* 实验环境：

虚拟机软件：VirtualBox6.16

虚拟机系统：ubuntu18.04

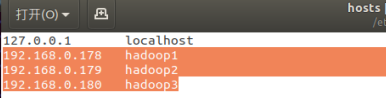
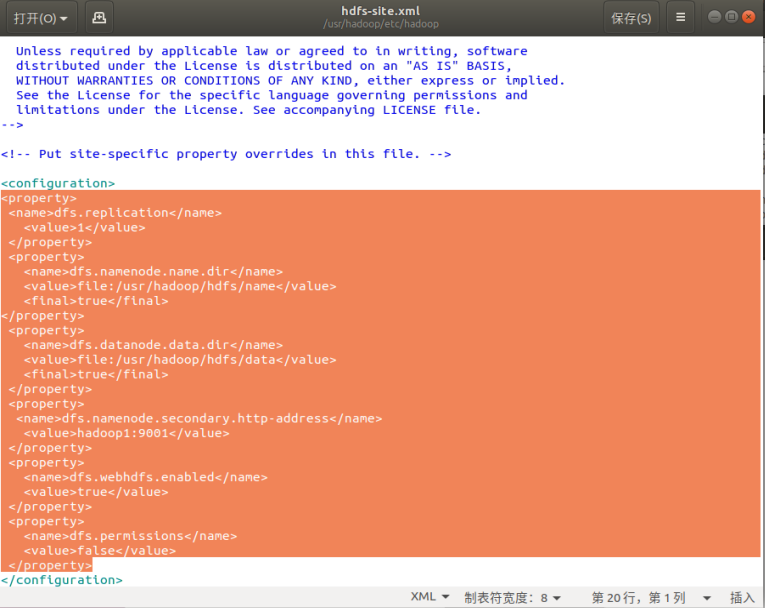
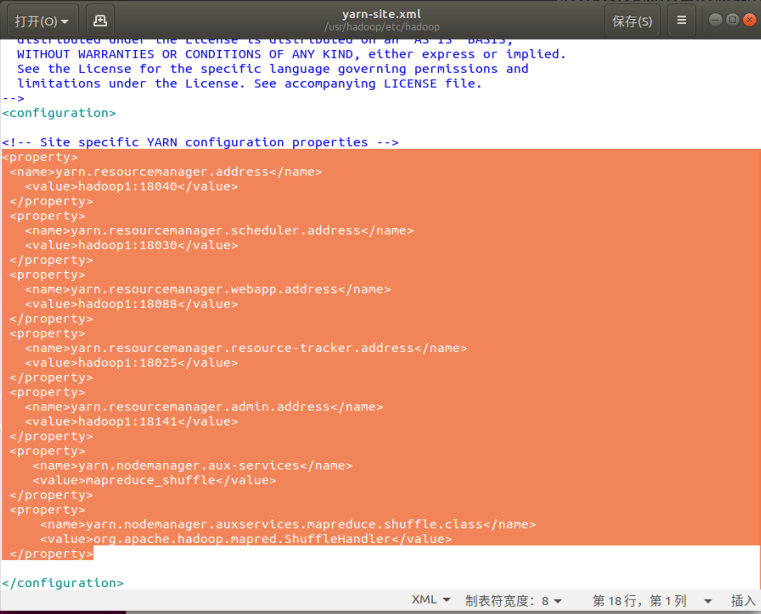
