（二）. Hadoop 练习

1、实验目的

搭建 Hadoop 运行环境，了解其基本操作。

2、实验内容

（1）搭建 Hadoop

在个人电脑上搭建 Hadoop，操作系统 Linux/Windows 都可以，可使用虚拟机，单节点（如果时间充裕，可以搭建多节点）。

参考：

http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

https://wiki.apache.org/hadoop/Hadoop2OnWindows

网上还有很多中文资料，可自行搜索。

（2）运行第一个 Hadoop 实例

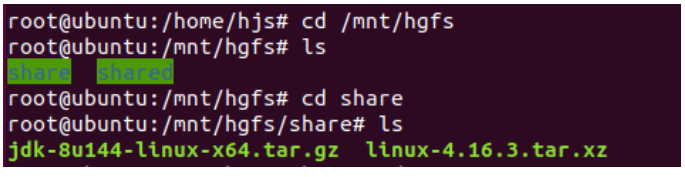
Wordcount 是 Hadoop 中的 HelloWorld 的程序。尝试正确运行 Wordcount。

3、实验报告（Linux 环境下搭建 Hadoop）

(1)实验环境：
虚拟机环境：VMware Workstation Pro
Linux 内核版本:4.18.0-17-generic
Ubuntu 版本：18.04.2 LTS

(2)配置 jdk
我的电脑里存有之前下载过的 linux 版本的 jdk，所以我可以将这个 jdk 存放到主机和虚拟机的共享文件夹里。然后再将这个 jdk 移动到我想存放目录上即可。

cd /mnt/hgfs/share #移动到我的共享文件夹
ls #查看 jdk 安装包是否已经存在在共享文件夹里



cd /usr/local
mkdir java #创建 java 文件夹目录
cd java #进入 java 文件夹
mv /mnt/hgfs/share/jdk-8u144-linux-x64.tar.gz /usr/local/java/ #移动 jdk 到 java 文件夹下
tar zxvf jdk-8u144-linux-x64.tar.gz #解压 jdk 压缩包
ls #查看解压后的当前目录如下，可见解压后多了一个 jdk1.8.0_144 文件夹，证明解压成功

```
root@ubuntu:/usr/local/java# ls
jdk1.8.0_144  jdk-8u144-linux-x64.tar.gz
```

vim /root/.bashrc #配置 jdk 全局环境变量

在打开的文件末尾加入如下代码后，保存并退出：

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_144
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_144
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

source ~/.bashrc #激活配置文件
java -version #在终端检测是否配置成功

```
root@ubuntu:/usr/local/java# java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
```

(3)配置 SSH
apt-get install ssh #安装 ssh

ssh-keygen -t rsa #首先生成密钥对，然后在接下来的设置中不断按 enter，将生成的密钥对保存在.ssh/id_rsa 文件中

cd ~/.ssh #检验是否生成.ssh 目录，并切换到该目录下，如果密钥对生成失败，是没有该文件夹的

```
root@ubuntu:/# cd /root/.ssh
root@ubuntu:~/.ssh# ls
id_rsa  id_rsa.pub
```

cp id_rsa.pub authorized_keys #把密钥追加到授权的 key 里面去

```
root@ubuntu:~/.ssh# cp id_rsa.pub authorized_keys
root@ubuntu:~/.ssh# cd ~/.ssh
root@ubuntu:~/.ssh# ls
authorized_keys  id_rsa  id_rsa.pub
```

ssh localhost #无密码连接 ubuntu 主机，出现以下提示说明连接成功

```
root@ubuntu:~# ssh localhost
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-17-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

139 packages can be updated.
0 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Mon Apr 22 12:05:36 2019 from 127.0.0.1
```

