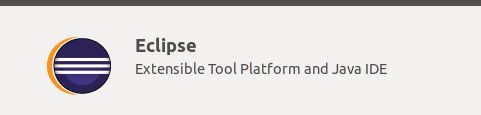
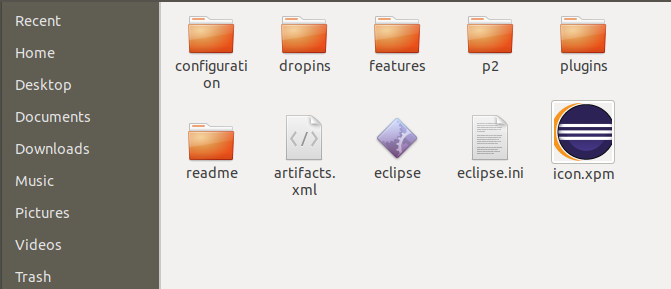
1.环境配置

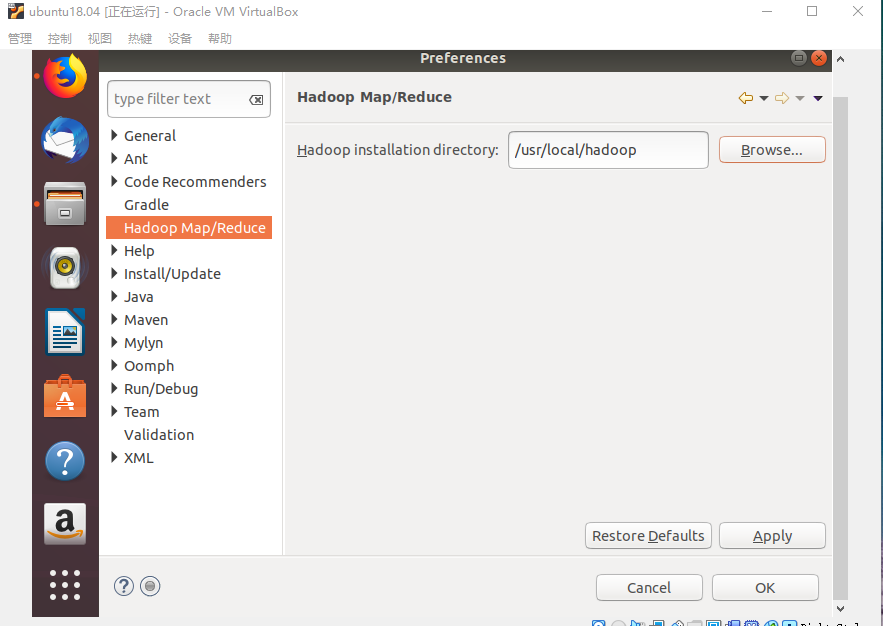
在软件中心找到eclipse，并安装

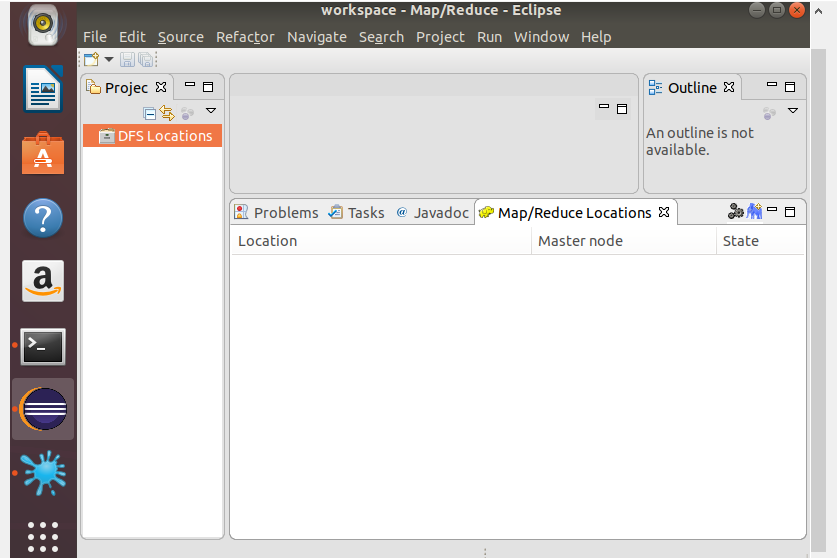


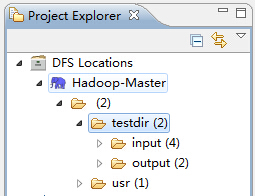
Eclipse安装路径/usr/lib/，并将对应的 将hadoop-eclipse-kepler-plugin-2.10.1.jar（我自己用ant编译成功hadoop-eclipse-kepler-plugin-2.10.1.jar）放到eclipse解压文件夹的plugins文件夹下，以便顺利启动。

