《金融大数据处理技术》课程

实验二报告

191870068 嵇泽同

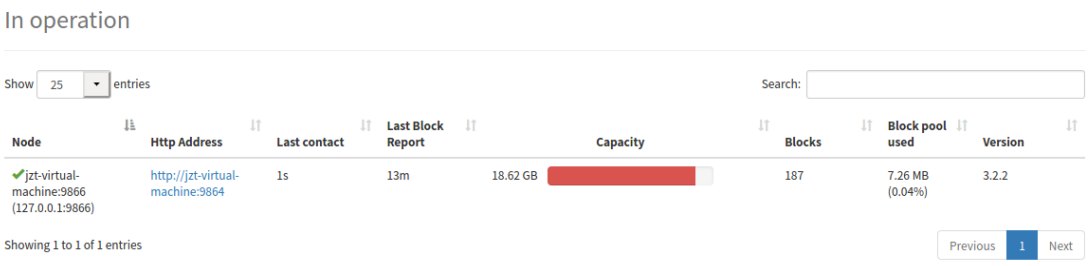
《金融大数据处理技术》课程
实验二报告
191870068 嵇泽同

一、运行结果

单机模式：

```
jzt@jzt-virtual-machine: ~/hadoop_install/hadoop-3.2.2
                Reduce shuffle bytes=25
                Reduce input records=1
                Reduce output records=1
                Spilled Records=2
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=33
                Total committed heap usage (bytes)=292397056
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=123
        File Output Format Counters
                Bytes Written=23
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ cat output/
*
1       dfsadmin
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$
```

单机伪分布：

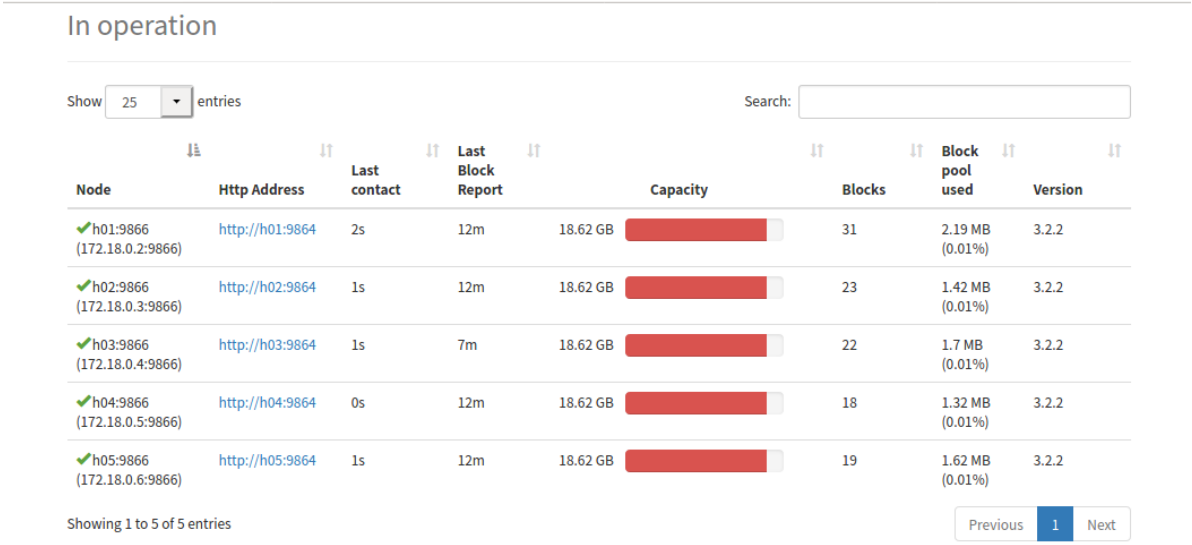



```
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ bin/hdfs dfs -cat /usr/jzt/output2/*

1       dfswoshinidie
1       dfsadmin
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$
```

