华中农业大学学士学位论文

基于 Python 的主题爬虫设计与实现

Design and Implementation of Web Theme Crawler

based on Python

姓　　名： 郑泉石

学　　号： 2013310200408

专　　业： 计算机科学与技术

学 位 类 型： 工学学士学位

指 导 教 师： 王建勇　讲师

华中农业大学信息学院

中国 武汉

目录

摘要

在网络高速发展，软硬件技术日新月异的今天，各行各业的人们都普遍依赖互联网来获得资讯信息和生活上的便利。同时也有越来越多依赖于网络的电商平台出现，电商平台的优势也日益明显。针对目前电商平台越来越多，商品类型，数量随之越来越繁杂，使用户无法方便“货比三家”的情况，为了方便用户更有效率的了解各电商平台的商品信息，本文提出设计一个面向电商平台的主题爬虫，也是相应电商管理系统的重要组成部分。目前网络爬虫技术已越来越受到重视，但是针对电商平台的主题爬虫存在使用不方便且普遍被掌握在提供付费爬取服务的厂商手中的问题，因此针对电商数据采集的主题爬虫研究是很有意义的。

本文从当前电商平台现状为背景出发，通过分析相关爬虫知识，爬虫算法，电商平台现主要的反爬虫技术，相关应对反爬虫技术的措施与算法，爬取资讯处理等相关技术。设计了一个面向电商平台的网络爬虫，通过该爬虫系统可以对多种主流电商平台商品信息进行高效的采集与识别。

爬虫本身代码和应对反爬虫的技术是爬虫系统关键技术，因此本文主要对爬虫技术和“反反爬虫技术”进行分析和阐述，传统的电商主题爬虫大多通过调取电商平台 API，通过解析平台网页页面的 HTML 解析获得商品信息，但是如今的电商平台越来越注重于保护自己的数据，因此会刻意的在网页中加入动态加载，JS 动态生成信息或者加入认证环节或者发送经过压缩加密过的文本等等多种多样的反爬虫技术，并且反爬虫技术会时常更新变化，在很大程度上加大了爬取信息的代价和成本，因此传统的电商爬虫经常会出现爬取信息不全，爬取信息有误，或是根本无法爬取，或是陷入了爬虫陷阱，出现爬取效率低等等问题。而在经过长时间，持续，反复的对主流电商网站的研究与分析后。本文提出的算法有效解决了传统电商主题爬虫程序的使用不方便，爬取质量低的问题。通过对该电商主题爬虫的测试和与其他商用电商主题爬虫对比，发现本系统在爬取数据的正确率和支持平台数量上都有着明显的提高。

基于以上的研究，本文设计了基于 python 语言的面向于亚马逊，淘宝，速卖通，ebay 等国内外主流电商平台的主题网络爬虫系统，并在此基础上添加数据库认领模块，web 前端模块，形成一个完整，系统的电商主题爬虫系统。

关键词：电商平台；主题爬虫；反爬虫技术；python

Abstract

Nowadays , Internet technology is developing rapidly ,people with different work all rely on Internet to gain information and convenience in daily life .On the same time, various e-commerce sites with their own evident advantages have been showing up .To the situation that users are not able to evaluate their selected goods around various goods in various sites conveniently ,this article puts forward a design and implementation of web theme crawler oriented to e-commerce sites which is also a important part of e-commerce manage system .Web crawler technology has been attached importance to nowadays，existing crawler oriented to e-commerce sites are commonly inconvenient to use and in the hands of paid service providers .Therefore ,the research of web theme crawler oriented to e-commerce sites is of great meaning .

This article start from the circumstance now of e-commerce sites ,by analyzing related technology ,knowledge and algorithm, designs a system of web crawler oriented to e-commerce sites which is approaching a high efficient way to identify and collect information from main e-commerce sites.

The algorithm of web crawler and technology against anti-crawler technology are the core of this system .Traditional e-commerce oriented web crawlers usually take the sites' APIs to gain html files then collect information by analyzing html files .But e-commerce sites nowadays attach great importance to their datas ,it is of great possibility that they will take some measures to interfere the process .And this will cause a increasing price of collecting information ,it also cause troubles like information missing ,mistake information or unable to crawler ,etc .However after a long time and persistent research to main e-commerce sites, this article put forward a algorithm which resolves the inconvenience and inefficiency of traditional web crawler .By the comparing among other commercial e-commerce oriented web crawlers ,this crawler systems attains a obvious increasing on accuracy and the number of supported platform.

Basing on above research ,this article designs a web theme crawler based on Python oriented to main e-commerce sites like amazon ,taobao ,aliexpress ,ebay.etc .then adds SQL database ,web UI on this system ,to form a integral ,systematic e-commerce theme crawler system.

Keywords： e-commerce sites ; theme crawler ; anti-crawler technology ; Python

1 绪论

1.1 课题背景及其意

截至 2016 年 6 月，我国网络购物用户规模达到 4.48 亿，较 2015 年底增加 3448 万，增长率为 8.3%，我国网络购物市场依然保持快速、稳健增长趋势。其中，我国手机网络购物用户规模达到 4.01 亿，增长率为 18.0%，手机网络购物的使用比例由 54.8%提升至 61.0%。日渐成熟的中国电商行业带动了社会多种行业的迅速发展（中国互联网络信息中心，2016）。

在中国，主流的国内电商有淘宝，京东等，在国际市场上，主流的电商平台还有 amazon , wish ， ebay，walmart，aliexpress 等；作为电子商务领域的领军企业，amazon 从最初的图书售卖逐渐发展成为今天的综合性商城（Analysis I S，2014）。这些主流的电商平台上商品齐备，可以找到包括书籍，服饰，电子产品等各种各样的商品。目前，线上销售环节也成为各个销售厂商不可忽视的部分；从侧重于发展的角度来看，电子商务可以被定义为一种现代的商业模式，他使组织，商户，用户需求更明显，从而可以再提高产品服务质量，增加流通速度的同时减少商业成本（Nistor C et al，2010）。

