彳国科修技术衣偿

硕士学位论文



支持**AJAX**的网络爬虫系统

设计与实现

|  |  |
| --- | --- |
| 作者姓名： | 曾伟辉 |
| 学科专业： | 模式识别与智能系统 |
| 导师姓名： | 李淼研究员 |
| 完成时间： | 二O。九年五月十二日 |

University of Science and Technology of China

A dissertation for master5s degree



Design and Implementation of  
a Web Crawler System  
Supported AJAX

Author's Name: Weihui Zeng

speciality: Pattern Recognition and Intelligent System Supervisor: Prof. Miao Li

Finished time: May 12lh, 2009

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成 果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写 过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确 的说明。

作者签名： 签字日期：

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥 有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交 论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入有关数据 库进行检索，可以釆用影印、缩印或扫描等复制手段保存、汇编学位论文。本人 提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

口公开 口保密（ 年）

作者签名： 导师签名：

签字日期： 签字日期：

摘要

2005年J. J. Garrett提出了建立在JavaScript+XML基础上的AJAX的定义 与概念。该技术的特点是：用户操作与服务器响应异步化，达到了节约服务器空 间，提高网络交互效率的目的。因而被各大网站相继釆用。但由于目前的网络爬 虫技术在对AJAX框架网站的解析过程中存在着：URL参数几何级数的无效组合、 不能够识别事件触发顺序、缺乏错误捕获和处理能力等问题，导致大量数据不能 被搜索引擎有效检索。

本文针对上述问题，在对网页代码进行网页分析之后，通过研究基于对象的 程序切片算法，利用脚本执行引擎对切片进行计算执行，将与URL相关的切片程 序计算出来，重建网页D0M树。经过脚本执行引擎与切片模块的互操作，依据脚 本执行之后的网页DOM树抓取网页中的链接，最终解决AJAX框架中URL提取以 *及异步*JavaScript网络爬虫系统的关键技术问题，达到获取脚本中URL的目的。

本文实现了对AJAX站点的链接进行抓取的目的，设计出支持AJAX站点的网 络爬虫系统。在理论和技术方法上总结并提出AJAX框架网站中URL关联信息提 取、切片代码的有序执行以及程序切片模块、爬虫模块、脚本执行模块之间的互 操作，为AJAX框架网站网络爬虫提供了新的解决方案，设计实现了支持AJAX 站点的网络爬虫系统。

关键字：AJAX网络爬虫异步交互脚本解析

**ABSTRACT**

Jesse James Garrett put forward the concept of AJAX based on the JavaScipt and xml in 2005. The AJAX technologies improved technologies of web site development, increased user's interactive experience, and the client does not need to install any plug-ins. As a result, the AJAX technology has aroused widespread concerns in the area of the internet. However, because the current web crawler technologies cannot identify trigger order in the course of URL resolution of the AJAX framework, it led to the result that a lot of data are not able to be effectively searched by search engines.

In this thesis, to address the issue, by studying the object-based program slicing algorithm, we computed and implementated the slice with the script administer engine to rebuild the page DOM tree after analyzing the web code on the page. We can crawl on links in the page according to the web page DOM tree after the script implementation, and finally resolved the key technology question of the URL extraction and asynchronous JavaScript web crawler system in the framework of AJAX to fetch the URL.

We have achieved the purpose of crawling the AJAX site, designed the system of AJAX web site crawler in this thesis. In the theoretical and technical methods. We summed up and put forward the methods of extraction the URL information which the AJAX frame web site related, the orderly implementation of the code sections, as well as the interoperability between the program slicing module, reptiles module, script execution modules, and provided a new solution to the AJAX framework web sites crawler, designed and implemented the system of AJAX web site crawler.

**Key Words:** AJAX, Web Crawler, Asynchronous Interaction, Script Analysis

[摘要 I](#bookmark18)

[ABSTRACT II](#bookmark21)

[第1章绪论 1](#bookmark53)

1.1选题背景和意义 1

1.2相关研究 3

[1.3本文工作 5](#bookmark66)

[1.4本文结构 6](#bookmark77)

1. [5本章小结 6](#bookmark80)

[第2章网络爬虫以及相关研究 7](#bookmark83)

1. [1网络爬虫 7](#bookmark86)
2. [1. 1基本原理 7](#bookmark89)
3. [1.2工作流程 7](#bookmark92)

[2.2网络爬虫抓取策略 8](#bookmark99)

* 1. [1广度优先策略 8](#bookmark102)

1. [2深度优先策略 8](#bookmark105)
2. 2. 3最佳优先策略 8
3. [3深层网络爬虫 9](#bookmark108)
4. [3. 1深层网络爬虫的定义和研究目标 9](#bookmark111)

[2. 3. 2深层网络爬虫研究进展 11](#bookmark135)

[2. 3. 2. 1基于领域知识的爬虫技术 11](#bookmark138)

2. 3. 2. 2基于网页结构分析的爬虫技术 13

2. 3. 2.3基于脚本语言分析的爬虫技术 15

2. 3. 3深层网络爬虫的研究趋势 15

[2. 3. 3. 1关于AJAX技术的深层网络爬虫 16](#bookmark145)

[2. 3. 3. 2多媒体网络爬虫研究 16](#bookmark148)

[2. 3. 3. 3 对等网络P2P (peer-to-peer) 16](#bookmark151)

[2. 4 本章小结 16](#bookmark154)

[第3章AJAX网络爬虫系统设计方案 17](#bookmark157)

1. [1相关概念 17](#bookmark160)
2. [1. 1 Web2. 0 17](#bookmark163)
3. [1. 2 JavaScript 18](#bookmark166)
4. [1.3 AJAX及其特点 18](#bookmark170)
5. [2 AJAX网络爬虫系统总体设计方案 19](#bookmark179)
6. [3 AJAX网络爬虫的技术难点及解决方案 20](#bookmark182)

[3.4本章小结 22](#bookmark195)

[第4章AJAX网络爬虫系统实现 23](#bookmark198)

1. [1相关技术 23](#bookmark201)
2. [1. 1 HTTP 协议 23](#bookmark204)
3. [1.2 HTMI *24*](#bookmark228)
4. 1.3 XML 24
5. 1.4 XHTML 26

*4. 1.* 5 DOM 27

1. [1.6 XPath 27](#bookmark234)

[4. 1. 7 Web信息抽取、Web信息检索、Web挖掘 28](#bookmark248)

4. 1.7. 1网络编程基本概念及两种传输协议简介 28

[4. 1. 7. 2 Java网络编程 28](#bookmark266)

4. 2网页抓取模块 29

* 1. 1网页抓取模块流程 29

*4. 2. 2*网页抓取模块的相关实现 29

4. 2. 2. 1网络编程基本概念及两种传输协议简介 29

4. 2. 2. 2 Java网络编程 30

[4.3网页分析模块 32](#bookmark275)

[4. 3. 1网页中嵌入JavaScript的方法 32](#bookmark278)

[4. 3. 2网页分析模块流程 32](#bookmark281)

[4.4程序切片模块 33](#bookmark284)

[4.4.1程序切片流程 33](#bookmark287)

[4. *4. 2*关键算法 34](#bookmark290)

[4.5脚本执行模块 34](#bookmark293)

[4. 6程序切片模块、爬虫模块、脚本执行模块之间的互操作 35](#bookmark296)

[4. 6. 1构建JavaScript切片的有限状态机 35](#bookmark299)

[4.6. 2程序切片模块、脚本执行模块之间的互操作 35](#bookmark302)

[4.6.3程序切片模块、爬虫模块、脚本执行模块之间的互操作 36](#bookmark305)

[4. 7 DOM支持模块 36](#bookmark308)

[4.8页面生成模块 37](#bookmark311)

1. [9本章小结 38](#bookmark314)

[第5章AJAX网络爬虫系统测试 39](#bookmark317)

1. [1软件测试 39](#bookmark320)

[5.2测试内容 40](#bookmark323)

* 1. [1 AJAX框架站点的测试 40](#bookmark326)

1. [2. 2非AJAX框架站点的测试 41](#bookmark329)

[5.3结论 41](#bookmark332)

[第6章总结与展望 44](#bookmark335)

1. [1总结 44](#bookmark338)
2. [2展望——不足与改进 44](#bookmark343)

[参考文献 46](#bookmark346)

[致谢 49](#bookmark349)

[在读期间发表的学术论文与取得的研究成果 50](#bookmark352)

第1章绪论

1.1选题背景和意义

自2004年Web2. 0概念提出之后，互联网应用方式与几年前相比发生了很大 的变化。在此之前，人们应用互联网的主要方式是通过门户网站浏览新闻。而现 在以虚拟社区，SNS (Social Network Software) , Blog (博客)，Wiki (百科 全书)，RSS (Rich Site Summary)等为代表的更注重用户交互体验的Web应用 模式已经成为新的技术发展方向。

从技术上分析，传统的Web应用模式是通过表单形式完成的。当用户浏览网 页时，填写并提交访问表单，向Web服务器发送请求，服务器接收处理并返回一 个新的网页。该做法浪费了许多带宽。同时由于每次应用的交互都需要向服务器 发送请求，导致用户界面的响应远远慢于本地应用的需要。

1998年，微软开发出了允许从客户端脚本向服务器发送HTTP请求的组件 (XMLHTTP),奠定了 “支持异步请求”的技术基础。2005年美国用户体验咨询公 司总裁J. J. Garrett在《AJAX: A New Approach to Web Applications》文章中， 提出了建立在XMLHTTP组件之上，将JavaScript. XML的几种技术以新的方式组合 而成的“AJAX( ： Asynchronous JavaScript and XML)”的定义与概念［Jesse James Garrett, 2005］。该技术的特点是：用户操作与服务器响应的异步化。用户访问 以这种技术建设的网站时，通过异步模式向服务器发送并取回需要的数据，减少 了服务器和浏览器之间交换的数据量，也减少了Web服务器的处理时间；同时由 于该技术采用JavaScript处理来自服务器的响应，异步交互网络引擎运行在客户 端，可以使很多的处理工作在发出请求的客户机完成，提高了应用响应的速度， 降低了服务器的负载。

由于该技术釆用了异步交互网络应用方式，符合Web2. 0时代的需要，代表 了互联网领域新技术的发展，在提升用户交互体验的同时，又不需要在客户端安 装插件。因此，一经提出就引起了互联网领域的广泛关注。2004年以来，Google 在它的应用程序中成功地使用了异步交互网络技术，如Google讨论组、Google 地图、Gmail等。同时由于该技术支持Mozilla/Gecko,而被各大网站相继釆用。 各种异步交互网络框架如GWT, Atlas, Dojo也应运而生。

但随之而来的却是新的问题：由于AJAX框架网站是构建在异步JavaScript 基础上的Web应用技术，而传统Web搜索引擎中网络爬虫(Web Crawler)并不能 解决AJAX框架网站中的Web地址(URL)提取［曾伟辉，2008］。一方面是大量基 于AJAX的网站正在不断的涌现，另一方面这些网站的URL被搜索引擎忽略。这 意味着越来越多有意义的数据将无法通过搜索引擎检索。这一问题，也引起了国 内外学者的广泛关注，研究支持AJAX的网络爬虫对互联网领域的技术理论发展 和应用有重要意义［孙彬等，2007；郑冬冬等，2005］。

由于AJAX是建立在JavaScript基础之上的，而JavaScript是一种基于对 象(Object)和事件驱动(Event Driven)的语言。它所具有的弱变量、事件驱动执 行、异步回应请求的特点，使得目前网络爬虫技术在解决异步交互网络的Web地 址解析方面,还存在着如下的困难［Manuel Ivarez等，2001］:

1、 在解析异步交互网络地址(URL)过程中必须考虑JavaScript中事件触 发的顺序，而目前爬虫技术不能够识别事件触发顺序，无法构造出完整的URL。

2、 异步交互网络技术构造的网页中的URL通常带有参数，目前的爬虫技术 在解析URL过程中，不能够识别无效参数组合，从而导致大量无效空地址及网络 黑洞的产生。

3、 JavaScript是弱变量、解释型代码，其错误只有运行时才能被捕获。而 目前爬虫技术缺乏对这种错误捕获和处理能力，因而会造成地址解析过程中的异 常中断。

除上述问题外，异步交互式网络的AJAX框架所具有的如下特点也影响了现 有网络爬虫技术在其中的应用：

1、 AJAX框架层出不穷，据不完全统计，当前AJAX框架有200余种而且这 —数字随着时间的推移还会提高，这为设计通用于各种AJAX框架的爬虫带来了 很大的困难。

2、 AJAX框架有大量的表现层代码，网络爬虫在执行这些代码时将消耗许多 时间，而这些代码对提取URL没有任何作用，严重降低了爬虫的执行效率。

因此，开展异步交互网络地址解析方法硏究，对解决AJAX站点地址(URL) 的获取，检索隐藏在其中的大量数据，具有重要的理论意义与实用价值。

本实验室自80年代末从事网络搜索以及程序编译技术研究，在近年来的工 作中发现：利用程序切片［李必信.等，2006； M Harman等，2001；王伟等，2000； Harman M等，1995］在分解程序、捕获与程序某个变量在所有计算位置相关的程 序信息等方面的特性，有利于解决网络爬虫在解析JavaScript时找出URL变量 有影响的语句、控制谓词以及事件触发函数，降低代码执行的时间和空间复杂度 等问题，这主要表现在如下三个方面：

1、 程序切片技术能够自动剔除与目标变量不相关的函数和代码片断，可以 动态的分析解释型脚本语言执行过程中可能出现的错误，降低代码执行引擎所消 耗的时间。

2、 程序切片技术能够找出和目标变量相关的起始函数(事件)，为有限状 态机的执行提供了入口点(否则将不得不穷举所有可能事件)，避免爬虫陷入网 络黑洞。

3、 各个AJAX框架的主要区别集中在表现层的设计以及客户端与服务器的通 信协议(通信协议这一部分可以由代码执行引擎自动完成，无须爬虫干预)，通 过切片技术剔除表现层代码后，网络爬虫兼容各个AJAX框架将变得更加容易。

然而，利用程序切片技术解决异步交互网络中的地址解析，还面临着如下的 问题［李必信，2006］：

1、 目前的程序切片技术主要集中在处理面向过程和面向对象的语言，而AJAX 站点采用的是基于对象的语言(JavaScript),因此需要对基于对象语言的程序 切片方法开展相关研究。

2、 在解析AJAX网页时须考虑到事件触发的顺序，而传统爬虫技术不能够识 别事件触发顺序，无法构造出完整的URL,因此必须对JavaScript脚本的执行 算法开展相关研究。

本课题计划针对上述问题，开展相关研究，总结并提出基于AJAX站点脚本 切片以及地址信息获取方法，解决网络爬虫对异步交互网络中的地址动态跟踪与 捕获这一具有普遍意义的难题，为解析AJAX框架网站提供解决方案。

1.2相关研究

国内外有关网络搜索的研究十分活跃。自2000年以来，分别在美国、澳大 利亚等地举办了 WISE (WORKSHOP ON INFORMATION SYSTEMS ECONOMICS)、ACM International Workshop on Web Information and Data Management > ICDE (International Conference on Data Engineering)、ADC (Australasian