（此处本人所用的input文件与教程中不同）

基于Docker的Hadoop集群：



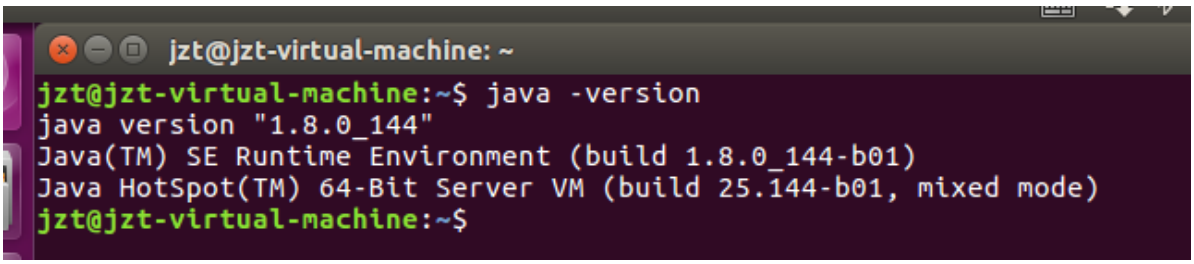

```
root@h01:/usr/local/hadoop/bin# ./hadoop fs -cat /output/part-r-00000
""AS    2
"AS     25
"AS-IS" 1
"Adaptation"    1
"COPYRIGHTS     1
"Collection"    1
"Collective     1
"Contribution"  2
"Contributor"   2
"Creative       1
"Derivative     2
"Distribute"    1
"French 2
"JDOM"  2
"JDOM", 1
"Java   1
"LICENSE").     2
"Legal  1
"License"       1
"License");     3
"Licensed       1
"Licensor"      3
```

（此处是对License.txt文件进行WordCount）

二、问题总结及解决方案

1. Java的安装配置

安装运行Hadoop首先需要安装Java。本人在安装并配置Java的过程中，即使参考了群里的若干教程链接也仍然遇到了一些问题。最终，本人参考网络上的教程https://blog.csdn.net/m0_37671741/article/details/100269736，终于顺利地完成了JDK1.8的安装：



2.Hadoop文件目录结构与教程中有较大出入

教程中一开始需要对Hadoop的若干文件配置进行修改，比如etc/hadoop/hadoop-env.sh。但是本人在下载的hadoop文件目录系统中并没有找到该路径下的对应文件。运用find命令查找该文件名后发现该文件在一个与教程中完全不同的路径下：

```
jzt@jzt-virtual-machine:/$ sudo find /home -name '*hadoop-env.sh*'
/home/jzt/hadoop_installs/hadoop-3.2.2-src/hadoop-common-project/hadoop-common/s
rc/main/conf/hadoop-env.sh
jzt@jzt-virtual-machine:/$
```

不仅如此，教程中后续需要通过bin/hadoop指令运行hadoop，但是本人的hadoop文件结构树中完全没有bin这个目录：

```
jzt@jzt-virtual-machine:~/hadoop_installs/hadoop-3.2.2-src$ ls
BUILDING.txt                hadoop-project
dev-support                 hadoop-project-dist
hadoop-assemblies           hadoop-tools
hadoop-build-tools          hadoop-yarn-project
hadoop-client-modules       Jenkinsfile
hadoop-cloud-storage-project LICENSE.txt
hadoop-common-project       NOTICE.txt
hadoop-dist                 patchprocess
hadoop-hdfs-project         pom.xml
hadoop-mapreduce-project    README.txt
hadoop-maven-plugins        start-build-env.sh
hadoop-minicluster
jzt@jzt-virtual-machine:~/hadoop_installs/hadoop-3.2.2-src$
```

上网查阅后发现是因为我犯了一个低级错误：

我在下载hadoop时面对source版本和binary版本的选项，由于不了解其区别而选择了source版本。而source版本是未编译的源代码版本、binary版本是编译后的版本，一般用户使用均使用binary版本，而本人所下载的是source版本，自然与binary版本有非常大的出入。出现该问题还是因为本人基础知识欠缺，吃一堑长一智。

3.配置环境变量失效

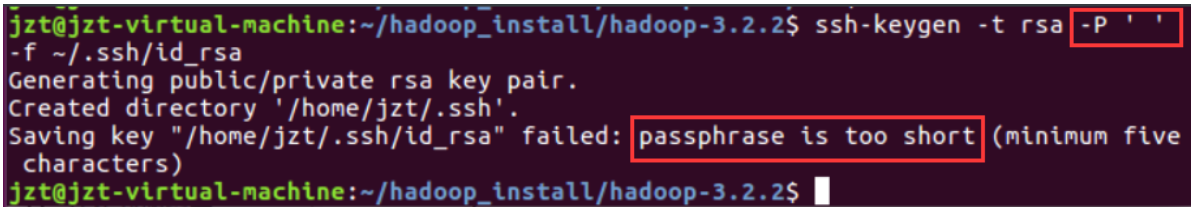
在修改若干文件、配置环境变量时，有时会出现明明已经进行了配置却还是无法正常使用的情况。后来仔细对照并修改尝试后发现是因为平时敲代码时养成了比如“=”符号左右两侧各敲一个空格的“优良习惯”，而在环境变量配置时“=”的两边是不能加空格的，加了空格后字体颜色的改变也表明了这一点：

```
export JAVA_HOME = /usr/local/java/jdk1.8.0_145
```

