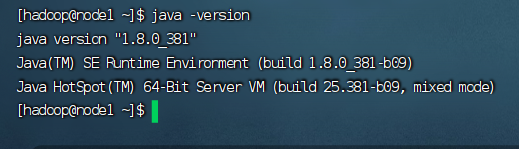
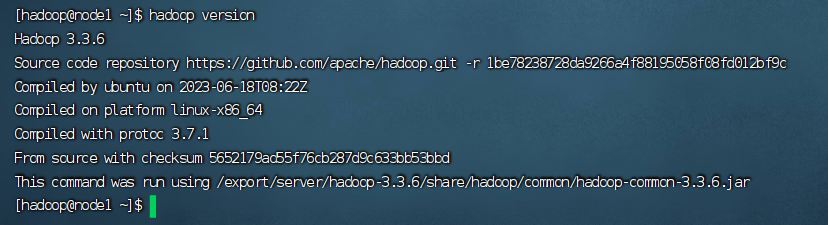
## 实验一：hadoop安装与配置实验

这一部分可以详细参考视频：<https://www.bilibili.com/video/BV1WY4y197g7/?spm_id_from=333.999.0.0&vd_source=cceb6047d20984cb221e76e98c4a5567>

采用的是在VM虚拟机上搭建CentOS，进而配置java jdk1.8.0以及hadoop3.3.6版本，同样配置了3个结点虚拟机，作为后续hdfs、yarn以及hive等内容。



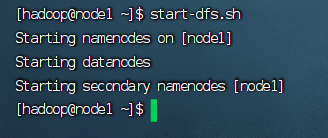


## 实验二：hdfs的配置、启动以及使用

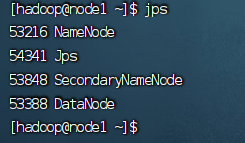
配置这一部分同样可以参照上面的视频，<https://www.bilibili.com/video/BV1WY4y197g7/?spm_id_from=333.999.0.0&vd_source=cceb6047d20984cb221e76e98c4a5567>

**实验配置、启停：**

1.配置完毕后，使用start-dfs.sh来启动hdfs：

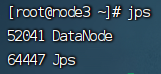
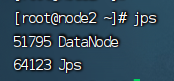


