

# 大数据数据初步分析

## 1、数据抓取：

### (1) 数据获取代码：

将从聚合数据网中得到的数据按照一定的格式输出：对历史的时间进行循环，按时间从 2018 年 1 月 1 日到 2018 年 6 月 30 日输出，依据当前日期对需要的日期进行运算。其中，依次对五个城市进行数据的抓取，并整合在同一个文本文件中。

```
public class App
{
    public static void main( String[] args )
    {
        String str = "";
        for(int dtime=-245;dtime<-65;dtime++){
            java.text.SimpleDateFormat format = new
java.text.SimpleDateFormat("yyyy-MM-dd");
            Calendar cal = Calendar.getInstance();// 取当前日期。
            cal = Calendar.getInstance();
            cal.add(Calendar.DAY_OF_MONTH, dtime);// 取当前日期的前N天。
            str =format.format(cal.getTime());
            //String res= GetCityList.weather("94", str);//安溪
            //String res= GetCityList.weather("119", str);//闽侯
            //String res= GetCityList.weather("818", str);//哈尔滨
            //String res= GetCityList.weather("1844", str);//上海
            String res= GetCityList.weather("83", str);//北京
            JSONObject obj=JSONObject.fromObject(res);

            String result=obj.getString("result");
            //此时result中数据有多个key,可以对其key进行遍历,得到对个属性
            obj=JSONObject.fromObject(result);
            //今日温度对应的key是today
            String city_id=obj.getString("city_id");//城市地区ID
            String city_name=obj.getString("city_name");//城市地区名称
```

```

String weather_date=obj.getString("weather_date");//天气日期
String day_weather=obj.getString("day_weather");// 白天天气
String night_weather=obj.getString("night_weather");//夜间天气
String day_temp=obj.getString("day_temp");//白天最高温度
String night_temp=obj.getString("night_temp");//夜间最低温度
String day_wind=obj.getString("day_wind");// 白天风向
String day_wind_comp=obj.getString("day_wind_comp");//白天风力
String night_wind=obj.getString("night_wind");// 夜间风向
String night_wind_comp=obj.getString("night_wind_comp");// 夜
间风力
String day_weather_id=obj.getString("day_weather_id");// 白
天天气标识
String night_weather_id=obj.getString("night_weather_id");//
夜间天气标识

```

```

List<String> list = new LinkedList<String>();
list.add(city_id);
list.add(city_name);
list.add(weather_date);
list.add(day_weather);
list.add(night_weather);
list.add(day_temp);
list.add(night_temp);
list.add(day_wind);
list.add(day_wind_comp);
list.add(night_wind);
list.add(night_wind_comp);
list.add(day_weather_id);
list.add(night_weather_id);

```

将取出的数据存在 C 盘目录下的 weather.txt 文件中

```

File file1 = new File("C:\\weather.txt");
try {
    FileWriter fw = new FileWriter(file1,true);
    BufferedWriter bw = new BufferedWriter(fw);

    for(int i = 0; i<list.size();i++){
        bw.write(list.get(i).toString()+" ");
        bw.flush();
        //System.out.println(list.size());
    }
    bw.newLine();
    bw.close();
    fw.close();

} catch (IOException e) {
    e.printStackTrace();
}

```

```
}  
}  
}  
}
```

## (2) 获取的数据：

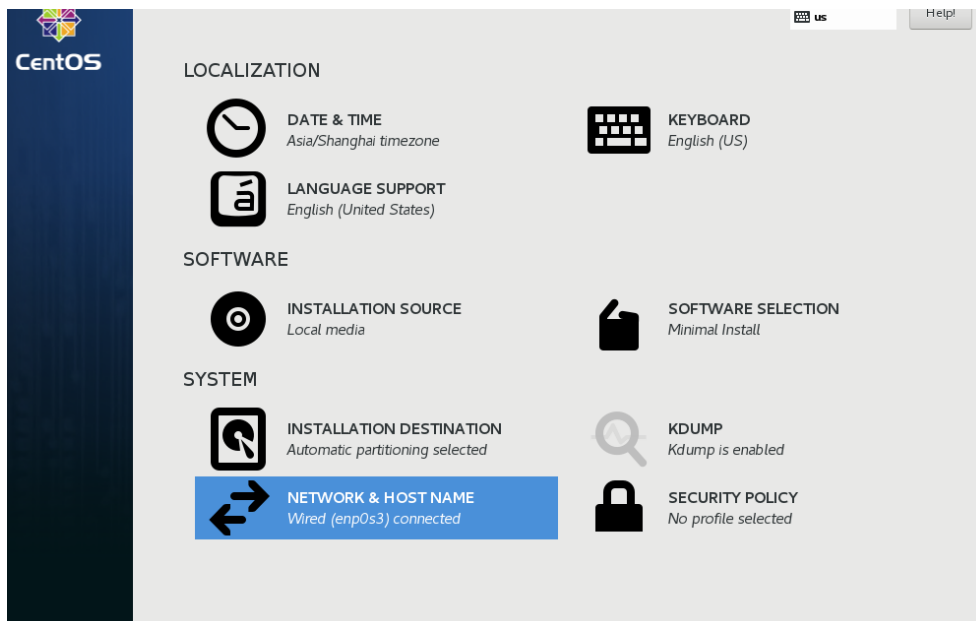
安溪.txt	2018/9/3 15:08	文本文档
北京.txt	2018/9/4 9:28	文本文档
哈尔滨.txt	2018/9/4 9:12	文本文档
汇总.txt	2018/9/4 10:23	文本文档
闽侯.txt	2018/9/3 16:13	文本文档
上海.txt	2018/9/4 9:12	文本文档