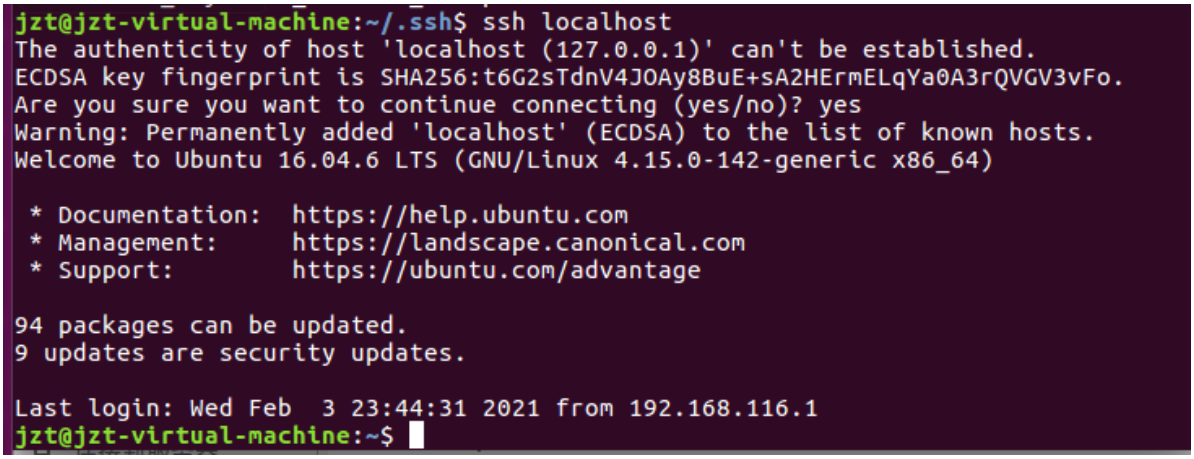
```
export JAVA_HOME=/usr/local/java/jdk1.8.0_144
```

4.免密SSH访问设置失败

问题如下图，提示密码太短：

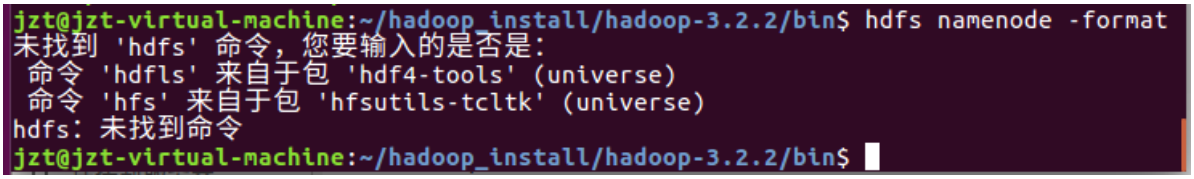


提示说密码最少需要五个字符。可那又如何设置免密呢？仔细分析观察后发现，原来是将指令从教程PDF中复制粘贴过来时两个单引号间多了一个空格，导致密码设置从原来的为空变成了一个空格，从而出现了密码过短的问题。将两个单引号间的空格删去即可成功设置免密SSH访问：

