

文章编号: 2096-1618(2016)02-0180-05

基于动态 Web 的 Python 多线程空气质量数据程序设计

陈琳, 李容
(四川省气象台, 四川 成都 610072)

摘要:环境气象预报是政府和环保部门应对空气重污染时的重要决策支撑,而气象部门环境气象业务要求必须实时快捷地获取当前空气质量监测数据。为快速、准确地获取到四川省空气质量数据,并生成 Micaps 格式产品,提出基于动态 Web 开发 Python 多线程程序的思路。程序具有较高性能,直接模拟浏览器定时调用网站底层数据库,获得数据效率更高,并且该程序具有较高地实时性和准确性。该程序已成功用于四川省气象台日常环境气象预报业务中。

关键词:计算机应用;气象信息技术;Python;多线程;空气质量数据;动态 Web;Micaps

中图分类号:TP311

文献标志码:A

0 引言

随着工业化、城市化进程的加快,城市空气污染愈显突出,而空气污染的形成与改善和气象条件密切相关,作为政府部门决策支撑的环境气象预报应运而生。气象部门环境气象预报业务需要高时效的空气质量数据、气象观测数据和数值预报产品作为预报数据支撑。空气质量数据包含定量描述空气质量状况的无量纲指数-空气质量指数(AQI),以及细颗粒物($PM_{2.5}$)、可吸入颗粒物(PM_{10})、二氧化硫(SO_2)、二氧化氮(NO_2)、臭氧(O_3)、一氧化碳(CO)等6种具体污染物指标,准确及时地获取空气质量数据是环境气象预报的基础^[1-4]。目前各省的环境监测总站将空气质量数据按时以数据交换方式提供给各级气象台。但是由于各方面的原因,环境监测总站提供给气象台的数据无具体值,且其数据实时性远不能达到环境气象预报业务的要求,此外,互联网上虽有当日空气质量监测实况,但是网上的各类数据无法加载到气象部门的专业业务系统(如Micaps系统等)中,对环境预报业务开展带来了诸多不便。文中主要阐述利用Python提供的处理动态Web标准库,生成Micaps格式的环境监测数据产品的具体实现方法。

1 相关概念

1.1 动态 Web

所谓的动态 Web,是指与静态网页相对的一种网

页编程技术。静态网页,随着html代码的生成,除非修改页面代码,页面的内容和显示效果就基本上不发生变化了。而动态 Web 则不然,动态网页是基本的html语法规则与Java、C#等高级程序设计语言、数据库编程等多种技术的融合,以期实现对网站内容和风格的高效、动态和交互式的管理。值得注意的是:动态页面的“动态”是网站与客户端用户互动的意思,根据服务器端数据库操作的结果发生改变^[5-7]。截至2015年底,动态 Web 占互联网网页总数的80%,文中需要获取数据的网页——四川环境空气质量发布系统就是一个动态 Web。

1.2 Python 脚本语言

Python 是一种简单易学,功能强大的开源编程语言,几乎能够在目前所有的操作系统上运行,它有高效率的多层数据结构,能简单而有效地实现面向对象编程^[8]。Python 简洁的语法和对动态输入的支持,再加上解释性语言的本质,使它在大多数平台上的很多领域都是一个理想的脚本语言,特别适用于快速地应用程序开发^[9]。特别是 Python 提供针对网络协议标准库,对网络协议的各个层次进行抽象封装,程序员就可以集中精力处理程序逻辑。其次,Python 非常擅长处理字节流的各种模式,具有很快的开发速度^[10-11]。

1.3 Python 多线程

Python 使用线程主要有两种方式:用函数或使用类来包装线程对象。使用函数实现多线程:调用线程 thread 模块的 thread.start_new_thread() 函数创建一个新的线程。程序使用函数方式调用线程。语法如下:

收稿日期:2016-04-06

基金项目:中国气象局2015年小型业务建设资助项目(气减函[2015]22号)

thread.start_new_thread (function, args [, kwargs])
其中 function 参数是你将要调用的线程函数;args 是讲传递给你的线程函数的参数,kwargs 是可选参数^[12]。关键代码如下:gethtml(url) 是获取全国城市的空气质量指数(AQI)排行榜的函数。

```
try:  
    thread.start_new_thread(gethtml(url), ("Thread-1", 2,))  
    thread.start_new_thread(gethtml(url), ("Thread-2", 4,))  
except: Exception,e:  
    returnException," :",e
```