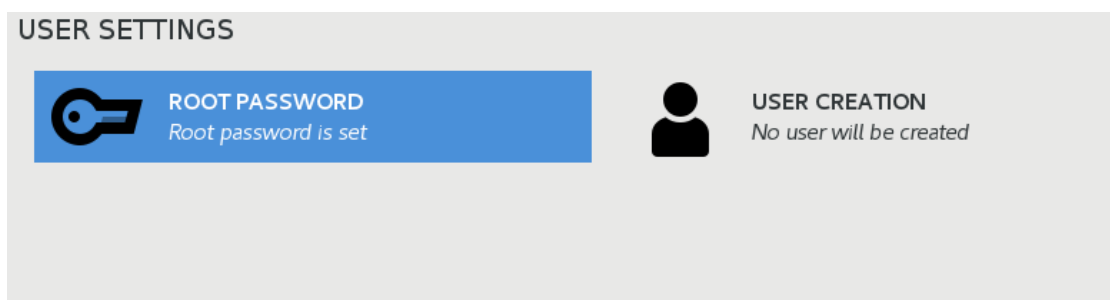
传入虚拟机用的是 weather.txt 文本文件，便于查看在这里改成对应的城市信息

## 2、环境搭建：

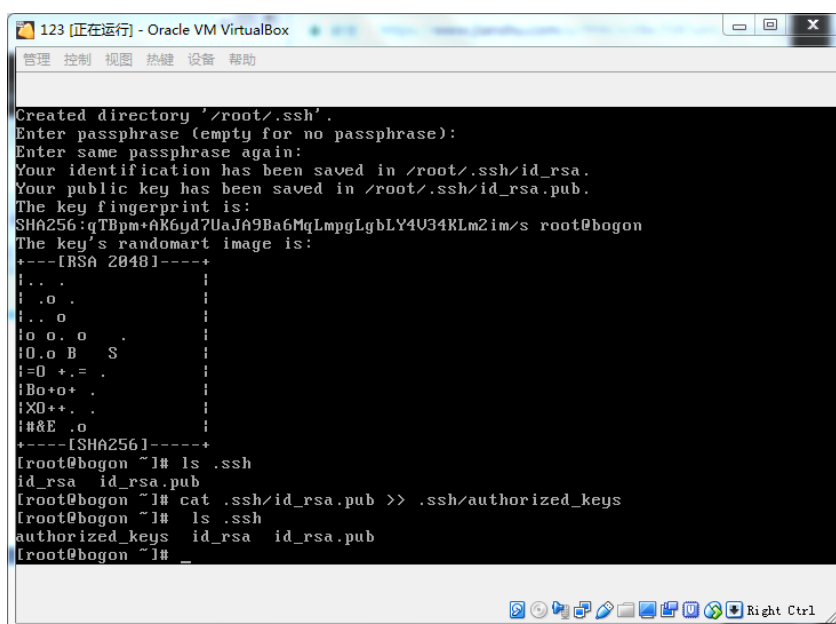
虚拟机安装,设置时区，分配内存空间，开启网络连接



在 ROOT PASSWORD 中设置密码为 123456



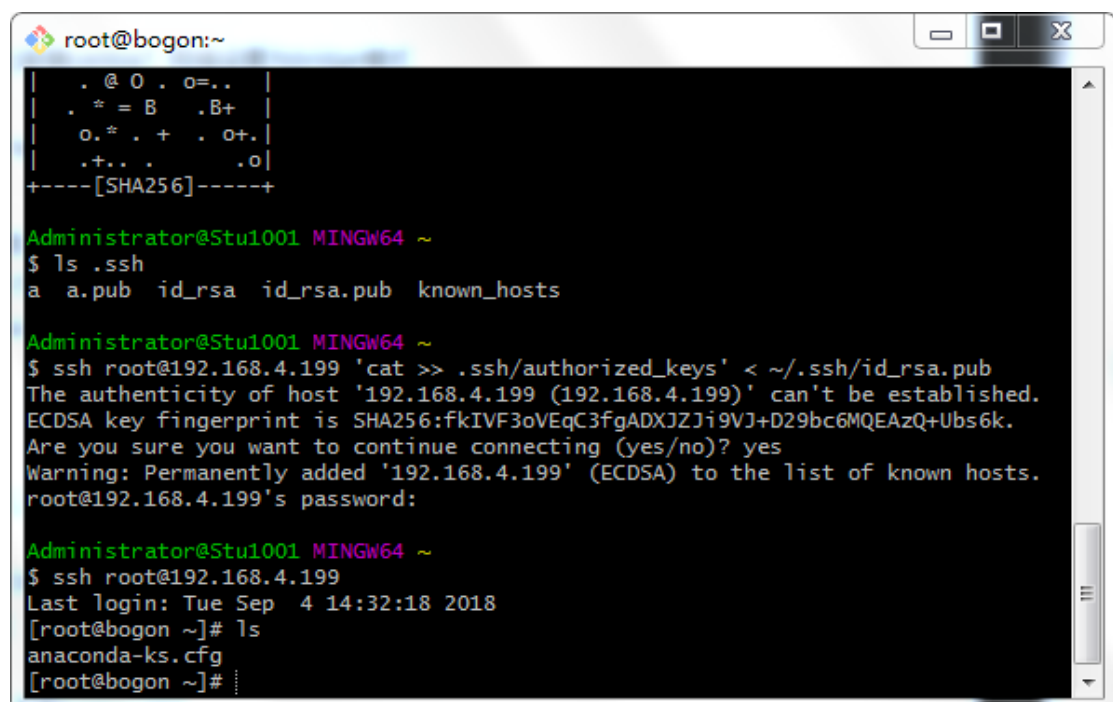
生成公钥和密钥，用于与本机的传递，可以免密登录



将网卡设置为桥接网卡，这样后面的 IP 地址就不会发生改变



将公钥送入本机



## JDK 和 HADOOP 压缩文件的传递与解压

```
root@bogon:~  
$ scp jdk-8u144-linux-x64.tar.gz root@192.168.4.199ssh: connect to host 192.168.4.199:22: Connection timed out  
lost connection  
  
Administrator@Stu1001 MINGW64 /c/Program Files/feiq/Recv Files/bigdata(3)  
$ scp jdk-8u144-linux-x64.tar.gz root@192.168.4.199:~/.  
jdk-8u144-linux-x64.tar.gz 100% 177MB 36.6MB/s 00:04  
  
Administrator@Stu1001 MINGW64 /c/Program Files/feiq/Recv Files/bigdata(3)  
$ scp jdk-8u144-linux-x64.tar.gz root@192.168.3.181:~/.  
[1]+ Stopped scp jdk-8u144-linux-x64.tar.gz root@192.168.3.181:~/.  
~/.  
  
Administrator@Stu1001 MINGW64 /c/Program Files/feiq/Recv Files/bigdata(3)  
$ scp hadoop-3.0.0.tar.gz root@192.168.4.199:~/.  
hadoop-3.0.0.tar.gz 100% 292MB 42.2MB/s 00:06  
  
Administrator@Stu1001 MINGW64 /c/Program Files/feiq/Recv Files/bigdata(3)  
$ ssh root@192.168.4.199  
Last login: Tue Sep 4 14:53:22 2018 from 192.168.4.176  
[root@bogon ~]# ls  
anaconda-ks.cfg hadoop-3.0.0.tar.gz jdk-8u144-linux-x64.tar.gz  
[root@bogon ~]#
```

```
root@bogon:~  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/rmid.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/jconsole.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/jdeps.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/jinfo.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/jdb.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/jmap.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/javac.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/wsimport.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/tnameserv.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/pack200.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/jcmd.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/javapackager.1  
jdk1.8.0_144/man/ja_JP.UTF-8/man1/jstat.1  
[root@bogon ~]# dir  
anaconda-ks.cfg hadoop-3.0.0.tar.gz jdk1.8.0_144 jdk-8u144-linux-x64.tar.gz  
[root@bogon ~]# ln -s jdk1.8.0_144/ jdk1.8  