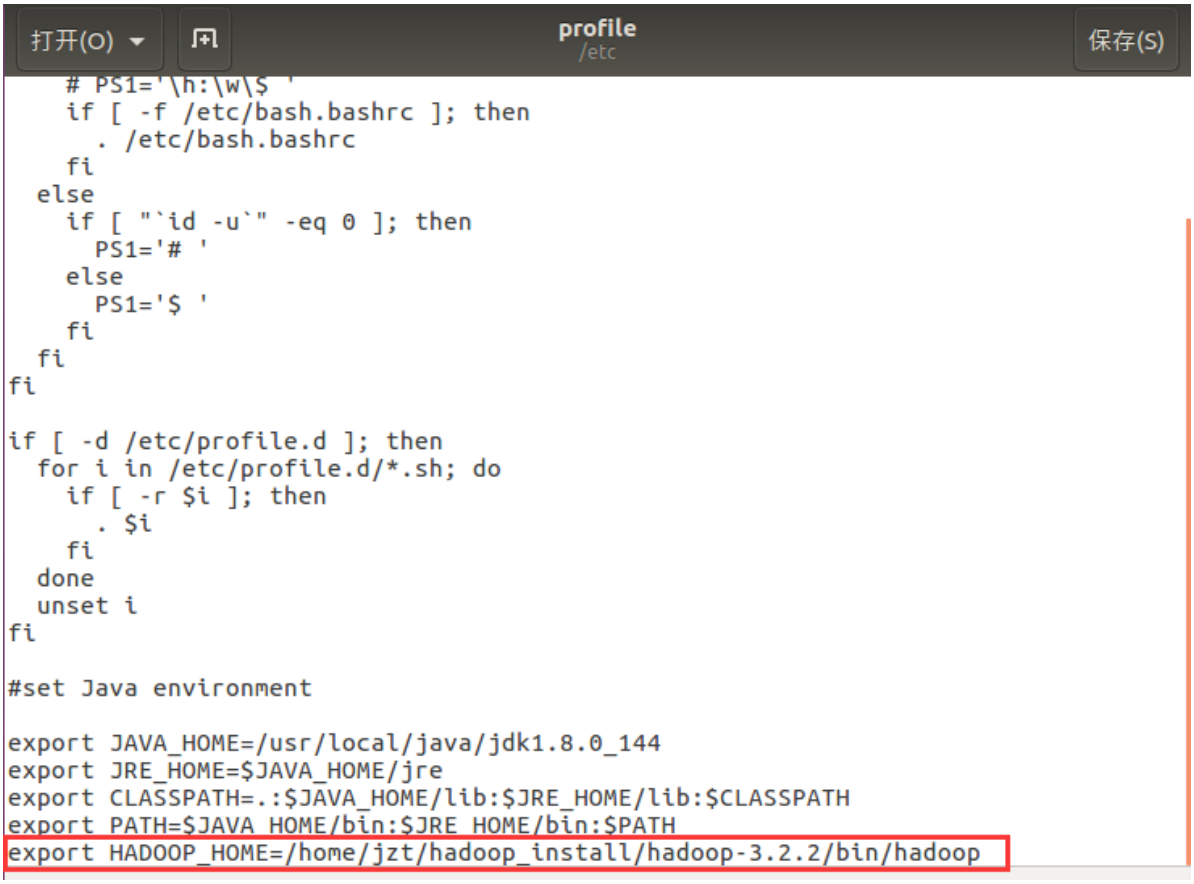


5.未找到hdfs指令

跟着教程进行时，在格式化NameNode时报错未找到'hdfs'命令：

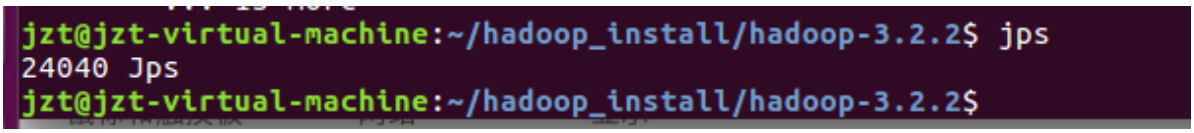


导致该问题的原因较为常见，是因为没有配置hdfs的环境变量导致找不到。配置hdfs环境变量的步骤教程中没有给出，本人自己揣测着进行了配置，之后就能成功运行hdfs指令（但后续步骤中发现该配置路径似乎有一些问题）：

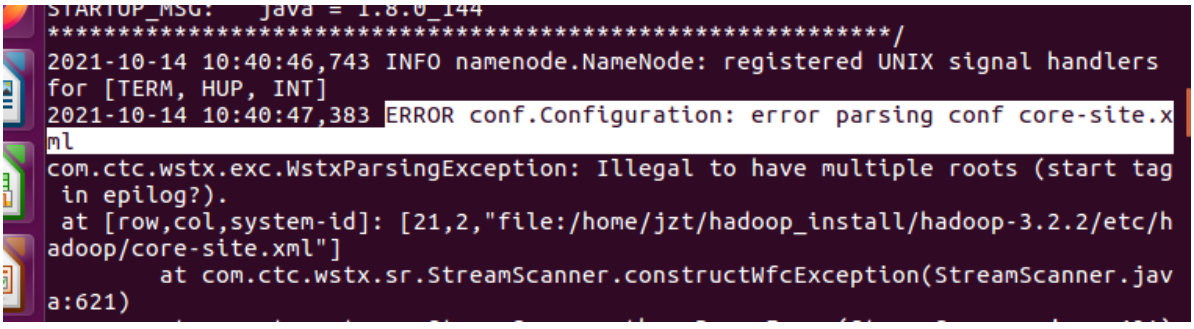


6.NameNode格式化失败

在输入NameNode格式化的指令后弹出大量白色信息，并且在最后几行没有报错提示。于是本人以为格式化成功了，并接着进行后续的步骤，却发现后续步骤无法成功顺利进行，比如jps指令查看发现缺少应有的数据：



回溯后才发现，在NameNode格式化时报出的一堆信息中“藏”了一个ERROR报错，如果不仔细看的话很容易忽视。该报错信息指出core-site.xml文件出错：



