# 并行计算 上机报告

上机题目：

1. 按照Hadoop安装运行说明文档中的指导，自己搭建伪分布式Hadoop环境，熟悉HDFS的常用操作(参考 **Hdoop实战**第31-36页)，运行WordCount程序，得到统计结果。请详细写出你每一步的操作,最好有截图,最后的结果部分必须有截图。
2. 实现一个统计输入文件中各个长度的单词出现频次的程序。

姓名： 张劲暾

学号： PB16111485

日期： 2019年5月25日

实验环境：

> **CPU：** *Intel® Core™ i7-6500U CPU @ 2.50GHz × 4*
>
> **内存：** *7.7 GiB*
>
> **操作系统：** *Ubuntu 19.04 64bit*
>
> **软件平台：**
>
> 1. *Hadoop 2.7.6*
> 2. *openjdk version "1.8.0_212"*
>
> ~~~~
> ```
>  *OpenJDK Runtime Environment (build 1.8.0_212-8u212-b03-0ubuntu1.19.04.2-b03)*
>
> ```*OpenJDK 64-Bit Server VM (build 25.212-b03, mixed mode)*
> ~~~~

目录

## 算法设计与分析

### 题目一

按照Hadoop安装运行说明文档中的指导，自己搭建伪分布式Hadoop环境，熟悉HDFS的常用操作(参考 Hdoop实战 第31-36页)，运行WordCount程序，得到统计结果。请详细写出你每一步的操作,最好有截图,最后的结果部分必须有截图。

实验步骤：

#### 创建Hadoop用户并设置密码

~~~
``` bash
$sudo useradd -m hadoop -s /bin/bash
```$sudo passwd hadoop
~~~

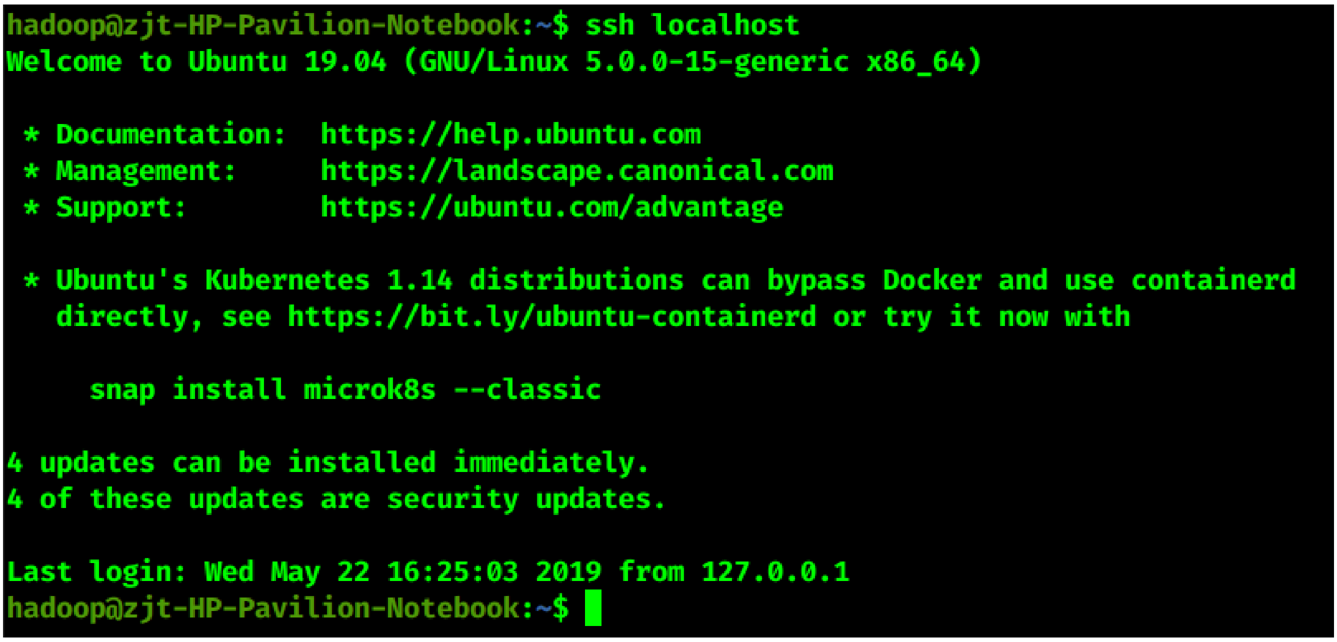
#### 为Hadoop用户增加管理员权限

~~~
``` bash
```$sudo adduser hadoop sudo
~~~

#### 切换到Hadoop用户下

#### 安装SSH、配置SSH无密码登陆

~~~
``` bash
$sudo apt-get update
$sudo apt-get install openssh-server
cd ~/.ssh/                      # 若没有该目录，请先执行一次ssh localhost
ssh-keygen -t rsa               # 会有提示，都按回车就可以
cat ./id_rsa.pub >> ./authorized_keys  #
```ssh localhost
~~~



```
hadoop@zjt-HP-Pavilion-Notebook:~$ ssh localhost
Welcome to Ubuntu 19.04 (GNU/Linux 5.0.0-15-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Ubuntu's Kubernetes 1.14 distributions can bypass Docker and use containerd
   directly, see https://bit.ly/ubuntu-containerd or try it now with

     snap install microk8s --classic

4 updates can be installed immediately.
4 of these updates are security updates.

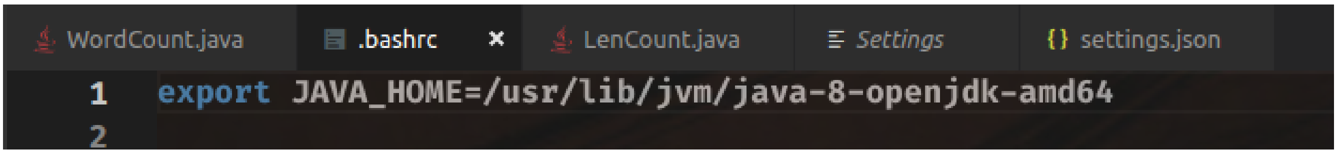
Last login: Wed May 22 16:25:03 2019 from 127.0.0.1
hadoop@zjt-HP-Pavilion-Notebook:~$
```

