**大数据数据初步分析**

# 数据抓取：

**package** com.example.weather;

**import** java.io.BufferedWriter;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**import** java.util.Calendar;

**import** java.util.Date;

**import** java.util.List;

**public** **class** WritetoFile {

**public** **static** **void** writeMethod() {

String fileName = "E:/wheather/xiameng.txt";

List<wheather> list=**new** ArrayList<wheather>();

String start = "2017-01-01";

String end = "2017-12-31";

SimpleDateFormat sdf = **new** SimpleDateFormat("yyyy-MM-dd");

Date dBegin=**null**;

Date dEnd=**null**;

**try** {

dBegin = sdf.parse(start);

dEnd = sdf.parse(end);

} **catch** (ParseException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**try** {

BufferedWriter out = **new** BufferedWriter(**new** FileWriter(fileName));

list=*getWheather*(dBegin,dEnd);

**for** (wheather w : list) {

out.write("weather\_date：" + w.getWeather\_date()

+ "day\_weather：" + w.getDay\_weather()

+ "night\_weather：" + w.getNight\_weather() + "day\_temp："

+ w.getDay\_temp() + "night\_temp：" + w.getNight\_temp()

+ "day\_wind：" + w.getDay\_wind() + "day\_wind\_comp："

+ w.getDay\_wind\_comp() + "night\_wind："

+ w.getNight\_wind() + "night\_wind\_comp："

+ w.getNight\_wind\_comp() + "\n");

}

//out.newLine(); // 注意\n不一定在各种计算机上都能产生换行的效果

out.close();

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

**public** **static** **void** main(String[] args) {

*writeMethod*();

}

**public** **static** List<wheather> getWheather(Date dBegin, Date dEnd) {

List<wheather> list=**new** ArrayList<wheather>();

Calendar calBegin = Calendar.*getInstance*();

// 使用给定的 Date 设置此 Calendar 的时间

calBegin.setTime(dBegin);

Calendar calEnd = Calendar.*getInstance*();

// 使用给定的 Date 设置此 Calendar 的时间

calEnd.setTime(dEnd);

// 测试此日期是否在指定日期之后

**while** (dEnd.after(calBegin.getTime())) {

wheather w = **new** wheather();

// 根据日历的规则，为给定的日历字段添加或减去指定的时间量

calBegin.add(Calendar.***DAY\_OF\_MONTH***, 1);

SimpleDateFormat formatter = **new** SimpleDateFormat("yyyy-MM-dd");

String dateString = formatter.format(calBegin.getTime());

w = WeatherReportByCity.*GetTodayTemperatureByCity*("146",dateString);

System.***out***.println(calBegin.getTime().toString());

list.add(w);