电商平台的发展与兴盛为消费者提供了广泛的选择空间，但随着电商平台数量的增加，商品品类的增多，往往使得消费者眼花缭乱，无法方便的获得自己真正所需的商品，也无法方便的在同类货品中“货比三家”获得最大的优惠额度。同时商家给出的各种各样的活动和广告也容易干扰到消费者浏览商品的效率，使消费者被迫接受到很多不相关的信息，降低使用的效率。

而现存的很多作为电商从业者辅助软件存在的电商爬虫软件，普遍存在使用不方便，支持平台有限，抓取算法更新滞后，抓取商品信息不全，抓取信息错误的问题，而且很多都被掌握在付费服务提供商的手中。因此，本文提出的面向电商的网络爬虫旨在对国内外的主流电商网站的店铺分类下所有商品或单个商品的主要信息：标题，简介，原价，促销价，商品图片等信息进行抓取。并存取在数据库中方便用户查看，用最快的时间去获得最关键全面的信息，方便消费者用最少的钱获得最合适的商品。同时也方便了电商行业的从业者以更低的代价去了解所处行业实时的状态，为用户提供必要的帮助和参考。

1.2 研究内容

本文主要是对于以下内容的研究：

（1）本文提到的针对各个主流电商平台网站的网页结构设计的爬虫程序，主要目的是通过调用网站 API 或者模拟浏览器获得网页的的 json，xml 格式数据，或者网站 html 文件进行解析，获得商品的种子列表，进而使用多线程去分别抓取每一个商品的信息

（2）搭建电商爬虫系统，从整体的需求入手进行整体框架和功能模块的设计与实现，整体使用 SQL 数据库+python 后台+web 前端的模式。

（3）针对涉及到的主流电商平台的反爬虫技术的研究，包含但不限于 xhr 动态加载，js 动态生成，xsrf ，cookie 认证，gzip 压缩返回内容。

1.3 网络爬虫及其应用

网络爬虫是一种按照人为设定规则自动抓取网页的程序；按照一定的顺序提取依次 URL 地址，下载指向的页面，分析页面内容，获取新的 URL 地址放入带爬行 URL 队列中，重复进行直至带爬行 URL 队列为空或满足终止条件，这一过程称为网络爬行（孙立伟等，2010）。一个完整的网络爬虫主要由两个功能模块构成：网络内容读取模块，网络内容分析模块（袁浩，黄烟波，2009）。按照系统结构和实现技术，爬虫可分为通用爬虫和聚焦爬虫，增量式爬虫，深层网络爬虫。本文所主要介绍的是聚焦网络爬虫。

聚焦网络爬虫，也称主题网络爬虫。是以普通爬虫为基础的，实际上它是对一个普通爬虫进行功能上的扩充（汪涛, 樊孝忠，2004）。它只爬取符合预先定义好的主题的网页的爬虫，它只处理与主题有关的页面，极大节省了硬件和软件资源，更新速度快；相对于通用爬虫来说，主题网络爬虫注重的是对主题相关资源的爬取，发现需要的信息资源（王琨，2015）。

网络爬虫通常由控制器，解析器，资源库三部分组成，控制器负责给每一个需要爬取的链接启动线程调用爬虫，解析器负责下载链接，进行分析，对结果进行处理的任务，资源库用来储存网页中下载的数据记录。

爬虫在现实中，被用于许多方面。例如，搜索引擎技术会利用爬虫获得最新的数据，也可以利用爬虫检查链接活性，确认 html 代码，抓取垃圾邮件地址等。而在现实应用中，通常爬虫算法都会被要求达到以下的要求：快速，礼貌的爬取，重复内容剔除，连续爬行（Hawking D，2006）。

1.4 基于 python 的爬虫系统设计

Python 是一门解释性的、面向对象的、动态语义特征的高层语言。它的高层次的内建数据结构,以及动态类型和动态绑定,这一切使得它非常适合于快速应用开发（Mark, Lutz et al，2002）。它具有强大和丰富实用的第三方标准库，使编程变得简洁快速（肖旻, 陈行，2014）。在 python 中，模块，字符串，函数等都是对象，对派生，重载，继承，多继承都完全支持。因此程序代码的复用性很高，此外，python 也非常适合用来编写分布式系统。集中式的 Crawler 构架已经不能满足目前互联网的规模，因此支持分布式的爬行，处理和协调好各节点之间的交互，也是一个重要议题（周德懋, 李舟军，2009）。

此外，爬虫系统使用了“前端—后台—数据库”的设计模式，将功能模块分开，降低模块间的耦合性。

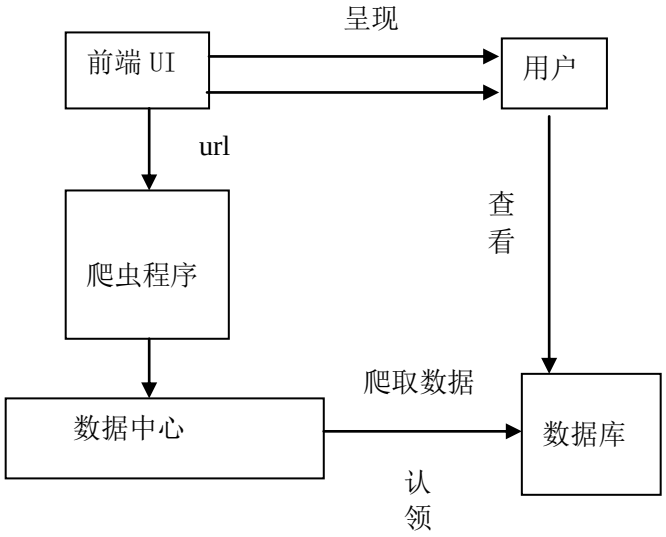

图 1　爬虫系统设计

Fig.1 Crawler system design

系统的模块间支持分布式，前端和爬虫后台间使用 python 的 web.py 开源框架，数据库使用 SQL，数据中心负责将用户认领的数据存入相应的表中，同时 UI 也可以查看数据库中数据。如果电商网站的反爬虫技术更新，可以方便的更新爬虫后台的代码，其他模块不受影响，方便系统后续的更新升级。