[root@bogon ~]# vi etc/profile  
[root@bogon ~]# vi /etc/profile  
[root@bogon ~]# source /etc/profile  
[root@bogon ~]# java -version  
java version "1.8.0_144"  
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)  
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)  
[root@bogon ~]#
```

```
root@bogon:~/hadoop-3.0.0
hadoop-3.0.0/etc/hadoop/kms-acls.xml
hadoop-3.0.0/etc/hadoop/hadoop-policy.xml
hadoop-3.0.0/etc/hadoop/capacity-scheduler.xml
hadoop-3.0.0/etc/hadoop/shellprofile.d/
hadoop-3.0.0/etc/hadoop/shellprofile.d/example.sh
hadoop-3.0.0/etc/hadoop/mapred-queues.xml.template
hadoop-3.0.0/etc/hadoop/yarn-env.sh
hadoop-3.0.0/etc/hadoop/container-executor.cfg
hadoop-3.0.0/etc/hadoop/hadoop-metrics2.properties
hadoop-3.0.0/etc/hadoop/mapred-env.cmd
hadoop-3.0.0/etc/hadoop/workers
hadoop-3.0.0/etc/hadoop/https-env.sh
hadoop-3.0.0/etc/hadoop/hdfs-site.xml
hadoop-3.0.0/etc/hadoop/user_ec_policies.xml.template
[root@bogon ~]# ls
anaconda-ks.cfg  hadoop-3.0.0.tar.gz  jdk1.8.0_144
hadoop-3.0.0     jdk1.8               jdk-8u144-linux-x64.tar.gz
[root@bogon ~]# ln -s hadoop-3.0.0 hadoop
[root@bogon ~]# ls
anaconda-ks.cfg  hadoop-3.0.0         jdk1.8          jdk-8u144-linux-x64.tar.gz
hadoop           hadoop-3.0.0.tar.gz  jdk1.8.0_144
[root@bogon ~]# ssh localhost
Last login: Tue Sep  4 15:21:50 2018 from ::1
[root@bogon ~]# cd hadoop
```

```
root@bogon:~/hadoop/etc/hadoop
[root@bogon hadoop]# cd etc
[root@bogon etc]# ls
hadoop
[root@bogon etc]# cd hadoop/
[root@bogon hadoop]# ls
capacity-scheduler.xml      kms-log4j.properties
configuration.xml           kms-site.xml
container-executor.cfg      log4j.properties
core-site.xml              mapred-env.cmd
hadoop-env.cmd             mapred-env.sh
hadoop-env.sh              mapred-queues.xml.template
hadoop-metrics2.properties mapred-site.xml
hadoop-policy.xml          shellprofile.d
hadoop-user-functions.sh.example  ssl-client.xml.example
hdfs-site.xml              ssl-server.xml.example
https-env.sh               user_ec_policies.xml.template
https-log4j.properties     workers
https-signature.secret     yarn-env.cmd
https-site.xml             yarn-env.sh
kms-acls.xml               yarn-site.xml
kms-env.sh
[root@bogon hadoop]# cp ~/hadoopfiles/* .
cp: 是否覆盖"./capacity-scheduler.xml"? y
cp: 是否覆盖"./configuration.xml"? y
```