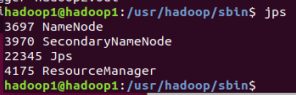
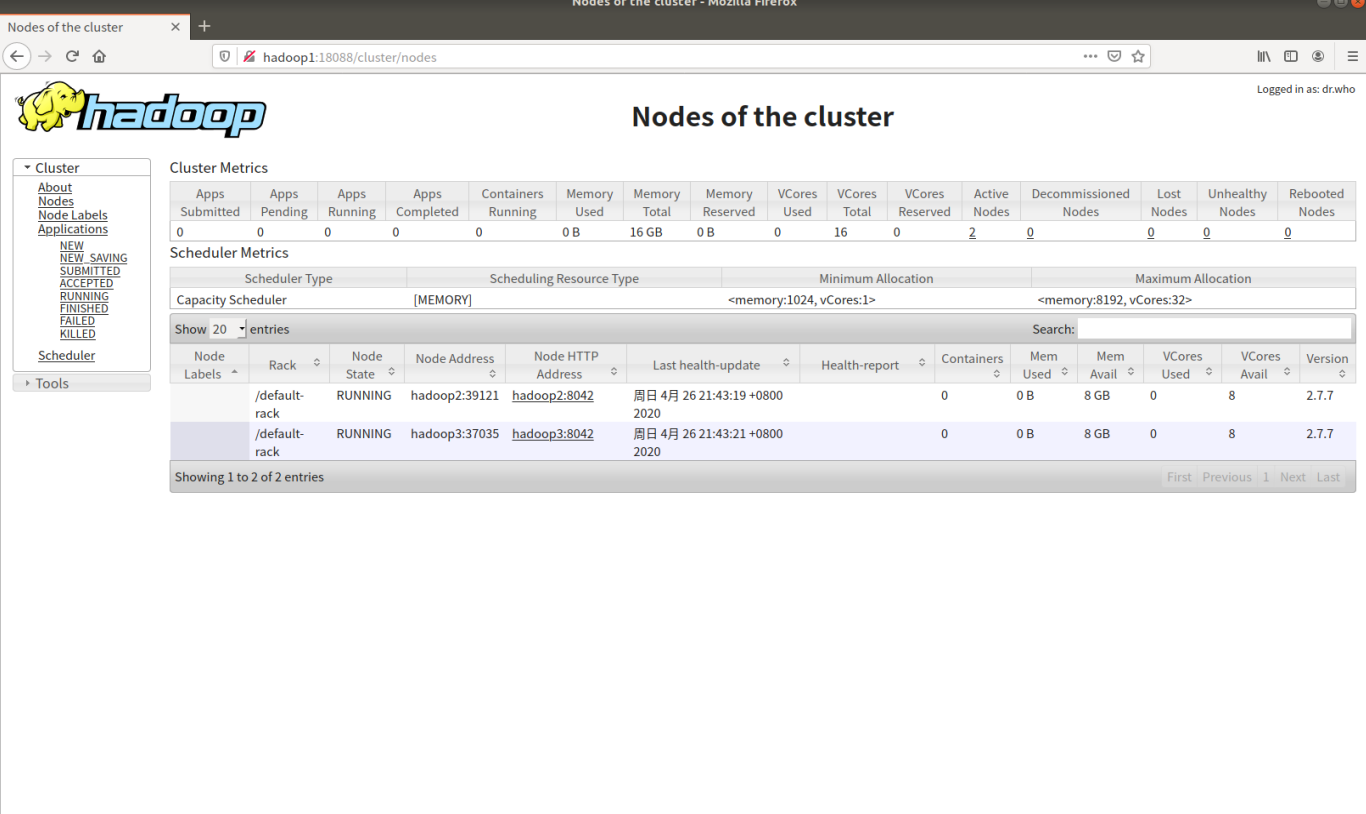
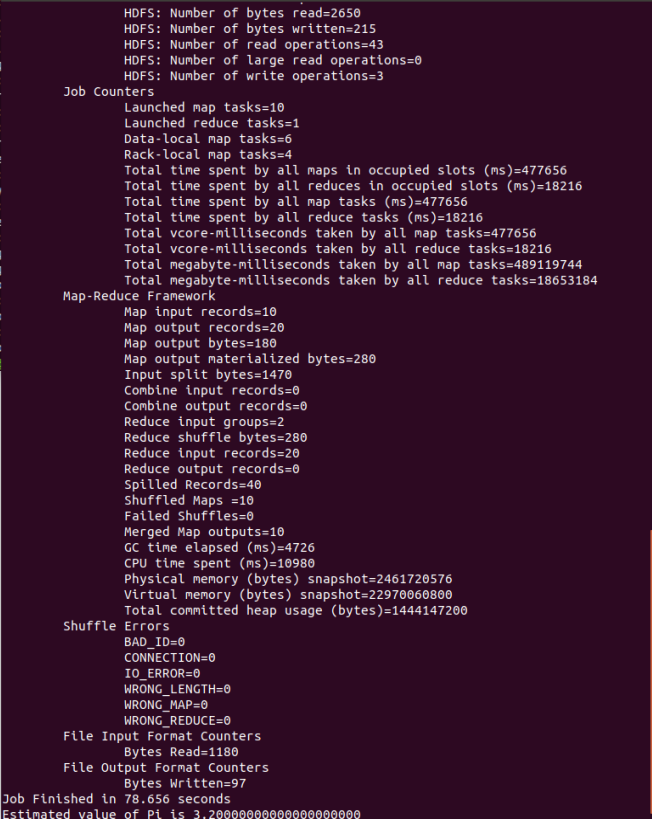
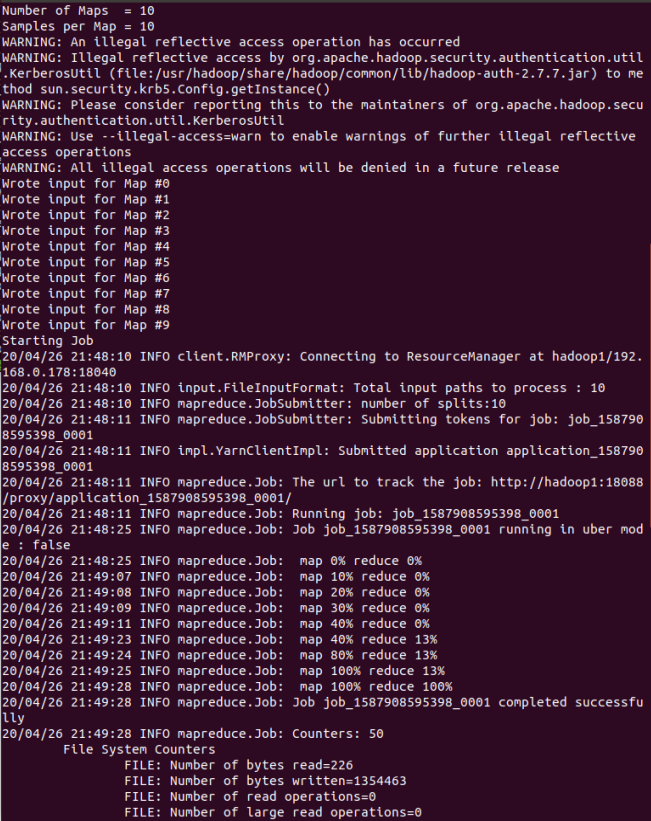
