基于 Python 的网络数据爬虫程序设计

张艳[1]，吴玉全[2]

（1. 江苏省宿迁高等师范学校学前三系，江苏 宿迁 223800；
2. 中国电信股份有限公司宿迁分公司，江苏 宿迁 223800）

摘　要： Python 语言是一种跨平台、面向对象的解释型编程语言，它的语法简洁，应用广泛且容易操作。与其他语言相比，基于 Python 的爬虫有很多优势。主要介绍了基于 Python 的爬虫技术，给出了利用 Python 进行网站数据的爬取程序设计，阐释了 Python 爬虫技术的先进性和便捷性。

关键词： Python 语言；网络爬虫；程序设计

DOI:10.16184/j.cnki.comprg.2020.04.010

1　概述

Web 已经成为日新月异迅速发展的网络信息技术中的信息载体，如何有效地提取和利用这些信息已经成为亟待解决的问题。利用搜索引擎可以获得互联网最有用的、可以免费公开访问的数据集，查找用户所需的价值数据或者相近的价值信息。作为搜索引擎的核心组成模块，网络爬虫在信息检索过程中有着举足轻重的地位。通过网络爬虫技术可以迅速找到这些被嵌入在网站的结构和样式中的有用信息，并给用户筛选出有价值的数据。因此，网络爬虫技术的研究，在很大程度上节省了更多的人力和物力资源，而且在搜索引擎的发展中具有十分重要的意义。

2　Python 简介

Python 是一种广泛使用、功能强大面向对象的程序设计语言，能够在短时间内简单有效地实现面向对象编程，Python 语言飞速发展，其简洁、免费、易学、兼容性好等特点受到众人喜爱[1]。

使用 Python 编写网络爬虫有其独特的优势。

（1）语言简洁，使用方便。与其他经常使用英语关键字和一些标点符号的语言相比，用 Python 书写的代码更容易阅读和理解，语法比较简单，其设计更简洁、方便、高效，也更容易为大众用户所使用。Python 易于配置的脚本特性，还使得它在处理字符方面也非常灵活。此外，Python 通过强大的爬虫模块，对抓取网页本身的接口操作和网页抓取后的处理都得心应手。

（2）提供功能强大的爬虫框架，各种爬虫框架方便高效地下载网页，这使得 Web 爬虫更高效地对数据进行爬取。

（3）丰富的网络支持库及网页解析器，Python 拥有便捷的库，包括 Request、gevent、redis、jieba、lxml、Pillow、pyquery、NLTK、BeautifulSoup 等。无论是最简单的爬虫程序还是复杂的爬虫系统，都可以利用它们轻松完成。

3　网络爬虫

3.1　定义

网络爬虫，主要用于收集互联网上的各种资源，它是搜索引擎的重要组成部分，是一个可以自动提取互联网上特定页面内容的程序，一段自动抓取互联网信息的程序称为爬虫，爬虫指的是：向网站发起请求，获取资源后分析并提取有用数据的程序，从技术层面来说就是通过程序模拟浏览器请求站点的行为，把站点返回的 HTML 代码、JSON 数据、图片、视频等爬到本地，进而提取自己需要的数据，存放起来使用[2]。网络爬虫架构如图 1 所示。