在 data 文件夹下面创建 namenode 和 datanode

```
root@bogon:~/hadoop/data

Administrator@Stu1001 MINGW64 /
$ ssh root
ssh root@1ssh: Could not resolve hostname root: Name or service not known

Administrator@Stu1001 MINGW64 /
$ ssh root@192.168.4.199

Last login: Tue Sep  4 15:37:57 2018 from 192.168.4.176
[root@bogon ~]#
[root@bogon ~]# cd hadoop
[root@bogon hadoop]# mkdir data
[root@bogon hadoop]# cd data
[root@bogon data]# mkdir namenode
[root@bogon data]# ls
namenode
[root@bogon data]# mkdir datanode
[root@bogon data]# ls
datanode namenode
[root@bogon data]#
```

对~/hadoop/etc 目录下的 profile 文件进行编辑,在尾部加上以下代码

```
export JAVA_HOME=~/.jdk1.8
export PATH=$PATH:$JAVA_HOME/bin
```

目录下配置环境需求的配置文件

在 ~/hadoop/etc/hadoop/目录下配置环境需求的配置文件

```
[root@localhost hadoop]# cd etc/
[root@localhost hadoop]# ls
capacity-scheduler.xml  hadoop-env.sh  httpfs-env.sh  kms-env.sh  mapred-env.sh
configuration.xml       hadoop-metrics2.properties  httpfs-log4j.properties  kms-log4j.properties  mapred-queue.properties
container-executor.cfg  hadoop-policy.xml  httpfs-signature.secret  kms-site.xml  mapred-site.xml
core-site.xml           hadoop-user-functions.sh.example  httpfs-site.xml  log4j.properties  shellprofile
hadoop-env.cmd          hdfs-site.xml  kms-acls.xml  mapred-env.cmd  ssl-client.x
```

编辑 core-site.xml,在以下位置加入该代码



```
<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
</property>
</configuration>
```

编辑 hdfs-site.xml, 在以下位置加入该代码

```
<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

编辑 hadoop-env.sh, 在以下位置加入该代码

```
# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=/root/jdk1.8
```

目录下配置环境需求的配置文件

在~/hadoop 目录下配置环境需求的配置文件编辑 sbin/start-dfs.sh 和 sbin/stop-dfs.sh 的配置文件

```
# limitations under the License.

HDFS_NAMENODE_USER=root
HDFS_DATANODE_USER=root
HDFS_SECONDARYNAMENODE_USER=root

# Start hadoop dfs daemons.
# Optionally upgrade or rollback dfs state.
# Run this on master node.

## startup-matrix
```

编辑 sbin/start-yarn.sh 和 sbin/stop-yarn.sh 的配置文件, 在该位置加上以下 2 条代码

```
# limitations under the License.