#### 安装JAVA环境

~~~
``` bash
$ $JAVA_HOME
bash: /usr/lib/jvm/java-8-openjdk-amd64: 是一个目录
```gedit ~/.bashrc
~~~

在文件最前面添加如下单独一行

~~~
``` bash
```export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
~~~



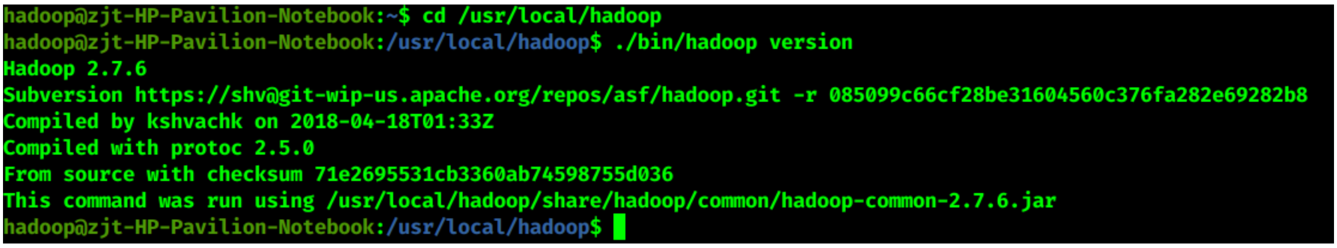
```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

~~~
``` bash
```$ ```source ~/.bashrc    # 使变量设置生效
~~~

#### 安装Hadoop

1. 下载安装包
2. 解压到安装路径

~~~
``` bash
    $ sudo tar -zxf ~/下载/hadoop-2.7.6.tar.gz -C /usr/local    # 解压到/usr/local中
    $ cd /usr/local/
    $ sudo mv ./hadoop-2.7.4/ ./hadoop                  # 将文件夹名改为hadoop
    $ sudo chown -R hadoop ./hadoop                           # 修改文件权限
    $ cd /usr/local/hadoop                                         # 进入Hadoop 目录下
```    $ ./bin/hadoop version                                        # 在该目录下执行该命令
~~~



```
hadoop@zjt-HP-Pavilion-Notebook:~$ cd /usr/local/hadoop
hadoop@zjt-HP-Pavilion-Notebook:/usr/local/hadoop$ ./bin/hadoop version
Hadoop 2.7.6
Subversion https://shv@git-wip-us.apache.org/repos/asf/hadoop.git -r 085099c66cf28be31604560c376fa282e69282b8
Compiled by kshvachk on 2018-04-18T01:33Z
Compiled with protoc 2.5.0
From source with checksum 71e2695531cb3360ab74598755d036
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-2.7.6.jar
hadoop@zjt-HP-Pavilion-Notebook:/usr/local/hadoop$ 
```

#### Hadoop伪分布式配置

/usr/local/hadoop/etc/hadoop/core-site.xml 设置：

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

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

<configuration>
        <property>
             <name>hadoop.tmp.dir</name>
             <value>file:/usr/local/hadoop/tmp</value>
             <description>Abase for other temporary directories.</description>
        </property>
        <property>
             <name>fs.defaultFS</name>
             <value>hdfs://localhost:9000</value>
        </property>
</configuration>
```

/usr/local/hadoop/etc/hadoop/hdfs-site.xml 设置：

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
```

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

<configuration>
        <property>
             <name>dfs.replication</name>
             <value>1</value>
        </property>
        <property>
             <name>dfs.namenode.name.dir</name>
             <value>file:/usr/local/hadoop/tmp/dfs/name</value>
        </property>
        <property>
             <name>dfs.datanode.data.dir</name>
             <value>file:/usr/local/hadoop/tmp/dfs/data</value>
        </property>
</configuration>
```

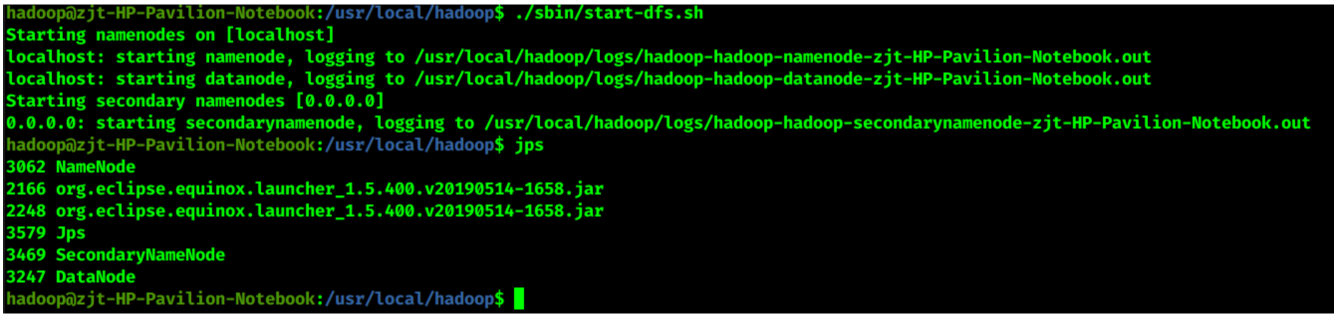
执行 NameNode的格式化：

```
/usr/local/hadoop$./bin/hdfs namenode -format
```