2　基于 web 的网络爬虫的工作原理及算法介绍

2.1　基础算法

2.1.1　html 解析

Web 浏览器所获得的网站内容一般都是 html 页面，通过 css 文件对其进行渲染；Web 页面是一种含有丰富链接结构的半结构化文档，其中链接结构是爬虫工作的基础（刘汉兴，刘财兴，2008）。而所需要的信息都包含在 html 页面内，html 文件内一般包含静态的标签体和动态执行的 js 代码，对于反爬意识不强的网站来说，所需的数据经常直接嵌在 html 的标签体内。向 Web 服务器提交对 HTML 页面的请求。提交后，服务器给出应答（王锋等，2010）。经过对标签体的解析，就能得到所需的内容，对于有反爬措施的网站一般也能从 html 得到所需的部分信息。

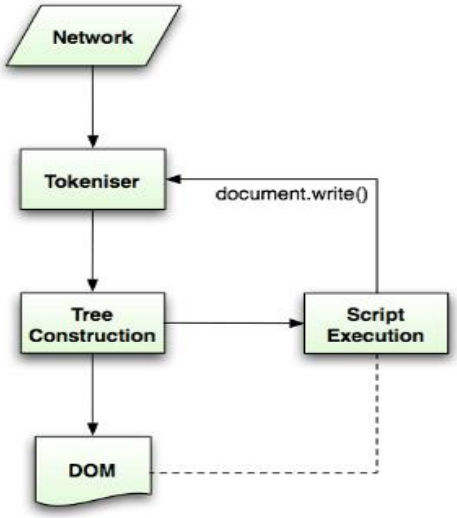

图 2 Html 解析过程

Fig.2 Html analyzing process

Html 不能被一般的自底向上或自顶向下的解析器解析，html 在解析时一般被标识解析为解析树，因此每一个元素都会有自己唯一对应的路径，在定位元素时，一般使用 css 选择器，xpath 和正则表达式也能达到相同的效果。

在实际使用时，本文使用了 python 的 pyQuery 库，通过将 html 文件转化为 pq 对象，指明元素路径后，便能得到所需要的内容。

2.1.2 广度优先搜索算法

广度优先算法是在搜索时，优先顾及当前搜索层次水平范围的搜索算法。该算法通常是为了尽可能多的搜集到种子链接，因此在店铺分类下的商品抓取通常使用此算法。

该算法的基本思想是通过一个 URL 链接入口，搜索完该页面上所有符合要求的链接，然后以每个链接为起点搜索下一个深度，直到所有的层次都被遍历或者达到预期深度后方才停止。该算法适合预期层次不深或网站规模较小的网站，否则容易出现数据量太大导致内存溢出的情况。

在图 3 中，依据广度优先搜索，正确的搜索顺序为 1，2，5，7，3，4，6，8。

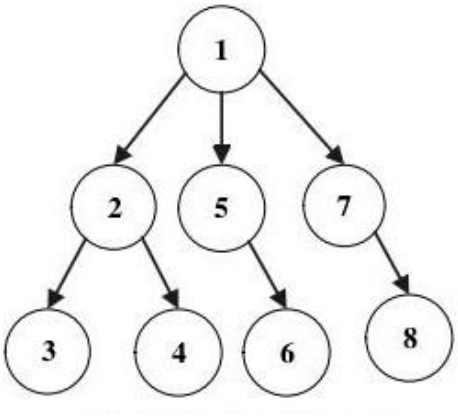

图 3　广度优先遍历

Fig.3 Breadth-first traversal

2.2　“反反爬虫”技术及其应用

2.2.1　反爬虫技术

随着国内外电商平台发展和技术日趋成熟，电商平台网站也越来越重视对站内商品信息的保护，因此绝大部分的电商平台都会为自己的网站写入或多或少的反爬虫保护。

针对淘宝，京东，亚马逊，ebay，wish，aliexpress，walmart 等国内外主流电商平台的研究，它们使用的反爬虫技术总结如下：

（1）动态加载：

动态网页是一个展现对请求回应中包含的特定信息的模板，他的动态体现在客户，服务两端的共鸣和交互上，动态网页在每次加载时都会变化（不需要任何人去特意制造改变）而且他们的变化取决于用户的操作，比如点击文字或图片（Soulemane M et al，2012）。而随着动态网站开发技术的发展，ASP,JSP,PHP 等都采用了页面参数传递的机制获得用户输入的参数信息（郑力明, 易平，2009）。

在这类网站里，商品价格等信息并不是静态嵌入在 html 文本中，而是随后通过 ajax 异步加载生成，有可能是 JS 动态生成，也有可能是调用别的接口获得 json 或 xml 格式信息通过解析后渲染到网页中，因此会出现浏览器中可见，元素路径可正确获得，但是在爬虫程序中模拟访问得到 html 网页进行解析后，抓取内容为空的情况。这是目前见得最多的反爬措施。使用此技术的网站代表：淘宝，京东，wish。

（2）Referer 字段认证：

在这类网站里，访问其站内任何网站，都会检查请求中的 header 里的 referer 字段，此字段一般为其网站的网址，否则该网站的服务器无论你访问任何页面都会干净利落的拒绝你的请求。使用此技术的网站代表：walmart。

（3）User-Agent 字段认证：

在这类网站中，网站会检查请求中 header 中的 User-Agent 字段，其要求访问网站的是真实存在的设备，如果是编程语言库中默认的 User-Agent 字段则会被拒绝访问。使用此技术的网站：几乎所有电商网站。

（4）Cookie，xsrf 证书认证：

在这类网站中，在访问网站或者进行特定动作，例如登陆之后网站服务器会返回包含 cookie 相关的信息，并要求浏览器进行 cookie 相关字段的设置，当服务器检测到 cookie 字段不对后，会要求登录或者拒绝访问。