YARN_RESOURCEMANAGER_USER=root
YARN_NODEMANAGER_USER=root

## @description  usage info
## @audience     private
```

在 hadoop 目录下运行以下命令清空 namenode 节点的内容

```
[root@localhost hadoop]# bin/hdfs namenode -format
```

之后运行 sbin/start-dfs.sh 与 sbin/start-yarn.sh 命令去产生新节点(每次重启都需运行)

```
[root@localhost hadoop]# sbin/start-dfs.sh
```

环境配置完成后,在 hadoop 目录下运行 jps 会出现四个节点,代表环境配置成功。

```
[root@localhost hadoop]# jps
1728 DataNode
5585 Jps
1605 NameNode
1979 SecondaryNameNode
[root@localhost hadoop]#
```

```
[root@localhost hadoop]# bin/hdfs dfs -cat /w/o/*
安溪--2018-01 白天平均温度是17℃ 夜间平均温度是10℃
安溪--2018-02 白天平均温度是18℃ 夜间平均温度是9℃
安溪--2018-03 白天平均温度是24℃ 夜间平均温度是13℃
安溪--2018-04 白天平均温度是28℃ 夜间平均温度是17℃
安溪--2018-05 白天平均温度是31℃ 夜间平均温度是22℃
安溪--2018-06 白天平均温度是30℃ 夜间平均温度是23℃
闽侯--2018-01 白天平均温度是15℃ 夜间平均温度是8℃
闽侯--2018-02 白天平均温度是16℃ 夜间平均温度是7℃
闽侯--2018-03 白天平均温度是22℃ 夜间平均温度是12℃
闽侯--2018-04 白天平均温度是26℃ 夜间平均温度是16℃
闽侯--2018-05 白天平均温度是31℃ 夜间平均温度是22℃
闽侯--2018-06 白天平均温度是30℃ 夜间平均温度是23℃
```

实现 3 台虚拟机的相连：

编辑 etc 目录下的 hosts 文件，除了本机加上其他两个虚拟机的地址与命名

```
[root@slave2 ~]# vi /etc/hosts
```

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.4.212 master
192.168.4.229 slave2
192.168.4.226 slave1
```

编辑 etc 目录下的网络配置,加上主机名呢 GATEWAY 以及 NETWORKING

```
[root@slave2 ~]# vi /etc/sysconfig/network
[root@slave2 ~]#
```

```
# Created by anaconda
NETWORKING=yes
HOSTNAME=slave2
GATEWAY=192.168.4.1
```

将本机的公钥导入到其余两台虚拟机中,分别为 master 和 slave1 , 在导入公钥前先清除.ssh 目录下的 uknow\_host

```
[root@slave2 ~]# cd .ssh
[root@slave2 .ssh]# ls
authorized_keys  id_rsa.pub  known_hosts
[root@slave2 .ssh]# rm know_hosts
```

```
[root@slave2 ~]# ssh root@master 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
[root@slave2 ~]# ssh root@slave1 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

上述步骤需同时在其余两台虚拟机上进行相同的配置

实现免密访问后在 master 端进行操作,可分别进入 slave1 和 slave2 端进行文件的删除、修改和创建,通过以下截图可发现进入 slave1 和 slave2 后成功创建了 slave1.txt 和 slave2.txt 文件,至此已成功实现 3 台虚拟机的互连。

```
[root@slave2 ~]# ssh slave1
Last login: Tue Sep  4 13:43:22 2018 from 192.168.4.212
[root@slave1 ~]# ls
anaconda-ks.cfg  hadoop-3.0.0  hadoopfiles  jdk1.8  jdk-8u144-linux-x64.tar.gz  weatherall.txt  wordcount-0.0.1.jar
hadoop           hadoop-3.0.0.tar.gz  hadoopfiles.zip  jdk1.8.0_144  weather-0.0.1.jar  weather.txt
[root@slave1 ~]# mkdir slave1.txt
[root@slave1 ~]# ls
anaconda-ks.cfg  hadoop-3.0.0  hadoopfiles  jdk1.8  jdk-8u144-linux-x64.tar.gz  weather-0.0.1.jar  weather.txt  wordcount-0.0.1.jar
hadoop           hadoop-3.0.0.tar.gz  hadoopfiles.zip  jdk1.8.0_144  slave1.txt  weatherall.txt  wordcount-0.0.1.jar
```