(4)安装和配置 Hadoop

mkdir ~/hadoop #在当前用户目录下创建 hadoop 文件夹用于存放 hadoop

同样在该目录下下载 hadoop-3.1.0.tar.gz

tar xzvfhadoop-3.1.0.tar.gz #解压得到如下文件夹 hadoop-3.1.0

```
root@ubuntu:~/hadoop# ls
hadoop-3.1.0  hadoop-3.1.0.tar.gz
```

cd hadoop-3.1.0/etc/hadoop #进入这个文件夹配置环境

vi hadoop-env.sh #编辑配置文件

exportJAVA_HOME=/usr/local/java/jdk1.8.0_144

exportHADOOP_HOME=/root/hadoop/hadoop-3.1.0

exportPATH=$PATH:/root/hadoop/hadoop-3.1.0/bin

```
# It uses the format of (command)_(subcommand)_USER.
#
# For example, to limit who can execute the namenode command,
# export HDFS_NAMENODE_USER=hdfs
export JAVA_HOME=/usr/local/java/jdk1.8.0_144
export HADOOP_HOME=/root/hadoop/hadoop-3.1.0
export PATH=$PATH:/root/hadoop/hadoop-3.1.0/bin
"hadoop-env.sh" 425L, 16520C
```

vi core-site.xml #配置 core-site.xml

在文件末尾加入如下配置代码：

```
<configuration>
   <!-- 指定 HDFS 老大（namenode）的通信地址 -->
      <property>
              <name>fs.default.name</name>
              <value>hdfs://localhost:9000</value>
      </property>
   <!-- 指定 hadoop 运行时产生文件的存储路径 -->
      <property>
              <name>hadoop.tmp.dir</name>
        <value>/root/hadoop/hadoop-3.1.0/hadoop_tmp</value>
        <description>A base for other
temporarydirectories.</description>
      </property>
</configuration>
```

vi mapred-site.xml #配置 mapred-site.xml

在文件末尾加入以下几行配置代码

```
<configuration>
   <property>
      <name>mapred.job.tracker</name>
      <value>localhost:9001</value>
   </property>
</configuration>
```

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
       <name>mapred.job.tracker</name>
       <value>localhost:9001</value>
    </property>
</configuration>
```

vi hdfs-site.xml #配置 hdfs-site.xml
在文件末尾加入几行配置代码

```
<configuration>
   <property>
      <name>dfs.replication</name>
      <value>1</value>
   </property>
</configuration>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
       <name>dfs.replication</name>
       <value>1</value>
    </property>
</configuration>
```

LibreOffice Writer

vi /etc/profile #为 Hadoop 配置环境变量
在文件末尾添加和修改如下：

```
export HADOOP_HOME=/root/hadoop/hadoop-3.1.0
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$HADOOP_HOME/bin:$PATH
```

source /etc/profile #更新配置文件

(5)运行 Hadoop

hdfs namenode -format #格式化 namenode

```
2019-04-22 17:17:07,141 INFO util.GSet: 0.029999999329447746% max memory 873 MB = 268.2 KB
2019-04-22 17:17:07,141 INFO util.GSet: capacity      = 2^15 = 32768 entries
2019-04-22 17:17:51,263 INFO namenode.FSImage: Allocated new BlockPoolId: BP-456944410-192.168.138.129-1555924671256
2019-04-22 17:17:51,367 INFO common.Storage: Storage directory /root/hadoop/hadoop-3.1.0/hadoop_tmp/dfs/name has been successfully formatted.
2019-04-22 17:17:51,408 INFO namenode.FSImageFormatProtobuf: Saving image file /root/hadoop/hadoop-3.1.0/hadoop_tmp/dfs/name/current/fsimage.ckpt_0000
000000000000000 using no compression
2019-04-22 17:17:51,546 INFO namenode.FSImageFormatProtobuf: Image file /root/hadoop/hadoop-3.1.0/hadoop_tmp/dfs/name/current/fsimage.ckpt_00000000000
00000000 of size 386 bytes saved in 0 seconds .
2019-04-22 17:17:51,608 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2019-04-22 17:17:51,639 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/192.168.138.129
************************************************************/
```

cd ../.. #返回到 hadoop-3.1.0 目录

sbin/start-all.sh #开启所有进程

结果显示 NameNodes、DataNotes、SecondaryNameNodes、ResourceManager、NodeManagers 守护进程都开启失败，报错信息如下：