Xsrf 是指跨站请求伪造，有些网站为了提高安全性会进行 xsrf 证书的验证，xsrf 的验证字段一般在 cookie 内，单独拿出来是为了列举 cookie 验证的一个例子。值得说明的是，cookie 验证可能验证 cookie 内多种内容，可能包括 xsrf 验证，但远不限于此。使用此技术的网站代表：aliexpress，wish。

（5）Gzip，br 压缩加密内容：

在这类网站中，有时会忽略用户要求返回内容格式字段，默认返回经 gzip 或其他方式压缩过的内容，此时用户无论进行解析或更换编码都会产生乱码，但是，这样做也会减少请求交换的内容大小，提升效率。使用此技术的网站代表：ebay，

（6）JavaScript 动态执行：

这个技术一般和前面部分技术一起使用，有时用来生成 cookie 中某些特定字段，如 xsrf_token，有时是将 API 返回的内容用 js 函数进行处理动态嵌入网页，使用灵活。在不知道该函数体内容情况下，基本不可能去执行针对性的“反反爬措施”。

（7）爬虫陷阱：

这个一般针对于没有设置爬取深度预期值的爬虫程序，采用该技术的网站会在网页中嵌入自我包含的或是没有结束的（例如日历中下一天）链接，导致爬虫陷入无止境的遍历中。但对于爬取深度并不算深的电商网站，使用的频率并不算高。

2.2.2 反反爬虫技术

“反反爬虫技术”顾名思义，是针对电商网站出于保护数据，节约服务器处理成本的考虑而使用的反爬虫技术的针对性措施。而在成文参考其他文献时并没有发现对此措施有统一明显的称呼，因此本文暂且称其为“反反爬虫技术”。

针对于以上的 2.2.1 节中所提到的反爬虫措施，经过自己的研究，总结了“反反爬虫技术”如下：

（1）动态加载：

对于这种情况，一般会抓取浏览器在访问该网站后所有的网络请求，尤其关注 XHR 以及 JS 请求，检查他们所有请求返回值中是否包含所需的信息内容，虽然访问一个页面会发现很多很多的 XHR 与 JS 请求，但是幸运的是，包含重要动态生成内容的请求一般在比较靠前的位置，因为在查找时实际工作量会比想象中要少。

当发现某个或几个请求中包含所需的信息后，随后会锁定这些请求，并把它们所请求的网址 API 记录下来，观察它们请求时是用的 post 还是 get 请求，然后再记录下他们发出请求时 header 的内容。记录下这些内容后，再用程序代码去模仿这一请求得到 html 或者 json 或者 xml 格式的数据后，在进行相应的解析，得到所需的信息。同时，在使用 get 方法时，参数被当成 http 请求中 url 的一部分，post 方法中，参数被放在 http 请求中（Madhavan et al，2008）。这些参数也能够帮助分析网站的活动。

（2）Rerferer 字段及 User-agent 字段认证：

应对这种技术，一般是人为去设置一下，一般 Referer 字段都是访问网站的主域名，观察设置一下就好了，而 User-Agent 字段则将浏览器的 User-Agent 字段复制下来作为自己程序的代理就可以了，理论上任何浏览器都可以，本文使用的是 Chrome 浏览器的代理。

（3）Cookie 及 xsrf 认证：

面对 Cookie 验证，是比较麻烦的，因为 Cookie 中的字段有可能是服务器随机返回的，也有可能是本地 JS 动态生成的，需要一层层的向上抓取并分析请求，必要时需要模拟执行 js 代码，同时 Cookie 中有些字段是必须的，有些字段却是可有可无的。而且 Cookie 的更新频率太快，若频繁操作可能几分钟内变化多次，即使不进行操作，一般 24 小时后也会失效，因此分析起来代价太大。

因此目前最好的方法是用浏览器访问，同时复制下来服务器返回的 Cookie，用到自己代码中，但是缺陷是不方便多次使用。

（4）Gzip 压缩：

设置专门的检测模块，在网络请求后检测 response 中头部压缩编码信息。常见的有 gzip，brotli，deflate 压缩，然后调用相应解压方式解压。

值得一提的是，在 Amazon 中的图片采用 base64 编码直接在网页中解码显示，而不是以 url 的形式展现。这样做会减少服务器访问负担，但是增加本地代码运行量。

（5）JavaScript 动态生成：

一般来说，JS 代码有时会被直接嵌在 html 网页中，或者 JS 代码的地址会被嵌入在 html 文件中，这时需要查找并模拟运行该 JS 代码。JS 代码的查找和参数获取都比较耗费精力，需要一定的耐心。一般来说 JS 函数体会十分的复杂，在抓不到

函数的情况，基本没有希望去模拟复现出函数。也有的 JS 代码会直接访问 API，此时只要抓取到 API 就能越过 JS 代码达到目的。

（6）爬虫陷阱：

在爬取电商网站时，所需的深度并不是很深，因此电商网站并不太经常使用爬虫陷阱，但是为了保持代码的健壮，可考虑设置爬取深度的预期值，这样在超过预期深度时，会控制爬虫程序终止爬取并立即返回所爬取到的数据。