开启hadoop，并上传文件：





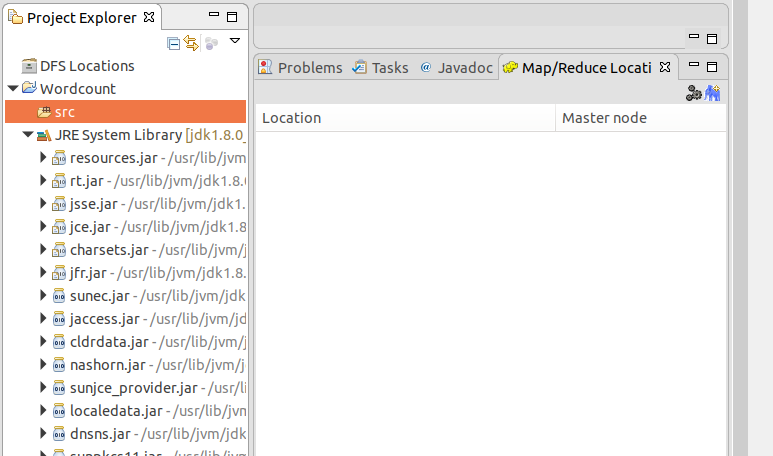


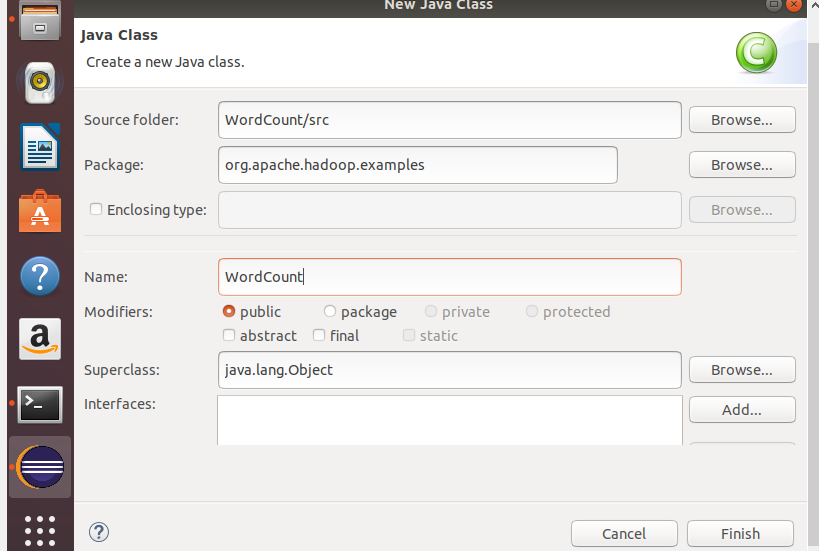


配置完成即可查看、操作文件，并且可以完成下面的编程执行工作。

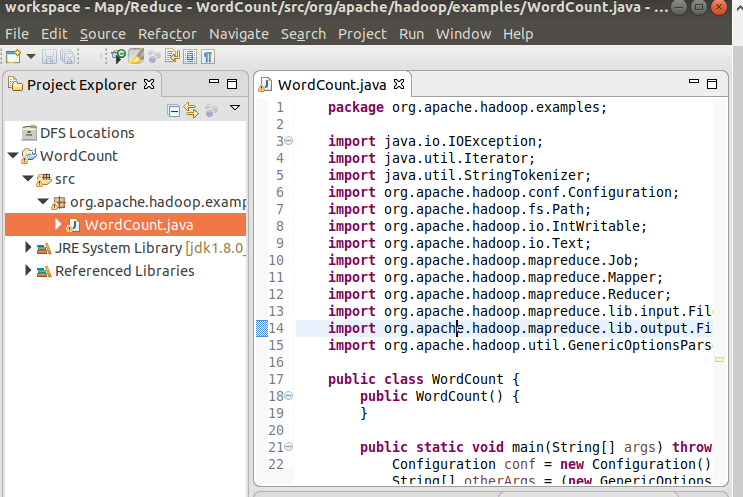
使用eclipse执行wordcount：

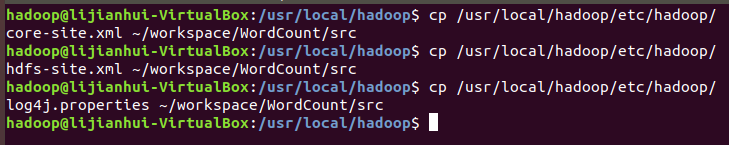
创建wordcount项目：



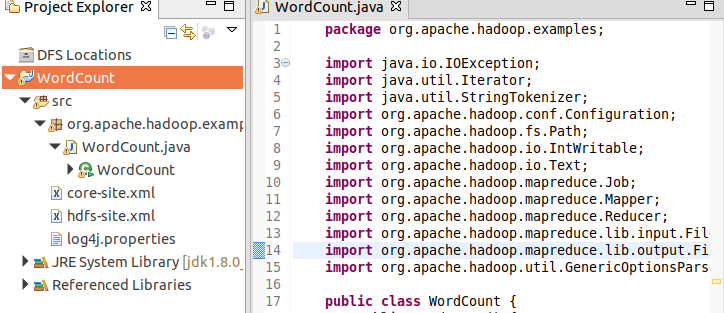


将wordcount代码复制到java文件中：

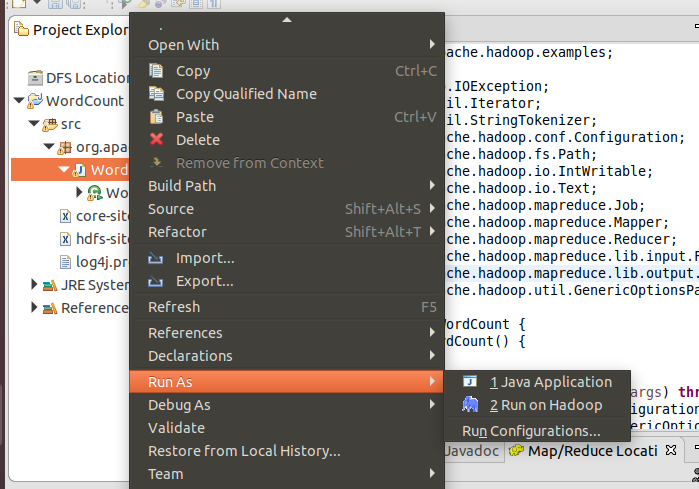




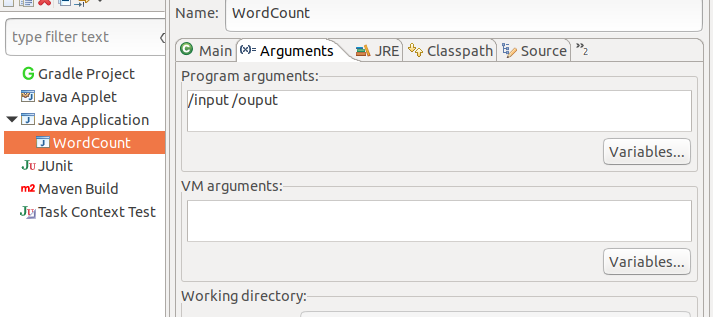
刷新后：



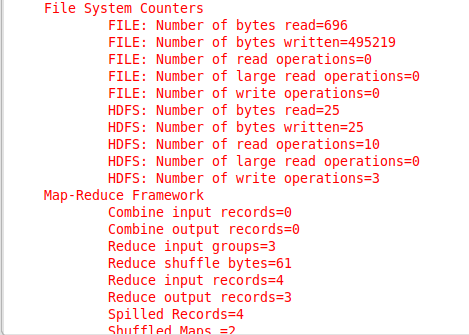