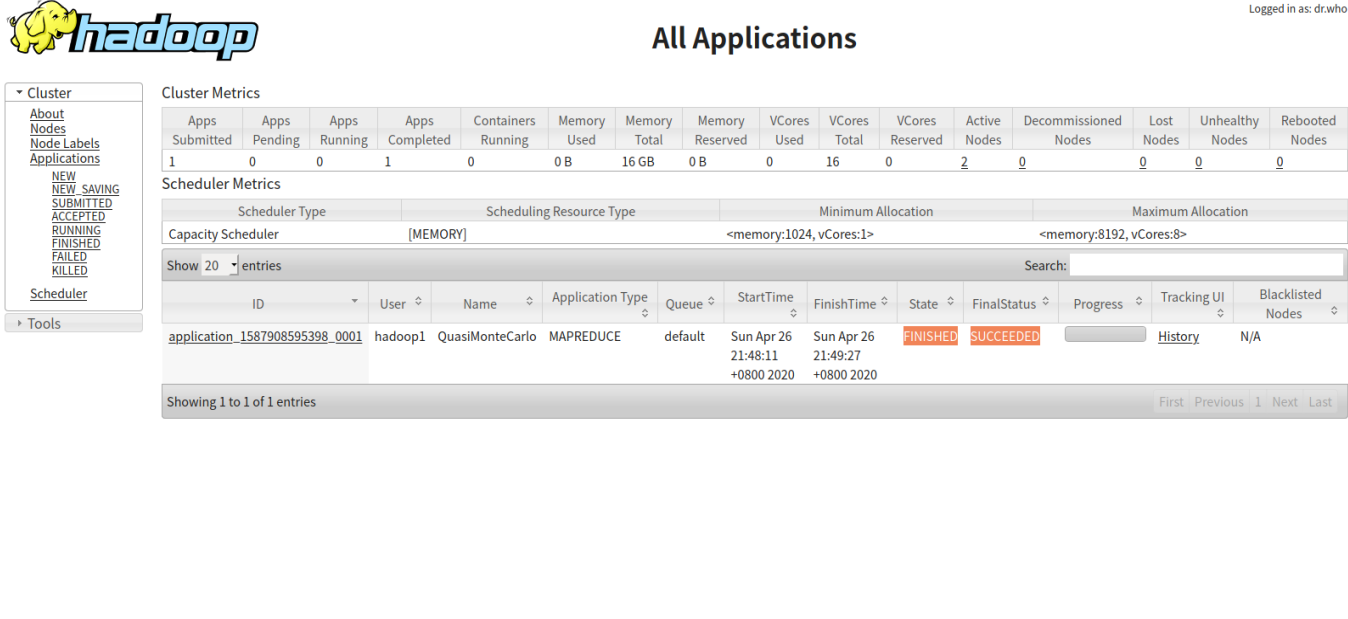
虚拟机配置：1\*namenode：2c4g、2\*datanode：1c2g

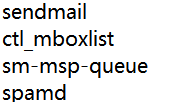
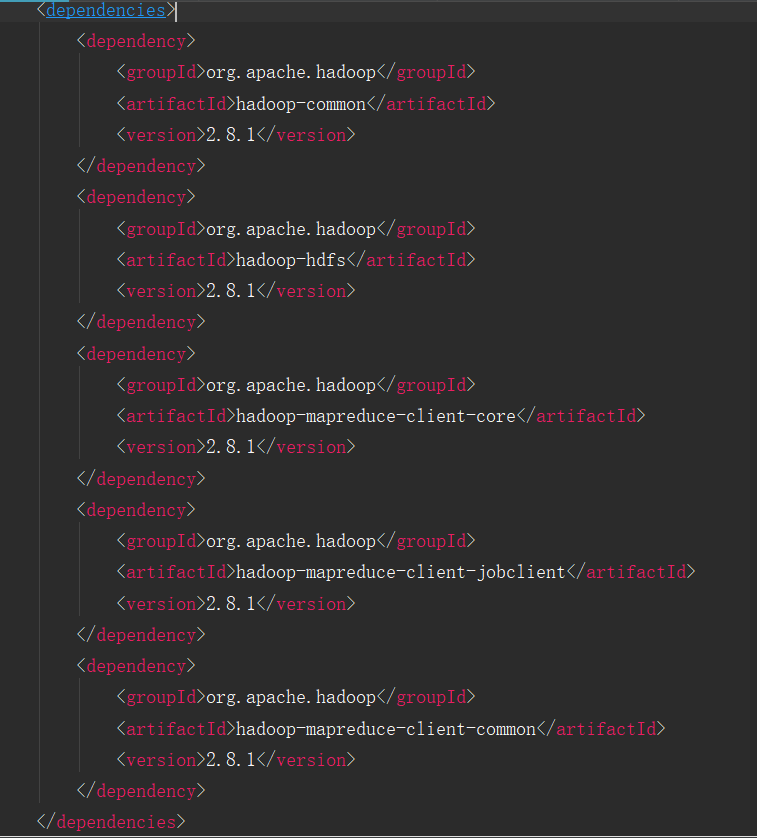
