Computer System Networking and Telecommunications
计算机系统网络和电信，2019,1(2)
ISSN:2661-3719 (Print); 2661-3727 (Online)

Java - based Jingdong Mall Crawler Implementation

Yawen TANG Zhengde BAO Chenxi LI
School of Computer and Software, Jincheng College, Sichuan University, Chengdu, 611731

Abstract

Web crawler catching information is similar to spider catching mosquitoes, which is a program that can use Python, Java and other programming languages to achieve, so as to automatically obtain useful information on the network according to the rules specified in the program, and to filter and analyze to maximize the value of data. This paper summarizes the technology involved in the design of crawler, and USES Java language to design a dynamic web crawler system based on the large e-commerce platform shopping jingdong mall, analyzes the working principle of the crawler program, and shows the accuracy and speed of the crawler data collection.

Key Words

Jingdong mall, Web Crawlers, Data Mining, Java

DOI:10.18686/jsjxt.v1i2.686

基于 Java 的京东商城爬虫实现

唐娅雯 鲍正德 李晨曦
四川大学锦城学院计算机与软件学院，四川成都，611731

摘 要

网络爬虫捕捉信息类似蜘蛛捕捉蚊虫，是一个能利用 Python、Java 等编程语言实现的一个程序，从而按程序指定规则自动获取网络上有利用价值的信息，并加以筛选分析让数据价值最大化。本文概述了爬虫设计所涉及的技术，并利用 Java 语言基于大型电商购物平台京东商城设计了一个动态网页爬虫系统，浅析了爬虫程序的工作原理，展现了爬虫采集数据的准确度及速度。

关键字

京东商城；网络爬虫；数据挖掘；Java

1.引言

信息化时代的互联网、计算机网络等技术促进着人们生活方式的变革，电商购物平台使用率已逐渐超过传统购物平台利用率。基于数据的电商平台投入使用，随之而来的是海量的数据亟待高效的存储或处理，抓住有效信息合理利用才能让数据挖掘的作用发挥到极致。爬虫作为网页上捕捉如蚊虫般微小而杂乱的数据的“蜘蛛”，有助于准确捕捉分析电商购物平台的数据。为方便学习研究京东购物商城的数据，利用 Java 语言设计设计一个高效网络爬虫系统以挖掘电商平台的有益于优化决策的数据。

2.爬虫设计

2.1 主要使用的技术

2.1.1 HttpClient 技术

是一个 Apache Jakarta Common 下的支持 HTTP 协议的客户端编程工具包，以其高效的效率、丰富的功能而受青睐，HttpClient 的新版本相比传统 JDK 自带的 URLConnection 更为灵活，并支持 HTTP 协议最新的版本和建议。不仅简化了客户端发送 Http 请求的操作，同时让开发人员测试接口更为便利，以其简洁性加速了开发效率、增强了代码健壮性。

2.1.2 Jsoup 技术

作为 Java 的一款 HTML 解析器，可直接解析某个 URL 地址，HTML 文本内容。

1）能从一个 URL、文件或字符串中解析 HTML；

2）可使用 DOM 或 CSS 选择器查找、取出数据；

3）可操作的 HTML 元素、属性、文本；

2.1.3 Java 多线程编程技术

一条线程值得是进程中一个单一的顺序控制流，一个进程中可以并发多个线程，每条线程并行执行不同的任务。[1]

多线程是多任务的一种特别的形式，但多线程使用了更小的资源开销。多线程能够满足编写高效率的程序来达到充分利用 CPU 的目的。

2.1.4 阻塞队列技术

阻塞队列不同于普通队列，当队列为空时，从队列中获取元素的操作将会被阻塞，或者当队列满时，网队列里添加元素将会被阻塞。[2]试图从空的阻塞队列中获取元素的线程将会被阻塞，指导其他的线程往空的队列插入新的元素。在 Java 1.5 之后，java.concurrent 包下提供了主要有以下几个阻塞队列：