3 基于电子商务平台的爬虫设计

3.1 淘宝平台爬虫设计

淘宝作为国内处于领导地位的电商平台，对数据的保护意识很强，会采取反爬措施而且更新频率比较高。

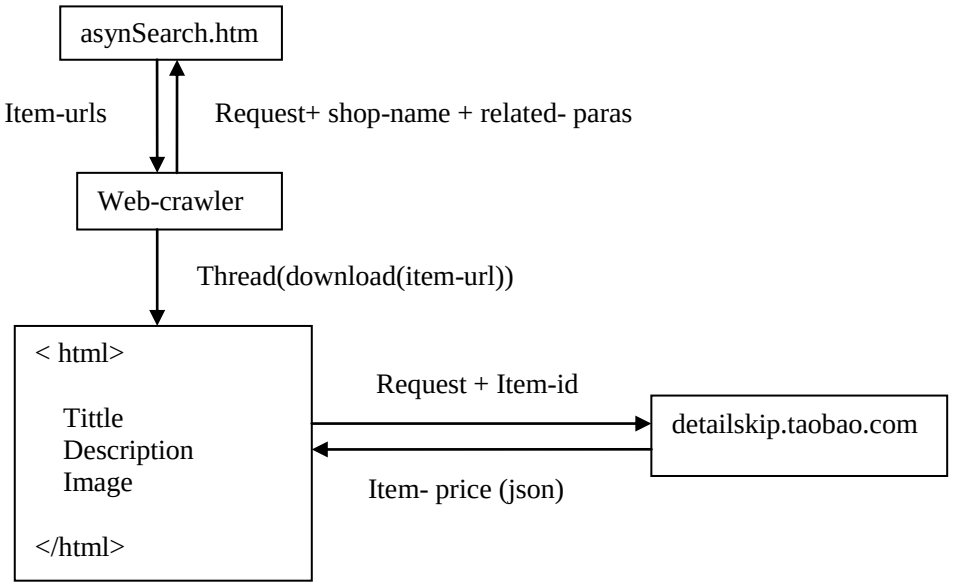

图 4 淘宝平台爬虫设计

Fig.4 Crawler design for taobao

在针对淘宝平台编写爬虫系统时，先从种子 url 中获得店铺名字等相关参数，然后调用淘宝网站 asynSearch 接口，获得店铺分类下所有商品的链接，然后爬虫程序会调用多线程对每一个商品分配一个线程进行抓取。单个线程会对每一个商品链接发出下载请求，最终获得单个商品的 html 页面，其中商品名称，简介，图片链接都是静态嵌入 html 中，可以经过 html 解析获得，但商品的原价促销价等价格信息是动态加载的，此时需从 html 文件中再抓取商品 id 等信息，向淘宝的 detailskip 接口发出请求。此接口会返回 json 数据格式的返回值，对此数据进行 json 解析，就能抓取到商品的价格信息。

经过对整套流程的分析，不难发现虽然本文是从网店分类目录下抓取商品信息，但是实际上本文整套抓取流程并不会访问到网店的页面，而在网店页面下，虽然浏览器可见商品列表及信息，但爬虫程序实际上并抓取不到任何信息。这里的商品列表，包括单个商品价格等重要信息都是动态生成的。

3.2 aliexpress 与 walmart 平台爬虫设计

在这里将这两个平台的爬虫设计写在一起是因为，这两个电商平台网站都会检查请求字段。若没有网站所要求的字段或者字段不正确，服务器会拒绝程序的所有请求。这两个网站没有类似淘宝，京东的动态加载的机制，因为经过服务器的字段检测后，服务器便认为请求是由真实存在的设备发出的。

其中，aliexpress 即速卖通，是阿里集团面向国际市场推出的电商网站。其服务器会检查所有请求中的 cookie，以鉴别用户身份，对于没有携带 cookie 或者 cookie 无法识别的请求，其服务器会统一返回登录页面。因此若要爬取速卖通数据，必须首先注册至少一个速卖通网站的账号。当用户登陆后，服务器会返回给用户一个区分用户身份的字段携带在 cookie 内，同时处理用户的请求。用户身份字段是经过秘钥，加密函数处理过的，并且一段时间后会更新字段。因此，若要破译速卖通的这个机制代价会比较大，另一个方法是用代码在登录界面携带账号密码模拟登陆，或者用浏览器直接登录后记录下 cookie，用在程序中。

Walmart 则是国际知名的零售巨头沃尔玛的线上商城，walmart 的商品种类多且全，在向其服务器发送请求时，服务器会检查请求头部 Referer 字段，检查其值是否为沃尔玛网站的主域名。面对 walmart 这种情况，只需在发送请求时，人为将请求头部设置一个 Rerferer 字段，值为沃尔玛商城的主域名即可。

值得注意的是，以上两个网站都会拒绝程序库中默认的 User-Agent，需要人为设置为浏览器的代理字段。

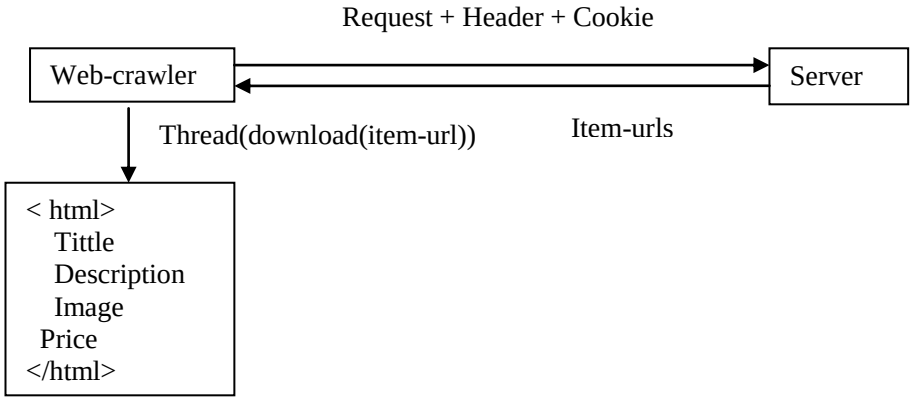

图 5　沃尔玛，速卖通平台爬虫设计

Fig.5　Crawler design for walmart and aliexpress

3.3 Wish 平台爬虫设计

Wish 平台是近几年电商界的新贵，专注于移动 APP 的，面向于国际贸易的电商。

在以上所有本文研究过的平台中，wish 平台的安全性方面是做的最好的。首先 wish 是具有 xsrf 认证即跨站请求伪造认证，防止有程序恶意冒充用户访问网站，而且此网站与速卖通一样是要求登录的，服务器会检查用户 cookie 中身份认证，发现不合法用户会拒绝所有的请求。