出错的core-site.xml文件如下：

```
-->

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

<configuration>
</configuration>
<configuration>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://localhost:9000</value>
        </property>
</configuration>
                                                              26,1          底端
```

思考后恍然大悟：问题出在一开始修改文件配置时，直接在已有文件尾部追加，导致出现了两个标签对。由于以前类似的修改配置文件、配置环境时往往都是尾部直接追加而较少去修改已有文件，再加上写css的习惯（css中多一个空HTML标签对往往没什么关系，格式也与这里的配置格式很相似），所以直接重新加了一个标签对。仔细想一想，其实原有的空标签对的意思应该就是让用户在这里进行配置，不应该再新建一对标签对。

删除空对后即可成功运行后续指令：

```
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [jzt-virtual-machine]
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ jps
25063 DataNode
25435 Jps
25244 SecondaryNameNode
24942 NameNode
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$
```

7.mkdir指令报错

在创建执行MapReduce作业的HDFS目录时提示mkdir指令为"Unknown command":

```
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ bin/hdfs dfs -mkdir /user
-mkdir: Unknown command
Usage: hadoop fs [generic options]
        [-appendToFile <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
```

本人怎么看都没问题，百思不得其解。上网查询后才发现是将指令从PDF中复制过来时“-”有问题：

执行 ./bin/hdfs dfs -mkdir input 时出现： -mkdir:Unknow command 的情况及解决办法

原创 爱跑步的mango 2020-03-18 12:23:06 2323 收藏 版权

文章标签： hadoop

我在ubuntu的hadoop执行 ./bin/hdfs dfs -mkdir input 时出现： -mkdir:Unknow command

```
hadoop@guyue-virtual-machine:/usr/local/hadoop$ ./bin/hdfs dfs -mkdir input
-mkdir: Unknown command
```

代码没有错， 应该是我直接在网页上复制粘贴过来，“-mkdir”的“-”字符编码错误吧，自己手动输入就可以了！

```
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ bin/hdfs dfs -mkdir /user
/jzt
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ bin/hdfs dfs -mkdir /user
```

可以发现手动输入的“-”和直接复制过来的“-”在长度上确实不一样！

8.使用YARN后运行MapReduce程序出现一系列错误

直接运行MapReduce指令报出如下错误，反复阅读报错信息仍然看不明白：

```
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ bin/hadoop jar share/hado
op/mapreduce/hadoop-mapreduce-examples-3.2.2.jar grep input output 'dfs[a-z.]+'
2021-10-14 16:03:37,037 INFO client.RMProxy: Connecting to ResourceManager at /0
.0.0.0:8032
java.net.ConnectException: Call From jzt-virtual-machine/127.0.1.1 to localhost:
9000 failed on connection exception: java.net.ConnectException: 拒绝连接; For mo
re details see:  http://wiki.apache.org/hadoop/ConnectionRefused
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
        at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstruct
orAccessorImpl.java:62)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingC
onstructorAccessorImpl.java:45)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
        at org.apache.hadoop.net.NetUtils.wrapWithMessage(NetUtils.java:836)
        at org.apache.hadoop.net.NetUtils.wrapException(NetUtils.java:760)
        at org.apache.hadoop.ipc.Client.getRpcResponse(Client.java:1566)
        at org.apache.hadoop.ipc.Client.call(Client.java:1508)
        at org.apache.hadoop.ipc.Client.call(Client.java:1405)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEng
ine.java:233)
        at org.apache.hadoop.ipc.ProtobufRpcEngine$Invoker.invoke(ProtobufRpcEng
ine.java:118)
        at com.sun.proxy.$Proxy9.delete(Unknown Source)
```

上网查阅后似乎可能是虚拟机内存不够的问题，于是在yarn-site.xml中调小了所需内存：

```
        <property>
                <name>yarn.scheduler.minimum-allocation-mb</name>
                <value>1024</value>
        </property>

        <property>
                <name>yarn.scheduler.maximum-allocation-mb</name>
                <value>2048</value>
        </property>
```

并且在虚拟机设置中增大了虚拟机内存并重启（调整虚拟机内存的步骤在此次实验中重复了多次。很多时候即使多给了一些内存仍然不够，在用docker构建hadoop集群时尤其如此）。重启后发现一个问题：在配置HDFS时没有配置NameNode和tmp文件的相关信息，而默认的tmp文件每次重新开机都会被清空，因此格式化信息会丢失。为了不出现这种情况，需要对core-site.xml文件进行配置添加：

```
<configuration>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://localhost:9000</value>
        </property>

        <property>
                <name>hadoop.tmp.dir</name>
                <value>/home/jzt/hadoop_install/hadoop-3.2.2/hadoop_tmp</value>
        </property>
</configuration>
                                                              29,1          底端
```