1）ArrayBlockKingQueue：基于数组实现的一个阻塞队列，在创建 ArrayBlockingQueue 对象时必须制定容量大小。并且可以指定公平性与非公平性，默认情况下为非公平的，即不保证等待时间最长的队列最优先能够访问队列。

2）LinkedBlockingQueue：基于链表实现的一个阻塞队列，在创建 LinkedBlockingQueue 对象时如果不指定大小，则默认为 Integer.MAX_VALUE。

3）PriorityBlockingQueue：以上 2 中队列都是先进先出队列，而 PriorityBlockingQueue 不是，它按照元素的优先级对元素进行排序，按照优先级顺序出队，每次出队的元素都是优先级最高的元素。[3]

4）DelayQueue：基于 PriorityQueue，一种延时阻塞队列，DelayQueue 中的元素只有当其指定的延迟时间到了，才能够从队列中获取到该元素。DelayQueue 也是一个无界队列，因此队列中插入数据库的操作永远不会被阻塞，而只有获取数据的操作才会被阻塞。

2.2 项目核心流程及代码

2.2.1 编写 HttpClient 工具类

```
public class HttpClientUtils {
    //创建 httpclient 连接池
    private static
PoolingHttpClientConnectionManager
connectionManager;
    static{
        connectionManager=new
PoolingHttpClientConnectionManager();
        //定义连接池最大连接数
        connectionManager.setMaxTotal(200);
        //对指定的网址最多只有 20 个连接

connectionManager.setDefaultMaxPerRoute(20);
    }
    private static CloseableHttpClient
getCloseableHttpClient(){
        CloseableHttpClient httpClient =
HttpClients.custom().setConnectionManager(connectionM
anager).build();
        return httpClient;
    }
    private static String execute(HttpRequestBase
httpRequestBase) throws IOException {

httpRequestBase.setHeader("User-Agent","Mozilla/5.0
(Windows NT 10.0; Win64; x64; rv:62.0)
Gecko/20100101 Firefox/62.0");
        //设置超时时间
        RequestConfig config =
RequestConfig.custom().setConnectionRequestTimeout(5
000).setConnectTimeout(5000).setSocketTimeout(10 *
1000).build();
        httpRequestBase.setConfig(config);
        CloseableHttpClient httpClient =
getCloseableHttpClient();
        CloseableHttpResponse response =
httpClient.execute(httpRequestBase);
        String html =
EntityUtils.toString(response.getEntity(), "utf-8");
```

```
            return html;
        }
        public static String doGet(String url) throws
IOException {
            HttpGet httpGet = new HttpGet(url);
            String html = execute(httpGet);
            return html;
        }
        public static String doPost(String url,
Map<String,String> params) throws IOException {
            HttpPost httpPost = new HttpPost(url);
            List<BasicNameValuePair> list = new
ArrayList<>();
            for (String key : params.keySet()) {
                list.add(new
BasicNameValuePair(key,params.get(key)));
            }
            UrlEncodedFormEntity entity = new
UrlEncodedFormEntity(list);
            httpPost.setEntity(entity);
            return execute(httpPost);
        }
    }
```