2 基于动态 Web 的 Python 环境监测数据产品程序设计

程序设计主要包括 3 个模块,首先分析四川环境空气质量发布系统的网站源码,找到获得数据的 Request URL,从 Request URL 获得数据、匹配站点生成 Micaps 格式产品和生成可执行文件。从 Request URL 获得数据是基础,只有下载 Request URL 提供的数据,才能匹配预先设置的四川 21 个地市州站点信息,得到各站点环境监测数据。程序设计流程如图 1 所示。

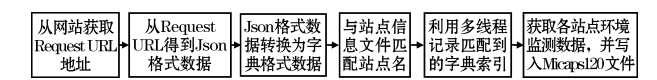


图 1 程序设计流程

2.1 从 Request URL 获得数据

统一资源定位符(URL)是对可以从互联网上得到的资源的位置和访问方法的一种简洁的表示,是互联网上标准资源的地址。互联网上的每个文件都有一个唯一的 URL,包含的信息指出文件的位置以及浏览器应该怎么处理它^[13]。

因四川环境空气质量发布系统既是一个动态网页,又没有发布 OpenAPI,只有先分析整个网页。打开 Chrome 浏览器,按下键盘 F12 即可弹出“开发者工具”。如图 2 所示,此时选中“Network”标签,在地址栏中输入此页面“http://newair.scemc.cn:5100/publish/index.html”,浏览器会分析出此次响应的全过程,而方框内的文件,就是此次响应中,浏览器和 Web 后端的所有通信。而标记的函数,就是此次响应中,获取每小时四川省环境监测数据的函数。在图 1 的“Header”一栏中明确地显示出了此次响应的过程:对目标 URL,用

POST 的方式,而 Request URL:http://newair.scemc.cn:5100/publish/getAllCity24HRealTimeAQIC,也就是获取四川省环境监测数据产品的 URL。

如果直接利用 Request URL 读取数据库,某些网站对直接读取行为一律拒绝,这时就需要设置 Request Headers 中的 user-Agent,把获取数据库的行为伪装成浏览器行为。user-Agent 中提供给了浏览器将要发送给服务器端的 http 请求头中 user-agent 的信息,如果无 User-Agent,网站服务器会识别你为非法浏览而屏蔽你的 IP。设 user_agent = 'Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/44.0.2403.155 Safari/537.36',这样就把获取网站数据的行为伪装成 Windows 浏览器访问网站。

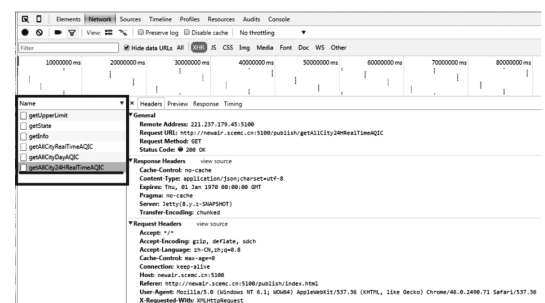


图 2 网站“开发者工具”界面

在 Python 语言众多网络函数库中,功能最强大的是 Urllib、Urllib2 库。Urllib、Urllib2 库提供一系列用于操作 URL 的功能,可以像读取本地文件一样读取 www 和 ftp 上的数据。Urllib2 库的 request 模块可以非常方便地抓取 URL 内容,也就是发送一个 GET 请求到指定的页面,然后最后返回 JSON 数据^[14]。具体流程如下:定义一个函数 gethtml(url):获取四川省 21 个城市的空气质量实时数据,返回的是 JSON 格式的一个数组,其中每项的信息有:包括评价时段、监测点位名称及位置、各监测项目的浓度、空气质量数据质量标示符、空气质量分指数、空气质量指数、首要污染物及空气质量级别等。