Database Conference）等会议；我国自2003年以来已分别由北京大学等研究单 位举办了六届全国搜索引擎和网上信息挖掘学术研讨会；自2004年以来由中国 计算机学会电子政务与办公自动化专委会主办了四届Web信息系统及其应用会 议。这些会议都对网络爬虫，以及站点搜索中脚本解析进行了深入的讨论。

目前，对于站点搜索中JavaScript脚本的处理，国内外的研究大多是采用 标准浏览器API构建迷你浏览器替代Web浏览器（如IE）的方法来处理脚本执 行代码［Manuel Alvarez 等，2001］ □西班牙 A Coruna 大学［Manuel Alvarez 等，2004］ 提出了一种使用标准浏览器API自动构建迷你浏览器替代http客户端处理脚本 执行代码，通过自底向上的递归算法来处理动态页面元素，提取动态网页中的 URL的方法。美国IBM公司［2005］开发的Web搜寻器，釆用在文档内容中查找有 可能是JavaScript语句中的URL的字符串，来达到提取动态网页中URL的目的。 美国SUNY大学釆用Rhino引擎，并扩展D0M解析器和监听模型来模拟执行 JavaScript脚本,提取脚本中的URL ［Ryan Levering等,2006］。瑞士苏黎世联 邦理工学院Cristian Duda等人［2008］设计的关于AJAX的网络爬虫，通过采用 广度优先方法，设置不同状态和不同时间的调用限制、最大深度限制以及相似度; 保持上下文信息：将所有的JavaScript变量作为状态的一部分加以维持；爬取 个性化信息，要求最少量的用户介入，只实现了企业级和用户说明的爬取。但是 对于一些与图书资料和IR社区相关的少数网页，还没有涉及到。荷兰代尔夫特 大学的Ali Mesbah等人［2008］对此也作了深入的探讨，通过动态分析和重建用 户界面状态变化来对AJAX页面进行爬取，动态分析一个能够模拟各种向导路径 以及AJAX应用程序中状态的状态流图，按照状态流图来建立静态页面。这样就 可以将动态交互的AJAX页面变成多个静态页面，使用一般爬虫就可以爬取。

中国科学院计算技术研究所利用Mozilla基金会开源的JavaScript引擎 SpiderMonkey,通过构造浏览器内置对象的方法提取动态网页中的URL ［Manuel Alvarez等，2004；王映等，2004］,浙江大学设计的支持AJAX网络爬虫原型系 统AjaxCrawler,采用了基于传统爬虫、JavaScript解释器和D0M支持的方法对 页面后加载内容进行了分析。

以上方法主要集中在对JavaScript脚本中的URL字符串常量的分析方面， 而对于需要按照特定的顺序触发事件才能获取完整URL变量的问题还缺乏有效 的解决方法［王映等，2004］ o

程序切片技术从1979年M. Weiser提出以来在国内外引起了人们的广泛关 注，目前已经有了许多程序切片工具。其中有美国、奥地利等研究机构分别开发 的支持 C、Ada、Oberon> Oberon-2、Java 等语言的切片工具,如：Wisconsin, Chopshop、Ghinsu, Menagerie、Unravel (支持 ANS I C)、PSS/ Ada 系统、The Oberon SI icing Tool (OST)、The Linz Oberon Slicing System > Indus > GrammaTech-. CodeSurfer、slicing tool、Spyder 等；我国山东大学、西安电 子科技大学等院校也相继展开了对程序切片技术的研究，南京大学、东南大学等 单位开发出的基于分层切片模型的Java程序切片工具[张勇翔，2000],目前已 经用于OOPQL语言环境中，并提供面向对象的Java程序的切片查询服务。而针 对基于对象的JavaScript语言的程序切片技术还没有开展系统的研究[李必信， 2006]O

2004年开始，中科院合肥智能所在国家科技攻关计划的支持下，开展了深 层网络信息获取方面的研究工作。针对动态网页信息获取问题，提出了半结构化 Web信息抽取算法，基于正则表达式的JavaScript脚本中Web地址过滤算法[王 伟等，2000],结合准对象交换模型(object ex-change model,简称OEM)和文档 结构抽取规则，研制了面向特定领域的主题网络爬虫以及Web信息提取模型。

1.3本文工作

本文针对传统网络爬虫无法爬取AJAX框架网站这一问题，提出了支持AJAX 框架的网络爬虫系统，为AJAX站点链接的爬取提供了新的解决方案，设计并实 现了支持AJAX框架的网络爬虫系统。主要研究以下两个方面：

1、 研究基于对象的JavaScript程序切片算法

1. 构建出JavaScript程序的依赖关系
2. 基于对象的系统依赖图构造方法研究
3. URL相关的程序片段切分与计算方法

2、 研究脚本代理执行技术及与切片模块的互操作方法

1. 构建JavaScript切片的有限状态机
2. 程序切片模块、脚本执行模块之间的互操作
3. 模块之间的数据传递方式设计

1.4本文结构

第二章介绍了网络爬虫以及相关技术。首先介绍了网络爬虫的工作原理、工 作流程以及现有传统网络爬虫所使用的几种抓取策略。接着介绍了深层网络爬虫 的相关知识。在对深层网络爬虫的定义和研究目标、深层网络爬虫研究进展介绍 分析的基础上，对深层网络爬虫的研究趋势作了分析与展望，充分强调了深层网 络爬虫的研究意义，也进一步说明了本论文研究的理论和现实基础；

第三章介绍了 AJAX网络爬虫系统设计方案。介绍了 AJAX网络爬虫系统涉及 到的相关概念，接着介绍了支持AJAX网络爬虫系统的总体设计方案，并对详细 介绍了 AJAX网络爬虫的技术难点及本文的解决方案；

第四章重点阐述了 AJAX网络爬虫系统实现。主要介绍了 AJAX框架网纟各爬虫 的具体实现，首先介绍了实现网络爬虫相关的技术，如HTTP协议、HTML语言等， 接着介绍了 AJAX框架网络爬虫系统各模块实现的关键算法、流程图以及涉及到 的相关技术；

第五章是对AJAX网络爬虫系统的测试。通过对AJAX框架站点和非AJAX框 架站点应用所设计的支持AJAX的网络爬虫系统进行测试，以及和传统网络爬虫 对上述网站的抓取做对比，得出实验结果并加以分析；

第六章是总结与展望，对本文的工作作了进一步的总结，对不足之处进行了 说明，并提出了改进建议。

1.5本章小结

本章首先介绍了选题背景和意义，接着介绍了相关的研究状况，最后大致概 括了论文主要的研究内容，以及章节的部署。

第2章网络爬虫以及相关研究

网络爬虫是一种自动提取网页的系统，它通过下载Web页面，并沿着已爬行 页面中的超链接来遍历Web,收集Web页面。Web爬虫通常用于通用搜索引擎中, 作为搜索引擎的页面收集系统，它通常以广度优先的模式，即无选择性遍历Web, 力求在限定的爬行周期内收集到尽可能多的Web页面。它的原理是从一个原始或 若干初始网页的URL种子开始，建立URL连接，对HTML文件进行解析，取出其 页面中的子链接，在抓取网页的过程中，不断从当前页面上抽取新的URL放入队 列。所有被爬虫抓取的网页将会被系统存贮，进行一定的分析、过滤，并建立索 引，以便之后的查询和检索。

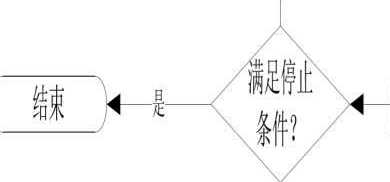
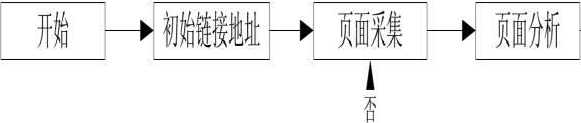
2. 1网络爬虫

2. 1. 1基本原理

网络爬虫是通过网页的链接地址来寻找网页，从网站某一个页面（通常是首 页）开始，读取网页的内容，找到在网页中的其他链接地址，然后通过这些链接 地址寻找下一个网页，这样一直循环下去，直到把这个网站所有的网页都抓取完 为止。如果把整个互联网当成一个网站，理论上讲网络爬虫可以把互联网上所有 的网页都抓取下来。

2. 1.2工作流程

通用网络爬虫是从一个或若干个初始网页的上的URL开始，获得初始网页上 的URL列表，在抓取网页过程中，不断从当前页面上抽取新的URL放入待爬行队 列，直到满足系统的停止条件［刘金红等，2007］。



其各个部分的主要功能介绍如下：

1、 页面采集模块：该模块是爬虫和因特网的接口，主要作用是通过各种We b协议(一般以HTTP. FTP为主)来完成对网页数据的采集，然后将采集到的页面 交由后续模块作进一步处理。

2、 页面分析模块：该模块的主要功能是将页面釆集模块釆集下来的页面进 行分析，提取其中满足用户要求的超链接，加入到超链接队列中。页面链接中给 出的URL一般是多种格式的，可能是完整的包括协议、站点和路径的，也可能是 省略了部分内容的，或者是一个相对路径。所以为处理方便，一般先将其转化成 统一的格式。

3、 链接过滤模块：该模块主要是用于对重复链接和循环链接的过滤。

4、 初始链接地址：提供URL种子，以启动爬虫。

2.2网络爬虫抓取策略

在抓取网页的时候，网络爬虫一般有三种策略：广度优先、深度优先和最佳 优先。

2.2.1广度优先策略

广度优先搜索策略是指在抓取过程中，在完成当前层次的搜索后，才进行下 —层次的搜索。该算法的设计和实现相对简单。在目前为覆盖尽可能多的网页， 一般使用广度优先搜索方法［eTesting L, 2000］。也有很多研究将广度优先搜索 策略应用于聚焦爬虫中。其基本思想是认为与初始URL在一定链接距离内的网页 具有主题相关性的概率很大。另外一种方法是将广度优先搜索与网页过滤技术结 合使用，先用广度优先策略抓取网页，再将其中无关的网页过滤掉。这些方法的 缺点在于，随着抓取网页的增多，大量的无关网页将被下载并过滤，算法的效率 将变低。对于下图2. 2,使用广度优先策略抓取的顺序为：A-B、C、D、E、F-G、 H-Io

2. 2. 2深度优先策略

深度优先是指网络蜘蛛会从起始页开始，一个链接一个链接跟踪下去，处理 完这条线路之后再转入下一个起始页，继续跟踪链接。这个方法有个优点是网络 蜘蛛在设计的时候比较容易。对于下图2. 2,使用深度优先策略抓取的顺序为： A-F-G； E-H-I : B； C； Do深度优先在很多情况下会导致爬虫的陷人(trapped) 问题，目前常见的是广度优先和最佳优先方法。［周立柱等，2005］

最佳优先策略按照一定的网页分析算法，预测候选URL与目标网页的相似 度，或与主题的相关性，并选取评价最好的一个或几个URL进行抓取。它只访问 经过网页分析算法预测为“有用”的网页。存在的一个问题是，在爬虫抓取路径 上的很多相关网页可能被忽略，因为最佳优先策略是一种局部最优搜索算法。因 此需要将最佳优先结合具体的应用进行改进，以跳出局部最优点。研究表明，这 样的闭环调整可以将无关网页数量降低30%〜90%。

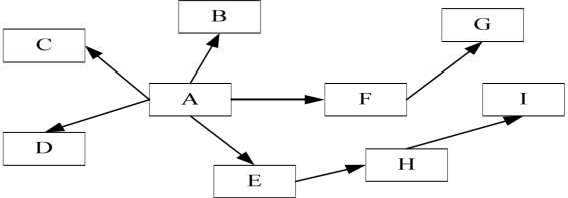


图2. 2抓取策略

2. 3深层网络爬虫

随着人们对信息重要性的认识不断加深，获取信息的方式越来越多，作为各 种信息的载体，网络蕴含着海量的丰富资源，包罗万象，是人类一笔宝贵的知识 财产。如何快捷的从网络上获取到自己所需要的有用信息成为人们急需解决的难 题。各种搜索引擎应运而生，例如传统的通用搜索引擎AltaVista, yahoo!、go ogle等。而通用的搜索引擎存在着一定的局限性，尤其是它们无法搜索到深层 网络(Deep Web)的信息。据BrightPlanet公司技术白皮书［2000］ , Deep Web 里包含的可访问公共信息容量是我们熟知的Surface Web的400-500倍。可见， 研究和挖掘Deep Web对于提高搜索覆盖率和准确率有着非常重要的意义。

2. 3. 1深层网络爬虫的定义和研究目标

互联网上的网页按其存在方式可以分为“表层网” (Surface Web)和“深 层网”(Deep Web,也有称 Invisible Web, HiddenWeb)［李涛等,2007］。

Surface Web是指传统网页搜索引擎可以索引的页面，以超链接可以到达的 静态网页为主构成的Web页面。

Deep Web 最初由 Dr. Jill Ellsworth 于 1994 年提出［郑冬冬等,2005］, 指那些由普通搜索引擎难以发现其信息内容的Web页面。

Christ Sherman、Gary Price 等人［2000］对 Deep Web 定义为：虽然通过 互联网可以获取，但普通搜索引擎由于受技术限制而不能或不作索引的那些文本 页、文件或其它通常是高质量、权威的信息。

文献［郑冬冬等，2006］对深层网络的定义为：网络上大部分内容是不能通过 静态链接获取的，特别是大部分隐藏在搜索表单之后的页面只有通过用户键入一 系列关键词才可以获得。这些页面被称为Hidden Web或Deep Web。

文献［李涛等，2007］对深层网络的定义为：Deep Web指那些存储在网络数 据库里、不能通过超链接访问而需要通过动态网页技术访问的资源集合。其中网 络数据库包括搜索引擎数据库、在线专业数据库和站内搜索数据库，统称为可搜 索数据库(Searchable Database)。

广义上来说，深层网络包含四个方面［Manuel Alvarez等，2004］：

1、 通过填写表单形成对后台再现数据库的査询而得到的动态页面。

2、 由于缺乏被指向的超链接而没有被搜索引擎索引到的页面。

3、 需要注册或其他限制访问的页面。

4、 可访问的非网页文件，如pdf, word文档等。

因为担心爬虫会陷入巨量动态网页库而浪费网络带宽资源和存储资源，以及 目前的技术还无法发现潜藏在网络数据库中的信息，所以传统搜索引擎，比如A ItaVista, yahoo!、google^百度等网页搜索引擎，一般只索引Surface Web中 由超链接可以到达的静态网页、文件等资源，却不索引或很少索引Deep Web中 的资源。

2000年Bright Planet公司曾针对Deep Web作过一个详细的调查，并将其 调查白皮书(The Deep Web: Surfacing Hidden Value)发布在互联网站点上。 该白皮书中提到的几点调查发现［2000］：

1. Deep Web里包含的可访问公共信息容量是我们通常定义的www的400 一 500 倍。
2. Deep Web包含7500TB的信息，而Surface Web包含的信息容量只有1 9TBo
3. Deep Web包含近5500亿独立文档，相对应的Surface Web只包含10 亿。
4. 现有的Deep Web站点已经超过100, 000个。
5. 60个最大Deep Web站点都已包含大约750TB信息，足以超过Surface W eb所包含信息的40倍。
6. 平均看，Deep Web站点的月访问量比Surface Web站点高出50%,并 且与Surface Web站点相比有更多的链接。可是那些典型的Deep Web站点在互 联网搜索领域却不知名。
7. Deep Web是互联网新信息增长的最大来源。
8. Deep Web站点在信息内容范围上比一般Surface Web站点更专更深。
9. Deep Web包含的有效高质内容总量至少是Surface Web的1000到200

