实验：文档倒排索引

班级： 学号： 姓名：

一、实验要求

文档倒排索引实验，是通过搭建Hadoop分布式集群，利用MapReduce算法，实现对海量数据文件的倒排索引，并将结果存入Hbase数据库中。

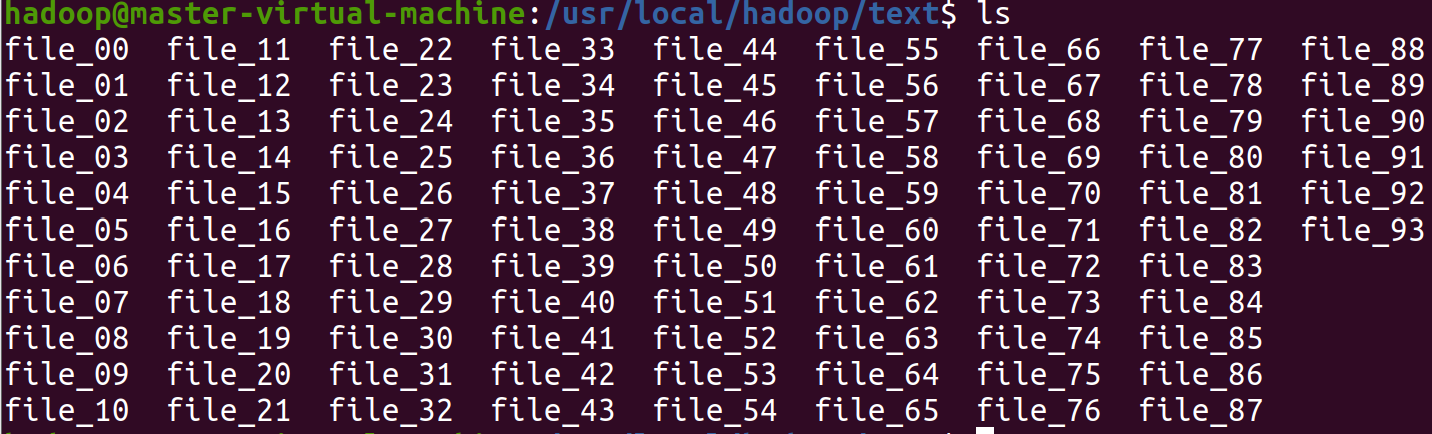
目标就是通过对Hadoop分布式框架以及Hbase架构的学习，能够在Linux操作系统环境下搭载自己的Hadoop，Hbase分布式框架，实现对海量数据的倒排索引，并将MapReduce后的数据直接插入到Hbase所建立的表中。