开启 NameNode 和 DataNode 守护进程

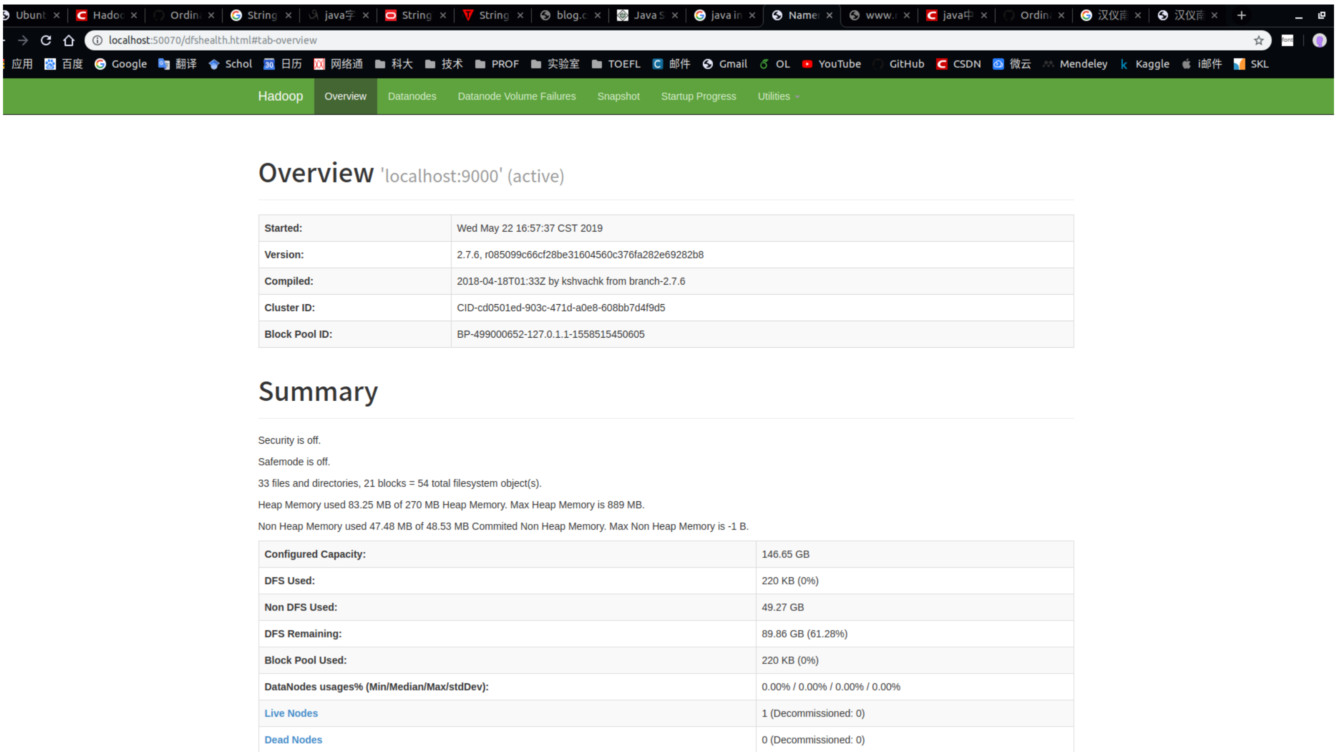
```
/usr/local/hadoop$./sbin/start-dfs.sh
```

通过命令 jps 来判断是否成功启动



```
hadoop@zjt-HP-Pavilion-Notebook:/usr/local/hadoop$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-zjt-HP-Pavilion-Notebook.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-datanode-zjt-HP-Pavilion-Notebook.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-secondarynamenode-zjt-HP-Pavilion-Notebook.out
hadoop@zjt-HP-Pavilion-Notebook:/usr/local/hadoop$ jps
3062 NameNode
2166 org.eclipse.equinox.launcher_1.5.400.v20190514-1658.jar
2248 org.eclipse.equinox.launcher_1.5.400.v20190514-1658.jar
3579 Jps
3469 SecondaryNameNode
3247 DataNode
hadoop@zjt-HP-Pavilion-Notebook:/usr/local/hadoop$ 
```

访问 Web 界面 http://localhost:50070 查看 NameNode 和 Datanode 信息



localhost:50070/dfshealth.html#tab-overview

Overview 'localhost:9000' (active)

Started:	Wed May 22 16:57:37 CST 2019
Version:	2.7.6, r085099c66cf28be31604560c376fa282e69282b8
Compiled:	2018-04-18T01:33Z by kshvachk from branch-2.7.6
Cluster ID:	CID-cd0501ed-903c-471d-a0e8-608bb7d4f9d5
Block Pool ID:	BP-499000652-127.0.1.1-1558515450605

Summary

Security is off.

Safemode is off.

33 files and directories, 21 blocks = 54 total filesystem object(s).

Heap Memory used 83.25 MB of 270 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 47.48 MB of 48.53 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	146.65 GB
DFS Used:	220 KB (0%)
Non DFS Used:	49.27 GB
DFS Remaining:	89.86 GB (61.28%)
Block Pool Used:	220 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)