0倍。

（10） Deep Web的信息内容与所有的信息需求、市场和领域高度相关；

（11） 超过一半的Deep Web内容都保存在专业领域的数据库中。

（12） 95%的Deep Web信息都是面向公共访问的，而不是需要付费或者订阅 的。

UIUC大学在2004年对深层网络作了一次估算［Chang K, 2004］,推测出整个 Web ±有307000个网络数据库站点，450000个网络数据库，比Bright Planet 在2000年估计的50000个又翻了许多倍。

通过上述研究可见，针对Deep Web的研究是进一步提高互联网信息获取质 量和数量的有效途径及可靠保障。但是利用通用网络爬虫在处理深层网络时通常 会遇到如下问题［Manuel Alvarez等，2004］:

1、 不具备处理浏览器端脚本代码中所有可能产生新的浏览器导航信息的机 制。

2、 不具备处理会话持久的相关机制，例如cookieo

3、 许多站点使用了复杂的重定向技术。

4、 不能处理的其他客户端技术，例如java applet, flash等。

5、 一些复杂的HTML代码，例如frames嵌套，https技术等。

2. 3. 2深层网络爬虫研究进展

现有的Deep Web爬虫技术大部分都是基于表单填写的爬虫研究，按照表单 填写方法基本可分为两大类：1、基于领域知识的表单填写。这种方法一般都有 一个本体库，通过语义分析来选取合适的关键词组合填写表单；2、基于网页结 构分析的表单填写。这种方法一般没有领域知识或者仅有有限的领域知识，将网 页表单构建成DOM树，在DOM树中提取表单各字段值。此外，还有一些爬虫技术 能够处理JavaScript语言。

2.3.2. 1基于领域知识的爬虫技术

Raghavan S.等人［2000］提出的IIIWE系统中Crawler管理器负责管理搜集 过程。它对下载的Web页面进行分析，包含表单的页面被送到表单处理器中专门 处理。表单处理器先从页面中抽取出表单，再从预先准备好的数据集中选择数据 自动地完成表单的填写，然后将合成的URL提交Crawler管理器去下载相应的结 果页面。由于需要系统自动完成表单填写，所以要求用户事先准备相应的表单数 据集。它通过LVS表来定义领域知识，通过视觉距离和文本相似度来识别表单。 不足之处是每个表单项只跟一个文本相关联，不能站在全局的观点上来处理表单 项，且不能处理JavaScript脚本。

Yiyao Lu［2007］等人提出一种获取Form表单的信息的多注解方法。该方法 首先将数据单元按语义平均分配到不同的组中，接着对每一组从不同的方面进行 注解，并且集合各种注解结果来预测一个最终的注解标签。但是当在相同的SRR 中的一个属性有多重数据单元时（例如一本书有多个作者），这种方法将会出错， 所以在这方面还需进一步完善。

Yoo Jung An ［2007］等人提出了一种通过分析网络语义自动生成领域本体知 识的方法，并用这种方法来获取deep web中的数据。

这个过程分成两大步骤：

第一步，从deep web中抽取属性。

文章将deep web中属性分成两大类：1、包含在网页标签中的（属性-值）。 2、查询界面中的提示字符串。文章利用wordNet以及自动属性抽取算法（AAE） 从网页中抽取这些属性。

第二步，利用属性间的IS-A关系构建有向无环图（DAG）,利用wordNet提 供的公共顶点将各个DAG合并，直到不能合并为止。

Zhen Zhang等人［2005］提出一种轻量级的基于领域知识的自动表单填写框 架。框架的核心是一种基于类型的搜索驱动方法，通过这种方法，能够将查询重 定向到一组相关资源的集合。

严亚兰［2003］提出了一种面向动态网页爬行的Crawler架构。它包括crawl 模块、表单分析模块、表单处理模块、结果分析模块、语义词表管理模块、UR Ls集和语义词表，并与搜索引擎中的索引数据集或网页数据集发生信息交互。

crawl模块控制并执行所有的爬行过程。crawl模块先从一个种子URLs集 出发开始爬行，对爬回的静态网页，crawl模块进行以下几方面的处理:从网页 中抽取所有的由链接指向的URLs ,并将这些URLs存入到URLs集中；将爬回 的网页保存在网页数据集中（如果存在），或者保存在缓存中足够长的时间，一 方面使索引模块完成索引任务并将索引数据保存在索引数据集中，另一方面保证 表单分析模块完成对网页表单的分析。URLs集为crawl提供将要爬行的静态网 页地址，表单处理模块为crawl提供将要爬行的动态网页的表单提交信息。era wl模块对爬回的动态网页除了执行以上几方面的处理之外，还将网页提交给结 果分析模块进行分析，以优化后面将要进行的动态网页爬行。同一表单的动态网 页爬行数量由表单处理模块决定，爬行完某一个表单的预定数量动态网页之后， crawl从URLs集中选择下一个将要爬行的URL地址重复前面的过程。

郑冬冬等人［2006］提出的爬虫，首先针对站点接口产生一个查询；然后下 载结果的索引列表页面：最后根据结果的索引下载具体页面。其中査询关键字选 择策略釆用了三种方式：随机选择方式，根据词频选择方式，适应性策略方式：

分析以前査询返回的结果文档，然后估计下次使用哪个关键词可以返回最多的文 档资资源。估计匹配的页面数量时综合考虑独立估计量和zipf估计量［2003］的 情况，其中术语频率f被定义为f=a(c+b)-r。式中：c为术语的排列等级，a, b, r为由文本集决定的常量。可以使用上述公式来估计P(q )的值。查询选择算法 主要是通过Efficiency (qi)=P(qi) / Cost(qi)来计算查询qi单位代价下获取新 的文档页面数，以此获得效能值最大的那个关键词。

Manue 1 Alvarez等人［2007］提出的DeepBot基于一系列的领域知识，每 种领域定义描述了一个数据采集任务。它采用了基于视觉距离和角度的方法计算 表单各项的最佳项值，距离超出阚值的文本舍弃，距离相等的文本比较该文本与 表单项的角度，表单项与文本之间的文本相似度通过文献［Cohen, W等，2003］中 的TFIDF和Jaro-Winkler edit-distance算法来计算。并且通过各表单项的相 似度计算表单与主题的相似度，来识别与主题相关的表单，然后模拟填写表单， 基于对象和客户端脚本的事件触发机制提交执行表单来得到hidden web中隐藏 的信息。这种方法不适于快速搜索和加密的表单。

1. 3. 2. 2基于网页结构分析的爬虫技术

Bergholz等人［2003］针对全文本搜索表单的Deep Web进行处理，这种网 页的搜索表单只有一个输入关键字的表单项。Alexandros Ntoulas等人［2005］ 在文献中可以针对前面的搜索结果自动产生新的搜索关键词，并对它们进行优先 级排序来获得隐藏在表单后的信息。优先级通过E(q)=P(q)/C0ST(q)来计算，其 中COST(q)为查询q的代价(通常为时间，以及带宽的消耗)，P(q)为査询q的 召回率，即单项的表单。返回的网页数占总网页数目的百分比。该方法优点是可 以使用最少的提交次数得到用户想要的数据，但是该方法不能处理含多形式表

Stephen. W. Liddle等人［2002］提出一种可以抽取出隐藏在表单后面的全部 内容的方法。首先，根据网页信息构建D0M树，然后在D0M树中提取form子树。 解析form子树，将form各个字段的取值提取出来，构造出带参数的URL,如果 某一字段默认值为空，则保留为空(例如文本框)，然后向服务器发送这个url 请求。如果数据召回率、查询次数、接收总字节长度、爬行时间、连续返回的空 值次数这几项指标的某一项达到预定值则结束，否则取表单各字段值的下一组待 选组合，继续向服务器发送请求。这种方法缺点是不能处理含有textbox表单项 的表单。

Cope JE2003］等人提出了一种判断页面是否含有查询接口的方法：基于查询 接口特征，利用决策树实现对接口的识别，其过程分为两大步骤：根据一定判别 规则产生查询接口，然后采用C4.5决策树算法，根据决策树找出真正的查询接 口。

Ali I. El-Desouky等人［2006］提出一种LEHW的方法,该方法通过一个解 析器(将HTML网页表示成一个D0M树形式)来区分(S-A)和(M-A),索引表 单通过两种不同的数据结构，一种是针对S-A表单，另一种是针对M-A表单，然 后分别通过页面标签判断进行处理。这种方法不仅可以对multi-attribute (M- A)型表单进行处理，而且对现有的single-attribute (S-A)型表单处理技术 进行改进。但是不足之处在于LEHW方法在提取标签方面精确度没有HIWE［Ragha van等，2000］方法高。

陈坷等人［2007］提出了一种策略，以两阶段采样策略确定是否充分获取了后 台数据库数据，首先使用表单提供的默认值来提交表单，然后对表单元素值组合 进行釆样以确定默认值提交是否返冋了后台数据库的所有数据，若返冋了后台数 据库所有或大部分数据则可以结束提交过程，否则，在爬虫所具有资源限制范围 内穷尽所有可能值的组合。在采样查询后台数据库时，主要釆用了分层采样和复 本检测的方法。这种方法不足之处在于它只能获取一部分Hidden Web页面，且 无法处理文本域元素。

Luciano Barbosa等人在文章［2007］中描述一种自适应抓取策略，以有效地 找到切入点隐藏web资源。事实上，隐藏网页的来源是稀疏分布的，这使深层网 络中主题定位非常有挑战性。作者在处理这个问题通过网页的内容将爬行定位到 某一个主题，优先考虑那些和主题相关的链接，并跟踪那些可能不会导致立即受 惠的链接。作者提出了一个新的框架，以自动学习模式并调整爬行方向，从而大 大减少了手工设置和调整。通过实验证明，这种方法收获率要比只使用一个固定 的主题提高三倍。由此我们可以在设计爬虫时适当考虑那些非主题链接，在选取 链接时，自动学习模式是值得借鉴的

孙彬等人在文章［2007］中提出一种基于XQuery的搜索系统。系统对表单和 特殊页面标记切换进行模拟，把页面关键字切换信息描述为三元组单元，按照一 定规则先进行盲搜索，排除无效的表单。然后将web文档构造成DOM树，利用X Query完成文字属性映射到表单字段的识别过程。此方法在处理常规结构的站点 时其爬行覆盖率达到了 71%以上。但是对于现在大量出现的非常规结构的站点， 我们也应该给予重视，提岀好的处理技术才行。

宋晖等人［2002］提出了一套可以自动地査询Hidden Web信息的系统HWIR (Hidden Web Information Retrieval) □用户可以输入索引主题及相关文本来检 索包含了这些Hidden Web信息的Web页面。HW1R利 用Crawler来收集网页， 使用对象抽取技术从网页中分析Hidden Web中数据库的人口 Form表单，然后 自动建立Hidden web信息的索引，用户可以通过结构化的查询获取自己需要的 Hidden web 信息的网页。文中采用的 TrOE (Tag-Tree-based Object Extracti on）技术是一种基于标记树结构的表单抽取算法。它首先将web页面表示成树型 结构，整个工作由四个步骤组成：网页标准化，树结构生成，表单分析，HwE对 象索引。先为每一个form子树建立一个HwE对象，然后找出该子树范围内所有 标识的表单成员，得到抽取对象的属性A和操作M,再完成对象名K的抽取。按 照深度优先原则遍历整个树，将前后两个距离结点最近的叶子结点作为对象取值 的候选结点。如果有一个候选结点是该结点的前序结点，则另一个结点被抛弃。 如果没有一个候选结点是结点的前序结点，则取后面的候选结点。如果没有候选 结点，则取其中一■个值（根据不同的InputType选择方法不同）。HWE对象的检 索归结于文本数据相似性利用基本的词频统计方法VSM（Vector Space Mode）算 法构造查询算法。文中对象抽取主要针对HTML、XML中的文本信息。对用ASP、 JSP的代码没有进行分析，而很多的Hidden Web信息的入口就隐藏在这些代码 中。

2. 3. 2. 3基于脚本语言分析的爬虫技术

目前基于脚本语言的爬虫技术，通用的方法［王映等，2004］是用脚本分析引 擎模拟浏览器的动作，真正执行脚本代码，这样无论网页中的脚本程序多么复杂, 都能得到正确结果。开放源码的JaveiScript引擎SpiderMonkey提供了一个最基 本的旦易于扩展的JavaScript分析器。通过包装SpiderMonkey,使其包装后的 接口能接收从页面提取的JavaScript代码，返回执行JavaScript后得到的所有 URL,从而完成爬虫任务。

Manuel Alvarez等人［2001］提出另一种解决客户端隐藏网络的解决方案。这个 方案主要包含三个步骤：

1、 将网页文档当作路由表来处理会话持久问题。

2、 使用标准浏览器API自动构建迷你浏览器替代http客户端处理脚本执行 代码，页面重定向。

3、 通过一种自底向上的递归算法来处理弹出菜单，以及其他动态页面元素。 上面的两种解决方案为我们着手脚本代码的研究指明了方向，我们可以在此

基础上进一步的改进及完善。

2.3.3深层网络爬虫的研究趋势

从网络信息生产的趋势看，越是价值高、规模大的信息往往越深藏在深层网 络中，而现在大部分的网络爬虫都无法对深层网络中的Flash和Script等动态 网页和数据库进行釆集。当前对于Deep Web爬虫技术的研究大多只是针对表单 的，少数针对JavaScript脚本代码的Deep Web处理［王映等，2004］。深层网络 爬虫的研究将趋向于以下几个方面：

2. 3. 3. 1关于AJAX技术的深层网络爬虫研究

AJAX技术现在已经被广泛的使用在网页中。Google的Orkut, Gmail,以及 最近的 beta 版的 Google Groups、Google Suggest 和 GoogleMaps,都应用了 A JAX这项技术。Flickr、Amazon的A9. com搜索引擎也釆用了类似的技术。Mic rosoft 已经推出了 Atlas 的 8 版，它在 ASP. NET 中实现了 Ajax。BEA Systems 公司正在把AJAX功能构建到它的门户产品中并把AJAX API加入运行时工具。S un Microsystems公司计划把Ajax加入Java Server Faces。企业服务总线供应 商CapeClear Software公司则计划把AJAX工具加入它的以SOA为中心的产品 中。AJAX的广泛研究和应用，使得以AJAX为基础的新一代javascript网络站 点抽取信息问题的研究显得越来越重要。

通过我们对国内外研究进展的探讨与分析，目前尚未有人针对此类网页进行 爬虫技术的研究。这方面的研究也将成为Deep Web爬虫需要处理的技术难点之

**O**

2. 3.3. 2多媒体网络爬虫研究

随着超媒体技术和宽带网技术的发展，开发可査寻图像、声音、图片和电影 的搜索引擎是一个新的方向[彭建荣等，2006] o因特网上图形、图像、视频 音 频、动画等多媒体信息正日渐丰富，与此同时，用户对其检索的要求也在不断增 长.伴随着搜索引擎的发展，各种基于网络的多媒体爬虫技术研究将会成为爬虫 研究的新方向。

