基于 Django 框架的关键词排名监控系统设计

濮文强，曹　磊，夏　斌
（上海海事大学 信息工程学院，上海 201306）

摘　要： 在 B2C 的电商交易平台上，商品的排名在很大程度上决定着商品的销量，但人工查看商品排名耗费时间且效率低下。目前市场上一些第三方查询工具不仅费用高，且查询时间较长。为更好地监控电商商品的排名信息，设计了一款基于 Web 的商品排名查询工具，实现了对商品的任意关键词进行快速的排名查询，且对已查询的关键词排名进行实时监控，定时更新其排名并提示相应变化情况。本系统主要基于 Python 语言进行开发，采用了 Django 框架进行 Web 平台的搭建。将该系统部署在安全稳定的 AWS 亚马逊云平台上进行使用，经过上线测试该系统达到了预期的效果。

关键词： 电商数据；排名；爬虫；Django
中图分类号： TQ35　　**文献标识码：** A　　DOI：10.19358/j.issn.1674-7720.2017.20.027
引用格式： 濮文强，曹磊，夏斌. 基于 Django 框架的关键词排名监控系统设计[J]. 微型机与应用，2017，36(20)：97-100.

Design of keyword ranking monitor system based on Django framework

Pu Wenqiang，Cao Lei，Xia Bin
（College of Information Engineering，Shanghai Maritime University，Shanghai 201306，China）

Abstract： On electronic business trading platform of B2C，the ranking of goods determines the sales of goods to a large extent，but viewing the product ranking artificially is time-consuming and inefficient. At present some third-party query tools in the market are expensive，and the query time is long. In order to monitor the ranking information of electronic business goods better，this paper designs a Web-based goods ranking query tool，to achieve a quick ranking query using any keyword of the goods，monitor the keyword ranking queried in real time，update the ranking regularly and prompt the corresponding changes. The system is developed based on Python language，using Django framework for Web platform structure. The system is deployed on AWS Amazon Cloud platform which is secure and stable，and achieves the desired results after online test.

Key words： electronic business data；ranking；the crawler；Django

0　引言

亚马逊作为当前国际的电商平台，其拥有十多个国际站点，对于海量的商品数据，亚马逊独有的 ASIN 码，有效地管理同一商品在不同国家的商品详情[1]。商品的销量和其排名情况关联度较高，排名越靠前的商品会被更早地浏览及购买，因此想要更高的销量就要对自己的商品的排名进行监控与提升。

文本旨在设计一种基于 B/S 结构的商品关键词排名监控，用户可建立自己的账户去添加在不同站点下想要查询的商品名称以及对应关键词下的排名信息，并且系统可自动对商品排名信息进行更新并提示排名的变化指标，整个系统提供批量查询以及管理等功能。在 B/S 结构下也要考虑优化用户的体验，系统在设计时需要将效率与准确性作为设计原则。

1　系统设计

1.1　系统架构

本系统是基于 B/S（浏览器/服务器）结构，这种结构将系统功能实现的核心部分集中到服务器上，用户无需下载与更新客户端，简化了系统的开发、维护和使用。客户机上一个浏览器即可与服务器进行数据交互[2]。

系统由 Web 框架 Django 搭建而成，Django 是开源的 Web 应用框架，由 Python 语言开发，采用了 MVC 的设计模式，将业务逻辑层、前端视图层、数据模型层以高内聚低耦合实现开发。采用 Django 可以简便、高效地开发基于数据库驱动的网站。Django 的优点是：(1) ORM 对象关系映射，便捷的数据模型设计与交互；(2) 管理员的管理界面；(3) URL 匹配；(4) 可扩展的模板语言；(5) 表单模型；(6) Cache 系统；(7) 内置国际化[3]。