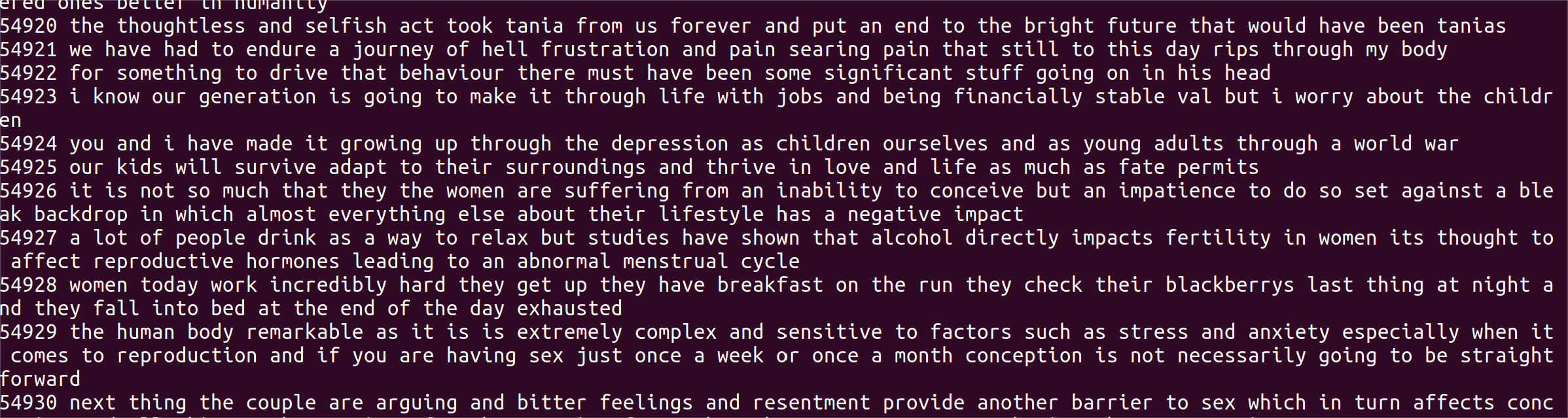
二、数据准备与说明

实验的原始文件：压缩文件500MB，解压后1.32GB

按照每100000行为一个文件进行划分，共划分出93个文件，文件名格式为file\_

划分代码为：split -l 100000 sentences.txt -d -a 2 file\_

划分出的文件为：

源数据内容为带行号的英文文本，每个单词间用空格分割：

三、环境简要说明

Linux操作系统，在VMware16pro 上运行ubuntu-20.04.5虚拟机。

共一台虚拟机hadoop

Jdk版本：java-8-openjdk-amd64

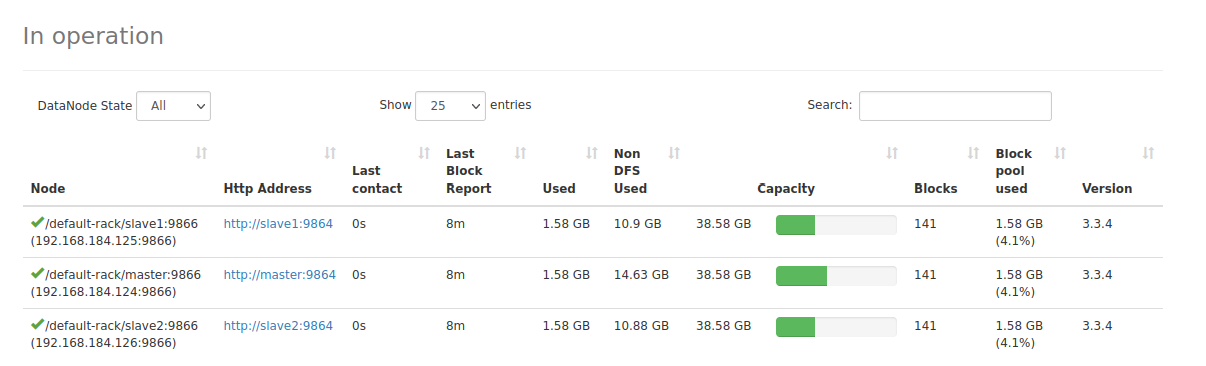
路径：/usr/lib/jvm/java-8-openjdk-amd64

(直接sudo apt-get install openjdk-8-jdk安装就好)

**Hadoop-3.3.4**

配置：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 主机名 | IP | HDFS | YARN | Zookeeper | Hbase |
| master | 192.168.x.124/24 | NameNode,DataNode | NodeManager,ResourseManager | QuorumPeerMain | HMaster |
| slave1 | 192.168.x.125/24 | DataNode,SecondaryNameNode | NodeManager | QuorumPeerMain | HRegionServer |
| slave2 | 192.168.x.126/24 | DataNode | NodeManager | QuorumPeerMain | HRegionServer |