```
root@ubuntu:~/hadoop/hadoop-3.1.0# sbin/start-all.sh
Starting namenodes on [localhost]
ERROR: Attempting to operate on hdfs namenode as root
ERROR: but there is no HDFS_NAMENODE_USER defined. Aborting operation.
Starting datanodes
ERROR: Attempting to operate on hdfs datanode as root
ERROR: but there is no HDFS_DATANODE_USER defined. Aborting operation.
Starting secondary namenodes [ubuntu]
ERROR: Attempting to operate on hdfs secondarynamenode as root
ERROR: but there is no HDFS_SECONDARYNAMENODE_USER defined. Aborting operation.
Starting resourcemanager
ERROR: Attempting to operate on yarn resourcemanager as root
ERROR: but there is no YARN_RESOURCEMANAGER_USER defined. Aborting operation.
Starting nodemanagers
ERROR: Attempting to operate on yarn nodemanager as root
ERROR: but there is no YARN_NODEMANAGER_USER defined. Aborting operation.
```

错误处理 1：

在以下文件顶部添加参数

vim sbin/start-dfs.sh

vim sbin/stop-dfs.sh

内容如下:

HDFS_DATANODE_USER=root

HADOOP_SECURE_DN_USER=hdfs

HDFS_NAMENODE_USER=root

HDFS_SECONDARYNAMENODE_USER=root

```
#!/usr/bin/env bash
HDFS_DATANODE_USER=root
HADOOP_SECURE_DN_USER=hdfs
HDFS_NAMENODE_USER=root
HDFS_SECONDARYNAMENODE_USER=root
```

错误处理 2：
在以下文件顶部添加参数
vim sbin/start-yarn.sh
vim sbin/stop-yarn.sh

内容如下：
YARN_RESOURCEMANAGER_USER=root
HADOOP_SECURE_DN_USER=yarn
YARN_NODEMANAGER_USER=root

```
#!/usr/bin/env bash
YARN_RESOURCEMANAGER_USER=root
HADOOP_SECURE_DN_USER=yarn
YARN_NODEMANAGER_USER=root
```

修改后再次执行：
sbin/start-all.sh