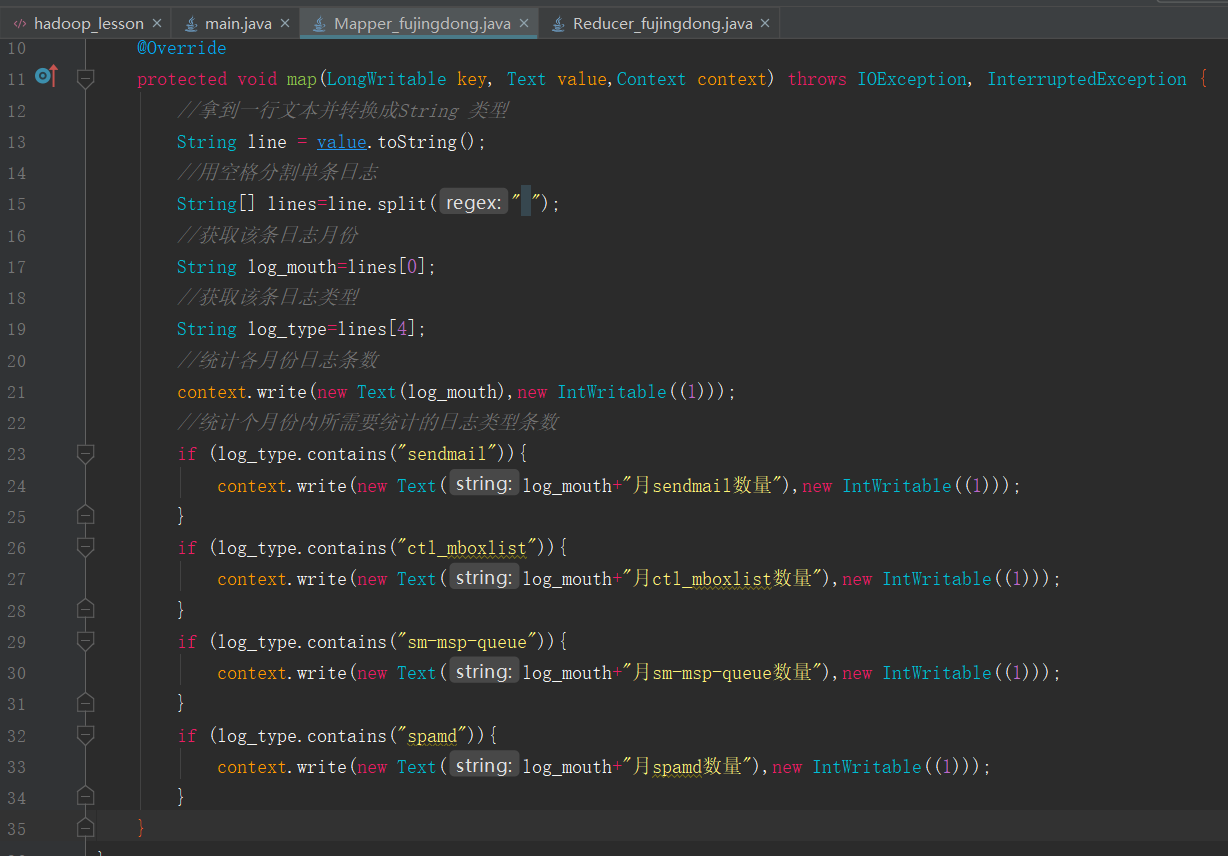
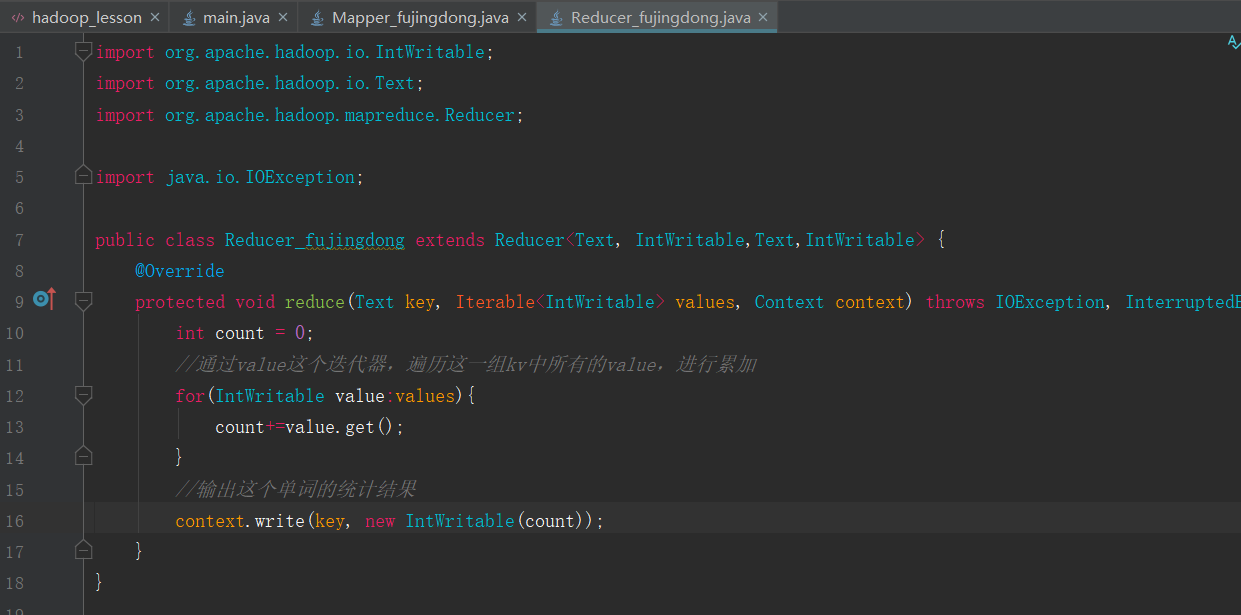
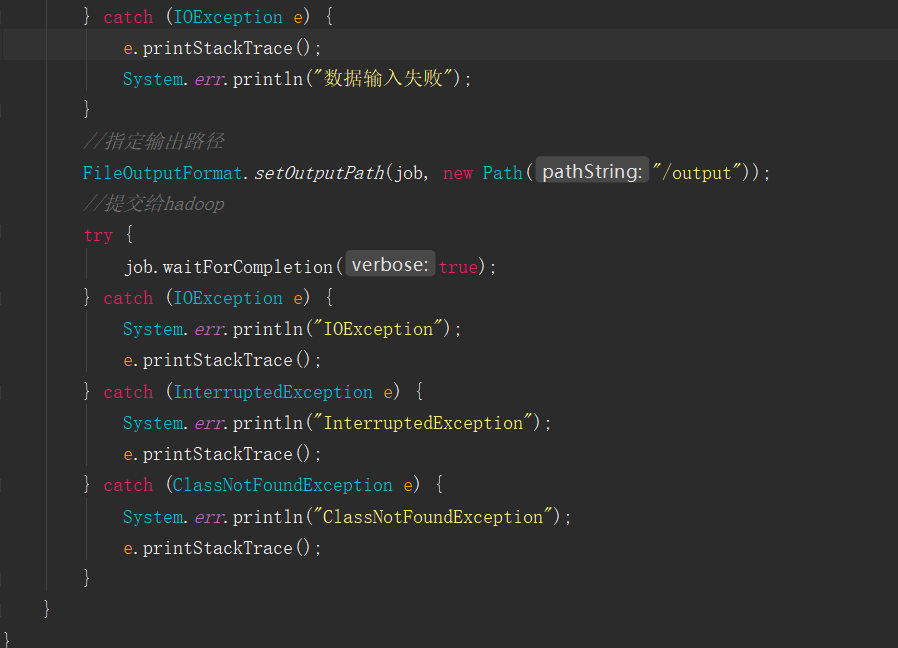
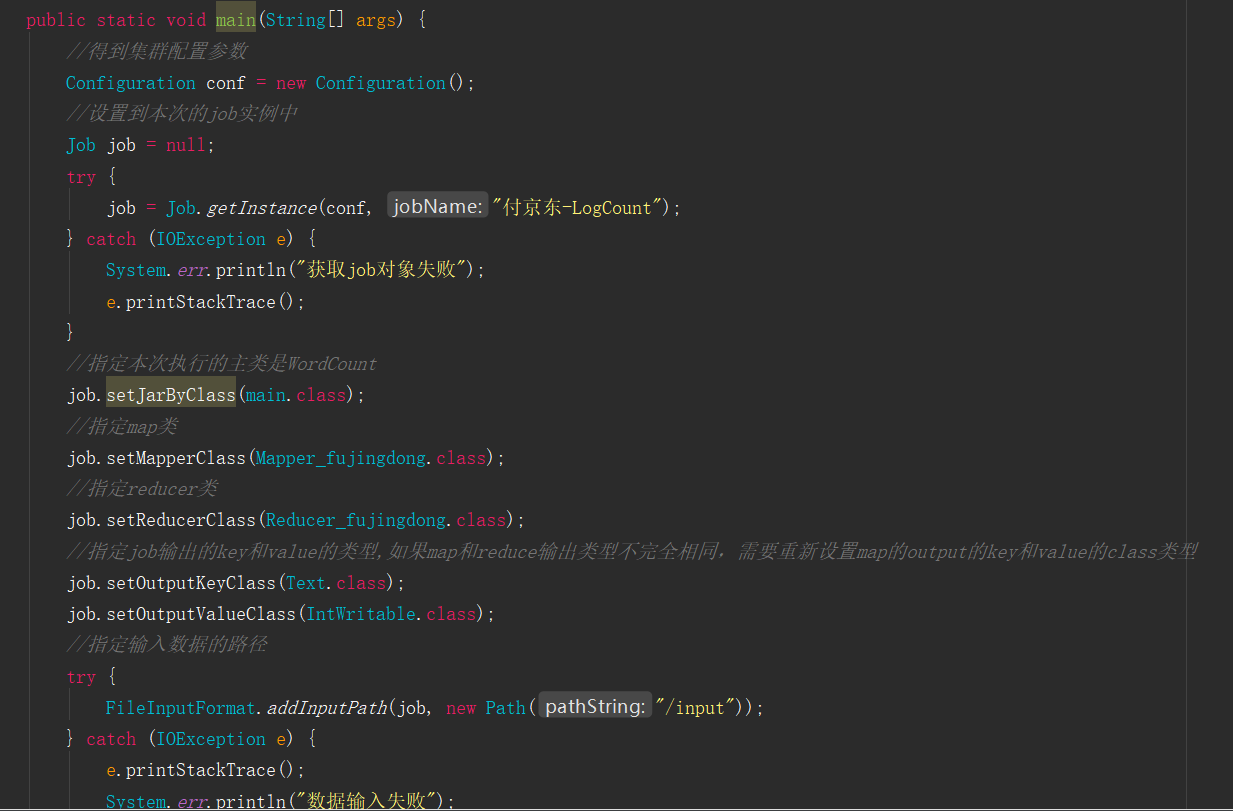