2. 3. 3. 3 对等网络 P2P (peer-to-peer)

对等网络在加强网络上人的交流、文件交换、分布计算等方面大有前途。长 期以来，人们习惯的互联网是以服务器为中心，人们向服务器发送请求，然后浏 览服务器回应的信息。而P2P所包含的技术就是使联网电脑能够进行数据交 换.但数据是存储在每台电脑里，而不是存储在既昂贵又容易受到攻击的服务器 里。网络成员可以在网络数据库里自由搜索、更新、冋答和传送数据。所有人都 共享了他们认为最有价值的东西，这将使互联网上信息的价值得到极大的提升。

2. 4本章小结

本章首先介绍了网络爬虫的工作原理、工作流程以及现有传统网络爬虫所使 用的几种抓取策略。接着介绍了深层网络爬虫的相关知识。在对深层网络爬虫的 定义和研究目标、深层网络爬虫研究进展介绍分析的基础上，对深层网络爬虫的 研究趋势作了分析与展望。

第3章AJAX网络爬虫系统设计方案

自Jesse James Garrett提出了 AJAX概念以来，由于AJAX在提升用户交互 体验的同时，又不需要在客户端安装插件。因此，一经提出就引起了互联网领域 的广泛关注。但目前的网络爬虫技术在AJAX框架的URL解析过程中存在着不能 够识别事件触发顺序等问题，导致大量数据不能被搜索引擎有效检索。本论文针 对此问题，通过研究基于对象的程序切片算法，以及脚本执行引擎与切片模块的 互操作，最终解决AJAX框架中URL提取以及异步JavaScript网络爬虫系统的关 键技术问题。

3.1相关概念

3. 1. 1 Web2. 0

Web2.0［佚名，2005］是相对Webl. 0 （2003年以前-的互联网模式）的新的一 类互联网应用的统称，是一次从核心内容到外部应用的革命。由Webl. 0单纯通 过网络浏览器浏览html网页模式向内容更丰富、联系性更强、工具性更强的We b2. 0互联网模式的发展已经成为互联网新的发展趋势。

Webl.O到Web2.0的转变，具体的说，从模式上是单纯的“读”向“写”、 “共同建设”发展；由被动地接收互联网信息向主动创造互联网信息迈进！从基 本构成单元上，是由“网页”向“发表/记录的信息”发展；从工具上，是由互 联网浏览器向各类浏览器、rss阅读器等内容发展；运行机制上，由“Client S erver”向"Web Services"转变；作者由程序员等专业人士向全部普通用户发 展：应用上由初级的“滑稽”的应用向全面大量应用发展。

总之，Web2. 0 是以 Flickr、Craigslist、Linkedin, Tribes、Ryze> Fri endster, Del. icio. us> 43Things. com 等网站为代表，以 Blog、TAG、SNS、RS S、wiki等应用为核心，依据六度分隔、XML、AJAX等新理论和技术实现的互联 网新一代模式。

自2004年Web2. 0概念提出之后，互联网应用方式与几年前相比发生了很大 的变化。在此之前，人们应用互联网的主要方式是通过门户网站浏览新闻。而现 在以虚拟社区，SNS （Social Network Software） , Blog （博客），Wiki （百科 全书），RSS （Rich Site Summary）等为代表的更注重用户交互体验的Web应用 模式已经成为新的技术发展方向。

1. JavaScr i pt

JavaScript是一种基于对象的编程语言，不同于其它面向对象的语言，它 没有类的概念，只有对象。JavaScript语言的对象有三个来源：JavaScript内 部对象、主机环境中对象、程序创建的对象。

任何对象都可以作为原型对象与另一个对象联系起来，允许后一个对象来共 享前一个对象的所有属性。JavaScript提供动态继承，继承可以根据单个对象 的不同而不同。而且还支持不需要任何声明的函数，函数可以是对象的属性，作 为宽松类型的方法被调用执行。

任一个对象都可以定义自己的属性，不管是在创建阶段还是在运行阶段。J avaScript在运行时可以动态的增加或者删除任何对象的属性。如果为一组对象 的原型对象增加一个属性的话，那么所有继承于这个原型对象的所有对象都可获 取这个新增加的属性。

JavaScript没有Java等面向对象语言所具有的静态类型，也没有严格的类 型检查机制。但JavaScript支持大部分的Java语言的语法和控制流结构。Jav a等面向对象语言的类是通过声明来创建并在编译阶段就固定好了的，而JavaS cript支持基于以下几个基本数据类型的运行时类系统：数字类型，布尔类型和 字符串类型。

1. AJAX及其特点

1998年，微软开发出了允许从客户端脚本向服务器发送HTTP请求的组件(X MLHTTP),奠定了 “支持异步请求”的技术基础。2005年美国用户体验咨询公司 总裁 J. J. Garrett[2005]在《AJAX: A New Approach to Web Applications》文 章中，提出了建立在XMLHTTP组件之上，将JavaScript, XML的几种技术以新的 方式组合而成的"异步交互网络开发技术(AJAX： Asynchronous JavaScript a nd XML) ”的定义与概念。

AJAX 是 Asynchronous JavaScript and XML 的缩写，Asynchronous 是"异 步”的意思。它是如何构成的呢？ AJAX的提出者Jesse James Garrett是这样 阐述的：

1. 使用XHTML和CSS标准化显示
2. 使用DOM (Document Object Model)动态显示和交互
3. 使用XML和XSLT进行数据交互和处理
4. 使用XMLHttpRequest进行异步数据获取
5. 使用JavaScript整合所有这一切(这部分被称为AJAX Engine)

所以说AJAX并不是一门全新的技术，它所包含的技术可能5年前甚至更早 就出现了，它只是才被人们重视，并适时的出现于Web2. 0应用中。

从技术上分析，传统的Web应用模式是通过表单形式完成的。当用户浏览网 页时，填写并提交访问表单，向Web服务器发送请求，服务器接收处理并返回一 个新的网页。该做法浪费了许多带宽；同时由于每次应用的交互都需要向服务器 发送请求，导致用户界面的响应远远慢于本地应用的需要。而AJAX技术的特点是： 用户操作与服务器响应的异步化。用户访问以这种技术建设的网站时，通过异步 模式向服务器发送并取回需要的数据，减少了服务器和浏览器之间交换的数据 量，也减少了Web服务器的处理时间；同时由于该技术采用JavaScript处理来自 服务器的响应，异步交互网络引擎运行在客户端，可以使很多的处理工作在发出 请求的客户机完成，提高了应用响应的速度，降低了服务器的负载。

3. 2 AJAX网络爬虫系统总体设计方案

AJAX框架网络爬虫系统主要包括：网页抓取模块、网页分析模块、程序切 片模块、脚本执行模块、DOM支持模块和页面生成模块。其工作原理图如图3.1 所示。

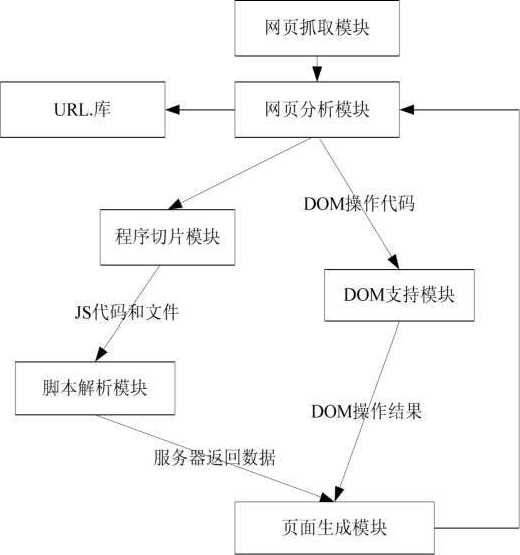


图3.1 AJAX框架爬虫系统工作原理

网页抓取模块首先对初始的URL相应网页进行下载，抓取页面代码。网页分 析模块是对抓取到的页面代码进行分析，判断出是否存在JavaScript代码。程 序切片模块对JavaScript代码部分进行处理，切分出与URL相关的切片。脚本 执行模块是来执行相应的切片程序，获取URL,交由网络爬虫模块进一步爬取。 DOM支持模块是对网页分析结果进行建树，支持类似浏览器中的DOM操作。页面 生成模块是将上述结果重新组合，生成页面内容，交由网页分析模块进行抓取。 依次下去，达到对整个页面URL信息的抽取，从而实现对AJAX框架网站链接抓 取的目的。

3. 3 AJAX网络爬虫的技术难点及解决方案

由于AJAX釆用了异步交互网络应用方式，符合Web2. 0时代的需要，在很大 程度上提高了用户体验。AJAX己经成为开发以用户为中心的Web应用程序的实 事标准，是用户体验的一次革命性进步。AJAX所釆用的技术是标准支持的，无 需下载插件和程序。因此，一经提出就引起了互联网领域的广泛关注。2004年 以来，Google在它的应用程序中成功地使用了 AJAX技术，如Google讨论组、G oogle地图、Gmail等。同时由于该技术支持Mozilla/Gecko,而被各大网站相 继釆用。各种异步交互网络框架如GWT, Atlas, Dojo也应运而生。

但随之而来的却是新的问题：由于AJAX框架网站是构建在异步JavaScript 基础上的Web应用技术，而传统Web搜索引擎中网络爬虫(Web Crawler)并不能 解决异步交互网络中的Web地址(URL)提取［曾伟辉等，2008］。一方面是大量 基于AJAX的网站正在不断的涌现，另一方面这些网站的URL被搜索引擎忽略［曾 伟辉等，2008］。这意味着越来越多有意义的数据将无法通过搜索引擎检索。这 一问题，也引起了国内外学者的广泛关注，研究异步交互网络的地址解析方法对 互联网领域的技术理论发展和应用有重要意义［郑冬冬等，2005：孙彬等，2007］。

由于AJAX是建立在JavaScript基础之上的，而JavaScript是一•种基于对 象(Object)和事件驱动(Event Driven)的语言。它所具有的弱变量、事件驱动执 行、异步回应请求的特点，使得目前网络爬虫技术在解决异步交互网络的Web地 址解析方面，还存在着如下的困难［Manuel Ivarez等，2001］：

1、 在解析异步交互网络地址(URL)过程中必须考虑JavaScript中事件触 发的顺序，而目前爬虫技术不能够识别事件触发顺序，无法构造出完整的URL。

2、 异步交互网络技术构造的网页中的URL通常带有参数，目前的爬虫技术 在解析URL过程中，不能够识别无效参数组合，从而导致大量无效空地址及网络 黑洞的产生。

3、 JavaScript是弱变量、解释型代码，其错误只有运行时才能被捕获。而

目前爬虫技术缺乏对这种错误捕获和处理能力，因而会造成地址解析过程中的异 常中断。

除上述问题外，异步交互式网络的AJAX框架所具有的如下特点也影响了现 有网络爬虫技术在其中的应用：

1、 AJAX框架有大量的表现层代码，网络爬虫在执行这些代码时将消耗许多 时间，而这些代码对提取URL没有任何作用，严重降低了爬虫的执行效率。

2、 AJAX框架层出不穷，据不完全统计，当前AJAX框架有200余种而且这 一数字随着时间的推移还会提高，这为设计通用于各种AJAX柩架的爬虫带来了 很大的困难。

因此，开展支持AJAX框架网站网络爬虫的研究，对解决AJAX站点地址（U RL）的获取，检索隐藏在其中的大量数据，具有重要的理论意义与实用价值。

在近年来的工作中本人发现：利用程序切片［李必信，2006；王伟，2000： H arman M, 1995, 2001］在分解程序、捕获与程序某个变量在所有计算位置相关的 程序信息等方面的特性，有利于解决网络爬虫在解析JavaScript时找出URL变 量有影响的语句、控制谓词以及事件触发函数，降低代码执行的时间和空间复杂 度等问题，这主要表现在如下三个方面：

1、 程序切片技术能够自动剔除与目标变量不相关的函数和代码片断，可以 动态的分析解释型脚本语言执行过程中可能出现的错误，降低代码执行引擎所消 耗的时间。

2、 程序切片技术能够找出和目标变量相关的起始函数（事件），为有限状 态机的执行提供了入口点（否则将不得不穷举所有可能事件），避免爬虫陷入网 络黑洞。

3、 各个AJAX框架的主要区别集中在表现层的设计以及客户端与服务器的通 信协议（通信协议这一部分可以由代码执行引擎自动完成，无须爬虫干预），通 过切片技术剔除表现层代码后，网络爬虫兼容各个AJAX框架将变得更加容易。

然而，利用程序切片技术解决异步交互网络中的地址解析，还面临着如下的 问题［7］：

1、 目前的程序切片技术主要集中在处理面向过程和面向对象的语言，而AJ AX站点采用的是基于对象的语言（JavaScript）,因此需要对基于对象语言的 程序切片方法开展相关研究。

2、 在解析AJAX网页时须考虑到事件触发的顺序，而传统爬虫技术不能够识 别事件触发顺序，无法构造出完整的URL,因此必须对JavaScript脚本的执行 算法开展相关研究。

3. 4本章小结

本章首先介绍了 AJAX网络爬虫系统涉及到的相关概念，接着介绍了支持AJ AX网络爬虫系统的总体设计方案，并详细介绍了 AJAX网络爬虫的技术难点及解 决方案。

第4章AJAX网络爬虫系统实现

4.1相关技术

本文的AJAX网络爬虫系统实现中涉及到很多标准技术，这些标准技术是本 文的工作基础。本节将一一介绍这些技术。

1. HTTP 协议

HTTP(超文本传输协议)是为分布式和合作式超媒体信息系统设计的协议，是 —种无状态、无连接、面向对象的协议，具有简单、灵活等特点，能满足简单数 据传输的需要，是WWW的标准通信协议.网络爬虫在万维网上的信息采集都是通 过HTTP协议实现的，下面介绍实现网络爬虫中需要用到的HTTP主要方法和信息 头格式[吕秀娟,2002]

一、HTTP的信息头

HTTP协议利用信息头来描述每次事务的处理，主要的信息头和信息域的定 义如下：

1. 请求头(RequestHeaders)

User-Agent：客户名及版本号：

From:客户端使用者信息；

Accept:客户端可接受的数据类型；

Accept-Charset :客户端可接受的字符集；

Accept-Language:客户端能理解的语言。

1. 应答头(ResponseHeaders)

HTTP-Version:服务器支持的HTTP版本号；

Status-code：服务器处理本次请求的状态码，常见的有：

200正常

400请求错误

404请求的资源未找到

500服务器内部错误

301请求的资源己经移动

302请求的资源被临时移动

Server： Web服务器软件名和版本号；

Last-Modified：客户端请求资源的最后修改时间；

Content-Length：本次传输数据的长度；

Content-Type：本次传送数据的类型，由服务器根据文件扩展名决定；

Content-Transrer-Encoding：表示服务器采用的编码机制； Content-Language：标识本次传送数据釆用的语言种类。

二、HTTP的主要方法

HTTP方法(Methd。)描述了在指定资源上将要执行的动作，HTTP协议中定义 了多种方法，其中最常用的如下：

1. GET：负责取回指定资源
2. HEAD:请求服务器传送指定资源的大小、最后更改时间等描述性信息
3. POST:用于客户端向服务器发送数据，主要是发送HTTP F0MR内容，供 服务器端处理。
4. HTML (Hyper Text Markup Language)