****

1. 环境配置

创建一台虚拟机

登录hadoop 用户后 更新系统软件：

更新软件列表：sudo apt-get update

更新软件：sudo apt-get upgrade

①编辑静态网络配置信息：

sudo gedit /etc/netplan/01-network-manager-all.yaml

network:

ethernets:

ens33:

dhcp4: no

#这里注意要修改第三个为自己的网络配置

addresses: [192.168.~.124/24]

optional: true

#这里注意要修改第三个为自己的网络配置

gateway4: 192.168.~.2

nameservers:

addresses: [8.8.8.8,8.8.8.4]

version: 2

renderer: NetworkManager

更新网络配置：sudo netplan apply

查看是否成功：ip addr

让系统随机分配网络：sudo dhclient -v

②主机名和ip地址映射

sudo gedit /etc/hosts

192.168.~.124 master

192.168.~.125 slave1

192.168.~.126 slave2

③安装ssh-server

sudo apt-get install openssh-server

④安装、配置jdk

sudo apt-get install openjdk-8-jdk

查看java是否安装成功：java -version

配置java环境：gedit ~/.bashrc

在最前面加入：export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

使环境变量生效：source ~/.bashrc

使用：java -version 查看jdk的版本 whereis java 查看jdk的路径

⑤安装hadoop

解压文件到 /usr/local/ 目录下 改名为 hadoop

然后修改文件的权限：sudo chown -R hadoop ./hadoop (hbase)

（将文件置为 hadoop 用户所有）

查看hadoop是否安装成功：

进入目录：cd hadoop

执行指令：./bin/hadoop version

进行Hadoop 完全分布式的配置

在hadoop路径下创建文件夹tmp: mkdir tmp

**gedit ./etc/hadoop/workers**

master

slave1

slave2

**gedit ./etc/hadoop/core-site.xml**

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<!-- hadoop 存储路径-->

<value>/usr/local/hadoop/tmp</value>

<description>A base for other temporary directories.</description>

</property>

<property>

<name>fs.defaultFS</name>

<!-- 告诉 NN 在那个机器，NN 使用哪个端口号接收客户端和 DN 的RPC请求-->

<value>hdfs://master:9000</value>

</property>

</configuration>

**gedit ./etc/hadoop/hdfs-site.xml**

<configuration>

<property>

<!-- 设置默认副本数量为3-->

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

<!-- 主namenode 信息存储目录-->

<name>dfs.namenode.name.dir</name>

<value>/usr/local/hadoop/tmp/dfs/name</value>

</property>

<property>

<name>dfs.namenode.secondary.http-address</name>

<value>slave1:50090</value>

</property>

<property>

<!--datanode 存储位置-->

<name>dfs.datanode.data.dir</name>

<value>/usr/local/hadoop/tmp/dfs/data</value>

</property>

</configuration>

**gedit ./etc/hadoop/yarn-site.xml**

<configuration>

<property>

<name>yarn.resourcemanager.hostname</name>

<value>master</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

**gedit ./etc/hadoop/mapred-site.xml**

<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>

<property>

<name>yarn.app.mapreduce.am.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>

</property>

<property>

<name>mapreduce.map.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>

</property>

<property>

<name>mapreduce.reduce.env</name>

<value>HADOOP\_MAPRED\_HOME=/usr/local/hadoop</value>

</property>

</configuration>

克隆出两台虚拟机slave1,slave2

修改主机名：sudo gedit /etc/hostname

修改网络配置信息：sudo gedit /etc/netplan/01-network-manager-all.yaml

addresses: [192.168.x.125/24]

addresses: [192.168.x.126/24]