2.使用jps来查看进程信息：



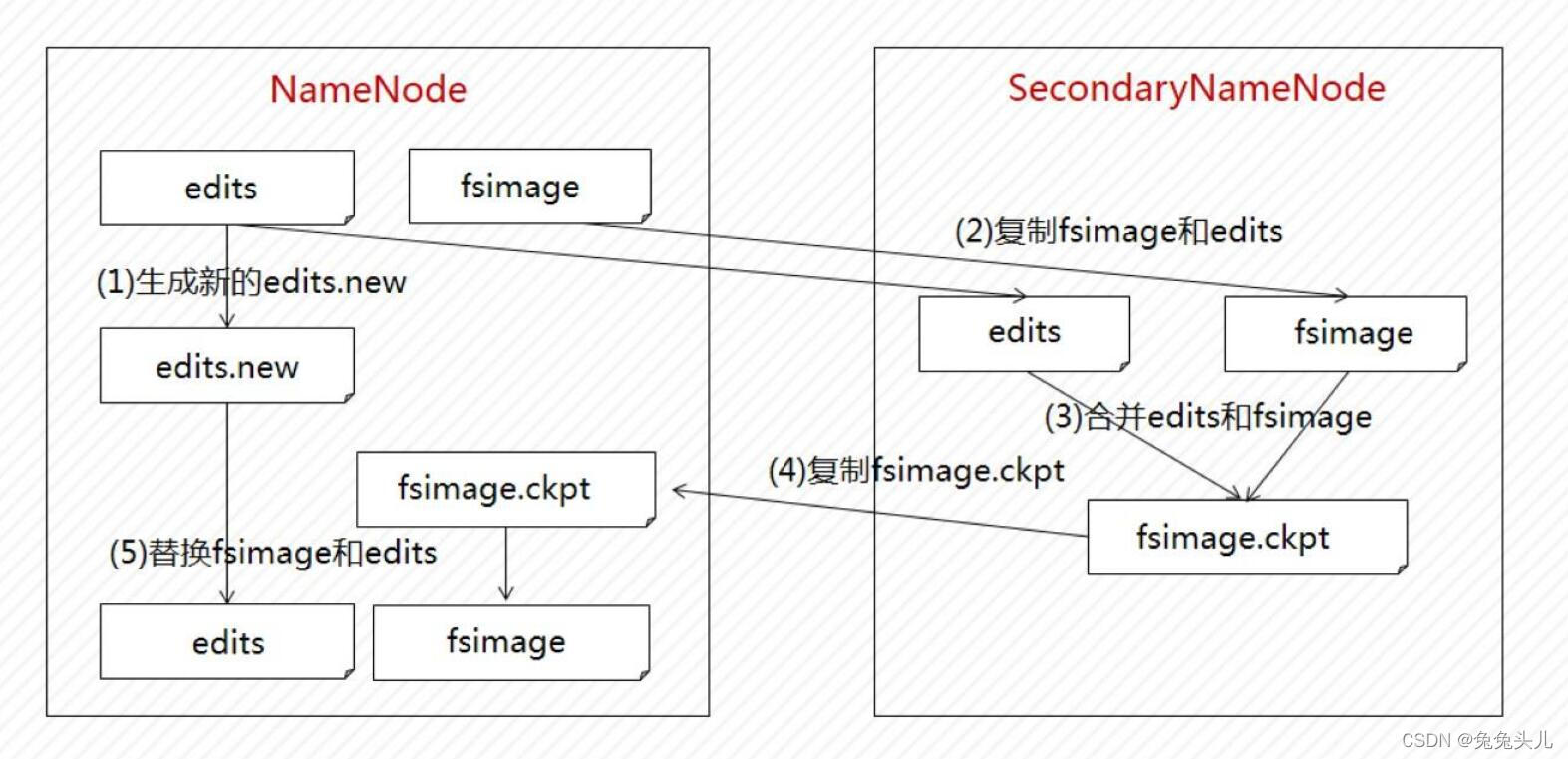
发现有NameNode、SecondaryNameNode以及DataNode。

通过在每个结点使用jps，还可以发现，node2和node3结点都具有DataNode进程。



NameNode负责管理维护hdfs，并且要接收客户端请求上传、下载、创建目录等，维护了fsimage和edits文件（操作日志文件）。

SecondaryNameNode负责定期将fsimage和edits下载到本地，最后将合并后的上传回NameNode。

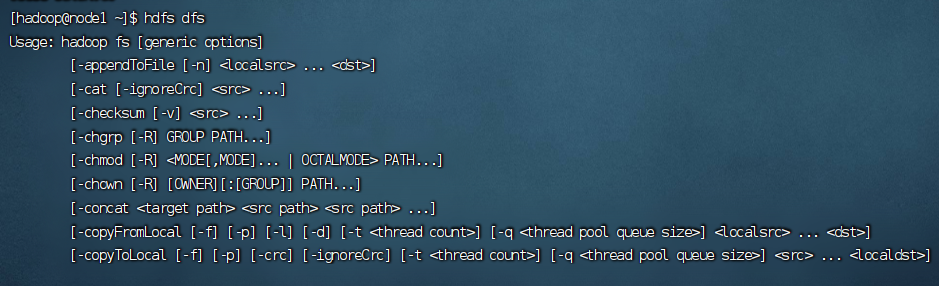


DataNode负责存储数据块，负责客户端对数据块的IO请求，DataNode定时和NameNode进行通信来接受指令。

**实验内容：**

这一部分同样可以参考上述的教程中内容。

使用hdfs dfs指令来查看可以使用的指令：



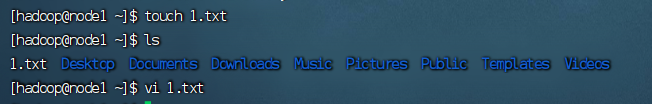
在某个路径下创建文件夹hdfs dfs -mkdir [-p] <path>

上传文件到hdfs指定目录下hdfs dfs -put [-f] [-p] <localsrc> -f覆盖目标文件，-p保留访问和修改时间、所有权和权限 localsrc本地文件系统