在调整内存后发现报出了新的错误：

```
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ sbin/start-dfs.sh
ERROR: Cannot execute /home/jzt/hadoop_install/hadoop-3.2.2/bin/hadoop/libexec/h
dfs-config.sh.
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$
```

该报错信息较为清楚：无法执行该路径下的文件。查看后发现确实不存在这个路径。该文件（“hdfs-config.sh”）实际存在路径通过find指令查找出如下：

```
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ find ./ -name '*hdfs-conf
ig*'
./libexec/hdfs-config.sh
./libexec/hdfs-config.cmd
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$
```

那为什么系统会去找这个错误的路径呢？回溯先前进行过的操作并进行分析后不难得出是在配置HADOOP_HOME（在本报告上述第五个遇到的问题中被提及）时配置了错误的环境变量路径，导致系统找到了错误的路径。

错误的路径：

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_144
export JRE_HOME=$JAVA_HOME/jre
export CLASSPATH=.:$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
export HADOOP_HOME=/home/jzt/hadoop_install/hadoop-3.2.2/bin/hadoop
```

结合报错信息中给出的系统所查找的路径和该文件实际所在的路径，不难分析得出应该删去上图中红色框出的路径部分。

在修改环境变量后再次运行MapReduce程序，仍然报错。报错信息中有如下行：

```
2021-10-17 16:36:41,971 INFO conf.Configuration: resource-types.xml not found
```

在网络上查找后，有的解决方法提出需要配置很多环境变量。而这些环境变量在教程中均未提及：

```
export JAVA_HOEM=/usr/java/default
export HADOOP_HOME=/opt/hadoop
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_YARN_HOME=$HADOOP_HOME
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_CONF_DIR=$HADOOP_HOME
export HADOOP_LIBEXEC_DIR=$HADOOP_HOME/libexec
export JAVA_LIBRARY_PATH=$HADOOP_HOME/lib/native:$JAVA_LIBRARY_PATH
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export PATH=.:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export CLASSPATH=$JAVA_HOME/lib:$JRE_HOME/lib:$CLASSPATH
```

并且还需要修改mapred-site.xml：

```
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOP_HOME}</value>
</property>
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=${HADOOP_HOME}</value>
</property>
 <property>
        <name>mapreduce.application.classpath</name>
        <value>
            ${HADOOP_HOME}/etc/hadoop,
            ${HADOOP_HOME}/share/hadoop/common/*,
            ${HADOOP_HOME}/share/hadoop/common/lib/*,
            ${HADOOP_HOME}/share/hadoop/hdfs/*,
            ${HADOOP_HOME}/share/hadoop/hdfs/lib/*,
            ${HADOOP_HOME}/share/hadoop/mapreduce/*,
            ${HADOOP_HOME}/share/hadoop/mapreduce/lib/*,
            ${HADOOP_HOME}/share/hadoop/yarn/*,
            ${HADOOP_HOME}/share/hadoop/yarn/lib/*
        </value>
    </property>
```

在进行上述修改调整后，再次运行MapReduce程序时终于能看到map和reduce进度顺利不断增加了！

9.MapReduce最后一步报错

在上一个问题解决后终于能看到map和reduce进度不断增加了，但是在进行到最后一步时却报错：

