

# Hadoop 数据分析

## 一、数据获取

### 1. 代码

```
String date = "";
for (int dtm = -185; dtm < -150; dtm++) { // 循环获取前185天起35天的天气信息
    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, dtm); // 取当前日期的前N天。
    date = format.format(cal.getTime());
    String res = GetCityList.weather("120", date); // 查询所需的天气
    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, 得到每个属性
    obj = JSONObject.fromObject(result); // 今日温度对应的key是today
    String city_id = obj.getString("city_id"); // 城市id
    String city_name = obj.getString("city_name"); // 城市名称
    String weather_date = obj.getString("weather_date"); // 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);

    File file1 = new File("E:\\weather\\weather.txt"); // 数据写入文件中
    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();
        }
        bw.newLine(); bw.close(); fw.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

## 2. 获取结果

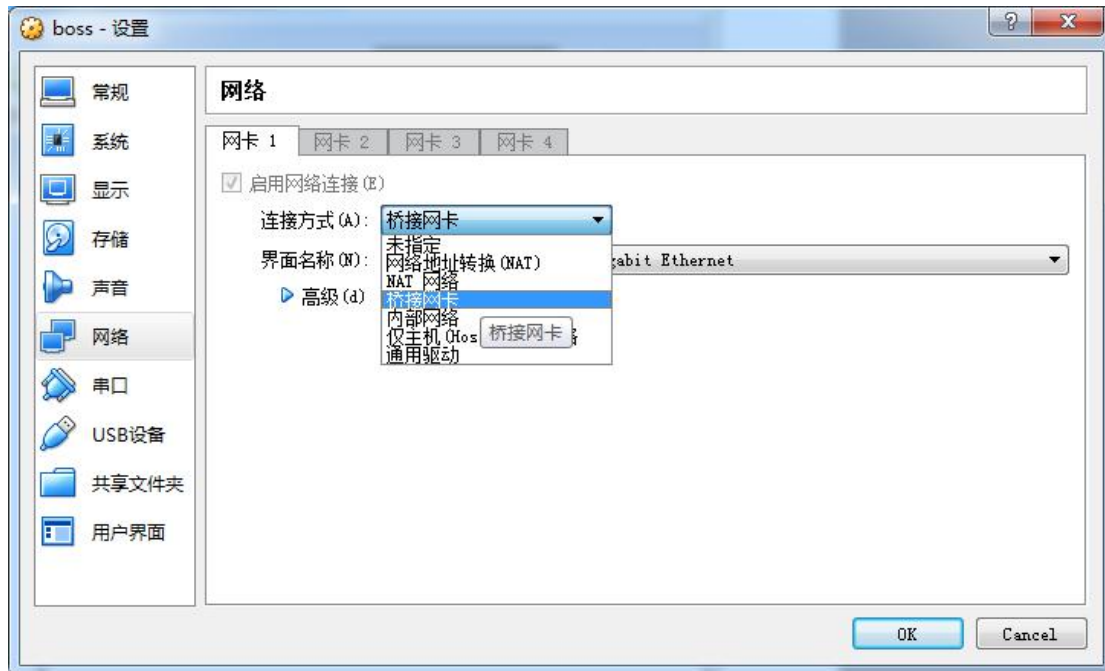
weather.txt - 记事本									
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)									
120	奥清	2018-03-04	阴 多云	31℃ 19℃	微风 <3级	微风 <3级	02	01	
120	奥清	2018-03-05	阴 阴	26℃ 13℃	西北风 <3级	东南风 <3级	02	02	
120	奥清	2018-03-06	阴 多云	18℃ 11℃	东风 3-4级	东南风 <3级	02	01	
120	奥清	2018-03-07	阴 阴	21℃ 11℃	东南风 <3级	西风 <3级	02	02	
120	奥清	2018-03-08	阴 多云	15℃ 5℃	西北风 3-4级	西北风 <3级	02	01	
120	奥清	2018-03-09	晴 多云	18℃ 4℃	东风 <3级	东南风 <3级	00	01	
120	奥清	2018-03-10	多云 雾	21℃ 4℃	东南风 <3级	西南风 <3级	01	18	
120	奥清	2018-03-11	晴 雾	23℃ 6℃	东南风 <3级	西南风 <3级	00	18	
120	奥清	2018-03-12	雾 多云	26℃ 11℃	东风 <3级	西风 <3级	18	01	
120	奥清	2018-03-13	多云 多云	25℃ 12℃	东风 <3级	西风 <3级	01	01	
120	奥清	2018-03-14	阴 阴	25℃ 14℃	东风 <3级	北风 <3级	02	02	
120	奥清	2018-03-15	阴 阴	21℃ 16℃	北风 <3级	西北风 <3级	02	02	
120	奥清	2018-03-16	阴 阴	23℃ 12℃	东南风 <3级	东南风 <3级	02	02	
120	奥清	2018-03-17	阴 多云	21℃ 13℃	东南风 <3级	东南风 <3级	02	01	
120	奥清	2018-03-18	多云 阴	26℃ 15℃	东北风 <3级	西北风 <3级	01	02	
120	奥清	2018-03-19	阴 阴	26℃ 14℃	西北风 <3级	东南风 <3级	02	02	
120	奥清	2018-03-20	阴 阴	15℃ 9℃	西北风 <3级	西风 3-4级	02	02	
120	奥清	2018-03-21	多云 多云	22℃ 8℃	西北风 3-4级	西北风 <3级	01	01	
120	奥清	2018-03-22	晴 多云	22℃ 8℃	东风 <3级	西风 <3级	00	01	
120	奥清	2018-03-23	多云 多云	25℃ 9℃	东风 <3级	西风 <3级	01	01	
120	奥清	2018-03-24	多云 阴	24℃ 12℃	东南风 <3级	西风 <3级	01	02	
120	奥清	2018-03-25	阴 阴	22℃ 15℃	东风 <3级	西风 <3级	02	02	
120	奥清	2018-03-26	阴 阴	22℃ 14℃	西北风 <3级	南风 <3级	02	02	
120	奥清	2018-03-27	多云 阴	24℃ 14℃	西南风 <3级	东风 <3级	01	02	
120	奥清	2018-03-28	多云 多云	28℃ 14℃	东南风 <3级	西风 <3级	01	01	
120	奥清	2018-03-29	多云 多云	25℃ 16℃	东风 <3级	北风 <3级	01	01	
120	奥清	2018-03-30	多云 多云	25℃ 15℃	微风 <3级	微风 <3级	01	01	
120	奥清	2018-03-31	多云 多云	28℃ 12℃	东南风 <3级	西风 <3级	01	01	
120	奥清	2018-04-01	多云 多云	30℃ 14℃	东南风 <3级	西风 <3级	01	01	
120	奥清	2018-04-02	多云 阴	29℃ 17℃	东风 <3级	东南风 <3级	01	02	
120	奥清	2018-04-03	多云 多云	30℃ 19℃	微风 <3级	微风 <3级	01	01	
120	奥清	2018-04-04	阴 多云	30℃ 20℃	南风 <3级	东风 <3级	02	01	
120	奥清	2018-04-05	阴 阴	29℃ 19℃	东风 <3级	西风 <3级	02	02	
120	奥清	2018-04-06	阴 阴	22℃ 13℃	西北风 3-4级	西北风 <3级	02	02	
120	奥清	2018-04-07	阴 阴	18℃ 7℃	微风 <3级	微风 <3级	02	02	

## 二、Hadoop 环境搭建

### 1. 虚拟机安装（Linux）



## 2. 虚拟机配置