下载hdfs文件 hadoop fs -get [-f] [-p] <stc> <localdst>

**1.向HDFS中上传任意文本文件，如果指定的文件在HDFS中已经存在，由用户指定是追加到原有文件末尾还是覆盖原有的文件。**

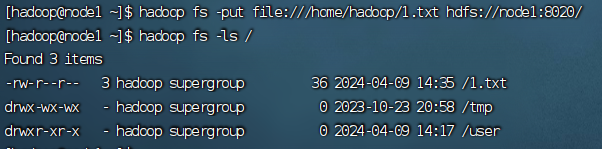
首先使用touch 1.txt来作为上传的：（使用cat、vim）



使用指令：hadoop fs -put file:///home/hadoop/1.txt hdfs://node1:8020/

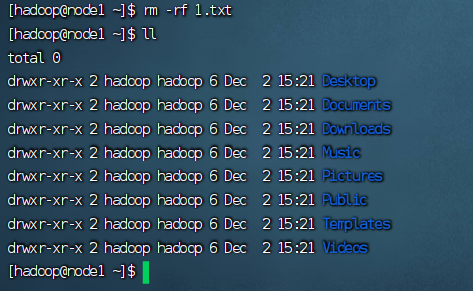
将我们刚才创建的1.txt上传到hdfs节点中。

再使用hadoop fs -ls / 即可查看是否上传成功：

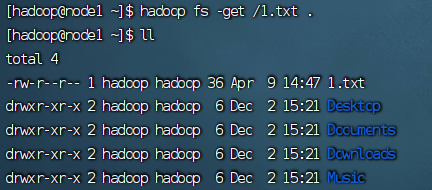


**2. 从HDFS中下载指定文件，如果本地文件与要下载的文件名称相同，则自动对下载的文件重命名**

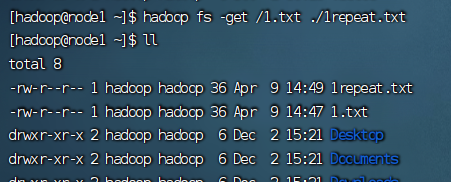
首先在本地删除掉1.txt文件：rm -rf 1.txt



然后使用hdfs dfs -get /1.txt . 命令来下载hdfs节点的文件：

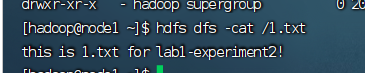


假设需要重复下载可以直接重命名，hdfs dfs -get /1.txt ./1repeat.txt，再使用ll来观察是否重命名成功，可见下图：



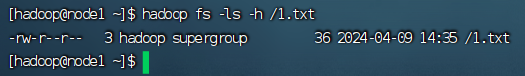
**3. 将HDFS中指定文件的内容输出到终端中**

使用-cat来读取文件所有内容，hdfs dfs -cat /1.txt。



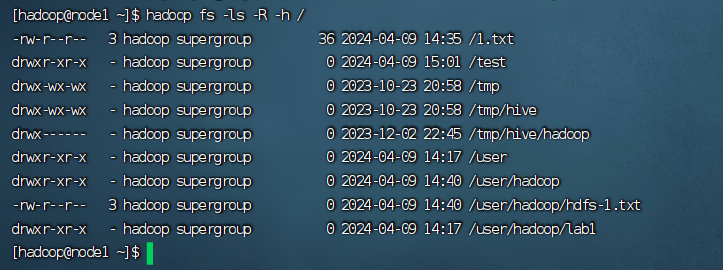
**4. 显示HDFS中指定的文件的读写权限、大小、创建时间、路径等信息**

使用指令hadoop fs -ls -h /1.txt，即可，如下图所示。



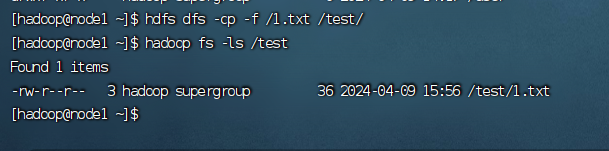
**5. 给定HDFS中某一个目录，输出该目录下的所有文件的读写权限、大小、创建时间、路径等信息，如果该文件是目录，则递归输出该目录下所有文件相关信息；**