```
[root@slave1 ~]# ssh slave2
Last login: Tue Sep  4 13:45:17 2018 from 192.168.4.202
[root@slave2 ~]# ls
anaconda-ks.cfg  hadoop-3.0.0  hadoopfiles  jdk1.8  jdk-8u144-linux-x64.tar.gz  weather-0.0.1.jar  weather.txt  xiamen.txt
a.txt            hadoop-3.0.0.tar.gz  hadoopfiles.zip  jdk1.8  jdk-8u144-linux-x64.tar.gz  weatherall.txt  wordcount-0.0.1.jar
[root@slave2 ~]# mkdir slave2.txt
[root@slave2 ~]# ls
anaconda-ks.cfg  hadoop-3.0.0  hadoopfiles  jdk1.8  jdk-8u144-linux-x64.tar.gz  slave2.txt  weather-0.0.1.jar  weather.txt  xiamen.txt
a.txt            hadoop-3.0.0.tar.gz  hadoopfiles.zip  jdk1.8.0_144  weatherall.txt  wordcount-0.0.1.jar
```

```
[root@slave2 ~]# ssh slave1
Last login: Tue Sep  4 13:43:22 2018 from 192.168.4.212
[root@slave1 ~]# ls
anaconda-ks.cfg  hadoop-3.0.0  hadoopfiles  jdk1.8  jdk-8u144-linux-x64.tar.gz  weatherall.txt  wordcount-0.0.1.jar
hadoop           hadoop-3.0.0.tar.gz  hadoopfiles.zip  jdk1.8.0_144  weather-0.0.1.jar  weather.txt
[root@slave1 ~]# mkdir slave1.txt
[root@slave1 ~]# ls
anaconda-ks.cfg  hadoop-3.0.0  hadoopfiles  jdk1.8  jdk-8u144-linux-x64.tar.gz  weather-0.0.1.jar  weather.txt  wordcount-0.0.1.jar
hadoop           hadoop-3.0.0.tar.gz  hadoopfiles.zip  jdk1.8.0_144  slave1.txt  weatherall.txt  wordcount-0.0.1.jar
Last login: Tue Sep  4 13:45:17 2018 from 192.168.4.202
[root@slave2 ~]# ls
anaconda-ks.cfg  hadoop-3.0.0  hadoopfiles  jdk1.8  jdk-8u144-linux-x64.tar.gz  weather-0.0.1.jar  weather.txt  xiamen.txt
a.txt            hadoop-3.0.0.tar.gz  hadoopfiles.zip  jdk1.8  jdk-8u144-linux-x64.tar.gz  weatherall.txt  wordcount-0.0.1.jar
[root@slave2 ~]# mkdir slave2.txt
[root@slave2 ~]# ls
anaconda-ks.cfg  hadoop-3.0.0  hadoopfiles  jdk1.8  jdk-8u144-linux-x64.tar.gz  slave2.txt  weather-0.0.1.jar  weather.txt  xiamen.txt
a.txt            hadoop-3.0.0.tar.gz  hadoopfiles.zip  jdk1.8.0_144  weatherall.txt  wordcount-0.0.1.jar
```

### 3、数据分析

#### (1) 数据分析代码：

#### JAVA 代码：

Map阶段的代码：

```
protected void map(LongWritable key, Text value,
Mapper<LongWritable, Text, Text, Text>.Context context)
    throws IOException, InterruptedException {
    String line = value.toString();//按行获取字符引用
    String[] words = line.split(" ");//以空格分离获取的当行数据
    String id = words[0];//获取第一个的数据
    String cityname = words[1];//获取第二个数据
    String datetime = StringUtils.substringBeforeLast(words[2], "-");
    //以StringUtils对象的方法去取出当前日期的月份
    String[] date = datetime.split("-");//分离日期，拆成年月
    String tmp=StringUtils.substringBefore(words[5], "°C");
    //取出温度的整型
    String tmp1=StringUtils.substringBefore(words[6], "°C");
    //取出温度的整型
    context.write(new Text(cityname+" "+date[0]+"年"+date[1]+"月"), new
Text(tmp+"--"+tmp1));
    //将MAP后的数据作为Reduce的输出，数据key为城市与时间，values值为对应日间
    与夜间的温度
}

}
```

Reduce 阶段的代码：

```
protected void reduce(Text key, Iterable<Text> values,
Reducer<Text, Text, Text, Text>.Context context)
```

```

throws IOException, InterruptedException {

    Integer tsum=0;
    Integer nsum=0;
    String s=null;
    int tavgtemperature=0;
    int navgtemperature=0;
    Text t=null;
    List<Integer> list = new ArrayList();
    int i=0;
    for(Text value : values) {
        s=value.toString(); //单行数据的引用
        String[] words = s.split("--"); //分割取出白天温度与
        夜间温度

        tsum += Integer.parseInt(words[0]); //对白天温度进行累加
        nsum += Integer.parseInt(words[1]);
        list.add(Integer.parseInt(words[0]));
        list.add(Integer.parseInt(words[1]));
        i++;
    }
    int min= Collections.min(list); //取出List集合中的最小值
    int max= Collections.max(list); //取出List集合中的最大值
    tavgtemperature=tsum/i; //计算白天平均温度
    navgtemperature=nsum/i; //计算夜间平均温度
    t =new Text("白天平均温度是"+tavgtemperature+"℃      夜
    间平均温度是"+navgtemperature+"℃"+"      本月最高温度是"+max+"℃
    本月最低温度是"+min+"℃"); //按该形式输出

    context.write(key,new Text(t)); //Reduce阶段输出的key为Map
    阶段的输出key, values为上述t的格式

}

}

```

MapReduce 阶段：

```

public class WordCountMapReduce
{
    public static void main( String[] args ) throws Exception
    {
        Configuration cfg = new Configuration();
        Job job = Job.getInstance(cfg, "wordcount");
        job.setJarByClass(WordCountMapReduce.class);