2.2.1 创建线程池和队列，开启线程

```
    //创建 Dao 对象
        static ProductDao productDao = new
ProductDao();
        //创建线程池
        static ExecutorService threadPool =
Executors.newFixedThreadPool(20);
        //创建原生阻塞队列   队列最大容量为 1000
        static BlockingQueue<String> queue=new
ArrayBlockingQueue<String>(1000);
        public static void main(String[] args) throws
IOException, InterruptedException {
            //监视队列大小的线程
            threadPool.execute(new Runnable() {
                @Override
                public void run() {
                    while(true){
                        try {
                            Thread.sleep(1000);
                        } catch
(InterruptedException e) {
                            e.printStackTrace();
                        }
                        //获得队列当前的大小
                        int size = queue.size();
                        System.out.println("当前队
列中有"+size+"个 pid");
                    }
                }
            });
            //开启 10 个线程去解析手机列表页获得
的 pids
            for (int i = 1; i <=10; i++) {
                threadPool.execute(new Runnable() {
                    @Override
                    public void run() {
                        while (true){
                            String pid=null;
                            try {
                                //从队列中取出
pid
                                pid =
queue.take();
                                Product product =
parsePid(pid);
                                //存入数据库

productDao.addProduct(product);
                            } catch (Exception e) {

e.printStackTrace();
                                try {
                                    //出现异常
则放回队列

queue.put(pid);
                                } catch
(InterruptedException e1) {
```

```
e1.printStackTrace();
                                        }
                                    }
                                }
                            }
                        });
                    }
        url="https://search.jd.com/Search?keyword=%E6%89%8B%E6%9C%BA&enc=utf-8&page="+(2*i-1);
                        String html =
HttpClientUtils.doGet(url);
                        parseIndex(html);
                    }
        }
```

2.3 核心实现过程的简要说明

2.3.1 关键词匹配

通过某一个指定的关键词匹配相关的手机商品信息详情页面。与关键词匹配则将该商品的数据按照预先设定好的字段分别存入数据库表中，并返回该商品包含的关键词字符串。筛选出于指定关键字相关的商品。

2.3.2 URL 队列

在数据库存放的 URL 队列中，需要将待爬取的队列和已爬取的队列分开。待爬队列是还没有爬取过的 URL；已爬取的队列存放的是爬取过的 URL。[4]为了对已经爬取过的页面进行去重处理，避免重复爬取同一个页面，首先判断 URL 是否存在于其中的一个队列中，将已存在 URL 的爬取队列中加入新的 URL；其次，初始化待爬取的队列，将 URL 作为种子 URL 开始爬取相关商品数据，从而提升爬虫工作效率和减少爬取数据的时间。

2.3.3 爬取频率控制

京东商城等大型电商平台对于异常访问流量的实施了访问限制和反爬虫机制，同样地对于普通用户的访问也有相类似的访问限制。在爬虫运行的过程中，如果爬虫访问过于频繁超出常规操作，该 IP 或者是该用户将会被封停，导致爬取数据的频率降低。为避免因用户被封停而降低爬取数据的频率，可设置单个 IP 的访问时长在保证效率的同时尽可能趋于正常水平，同时设置 IP 池，以用多个 IP 对数据进行爬取的方法保障用户处理大量数据时的需求。

2.3.4 URL 权重设置

由于网站的客户浏览量较大，商品信息中用户的评论会不定时更新，在一段时间后对于已经爬取过的商品信息，页面的用户评论数据会随之发生改变。为此不能把已爬取过的 URL 长时间地放在已爬取队列中，并且需要定时清理部分 URL，否则会导致内存产生溢出。所以需要根据计算机内存控制好已爬取 URL 的数量，在数量过多时释放部分 URL。同时对每个 URL 设置权值，控制好其在已爬取队列中的时间。[5]