HTML是一个用于创建网页的标记语言，它是一个由W3C组织创建并且维护 的Internet标准，通常表现为连接到互联网的计算机中的文本文件。这些文件 中包含很多标记，这些标记是告诉浏览器如何表现或处理文本内容的指令。

HTML在展示数据方面十分成功，但是对于机器来说，一个HTML文档是很难 理解的。因为IITML着重于数据的表现而非数据的描述。比如<b><i>这样的标记 根本无法获得它们所包含文本究竟是什么内容，着重于数据描述的是另外一个语 言 XML (周津,2004) O

1. XML (EXtens i bIe Markup Language)

XML的全名是可扩展标记语言(Extensible Markup Language),它允许开发 人员制定自己的标记，从而使得文档具备自描述性。

XML有两个先驱——SGML和HTMLo这两个语言都是非常成功的标记语言， 但是它们都在某些方面存在着与生俱来的缺陷。SGML (Standard Generalized Markup Language)的全称是标准通用标记语言，它为语法标记提供了异常强大的 工具，同时具有极好的扩展性，因此在分类和索引数据中非常有用。但是，SGM L非常复杂，并且价格昂贵。几个主要的浏览器厂商都明确拒绝支持SGML,使S GML在网上传播遇到了很大障碍。相反，超文本标记语言HTML (HyperText Mark up Language)免费、简单，在世界范围内得到了广泛的应用。它侧重于主页表现 形式的描述，大大丰富了主页的视觉、听觉效果，为推动WWW的蓬勃发展、推 动信息和知识的网上交流发挥了不可取代的作用。可是，HTML也有如下几个致 命的弱点，这些弱点逐渐成为HTML继续发展应用的障碍(周津，2004)。

1、HTML是专门为描述主页的表现形式而设计的，它疏于对信息语义及其内 部结构的描述，不能适应日益增多的信息检索要求和存档要求。

2、 HTML对表现形式的描述能力实际上也还非常不够，它无法描述矢量图形、 科技符号和一些其他的特殊显示效果。

3、 HTML标记集变得日益臃肿，而其松散的语法要求使得文档结构混乱而缺 乏条理，导致浏览器的设计越来越复杂，降低了浏览的时间效率与空间效率。

正因为如此，1996年人们开始致力于描述一个标记语言，它既具有SGML的 强大功能和可扩展性，同时又具有HTML的简单性。XML就是这样诞生的。正像 SGML和HTML -样，可扩展标记语言XML也是一种标记语言，它通过在数据中加 入附加信息的方式来描述结构化数据。不过，XML并非像HTML那样，只提供一 组事先已经定义好的标记，准确地说，它是一种元标记语言，允许程序开发人员 根据它所提供的规则，制定各种各样的标记语言。

XML的语法很严格。形式良好(well formed)限制是XML文档的最低要求。 不满足形式良好限制的文档就不是XML文档，解析器将无法读取这种文档。简 单的说，XML文档应有且只有一个根元素，所有开始标记要有相应的结束标记， 元素要正确的嵌套，所有属性值要放到引号中等。除了把XML文档看成符合一 定规则的字符序列外，还可以把XML文档看作一棵树(tree),有一个根节点，包 含不同的子节点，其中有些子节点自己又有子节点，而有些则是叶节点，没有子 节点。XML树中大致可以分出5种不同节点：

1、 根节点

也称为文档节点，是一个抽象节点。包含整个XML文档，其子节点包括注 释语句、处理指令和文档的根元素。

2、 元素

XML元素，具有名称、一组属性、一组范围名称空间和一列子节点。

3、 文本节点

两个标记(或者任何其它非文本节点)之间的字符数据。

4、 注释语句

<!-注释语句一>之类的XML注释语句。注释语句的内容是其数据。注释语句 没有任何子节点。

5、 处理指令

<?xml-stylesheet type="text/css" href="order. css"?>之类的处理指 令。处理指令具有目标和值，没有任何子节点。

一个XML文档的大部分内容都是元素和文本节点。文本节点是文档的数据部 分，元素节点则是XML文档的结构部分。一般来说应用程序主要就是处理元素 节点和文本节点。下面将详细介绍元素。

XML的基本单元是元素。从语法上来说，一个起始标记、对应的结束标记以 及标记之间的内容就是一个元素，例如＜Note＞……＜/Note＞;从逻辑上来说，每 个元素有四个关键部分：名称、元素属性、元素名称空间、元素内容。名称就是 标记的名字，如Note,元素属性是标记中所含的名称/值对，名称空间不属于本 文的考虑范围，元素内容是标记之间的数据，可以包含文本或者一个或多个子元 素。这样，这些元素就构成了一个树的结构。实际上，XML文档从逻辑上就是一 个树模型。正因为如此，对XML文档进行转换以及提取数据都变得很容易。

XML元素可以在开始标记中包含属性，就像HTML -样。属性提供了元素的 附加信息。如〈person sex="female"〉 〈/person〉中，person 元素具有一•个 属性sex,对应的值为female。一个元素可以包含多个属性，这些属性是没有顺 序的。(于静，2008)

除了形式良好外，XML还可以加上有效性限制。合法的数据不一定有效，比 如身高3米。可以使用模式语言(schema language)来对XML文档进行验证。主 要的模式语言有 DTD (Document Type Definition)和模式(Schema) o DTD 可以 定义文档包含什么元素、每个元素包含什么、按什么顺序以及每个元素有什么属 性；W3C XML模式(XML SCHEMA)语言克服了 DTD的几个局限。首先，模式用X ML实例文档来编写、使用标记、元素和属性；第二，模式完全支持名称空间； 第三，模式可以指定元素的数据类型，验证文档时不仅可以根据元素结构，而且 可以根据元素内容。

XML关于文档浏览的基本思想是将数据与数据的显示分别定义，XML文档本 身不涉及各种数据的具体显示方式，文档的显示实际上是通过一个外部样式表， 又称为样式单来描述的。内容与形式相分离使XML文档更偏重于数据本身，而不 受显示方式的细枝末节的影响，而旦通过定义不同的样式单可以使相同的数据呈 现出不同的显示外观，实现XML数据的可重用性。样式单(StyleSheet)是一种 描述结构文档表现方式的文档，具有表达效果丰富、文档体积小、便于信息检索、 可读性好等优点。迄今为止W3C己经给出了两种样式单语言的推荐标准，一种是 层叠样式单CSS (Cascading StyleSheets),另一■种是可扩展样式单语言XSL(e Xtensible Stylesheet Language)» CSS主要用于文档的显示，XSL功能很强大, 它包含两个互为补充的组件：XSL转换(XSLT Transformation)与XSL格式化对 象(XSL-FO, XSL Formatting Object)o

4. 1.4 XHTML

HTML是一种标记语言，有一套预定义的标记。但是HTML并不符合严格XML 的语法，而且由于各浏览器厂商的竞争，很多语法错误的HTML文档也都能被浏 览器显示出来，这使得大量的HTML文档根本无法直接用于解析。为了改变这种 状态，W3C推出了 HTML的替代标准XHTMLo与HTML不同的是,XHTML严格建立 在XML基础之上，并且明确定义了合式的文档规则。这意味着，可以像对待一般 XML文档一样对待XHTML文档，这样我们就可以利用各种强大的XML标准技术来 操纵XHTML文档，从而可以大大简化应用程序的开发和维护。

4.1.5 DOM (Document Object Model)

DOM是HTML和XML文档的一个应用程序接口(API),是W3C制定的标准 接口规范。它定义了 HTML文档和XML文档的逻辑结构，给出了一种访问和处理 HTML文档和XML文档的方法(张秀清，2006)。DOM接口提供了一种通过分层对 象模型来访问XML文档信息的方式，这些对象模型依据XML的文档结构，即元素 间有序的嵌套结构，形成一棵对象节点树，本文中称此结构为DOM树，它将XML 文档表示为节点构成的树。org.w3c.dom包中的不同接口可以表示元素、属性、 已分析的字符数据、注释和处理指令，它们都是公用Node接口的子接口。Node 接口提供了在树中导航与处理的基本方法。

4. 1. 6 XPath

XPath是第四代声明式语言，用于定位XML文档中的节点。XPath位置路径 指定需要文档中的哪些节点，而不指定寻找这些节点的算法。只要向方法传递一 个XPath语句，XPath引擎就会负责确定如何寻找满足这个表达式的所有节点。 这比自己用DOM, SAX与JDOM编写详细的搜索与导航代码健壮得多，即使文挡 格式不符合预期，XPath也通常能搜索成功，例如，文本段落中间的说明可能会 破坏连续文本的DOM代码，但这不会影响XPatho许多XPath表达式还可以对付 更大的变化，如改变祖先元素的名称或名字空间，重排元素的子节点，以及在树 层次中增加或减少整个层级。(于静，2008)

大致来说，在Java程序中使用XPath与在Java程序中使用SQL差不多。要 从数据库中取得信息，就要编写SQL语句表示需要什么信息，并且JDBC会取得 所要的信息。你不知道也不关心JDBC如何与数据库通信。同样，在XML中，只 要编写一个XPath表达式来表示需要什么信息，XPath引擎会到XML文档中取得 该信息，你不知道也不关心XPath搜索XML文档的算法。

定位路径(Location Step)是XPath的基本结构，定位路径选择XML文档 中的一组节点。每个定位路径包括一个或几个定位步(location step),每个定 位步包括三个部分：一个轴(axis)、一个节点测试(nodetest)和一个或几个谓 词(predicate)。轴与节点测试用双冒号(：：)分开，每个谓词放在方括号中，即 axisname::nodetest[predicate]0

1、轴：轴定义了要定位的节点相对于上下文的方向和范围。如：

1. Self：节点本身
2. Child:上下文节点的所有子节点
3. Parent：父节点
4. Attribute：上下文节点的属性

2、 节点测试：允许从指定的轴中选择特定的元素或者其他节点类型，来进 一步缩小节点定位范围。如下：

1. Name：具有指定名称的任何元素或属性
2. \* :沿着属性轴，星号匹配所有属性节点，沿着名称空间轴，星 号匹配所有名称空间节点，沿着其他轴，星号匹配所有元素节点。文本节点不属 于元素节点。
3. Test :任何文本节点
4. Node：任何节点，包括文本节点或元素节点或其他任何节点

3、 谓词：谓词是方括号中的XPath表达式，对所选择的每个节点求值。如 果谓词为true,则节点保留在节点集中；如果谓词为false,则节点不保留在 节点集中。

4. 1. 7 Web信息抽取、Web信息检索、Web挖掘

4. 1.7.1信息抽取与信息检索

信息抽取与信息检索存在差异，主要表现在三个方面：(李保利等，2003)

1. 功能不同。信息检索系统主要是从大量的文档集合中找到与用户需求相 关的文档列表；而信息抽取系统则旨在从文本中直接获得用户感兴趣的事实信 息。
2. 处理技术不同。信息检索系统通常利用统计及关键词匹配等技术，把文 本看成词的集合(bags of words),不需要对文本进行深入分析理解；而信息 抽取往往要借助自然语言处理技术，通过对文本中的句子以及篇章进行分析处理 后才能完成。
3. 适用领域不同。由于采用的技术不同，信息检索系统通常是领域无关的, 而信息抽取系统则是领域相关的，只能抽取系统预先设定好的有限种类的事实信 息。

另一方面，信息检索与信息抽取又是互补的。为了处理海量文本，信息抽取 系统通常以信息检索系统(如文本过滤)的输出作为输入；而信息抽取技术又可 以用来提高信息检索系统的性能。二者的结合能够更好地服务于用户的信息处理 需求。

4. 1.7. 2 Web信息抽取与Web挖掘

Web信息抽取利用信息抽取技术，将半结构化数据转化为结构化数据，存储 于传统的数据库中，利用基于数据库的数据挖掘算法提取有用的知识，这是一种 离线式Web数据挖掘方式。

而Web挖掘另一种是在线方式，将Web信息提取技术与数据挖掘算法结合为 一体，直接对提取的数据进行挖掘。由于一次抽取的信息样本一般较小，甚至数 据是不完整的，而且信息是增量式的，需要挖掘算法具有处理这些特性的能力， 如增量式决策树方法等。Web数据挖掘技术首要解决半结构化数据源模型和半结 构化数据模型的查询与集成问题。

信息抽取是文本挖掘当中最重要的模块，实际上，在许多文章中都将这两者 看作是同样的概念，当然实际上他们并不是对等的。信息抽取的目的是从文本中 扫描并抽取所需要的事实，Grishm (1997)给信息抽取技术下了一个清晰的定义： 信息抽取系统应该是由应用领域决定，而且在进行训练之后生成的-个项集。B asil (1997)在他的论文中，强调了词典在信息抽取和自然语言处理中的作用。 在信息抽取中，词典用来发现事实以及它们之间的相互关系.为了进行信息抽取, 有许多种不同的技术来获得特定领域内的词典。

1. 2网页抓取模块

4.2.1网页抓取模块流程

网页抓取模块主要是通过URL,发起HTTP请求，从而获取相应的网页。根 据返回的状态码判定当前URL是否有错，若有错，则返回错误信息。如果无法连 接或者连接超时，根据设定的重连次数来确定是否重新连接。最后返回得到的页 面或者错误信息。

4. 2. 2网页抓取模块的相关实现

4.2. 2. 1网络编程基本概念及两种传输协议简介

1. 网络基础知识

网络编程的目的就是指直接或间接地通过网络协议与其他计算机进行通讯。 网络编程中有两个主要的问题，一个是如何准确的定位网络上一台或多台主机， 另一个就是找到主机后如何可靠高效的进行数据传输。在TCP/IP协议中IP层主 要负责网络主机的定位，数据传输的路由，由IP地址可以唯一地确定Internet 上的一台主机。而TCP层则提供面向应用的可靠的或非可靠的数据传输机制，这 是网络编程的主要对象，一般不需要关心IP层是如何处理数据的。

目前较为流行的网络编程模型是客户机/服务器(C/S)结构，即通信双方的一 方作为服务器等待客户提出请求并予以响应。客户则在需要服务时向服务器提出 申请。服务器一般作为守护进程始终运行，监听网络端口，一旦有客户请求，就 会启动一个服务进程来响应该客户，同时自己继续监听服务端口，使后来的客户 也能及时得到服务。

1. 两种传输协议：TCP和UDP

虽然TCP/IP协议的名称中只有TCP这个协议名，但是在TCP/IP的传输层 存在TCP和UDP两个协议。

1. TCP是Transfer Control Protocol的简称,是一种面向连接的通信协 议。其目的是提供大量数据的传输并确保其传输无误，具有错误侦测、数据复原 及数据重送等功能。发送方和接收方的两个socket之间建立连接，以便在TCP 协议的基础上进行通信，当一个socket (通常都是server socket)等待建立连接 时，另一个socket可以要求进行连接，一旦这两个socket连接起来，它们就可 以进行双向数据传输，双方都可以进行发送或接收操作.
2. UDP是User Datagram Protocol的简称，是一种无连接的通信协议。 其目的是在于处理传输少量的数据。
3. 两种协议的比较：

