哈尔滨工业大学（深圳）

大数据实验指导书

实验一 Hadoop 环境配置与基本操作

1. 实验目的

1. 熟悉 Hadoop 单机环境和伪分布式环境的配置方法；
2. 熟悉命令行运行 Mapreduce 作业的原理和操作。

2. 实验内容

1. 根据实验指导书，独立搭建 Hadoop 单机环境和伪分布式环境；
2. 使用 MapReduce 实现 wordCount 任务，实验步骤需截图。

3. 实验环境

1. Ubuntu 16.04
2. Hadoop 2.7.2

4. 实验步骤

4.1 创建 Hadoop 用户

1. 创建 hadoop 用户并使用 bash 作为 shell

```
sudo useradd -m hadoop -s /bin/bash
```

2. 设置 hadoop 用户密码

```
sudo passwd hadoop
```

3. 为 hadoop 用户增加管理员权限

```
sudo adduser hadoop sudo
```

4.2 安装 SSH、配置 ssh 免密登录

1. 切换至 hadoop 用户（务必切换到 hadoop 用户再执行下面的步骤）

```
su - hadoop
```

2. 更新 apt

```
sudo apt-get update
```

3. 安装 ssh 服务器

```
sudo apt-get install openssh-server
```

4. 删除原有的.ssh 文件夹（若不存在无需删除）

```
rm -r ~/.ssh
```

5. 利用 ssh-keygen 生成密钥

```
ssh-keygen -t rsa
```

6. 将密钥加入到授权中

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

7. 测试 ssh 无密码连接

```
ssh localhost
```

4.3 安装 Java 环境

1. 安装 OpenJDK 8

```
sudo apt-get install openjdk-8-jre openjdk-8-jdk -y
```

2. 设置 JAVA_HOME

```
sudo vim /etc/profile
```

在文件最后面添加如下一行代码（**=号前后没有空格**）：

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

使环境变量生效：

```
source  /etc/profile
```

检查是否生效：

```
echo $JAVA_HOME
```

4.4 配置 Hadoop 单机环境

1. 解压 hadoop 安装包到/usr/local 文件夹下并重命名

```
sudo tar -xzvf hadoop-2.7.2.tar.gz -C /usr/local
sudo mv /usr/local/hadoop-2.7.2 /usr/local/Hadoop
sudo chown  -R hadoop:hadoop /usr/local/Hadoop
```

使用 tar 命令解压时，需指定 hadoop 安装包的绝对路径，如在桌面上：/home/lenovo/桌面/hadoop-2.7.2.tar.gz（可通过右键查看文件属性得到该路

径）。
注意：请务必使用 chown 命令修改解压后的 hadoop 目录的权限。

2.配置 Hadoop 环境变量

```
sudo vim /etc/profile
```

在文件最后面添加如下两行代码（$PATH 不要写成 SPATH）：

```
export HADOOP_HOME=/usr/local/Hadoop
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

使环境变量生效：

```
source  /etc/profile
```

3.配置 Hadoop 中的 JAVA_HOME

```
vim /usr/local/Hadoop/etc/hadoop/hadoop-env.sh
```

找到其中的 JAVA_HOME 配置,将其改为之前配置的 JAVA_HOME 对应的路径:
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

4.运行 Hadoop 单机环境（非分布式）
a)设置输入的路径 input，在此进行统计的普通文本文件，注意修改目录权限。

```
sudo mkdir -p /usr/test/hadoop
sudo chown  -R hadoop:hadoop /usr/test/hadoop
mkdir -p /usr/test/hadoop/input
```

b)将需要做词频统计的文件拷贝到 input 路径下。以下示例为将 hadoop 目录下的 README.txt 文件做词频统计：

```
cp /usr/local/Hadoop/README.txt /usr/test/hadoop/input/
```

请大家提前对爬取到的中文文本用 jieba 做分词处理（空格隔开），包括去除停用词（停用词表可在网上下载）。将预处理后的文本拷贝到 input 路径下。

c)使用 Hadoop 命令进行统计测试（不需要创建 output 目录）,以下为一条命令：