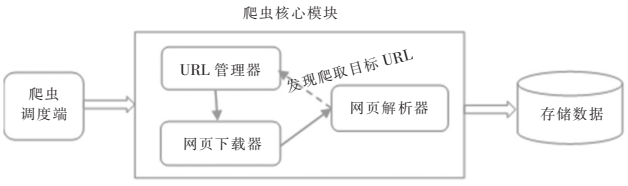

图 1　网络爬虫架构图

（1）爬虫调度端是程序的入口，主要负责爬虫程序的控制，这包括爬虫程序的启动、执行和停止，或者监视爬虫中的运行情况。

（2）爬虫核心模块包括 URL 管理器、网页下载器和网页解析器 3 个部分。1）等待爬取的 URL 数据和已

作者简介： 张艳（1981–），女，讲师，硕士，研究方向：计算机软件及理论。

经爬取好的 URL 数据是由 URL 管理器来管理，URL 管理器中的数据存储方式有 Python 内存、关系数据库和缓存数据库组成；2）等待爬取的 URL 数据通过网页下载器下载其对应的网页并存储为一个字符串，网页解析器再对传送过来的字符串进行解析，由 request 和 urllib2 实现 URL 并获取网页内容；3）网页解析器，一方面通过正则表达式、html.parser、BeautifulSoup、lxml 等实现解析，解析出有价值的数据，另一方面由于每一个页面都有很多指向其他页面的网页，这些 URL 被解析出来之后，可以补充进 URL 管理器[3]。爬虫调度端、爬虫核心模块和存储数据这 3 部分就组成了一个可以将互联网中所有的网页抓取下来的网络爬虫架构。

3.2 工作流程

网络爬虫工作首先明确要爬取的网站和数据，选择合适的方法来抓取数据，再将解析下载下来的网页和价值数据持久化，保存到数据库中。网络爬虫的基本工作流程如图 2 所示。

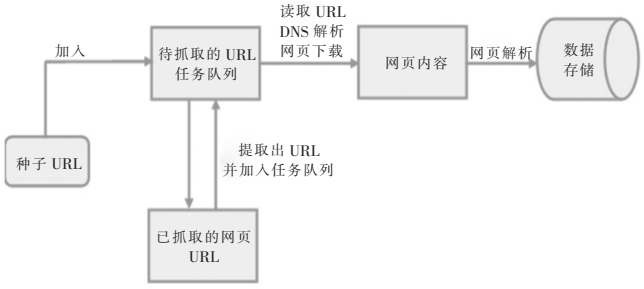

图 2 网络爬虫工作流程图

4 爬虫案例实施

通过 Python3.0 实现任意网页数据的爬虫，并将网页保存到本地来完成简单的网络爬虫程序。程序将编码方式设为可输出中文的 utf-8 形式，首先定义带参的页面爬取函数，该函数通过 requests 库的 get() 函数爬取所需页面内容，并将结果进行打印输出。

```
#coding:utf-8# 网络爬虫,尝试从网络上爬取整个网站
import urllib
import urllib.request
import re
    def get_html(url): # 根据给定的网址来获取网页信
# 息,得到的 html 就是网页的源代码
                page = urllib.request.urlopen(url) # 使用
#urllib.request.urlopen 打开页面
    html = page.read() # 使用 read 方法保存 html 代码
    return html.decode('utf-8')
    def ProcessLink(Urllist): # 读取 URL 并解析,保存
# 到本地
    print (len(Urllist))
    x=0
      for url in Urllist: # 将 Urllist 中保存的网页保存到
# 本地
          if url[0:4]! ='http':
          continue
       print (url)
       fileUrl = 'link\\' + str(x) + '.htm'
       print (fileUrl)
       with open(fileUrl, 'w') as f:
          html_code = get_html(url)
          print (len(html_code))
          if len(html_code)>0:
             f.write(html_code)
       x+=1
if _name_ == '_main_':
    url='http:// www.w3cschool.cn /'
    #reg_img=re.compile(r'src="(.+?\.jpg)" width')
    reg=re.compile(r'href="(.+?\.htm)"')
    #reg_img = re.compile(reg)# 若需要编译一下,可
# 运行得更快
    html_code=get_html(url) # 获取该网址网页详细信
# 息,得到的 html 就是网页的源代码
    linklist = reg.findall(html_code)# 与 html 进行匹配
    ProcessLink(linklist) # 从网页源代码中分析并下载
# 保存数据
```

5 结语

通过具有强大功能的网络爬虫技术，不但可以在短时间内提取出用户所需要的各种类型的信息数据，还可以挖掘出深层次更有价值的数据。拥有强大功能的 Python 语言可以为各种类型的软件工具包提供重要支持，在一定程度上，它还可以实现提取各种 Web 信息和数据，为面向主题的用户查询准备数据资源实现了网页数据抓取与分析。

参考文献

[1] 仇明. 基于 Python 的图片爬虫的程序设计 [J]. 工业技术与职业教育，2019，(3).
[2] 魏程程. 基于 Python 的数据信息爬虫技术 [J]. 电子世界，2018，(11).
[3] 吴永聪. 浅谈 Python 爬虫技术的网页数据抓取与分析 [J]. 计算机时代，2019，(8).