并行计算实验 4: MapReduce

傅申 PB20000051

1. 实验目的

如下:

- 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;
- 运行 `WordCount` 程序, 得到统计结果;
- 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.

2. 实验环境

本次实验在我自己的笔记本上运行, 硬件参数如下:

CPU Intel i5-1035G1, 4 核 8 进程.
内存 双通道 16 GB DDR4 3200 MHz 内存.

相应的软件环境如下:

OS GNU/Linux 6.1.31-2-MANJARO x86_64.
Hadoop 1.0.4

3. 实验过程与结果

本次实验中, 目录结构如下 (忽略输出目录 `output` 中的文件):

```
.
├── build
│   ├── lengthcount
│   │   ├── classes
│   │   └── LengthCount.jar
│   └── wordcount
│       ├── classes
│       └── WordCount.jar
├── input
│   ├── lengthcount
│   │   ├── input1.txt
│   │   └── input2.txt
│   └── wordcount
│       ├── input1.txt
│       └── input2.txt
├── output
│   ├── lengthcount
│   └── wordcount
└── src
    ├── LengthCount.java
    └── WordCount.java
```

3.1. 实现 `LengthCount`

基于 `WordCount.java` 实现 `LengthCount.java`, map 部分修改如下:

```
public static class TokenizerMapper
    extends Mapper<Object, Text, IntWritable, IntWritable>
{
    private final static IntWritable one    = new IntWritable(1);
    private IntWritable              length = new IntWritable();

    public void map(Object key, Text value, Context context)
        throws IOException, InterruptedException
    {
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            length.set(itr.nextToken().length());
            context.write(length, one);
        }
    }
}
```

reduce 部分只需要修改参数类型即可:

```
public static class IntSumReducer
    extends Reducer<IntWritable, IntWritable, IntWritable, IntWritable>
{
    private IntWritable result = new IntWritable();

    public void reduce(IntWritable key, Iterable<IntWritable> values,
                       Context context)
        throws IOException, InterruptedException
    {
        // ...
    }
}
```

同时, 在 `main` 函数中还需要设置任务的输出键类型为 `IntWritable`:

```
job.setOutputKeyClass(IntWritable.class);
```

3.2. 搭建 Hadoop 环境并运行程序

安装 OpenJDK, SSH 和 Hadoop 并进行相应配置后, 先格式化 HDFS, 并启动 Hadoop (由于某种原因, 我需要手动启动 SecondaryNameNode, DataNode 和 TaskTracker):

```
$ hadoop namenode -format
$ start-all.sh
$ hadoop-daemon.sh start secondarynamenode
$ hadoop-daemon.sh start datanode
$ hadoop-daemon.sh start tasktracker
```

运行 `jps`, 输出如下:

```
$ jps
310789 NameNode
316406 TaskTracker
316112 DataNode
311245 JobTracker
318111 Jps
315519 SecondaryNameNode
```

然后在 HDFS 中添加文件和目录:

```
$ hadoop fs -mkdir /user/fushen/wordcount/input
$ hadoop fs -mkdir /user/fushen/lengthcount/input
```

将本地 `input` 目录中的文件上传到 HDFS 中:

```
$ cd input/wordcount
$ hadoop fs -put ./input*.txt /user/fushen/wordcount/input
$ hadoop fs -ls /user/fushen/wordcount/input
Found 2 items
-rw-r--r--   1 fushen supergroup         29 2023-06-11 21:17 /user/fushen/
wordcount/input/input1.txt
-rw-r--r--   1 fushen supergroup         29 2023-06-11 21:17 /user/fushen/
wordcount/input/input2.txt

$ cd ../lengthcount
$ hadoop fs -put ./input*.txt /user/fushen/lengthcount/input
$ hadoop fs -ls /user/fushen/lengthcount/input
Found 2 items
-rw-r--r--   1 fushen supergroup         36 2023-06-11 21:23 /user/fushen/
lengthcount/input/input1.txt
-rw-r--r--   1 fushen supergroup         18 2023-06-11 21:23 /user/fushen/
lengthcount/input/input2.txt
```

编译 `WordCount.java` 和 `LengthCount.java` 并打包成 `WordCount.jar` 和 `LengthCount.jar`:

```
$ javac -classpath /home/fushen/hadoop/hadoop-1.0.4/hadoop-core-1.0.4.jar:/home/
fushen/hadoop/hadoop-1.0.4/lib/commons-cli-1.2.jar -d ./build/wordcount/classes ./
src/WordCount.java
$ jar -cvf ./build/wordcount/WordCount.jar -C ./build/wordcount/classes .

$ javac -classpath /home/fushen/hadoop/hadoop-1.0.4/hadoop-core-1.0.4.jar:/home/
fushen/hadoop/hadoop-1.0.4/lib/commons-cli-1.2.jar -d ./build/lengthcount/
classes ./src/LengthCount.java
$ jar -cvf ./build/lengthcount/LengthCount.jar -C ./build/lengthcount/classes .
```

运行 Hadoop 作业:

```
$ hadoop jar ./build/wordcount/WordCount.jar WordCount /user/fushen/wordcount/
input /user/fushen/wordcount/output
$ hadoop jar ./build/lengthcount/LengthCount.jar LengthCount /user/fushen/
lengthcount/input /user/fushen/lengthcount/output
```

取得结果:

```
$ cd output
$ hadoop fs -get /user/fushen/wordcount/output/part-r-00000 ./wordcount
$ hadoop fs -get /user/fushen/lengthcount/output/part-r-00000 ./lengthcount
```

`WordCount` 的运行结果如下:

```
$ cat input/wordcount/*
no zuo no die
you can you up
you can you up
no zuo no die
$ cat output/wordcount/part-r-00000
can     2
die     2
no      4
up      2
you     4
zuo     2
```

`LengthCount` 的运行结果如下:

```
$ cat input/lengthcount/*
a       bc      de      fg      hdk
e       gh      f       tt      dhs
b       cd      e       g
tt      dd
$ cat output/lengthcount/part-r-00000
1       6
2       8
3       2
```

可以看到, 结果正确.

最后, 关闭 Hadoop:

```
$ stop-all.sh
```

4. 实验总结

成功在 Hadoop 上运行了 `WordCount` 和 `LengthCount` 两个程序, 并取得了正确的结果.