即使通过了 wish 的身份验证，wish 平台中所有的内容也全部是动态生成然后通过 js 嵌入到网页中的。需要去调用 wish 的 API 获得数据解析，但是 wish 的 API 并不会返回全部的商品目录，他总共只会返回总商品中的几百个商品，而且每次请求只能拿取总共商品中随机的几十个商品，因此多次返回的商品可能会有重复，去处重复商品后，可能总共请求 200 个商品结果只有 100 左右有效的商品信息。因此不仅增大了爬虫程序的爬取代价，同时也能保证几乎不可能爬取到其所有的商品信息。针对于 wish 平台，实际的操作流程比较复杂，因此简化的爬取流程如下：

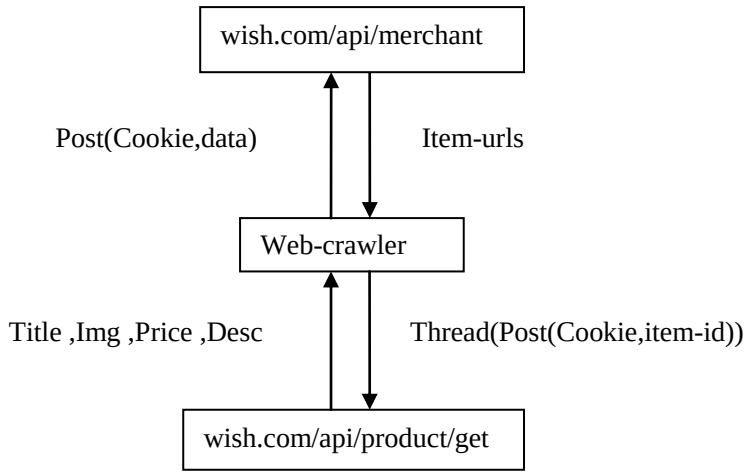

图 6 wish 平台爬取简化流程

Fig.6 Simplifying process flow of crawling wish

从图 6 中可看出，实际上 wish 平台的爬取和与 wish 网站的页面并没有任何关系了，网页上所有的数据完全是调用 API 后获得动态嵌入网页中的，其中值得一提的是，由于 merchant 接口单次请求返回的商品列表是完全随机的，且限定了总共的返回商品数量。因此调用这一接口后需进行去重操作且需要，多次重复的调用。而且每次请求中使用的是 post 方法，而不是其他平台常用的 get。并且较为麻烦的是，post 请求携带的参数颇多，且更新频繁，所以要求爬取程序每次爬取都要动态去生成，总的来说是以上介绍有比较有代表性的平台中爬取最为麻烦的了。

4 基于爬虫的系统设计与实现

整体的爬虫系统采用基于 web 的 ui 界面模块+爬虫模块+带有数据中心的数据库模块的设计，模块间相互独立，以下将分别介绍 这些部分。

4.1 前端界面模块

在前端本文采用了基于 web 的界面，界面使用 html+css 样式，后台的逻辑部分使用 JavaScript 与 Python 语言中的 web.py 库去实现。

用户在选择了是单个商品信息采集或者是店铺分类采信息集后，输入想要采集信息的店铺的网址，前端逻辑在检测输入网址是爬虫所支持的平台后，会将网址传送到爬虫后台进行爬取，进入下一个模块的处理环节。爬虫环节的处理流程本文将在爬虫模块中详细介绍。

而当爬虫模块爬取到有效信息后，在存入数据库之前，会将数据传输到数据中心模块中，而传送到数据中心的数据，用户都可以在前端界面中见到，方便用户进行处理与确认，当用户确认数据无误，进行认领操作后，数据才将真正写入数据库中。

	图片	标题	描述
	 来源：taobao	洗衣机专用漂浮除毛器清洁过滤网袋洗衣球去污吸毛去毛魔力滤毛器	出口国:亚洲 国家/地区:日本 品牌:angugu/安古古 货号:QMQ001 颜色分类:粉色2个 蓝色2个 粉色1个+蓝色1个 洗衣球优惠组合[购买搭配套餐] 商品类型:整理/收纳
	 来源：taobao	洗衣机专用文胸袋洗衣袋防变形洗护袋收纳袋加厚内衣网兜护洗袋子	出口国:亚洲 国家/地区:日本 品牌:angugu/安古古 货号:mx0304 颜色分类:A款-适合钢圈文胸[满2送1] B款-适合钢圈文胸[满2送1] C款-适合无钢圈文胸[满2送1] 商品类型:整理/收纳

描述	原价	促销价	创建时间	操作
出口国:亚洲 国家/地区:日本 品牌:angugu/安古古 货号:QMQ001 颜色分类:粉色2个 蓝色2个 粉色1个+蓝色1个 洗衣球优惠组合[购买搭配套餐] 商品类型:整理/收纳	24.75	9.90	2017-06-08 00:39:27	认领到 删除
出口国:亚洲 国家/地区:日本 品牌:angugu/安古古 货号:mx0304 颜色分类:A款-适合钢圈文胸[满2送1] B款-适合钢圈文胸[满2送1] C款-适合无钢圈文胸[满2送1] 商品类型:整理/收纳	17.25 - 22.25	6.90	2017-06-08 00:39:28	认领到 删除

图 7 抓取的数据

Fig.7 Crawl data

如若信息正确且符合用户的需求，则用户可以，点击操作栏中的认领按钮，通过数据中心将指定商品的信息写入到数据库中记录下来，如果商品信息不符合用户的需求，则用户可以通过点击删除按钮，将此条或多条商品数据移除，在此处将方便用户对数据进行批量操作。

在此处本文主要用到的是 JavaScript 语言动态生成信息表格与商品信息，利用 Python 语言及其第三方库 web.py 作为前端逻辑部分，负责与数据中心模块进行通信，对数据中心进行操作等。