UDP与TCP不同的是：UDP在传输数据之前不需要建立通信链接，仅须设置 计算机间的IP及使用相同的端口号，即可互相传输信息。因此UDP只提供单向 的数据传输，这样可以节省建立连接所需的时间，适用于在主机间做单向的数据 传输，但是UDP不提供数据错误的侦测以及数据重送等功能，因此不能确保数据 完整发送。而TCP因为要建立连接而对系统资源的要求比UDP多。

4. 2. 2. 2 Java网络编程

1. 统一资源定位器URL

URL表示Internet上某一资源的地址。通过URL可以访问Internet上的 各种网络资源，比如最常见的WWW, FTP站点。浏览器通过解析给定的URL可以 在网络上查找相应的文件或其他资源。

1. URL的组成

格式如下：〈通信协议〉://〈资源名〉

其中，所使用的通信协议如http、https, FTP、gopher或mail等，资源名是资 源的完整地址，包括主机地址、端口号、文件名或文件内部的引用等。例如： http://www. google, com/是协议名：//主机名

http://www. baidu. com/index. html 是协议名：//机器名 + 文件名

1. 创建URL

Java中专门在java, net中实现了类URL,我们可以通过四种方式初始化 一个URL对象：

public URL (String spec):

public URL (ULR context, String spec);

public URL (String protocol, String host, String file);

public URL (String protocol, String host, int port, String file);

1. 解析URL

一个URL对象生成后，其属性是不能被改变的，可以通过类URL所提供的方 法来获取这些属性：

方法说明

|  |  |
| --- | --- |
| getProlocol () getHost () getPort () getFile () getQuery () getPath () | 获取该URL的协议名  获取该URL的主机名  获取该URL的端口号，如果没有设置端口，返回一 1  获取该URL的文件名  获取该URL的査询信息  获取该URL的路径 |

gelAtlthorltyO获取该URL的权限信息

|  |  |
| --- | --- |
| getUserlnfo ()  getRef () | 获得用户信息 获得该URL的锚 |

1. 从URL读取Web资源

使用URL的方法openStream ()通过URL对象读取指定的WEB资源，方法Op enStream ()与指定的URL建立连接并返回InputStream类的对象以便从这一连接 中读取数据。

1. 通过 URLConnetction 连接 Web

通过URL的方法openStream()只能从网络上读取数据，如果同时需要输出 数，必须先与URL建立连接，然后才能对其进行读写，需要用到URLConnection 类。

在包java, net中同样定义类URLConnection,它表示Java程序和URL在网 络上的通信连接。当与一个URL建立连接时，首先要在一个URL对象上通过方 法openConnection ()生成对应的URLConnection对象.例如下面的程序段首先生 成一个指向地址<http://blog>. 163. com的对象,然后用openConnection()打开 该URL对象上的一个连接，返回一个URLConnection对象。如果连接过程失败， 将产生lOExcention异常。

Try (

URL net=new URL(' <http://blog>. 163. com):

URL Connection cnt=net. openConnection():

} catch ( MalformedURLExceptione) (〃创建 URL ()对象失败

}catch (lOExceptione) (//openConnection ()失败

4.3网页分析模块

4. 3. 1网页中嵌入JavaScr ipt的方法

―将 JavaScript 代码写在〈script type="text/javascript"〉和〈/script >之间，如：

<script type="text/javascriptz，>

//JavaScript 代码

</script>

这种是目前嵌入JavaScript的一■般方式。JavaScript代码可以在head区或b ody区，但有些区别。JavaScript代码随网页的其他元素顺序载入，载入后先经 过一次语法检查，然后顺序执行；如果JavaScript代码中不包含document, wr ite或document, writein语句，写在head区比较好，因为这样会先执行。

二 将JavaScript代码写在<script>标签中的src属性中，链接外部脚本， 这样可以实现代码与页面分离，如：

〈script src ="xxx. js">〈/script>

<script type="text/javascript" src ="xxx.js"〉〈/script>

三将JavaScript代码直接写在网页标签所支持的事件中，如：可以写在标 签A的href和标签FORM的action属性中。

4.3.2网页分析模块流程

本模块根据网页抓取模块获取的网页，遍历该网页的每个节点，通过上述对 JavaScript嵌入方式的分析来判断，如果是script节点，并且是JavaScript 类型，则分析是文件还是代码；若是文件，则通过HTTP请求获取此文件;若是

代码，直接将这些代码保存为本地文件。最后将所有JavaScript文件交给程序 切片模块来处理。流程如图4.1所示。

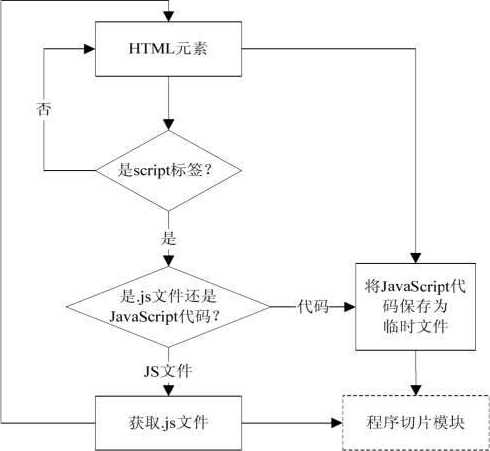


图4.1网页抓取流程图

4. 4程序切片模块

4.4.1程序切片流程

对JavaScript程序进行语法和语义分析，从而构建程序层次模型，构造对 象层次子图、控制依赖与数据依赖子图，从而构建基于对象的系统依赖图。再次 利用依据网页分析模块构建的切片准则来逆向遍历基于对象的系统依赖图，最后 计算出与URL相关的切片，交由脚本执行模块执行。流程如图4. 2所示。

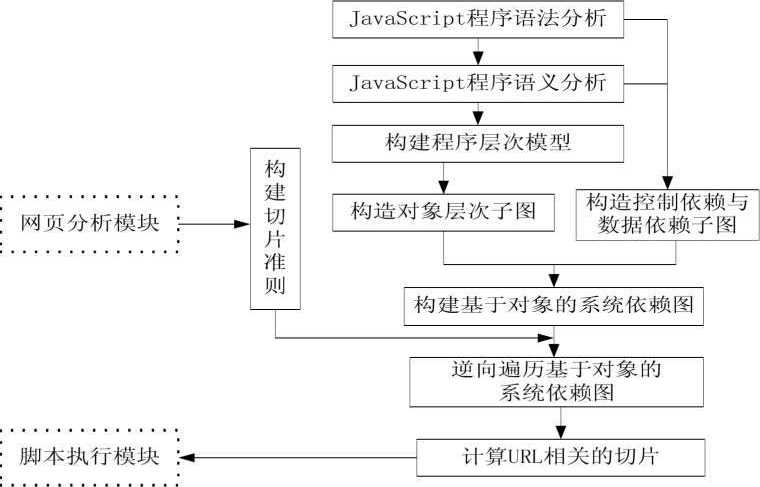


图4.2程序切片流程图

4. 4. 2关键算法

根据JavaScript自身的特点建立JavaScript语言的分层切片模型，釆用对 传统的系统依赖图进行基于对象扩充的方案进行JavaScript脚本切片，以获得 URL相关的程序片段。将基于对象的程序切片计算分成三部分：

一构建出JavaScript程序的依赖关系

通过对JavaScript源程序进行基于扫描的语法分析，找出所有的对象、变 量、方法的定义；按照逻辑结构将JavaScript分为对象层、方法层和语句层， 逐层抽取程序语句间的数据依赖和控制依赖关系。通过对JavaScript源程序中 赋值语句的左值右值，控制语句的谓词，函数调用语句对实参和全局变量的影响 以及对象继承时的多态进行语义分析；针对JavaScript中动态定义对象的模式, 实现对象中数据、方法的统一封装处理。

二基于对象的系统依赖图构造方法研究

根据数据依赖和控制依赖关系构造出由对象层次子图（OHG）、控制依赖子 图（CDG）、数据依赖子图（DDG）三个基本层次组成的基于对象的系统依赖图 （BOSDG） „ OHG描述了基本对象的结构信息和对象层次信息，其中的顶点包括 每个对象的对象首部顶点、定义在每个对象中的每个方法的方法首部顶点；边包 括每个对象得对象首部顶点到与其有继承关系的对象的相应对象首部顶点的继 承边，由方法首部表示的方法节点到定义该方法的对象的对象首部顶点的类成员 边。当一个对象和另一个对象或者系统结合时，通过对象首部节点和对象成员边 就能够方便的访问方法的信息。同时图中子对象没有重新表示从超对象中继承的 方法，因此消除了对继承方法的重复表示。CDG中描述了函数方法的具体语句实 现过程，采用静态后向切片的方法，包含了方法的多态性表示等。DDG中包括了 对象的实现，消息动态绑定到对象中的特定方法表示，对象间的数据依赖关系等。

三URL相关的程序片段切分与计算方法

利用两阶段图形可达性算法逆向遍历BOSDG （基于对象的系统依赖图）。首 先（1）在BOSDG上找出从节点n出发，沿（正向或逆向）数据依赖边或者控制依赖 边可以到达的节点进行标记，构成程序关于节点n（语句n）的程序切片。（2）标记 在BOSDG中与n相连的节点，然后标记跟这些节点相连的节点，依次计算到不能 找到新的节点为止。通过上述遍历过程中的节点标记，计算出URL相关的程序片 段。

4.5脚本执行模块

脚本执行模块是对程序切片模块计算出来的切片，利用状态控制引擎，对其 进行事件监听，进一步判断状态表中的事件、条件、动作是否满足条件，从而决 定是否执行状态表中的操作。执行操作由脚本执行引擎来执行，返回下一个状态, 判断是否结束状态，当结束时交由网页分析模块进行进一步的分析；当没有结束 时转到判断执行阶段进行再一次的判断执行。流程如图4. 3所示。

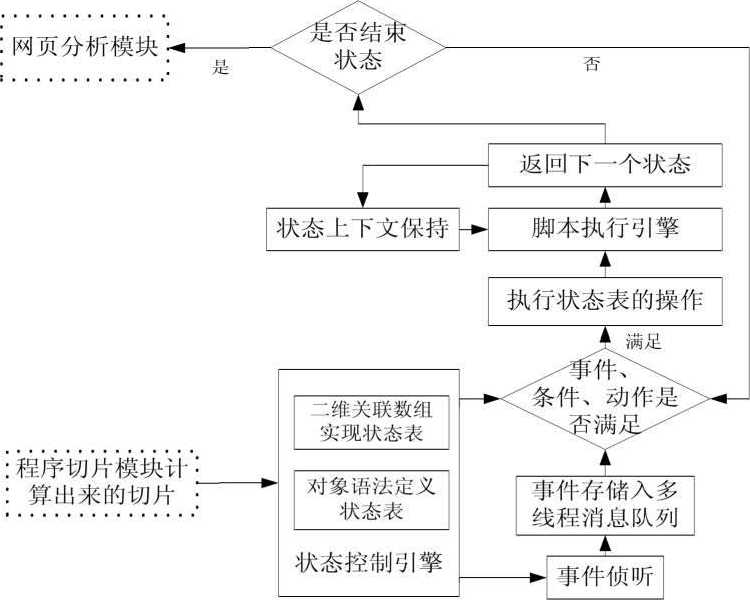


图4.3脚本执行流程图

4.6程序切片模块、爬虫模块、脚本执行模块之间的互操作

通过研究三种模块之间的互操作协议，来实现三者之间的数据传递，状态控 制等。同时保持了模块最大限度的独立，且能互相协作，提高了模块的高内聚性 和模块间的低耦合性，降低了程序的复杂度。

4.6. 1构建JavaScript切片的有限状态机

通过切片后得到的JavaScript代码来构造有限状态机的状态表，以表示所 有情况下响应每个事件所需的操作以及状态变量，为状态间的事件和转移组织动 作。将JavaScript函数存储于关联数组中，将状态表实现为一个二维的函数关 联数组，使用状态名称和事件名称作为索引。利用JavaScript中关联数组是对 象的特性，使用对象语法定义状态表，将状态表直接转换成代码。

4.6.2程序切片模块、脚本执行模块之间的互操作

启动事件侦听端口，用多线程消息队列存储事件，对于状态表中事件条件动 作、事件动作、事件条件、条件动作、条件、动作的转换，程序切片模块判断转 换条件、动作，条件满足时，脚本执行模块执行状态表中的操作，返回下一个状 态，并保留状态上下文。动态跟踪程序执行过程，截获所需的中间结果以及最终 结果。

4.6.3程序切片模块、爬虫模块、脚本执行模块之间的互操作

动态跟踪程序执行过程，将最终结果返冋给爬虫模块处理。爬虫模块继续利 用URL参数构建切片准则，调用程序切片模块得到该参数的代码片断，返回给脚 本执行模块，反复执行上述操作，直至URL完整构造出来。

其中JavaScript脚本的解释处理大体可分为五大模块：JavaScript编译模 块，JavaScript字节码解释执行模块，atom管理模块，垃圾收集模块以及标准 类管理模块。其体系结构和工作原理可参考NetScape公司开发的JavaScript 解释器引擎SpiderMonkey。

4.7 DOM支持模块

由于只有格式良好的(We 11-formed)的XML文档才能转换为D0M树，所以本 文首先使用开源nekohtml. jar将样本网页的HTML格式清理转换成XML文档。 HTML的标记间具有一定的嵌套结构，可以将其生成一颗文档对象模型Document Object Model (DOM)树，DOM树是一个对象化的HTML数据接口，一个与语言无关， 与平台无关的标准接口规范，它定义了 HTML文档的逻辑结构，给出了一种访问 和处理HTML文档的方法。DOM树结构简单清晰，意义表述明确，成为描述和操

作HTML文档的最流行的方式之一。它展现HTML层次化的文档结构，将I1TML语 言里面的标记（TAG）作为DOM树的节点，形成一种层次化的DOM树。（于静， 2008）

图4.4表示的是一棵简单、标准的D0M树。树中的每一个节点对应于HTML语 法里的TAG元素。图中的DOM树是图4. 1网页对应的HTML-DOM树的一个部分， 该DOM树对网页的DOM树进行了剪枝过滤和清洗，删除原有的部分与URL信息 无关的节点和对显示效果起修饰性作用的节点。由图可以看出，对于各个子信息, 有共同的祖先节点，该节点以下包含所有的子节点共同构成了所要抓取的链接信 息块。该信息块是一棵DOM子树。

html

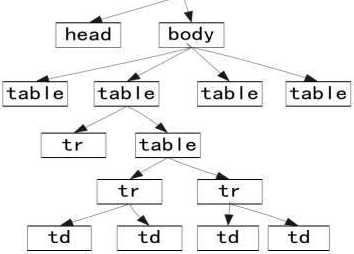


图4.4 -•个简单的网页DOM结构

由于JavaSript代码的作用，这棵DOM树会随着代码的执行而发生变化，从 而造成对AJAX框架网站链接爬取的困难。我们需要在对JavaScript解析之后， 重建D0M树。但是这种重建，会消耗大量的时间，所以在对页面分析，脚本解析 之后，经过程序切片的计算，我们只重建那些与URL信息相关的DOM树，这样相 对会节省一些时间。