```
2021-10-17 22:26:29,660 INFO mapreduce.Job:  map 0% reduce 0%
2021-10-17 22:26:43,876 INFO mapreduce.Job:  map 16% reduce 0%
2021-10-17 22:26:44,882 INFO mapreduce.Job:  map 19% reduce 0%
2021-10-17 22:26:57,987 INFO mapreduce.Job:  map 38% reduce 0%
2021-10-17 22:27:10,062 INFO mapreduce.Job:  map 41% reduce 0%
2021-10-17 22:27:11,067 INFO mapreduce.Job:  map 53% reduce 0%
2021-10-17 22:27:21,125 INFO mapreduce.Job:  map 66% reduce 18%
2021-10-17 22:27:22,129 INFO mapreduce.Job:  map 69% reduce 18%
2021-10-17 22:27:27,164 INFO mapreduce.Job:  map 69% reduce 23%
2021-10-17 22:27:30,186 INFO mapreduce.Job:  map 72% reduce 23%
2021-10-17 22:27:31,191 INFO mapreduce.Job:  map 81% reduce 23%
2021-10-17 22:27:32,195 INFO mapreduce.Job:  map 84% reduce 23%
2021-10-17 22:27:33,204 INFO mapreduce.Job:  map 84% reduce 28%
2021-10-17 22:27:41,252 INFO mapreduce.Job:  map 94% reduce 28%
2021-10-17 22:27:41,254 INFO mapreduce.Job: Task Id : attempt_1634478073803_0003
_m_000031_0, Status : FAILED
Error: java.io.FileNotFoundException: Path is not a file: /user/jzt/input2/shell
profile.d
        at org.apache.hadoop.hdfs.server.namenode.INodeFile.valueOf(INodeFile.ja
va:90)
```

上图中红框框出的信息其实看不出什么，关键在于红框下面的那一行信息：Path is not a file。阅读该路径后发现，该文件名为“shellprofile.d”，是input文件夹中众多文件的其中一个。“.d”后缀代表什么我不清楚，但是该MapReuduce程序执行的是grep，即文本查找，不常见的“.d”后缀结合报错信息不难得出结论：该文件极有可能不是文本文件，不能进行查找，因此报错。上网查阅后得知“.d”后缀文件是用gcc编译.c文件时用来供gdb调试的文件，也自然无法进行文本查找。由于本实验重点在于安装和配置，关于input文件到底是什么、输出结果如何其实并不重要，关键在于能够顺利执行即可。因此我在这里直接用了单机模式中的input文件夹内容作为input内容（进行了微调——添加了一行符合搜索条件的文本以区别两次运行的结果），而非像教程中所给出的那样将etc/hadoop目录下的所有文件作为input。这样做的好处是避免了可能出现本问题中所出现的无法进行查找的文件。此时再次运行MapReduce程序，终于顺利得出结果：

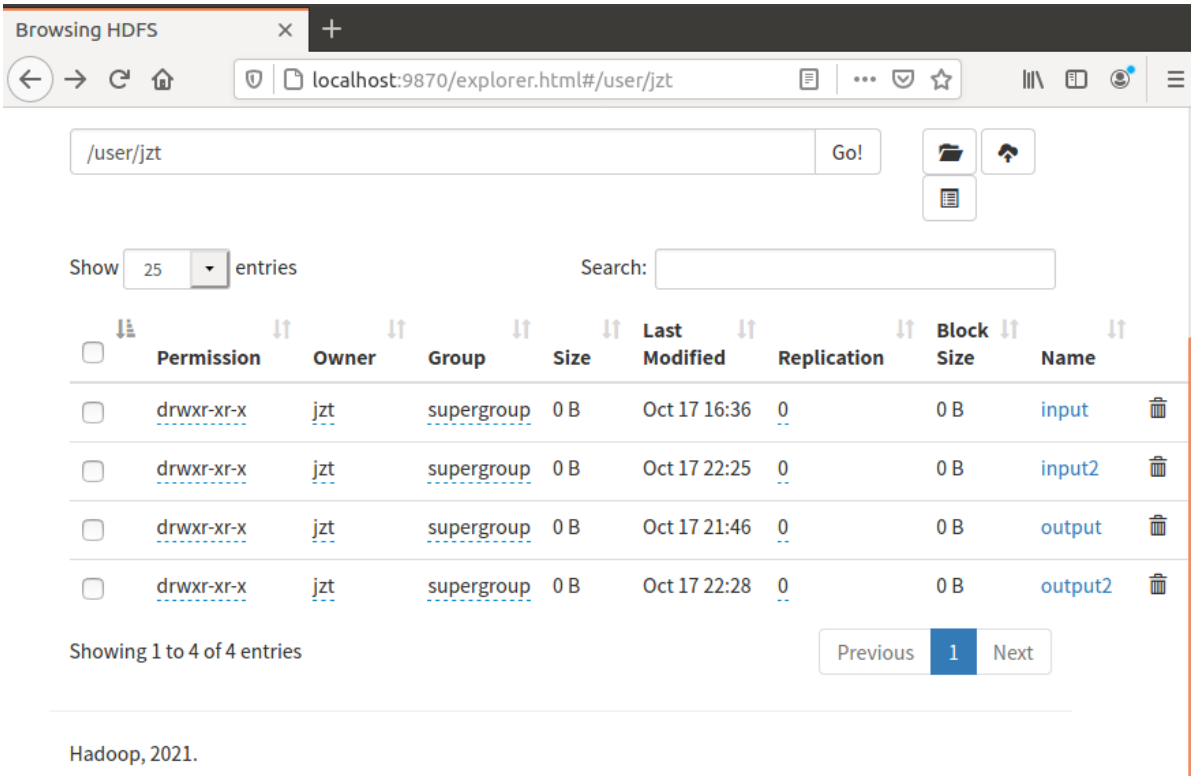
```
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$ bin/hdfs dfs -cat /usr/jzt/output2/*

