```

2.9：下载cdh源码准备编译

源码下载地址为：

http://archive.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-cdh5.14.2-src.tar.gz

下载源码进行编译

```
cd  /kkb/soft

wget http://archive.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-cdh5.14.2-src.tar.gz

tar -zxvf hadoop-2.6.0-cdh5.14.2-src.tar.gz -C ../servers/

cd  /kkb/install/hadoop-2.6.0-cdh5.14.2

编译不支持snappy压缩：

mvn package -Pdist,native -DskipTests -Dtar



编译支持snappy压缩：

mvn package -DskipTests -Pdist,native -Dtar -Drequire.snappy -e -X

编译完成之后我们需要的压缩包就在下面这个路径里面
```

2.10：常见编译错误

如果编译时候出现这个错误：

An Ant BuildException has occured: exec returned: 2

这是因为tomcat的压缩包没有下载完成，需要自己下载一个对应版本的apache-tomcat-6.0.53.tar.gz的压缩包放到指定路径下面去即可

这两个路径下面需要放上这个tomcat的 压缩包

```
/kkb/install/hadoop-2.6.0-cdh5.14.2/hadoop-hdfs-project/hadoop-hdfs-httpfs/downloads

/kkb/install/hadoop-2.6.0-cdh5.14.2/hadoop-common-project/hadoop-kms/downloads
```

2、hadoop集群的安装

安装环境服务部署规划

服务器IP	192.168.52.100	192.168.52.110	192.168.52.120
HDFS	NameNode		
HDFS	SecondaryNameNode		
HDFS	DataNode	DataNode	DataNode
YARN	ResourceManager		
YARN	NodeManager	NodeManager	NodeManager

YARN	NodeManager	NodeManager	NodeManager
服务器IP 历史日志服务器	**192.168.52.100** JobHistoryServer	**192.168.52.110**	**192.168.52.120**

第一步：上传压缩包并解压

将我们重新编译之后支持snappy压缩的hadoop包上传到第一台服务器并解压

第一台机器执行以下命令

```
cd /kkb/soft/



tar -zxvf hadoop-2.6.0-cdh5.14.2_after_compile.tar.gz -C ../install/
```

第二步：查看hadoop支持的压缩方式以及本地库

第一台机器执行以下命令

```
cd /kkb/install/hadoop-2.6.0-cdh5.14.2

bin/hadoop checknative
```

如果出现openssl为false，那么所有机器在线安装openssl即可，执行以下命令，虚拟机联网之后就可以在线进行安装了

```
yum -y install openssl-devel
```

第三步：修改配置文件

修改core-site.xml

第一台机器执行以下命令

```
cd /kkb/install/hadoop-2.6.0-cdh5.14.2/etc/hadoop

vim core-site.xml

<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://node01:8020</value>
    </property>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/kkb/install/hadoop-2.6.0-
cdh5.14.2/hadoopDatas/tempDatas</value>
    </property>
    <!--  缓冲区大小，实际工作中根据服务器性能动态调整 -->
    <property>
        <name>io.file.buffer.size</name>
        <value>4096</value>
    </property>
    <!--  开启hdfs的垃圾桶机制，删除掉的数据可以从垃圾桶中回收，单位分钟 -->
    <property>
```

```
        <name>fs.trash.interval</name>
        <value>10080</value>
    </property>
</configuration>
```

修改hdfs-site.xml

第一台机器执行以下命令

```
cd /kkb/install/hadoop-2.6.0-cdh5.14.2/etc/hadoop

vim hdfs-site.xml

<configuration>
    <!-- NameNode存储元数据信息的路径，实际工作中，一般先确定磁盘的挂载目录，然后多个目录用，进行分割    -->
    <!--   集群动态上下线
    <property>
        <name>dfs.hosts</name>
        <value>/kkb/install/hadoop-2.6.0-cdh5.14.2/etc/hadoop/accept_host</value>
    </property>
    <property>
        <name>dfs.hosts.exclude</name>
        <value>/kkb/install/hadoop-2.6.0-cdh5.14.2/etc/hadoop/deny_host</value>
    </property>
     -->
     <property>
            <name>dfs.namenode.secondary.http-address</name>
            <value>node01:50090</value>
    </property>
    <property>
        <name>dfs.namenode.http-address</name>
        <value>node01:50070</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:///kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/namenodeDatas</value>
    </property>
    <!--  定义dataNode数据存储的节点位置，实际工作中，一般先确定磁盘的挂载目录，然后多个目录用，进行分割  -->
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:///kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/datanodeDatas</value>
    </property>
    <property>
        <name>dfs.namenode.edits.dir</name>
        <value>file:///kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/dfs/nn/edits</value>
    </property>
    <property>
        <name>dfs.namenode.checkpoint.dir</name>
        <value>file:///kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/dfs/snn/name</value>
```