4.8页面生成模块

经过上述模块的配合，对页面元素进行分析和处理之后，可以直接通过页面 生成模块来生成新页面，从而进行再次的链接抓取。在页面中抓取链接，主要是 通过＜A＞标签的href属性来提取。页面链接抓取的流程如图4. 5所示。

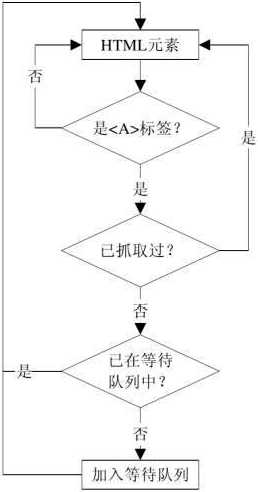


图4.5页面链接抓取流程图

4. 9本章小结

本章主要介绍了 AJAX框架网络爬虫的具体实现。首先介绍了实现网络爬虫 系统的相关技术，如HTTP协议，HTML语言等，接着介绍了 AJAX框架网络爬虫 系统各模块实现的关键算法、流程图以及涉及到的相关技术。

第5章AJAX网络爬虫系统测试

5.1软件测试

随着软件规模的不断扩大，软件设计的复杂程度不断提高，软件开发中出现 错误或缺陷的机会越来越多。同时，市场对软件质量重要性的认识逐渐增强，软 件测试在软件项目实施过程中的重要性日益突出。

软件测试就是在软件投入运行前，对软件需求分析、设计规格说明和编码的 最终复审，是软件开发过程的重要组成部分，是软件质量保证的关键步骤。它一 个系列过程活动，在软件项目的每一个阶段都要进行不同目的和内容的测试活 动，以保证各个阶段的正确性。软件测试的对象不仅仅是软件代码，还包括软件 需求文档和设计文档。测试的目标：以较少的用例、时间和人力找出软件中潜在 的各种错误和缺陷，以确保系统的质量。

软件测试从是否关心软件内部结构和具体实现的角度可分为白盒测试、黑盒 测试和灰盒测试。白盒测试也称结构测试或逻辑驱动测试，它是按照程序内部的 结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正 常进行，检验程序中的每条通路是否都能按预定要求正确工作。黑盒测试也称功 能测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看 作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在 程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使 用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程 序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。灰 盒测试，确实是介于白盒测试与黑盒测试之间的。可以这样理解，灰盒测试关注 输出对于输入的正确性，同时也关注内部表现，但不象白盒那样详细、完整，只 是通过一些表征性的现象、事件、标志来判断内部的运行状态，有时候输出是正 确的，但内部其实已经错误了，这种情况非常多，如果每次都通过白盒测试来操 作，效率会很低，因此需要釆取这样的一种灰盒的方法。

本文使用JUnit进行单元测试，测试驱动开发(TDD)是以测试作为开发过程 的中心，坚持在编写实际代码之前写好基于产品代码的测试代码。开发过程的目 标就是首先使测试能够通过，然后再优化设计结构。测试驱动开发式是极限编程 的重要组成部分，xUnit是一个基于测试驱动开发的测试框架，为开发过程中使 用测试驱动开发提供了一个方便的工具，从而快速地进行单元测试。xUnit的成 员有很多，包括JUnit, CPPUnit等，AJAX网络爬虫系统采用了川nit,它是一 个专门面向java的测试框架，用于白盒测试，对于AJAX网络爬虫系统各个模块编写测试代码并通过。

5.2测试内容

实验环境:WindowsXP, P4 CPU, 2. 66GHZ, 3. 25G 内存。

5.2.1 AJAX框架站点的测试

使用所实现的支持AJAX框架的网络爬虫系统和传统的网络爬虫分别抓取网 易博客首页的链接，分别记录了抓取的链接总数和总时间。

|  |  |  |
| --- | --- | --- |
| P- AJAXCr»rler«. ln£ -已事本 | |  |
| 丈件**CD** *d)*格式⑵ 査看① | 糖助**Qj)** |  |
| **fromCulture** |  | *4* |
| **2009.04.29 10:39:41.093** | **Add to imiList** | **http：Z/b1og. 163.com/xysa1on/blog/static/52352920094237l622217?** |
| **fromCulture** |  |  |
| **2009.04.29 10:39:41.093** | **Add to URLList** |  |
| **littp^bloy.!63.cx)iiiAvenliu<i|)eiyM)li>g/sl.iti(y618160732()0942261656576?fruinCullure** | | |
| **2009.04.29 10:39:41.093** | **Add to URLList** | **<http://blog.I63.com/xysalon/blog/static/523529200942493546103>?** |
| **fromCulture** |  |  |
| **2009-04-29 10:39:41.093** | **Add to URLList** | **http7/hlog.l63.com/qin.q>7hlog/stati(y2750B74520()942385430260?** |
| **fromCulture** |  |  |
| **2009.04.29 10:39:41.093** | **Add tu URLList** | **hltp^blog.!fi3.conVssnlyl 0(J/l>log/static/l 1563392009427103658317?** |
| **fromCulture** |  |  |
| **2009-04-29 10:39:41.093** | **Add to URLList** |  |
| **http：Z/blog.l63.com/xuefengslii200(J/l)log<Mdti(7665HM5200»42712654558nroinCullure** | | |
| **2009-04-29 10:39:41.093** | **Add tolTRLList** | **http:Z/blog.l63.coni/xuzhiTning668/blog/static/1252526820094249485551?** |
| **fromCulture** |  |  |
| **2009.04.29 10:39:41.093** | **Add to URLList** | **http^/corp. 163.com/index-gb.html** |
| **2009-04-29 10:39:41.125** | **link not this host** | **http**:〃**gb.corp. 163.com/gb/contachis.html** |
| **2009.04.29 10:39:41.125** | **Add to URLList** | **httpv/corp. 163.coni/yi)/iob/i ob.htnil** |
| **2009.04.29 10:39:41.125** | **Add to URLList** | **http:Mielp.163.8m/7bl3abkl** |
| **2009.04.29 10:39:41.125** | **Add to URLList** | **http**:〃**corp. 163.com/gb/legal/legal.html** |
| **2009.04.29 10:39:11.125** | **Add to URLList** | **<http://jz.163.coin>** |
| 支持**AJAX**网绻爬虫抓成链簽总数为**509**个 | |  |
| 屹费时间为**3704ms** |  |  |

支持AJAX的网络爬虫系统对网易博客首页的抓取结果

Crnvlers. I og -记■本 匚|叵|区j

**irttd)** 格£•#吝看領 袖助叩

httpv7blog.163.com/lntooldtraflord/blog/sta H</10235810200942651610143?~ fromSporthttp:㈣ log.163.com/intoo1dtMffonVblog/statid102358102009425358118117 fromS|K>rthtlp://l)log.l63.com/iniooldtrliffor(Vblog/static>'l 023581020094233463145 l?iromSf)ort http:〃 blog. 163.com/intooldtrafford/blog/stamyi 0235810200942414751948\*

fromSporthttp :^blog.l 63.com/intooldtraf ford/blog/staticZl 0235810200942265824292?fromSpor1 liltp:/7bluy.l63.a)ni/intuoldtrdffonVI)lvg/stdti(yi0235810200g42752325303?

fromSporthttpu/blog. 163.com/zhoupengduiyuan/blog/stdtifyi 64041902009425114824593?fromSport httpv>blog.! 63.com/chendaweiduiyuan/blog/st«iti(716396457200042210723906?fromSport <http://blog.l63.com/qushaobinduiYuan/blog/staticyi639B645200942684234661>?

fromSporthttp：Z^blog. 163.com/intooldtrof ford/Tilog/stati(yi 0235810200942752325303?fromSport hltp://blog.i 63.coni/intooldtrdfford/l)log/stati(yi 0235810200912651610143?

fromSporthttp:j7azurestars.blog. 163.com/blog/static/ll 83934222009427105220277/?fromTodaytoutlao http^/kexu('^ngshuhui.hlog.l63.com,/hlog/static,93596567200942693845616/7fromTo(iaytoutlao httpV/blog.l 63.com/pub/famous/all.html

http:^frccpilo.bloq. 163.com/?frompcoplc

httpv>blog.i 63.com/ja vascript; void [0);

http:〃 news. 163.com/special/000132 LCVha ngye.h tm I [http://blog.163.com/gpcat鱼](http://blog.163.com/gpcat%e9%b1%bc) 126/?fromexpert <http://blogJ63.com/gusunj/?tromexper>

侍统网络爬虫抓取懐接总数为**50**个

花费时冋为300ms

传统网络爬虫系统对网易博客首页的抓取结果

5.2.2非AJAX框架站点的测试

使用所实现的支持AJAX框架的网络爬虫系统和传统的网络爬虫分别抓取

http://soft.gougou.com的链接，接，分别记录了抓取的链接总数和总时间。

**E AJAKCrarlersoftgougou. log -**记事本

d84restype=24i(i = 6 2009.04.29 16:08:32.796 fd&restype=2&id=6 2009-04.29 16:08:32.796 70&restype=2&id = 6 2009.04.29 16:08:32.796 2009.04.29 16:08:32.796 d6%>ba%d(U restype=2& Id=6 2009.04-29 16:08:32.796

Add tolHlLList

Add to URLList

Add to URLList

**Add to URLList**

Add to URLList

65%61%64%65%72Arestype=2Aid=6 2009.04.29 16:08:32.796 Add to URLList

a84restype=2&id=6 2009.04.29 16:08:32.796 Add to URLList

cO%c3%e6%bb%ad%bl%a8&restype=2&id=6 2009.04.29 16:08:32.796 a6&restype=2&i(i=6 2009.04.29 16:08:32.796 2009.04.29 16:08:32.796 2009.04.29 16:08:32.796

Add to URLList

Add to URLList

Add to URLList

Add to URLList

支持**AJAX**网络爬虫抓瑕链表忘、数为**88**个

http77soft.gougou.convsearch?search=%c7%a7%c7,x.a7%be%b2%cc'/<.

http^7soft.gougou.com/search?search=%57%69%6e%61%6d%

http^soft.gouqou .convsearch? search =%bf%el %b9%b7 & re stype=2&id=6 http^soft.gougou.coin/seard)?search=%bf%el%ce%d2%d2%f4%cO%

http7/soft.gougou.convsearch?search=%41%64%6f%62%65%20%52%

httpV/soft.gougou.convsearch<search=%ba%c6%b7%bd%c6%b(i%cc,%.

http7/soft.gougou.convsearch?search=%43%6f%6f%70%65%6e%d7%

http^/soft.gou gou.com/sea rdi?searrh=%c(i%e6%bc^>d2%h 1 %a6%bl %

<http://pstatic.xunlei.com/abouVproduct/down-xl5.htnihttp://www.gou> gou .conVdu ty/du ty .h tm I http^wwAv.sznetl 10.gov.cn/inilex.jsp

花费时间为531ms

支持AJAX的网络爬虫系统对http://soft.gougou.com的抓取结果

**B Crawlerssoft. log -**记李本

文件9 郦fl。)格式**(Q)**查看叩\*?助咚

h <ftp://soft.gougou.co> nVsearch?search=%b4%f3\*^d6%c7%bb?i.db&restype=2&id=6 [http://soft.gougou.com/search?search=%cd%a^](http://soft.gougou.com/search?search=%cd%25a%5e)bb%5e.a8'lz%3c.cb%b34restype=2&i(i=6)[)](http://soft.gougou.com/search?search=%cd%25a%5e)bb%5e.a8'lz%3c.cb%b34restype=2&i(i=6)[bb^.a8'](http://soft.gougou.com/search?search=%cd%25a%5e)bb%5e.a8'lz%3c.cb%b34restype=2&i(i=6)[lz](http://soft.gougou.com/search?search=%cd%25a%5e)bb%5e.a8'lz%3c.cb%b34restype=2&i(i=6)[<.cb%b34restype=2&i(i=6](http://soft.gougou.com/search?search=%cd%25a%5e)bb%5e.a8'lz%3c.cb%b34restype=2&i(i=6) http^/soft.gon gou.com/search ?search=%b9%c9%c6%bl O;.(i6%ae%d0%c7&restype=2&id=6 http:^soft.gougou.a)iiVsearch?search=%b9%c9%b3%c7%c4%a3€X,c4%c2%b3%b4%b9%c«&restypc=2&id=6 http://sofl.gougou.(vm/sedrch?sedrdi=%c8%f0%d0%c7%bf%d8%bf%d8&restype=2&id=6 http^soft.gougou.com/search?search=%33%36%30%b0%b2%c8%ab%ce%c0%ca%bf&restype=24id=6 http7/soft.gougou.conVsearch?search=%bd%fO%c9%bd%c7,)!>e5%cO%ed%d7%aB%bc,K»d2&restype=2&id=6 httpV/soft.gougou.roin/search?search=%cd%ea%c3%c03；l(i0%h6%d4%dB&restype = 2&id=6 <http://soft.gougou.com/scarchTscarch> =%c7%a7%c7%a7%bc%b2%cc^.fd&restypc=2&id=6 http7/soft.gougou.convsearch?search=%57%69%6e%61%6d%70&restype=2&id=6

h ttp:〃sof t.gou gou .convsea rch ? search =%bf%el %b9%b7&re stype=2 &id=6 http://soft.gougou.conVsearch?search=%bf%el%ce%d2%d2%f4%c0%(16%ba%(10&restype=2&i(i=6http://soft.gougou.conVsearch?search=%41%64%6f4X.62%65%20%52%65%61%64%55%72Arestype=2Aid=6 http^soft.gougou.com/search?search=%ba%c6%b7%bd\*?4>c6%bd%cc,X.a8&restype=2&id=6 [http://soft.gougou.com/search?search=%43%6f%6f%70%65%6e%d7%c0%c3%e6%bb%ad%bl%a8Arestype=2&id=6](http://soft.gougou.com/search?search=%43%6f%6f%70%65%6e%d7%c0%c3%e6%bb%ad%25bl%a8Arestype=2&id=6) http：Z/，soft.gougou.conVsearch?searrh=%cd%e6%bcf^d2f'obi%a6%bt%a64restype=2&i(i=6 http://pstatic.xunlei.com./abouVproduct/do^vn-xl5.htm

http:ZZwww.goi】gou .conVdu ty/du ty.htm I <http://www.sznetl10.gov.cn/index.jsp> 传统网络爬虫抓取嚥总数为88个 花费时间为500ms|

传统网络爬虫系统对http://soft.gougou.com的抓取结果

5.3结论

对于AJAX站点的测试，我们经过如图5. 5和5. 6的对比，我们可以看出:支持AJAX站点的网络爬虫抓取的链接数明显多于传统网络爬虫，但与此同时， 所花费的时间也是传统网络爬虫的十多倍。可以说支持AJAX框架的网络爬虫对 AJAX站点的链接抓取是有效的，但是相应的代价是时间比较长。这个主要是因 为脚本解析模块需要花费的时间比较长，在解析之后重建DOM树也需要比较长的 时间。对于非AJAX站点的链接抓取，同样存在时间较长的问题，抓取能力与传 统爬虫相比，优势不大.