前端主要采用 HTML、Javascript、Query、bootstrap 相结合，具有简单明了的数据显示以及更方便的用户操作。

系统结构及逻辑流程如图 1 所示。

1.2　系统模块结构

该系统模块主要分为以下七个部分。

(1) 注册登录模块

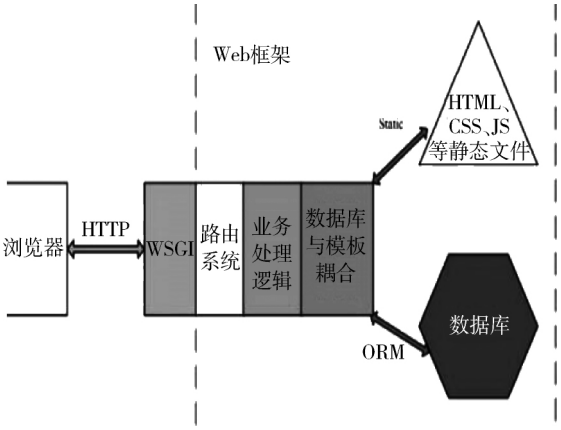

图1 系统组成结构及逻辑流程图

提供用户注册与登录，每一个用户拥有自己的商品排名信息库，用以保留每个用户查询记录以及管理商品查询记录，使得用户能更高效、快捷地管理自己的所关注的商品信息排名。

(2) 查询商品名称模块

由于相同的商品在不同国家的站点下商品名称不一样，本系统采用亚马逊提供的 ASIN 码去查询商品名称。

(3) 查询排名模块

系统业务的核心功能，用户可自定义多个关键词去查询商品在不同关键词下的排名情况。

(4) 文件上传模块

用户可通过模板文件一次多个产品及对应关键词，一次性查询多个商品的排名信息，查询结果可下载。

(5) 定时更新模块

通过定时任务对数据库中所有用户的商品进行关键词排名的更新，并分析与之前的排名变化趋势。

(6) 分页管理模块

用户可以查看到自己所有的查询记录以及变化趋势，对其进行增删改查等操作。

(7) 多线程网络爬虫模块

针对一个商品多个关键词同时查询的情况，创建对应个数的线程对排名数据进行抓取，避免同步逐个关键词查询而造成等待时间过长。大幅度地提高爬虫抓取信息的效率，提高用户的体验。

1.3 数据处理流程

用户在第一次进入系统时需要先注册，这是为了能对该用户所查询的商品进行记录与更新，登录完成之后，用户采用商品的 ASIN 首先查询在对应站点下的商品名称，而不是用商品名称去查询，这是为了让该系统可以服务于所有的站点。查询到商品名称后，即可以自定义添加多个关键词去查询在该关键词下的排名情况，在短暂的查询过程后，将结果显示在 Web 前端上，并且这些商品以及对应的关键词都会被保存到数据库中，用于定时地更新这些排名数据，并在用户下次进行查询时直接将更新结果显示了出来，而不需要再去等待查询结果就可以获取到最新的排名数据。用户在第二次登录之后就显示其查询的历史记录，不仅可以提高用户的查询效率，更减轻了服务器的实时压力。系统数据处理流程图如图 2 所示。

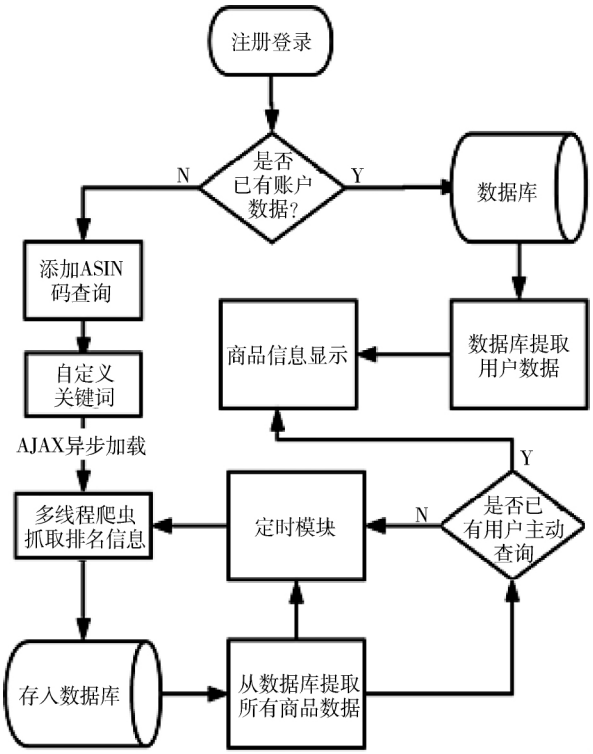

图2 系统数据处理流程图

2 系统实现

2.1 商品名以及排名数据抓取的实现

2.1.1 数据获取的方法

在没有官方提供的数据以及相关 API 接口调用的情况下，网络爬虫是获取网页数据信息的有力方法，也是该系统查询业务的核心。网络爬虫具有开发效率短、易编写、准确率高的特点。

2.1.2 多线程爬虫实现

多线程是指从软件或者硬件上实现多个线程并发执行的技术，能够在同一时间执行多个任务，进而提升整体处理效率。Python 提供了 threading 对多线程的支持，同步锁通过 my_lock = threading. RLock( ) 进行获取，threading. join( ) 将子查询线程保护，以便在子查询线程没有结束而执行主线程，因为主线程需要子线程所得出的数据。threading. setDaemon( ) 将线程声明为守护线程，必须在 start( ) 方法调用之前设置，如果不设置为守护线程，程序会被无限挂起。threading. start( ) 用以启动线程。

由于多个线程之中处理的关键词不同以及不同商品关键词数量不同，写入到的数据库位置也不相同，因此系统还需要对不同的线程进行标志以及与创建的线程数量进行对应，以保证数据写入的准确性。在遍历关