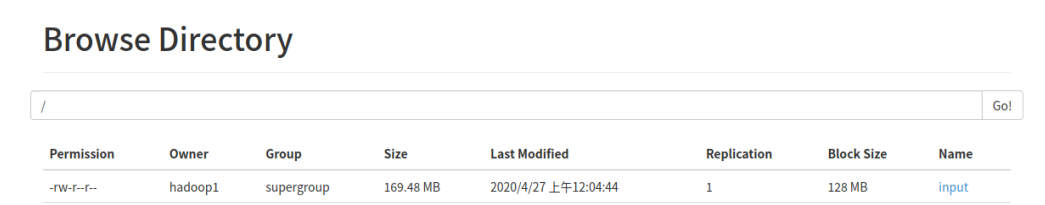
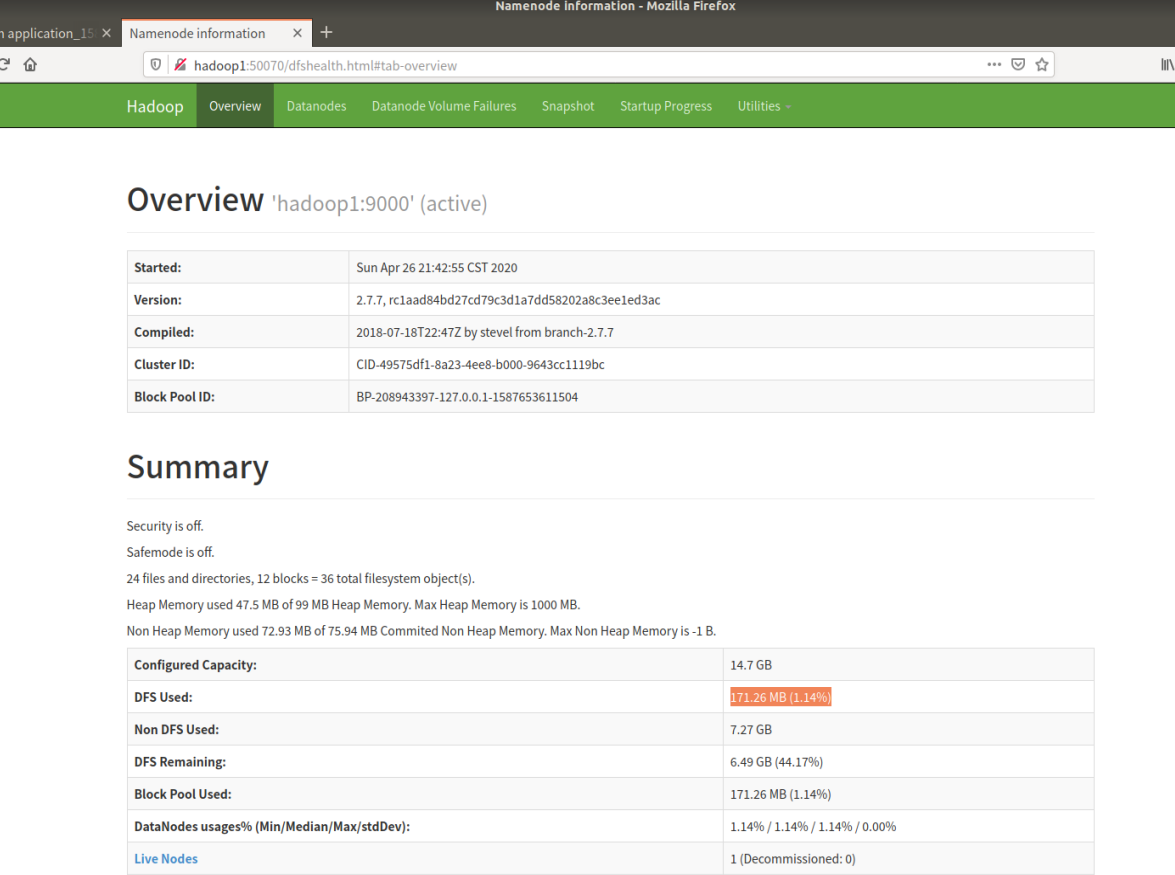
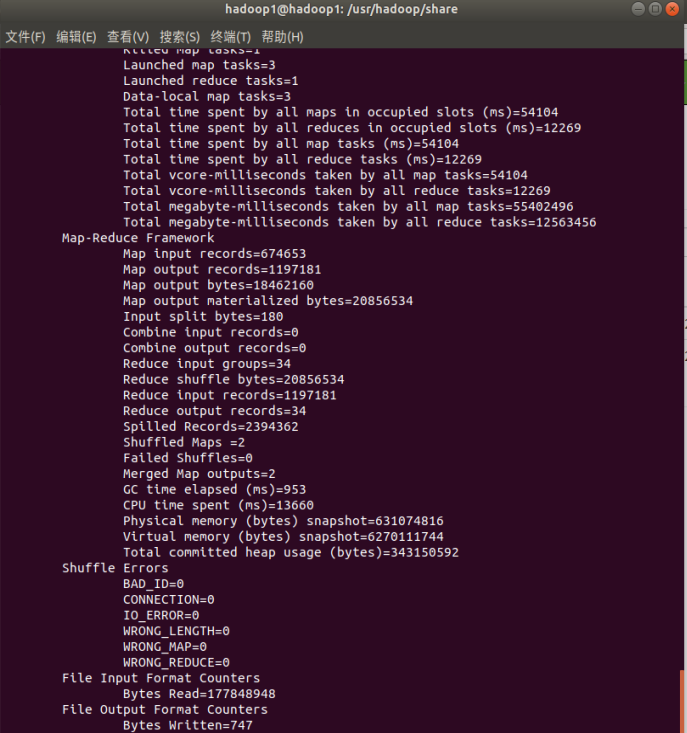
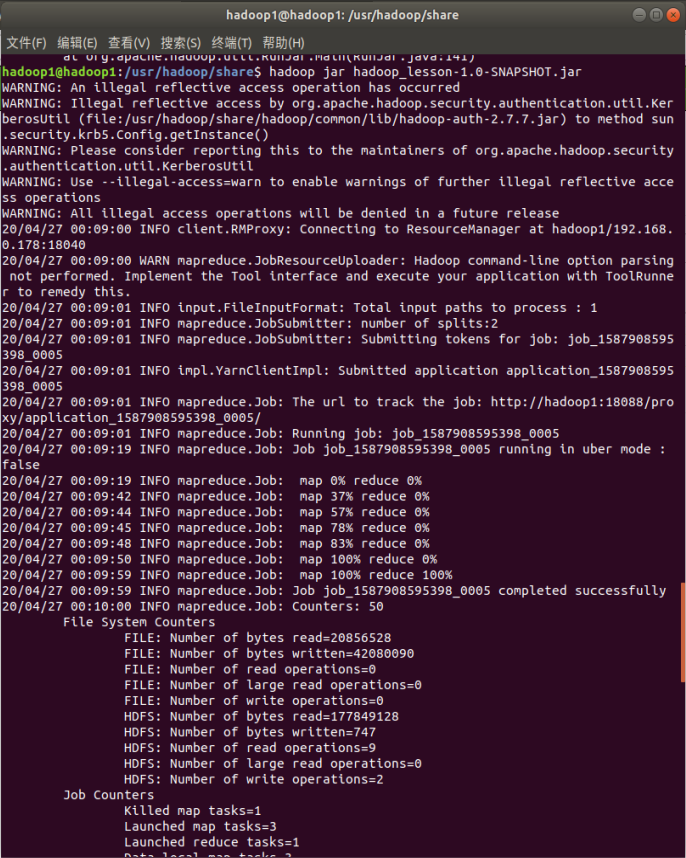