3.数据简单分析

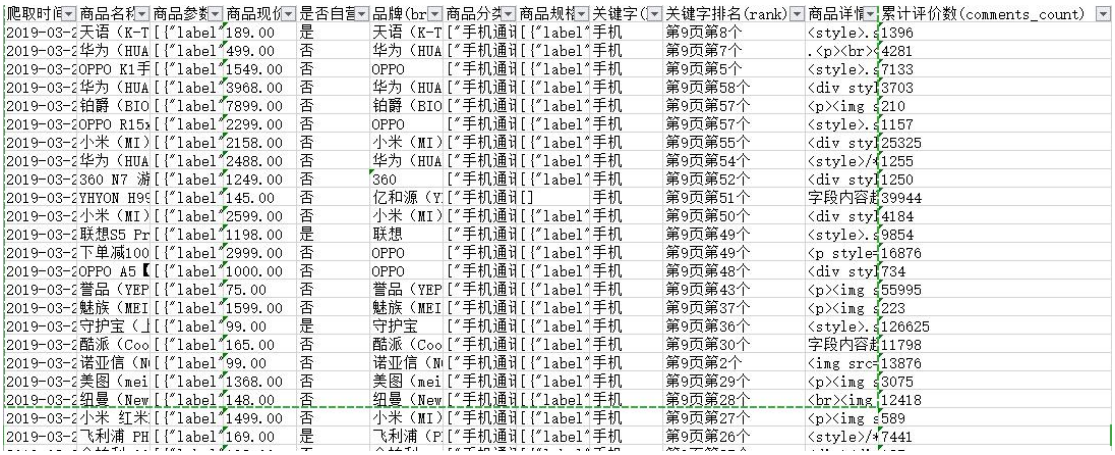
3.1 数据展示（部分）

经过清洗后的数据共 3651 条，包含爬取时间、商品名称、商品参数、商品现价、是否自营、品牌、商品分类、商品规格、关键字、关键字排名、商品详情、累积评价数 11 个字段。

3.2 简单的数据分析（分析结果仅来源于该爬虫数据）

3.2.1 关键字排名

以手机为关键字进行排名，前十五位排名分别是诺亚信（NOAIN）3310、华为荣耀畅玩 7 手机、天语（K-TOUCH）V6、Meitu 美图 T9、三星 Galaxy Folder2、中兴 ZTE Blade V10、魅族（MEIZU）16x、美图 T8s 手机、黑鲨 2 代 游戏手机、OPPO R17、vivo Z1、红米 Note7、纽曼（Newman）L660S、美图 V6、酷派（Coolpad）S618



爬取时间	商品名称	商品参数	商品现价	是否自营	品牌(brand)	商品分类	商品规格	关键字(keyword)	关键字排名(rank)	商品详情	累计评价数(comments_count)
2019-03-2	天语（K-T	[{"label	189.00	是	天语（K-T	["手机通讯	[{"label"	手机	第9页第8个	<style>.s	1396
2019-03-2	华为（HUA	[{"label	499.00	否	华为（HUA	["手机通讯	[{"label"	手机	第9页第7个	.<p> <	4281
2019-03-2	OPPO K1手	[{"label	1549.00	否	OPPO	["手机通讯	[{"label"	手机	第9页第5个	<style>.s	7133
2019-03-2	华为（HUA	[{"label	3968.00	否	华为（HUA	["手机通讯	[{"label"	手机	第9页第58个	<div styl	3703
2019-03-2	铂爵（BIO	[{"label	7899.00	否	铂爵（BIO	["手机通讯	[{"label"	手机	第9页第57个	<p><img s	210
2019-03-2	OPPO R15x	[{"label	2299.00	否	OPPO	["手机通讯	[{"label"	手机	第9页第57个	<style>.s	1157
2019-03-2	小米（MI）	[{"label	2158.00	否	小米（MI）	["手机通讯	[{"label"	手机	第9页第55个	<div styl	25325
2019-03-2	华为（HUA	[{"label	2488.00	否	华为（HUA	["手机通讯	[{"label"	手机	第9页第54个	<style>/*	1255
2019-03-2	360 N7 游	[{"label	1249.00	否	360	["手机通讯	[{"label"	手机	第9页第52个	<div styl	1250
2019-03-2	YHYON H9	[{"label	145.00	否	亿和源（Y	["手机通讯	[]	手机	第9页第51个	字段内容起	39944
2019-03-2	小米（MI）	[{"label	2599.00	否	小米（MI）	["手机通讯	[{"label"	手机	第9页第50个	<div styl	4184
2019-03-2	联想S5 Pr	[{"label	1198.00	是	联想	["手机通讯	[{"label"	手机	第9页第49个	<style>.s	9854
2019-03-2	下单减100	[{"label	2999.00	否	OPPO	["手机通讯	[{"label"	手机	第9页第49个	<p style=	16876
2019-03-2	OPPO A5【	[{"label	1000.00	否	OPPO	["手机通讯	[{"label"	手机	第9页第48个	<div styl	734
2019-03-2	誉品（YEP	[{"label	75.00	否	誉品（YEP	["手机通讯	[{"label"	手机	第9页第43个	<p><img s	55995
2019-03-2	魅族（MEI	[{"label	1599.00	否	魅族（MEI	["手机通讯	[{"label"	手机	第9页第37个	<p><img s	223
2019-03-2	守护宝（上	[{"label	99.00	是	守护宝	["手机通讯	[{"label"	手机	第9页第36个	<style>.s	126625
2019-03-2	酷派（Coo	[{"label	165.00	否	酷派（Coo	["手机通讯	[{"label"	手机	第9页第30个	字段内容起	11798
2019-03-2	诺亚信（N	[{"label	99.00	否	诺亚信（N	["手机通讯	[{"label"	手机	第9页第2个	<img src=	13876
2019-03-2	美图（mei	[{"label	1368.00	否	美图（mei	["手机通讯	[{"label"	手机	第9页第29个	<p><img s	3075
2019-03-2	纽曼（New	[{"label	148.00	否	纽曼（New	["手机通讯	[{"label"	手机	第9页第28个	 <img	12418
2019-03-2	小米 红米	[{"label	1499.00	否	小米（MI）	["手机通讯	[{"label"	手机	第9页第27个	<p><img s	589
2019-03-2	飞利浦 PH	[{"label	169.00	是	飞利浦（P	["手机通讯	[{"label"	手机	第9页第26个	<style>/*	7441