字典是 Python 中唯一内建的映射类型,且可存储任意类型对象,每个键值(key=>value)对用冒号(:)分割,每个对之间用逗号(,)分割,整个字典包括在花括号{}中,键必须是唯一的,但值则不必。值可以取任何数据类型,但键必须是不可变的,如字符串,数字或元组。因 Request URL 获取到本地的数据是 JSON 格式的,还须把 JSON 格式转换成字典格式:airproduct = json.loads(htmls.read())。转换成字典后数据如下所示:[{"data":[{"columns":{"AQI": "39", "CITY-

CODE": 5101," CITYNAME ":" 成都 "," CO ":" 1.259 "," CO_MARK ": true, ... }, { " columns": { " AQI": "45", " CITYCODE": 5103, " CITYNAME": " 自贡 ", " CO ":" 0.906 ", " CO_MARK ": true, ... } }]。此时,airproduct 是一个多维字典。上述流程关键代码如下:

```
def gethtml(url):
    headers = { 'User-Agent': user_agent } #标识键
    User-Agent'
    datax = urllib.urlencode(data) #表单数据
    req = urllib2.Request(url, datax, headers)
    response = urllib2.urlopen(req, timeout=4) #模拟浏览器提交数据。
    airproduct = json.loads(response.read())
    return airproduct
```

因从四川环境空气质量发布系统获取到本地的字典数据包含空气质量数据标示符等无用键如“CO_MARK”、“PM10_MARK”、“STATE”等,需要把这些键剔除。定义函数 Write_keys(airproduct),dic 提取某个站点空气质量数据的键列表;dic1 为自定义无用键的列表;dic2 是列表 dic 与 dic1 的相对补集;use_dic 是在列表['CODE','LON','LAT'] 右加列表 dic2,保证“CODE”(站点编号)、“LON”(站点经度)、“LAT”(站点纬度)键居列表 dic2 首位,use_dic 是有效键的列表,也是最后生成 micaps120 类文件站点要素的列表。

```
def Write_keys(airproduct):
    dic = airproduct[ 'data' ][ 0 ][ 'columns' ]. keys()
    dic1 = [ 'CODE', 'CICODE', " CO_MARK ", ..., " TIMEPOINT", 'STATE' ]
    dic2 = set(dic) - set(dic1)
    use_dic = [ [ 'CODE', 'LON', 'LAT' ] ]. __iadd__ (list(dic2))
    return use_dic
```

2.2 匹配站点生成 Micaps 格式产品

为加载在气象专业软件 Micaps 系统上,就需要把数据转换成 Micaps 120 类格式。具体步骤如下:在多维字典 airproduct 中,站点编号中以“CODE”键表示,21 个城市空气质量实时数据按照“CODE”键有序排列。首先生成 21 个地市级城市的经纬度信息的文件 geo.txt,如表 1 所示。获取的空气质量字典数据 airproduct 没有经纬度的键值,需对照站点信息表 geo.txt 的经纬度值补齐,把经纬度信息以字符串形式读入内

存中,以列表 linesid[]表示,站点的纬度和经度分别以 linesid[i].split('\t')[1]和 linesid[i].split('\t')[2]表示。关键代码如下:

```
for i in range(0, len(linesid)):
    airproduct[ 'data' ][ i ][ 'columns' ][ 'LON' ] = linesid[i].split( '\t' )[ 1 ]
    airproduct[ 'data' ][ i ][ 'columns' ][ 'LAT' ] = linesid[i].split( '\t' )[ 2 ]
```

表 1 西南区域 52 个城市经纬度信息

编号	纬度	经度	站点
5101	30.67	104.02	成都
5103	29.35	104.77	自贡
5104	26.58	101.72	攀枝花
5105	28.88	105.43	泸州
...
5134	27.90	102.27	西昌

