基于 Selenium 的 Python 网络爬虫的实现

花君林

（淮阴工学院计算机与软件工程学院，江苏 淮安 223003）

摘　要：随着大数据时代的到来，人们对数据的需求越来越大。尤其是商业数据，它的价值远远高出普通数据。而这些高价值数据往往被一些反爬机制保护着，为了解决这类问题，实现了一种基于 Selenium 的 Python 网络爬虫，它可以很好地解决此类问题，高效地爬取所需要的数据。

关键词：网络爬虫；Python 语言；Selenium 技术

DOI:10.16184/j.cnki.comprg.2017.15.010

1　爬虫流程

网络流程如图 1 所示。

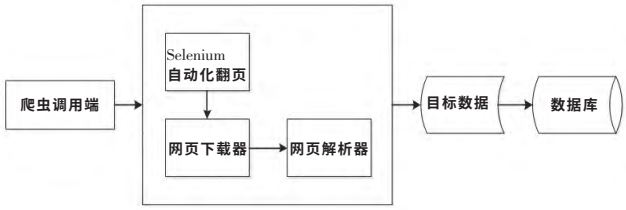

图 1　爬虫流程图

首先，需要认为的对爬虫主程序进行初始化信息，需要向爬虫提供一些参数，例如目标数据所在的初始网页地址，需要爬取的页面数、关键词等。然后，Selenium 开始工作，打开浏览器，输入网址，翻页。每次翻页过后，网页下载器便会获取当前页面的网页源代码，然后提交给网页解析器解析内容。通过一些合适方法将网页源代码中的目标数据提取出来，保存到数据库中，方便以后进行更进一步的处理。上述流程的时序图如图 2 所示。

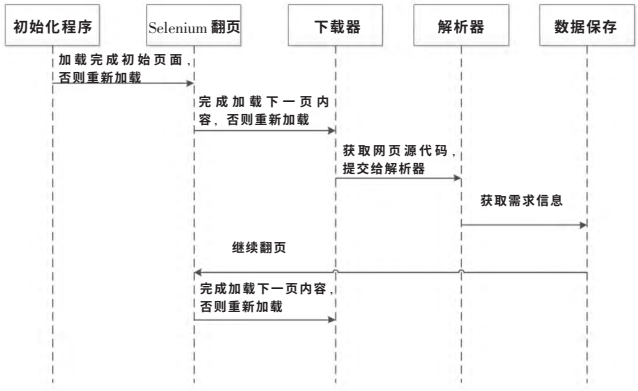

图 2　爬虫时序图

图 2 中最关键的点在于翻页之后，程序会在加载一段时间之后对加载的内容进行判断，是否已经加载完成，假如没有加载完成，那么，目标数据有可能会缺失甚至遗漏，在此，重新加载页面，直到加载完成后才进行下一步的处理。

2　爬虫核心技术

2.1　Selenium 和 WebDriver

Selenium 主要用于 Web 应用程序的自动化测试，但并不局限于此，它还支持所有基于 Web 的管理任务自动化，实现的爬虫便应用到了 Selenium。

Selenium 的特点如下：

（1）开源，免费。

（2）多浏览器支持：Firefox、Chrome、IE、Opera。

（3）多平台支持：Linux、Windows、MAC。

（4）多语言支持：Java、Python、Ruby、C#、JavaScript、C++。

（5）对 Web 页面有良好的支持。

（6）简单、灵活。

WebDriver 属于 Selenium 体系中设计出来操作浏览器的一套 API，它是按照 Server-Client 的经典设计模式设计的。下面是它的工作流程：

（1）WebDriver 启动目标浏览器，并绑定到指定端口。启动的浏览器实例将作为 WebDriver 的 Remote Server。

（2）Client 端通过 CommandExcuter 发送 HTTPRequest 给 Remote Server 的侦听端口。

（3）Remote Server 需要依赖原生的浏览器组件来转化浏览器的 native 调用。

作者简介：花君林（1993-），男，硕士，研究方向：化工大数据与云计算。

收稿日期：2017-04-27

Selenium 经历了两个版本，Selenium1.0 和 Selenium2.0。Selenium 不是由单独一个工具构成的，而是由一些插件、类库组成，每个部分都有其特点和应用场景。由于 WebDriver 是用于操作浏览器的一套工具，在 Selenium1.0 中是不包括它的。但是在 Selenium2.0 中将其加入了进来。所以，Selenium2.0=Selenium1.0+WebDriver。

使用便是最新版本的 Selenium2.0（以下均称为 Selenium）。在 Python 中，我们通过下面语句来导入 WebDriver 包：

```
From Selenium import webdriver
```

既然需要实现自动化操作，那么必须要让机器知道它们需要操作什么东西，之后就是如何操作。WebDriver 提供了相当丰富的方法来分辨页面上的不同元素。

在 Python 中，有 8 种定位方法，分别为：id，name，class，tag，link，partial link，XPath，CSS 定位方法。元素一旦定位好，就需要对其进行操作了，在这里介绍 3 种最常用的方法：

（1）clear()：用于清除文本输入框中的内容。

（2）send_keys()：模拟键盘向输入框里输入内容。

（3）click()：用来实现单机一个元素，也就是点击功能。（前提是元素可以被点击）。

如今大多数网页都是用 AJax 技术，网页的内容并不是同时被加载出来，而是按时间先后加载内容。这就需要程序设置元素等待，不然就会造成获取数据残缺甚至获取不到数据的情况。WebDriver 提供了两种类型的等待方法，使得抓取数据这一环节在网页加载完成后进行。

