**Hadoop安装及其应用**

**课程名称：　　Hadoop安装及其应用**

**专业：　　 计算机科学与技术**

**学　　号：　　 15196116**

**姓　　名：　　 邱来**

**班级：　　 15184113**

1. 实验内容及要求

通过hadoop的安装以及mapreduce框架实现排序算法，熟悉分布式环境下的编程。

1. 实验环境：  
    主机：Ubuntu16.04服务器版

Hadoop版本：2.7.3

JDK:openjdk version "1.8.0\_121"

Hadoop单节点，上面安装hdfs分布式文件系统

1. 实验步骤
2. 安装hadoop并配置；

安装默认的Java开发环境：

sudo apt-get install default-jdk

安装hadoop:

wget http://apache.mirrors.tds.net/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz

tar -xzvf hadoop-2.7.3.tar.gz

sudo mv hadoop-2.7.3 /usr/local/hadoop

设置hadoop运行环境：  
sudo vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh

export JAVA\_HOME=$(readlink -f /usr/bin/java | sed "s:bin/java::")

运行hadoop

/usr/local/hadoop/bin/hadoop 输出信息说明hadoop基本安装完成

配置hadoop支持hdfs文件系统：

设置PATH环境变量：

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/usr/local/hadoop/bin

设置localhost 的ssh无密码访问：

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys

配置文件：

etc/hadoop/core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

etc/hadoop/hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

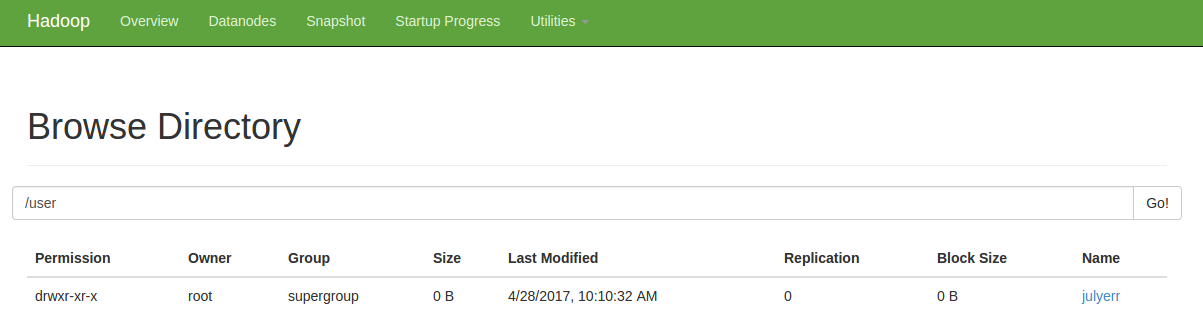
</configuration>

格式化文件系统：

bin/hdfs namenode -format

启动hdfs文件系统

sbin/start-dfs.sh



1. 编写mapreduce代码试下mapreduce的排序算法；

输入数据的格式：

12

893

23

核心代码：

//继承于map类

public static class DataSortMap extends Mapper<LongWritable, Text, IntWritable, IntWritable> {

protected void map(LongWritable key, Text value, Context context)

throws java.io.IOException ,InterruptedException {

String line = value.toString();

context.write(new IntWritable(Integer.parseInt(line)), new IntWritable(1)); //将文件中的数据传输给reducer函数

};

//继承于reduce类

public static class DataSortReduce extends Reducer<IntWritable, IntWritable, IntWritable, IntWritable> {

private static IntWritable linenum = new IntWritable(1);

protected void reduce(IntWritable key, Iterable<IntWritable> value, Context context)

throws java.io.IOException ,InterruptedException {

for(IntWritable val:value){

context.write(linenum, key);

linenum = new IntWritable(linenum.get()+1);

} //按照行号，使用默认的上升的顺序次序写入到文件中

};

}

public static void main(String[] args) {

if(args.length!=2){

System.out.println("args must be three");return ;

}

Configuration conf = new Configuration();

try {

Job job = Job.getInstance(conf,"DataSort");

job.setJarByClass(DataSort.class);

//修改输出和输出的类

job.setMapperClass(DataSortMap.class);

job.setReducerClass(DataSortReduce.class);

job.setOutputKeyClass(IntWritable.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0])); //输入文件路径

FileOutputFormat.setOutputPath(job, new Path(args[1])); //输出文件路径

System.exit(job.waitForCompletion(true) ? 0 : 1 ); //mapreduce过程结束，退出

} catch (Exception e) {

e.printStackTrace();

}

}

1. 调试运行。

部分文件的数据

5956

22

54

6

打包生成jar包：

Javac-classpath /usr/local/hadoop/share/hadoop/common/hadoop-common-2.7.3.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-cli-1.2.jar:/usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.7.3.jar -d classes

jar cvf DataSort.jar DataSort.java classes

向hdfs中添加输入文件：

hdfs dfs -put input/Test2\_input /user/julyerr/input

执行排序过程：  
hadoop jar DataSort.jar Sort /user/julyerr/input /user/julyerr/output

将结果复制到本地：

hdfs dfs -get /user/julyerr/output output/Test2\_output

输出结果:

1 2

2 6

3 15

4 22

5 26

6 32

7 32

8 54

9 92

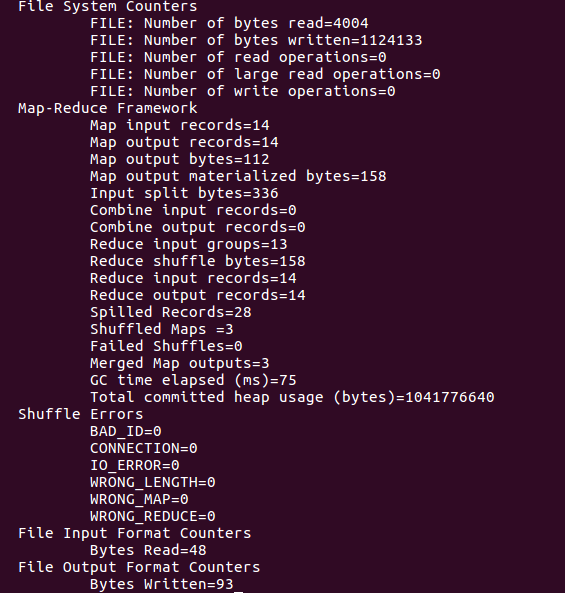
10 650

11 654

12 756

13 5956

14 65223



六 总结：

安装hadoop过程中尽管遇到了很多问题，但是理解Hadoop、hdfs、mapreduce架构，初步知道了Hadoop分布式环境下编程的流程，感觉很有趣，以后会更加深入学习Hadoop，提高自己的专业技能。