```
root@ubuntu:~/hadoop/hadoop-3.1.0# sbin/start-all.sh
WARNING: HADOOP_SECURE_DN_USER has been replaced by HDFS_DATANODE_SECURE_USER. Using value of HADOOP_SECURE_DN_USER.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
ubuntu: Warning: Permanently added 'ubuntu,192.168.138.129' (ECDSA) to the list of known hosts.
Starting resourcemanager
Starting nodemanagers
```

jps #验证是否开启成功
结果显示五个守护进程都全部启动了

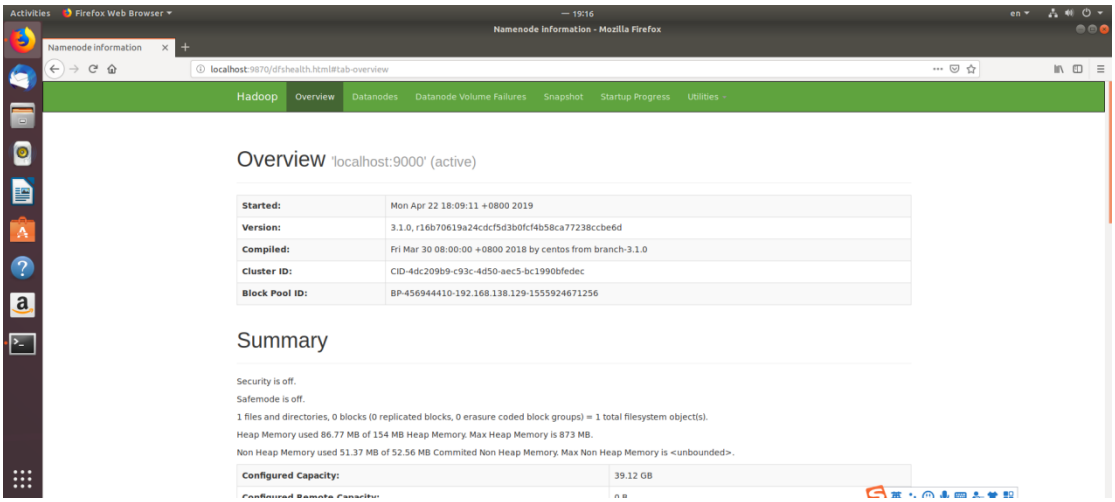
```
root@ubuntu:~/hadoop/hadoop-3.1.0# jps
32704 Jps
32321 NodeManager
31444 NameNode
32151 ResourceManager
31610 DataNode
31822 SecondaryNameNode
```

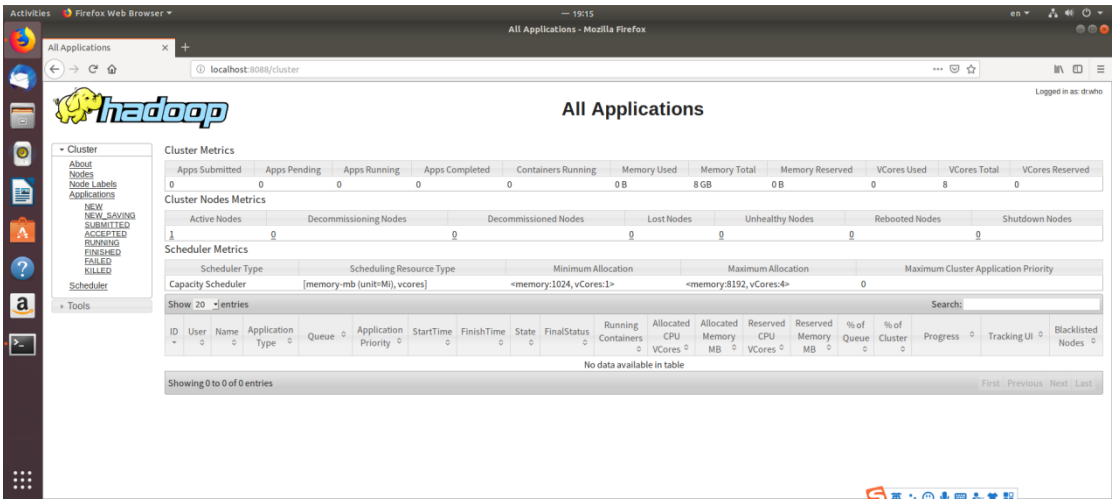
(6)运行 Hadoop 自带实例 Wordcount
ls /root/hadoop/hadoop-3.1.0/share/hadoop/mapreduce #显示 hadoop 下自带的样例程序(带有 example 字样的 jar 包)