1       dfswoshinidie
1       dfsadmin
jzt@jzt-virtual-machine:~/hadoop_install/hadoop-3.2.2$
```

输出结果的第二行是原有的input文件中的内容，第一行是我手动新填的符合搜索条件的文本。由此可见程序运行成功，得到了正确的结果。

10.HDFS目录混乱

由于一直是在命令行中进行操作，而且摸索过程中执行了一些“脏指令”，且有的指令执行到一半就被终止，因此导致HDFS目录较为混乱，拉取到本地后也较为混乱，出现output文件夹中还有output文件夹等问题，让我有点晕头转向。解决方法是登录localhost:9870，进行可视化查看：



这样查看，结构就非常清晰，内容也很清楚，执行指令时也可以有的放矢。

11.镜像下载文件失败

在构建基于docker的hadoop集群时，按照知乎上的步骤从华中科技大学的镜像上下载hadoop时提示connection refused：

```
root@5f3460df7352:~# wget http://mirrors.hust.edu.cn/apache/hadoop/common/hadoop-3.2.0/hadoop-3.2.0.tar.gz
--2021-10-17 17:49:04--  http://mirrors.hust.edu.cn/apache/hadoop/common/hadoop-3.2.0/hadoop-3.2.0.tar.gz
Resolving mirrors.hust.edu.cn (mirrors.hust.edu.cn)... 202.114.18.160
Connecting to mirrors.hust.edu.cn (mirrors.hust.edu.cn)|202.114.18.160|:80... failed: Connection refused.
root@5f3460df7352:~#
```

后来发现好像华中科技大学的镜像网站不知道为什么挂了（忍不住吐槽一下）。解决方法是从宿主机上挂载下载好的hadoop到容器里。

12.docker容器的hosts文件修改无法保存

在构建基于docker的hadoop集群时，需要修改容器的hosts文件来添加其他主机。我的第一反应就是直接vim修改。但是这样修改后，虚拟机一重启，所有的hosts修改都被复原了。我于是又尝试在源容器中修改hosts文件，然后再制作成镜像，以为以这样的镜像导出的容器都会有同样的、修改过的hosts文件。但是结果仍然是失败。上网查阅后得知，/etc/hosts, /etc/resolv.conf和/etc/hostname容器中的这三个文件不存在于镜像,而是存在于于/var/lib/docker/containers/，在启动容器的时候，通过mount的形式将这些文件挂载到容器内部。因此，如果在容器中修改这些文件的话，修改部分不会存在于容器的top layer，而是直接写入这三个物理文件中。每次Docker在启动容器的时候，都会重新构建新的/etc/hosts文件。

解决方法是在调用docker run指令创建容器时添加--add-host指令：

```
jzt@jzt-virtual-machine:~$ sudo docker run -it --network hadoop -h h03 --name h03 --add-host h01:172.18.0.2 --add-host h02:172.18.0.3 --add-host h03:172.18.0.4 --add-host h04:172.18.0.5 --add-host h05:172.18.0.6 newuhadoop /bin/bash
```

以该种方式创建的容器，hosts内容会被修改且修改不会随着重启等操作而复原。

13.ssh连接操纵其他主机时越来越卡

在此次实验中本人开启了五个容器来构建集群。其中有一步需要在五个容器中都修改一个文件。我直接采取了简单粗暴的ssh连接其他主机并修改的步骤：host1——ssh host2——修改host2的文件——ssh host3——修改host3的文件——……——ssh host5——修改host5的文件。但是随着操作的不断进行，操作变得越来越卡，甚至接近于卡死的程度。我当时不知道为什么，直到操作终于进行完、退出容器时我才发现了问题：修改完host5的文件后输入exit，此时回到的不是我以为的虚拟机宿主机，而是host4。再exit则回到host3……我在不断ssh的过程中忘了我一直在执行“套娃”的过程：host1操纵host2操纵host3操纵host4操纵host5。难怪会越来越卡！

三、思考及总结

由于本次实验涉及的主要是软件和环境的配置、运行，不涉及具体的代码和程序，因此在该过程中本人学到的主要是对于ubuntu的了解和掌握、对于软件安装和环境配置的一些经验和心得、对于虚拟机和docker的进一步了解和掌握以及一些计算机的基础知识等等，比如source版本和binary版本的区别、配置环境变量的要点、复制指令时需要多注意、要善于发现和阅读报错信息等等。常言道：“磨刀不误砍柴工”，在本次实验中，我更多地学到的就是如何更好、更快地“磨刀”，其实如果对于“磨刀”掌握不精，那势必将“大误砍柴工”：别人都已经在写算法跑代码了，你还在苦兮兮地配环境呢！