        FileInputFormat.setInputPaths(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
    }
}

```

```

        job.setMapperClass(WordCountMapper.class);
        job.setReducerClass(WordCountReducer.class);

        job.setMapOutputKeyClass(Text.class);
        job.setMapOutputValueClass(Text.class);

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

        boolean b = job.waitForCompletion(true);
        if(!b) {
            System.out.println("wordcount task fail!");
        }
    }
}

```

## 分布式环境运行：

将 JAVA 代码的 JAR 包导入到虚拟机中

```

Administrator@Stu1001 MINGW64 /e/feiq/Recv Files/weathercount/target
$ scp weather-0.0.1.jar root@192.168.4.212:~/
weather-0.0.1.jar                                100% 7036      1.8MB/s   00:00

```

将 JAVA 代码生成的数据导入到虚拟机中

```

Administrator@Stu1001 MINGW64 /e/bigdata
$ scp weather.txt root@192.168.4.212:~/
weather.txt                                       100%   73KB   14.9MB/s   00:00

```

在 hadoop 目录下创建/weathercount/input 目录,并将文本文件传入该目录,用导入的 JAR 包去运行该文件,并将生成的文件放置在/weathercount/output 目录下

```

[root@localhost hadoop]# bin/hdfs dfs -mkdir -p /weathercount/input
[root@localhost hadoop]# bin/hdfs dfs -put ~/weather.txt /weathercount/input
[root@localhost hadoop]# bin/hadoop jar ~/weather-0.0.1.jar demo.mr.wordcount.WordCountMapReduce /weathercount/input /weathercount/output

```

## (2) 分析的数据：

在 hadoop 目录下,运行 bin/hdfs dfs -cat /weathercount/output/\*

查看分析后的数据：



只统计白天与夜间的平均温度

```
上海 2018年01月 白天平均温度是7℃ 夜间平均温度是2℃
上海 2018年02月 白天平均温度是9℃ 夜间平均温度是2℃
上海 2018年03月 白天平均温度是16℃ 夜间平均温度是9℃
上海 2018年04月 白天平均温度是22℃ 夜间平均温度是14℃
上海 2018年05月 白天平均温度是26℃ 夜间平均温度是19℃
上海 2018年06月 白天平均温度是28℃ 夜间平均温度是22℃
北京 2018年01月 白天平均温度是1℃ 夜间平均温度是-6℃
北京 2018年02月 白天平均温度是4℃ 夜间平均温度是-5℃
北京 2018年03月 白天平均温度是14℃ 夜间平均温度是2℃
北京 2018年04月 白天平均温度是20℃ 夜间平均温度是9℃
北京 2018年05月 白天平均温度是27℃ 夜间平均温度是16℃
北京 2018年06月 白天平均温度是32℃ 夜间平均温度是21℃
哈尔滨 2018年01月 白天平均温度是-14℃ 夜间平均温度是-24℃
哈尔滨 2018年02月 白天平均温度是-9℃ 夜间平均温度是-21℃
哈尔滨 2018年03月 白天平均温度是2℃ 夜间平均温度是-9℃
哈尔滨 2018年04月 白天平均温度是14℃ 夜间平均温度是3℃
哈尔滨 2018年05月 白天平均温度是22℃ 夜间平均温度是10℃
哈尔滨 2018年06月 白天平均温度是26℃ 夜间平均温度是16℃
安溪 2018年01月 白天平均温度是17℃ 夜间平均温度是10℃
安溪 2018年02月 白天平均温度是18℃ 夜间平均温度是9℃
安溪 2018年03月 白天平均温度是24℃ 夜间平均温度是13℃
安溪 2018年04月 白天平均温度是28℃ 夜间平均温度是17℃
安溪 2018年05月 白天平均温度是31℃ 夜间平均温度是22℃
安溪 2018年06月 白天平均温度是30℃ 夜间平均温度是23℃
闽侯 2018年01月 白天平均温度是15℃ 夜间平均温度是8℃
闽侯 2018年02月 白天平均温度是16℃ 夜间平均温度是7℃
闽侯 2018年03月 白天平均温度是22℃ 夜间平均温度是12℃
闽侯 2018年04月 白天平均温度是26℃ 夜间平均温度是16℃
闽侯 2018年05月 白天平均温度是31℃ 夜间平均温度是22℃
闽侯 2018年06月 白天平均温度是30℃ 夜间平均温度是23℃
[root@localhost hadoop]#
```

统计五个城市每个月的白天与夜间平均温度、单月最高气温与单月最低气温