#### 在HDFS上创建目录上传输入文件

```
/usr/local/hadoop$ ./bin/hdfs dfs -mkdir input
/usr/local/hadoop$ ./bin/hdfs dfs -put ./etc/hadoop/*.xml input
/usr/local/hadoop$ ./bin/hdfs dfs -ls input
Found 8 items
-rw-r--r--   1 hadoop supergroup       4436 2019-05-22 17:01 input/capacity-scheduler.xml
-rw-r--r--   1 hadoop supergroup       1116 2019-05-22 17:01 input/core-site.xml
-rw-r--r--   1 hadoop supergroup       9683 2019-05-22 17:01 input/hadoop-policy.xml
-rw-r--r--   1 hadoop supergroup       1188 2019-05-22 17:01 input/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup        620 2019-05-22 17:01 input/httpfs-site.xml
-rw-r--r--   1 hadoop supergroup       3518 2019-05-22 17:01 input/kms-acls.xml
-rw-r--r--   1 hadoop supergroup       5540 2019-05-22 17:01 input/kms-site.xml
-rw-r--r--   1 hadoop supergroup        690 2019-05-22 17:01 input/yarn-site.xml
/usr/local/hadoop$ ./bin/hadoop fs -ls -R
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:01 input
-rw-r--r--   1 hadoop supergroup       4436 2019-05-22 17:01 input/capacity-scheduler.xml
-rw-r--r--   1 hadoop supergroup       1116 2019-05-22 17:01 input/core-site.xml
-rw-r--r--   1 hadoop supergroup       9683 2019-05-22 17:01 input/hadoop-policy.xml
-rw-r--r--   1 hadoop supergroup       1188 2019-05-22 17:01 input/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup        620 2019-05-22 17:01 input/httpfs-site.xml
-rw-r--r--   1 hadoop supergroup       3518 2019-05-22 17:01 input/kms-acls.xml
-rw-r--r--   1 hadoop supergroup       5540 2019-05-22 17:01 input/kms-site.xml
-rw-r--r--   1 hadoop supergroup        690 2019-05-22 17:01 input/yarn-site.xml
/usr/local/hadoop$ ./bin/hadoop fs -mkdir wordcount/input
/usr/local/hadoop$ ./bin/hadoop fs -put ~/ParallelComputingAlgorithm/MapReduce/input3.txt  wordcount/input
/usr/local/hadoop$ ./bin/hadoop fs -put ~/ParallelComputingAlgorithm/MapReduce/input2.txt  wordcount/input
/usr/local/hadoop$ ./bin/hadoop fs -put ~/ParallelComputingAlgorithm/MapReduce/input1.txt  wordcount/input
/usr/local/hadoop$ ./bin/hadoop fs -ls -R
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:35 input
-rw-r--r--   1 hadoop supergroup       4436 2019-05-22 17:01 input/capacity-scheduler.xml
-rw-r--r--   1 hadoop supergroup       1116 2019-05-22 17:01 input/core-site.xml
-rw-r--r--   1 hadoop supergroup       9683 2019-05-22 17:01 input/hadoop-policy.xml
-rw-r--r--   1 hadoop supergroup       1188 2019-05-22 17:01 input/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup        620 2019-05-22 17:01 input/httpfs-site.xml
-rw-r--r--   1 hadoop supergroup       3518 2019-05-22 17:01 input/kms-acls.xml
```

```
-rw-r--r--   1 hadoop supergroup       5540 2019-05-22 17:01 input/kms-site.xml
-rw-r--r--   1 hadoop supergroup        690 2019-05-22 17:01 input/yarn-site.xml
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:36 wordcount
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:36 wordcount/input
-rw-r--r--   1 hadoop supergroup        464 2019-05-22 17:36 wordcount/input/input1.txt
-rw-r--r--   1 hadoop supergroup        511 2019-05-22 17:36 wordcount/input/input2.txt
-rw-r--r--   1 hadoop supergroup        643 2019-05-22 17:36 wordcount/input/input3.txt
```

#### WordCount.java程序解析

~~~
``` java
 import org.apache.hadoop.fs.FileSystem;```java
 import org.apache.hadoop.fs.FileSystem;
 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;

 import java.io.IOException;
 import java.util.StringTokenizer;

 public class WordCount {
     /***************************
     * MapReduceBase类:实现了Mapper和Reducer接口的基类（其中的方法只是实现接口，而未作任何事情）
     * Mapper接口：
     * WritableComparable接口：实现WritableComparable的类可以相互比较。所有被用作key的类应该实现此接口。
     * Reporter 则可用于报告整个应用的运行进度，本例中未使用。
     ***************************/
     public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>{
         /***************************
         * LongWritable, IntWritable, Text 均是 Hadoop 中实现的用于封装 Java 数据类型的类，这些类实现了
 WritableComparable接口，
         * 都能够被串行化从而便于在分布式环境中进行数据交换，你可以将它们分别视为long,int,String 的替代品。
         ***************************/
         private final static IntWritable one = new IntWritable(1);
         private Text word = new Text();//Text 实现了BinaryComparable类可以作为key值
         /***************************
         * Mapper接口中的map方法：
         * void map(K1 key, V1 value, OutputCollector<K2,V2> output, Reporter reporter)
         * 映射一个单个的输入k/v对到一个中间的k/v对
         * 输出对不需要和输入对是相同的类型，输入对可以映射到0个或多个输出对。
         * OutputCollector接口：收集Mapper和Reducer输出的<k,v>对。
         * OutputCollector接口的collect(k, v)方法:增加一个(k,v)对到output
         ***************************/
         public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
             // 用StringTokenizer作为分词器，对value进行分词
             StringTokenizer itr = new StringTokenizer(value.toString());
             // 遍历分词后结果
             while (itr.hasMoreTokens()) {
                 // 将String设置入Text类型word
                 word.set(itr.nextToken());
~~~

```
                // 将(word,1)，即(Text,IntWritable)写入上下文context，供后续Reduce阶段使用
                context.write(word, one);
            }
        }
    }

    // IntSumReducer作为Reduce阶段，需要继承Reducer，并重写reduce()函数
    public static class IntSumReducer extends Reducer<Text,IntWritable,Text,IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException {
            int sum = 0;
            // 遍历map阶段输出结果中的values中每个val，累加至sum
            for (IntWritable val : values) {
                sum += val.get();
            }
            // 将sum设置入IntWritable类型result
            result.set(sum);
            // 通过上下文context的write()方法，输出结果(key, result)，即(Text,IntWritable)
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        if (otherArgs.length < 2) {
            System.err.println("Usage: wordcount <in> [<in>...] <out>");
            System.exit(2);
        }
        // 构造一个Job实例job，并命名为"word count"
        Job job = Job.getInstance(conf, "word count");
        // 设置jar
        job.setJarByClass(WordCount.class);
        job.setMapperClass(TokenizerMapper.class);  // 为job设置Mapper类
        job.setCombinerClass(IntSumReducer.class);  // 为job设置Combiner类
        job.setReducerClass(IntSumReducer.class);   // 为job设置Reduce类