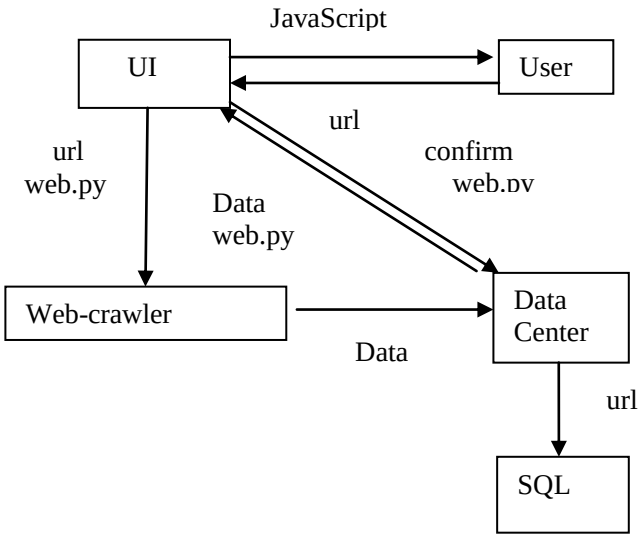

图 8 UI 界面工作流程

Fig.8 Working process flow of UI

4.2 爬虫模块

爬虫模块作为本系统中最为核心的一个模块，爬虫模块的功能与实现已在本文第二，三章做出了详细的描述。爬虫模块在整个系统中主要负责爬取由用户输入的链接指向的单个或多个商品的信息，以及对抗各个平台多种多样的反爬措施，为用户爬取到有价值的信息；爬虫获得页面信息后存贮器将页面及相关信息储存为本地文件，这些数据包括页面的 URL 地址，指向本页的 URL 地址，数据大小等（罗刚, 王振东，2010）。

在本系统中，爬虫模块全部都由 Python 语言写成，因为电商平台的反爬措施会时常更新，因此本模块的特殊性在于也需时常随着电商平台的更新而维护，作为整个系统中更新频率最高的一个模块，必须具有很高的独立性，以保证此模块的更新不需要波及到其他模块更改，以最大限度的减少系统后期维护的代价。

本模块中主要涉及到的技术是 html 页面解析与“反反爬技术”。这两项技术在本文的第二章与第三章有着详细的介绍，因此在此不再加赘述。在实际实现过程

中，除了这两项技术外，还需用到多线程调度和容错机制，还尤其需要注意编码问题。

在爬取分类目录下多个商品时，若是使用传统串行爬取，会导致系统爬取效率低下，当商品数量庞大时，系统效率会尤其低下，而且单次爬取的不顺利会严重的影响后续爬取。在此用到 Python 语言的多线程库，为每一个爬取商品单独指定一个线程，优化分配系统资源，使系统效率最大化；爬虫算法主要实现的是获取商品名称列表功能，通过反复调用 API 来获取商品信息，然后将商品信息插入到商品名称列表中，最终获得电子商品相关的商品名称列表（王弘巍，2012）。

此外，由于会在短时间内爬取大量的商品信息，因此有可能会有个别商品在爬取时出现错误与异常，如果不对这些错误进行捕捉处理，那么他们可能会出现 bug 重则导致系统意外停止或崩溃，轻则拖低系统效率。因此，需要在每次爬取时对错误进行捕捉和处理。在这里需要对每一个线程的运行进行监测，若出现错误，及时处理。在本系统中采取的措施是暂停该线程的运行，输出错误原因，记录下相关信息，等到系统运行结束，视情况重新抓取或放弃抓取。并且在抓取时，由于电商平台的服务器可能会有各种各样的状况，因此也需要在向平台服务器提出请求的环节进行错误和异常的捕捉，及时进行错误类型的分析，方便系统采取相应措施。这些都是容错机制的一部分，系统容错机制设置的好坏与否会直接关系到系统的健壮性。

最后还有一个需要注意的问题，电商网站网页的编码不尽相同，并有可能更新改变。并不是统一设置为“utf-8”就能一劳永逸的。因此本文推荐在抓取网页信息时，顺带抓取网页中的编码项以识别网页编码，但是这样做依然可能出现乱码，此时便需要人为查看并修复编码问题了。

4.3 数据库模块

数据库模块左右系统的最后一个模块，拥有着数据中心与数据库两个部分，以下将分别介绍。

4.3.1 数据中心

数据中心作为数据在写入数据库前必须经过的一个功能模块，其承担着对数据进行必要的处理，例如格式，编码，字符串处理，并将处理好的数据按属性字段返回给前端 ui 界面展示给用户，并接受处理用户的要求，对数据进行相应写入数据库操作或删除操作。

在数据库前加上这样一个模块，主要是考虑到爬虫模块爬取到的数据，不一定全是符合用户要求的，或者说数据中只有一部分是用户需要的，甚至可能因为电商网站的更新导致抓取数据不全，数据错误，数据乱码，数据非法等等情况。如果不设置这个模块抓取后直接写入数据库，会导致数据库数据质量不高，有错误和乱码问题，数据库数据冗余，规模庞大，导致数据库维护困难，从而导致各种后续的问

题，严重影响系统正常工作。虽然数据库方面可以进行相应的删除，查重等维护操作，但是频繁的不必要的操作数据库，势必会影响系统效率，而且数据库可能是多个不同系统共用的，如此的访问方式会引起安全问题，并且也不符合一个健康的多用户的分布式系统的设计准则。

因此本文在数据库前写这样一个模块，对数据进行处理，本模块的代码规模并不大，主要对数据进行一些规则性的处理，保证数据格式统一。并根据用户的要求对数据进行相应的处理，最后符合用户要求的数据才会真正的进入数据库。这样会增加整个系统的安全与健壮性。

本模块全部由 Python 写成，拥有数据处理与数据删写两部分。

4.3.2 数据库

数据库作为本系统永久储存数据的部分，要求安全，健壮，可维护。经过选择，本系统的数据库采用 SQL 数据库。SQL 数据库功能强大，简单易用，适合分布式系统的编写；考虑到本系统的应用规模和未来的业务扩展性能上的需要以及 MySQL 本身的性能，成本，可靠性和开源方面的优势（DuBois P et al，2011），本文采用 SQL 数据库，数据库拥有多个字段涵盖了商品信息的各个维度，方便日后的维护。

值得一提的是，在设计数据库时，一定需要严格按照数据库的设计规范来设计数据库。所谓数据库设计就是规范和结构化数据库中的数据对象以及这些数据对象之间关系的过程。数据库中的数据结构以及种类及其对象之间的关系建立都是影响数据库效率的重要决定因素。