```
hadoop-mapreduce-client-app-3.1.0.jar     hadoop-mapreduce-client-hs-plugins-3.1.0.jar       hadoop-mapreduce-client-shuffle-3.1.0.jar   lib-examples
hadoop-mapreduce-client-common-3.1.0.jar  hadoop-mapreduce-client-jobclient-3.1.0.jar        hadoop-mapreduce-client-uploader-3.1.0.jar  sources
hadoop-mapreduce-client-core-3.1.0.jar    hadoop-mapreduce-client-jobclient-3.1.0-tests.jar  hadoop-mapreduce-examples-3.1.0.jar
hadoop-mapreduce-client-hs-3.1.0.jar      hadoop-mapreduce-client-nativetask-3.1.0.jar       jdiff
```

hdfs(NameNode)管理界面：http://localhost:9870
MR 的管理界面：http://localhost:8088

我第一次访问 hdfs 管理界面时，研究了很久才发现是因为我安装的不是老师在课堂上讲解过的 Hadoop2 版本，而是 Hadoop3。现在，Hadoop3 以上版本的 webUI 在访问 hdfs 管理界面时，端口号 50070 都改为了 9870。特此记录。





#在 hdfs 云端创建一个/data/input 文件夹
bin/hdfs dfs -mkdir -p /data/input

touch words.txt #创建一个 words.txt 文件
vim words.txt #编辑 words.txt 内容如下

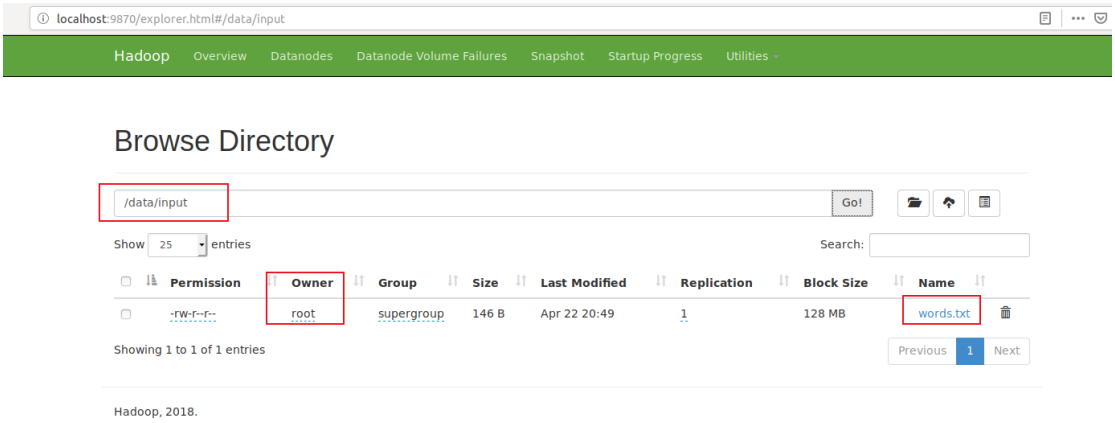
```
hello hjs
hello hdu
hello java
study python
hello javaScript
hello hadoop
goodbye hadoop
goodbye hjs
finish shangji2
study springBoot
goodbye hdu
```

bin/hdfs dfs -put words.txt /data/input #将当前目录的 word.txt 上传到云端，由主机进行文件的分布式存储

bin/hdfs dfs -ls /data/input #查看云端的/data/input 文件夹下有哪些文件

```
root@ubuntu:~/hadoop/hadoop-3.1.0# bin/hdfs dfs -ls /data/input
Found 1 items
-rw-r--r--   1 root supergroup        146 2019-04-22 20:49 /data/input/words.txt
root@ubuntu:~/hadoop/hadoop-3.1.0#
```

hadoop 可视化界面查看文件：



#运行 share/hadoop/mapreduce/ hadoop-mapreduce-examples-3.1.0.jar 这个 java 程序，调用 wordcount 方法。
#输入参数：/data/input/words.txt
#输出参数：/data/output/count_result ， 保存处理后的数据的文件夹名字

```
bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.1.0.jar
wordcount /data/input/words.txt /data/output/count_result
```

#查看运行结果
bin/hdfs dfs -cat /data/output/count_result/part-r-00000

```
root@ubuntu:~/hadoop/hadoop-3.1.0# bin/hdfs dfs -cat /data/output/count_result/part-r-00000
finish  1
goodbye 3
hadoop  2
hdu     2
hello   5
hjs     2
java    1
javaScript      1
python  1
shangji2        1
springBoot      1
study   2
```