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        for (int i = 0; i < otherArgs.length - 1; ++i) {
            FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
        }
        FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
        // 等待作业job运行完成并退出
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}、
```

#### WordCount.java编译打包

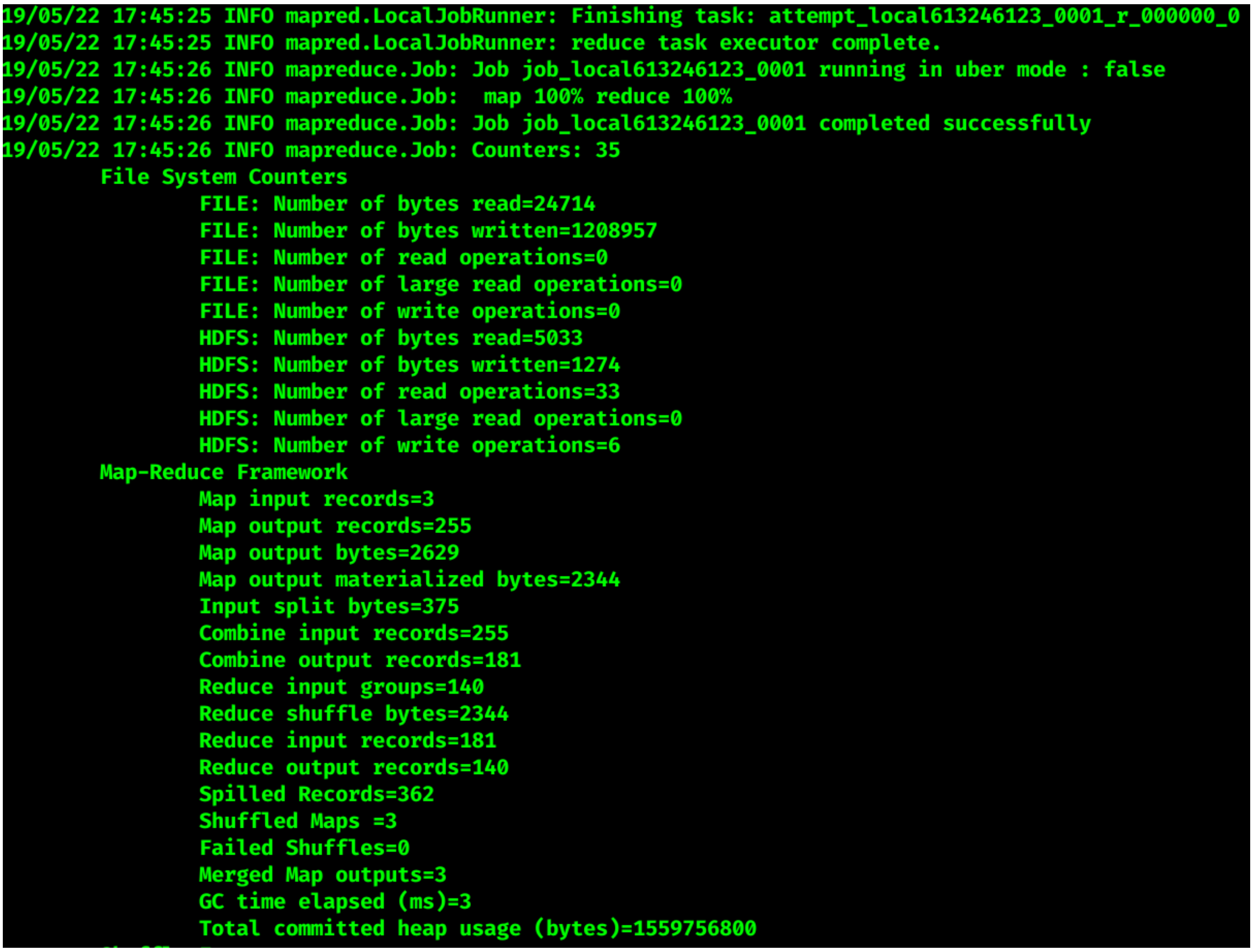
~~~
``` bash
$ cd wordcount_hadoop/
$ ls
WordCount.java
$ mkdir ./classes
$ javac  -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-
2.7.6.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-
2.7.6.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d ./classes/ WordCount.java
$ jar -cvf ./WordCount.jar -C ./classes  .
已添加清单
正在添加: WordCount$TokenizerMapper.class(输入 = 1736) (输出 = 754)(压缩了 56%)
正在添加: WordCount$IntSumReducer.class(输入 = 1739) (输出 = 737)(压缩了 57%)
正在添加: WordCount$TokenizerMapper$CountersEnum.class(输入 = 1021) (输出 = 507)(压缩了 50%)
正在添加: WordCount.class(输入 = 1907) (输出 = 1038)(压缩了 45%)
```
~~~

#### 在HDFS上执行WordCount.jar

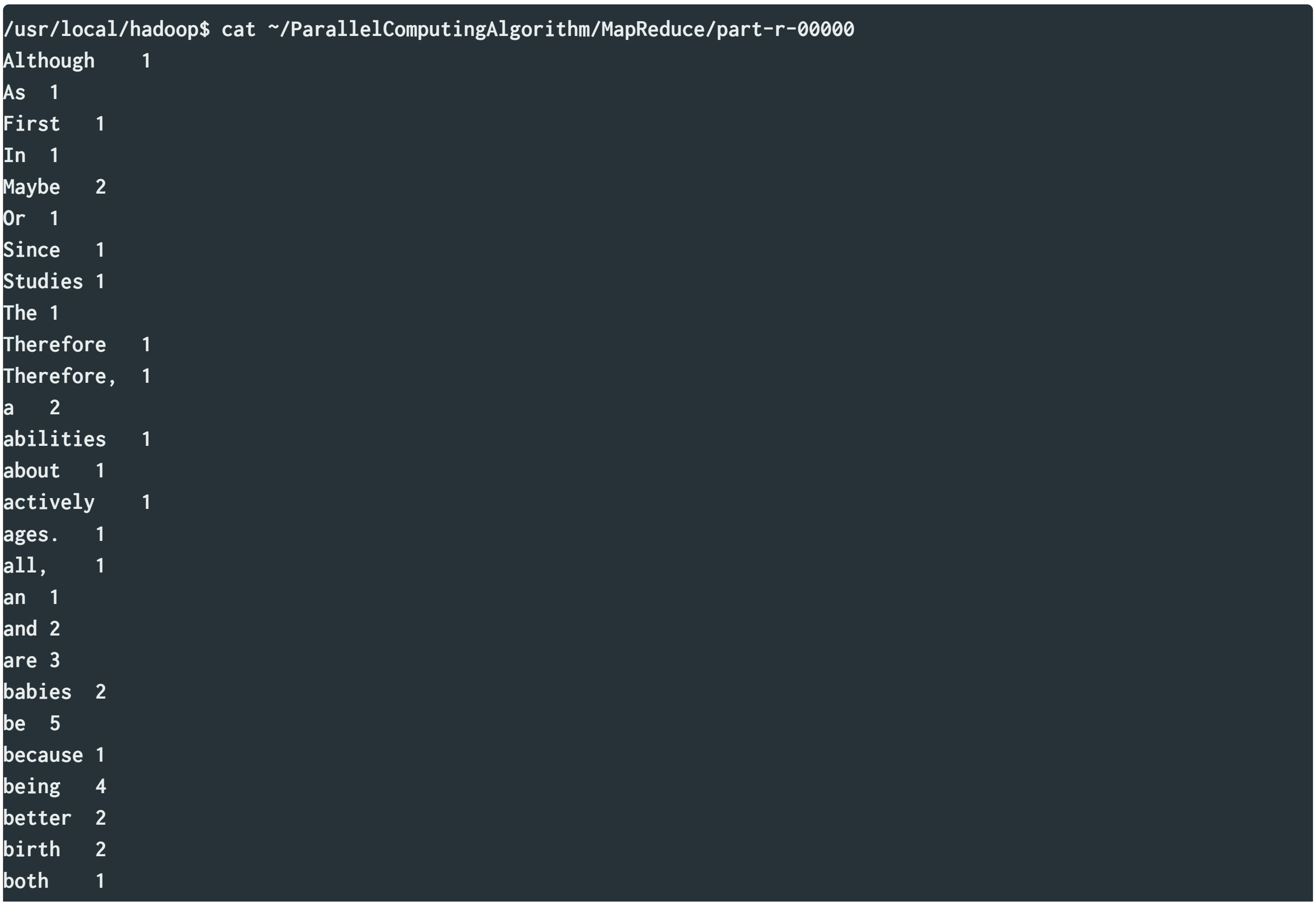
~~~
``` bash
/usr/local/hadoop$ ./bin/hadoop jar ~/ParallelComputingAlgorithm/MapReduce/wordcount_hadoop/WordCount.jar
WordCount wordcount/input wordcount/output
/usr/local/hadoop$ ./bin/hadoop fs -ls -R
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:35 input
-rw-r--r--   1 hadoop supergroup       4436 2019-05-22 17:01 input/capacity-scheduler.xml
-rw-r--r--   1 hadoop supergroup       1116 2019-05-22 17:01 input/core-site.xml
-rw-r--r--   1 hadoop supergroup       9683 2019-05-22 17:01 input/hadoop-policy.xml
-rw-r--r--   1 hadoop supergroup       1188 2019-05-22 17:01 input/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup        620 2019-05-22 17:01 input/httpfs-site.xml
-rw-r--r--   1 hadoop supergroup       3518 2019-05-22 17:01 input/kms-acls.xml
-rw-r--r--   1 hadoop supergroup       5540 2019-05-22 17:01 input/kms-site.xml
-rw-r--r--   1 hadoop supergroup        690 2019-05-22 17:01 input/yarn-site.xml
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:45 wordcount
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:36 wordcount/input
-rw-r--r--   1 hadoop supergroup        464 2019-05-22 17:36 wordcount/input/input1.txt
-rw-r--r--   1 hadoop supergroup        511 2019-05-22 17:36 wordcount/input/input2.txt
-rw-r--r--   1 hadoop supergroup        643 2019-05-22 17:36 wordcount/input/input3.txt
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:45 wordcount/output
-rw-r--r--   1 hadoop supergroup          0 2019-05-22 17:45 wordcount/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup       1274 2019-05-22 17:45 wordcount/output/part-r-00000
```
~~~

#### 获取输出，实验成功

