

# 并行计算 Lab4: MapReduce

## 1. 实验简介

Hadoop 是要给开源框架，它遵循谷歌的方法实现了 MapReduce 算法，用以查询在互联网上分布的数据集。

## 2. 配置过程

### 2.1 准备工作

一下过程主要参考助教给出的压缩包中的 `hadoop安装运行说明.txt`

安装 JDK: 参考 [如何在 Ubuntu 20.04 上安装 Java](#), 目前安装的是 OpenJDK 11

ssh 已安装, `openssh-server` 需要安装。

sshd 服务已经启动, ssh 密钥对已经生成。

公钥添加: `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`

`ssh localhost` 可以连接。

### 2.2 安装配置 Hadoop

解压并重命名到用户主目录, 将 hadoop 路径加入 path。

修改 `~/hadoop/conf/hadoop-env.sh`, 在该文件最后一行添加 JAVA\_HOME 的路径:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

### 2.3 配置单机模式

对 `conf` 目录下面的配置文件不做修改即为单机模式

### 2.4 配置伪分布模式

#### 2.4.1 修改 `core-site.xml` 文件

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

告诉 Hadoop 使用 HDFS 作为默认的文件系统, 并将 namenode 的地址设置为 localhost:9000

## 2.4.2 修改 mapred-site.xml 文件

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
</configuration>
```

告诉 Hadoop 使用 JobTracker 作为默认的任务调度程序，并将其地址设置为 localhost:9001

## 2.4.3 修改 hdfs-site.xml 文件

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

告诉 Hadoop 将数据副本数量设置为 1，这是为了减少数据冗余和存储开销。

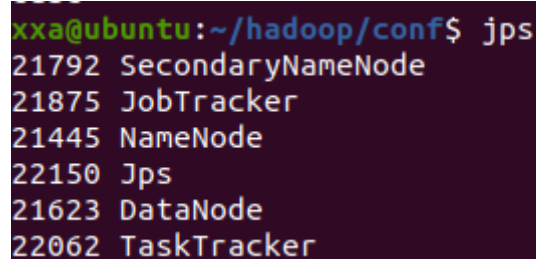
# 3. 使用 Hadoop

## 3.1 启动 Hadoop

格式化：`~/hadoop/bin/hadoop namenode -format`

启动 Hadoop：`~/hadoop/bin/start-all.sh`

成功启动：



```
xxa@ubuntu:~/hadoop/conf$ jps
21792 SecondaryNameNode
21875 JobTracker
21445 NameNode
22150 Jps
21623 DataNode
22062 TaskTracker
```

## 3.2 在 HDFS 中添加文件和目录

创建文件夹：`hadoop fs -mkdir /user/kke/wordcount/input`

将文本文件从本地目录上传到 HDFS 中：

`hadoop fs -put ./input1.txt /user/kke/wordcount/input`

查看是否上传成功：

```

xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/hadoop$ hadoop fs -lsr /
Warning: $HADOOP_HOME is deprecated.

WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:
/home/xxa/hadoop/hadoop-core-1.0.4.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.
util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
drwxr-xr-x - xxa supergroup 0 2023-05-02 05:41 /tmp
drwxr-xr-x - xxa supergroup 0 2023-05-02 05:41 /tmp/hadoop-xxa
drwxr-xr-x - xxa supergroup 0 2023-05-02 05:41 /tmp/hadoop-xxa/mapred
drwx----- xxa supergroup 0 2023-05-02 05:41 /tmp/hadoop-xxa/mapred/system
-rw----- 1 xxa supergroup 4 2023-05-02 05:41 /tmp/hadoop-xxa/mapred/system/jobtracker.info
drwxr-xr-x - xxa supergroup 0 2023-05-02 05:44 /user
drwxr-xr-x - xxa supergroup 0 2023-05-02 05:44 /user/kke
drwxr-xr-x - xxa supergroup 0 2023-05-02 05:44 /user/kke/wordcount
drwxr-xr-x - xxa supergroup 0 2023-05-02 05:46 /user/kke/wordcount/input
-rw-r--r-- 1 xxa supergroup 35 2023-05-02 05:46 /user/kke/wordcount/input/input1.txt
-rw-r--r-- 1 xxa supergroup 17 2023-05-02 05:46 /user/kke/wordcount/input/input2.txt

```

## 3.3 运行 Hadoop 作业

编译 WordCount.java

这个代码用于统计给定文本文件中每个单词出现的次数。

```

javac -classpath /home/xxa/hadoop/hadoop-core-
1.0.4.jar:/home/xxa/hadoop/lib/commons-cli-1.2.jar -d ./classes/
./WordCount.java

```

该命令使用 Java 编译器 ( `javac` ) 将 `WordCount.java` 文件编译成字节码文件 ( `WordCount.class` ), 并将其存储在 `./classes/` 目录中。

```

jar -cvf ./WordCount.jar -C ./classes .

```

该命令使用 Java 档案工具 ( `jar` ) 将 `./classes/` 目录中的所有字节码文件打包成 `WordCount.jar` 文件。打包后的 `WordCount.jar` 文件包含了编译好的 Java 字节码文件和其它资源, 可以在 Hadoop 集群上运行 MapReduce 作业。

运行 Hadoop 作业:

```

hadoop jar ./WordCount.jar wordCount /user/kke/wordcount/input
/user/kke/wordcount/output

```

如果提示你说输出文件夹已经存在, 那么则执行如下命令删除:

```

hadoop fs -rmr /user/kke/wordcount/output

```

## 3.4 获得运行结果

```

hadoop fs -ls /user/kke/wordcount/output

```

```

xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount$ hadoop fs -ls /user/kke/wordcount/output
Warning: $HADOOP_HOME is deprecated.

WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/home/xxa/hadoop/had
op-core-1.0.4.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Found 3 items
-rw-r--r-- 1 xxa supergroup 0 2023-05-02 06:09 /user/kke/wordcount/output/_SUCCESS
drwxr-xr-x - xxa supergroup 0 2023-05-02 06:08 /user/kke/wordcount/output/_logs
-rw-r--r-- 1 xxa supergroup 67 2023-05-02 06:09 /user/kke/wordcount/output/part-r-00000

```

```

hadoop fs -get /user/kke/wordcount/output/ ./

```

```
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount$ ls
classes output WordCount.jar WordCount.java
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount$ cd output/
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount/output$ ls
_logs part-r-00000 _SUCCESS
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount/output$ cat part-r-00000
a      1
b      1
bc     1
cd     1
dd     1
de     1
dhs    1
e      2
f      1
fg     1
g      1
gh     1
hdk    1
tt     2
```

## 3.5 关闭 Hadoop 集群

```
stop-all.sh
```

## 4. 题目2：统计输入文件中各个长度的单词出现频次

### 4.1 实现 Java 代码

在助教提供的代码基础上修改。

首先，分析助教提供的 Java 代码的整体执行逻辑：

这段代码是一个 Hadoop MapReduce 程序，用于统计给定文本文件中每个单词出现的次数。

在执行该程序时，Hadoop 将根据输入路径中指定的文件或目录，将数据分片并分配给不同的 Mapper 进行处理，然后将 Mapper 的输出结果按照 Key 进行排序和分组，再传递给 Reducer 进行归并，最终输出每个单词的出现次数到输出路径中的文件中。

Mapper 将输入的文本数据按照空格进行分词，构造 <单词,1> 键值对。

Reducer 将 Mapper 输出的键值对进行归并，并输出每个单词出现的总次数。

main 方法负责组装 MapReduce 作业的各个组件，并提交作业到 Hadoop 集群进行运行。

- 创建一个 `Configuration` 对象，并使用 `GenericOptionsParser` 从命令行参数中获取输入和输出路径。
- 创建一个新的 `Job` 对象，并为其设置作业名称和所需的组件，包括 Mapper、Combiner、Reducer、输入路径和输出路径。
- 等待作业完成并打印作业的执行结果。

相比之下，我们要实现的代码的功能是**统计各个长度的单词出现频次**，相比于统计各单词出现的频率，要修改的内容不多。

首先，程序的整体功能是：

该程序是要在 Hadoop 上进行单词计数的功能，根据单词长度进行统计。

在 Mapper 中，使用 `StringTokenizer` 对输入文本进行分词，然后遍历每个分词，统计单词长度，并将单词长度作为键，值为 1 传递给 Reducer。

在 Reducer 中，将相同键的值进行累加，并输出结果。

在 main 方法中，设置 Mapper、Combiner 和 Reducer，并设置输入路径和输出路径，然后提交作业等待运行结果。

根据以上分析，我们要在源代码的基础上做出以下修改：

- 在 TokenizerMapper 的 map 方法中，while 循环遍历每个单词时，存入 context 的键值对是 `<单词长度, one>`。
- IntSumReducer 无需改变。
- main 方法也基本无需修改。

此外，需要修改个别变量名称。完整代码见附件。

## 4.2 自动生成输入文件

要求每行单词之间用空格间隔。

我的实现思路是：

- 行数随机，在 2 到 101 之间；
- 每行 10 个单词；
- 单词是字母的随机组合，每个单词长度不超过 20；

## 4.3 在 Hadoop 上测试

编译 WordCount2.java，打包，在 Hadoop 上运行。

获取运行结果。

```
hadoop fs -ls /user/kke/wordcount/output
```

```
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount$ hadoop fs -ls /user/kke/wordcount/output
Warning: SHADOOP_HOME is deprecated.

WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/home/xxa/hadoop/hadoop-core-1.0.4.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Found 3 items
-rw-r--r-- 1 xxa supergroup      0 2023-05-02 07:08 /user/kke/wordcount/output/_SUCCESS
drwxr-xr-x - xxa supergroup      0 2023-05-02 07:07 /user/kke/wordcount/output/_logs
-rw-r--r-- 1 xxa supergroup    12 2023-05-02 07:08 /user/kke/wordcount/output/part-r-00000
```

```
hadoop fs -get /user/kke/wordcount/output/ ./
```

```
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount$ hadoop fs -get /user/kke/wordcount/output/ ./
Warning: SHADOOP_HOME is deprecated.

WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/home/xxa/hadoop/hadoop-core-1.0.4.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount$ ls
classes  output  WordCount2.jar  WordCount2.java  WordCount.jar  WordCount.java
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount$ cd output/
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount/output$ ls
_logs  part-r-00000  _SUCCESS
xxa@ubuntu:~/Desktop/Parallel-Computing-Labs/Lab4/wordcount/output$ cat part-r-00000
1      6
2      8
3      2
```

与期望输出相吻合。

## 5. 附录

- PB20061343\_徐奥\_实验四.pdf：实验报告
- wordCount2.java：统计各个长度的单词出现频次的源代码

- `Genwords.c`: 生成 input 的 c 代码