```
base login: Thu Sep 8 15:10:31 08 ptc/0  
[root@master ~]# yum install net-tools_
```

```
[root@master ~]# ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.4.197 netmask 255.255.255.0 broadcast 192.168.4.255  
    inet6 fe80::9bce:4b56:a79b:f8c2 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:5e:c1:2c txqueuelen 1000 (Ethernet)  
    RX packets 54 bytes 5681 (5.5 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 60 bytes 5286 (5.1 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1 (Local Loopback)  
    RX packets 64 bytes 5568 (5.4 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 64 bytes 5568 (5.4 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
[root@master ~]#
```

```
root@master:~  
127.0.0.1 localhost localhost.1  
::1 localhost localhost.1  
  
192.168.4.197 master  
192.168.4.224 man001  
[root@master ~]# vi /etc/hosts
```

## 3. 本地传输文件到虚拟机并安转 (jdk、hadoop)

```
Administrator@PC201803080955 MINGW64 /e/data (master)  
$ scp jdk-8u144-linux-x64.tar.gz root@192.168.4.197:~/.
```



```
[root@master ~]# tar xvf jdk-8u144-linux-x64.tar.gz |
```

```
[root@master ~]# ln -s jdk1.8.0_144/ jdk 1.8
```

```
[root@master ~]# vi /etc/profile //编辑 profile 文件
```

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

//设置 jdk 的环境变量

```
[root@master ~]# source /etc/profile //使其生效
```

```
[root@master ~]# 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@master ~]# |
```

//jdk 环境搭建成功

```
Administrator@PC201803080955 MINGW64 /e/data (master)
$ scp hadoop-3.0.0.tar.gz root@192.168.4.197:~/.
```

```
[root@master ~]# hadoop
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
or hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
where CLASSNAME is a user-provided Java class

OPTIONS is none or any of:
```

//同 jdk 环境搭建

```
Administrator@PC201803080955 MINGW64 /e/data (master)
$ scp hadoopfiles.zip| root@192.168.4.197:~/.
```

```
[root@master ~]# yum install unzip
```

```
[root@master ~]# unzip hadoopfiles.zip |
```

## 4. Hadoop 环境设置

```
# limitations under the License.