图 1 部分数据

关键字排名前十五位	
型号	排名
诺亚信（NOAIN）3310	1
华为荣耀畅玩 7 手机	2
天语（K-TOUCH）V6	3
Meitu 美图 T9	4
三星 Galaxy Folder2	5
中兴 ZTE Blade V10	6
魅族（MEIZU）16x	7
美图 T8s 手机	8
黑鲨 2 代 游戏手机	9
OPPO R17	10
vivo Z1	11
红米 Note7	12
纽曼（Newman）L660S	13
美图 V6	14
酷派（Coolpad）S618	15

表 1 关键词排名前 15 位

从以上数据可以看出，在关键词排名方面，用户的关注度并不一定是当年比较火热的国产高端智能手机，也可能是像诺亚信这种老年手机，或者是华为荣耀这种千元机。而图中大部分的手机除三星 Galaxy Folder2 外，其余的手机都属于以上两者的范围之内。一定程度上也反应出随着我国人口老龄化的加剧，老年人正在成为手机市场一个重要的消费群体。而千元机方面，主力的购买人群也是学生或者是收入处于相对较低水平的消费者。

3.2.2 价格排名

价格排名前十的手机分别是詹姆士（GEMRY）S100 高端商务手机、NOKIA 9 PureView（6GB+128GB 版）、OPPO 【兰博基尼版】FindX（8GB+512GB 版）、铂爵（BIOJUET）V8 商务高端手机、三星 W2018（6+256G 版）、华为 Mate RS（6+256G 版）、Apple 苹果 iPhone XS Max (A2104)、Apple iPhone XS、华为 mate20 Pro 8G+256G（UD 屏内指纹版）、iPhone Xs Max 全网通 64G 版。

ISSN:2661-3719 (Print); 2661-3727 (Online)