四川环境监测总站环境监测数据每小时更新一次,程序需每小时生成一个 Micaps 120 类数据文件,例如 16033018.000。引入 Python 中的 datetime 库,截取时间的字符串作为文件名。保存文件的地址 path = " d: \ \" + datetime.now().strftime('%Y% m% d% H') [2 : 10] + ".000"。最后把监测数据各列表分别写入到文件中,生成 Micaps 120 格式产品,生成文件见图 3,2016 年 3 月 30 日 18 时的空气质量监测产品在 Micaps 3.2 系统显示界面见图 4。在生成的 Micaps 120 类文件中,空气质量监测要素依次为 AQI(空气质量指数(AQI))、quality(空气质量 AQI 等级)、CO(一氧化碳 1 小时平均)、NO₂(二氧化氮 1 小时平均)、O₃(臭氧 1 小时平均)、O3_8 h(臭氧 8 小时滑动平均)、PM₁₀(颗粒物(粒径小于等于 10 μm)、PM_{2.5}(颗粒物(粒径小于等于 2.5 μm) 1 小时平均)、SO₂(二氧化硫 1 小时平均)。关键代码如下:其中 site 是多维字典 airproduct 中某站点空气质量数据的子字典;use_dic 是生成文件某站点键的列表;ele_value 是 use_dic 对应键的值;Ls 为某站点所有 ele_value 的列表。

```
for i in airproduct[ 'data' ]:
    site = i[ 'columns' ]
    Ls = [ ]
    for j in use_dic:
        ele_value = site[j]
        if type(ele_value) == int or type(ele_value) == float:
            ele_value = str(ele_value)
        Ls.append(ele_value.encode( 'gbk' ))
```


16033018.000 - 记事本														
文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)										
di.amond 120 2016年03月30日18时空气质量监测														
5101	30.67	104.02	28	1	0.8	14	47	47	47	42	5			
5103	29.35	104.77	47	1	0.3	10	65	65	58	25	5			
5104	26.58	101.72	27	1	1	10	57	54	54	34	24			
5105	28.88	105.43	26	2	0.2	8	46	46	46	32	6			
5106	31.15	104.38	36	2	0.6	5	61	56	56	41	7			
5107	31.47	104.68	22	1	0.6	8	48	48	48	38	8			
5108	32.43	105.85	9	1	0.9	10	41	41	41	36	16			
5109	30.50	105.58	24	1	0.7	6	43	43	43	34	6			
5110	29.58	105.07	17	1	0.5	9	29	37	29	37	10			
5111	29.57	103.75	19	1	0.7	14	25	33	25	33	7			
5113	30.78	106.10	43	1	0.8	10	45	60	45	23	5			
5114	30.05	103.82	14	2	0.4	7	32	45	32	45	6			
5115	28.90	104.60	35	1	0.8	11	57	54	54	31	16			
5116	30.47	106.63	48	1	0.7	10	67	67	59	38	7			
5117	31.20	107.50	30	2	0.6	11	48	48	48	18	14			
5118	29.98	103.00	35	2	1	8	47	50	47	36	5			
5119	31.85	106.77	11	1	0.5	9	24	24	24	24	3			
5120	30.02	104.65	22	2	10.4	5	49	49	49	42	17			
5132	31.90	102.23	10	1	0.4	5	34	34	34	34	4			
5133	30.05	101.97	12	2	0.5	6	56	53	53	18	12			
5134	27.90	102.27	21	1	1.7	14	42	45	42	45	23			

图 3 Micaps 120 格式的环境监测数据产品

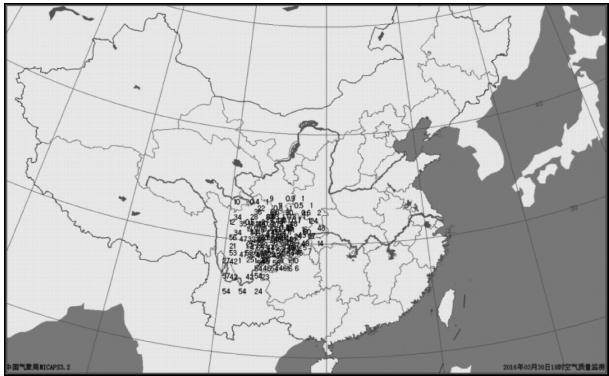


图 4 2016 年 3 月 30 日 18 时的空气质量监测产品在 Micaps 3.2 系统显示界面

2.3 生成可执行文件

为使 Python 脚本程序能在不用安装 Python 环境的 Windows 系统下运行,必须利用 py2exe 等工具把 Python 脚本(以 .py 为后缀)转换成 Windows 上的可独立执行的可执行程序(*.exe)。把文中的 Python 脚本命名为 airquality.py。首先在网上下载 py2exe 的安装程序 installer,安装后,写一个用于发布程序的设置脚本^[15],例如 setup.py。setup.py 代码如下:

```
from distutils.core import setup
import py2exe
setup(console=["airquality.py"])
```