键词的列表时，通过获取关键词的下标来标识是第几个关键词，再依次遍历进行创建线程，保证线程数量的正确性。

系统采用基础单线程爬虫对商品的名称进行获取，通过用户输入的 ASIN 码以及选择的站点在后台拼接成对应的 URL，对该 URL 进行 request 请求，再对返回的响应信息采用对应的正则表达式抓取到商品名称。当用户添加多个关键词以及用户上传文件时，由于获取的是多个信息，此时采用多线程网络爬虫，多线程以一种并发的形式去执行任务，提高爬虫的效率，缩短任务时间。但是多线程之间共享变量，在保存数据到数据库进行写入操作时，会覆盖掉其他写入操作，导致查询结果只有一个关键词排名被写入到数据库中。本系统采用多线程锁的机制，当每个关键词排名结果信息需对数据库进行操作时，则加上锁，保证该时刻开始到数据写入结束，只有该线程可以对数据库进行操作，保证了数据库的一致性。

2.1.3 数据抓取的正则表达式

正则表达式是计算机科学的一个概念。正则表通常被用来检索、替换那些符合某个模式（规则）的文本。对于信息的抓取，原理是对相应的 HTML 源代码采用正则表达式去检索所需信息。例如商品对应的商品名为 The Paw for Dogs，Large。在网页的源代码中，在 < img 标签下属性 alt 记录着商品名称。代码匹配代码如下

```
Pattern = re.compile('<img alt=(.*?)' src=')
```

通过 < img alt 去检索到对应标签内容信息，.* ? 代表匹配到（）内的任意文本，也就是商品名称，再通过 src = 对于尾部的限定，准确地获取到商品名称的信息。

2.1.4 排名数据抓取的实现

已经通过 ASIN 以及选择站点查询到对应的商品名称，这时将用户所提交的关键词以及对应该商品的站点构建成页面 URL，该 RUL 是具有 page 属性的，也就是具有翻页的功能，通过对相应页面的信息与商品名称进行对比，若当前页没有匹配到，则跳转到下一页继续匹配，若跳转到商品的第六页时依然没有匹配到，认为该商品排名比较靠后。排名的准确数据是通过匹配的页数以及在当前页的排名次第计算而来的。即排名 =（前页数 - 1）× 每页的商品数量 + 匹配页名次。

2.2 数据库设计

系统设计排名数据离不开数据库，采用 MySQL 数据库用于存储数据，其体积小，速度快，总体成本低，特别是开源的特点使得 MySQL 作为一般网站首选的数据库。

本系统的 Django 框架将把数据库的操作封装成 ORM（对象关系映射）类对应表，属性对应字段，对象对应记录，使开发者更注重于表的定义与操作，而不是 SQL（结构化查询语句）[4]。可在系统的 setting. py 中配置数据库参数，在 model. py 中定义相应的 class，运行 python manage. py syncdb，即可在数据库中生成相应的表。系统的表定义为 Loginuser（用户表），该表主要存储所有用户的信息，也是用于数据库多表查询时的外键表；Showproduct（商品显示表）前端显示用户所查询的商品以及所添加的关键词；Upfile（上传文件记录表）用于查询上传文件商品排名信息，便于用户下载；ProductRank（商品排名信息表）记录所有用户每一天的所有商品关键词排名信息，用于展示排名数据以及定时任务。

系统的外键关联表关系图如图 3 所示。

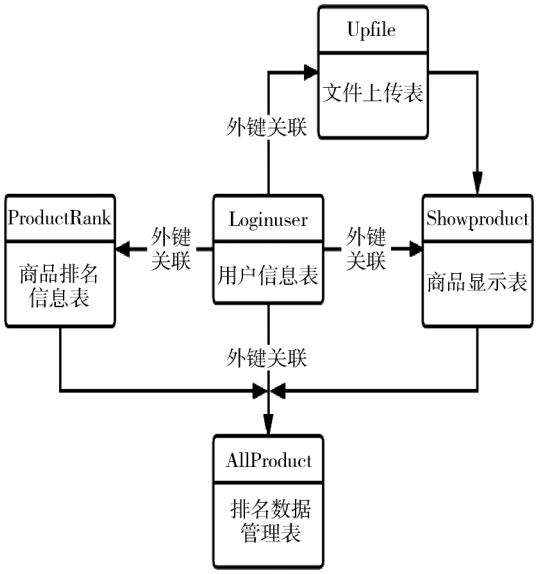

图 3 系统外键关联表关系图

2.3 定时更新的实现

2.3.1 Django 定时模块的原理

Django 框架的定时功能主要是利用 celery 模块实现[5]。这个模块实现方便只需要配置好定时的时间与任务即可，在 setting 文件中配置好定时的参数以及安装对应的插件，再创建 task. py 文件，将定时的任务代码函数编写在里面，但是需要加上装饰器@ task 才能将函数从普通函数包装成定时的函数，在程序运行时会自动扫描该文件。Django 框架的管理员界面可以更快地对定时任务的时间以及任务进行调整。