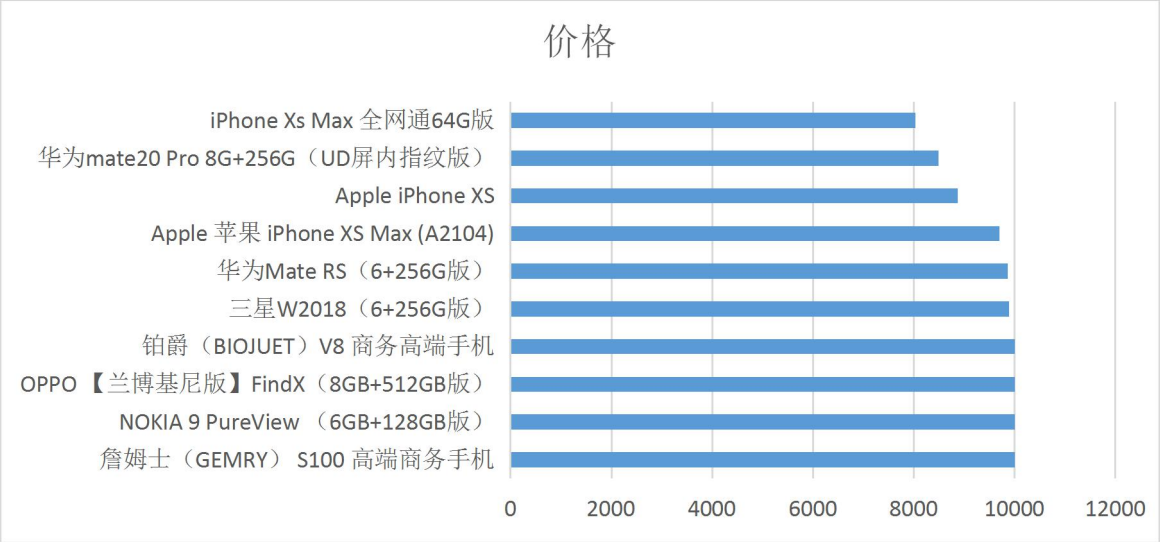


图 2 价格排名

从这个分析结果可以清晰的看出，该数据中价格最高的手机分别是詹姆士（GEMRY） S100 高端商务手机、NOKIA 9 PureView （6GB+128GB 版）、OPPO 【兰博基尼版】FindX（8GB+512GB 版）、铂爵（BIOJUET）V8 商务高端手机等四款手机，他们的价格均在 9999 未突破万元大关。而其余手机的价格与最高位相差不大。从手机型号来看，国产手机价格在前十位的少于国外品牌的手机，且排名相对靠后。说明国产手机在价格上，与国外手机品牌还存在着一定的差别。

4.结束语

采用 Java 结合流行的 HttpClient 等技术实现了一个灵活、高效、网页解析优秀的爬虫，为从数据流量巨大的京东商城灵活爬取数据奠定了重要的技术基础。通过该爬虫系统，负责后期数据可视化等的开发人员可用较快的速度获取所需信息从而节约时间、提升开发效率。爬虫系统设计者能实现大规模的京东商城购物平台数据的高效采集，并扩宽数据的采集面、提升数据采集量。实践结果证明，该系统具有可行性，相比于其他的数据采集方式，它更为实用。

参考文献

[1]纪莹莹.互联网 POI 同位模式挖掘方法研究[D].山东农业大学,2014.
[2]刘琛. 下一代网络业务执行环境中基于 SOA 的业务引擎的设计与实现[D].北京邮电大学,2010.
[3]董鹏.分布式实时事件服务的研究与实践[D].电子科技大学,2003.
[4]黎志雄,黄培灿.构建企业级的搜索爬虫[J].福建电脑,2008(12):93+97.
[5]陈珂,蓝鼎栋,柯文德,黎树俊,邓文天.基于 Java 的新浪微博爬虫研究与实现[J].计算机技术与发展,2017,27(09):191-196.

作者简介

第一作者：唐娅雯（1999-），女，汉，四川省资阳市，本科，四川大学锦城学院，研究方向：信息管理、J2EE
第二作者（通讯作者）：鲍正德（1989-），男，汉，黑龙江哈尔滨，研究生，四川大学锦城学院，研究方向：电子商务。
第三作者：李晨曦（1998-），男，汉，贵州省贵阳市，本科，四川大学锦城学院，研究方向：大数据技术开发