```
hadoop jar
/usr/local/Hadoop/share/hadoop/mapreduce/hadoop-mapreduce-
examples-2.7.2.jar wordcount /usr/test/hadoop/input
/usr/test/hadoop/output
```

d)查看输出目录

```
cat /usr/test/hadoop/output/part-r-00000
```

注意：Hadoop 默认不会覆盖结果文件，因此再次运行上面的命令会提示出错，需要先将/usr/test/hadoop/output 目录删除后再运行。

4.5 配置 Hadoop 伪分布式环境

1. 修改 core-site.xml 的配置文件（Hadoop 在运行时候的核心文件）

```
vim /usr/local/Hadoop/etc/hadoop/core-site.xml
```

添加：

```
<configuration>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>file:/usr/local/Hadoop/tmp</value>
    </property>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

2. 修改 hdfs-site.xml 的配置文件 （进行 HDFS 分布式储存的配置）

```
vim /usr/local/Hadoop/etc/hadoop/hdfs-site.xml
```

添加：

```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.namenode.name.dir</name>
        <value>file:/usr/local/Hadoop/tmp/dfs/name</value>
    </property>
    <property>
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/Hadoop/tmp/dfs/data</value>
    </property>
</configuration>
```

3.配置完成后，执行 NameNode 的格式化

```
/usr/local/Hadoop/bin/hdfs namenode -format
```

4.开启 NameNode 和 DataNode 守护进程，开启后通过 jps 命令查看

```
/usr/local/Hadoop/sbin/start-dfs.sh
jps
```

5.运行 Hadoop 伪分布式环境

a)在 HDFS 中创建用户目录

```
/usr/local/Hadoop/bin/hdfs dfs -mkdir -p /user/hadoop
```

b)设置输入的路径

```
/usr/local/Hadoop/bin/hdfs dfs -mkdir input
```

c) 将需要做词频统计的文件拷贝到 input 路径下

```
/usr/local/Hadoop/bin/hdfs dfs -put README.txt input
```

请大家提前对爬取到的中文文本用 jieba 做分词处理（空格隔开），包括去除停用词（停用词表可在网上下载）。将预处理后的文本拷贝到 input 路径下。

d)使用 Hadoop 命令进行统计测试，以下为一条命令：

```
hadoop jar /usr/local/Hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.2.jar wordcount input output
```

e)查看运行结果的命令（查看的是位于 HDFS 中的输出结果）

```
/usr/local/Hadoop/bin/hdfs dfs -cat output/part-r-00000
```

f）也可以将运行结果取回到本地

```
/usr/local/Hadoop/bin/hdfs dfs -get output ./output
cat /usr/local/Hadoop/output/part-r-00000
```

注意：Hadoop 运行程序时，输出目录不能存在，因此若要再次执行，需要执行如下命令删除 output 文件夹:

```
/usr/local/Hadoop/bin/hdfs dfs -rm -r output
```

6.启动 YARN

YARN 是从 MapReduce 中分离出来的，负责资源管理与任务调度。YARN 运行于 MapReduce 之上，提供了高可用性、高扩展性，伪分布式不启动 YARN 也可以，一般不会影响程序执行。

a) 修改配置文件 yarn-site.xml（进行 yarn 分析结构使用）

```
vim /usr/local/Hadoop/etc/hadoop/yarn-site.xml
```

添加：

```
<configuration>
    <property>
            <name>yarn.nodemanager.aux-services</name>
            <value>mapreduce_shuffle</value>
    </property>
</configuration>
```

b)修改配置文件 mapred-site.xml

```
cd /usr/local/Hadoop
mv ./etc/hadoop/mapred-site.xml.template ./etc/hadoop/mapred-site.xml
vim ./etc/hadoop/mapred-site.xml
```

添加：

```
<configuration>
    <property>
            <name>mapreduce.framework.name</name>
            <value>yarn</value>
        </property>
</configuration>
```

c)启动 YARN （需要先执行过./sbin/start-dfs.sh），开启后通过 jps 命令查看。

```
/usr/local/Hadoop/sbin/start-yarn.sh
jps
```

d)关闭 YARN

```
/usr/local/Hadoop/sbin/stop-yarn.sh
```