~~~
``` bash
/usr/local/hadoop$ ./bin/hadoop fs -get wordcount/output/part-r-00000 ~/ParallelComputingAlgorithm/MapReduce/
```
~~~



```
19/05/22 17:45:25 INFO mapred.LocalJobRunner: Finishing task: attempt_local613246123_0001_r_000000_0
19/05/22 17:45:25 INFO mapred.LocalJobRunner: reduce task executor complete.
19/05/22 17:45:26 INFO mapreduce.Job: Job job_local613246123_0001 running in uber mode : false
19/05/22 17:45:26 INFO mapreduce.Job:  map 100% reduce 100%
19/05/22 17:45:26 INFO mapreduce.Job: Job job_local613246123_0001 completed successfully
19/05/22 17:45:26 INFO mapreduce.Job: Counters: 35
        File System Counters
                FILE: Number of bytes read=24714
                FILE: Number of bytes written=1208957
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=5033
                HDFS: Number of bytes written=1274
                HDFS: Number of read operations=33
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=6
        Map-Reduce Framework
                Map input records=3
                Map output records=255
                Map output bytes=2629
                Map output materialized bytes=2344
                Input split bytes=375
                Combine input records=255
                Combine output records=181
                Reduce input groups=140
                Reduce shuffle bytes=2344
                Reduce input records=181
                Reduce output records=140
                Spilled Records=362
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=3
                Total committed heap usage (bytes)=1559756800
```



```
/usr/local/hadoop$ cat ~/ParallelComputingAlgorithm/MapReduce/part-r-00000
Although	1
As	1
First	1
In	1
Maybe	2
Or	1
Since	1
Studies	1
The	1
Therefore	1
Therefore,	1
a	2
abilities	1
about	1
actively	1
ages.	1
all,	1
an	1
and	2
are	3
babies	2
be	5
because	1
being	4
better	2
birth	2
both	1
```

```
brothers    1
busy    1
care    1
cause   1
certain 1
changes 1
children    2
cognitive   1
compared    1
cortisol    3
could   3
danger  1
different   2
direct  1
effects 2
emotional   1
excited 2
excitement  2
expecting   1
explain 1
explanation 1
exposed 1
feel    1
first-time  2
firstborn   4
firstborn.  2
for 2
from    2
geared  1
genes   2
genes,  1
genetically 2
happen  1
have    1
high    2
higher  2
human   1
identical   1
in  1
infants 3
infants,    1
infants.    1
intense 1
is  1
it  1
larger  1
lead    1
level   5
levels  2
may 2
monkey  1
monkeys 1
more    4
mothers 2
mothers,    1
necessary   1
nervous 1
nervous.    1
not 2
of  19
```