在Hadoop上运行

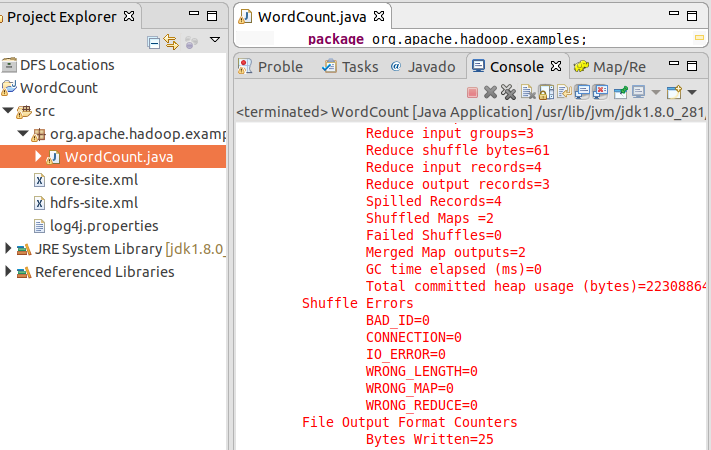


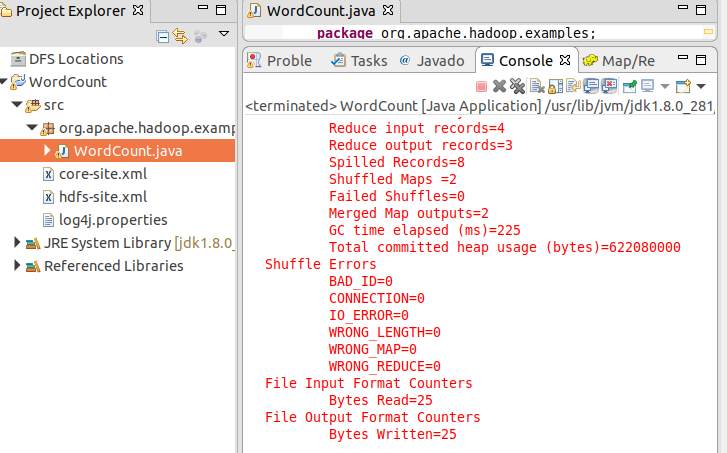
设置变量（要加斜杠，否则会报错）：



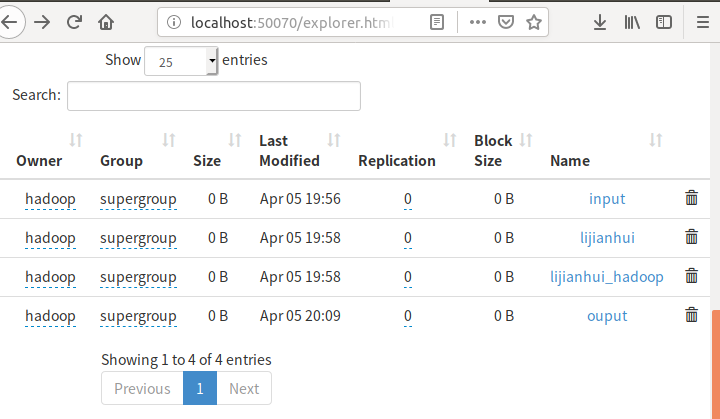
执行成功：



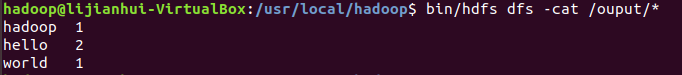




并且在web端可以看到出现了/output,成功：

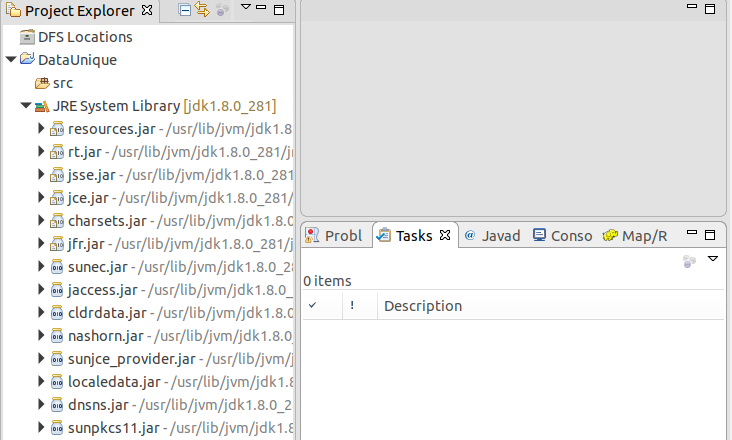


可以得到我们的词数：

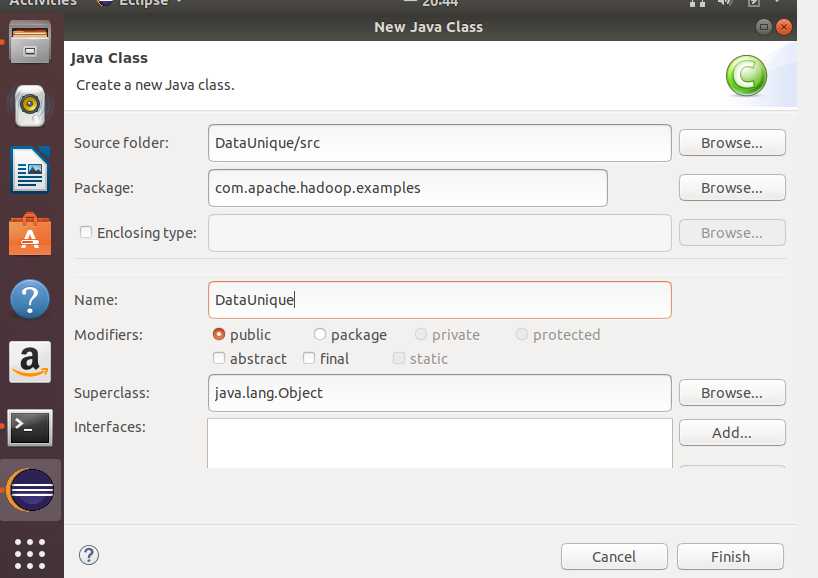


1. 数据去重：

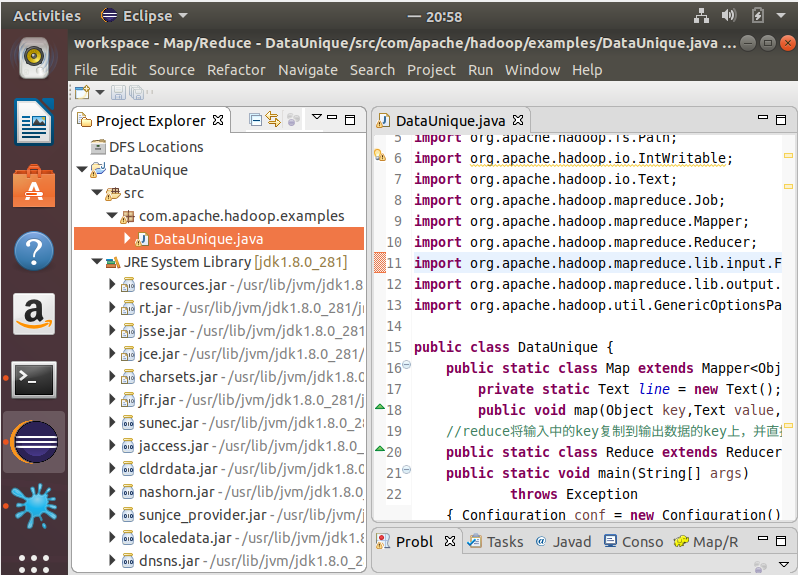
首先，建立项目：



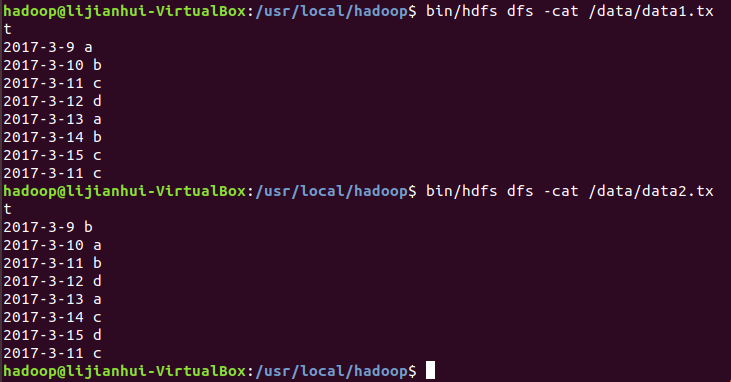
创建类：



写入程序：