```
[root@localhost hadoop]# bin/hdfs dfs -cat /w/o/*
上海 2018年01月 白天平均温度是7℃ 夜间平均温度是2℃ 本月最高温度是15℃ 本月最低温度是-2℃
上海 2018年02月 白天平均温度是9℃ 夜间平均温度是2℃ 本月最高温度是17℃ 本月最低温度是-2℃
上海 2018年03月 白天平均温度是16℃ 夜间平均温度是9℃ 本月最高温度是25℃ 本月最低温度是2℃
上海 2018年04月 白天平均温度是22℃ 夜间平均温度是14℃ 本月最高温度是30℃ 本月最低温度是7℃
上海 2018年05月 白天平均温度是26℃ 夜间平均温度是19℃ 本月最高温度是35℃ 本月最低温度是14℃
上海 2018年06月 白天平均温度是28℃ 夜间平均温度是22℃ 本月最高温度是36℃ 本月最低温度是18℃
北京 2018年01月 白天平均温度是1℃ 夜间平均温度是-6℃ 本月最高温度是7℃ 本月最低温度是-12℃
北京 2018年02月 白天平均温度是4℃ 夜间平均温度是-5℃ 本月最高温度是12℃ 本月最低温度是-10℃
北京 2018年03月 白天平均温度是14℃ 夜间平均温度是2℃ 本月最高温度是27℃ 本月最低温度是-4℃
北京 2018年04月 白天平均温度是20℃ 夜间平均温度是9℃ 本月最高温度是29℃ 本月最低温度是0℃
北京 2018年05月 白天平均温度是27℃ 夜间平均温度是16℃ 本月最高温度是34℃ 本月最低温度是9℃
北京 2018年06月 白天平均温度是32℃ 夜间平均温度是21℃ 本月最高温度是39℃ 本月最低温度是15℃
哈尔滨 2018年01月 白天平均温度是-14℃ 夜间平均温度是-24℃ 本月最高温度是-2℃ 本月最低温度是-35℃
哈尔滨 2018年02月 白天平均温度是-9℃ 夜间平均温度是-21℃ 本月最高温度是-2℃ 本月最低温度是-30℃
哈尔滨 2018年03月 白天平均温度是2℃ 夜间平均温度是-9℃ 本月最高温度是21℃ 本月最低温度是-24℃
哈尔滨 2018年04月 白天平均温度是14℃ 夜间平均温度是3℃ 本月最高温度是27℃ 本月最低温度是-6℃
哈尔滨 2018年05月 白天平均温度是22℃ 夜间平均温度是10℃ 本月最高温度是31℃ 本月最低温度是4℃
哈尔滨 2018年06月 白天平均温度是26℃ 夜间平均温度是16℃ 本月最高温度是37℃ 本月最低温度是13℃
安溪 2018年01月 白天平均温度是17℃ 夜间平均温度是10℃ 本月最高温度是25℃ 本月最低温度是3℃
安溪 2018年02月 白天平均温度是18℃ 夜间平均温度是9℃ 本月最高温度是26℃ 本月最低温度是1℃
安溪 2018年03月 白天平均温度是24℃ 夜间平均温度是13℃ 本月最高温度是29℃ 本月最低温度是7℃
安溪 2018年04月 白天平均温度是28℃ 夜间平均温度是17℃ 本月最高温度是34℃ 本月最低温度是9℃
安溪 2018年05月 白天平均温度是31℃ 夜间平均温度是22℃ 本月最高温度是36℃ 本月最低温度是18℃
安溪 2018年06月 白天平均温度是30℃ 夜间平均温度是23℃ 本月最高温度是36℃ 本月最低温度是20℃
闽侯 2018年01月 白天平均温度是15℃ 夜间平均温度是8℃ 本月最高温度是23℃ 本月最低温度是2℃
闽侯 2018年02月 白天平均温度是16℃ 夜间平均温度是7℃ 本月最高温度是24℃ 本月最低温度是-1℃
闽侯 2018年03月 白天平均温度是22℃ 夜间平均温度是12℃ 本月最高温度是29℃ 本月最低温度是4℃
闽侯 2018年04月 白天平均温度是26℃ 夜间平均温度是16℃ 本月最高温度是32℃ 本月最低温度是8℃
闽侯 2018年05月 白天平均温度是31℃ 夜间平均温度是22℃ 本月最高温度是38℃ 本月最低温度是15℃
闽侯 2018年06月 白天平均温度是30℃ 夜间平均温度是23℃ 本月最高温度是38℃ 本月最低温度是19℃
[root@localhost hadoop]#
```