```
older   2
or  5
order   1
order.  1
out 1
parent. 1
possible    1
potential   1
pregnant    2
random  1
rather  1
relatively  2
released.   1
releasing   1
responded   1
result  2
returning   1
sample  1
samples 1
sensitive   1
share   1
siblings    2
similar 1
simply  1
sisters.    1
situations. 1
size    1
small,  1
stimulating 1
stimulation 3
stimulation.    2
stress  1
studies 1
taking  1
tease   1
terms   1
than    1
that    2
the 19
their   4
they    2
to  9
too 1
towards 1
unfamiliar  1
usually 1
was 1
we  1
were    4
when    1
which   1
with    4
younger 2
```

### 题目二

实现一个统计输入文件中各个长度的单词出现频次的程序。

#### 输入生成代码

~~~
``` python
import numpy.random as random

def generate_random_str(randomlength=16):
    """
    生成一个指定长度的随机字符串
    """
    random_str = ''
    base_str = 'ABCDEFGHIGKLMNOPQRSTUVWXYZ'
    length = len(base_str) - 1
    for i in range(randomlength):
        random_str += base_str[random.randint(0, length)]
    return random_str

output = open("./input.txt","w")
count = random.randint(20,50)
for i in range(count):
    output.write(generate_random_str(random.randint(0, 10)) + " ")

output.close()
```
~~~

#### 在HDFS上创建目录上传输入文件

~~~
``` bash
/usr/local/hadoop$ ./bin/hadoop fs -mkdir lencount
/usr/local/hadoop$ ./bin/hadoop fs -mkdir lencount/input
/usr/local/hadoop$ ./bin/hadoop fs -put ~/ParallelComputingAlgorithm/MapReduce/input4.txt  lencount/input
/usr/local/hadoop$ ./bin/hadoop fs -put ~/ParallelComputingAlgorithm/MapReduce/input5.txt  lencount/input
/usr/local/hadoop$ ./bin/hadoop fs -put ~/ParallelComputingAlgorithm/MapReduce/input6.txt  lencount/input
/usr/local/hadoop$ ./bin/hadoop fs -put ~/ParallelComputingAlgorithm/MapReduce/input7.txt  lencount/input
/usr/local/hadoop$ ./bin/hadoop fs -put ~/ParallelComputingAlgorithm/MapReduce/input8.txt  lencount/input
/usr/local/hadoop$ ./bin/hadoop fs -ls -R
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:35 input
-rw-r--r--   1 hadoop supergroup       4436 2019-05-22 17:01 input/capacity-scheduler.xml
-rw-r--r--   1 hadoop supergroup       1116 2019-05-22 17:01 input/core-site.xml
-rw-r--r--   1 hadoop supergroup       9683 2019-05-22 17:01 input/hadoop-policy.xml
-rw-r--r--   1 hadoop supergroup       1188 2019-05-22 17:01 input/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup        620 2019-05-22 17:01 input/httpfs-site.xml
-rw-r--r--   1 hadoop supergroup       3518 2019-05-22 17:01 input/kms-acls.xml
-rw-r--r--   1 hadoop supergroup       5540 2019-05-22 17:01 input/kms-site.xml
-rw-r--r--   1 hadoop supergroup        690 2019-05-22 17:01 input/yarn-site.xml
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 20:47 lencount
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 20:48 lencount/input
-rw-r--r--   1 hadoop supergroup         58 2019-05-22 20:47 lencount/input/input4.txt
-rw-r--r--   1 hadoop supergroup        107 2019-05-22 20:48 lencount/input/input5.txt
-rw-r--r--   1 hadoop supergroup         77 2019-05-22 20:48 lencount/input/input6.txt
-rw-r--r--   1 hadoop supergroup        116 2019-05-22 20:48 lencount/input/input7.txt
-rw-r--r--   1 hadoop supergroup        127 2019-05-22 20:48 lencount/input/input8.txt
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:45 wordcount
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:36 wordcount/input
-rw-r--r--   1 hadoop supergroup        464 2019-05-22 17:36 wordcount/input/input1.txt
-rw-r--r--   1 hadoop supergroup        511 2019-05-22 17:36 wordcount/input/input2.txt
-rw-r--r--   1 hadoop supergroup        643 2019-05-22 17:36 wordcount/input/input3.txt
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:45 wordcount/output
-rw-r--r--   1 hadoop supergroup          0 2019-05-22 17:45 wordcount/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup       1274 2019-05-22 17:45 wordcount/output/part-r-00000
```
~~~

#### LenCount.java程序解析

~~~
``` java
import org.apache.hadoop.fs.FileSystem;
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;

import java.io.IOException;
import java.util.StringTokenizer;

public class LenCount {
    public static class CounterMapper extends Mapper<Object, Text, Text, IntWritable>{
        private final static IntWritable one = new IntWritable(1);
        private Text word_len = new Text();
        public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
            // 用StringTokenizer作为分词器，对value进行分词
            StringTokenizer itr = new StringTokenizer(value.toString());
            // 遍历分词后结果
            while (itr.hasMoreTokens()) {
                // 将String设置入Text类型word
                word_len.set(Integer.toString(itr.nextToken().length()));
                // 将(word,1)，即(Text,IntWritable)写入上下文context，供后续Reduce阶段使用
                context.write(word_len, one);
            }
        }
    }
    public static class IntSumReducer extends Reducer<Text,IntWritable,Text,IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException {
            int sum = 0;
            // 遍历map阶段输出结果中的values中每个val，累加至sum
            for (IntWritable val : values) {
                sum += val.get();
            }
            // 将sum设置入IntWritable类型result
            result.set(sum);
            // 通过上下文context的write()方法，输出结果(key, result)，即(Text,IntWritable)
            context.write(key, result);
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
        if (otherArgs.length < 2) {
            System.err.println(”Usage: wordlen <in> [<in>...] <out>”);
            System.exit(2);
        }
        // 构造一个Job实例job，并命名为"wordlen count"
~~~

~~~~
        Job job = Job.getInstance(conf, "wordlen count");
        // 设置jar
        job.setJarByClass(LenCount.class);
        job.setMapperClass(CounterMapper.class);    // 为job设置Mapper类
        job.setCombinerClass(IntSumReducer.class);  // 为job设置Combiner类
        job.setReducerClass(IntSumReducer.class);   // 为job设置Reduce类

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        for (int i = 0; i < otherArgs.length - 1; ++i) {
            FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
        }
        FileOutputFormat.setOutputPath(job, new Path(otherArgs[otherArgs.length - 1]));
        // 等待作业job运行完成并退出
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```
~~~~