设计数据库必须需要经过需求分析，概要设计以及详细设计阶段，必要时 E-R 图也可以帮助设计者实行有效的数据库设计；而要真正实现数据字典和 ER 图的实效性，就必须对数据库进行规范化的整理，对数据库使用过程中的各个名词进行专业化地统一，只有这样，才能够让数据字典和 ER 图起到应有的实效（郝进义，2012）。

一般来说，在设计一个高质量的数据库时，在以上三个阶段，都会有一些必须做的工作和步骤。

在需求分析阶段，需要做的工作有收集信息，标志对象，确定对象的属性以及确定对象间的关系。

在概要设计阶段，需要做的有，E-R 图的绘制，E-R 图与表的转换，三大范式规范化表格。所谓规范化是指关系型数据库中去除冗余数据的一个过程。往往一个符合规范的数据库的结构都会是最精简的形式而不会有冗余的行列和存在依赖关系的数据存在。想要获得一个高效的数据库，规范化是必不可少的一个步骤。

因此一个设计良好的数据库，能在保证数据完整性的情况下节省数据的储存空间，高效的查找效率，能够极大提升系统的效率和后续系统的开发。

5 总结与展望

5.1 工作总结

通过本文的面向电商的主题爬虫系统设计，我学习到了网络爬虫和网站结构以及系统设计相关的知识。本文主要的目的是实现爬虫相关技术，以及实现系统的相关设计，主要成果如下：

（1）对网络爬虫的理论和相关技术有了深刻的认识。对基础的网页解析算法，以及浏览器解析页面和渲染页面的机制有了了解。

（2）通过对淘宝，亚马逊，ebay，wish 等网站的研究，使我也对这些网站的结构和设计模式有了一定的了解。了解了他们从数据库中拿取数据，生成网页的方式与模式，对他们的安全机制也有了一定了解。

（3）了解了主流电商网站的反爬技术，学习了这些网站的安全部门人员为了保护公司的数据想出的各种巧妙的办法，同时自己也想出了针对这些反爬技术的措施，锻炼了自己的编程思维，提高了自己的编程水平。

（4）通过对整个系统的设计，锻炼自己整体系统设计思维，和着眼全局的眼光。终于理解了软件工程这门课程的精髓和良苦用心。

5.2 未来工作展望

虽然本文总体完成，但时间和精力以及学识的限制下，依然有一些略微的遗憾尚待学习和改进：

（1）软件的功能还比较单一，可以在软件中加入数据挖掘的功能，爬取每件商品的实时销量和评论，显示出每件商品实时的热度走势以及商品特征关键字。

（2）个人的审美和美观以及技术有限，系统的前端界面不够美观，提升界面的视觉感受和易用性。

（3）系统的算法和代码都有进一步提升和精简的空间。

参考文献

[1] 郝进义. 数据库设计规范及设计技巧研究[J]. 计算机光盘软件与应用, 2012,07(12):176-177.

[2] 刘汉兴, 刘财兴. 主题爬虫的搜索策略研究[J]. 计算机工程与设计, 2008, 29(12):3160-3162.

[3] 罗刚, 王振东. 自己动手写网络爬虫[M].北京：清华大学出版社, 2010.174-176

[4] 孙立伟, 何国辉, 吴礼发. 网络爬虫技术的研究[J]. 电脑知识与技术, 2010, 06(15):4112-4115.

[5] 汪涛, 樊孝忠. 主题爬虫的设计与实现[J]. 计算机应用, 2004, 24(s1):270-272.

[6] 王琨. 面向教育舆情的主题网络爬虫设计与实现[D]. 南华大学:南华大学图书馆, 2015.

[7] 王锋, 王伟, 张璟,等. 基于 Linux 的网络爬虫系统[J]. 计算机工程, 2010, 36(1):280-282.

[8] 王弘巍. 基于亚马逊网站的特定电子商品爬虫设计与实现[D]. 吉林大学：吉林大学图书馆, 2012.

[9] 肖旻, 陈行. 基于 Python 语言编程特点及应用之探讨[J]. 电脑知识与技术, 2014，08(12):8177-8178.

[10] 袁浩, 黄烟波. 网页标题分析对主题爬虫的改进[J]. 计算机技术与发展, 2009, 19(6):22-24.

[11] 周德懋, 李舟军. 高性能网络爬虫:研究综述[J]. 计算机科学, 2009, 36(8):26-29.

[12] 郑力明, 易平. 基于 HTMLParser 信息提取的网络爬虫设计[J]. 微计算机信息, 2009, 25(15):123-124.

[13] 中国互联网络信息中心，2016 年第 38 次中国互联网络发展状况统计报告[EB/OL].http://211.69.141.6/cache/5/03/cnnic.net.cn/f0644e33b3904bc04692a338099dbe79/P020160803367337470363.pdf ，2016-7.

[14] DuBois P, 杨晓云, 王建桥, 等. MySQL 技术内幕[M]. 北京:人民邮电出版社, 2011.305-306

[15] Analysis I S. Amazon.com, Inc. SWOT Analysis[J]. Amazon.com Inc.swot Analysis, 2014,01(12):1-2

[16] Hawking D. Web Search Engines: Part 1[M]. IEEE Computer Society Press, 2006.320-322

[17] Mark, Lutz, David,等. 《Python 语言入门》[J]. Internet:共创软件, 2002,08(10):86-86.

[18] Madhavan, Jayant, Ko, David, Kot,,等. Google's Deep Web crawl[J]. Proceedings of the Vldb Endowment, 2008, 1(2):1241-1252.

[19] Nistor C, Nistor R, Muntean M C. E-Commerce: Winners' Choice[C]. Venice, Italy: Wseas International Conference on Applied Computer Science. World Scientific and Engineering Academy and Society (WSEAS), 2010. 5-6

[20] Soulemane M, Rafiuzzaman M, Mahmud H. Crawling the Hidden Web: An Approach to Dynamic Web Indexing[J]. International Journal of Computer Applications, 2012, 55(1):7-15.