```
    </property>
    <property>
        <name>dfs.namenode.checkpoint.edits.dir</name>
        <value>file:///kkb/install/hadoop-2.6.0-
cdh5.14.2/hadoopDatas/dfs/nn/snn/edits</value>
    </property>
    <property>
        <name>dfs.replication</name>
        <value>2</value>
    </property>
    <property>
        <name>dfs.permissions</name>
        <value>false</value>
    </property>
<property>
        <name>dfs.blocksize</name>
        <value>134217728</value>
    </property>
</configuration>

```

修改hadoop-env.sh

第一台机器执行以下命令

```
cd /kkb/install/hadoop-2.6.0-cdh5.14.2/etc/hadoop

vim hadoop-env.sh
export JAVA_HOME=/kkb/install/jdk1.8.0_141

```

修改mapred-site.xml

第一台机器执行以下命令

```
cd /kkb/install/hadoop-2.6.0-cdh5.14.2/etc/hadoop

vim mapred-site.xml

<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
    <property>
        <name>mapreduce.job.ubertask.enable</name>
        <value>true</value>
    </property>
    <property>
        <name>mapreduce.jobhistory.address</name>
        <value>node01:10020</value>
    </property>
    <property>
        <name>mapreduce.jobhistory.webapp.address</name>
        <value>node01:19888</value>
```

```
    </property>
</configuration>

```

修改yarn-site.xml

第一台机器执行以下命令

```
cd /kkb/install/hadoop-2.6.0-cdh5.14.2/etc/hadoop

vim yarn-site.xml

<configuration>
    <property>
       <name>yarn.resourcemanager.hostname</name>
        <value>node01</value>
    </property>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>

```

修改slaves文件

第一台机器执行以下命令

```
cd /kkb/install/hadoop-2.6.0-cdh5.14.2/etc/hadoop

vim slaves
node02
node03

```

第四步：创建文件存放目录

第一台机器执行以下命令

node01机器上面创建以下目录

```
mkdir -p /kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/tempDatas
mkdir -p /kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/namenodeDatas
mkdir -p /kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/datanodeDatas
mkdir -p /kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/dfs/nn/edits
mkdir -p /kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/dfs/snn/name
mkdir -p /kkb/install/hadoop-2.6.0-cdh5.14.2/hadoopDatas/dfs/nn/snn/edits
```

第五步：安装包的分发

第一台机器执行以下命令

```
cd /kkb/install/

scp -r hadoop-2.6.0-cdh5.14.2/ node02:$PWD
scp -r hadoop-2.6.0-cdh5.14.2/ node03:$PWD

```

第六步：配置hadoop的环境变量

三台机器都要进行配置hadoop的环境变量

三台机器执行以下命令

```
vim  /etc/profile

export HADOOP_HOME=/kkb/install/hadoop-2.6.0-cdh5.14.2
export PATH=:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH

配置完成之后生效

source /etc/profile

```

第七步：集群启动

要启动 Hadoop 集群，需要启动 HDFS 和 YARN 两个集群。

注意：首次启动HDFS时，必须对其进行格式化操作。本质上是一些清理和准备工作，因为此时的 HDFS 在物理上还是不存在的。

```
bin/hdfs namenode  -format或者bin/hadoop namenode –format

```

单个节点逐一启动

```
在主节点上使用以下命令启动 HDFS NameNode：
hadoop-daemon.sh start namenode

在每个从节点上使用以下命令启动 HDFS DataNode：
hadoop-daemon.sh start datanode

在主节点上使用以下命令启动 YARN ResourceManager：
yarn-daemon.sh  start resourcemanager

在每个从节点上使用以下命令启动 YARN nodemanager：
yarn-daemon.sh start nodemanager

以上脚本位于$HADOOP_PREFIX/sbin/目录下。如果想要停止某个节点上某个角色，只需要把命令中的start 改为stop 即可。

```

脚本一键启动

如果配置了 etc/hadoop/slaves 和 ssh 免密登录，则可以使用程序脚本启动所有Hadoop 两个集群的相关进程，在主节点所设定的机器上执行。

启动集群

node01节点上执行以下命令

```
第一台机器执行以下命令

cd /kkb/install/hadoop-2.6.0-cdh5.14.2/
sbin/start-dfs.sh
sbin/start-yarn.sh
sbin/mr-jobhistory-daemon.sh start historyserver

停止集群：

sbin/stop-dfs.sh

sbin/stop-yarn.sh

```

第八步：浏览器查看启动页面

hdfs集群访问地址

http://192.168.52.100:50070/dfshealth.html#tab-overview

yarn集群访问地址

http://192.168.52.100:8088/cluster

jobhistory访问地址：

http://192.168.52.100:19888/jobhistory