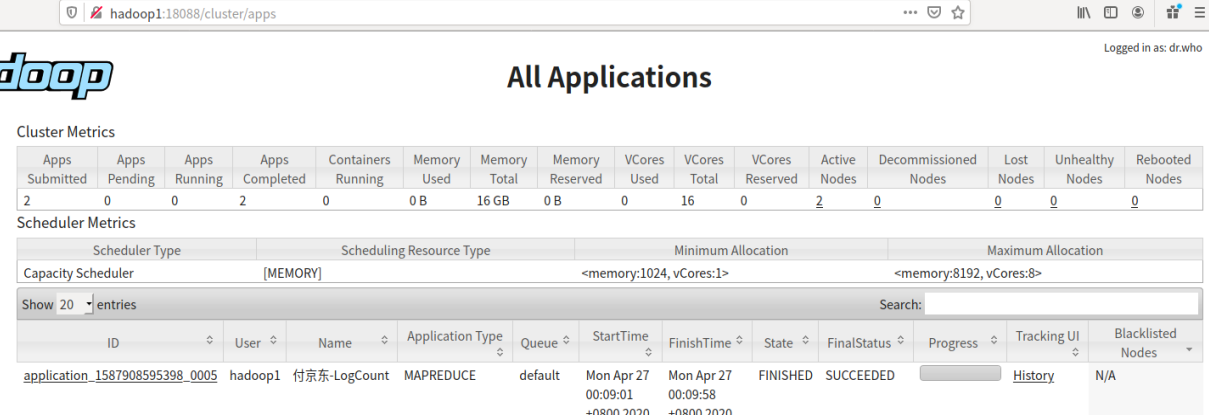
虚拟机数量：3

Jdk版本：9.0.1

Hadoop版本：2.7.7

* 环境搭建流程：

1. 准备上述软件和对应安装包
2. 安装并运行计划用于运行namenode、secondaryname和resourceManager的主虚拟机hadoop1,设置虚拟机网络为桥接模式
3. 修改网络设置为手动获取设置主机内网地址为192.168.0.178
4. 修改本机名称sudo gedit /etc/hostname 编辑为hadoop1
5. 修改网络映射sudo gedit /etc/hosts 增加如下两条（因为只用两台机子）便于使用名称访问
   1. 