1）显示等待

等待某个条件成立时继续执行，超过最大时长时抛出异常（TimeoutException）。

2）隐式等待

等待一段时间后检测某元素加载完成，然后继续执行，否则抛出异常（NoSuchElementException）。

设置元素等待可以保证爬虫可以抓取到完整且详细的数据，不会造成遗漏甚至爬虫停止运行的情况。在使用 Selenium 打开浏览器的时候，选择 PhantomJS 浏览器。PhantomJS 是一个拥有 JavaScript API 的无界面 WebKit 内核，与 HtmlUint 类似。因此，在启用它的时候只有内核在运行，不会出现窗口或者界面，可以大大提高爬虫抓取目标数据的效率。PhantomJS 用来渲染解析 JS，Selenium 用来驱动以及与 Python 的对接，Python 进行后期的处理，他们可以无缝对接。

2.2 网页下载器

网页下载器是爬虫程序的主要核心模块，因为目标数据都在网页的源代码中，而网页源代码只有通过网页下载器才可以获得。通过 Selenium 模块中自带的 page_source 方法获得。具体代码如下：

```
from Selenium import webdriver
brower=webdriver.PhantomJS()
html=browser.page_source
```

2.3 网页解析器

通过要对它进行解析，这是在获取源代码之后需要进行的非常重要的一步。在此，列出了一些解析源代码的方法。

（1）正则表达式

（2）beautifulsoup4

（3）html.parser

（4）lxml

第一种方法属于字符串式的模糊匹配模式，其他 3 种为结构化解析模式，它们都以 DOM 树结构为标准，进行标签结构的提取。结构化解析是指通过源代码的结构，一层一层往下进行搜索，而不像模糊解析，通篇查找。

使用了 pyquery 网页解析器。pyquery 是 JQuery 的 Python 实现，只要对 jQuery 有一定的了解，那么使用 pyquery 就会如鱼得水，大大提高提取目标数据的效率。pyquery 有非常强大的 css 选择器功能。PyQuery 可以将传入的 html 数据源初始化为一个操作对象。下面是代码示例：

```
from pyquery import PyQuery as pq
doc=pq(html)
```

这就创建了一个 pyquery 操作对象，这个对象有许多方法可以调用，例如 find()，parent()，children()。只要得到的类型还是 pyquery 类型，那么就可以继续使用pyquery 来进行操作，直到获取目标数据。代码示例如下：

```
items=doc('#mainsrp-itemlist .items .item').items()
for item in items:
    Title=item.find('.title').text()
```

（下转第 36 页）

随着计算机软件的飞速发展，计算机软件对硬件的要求不断提高，硬件配置越来越显得配置偏低。目前2008年引进的两套集群，由于资料处理精度不断提高，数据量不断加大，进行并行运算的能力较差。因此，可以考虑利用每个节点自带的硬盘，将其构建为集群分布式存储，形成两套较大的存储设备。同时，还可以利用Jason等软件进行并行反演等运算，使老旧资源重新得以利用，充分利用硬件资源。

参考文献

[1] 杨勇. 基于GlusterFS的分布式冗余存储 [J]. 西安文理学院学报（自然科学版），2010，04：23-25.

[2] 胡文波，徐造林. 分布式存储方案的设计与研究 [J]. 计算机技术与发展，2010，04：41-43.

[3] 刘仲，章文嵩，王召福，周兴铭. 基于对象存储的集群存储系统设计 [J]. 计算机工程与科学，2005，02:35-37.

[4] 金弟，庄锡进，曹晓初，王启迪. 基于多路径地震资料处理集群存储系统 [J]. 计算机研究与发展，2012，S1：66-68.

(上接第31页)

3 数据的保存

使用MySQL数据库进行目标数据的保存。使用了Python中的MySQLdb包。保存数据的步骤如下：

（1）创建数据库连接对象。

（2）创建游标对象。

（3）执行SQL语句。

（4）提交事务。

（5）关闭数据库连接。

代码示例如下：

```
import MySQLdb
conn=MySQLdb.connect()
cur=conn.cursor()
sqlstr="insert into product values('1','2','4')"
cur.execute(sqlstr)
cur.close();
conn.commit()
conn.close()
```

4 具体案例

以爬取淘宝宝贝数据为例，表1为该爬虫的核心算法。

表1 核心算法

本算法的具体参数为page_number（需要爬取的页面数），url（初始页面），KeyWord（关键词）。	
1	输入参数。Page_number=100,url=“https://www.taobao.com/"，KeyWord=“美食”
2	初始已爬页面数为0
3	Begin
4	输入搜索内容“美食”，url“https://www.taobao.com/”。开始搜索
5	While已爬取页面数<=100
6	Repeat
7	抓取单个商品目标数据
8	保存到mysql数据库
9	Until不存在没有被抓取的商品
10	已爬页面数加1
11	If已爬页面数达到100那么
12	Break；
13	End

5 结语

实现的网络爬虫可以非常高效地抓取目标数据，并且可以避开网页对网络爬虫的限制，乃至封杀。从应用方面来说，获取的目标数据保存在数据库中，下一步就可以对数据库中的数据进行数据挖掘以及数据分析，挖掘出更具价值的东西。

参考文献

[1] 小甲鱼. 零基础入门学习Python. 北京：清华大学出版社，2016.

[2] 虫师. Selenium2自动化测试实战——基于Python语言. 北京：电子工业出版社，2016.

[3] 理查德 劳森. 用Python写网络爬虫. 北京：人民邮电出版社，2016.