#### LenCount.java编译打包

~~~~
``` bash
$ cd ../lencount/
$ mkdir ./classes
$ javac  -classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-2.7.6.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.7.6.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar -d ./classes/ LenCount.java
$ jar -cvf ./LenCount.jar -C ./classes  .
已添加清单
正在添加: LenCount.class(输入 = 1902) (输出 = 1034)(压缩了 45%)
正在添加: LenCount$CounterMapper.class(输入 = 1843) (输出 = 805)(压缩了 56%)
`正在添加: LenCount$IntSumReducer.class(输入 = 1736) (输出 = 739)(压缩了 57%)
~~~~

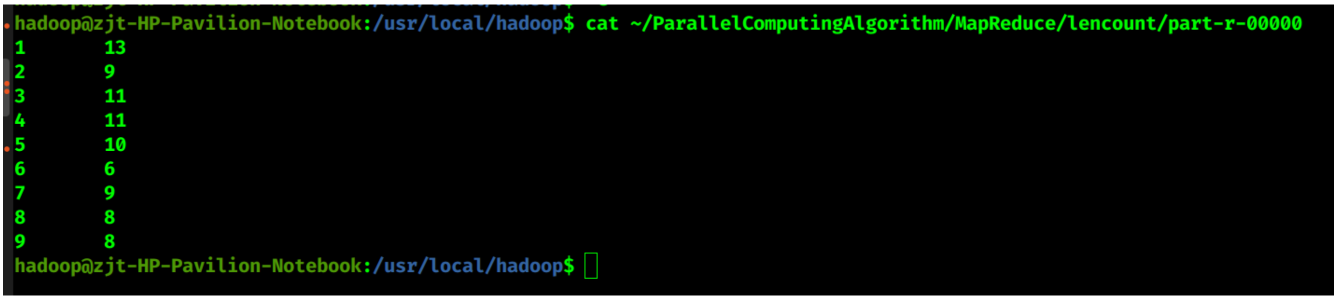
#### 在HDFS上执行LenCount.jar

~~~~
``` bash
/usr/local/hadoop$ ./bin/hadoop jar ~/ParallelComputingAlgorithm/MapReduce/lencount/LenCount.jar LenCount lencount/input lencount/output
...
/usr/local/hadoop$ ./bin/hadoop fs -ls -R
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:35 input
-rw-r--r--   1 hadoop supergroup       4436 2019-05-22 17:01 input/capacity-scheduler.xml
-rw-r--r--   1 hadoop supergroup       1116 2019-05-22 17:01 input/core-site.xml
-rw-r--r--   1 hadoop supergroup       9683 2019-05-22 17:01 input/hadoop-policy.xml
-rw-r--r--   1 hadoop supergroup       1188 2019-05-22 17:01 input/hdfs-site.xml
-rw-r--r--   1 hadoop supergroup        620 2019-05-22 17:01 input/httpfs-site.xml
-rw-r--r--   1 hadoop supergroup       3518 2019-05-22 17:01 input/kms-acls.xml
-rw-r--r--   1 hadoop supergroup       5540 2019-05-22 17:01 input/kms-site.xml
-rw-r--r--   1 hadoop supergroup        690 2019-05-22 17:01 input/yarn-site.xml
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 20:49 lencount
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 20:48 lencount/input
-rw-r--r--   1 hadoop supergroup         58 2019-05-22 20:47 lencount/input/input4.txt
-rw-r--r--   1 hadoop supergroup        107 2019-05-22 20:48 lencount/input/input5.txt
-rw-r--r--   1 hadoop supergroup         77 2019-05-22 20:48 lencount/input/input6.txt
-rw-r--r--   1 hadoop supergroup        116 2019-05-22 20:48 lencount/input/input7.txt
-rw-r--r--   1 hadoop supergroup        127 2019-05-22 20:48 lencount/input/input8.txt
~~~~

~~~~
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 20:49 lencount/output
-rw-r--r--   1 hadoop supergroup          0 2019-05-22 20:49 lencount/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup         40 2019-05-22 20:49 lencount/output/part-r-00000
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:45 wordcount
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:36 wordcount/input
-rw-r--r--   1 hadoop supergroup        464 2019-05-22 17:36 wordcount/input/input1.txt
-rw-r--r--   1 hadoop supergroup        511 2019-05-22 17:36 wordcount/input/input2.txt
-rw-r--r--   1 hadoop supergroup        643 2019-05-22 17:36 wordcount/input/input3.txt
drwxr-xr-x   - hadoop supergroup          0 2019-05-22 17:45 wordcount/output
-rw-r--r--   1 hadoop supergroup          0 2019-05-22 17:45 wordcount/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup       1274 2019-05-22 17:45 wordcount/output/part-r-00000
```
~~~~

#### 获取输出，实验成功

~~~~
``` bash
/usr/local/hadoop$ ./bin/hadoop fs -get lencount/output/part-r-00000
~/ParallelComputingAlgorithm/MapReduce/lencount/
/usr/local/hadoop$ cat ~/ParallelComputingAlgorithm/MapReduce/lencount/part-r-00000
```
~~~~



```
hadoop@zjt-HP-Pavilion-Notebook:/usr/local/hadoop$ cat ~/ParallelComputingAlgorithm/MapReduce/lencount/part-r-00000
1       13
2       9
3       11
4       11
5       10
6       6
7       9
8       8
9       8
hadoop@zjt-HP-Pavilion-Notebook:/usr/local/hadoop$ 
```

总结

通过算法实现锻炼了并行思维，熟悉了MapReduce分布式并行环境的使用。