然后在 python 的命令提示窗口,键入如下命令:python setup.py py2exe。命令执行后将产生一个名为 dist 的子目录,其中包含可执行程序 airquality.exe。把 airquality.exe 放在 Windows 环境下,在 Windows 的计划任务里,设置程序每小时执行一次,即可获取每小时更新一次的环境监测数据。

3 结束语

利用 Python 丰富的标准库及快速开发的特长,设计一个基于动态 Web 的 Python 多线程空气质量监测数据产品程序,实现每小时定时自动化生成 Micaps 格

式的四川省 21 个地市州空气质量监测产品。目前该程序已部署到四川省气象台日常天气预报业务中,其生成的空气质量监测产品为环境气象预报的研究工作提供了实况对比和技术支撑,满足了预报员对空气质量监测实况资料采集的需求。

参考文献:

[1] 刘新超,郭洁,申辉,等.成都市大气污染对儿童呼吸系统疾病的影响[J].高原山地气象研究,2015,(3):84-87.

[2] 冯良敏,陈朝平,龙柯吉,等.成都地区 2012 年 PM10 污染过程气象条件分析[J].山地气象研究,2014,(2):57-62.

[3] 张建忠,孙瑾,王冠岚,等.北京地区空气质量指数时空分布特征及其与气象条件的关系[J].气象与环境科学,2014,37(1):33-39.

[4] 秦福生,周岩,王淑琴,等.郑州市主要污染物污染特征及污染趋势分析[J].气象与环境科学,2007,30(4):63-65.

[5] 孙卫红.用动态 HTML 制作动态 Web 页面[J].计算机应用,1998,(9):27-29.

[6] 安海忠.动态 Web 应用高级开发[M].北京:人民交通出版社,2000:57-59.

[7] 魏兵海.基于 XML 相关规范集的动态 Web Service 框架系统[J].计算机科学,2004(6):68-71.

[8] 刘志凯,张太红,刘磊.基于 Web 的 Python3 编程环境[J].计算机系统应用,2015,24(7):236-239.

[9] 王大伟.基于 Python 的 Web API 自动化测试方法研究[J].电子科学技术,2015,2(5):573-581.

[10] Magnus Lie Hetland,司维,曾军葳,等. Python 基础教程[M].北京:人民邮电出版社,2014:243-245.

[11] 高森. Python 网络编程基础[M].北京:电子工业出版社,2007:326-327.

[12] 李俊丽.基于 Linux 的 python 多线程爬虫程序设计[J].计算机与数字工程,2015,(5):861-863.

[13] Elizabeth Castro. HTML XHTML CSS 基础教程.北京:人民邮电出版社,2007:35-37.

[14] TJ O'Connor,崔孝晨,武晓音. Python 绝技:运用 python 成为顶级黑客[M].北京:电子工业

出版社,2016:213–220.

[M]. 北京:电子工业出版社,2016:191–194.

[15] 刘凌霞,郝宁波,吴海涛. 21 天学通 Python

Python Multithreaded Air Pollution Products Program based on Dynamic Web

CHEN Lin, LI Rong

(Sichuan Provincial Meteorology Observatory, Chengdu 610071, China)

Abstract: Air pollution products as the government and the environmental protection departments to deal with heavy pollution weather decision support, attaches great importance to by the meteorological department. In order to quickly and accurately get to Southwest regional Air pollution products and generate the Micaps products, this paper developed a OpenAPI Python multithreaded program. The program is of high performance, it is directly simulates the browser to call at the bottom of the database, the fetching data is more efficient, and the program is with timeliness and accuracy. This program has been applied successfully in daily operations of the Sichuan Meteorological Observatory.

Key words: computer application; meteorological information technology; Python; multithreading; air pollution products; dynamic Web; Micaps