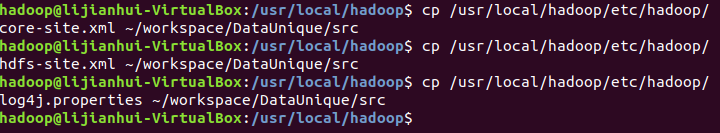
按照要求写入数据（此处查看hdfs中的data文件）：



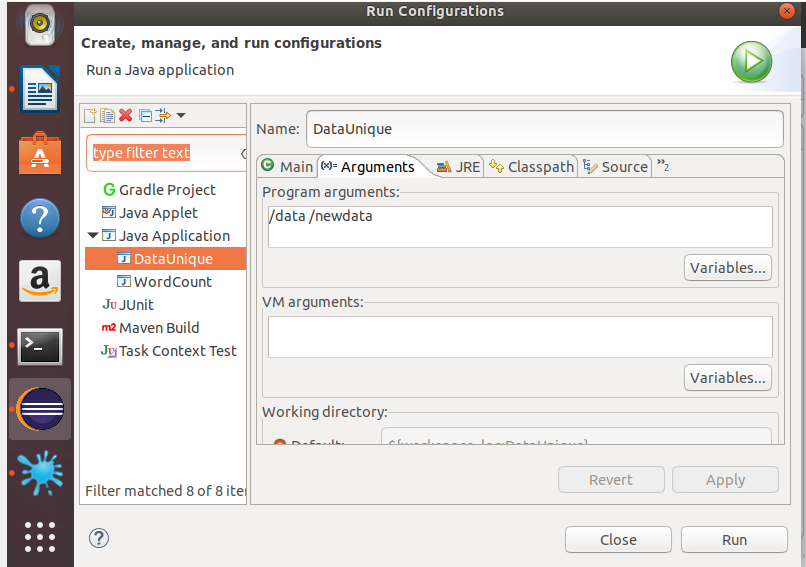
上传文件：



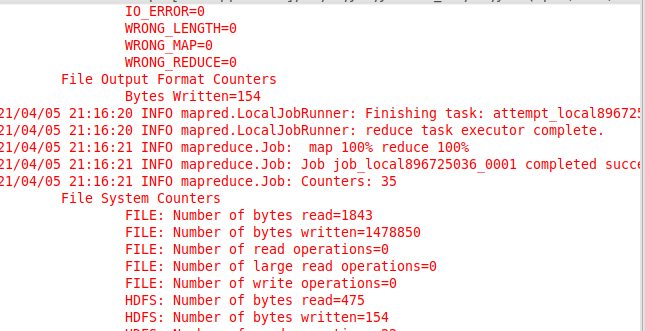
配置项目：

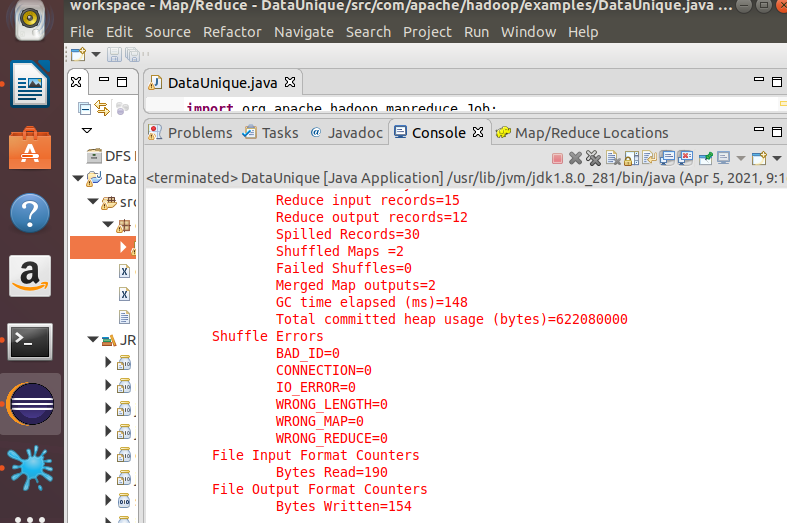


配置参数：

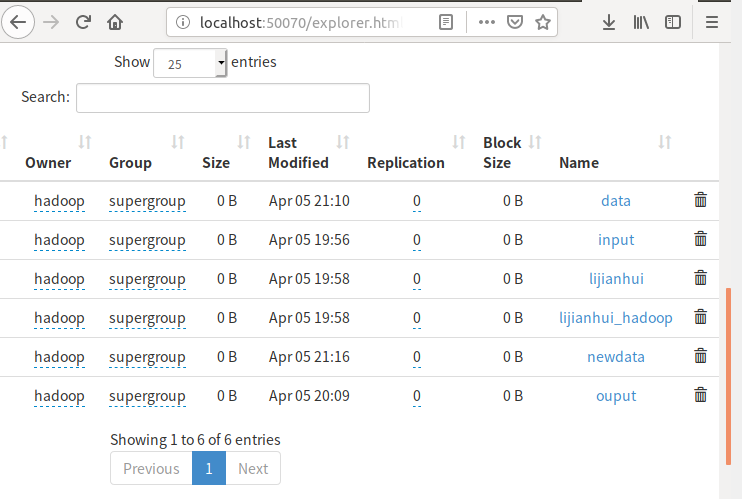


运行成功：

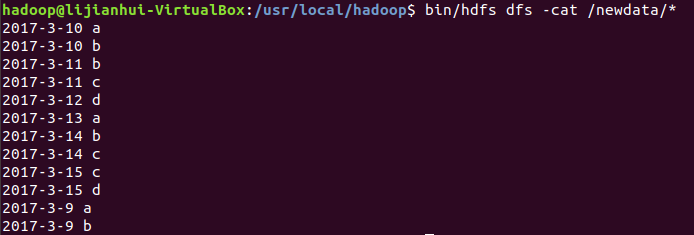




并且在后台管理系统中出现了newdata，顺利完成：



并且我们可以看到输出结果，正如预期：



源代码：

package com.apache.hadoop.examples;

import java.io.IOException;

import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.Mapper;