更新网络配置：sudo netplan apply

在三台虚拟机上生成ssh密匙：ssh-keygen -t rsa

（若原先有密匙需要进入cd ~/.ssh/删除）

在三台虚拟机上互传密匙，以便可以互相免密登录

ssh-copy-id slave1

ssh-copy-id slave2

~~~

执行namenode格式化：./bin/hdfs namenode -format (当提示输入Y/N时，输入Y)

开启namenode、datanode：./sbin/start-dfs.sh

查看是否开启成功：jps（4个信息JPS,SecondaryNameNode,NameNode,DataNode）

开启yarn：./sbin/start-yarn.sh

查看是否开启成功：jps (新加ResourceManager与NodeManager)

查看网页端的信息：http://master:9870

伪分布读取的是HDFS上的数据使用时需要上传数据到HDFS上

在HDFS上创建目录：./bin/hdfs dfs -mkdir -p /data/input （创建了data/input路径的文件夹）

上传文件到该目录下：./bin/hdfs dfs -put text/\* /data/input

查看是否上传成功：./bin/hdfs dfs -ls /data/input

HDFS 退出安全模式：./bin/hdfs dfsadmin -safemode leave

⑥安装zookeeper

在三台虚拟机上解压改名后授权：sudo chown -R hadoop ./zookeeper

在zookeeper路径下创建文件夹tmp: mkdir tmp

在tmp下创建文件夹myid：gedit myid 并分别输入1/2/3

配置zookeeper: cd zookeeper/conf/

cp zoo\_sample.cfg zoo.cfg

gedit zoo.cfg

修改： dataDir=/usr/local/zookeeper/tmp

增加：

server.1=master:2888:3888

server.2=slave1:2888:3888

server.3=slave2:2888:3888

在三台虚拟机上分别开启zookeeper：

./bin/zkServer.sh start

查询状态：./bin/zkServer.sh status

关闭：./bin/zkServer.sh stop

⑦安装Hbase

解压改名后授权：sudo chown -R hadoop ./hbase

查看是否解压成功：./bin/hbase version

配置Hbase:

**gedit ./conf/hbase-env.sh**

# 配置JDK安装路径

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

# 设置HBase的日志目录

export HBASE\_LOG\_DIR=/usr/local/hbase/logs

export HBASE\_CLASSPATH=/usr/local/hbase/conf

export HBASE\_PID\_DIR=/usr/local/hbase/pids

# 使用独立的ZooKeeper集群

export HBASE\_MANAGES\_ZK=false

**gedit ./conf/hbase-site.xml**

<configuration>

<property>

<!-- 同hadoop配置文件core-site.xml中fs.defaultFS的值 -->

<name>hbase.rootdir</name>

<value>hdfs://master:9000/hbase</value>

</property>

<property>

<!-- false是单机模式，true是分布式模式 -->

<!-- 当使用外置zookeeper时必须配置为true，仅在使用hbase自带zookeeper时配置为false -->

<name>hbase.cluster.distributed</name>

<value>true</value>

</property>

<property>

<!-- zookeeper的数据目录，启动hbase的用户需对此目录有读写权限 -->

<name>hbase.zookeeper.property.dataDir</name>

<value>/usr/local/zookeeper/tmp</value>

</property>

<property>

<name>hbase.zookeeper.quorum</name>

<value>master,slave1,slave2</value>

</property>

<property>

<!-- 指定HBase Master web页面访问端口-->

<name>hbase.master.info.port</name>

<value>60010</value>

</property>

<!-- 在分布式的情况下一定要设置，不然容易出现Hmaster起不来的情况 -->

<property>

<name>hbase.unsafe.stream.capability.enforce</name>

<value>false</value>

</property>

<property>

<!-- 指定HBase RegionServer web页面访问端口 -->

<name>hbase.regionserver.info.port</name>

<value>60030</value>

</property>

</configuration>

配置regionservers: gedit ./conf/regionservers

slave1

slave2

启动hbase：./bin/start-hbase.sh

查看是否启动成功：jps （多三个结点HRegionServer、HQuorumPeer、HMaster）

关闭hbase：./bin/stop-hbase.sh

打开shell：./bin/hbase shell

五、实验收获与体会（黑体四号字）

宋体小四号字，行距22磅

附录：程序清单及说明（黑体四号字）

宋体小四号字，行距22磅

（列出文件名及说明即可，不需要在此处复制代码，代码直接以源文件形式提供，但源文件中对代码要有必要的注释和说明）