使用hadoop fs -ls -R -h/

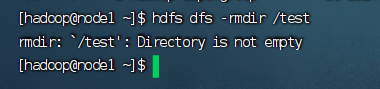


**7. 提供一个HDFS的目录的路径，对该目录进行创建和删除操作。创建目录时，如果目录文件所在目录不存在则自动创建相应目录；删除目录时，由用户指定当该目录不为空时是否还删除该目录；**

创建了一个test目录用来保存txt文件，选择删除当前文件：

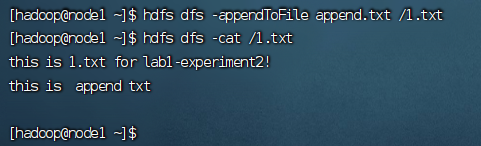


输入hdfs dfs -rmdir /test指令可见如下所示：



**8. 向HDFS中指定的文件追加内容，由用户指定内容追加到原有文件的开头或结尾；**

首先在本地创建一个append.txt文件用于将内容追加到文件结尾。



追加到文件开头：

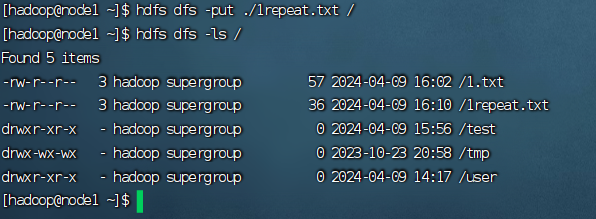
hdfs dfs -get text.txt

cat text.txt >> local.txt

hdfs dfs -copyFromLocal -f text.txt text.txt

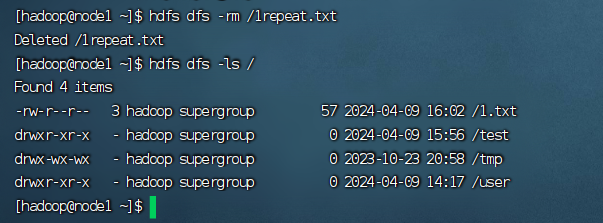
**9. 删除HDFS中指定的文件**

先将之前的1repeat.txt文件上传到hdfs中间：



现在执行命令删除这个文件：

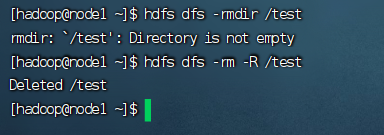
hdfs dfs -rm /1repeat.txt



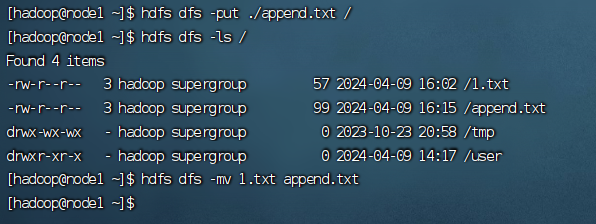
**10. 删除HDFS中指定的目录，由用户指定目录中如果存在文件时是否删除目录；**