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

import org.apache.hadoop.util.GenericOptionsParser;

public class DataUnique {

//map将输入中的value复制到输出数据的key上，并直接输出

public static class Map extends Mapper<Object,Text,Text,Text>{

private static Text line = new Text();

public void map(Object key,Text value,Context context) throws IOException,InterruptedException{

line = value;

context.write(line, new Text(""));

}

}

//reduce将输入中的key复制到输出数据的key上，并直接输出

public static class Reduce extends Reducer<Text,Text,Text,Text>{

public void reduce(Text key,Iterable<Text> values,Context context) throws IOException,InterruptedException{

context.write(key, new Text(""));

}

}

/\*\*

\* @param args

\*/

public static void main(String[] args) throws Exception{

// TODO Auto-generated method stub

Configuration conf = new Configuration();

String[] otherArgs = new GenericOptionsParser(conf,args).getRemainingArgs();

if(otherArgs.length != 2){

System.err.println("Usage WordCount <int> <out>");

System.exit(2);

}

Job job = new Job(conf,"DataUnique");

job.setJarByClass(DataUnique.class);

job.setMapperClass(Map.class);

job.setCombinerClass(Reduce.class);

job.setReducerClass(Reduce.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(Text.class);

FileInputFormat.addInputPath(job, new Path(otherArgs[0]));

FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}