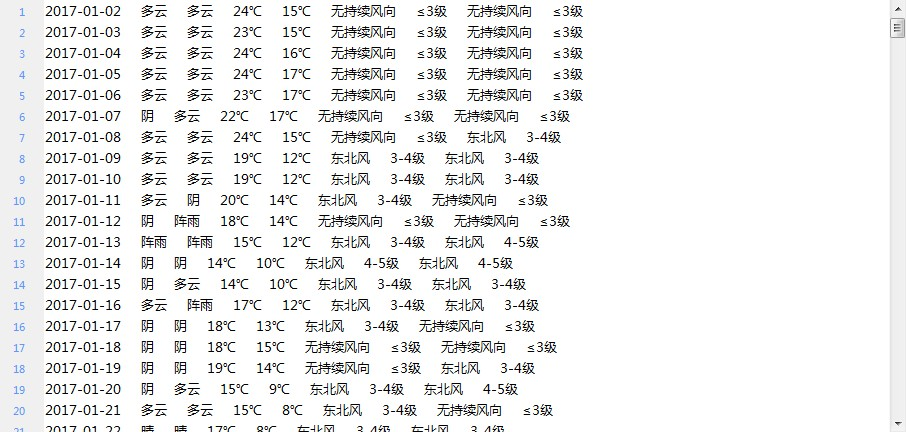
}

**return** list;

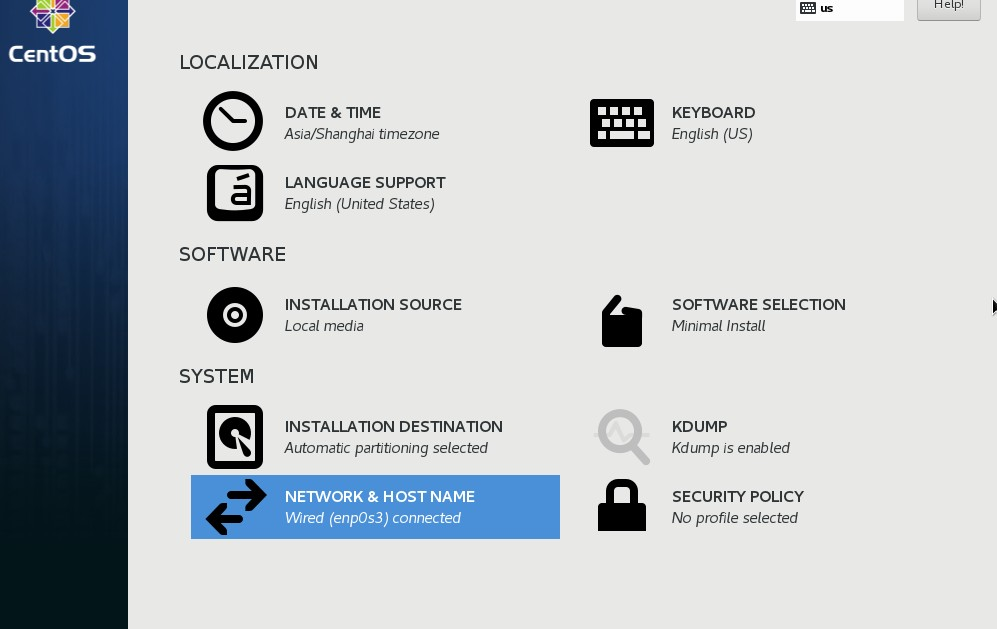
}

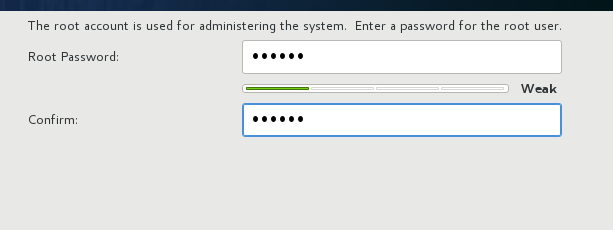
}

## 获取的数据：



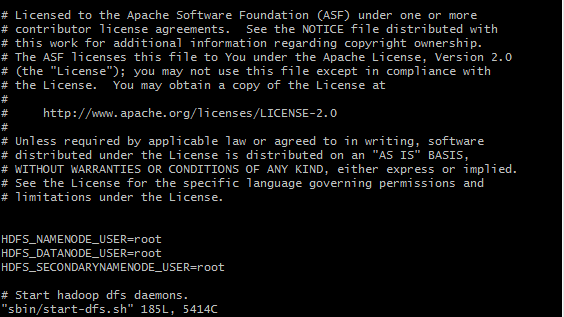
# 环境搭建



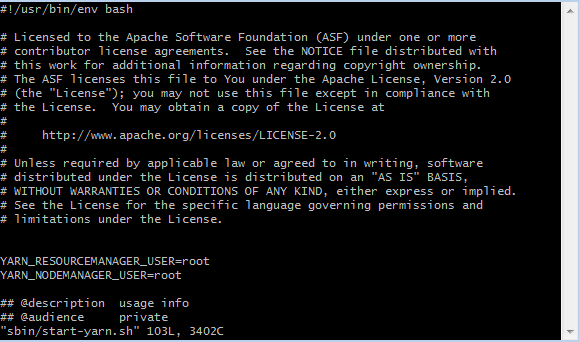


编辑以下文件



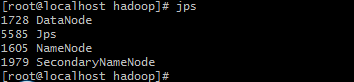








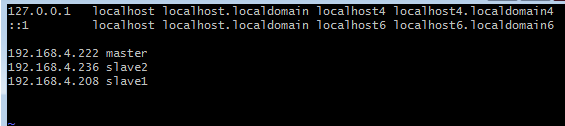




3台虚拟机相连：

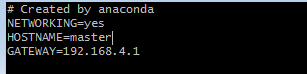
编辑etc目录下的hosts文件，除了本机加上其他两个虚拟机的地址与命名





编辑etc目录下的网络配置





将本机的公钥导入到其余两台虚拟机中,分别为master和slave1，



连接到另外两台虚拟机





上述步骤需同时在其余两台虚拟机上进行相同的配置

# 数据分析

## 数据分析代码：

**JAVA代码：**

Map代码：

**public** **class** WordCountMapper **extends** Mapper<LongWritable, Text , Text, Text> {

@Override

**protected** **void** map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, Text>.Context context)

**throws** IOException, InterruptedException {

String line = value.toString();

String[] words = line.split(" ");

String id = words[0];

String cityname = words[1];

String datetime = StringUtils.*substringBeforeLast*(words[2],"-");

String[] date = datetime.split("-");

String tmp=StringUtils.*substringBefore*(words[5], "℃");

String tmp1=StringUtils.*substringBefore*(words[6], "℃");

context.write(**new** Text(cityname+" "+date[0]+"年"+date[1]+"月"), **new** Text(tmp+"--"+tmp1));

}

}

Reduce代码：

**public** **class** WordCountReducer **extends** Reducer<Text, Text , Text, Text> {

@Override

**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** tavgtemperture=0;

**int** navgtemperture=0;

Text t=**null**;

**int** i=0;

**for**(Text value : values) {

s=value.toString();

String[] words = s.split("--");

tsum += Integer.*parseInt*(words[0]);

nsum += Integer.*parseInt*(words[1]);

i++;

}

tavgtemperture=tsum/i;

navgtemperture=nsum/i;

t =**new** Text("白天平均温度是"+tavgtemperture+"℃ 夜间平均温度是"+navgtemperture+"℃");

context.write(key,**new** Text(t));

}

}

MapReduce代码：

**public** **class** WordCountMapReduce

{

**public** **static** **void** main( String[] args ) **throws** Exception

{

Configuration cfg = **new** Configuration();

Job job = Job.*getInstance*(cfg, "worcount");

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) {

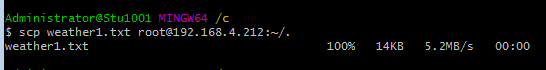
System.***out***.println("wordcount task fail!");

}

}

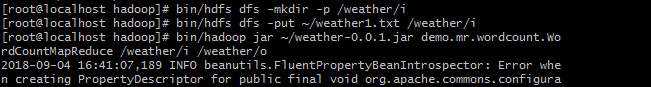
}

将以下两个文件传到虚拟机中



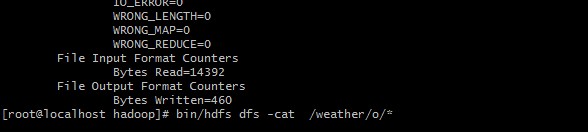


虚拟机下运行



## 分析的数据：

运行以下命令

查看分析后的数据：