HDFS_NAMENODE_USER=root
HDFS_DATANODE_USER=root
HDFS_SECONDARYNAMENODE_USER=root

# Start hadoop dfs daemons.
```

```
[root@master ~]# vi sbin/start-dfs.sh
```

```
[root@master hadoop]# vi sbin/stop-dfs.sh
```

```
[root@master hadoop]# vi sbin/start-yarn.sh
```

```
[root@master hadoop]# vi sbin/stop-yarn.sh
```

```
# limitations under the License.

HDFS_NAMENODE_USER=root
HDFS_DATANODE_USER=root
HDFS_SECONDARYNAMENODE_USER=root

# Stop hadoop dfs daemons.

# limitations under the License.

YARN_RESOURCEMANAGER_USER=root
YARN_NODEMANAGER_USER=root

## @description usage info
# limitations under the License.

YARN_RESOURCEMANAGER_USER=root
YARN_NODEMANAGER_USER=root

## @description usage info
```

## 5. Hadoop 启动

```
[root@master hadoop]# sbin/start-all.sh
Starting namenodes on [master]
上一次登录: 四 9月 6 16:01:13 CST 2018从 192.168.4.160pts/1 上
Starting datanodes
上一次登录: 四 9月 6 16:15:00 CST 2018pts/0 上
Starting secondary namenodes [master]
上一次登录: 四 9月 6 16:15:05 CST 2018pts/0 上
Starting resourcemanager
上一次登录: 四 9月 6 16:15:20 CST 2018pts/0 上
Starting nodemanagers
上一次登录: 四 9月 6 16:15:45 CST 2018pts/0 上
[root@master hadoop]# jps
2241 SecondaryNameNode
1794 NameNode
1910 DataNode
2487 ResourceManager
2903 Jps
2607 NodeManager
[root@master hadoop]# |
```

## 三、Hadoop 项目应用

### 1. 将项目的生成 jar 包和天气数据传输到 Linux 目录下

```
Administrator@PC201803080955 MINGW64 /e/data (master)
$ scp WeatherWork.jar root@192.168.4.197:~/hadoop/share/hadoop/mapreduce/.
WeatherWork.jar 100% 17KB 678.0KB/s 00:00
Administrator@PC201803080955 MINGW64 /e/data (master)
$ |
```

```
[root@master mapreduce]# ls
hadoop-mapreduce-client-app-3.0.0.jar
hadoop-mapreduce-client-common-3.0.0.jar
hadoop-mapreduce-client-core-3.0.0.jar
hadoop-mapreduce-client-hs-3.0.0.jar
hadoop-mapreduce-client-hs-plugins-3.0.0.jar
hadoop-mapreduce-client-jobclient-3.0.0.jar
hadoop-mapreduce-client-jobclient-3.0.0-tests.jar
hadoop-mapreduce-client-nativetask-3.0.0.jar
hadoop-mapreduce-client-shuffle-3.0.0.jar
hadoop-mapreduce-examples-3.0.0.jar
jdifff
lib-examples
sources
WeatherWork.jar
[root@master mapreduce]#
```

```
Administrator@PC201803080955 MINGW64 /e/data (master)
$ scp weather.txt root@192.168.4.197:~/hadoop/share/hadoop/hdfs
weather.txt                                100% 3168    839.5KB/s   00:00
```

## 2. 运行 hadoop

```
[root@master hdfs]# ls
hadoop-hdfs-3.0.0.jar                hadoop-hdfs-nfs-3.0.0.jar
hadoop-hdfs-3.0.0-tests.jar          jdifff
hadoop-hdfs-client-3.0.0.jar         lib
hadoop-hdfs-client-3.0.0-tests.jar  sources
hadoop-hdfs-httpfs-3.0.0.jar        weather.txt
hadoop-hdfs-native-client-3.0.0.jar webapps
hadoop-hdfs-native-client-3.0.0-tests.jar
[root@master hdfs]#
```

```
[root@master hdfs]# hadoop jar ../mapreduce/WeatherWork.jar demo.mr.wordcount.WordCountMapReduce /newinput /newoutput2
```

```
n uber mode : false
2018-09-06 16:53:53,858 INFO mapreduce.Job: map 0% reduce 0%
2018-09-06 16:54:20,976 INFO mapreduce.Job: map 100% reduce 0%
2018-09-06 16:54:40,179 INFO mapreduce.Job: map 100% reduce 100%
2018-09-06 16:54:43,366 INFO mapreduce.Job: Job job_1536221753267_0001 completed successfully
2018-09-06 16:54:44,165 INFO mapreduce.Job: Counters: 53
File System Counters
```

## 3. 查看 hadoop 运行结果

[illegible]