hdfs dfs -rmdir /test 如果这个文件夹内有内容，会告知用户

hdfs dfs -rm -R /test 使用-R强行删除这个文件夹



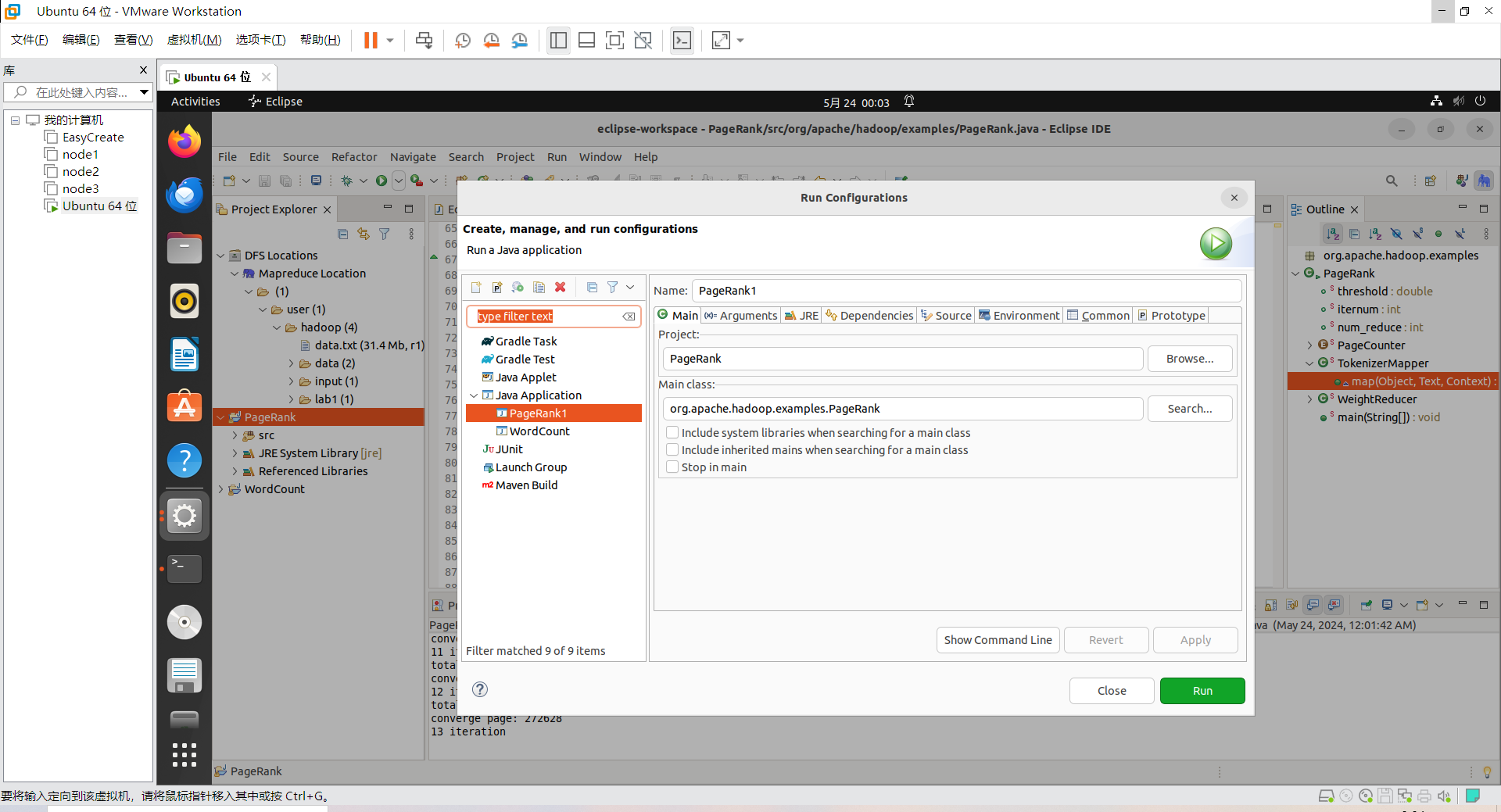
**11. 在HDFS中，将文件从源路径移动到目的路径**

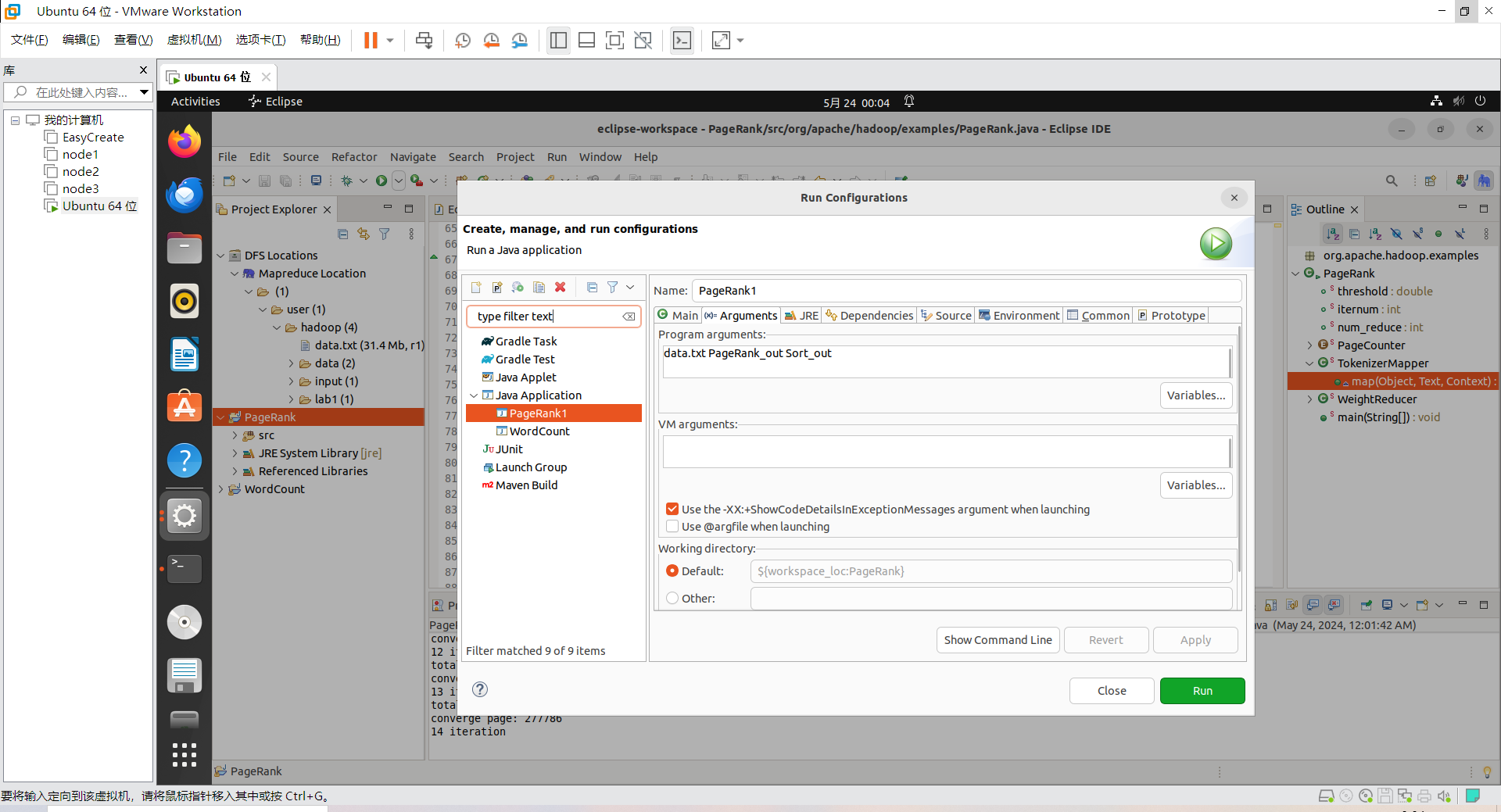


## 实验三：WordCount实验

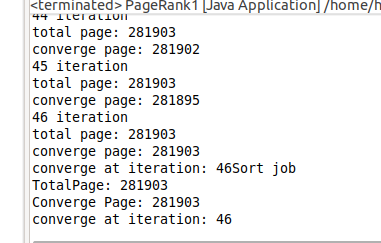
## 实验四：PageRank实验

在eclipse中编写pagerank的java文件，然后对左侧文件栏中的PageRank项目右键run as中的run configuration将这个添加成一个java application，并且修改main中的内容如下所示。





运行完毕：



保存在了/eclipse-workspace/PageRank中的PageRank\_out和Sort\_out文件夹中