2.3.2 系统定时的实现

本文设计的系统中通过检索数据库对所有用户的商品排名数据进行定时的更新。首先读取当前日期，并对日期减去一天得到昨天的数据，将昨天的数据的基本信息以今天的日期写入到数据库中，多线程爬虫去获取此刻的排名情况，并与昨天的排名进行比较，将趋势也写入到今日的记录中去，实现排名的更新操作。

2.4 AJAX 的前端实现

AJAX（Asynchronous Javascript And XM）是一种交互式网页应用的网页开发，通过少量数据即可实现后台与服

务器的交互，使网页实现异步加载，简单地说就是可以在不重新加载整个页面即可更新页面的数据[3]。

本系统在效率原则下在关键词排名查询时采用 AJAX 技术，将商品的 ID 序号 POST 到后台，后台接受这个 ID，对应数据库找到对应商品，此时数据库中已经有关键词，将这条数据取出，再进行关键词排名抓取，将结果传到前段进行解析。整个过程数据提供轻量级 JSON 格式进行传输，并没有页面刷新，减轻了服务器压力，在多用户访问时不会出现等待情况，优化了体验效果。

3 系统测试

系统登录模块可实现将注册用户信息存入数据库中，并且验证用户信息是否正确。

通过 ASIN 码以及站点的商品名称查询，验证出查询的准确性，如图 4 所示。

图 4 系统外键关联表关系图

系统定时模块对数据排名进行对比，得出趋势，如图 5所示。

图 5 关键词排名变化趋势图

4 结论

在没有官方的数据获取 API 的情况下，基于网络爬虫来获取排名数据，利用 Web 框架搭建服务器方便快速地查询，并且定时监控排名变化趋势，既方便了使用者的操作，也节省了时间。系统的创新在于将传统的 C/S 结构转变为 B/S 结构，节省了客户端的维护更新，优化了用户体验；其次是充分利用亚马逊平台所提供的 ASIN 码实现跨国家站点的数据查询，不再是只能针对一个站点进行查询；最后将系统搭建在 AWS 的云平台上，可服务于所有的用户。经过多次实验验证，本系统基本实现了设计目标，在完成各项功能的同时优化了用户体验，提高了效率。

参考文献

[1] GOPALPUR C C, HALE C C. Online marketplace management system with automated pricing tool [P]. US: US7774238, 2010-08-10.

[2] 张友生，陈松乔. ClS 与 BlS 混合软件体系结构模型 [J]. 计算机工程与应用，2002，38(23)：138-140.

[3] 刘班. 基于 Django 快速开发 Wed 应用 [J]. 电脑知识技术，2009，5(7)：1616-1618.

[4] 王冉阳. 基于 Django 和 ORM 的 Web 开发 [J]. 电脑编程技巧与维护，2009，5(2)：56-58.

[5] SINGHAL N, DIXIT A, SHARMA A K. Design of a priority based frequency regulated incremental crawler [M]. LAP LAMBERT Academic Publishing, 2014.

[6] CRANE D, PASCARELLO E, JAMES D. Ajax in Action [M]. Manning Publications Co., 2005.

(收稿日期：2017-03-24)

作者简介：

濮文强(1994－)，男，硕士研究生，主要研究方向：商务数据挖掘与处理。

曹磊(1989－)，男，博士研究生，主要研究方向：脑机接口与智能信息处理。

夏斌(1975－)，通信作者，男，博士，副教授，硕士生导师，主要研究方向：脑-机接口、云计算及人工智能。E-mail：binxia@shmtu.edu.cn。

(上接第 96 页)

[11] 赵晶晶，吕雪，符杨，等. 基于双馈感应风力发电机虚拟惯量和桨距角联合控制的风光柴微电网动态频率控制 [J]. 中国电机工程学报，2015，35(15)：3815-3822.

[12] 侍乔明，王刚，马伟明，等. 直驱永磁风电机组虚拟惯量控制的实验方法研究 [J]. 中国电机工程学报，2015，35(8)：2033-2042.

[13] 蒋文韬，付立军，王刚，等. 直驱永磁风电机组虚拟惯量控制对系统小干扰稳定性影响分析 [J]. 电力系统保护与控制，2015，43(11)：33-40.

(收稿日期：2017-03-22)

作者简介：

李洋(1993－)，男，硕士研究生，主要研究方向：新能源发电技术及智能微电网控制技术。

王春明(1966－)，男，硕士，副教授，主要研究方向：新能源发电与智能微电网技术。

侯朋飞(1989－)，男，博士研究生，主要研究方向：新能源发电技术。