6. 检查service iptables status确保防火墙未激活
7. 使用sudo gedit /etc/apt/source.list 将apt默认源换为任一国内源避免被墙
8. 设置该虚拟机的root用户密码
   1. 使用sudo passwd //设置root用户默认密码
9. 解压缩jdk改变其根目录为/usr/jdk，设置其环境变量
   1. sudo gedit /etc/profiles 打开改文件在最后添加如下语句
   2. export JAVA\_HOME=/usr/jdk
   3. export JRE\_HOME=$JAVA\_HOME/jre
   4. export CLASSPATH=.:$JAVA\_HOME/lib:$JRE\_HOME/lib:$CLASSPATH
   5. export PATH=$JAVA\_HOME/bin:$JRE\_HOME/bin:$JAVA\_HOME:$PATH
10. 解压缩hadoop并改变其目录为/usr/hadoop，设置其环境变量
    1. sudo gedit /etc/profiles 打开改文件在最后添加如下语句
    2. export HADOOP\_HOME=/usr/hadoop/
    3. export PATH=$HADOOP\_HOME/bin:$PATH
    4. export HADOOP\_CONF\_DIR=$HADOOP\_HOME/etc/hadoop
11. sudo apt-get install ssh 安装ssh，建立本机ssh无密码登陆
    1. sudo service ssh start //开启ssh服务
    2. ssh-keygen -t rsa -P "" //生成默认空rsa密码，之后会在.ssh目录产生id\_rsa和id\_ras.pub两个文件
    3. cd ~/.ssh //进入.ssh目录
    4. cat id\_rsa.pub >> authorized\_keys //将公钥追加到授权文件中
12. 关闭hadoop1虚拟机复制vdi盘片并使用该盘片创建用于运行datanode的虚拟机hadoop2和hadoop3，根据步骤2、3设置网络桥接模式并且内网地址改为192.168.0.179和192.168.0.179，然后启动hadoop1和hadoop2和hadoop3三台虚拟机
13. 根据步骤4修改hadoop2、hadoop3虚拟机hostname为hadoop2、hadoop3
14. 建立hadoop1和hadoop2和hadoop3之间的ssh无密码登陆
    1. 开启所有机器ssh服务
    2. 使用scp命令转移相互的公钥到.ssh目录
    3. 使用cat命令将对方机器的公钥添加到授权文件
    4. 相互使用ssh登录一次机器
15. 设置在hadoop1的HADOOP\_HOME目录下简历四个文件夹
    1. HADOOP\_HOME/hdfs
    2. HADOOP\_HOME/hdfs/tmp
    3. HADOOP\_HOME/hdfs/name
    4. HADOOP\_HOME/hdfs/data
16. 进入hadoop1的HADOOP\_HOME/etc/hadoop目录进行hadoop设置对应文件
    1. core-site.xml：设置namenode地址和端口以及临时文件目录
    2. hadoop.env和yarn.env中的“export JAVA\_HOME=”的内容换成本机jdk实际安装目录“/usr/jdk”
    3. hdfs-site设置namenode和datanode对应文件夹、secondaryname地址端口等：
    4. mapred-site.xml.templete 去掉.templete后缀，然后编辑增加内容指定mapreduce框架为yarn：
    5. slaves文件内编辑hadoop2，使hadoop2位slaves
    6. yarn-site.xml编辑以下内容：主要用以设置resources对应模块接口位置
17. 将HADOOP\_HOME/etc/hadoop配置文件夹使用scp迁移到hadoop2对应目录位置上覆盖该机器的原配置文件夹
18. 确保所有机器当前使用的标准用户对HADOOP\_HOME的目录和子目录有读写权限，否则需要使用chmod为当前系统标准用户增加读写权限
    1. chmod -R 777 /{dir} //-R表示所有子目录，777表示所有用户有读写和执行权限
19. hadoop1和hadoop2、3上使用hadoop namenode -format初始化hadoop
    1. 如果因为权限初始化失败可以使用root用户在HADOOP\_HOME/bin/hdfs使用该命令并重新获取HADOOP\_HOME下所有子目录权限
20. 在hadoop1的HADOOP\_HOME/sbin目录下使用./start-all.sh启动hadoop
21. 启动成功后可以看到：
    1. hadoop1上使用jps命令查看该机器上相关进程有secondnarynamenode，namenode，jps，resourcemanager：
    2. hadoop2和3上使用jps命令查看该机器上相关进程有datanode，jps，nodemanager：
    3. 进入resourcemanager的webapp的页面查看运行无误：
22. 运行hadoop内置的求圆周率的样例，pi为类名，第一个10位map次数，第二个10位随机点生成次数：
    1. hadoop jar /usr/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.7.jar pi 10 10：
    2. resourcemanager：
    3. 结果为3.200000000000000000，可见运行正常

* 编写测试样例
  + 有一个邮件日志数据集：
  + 总共674653条日志包含日志格式如下：
  + 需求：统计每月的日志条数、各月的这四种类型（数据集实际不止这四种）的条数分别是多少
  + 编写mapreduce程序：
    - 新建maven项目
    - 导入hadoop的各种依赖：
    - 编写mapper：
    - 编写reducer：
    - 主函数写任务配置：
    - maven打包配置：
    - 项目打成jar包，传到hadoop1上
    - 数据集文件maillog穿到hadoop1上，并使用hdfs -put命令将maillog数据文件加入到hdfs中，查看hdfs状况：
    - 使用hadoop jar命令运行logCount程序：
    - 查看运行状态：
    - 查看运行结果：