抓取的链接总数对比

600

抓取总数(个)

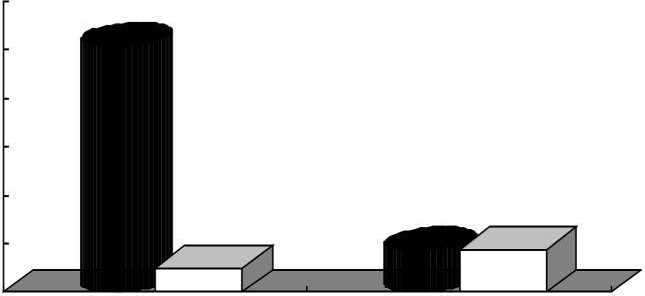
500

400

300

200

100

0

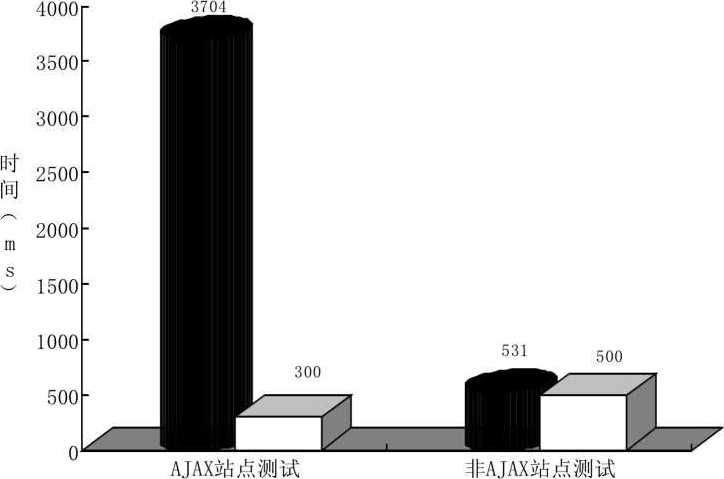
AJAX站点测试

非AJAX站点测试

■支持AJAX的网络爬虫

口传统网络爬虫

抓取的链接所需时间对比



■支持AJAX的网络爬虫

□传统网络爬虫

由上面的测试分析结论，我们可以进一步得出如下结论：

1从抓取链接总数来看，本文所设计的支持AJAX的网络爬虫对于非AJAX 站点，抓取链接总数与传统爬虫相同；对AJAX站点的所抓取的链接总数远远超 出传统爬虫，由此可见，本文所设计的网络爬虫系统对于AJAX站点链接的抓取 是切实有效的。

2从抓取链接所需时间来看，本文所设计的支持AJAX的网络爬虫对于非 AJAX站点链接抓取速度与传统网络爬虫抓取速度相当，而对于AJAX站点抓取速 度由于链接抓取的困难，花费时间比较长，但是平均计算速度，两种爬虫抓取单 位链接所花费的时间基本相同。我们还需要进一步的改进算法，提高抓取速度， 以突显出支持AJAX站点的网络爬虫系统的优势。

第6章总结与展望

6.1总结

虽然网络爬虫技术很早就被提出来了，但是传统的网络爬虫只是针对静态网 页，并不适合AJAX框架网站。现有深层网络爬虫技术无法有效的对AJAX站点链 接进行抓取。这是由于目前的网络爬虫技术在对AJAX框架网站的解析过程中存 在着：URL参数几何级数的无效组合、不能够识别事件触发顺序、缺乏错误捕获 和处理能力等问题，导致大量数据不能被搜索引擎有效检索。

本文针对上述问题，在对网页代码进行网页分析之后，釆用基于对象的程序 切片算法，利用脚本执行引擎对切片进行计算执行，将与URL相关的切片程序计 算出来，重建网页DOM树。经过程序切片与脚本执行引擎之间的互操作，根据脚 本执行之后的网页DOM树抓取网页中的链接，最终解决AJAX框架中URL提取以 及异步JavaScript网络爬虫系统的关键技术问题，以达到抓取AJAX站点URL 的目的。

本文从理论和技术方法上总结并提出了 AJAX框架网站中URL关联信息提取 方法、切片代码的有序执行以及程序切片模块研究，爬虫模块，脚本执行模块之 间的互操作协议，并为AJAX框架网站网络爬虫提供了新的解决方案，设计实现 了支持AJAX站点的网络爬虫系统。其创新之处有以下两点：

1、 在面向对象语言切片技术的基础上，提出了基于对象语言的切片算法；

JavaScript语言是基于对象的语言。在对JavaScript程序进行语法和语义 分析构建JavaScript程序的依赖程序层次模型，构造对象层次子图、控制依赖 与数据依赖子图，从而构建基于对象的系统依赖图。再次利用依据网页分析模块 构建的切片准则来逆向遍历基于对象的系统依赖图，最后计算出与URL相关的切 片。

2、 在现有网络爬虫技术基础上，提出了基于程序切片算法的支持AJAX的网 络爬虫。

基于程序切片算法的AJAX框架网络爬虫能较快的剔除无关因素,只对与URL 相关的切片进行计算，这样能快速对有价值的脚本分离执行，从而可以有效提高 支持AJAX的网络爬虫的抓取效率。

6.2展望——不足与改进

探讨AJAX框架网络爬虫解决方案是本论文研究工作的重点。目前的研究工 作虽已取得了一定的积极的研究成果，但同时也遇到了不少问题。与传统爬虫相 比较，支持AJAX站点的网络爬虫在处理中多了程序切片、脚本解析以及DOM操 作三个步骤，特别是脚本执行和程序切片部分，在同等条件下会减慢爬虫抓取网 页的速度，对爬虫的性能影响很大。要实现AJAX框架网站链接的抓取的高效率， 还需要对程序切片规则和脚本解析的效率问题进行进一步的研究。由于程序切片 规则需要対AJAX框架进行分析之后获取，所以需要随着AJAX框架的不断变化而 不断的学习和完善。而脚本解析需要花费大量的时间，这使得网络爬虫消耗时间 比较长，不断完善程序切片规则，提高脚本解析的效率是亟待解决的问题。

本文初步探讨了支持AJAX网络爬虫的设计与实现，随着Web2. 0概念在Web 开发中的深入，将会有更多的技术用于提升用户交互体验和界面上，对于搜索引 擎的爬虫抓取网页带来更大的挑战，在不久的将来，随着研究的深入，相信会出 现更多支持AJAX等技术生成动态内容的网络爬虫，适应于所有形式网站的网络 爬虫的出现也是指日可待的，从而为搜索引擎查询结果的查准率和查全率提供有 力支持，最终让网络成为我们最有力的助手，为我们提供最迅速，最便捷的信息 服务。

参考文献

陈珂，陈小英,徐科.2007. Hidden Web信息获取［J］.计算机时代,54-56

蒋勇.2007,基于网络的农业信息抽取与数据挖掘系统［D］:［硕士］ .合肥：中国科学技术 大学，6-9.

李保利，陈玉忠，俞士汶.2003.信息抽取研究综述［J］.计算机工程与应用，8-17.

李必信.2006.程序切片技术及其应用［M］.北京:科学出版社,1-14.

李涛，陈鹏，李哲.2007.深度Web资源探测系统的研究与实现［J］. 185-187

刘金红，陆余良.2007.主题网络爬虫研究综述［J］.计算机应用研究.24(10):26-29

吕秀娟.2002.搜索引擎中资源获取的设计与实现［D］:［硕士］.南京：南京理工大学13-17 彭建荣，罗永会.2006,搜索引擎的基本原理及发展趋势［J］,电脑知识与技术,84-85 孙彬，王东，李娟.2007.基于XQuery的Deep Web搜索系统的设计与实现［J］.科学技术与工 程 7(16):4080-4084

宋晖张岭叶允明马范援.2002.基于标记树对象抽取技术的Hidden Web获取研究［j］. 计算机工程与应用.9-12, 15

王伟,陈平.2000.程序切片技术综述［J］.微电子学与计算机.25-27, 48

王映，于满泉,李盛韬，王斌，余智华.2001. JavaScript引擎在动态网页釆集技术中的应用 ［J］.计算机应用，33-36.

严亚兰.2003.面向动态网页爬行的Crawler架构［J］.图书情报知识，51-53

佚名.2005. WEB2. 0 概念诠释［J］. <http://tech>. 163. com/05/0912/11/1TERVH5N000915BF. html

佚名.2005. web 搜寻器［J］. <http://publib>. boulder, ibm. com/infocenter/wsiihelp/v8r 3/topic/com. ibm. websphere, ii. esearch, ad. doc/administering/iiysacweb. htm.

于静.2008.半结构化网络信息抽取技术及应用研究［D］:［硕士］ .合肥：中国科学技术大 学，12-17.

曾伟辉，李淼.2008.深层网络爬虫研究综述［J］.计算机系统应用，122-126

张秀清. 2006. WEB信息提取以及数据仓库的研究与实践［D］:［硕士］.北京：中国科学院 研究生院，6-8.

张勇翔.2000. JAVA语言的程序切片生成技术［J］, 7-10.

郑冬冬，赵朋朋，崔志明.2005. DeepWeb爬虫研究与设计［J］.清华大学学报(自然科学 版),45(S1):1896-1902

郑冬冬，崔志明.2006. Deep Web爬虫爬行策略研究［J］.计算机工程与设计，27 (17)： 3154-3158

周津.2004.基于XML的网页信息抽取[D]:[硕士] .合肥：中国科学技术大学，27-30. 周立柱，林玲.2005.聚焦爬虫技术研究综述[J].计算机应用,25(9):1965 - 1969

Ali I. El-Desouky, Hesham A. Ali, Sally M. El-Ghamrawy. 2006. An Automatic Labe 1 Extraction Technique for Domain-Specific Hidden Web Crawling (LEHW) [C]. The 2006 International Conference on Computer Engineering and Systems:454-459

Ali Mesbah, Engin Bozdag, Arie van Deursen. 2008. Crawling AJAX by Inferring User Interface State Changes[C]. Proceedings of the 2008 Eighth International Conference on Web Engineering.122-134

Bergholz, A. , Chidlovskii, B. 2003. Crawling for Domain-Specific Hidden Web Resources [J]. Web Information Systems Engineering. 125-133

BrightPlanet. com. 2000. The deep web: Surfacing hidden value. Accessible at http: //www. completeplanet. com/Tutorials/DeepWeb [EB/OL]

Chang K C C, He B, Li C, et al. 2004. Structured databases on the web: Observations and implications [C] . SIGMOD Record, 33 (3)

Cohen W, Ravikumar Pradeep, Fienberg Stephen E. 2003. A Comparison of String Distance Metrics for Name-Matching Tasks[C]. Proceedings of 1JCAI-03 Workshop on Information Integration on the Web (IIWeb-03) 73-78

Cope J, Cr a swell N, Hawking D. 2003. Automated discovery of search intefaces on the Web[C]. Proceedings of the 14th Australian Database Conference (ADC 2003) : 181-189

Cristian Duda, Gianni Frey, Donald Kossmann, Chong Zhou. 2008. AJAXSearch: Crawl ing, Indexing and Searching Web 2. 0 Applications[C]. Proceedings of the VLDB 1440-1443 eTesting L. 2000. Google Web Search Engine Evaluation.

Harman M, Danicic S. 1995. Using program slicing to simplify testing[C]. The EuroSTAR, 94 conference and Software Quality Engineering. 143-162

Harman M, RM Hierons. 2001. An overview of program slicing [J]. Software Focus.

Jesse James Garrett, 2005. Ajax: A New Approach to Web Applications. Accessible at <http://ww>. adaptivepath. com/ideas/essays/archives/000385. php.

J Roberto Basili, Maria Teresa Pazienza. 1997. Lexical Acquisition and Information Extraction [J]. Information Extraction： A Multidisciplinary Approach to an Emergine Information Technology, 1299： 44—72.

Lpeirotis P, Gravano L. 2002. Distributed search over the hidden web： Hierarchical database sampling and selection[C]. Proceedings of the 28th international conference on Very Large Data Bases 394-405

Luciano Barbosa, Juliana Freire. 2007. An Adaptive Crawler for Locating Hidden-Web

Entry Points[C]. WWW2007 441-450

Manuel Alvarez, Alberto Pan, Juan Raposo, Angel Vina. 2001. Client-Side Deep We b Data Extraction extended paper. Accessible at <http://ww>. tic. udc. es/^mad/pu blications/csdeepWeb extended, pdf.

Manuel Alvarez, Alberto Pan, Juan Raposo, Angel Vina. 2004. Client-Side Deep Web Data Extract ion[C]. Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business. 158-161

Manuel Alvarez, Juan Raposo, Alberto Pan, Fdel Cacheda, Fernando Bellas, Victor Carneiro, 2007. DeepBot: A Focused Crawler for Accessing Hidden Web Content[C]. Proceedings of the 3rd international workshop on Data engineering issues in E-commerce and services: In conjunction with ACM Conference on Electronic Commerce, 18-25,

Ntoulas Alexandros, Zerfos Petros, Cho Junghoo. 2005. Downloading Textual Hidden Web Content Through Keyword Queries[C]. Proceedings of the 5th ACM/1EEE Joint Conference on Digital Libraries. 100-109

Raghavan S., Garcia-Molina, H. 2000. Crawling the Hidden Web. Technical Report.2 000-36, Computer Science Department, Stanford University. Available at http:// ilpubs. Stanford, edu:8090/456/1/2000-36. pdf)

Ralph Grishman. 1997. Information Extraction： Techniques and chai lenges[M]. Springer Berlin / Heidelberg 1299： 10—27.

Ryan Levering, Michal Cutler. 2006.The Portrait of a Common HTML Web Page[C]. Proceedings of the 2006 ACM symposium on Document engineering. 198-204

Stephen W. Liddle David W. Embley Del T. 2002. Extracting Data Behind Web Forms[C].

Scott Sai Ho Yau Proceedings of the 28th VLDB Conference, HongKong, China, 402-413

Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng. 2007 Annotating Structured Data of the Deep Web. Data Engineering[C], IEEE 23rd International Conference on 376-385

Yoo Jung An, James Geller, Yi~Ta Wu and Soon Ae Chun. 2007. Automatic Generation of Ontology from the Deep Web[C]. Proceedings of the 18th International Conference on Database and Expert Systems Applications 470-474

Zhang, Z., He, B., C. -C. Chang, K. 2005. Light-weight Domain-based Form Assistant: Querying Web Databases On the Fly[C]. Proceedings of the 31st VLDB Conference 97-108

致谢

三年的研究生生活即将划上句号，在这三年的学习和生活中，老师、同学、 亲人、朋友给予我的关心和帮助将永远留在我美好记忆的深处。

首先，我要衷心地感谢我的恩师李淼研究员，从论文的选题、开题、研究进 展，至最终论文的撰写，每个过程无一不凝结着导师辛勤的心血。在我攻读硕士 研究生期间，深深受益于李老师的关心、爱护和谆谆教导，李老师以她深厚的理 论知识、实践经验以及对研究方向的良好把握，指引我从事科学研究，使我在研 究过程中少走了很多弯路。李老师广博的学识、严谨的治学态度、开阔的思维方 式、一丝不苟的工作作风都给我留下了深刻的印象，并将使我受益终生。

在三年的学习和工作中，我还得到了智能农业实验室张建老师的很多帮助。 张老师不仅在理论方法和工作等方面给予我指导和支持，还在生活上给予了我无 微不至的关心和照顾，在此特向张老师表示由衷的谢意！

在整个论文的进展过程中，我得到了实验室小组李静、耿英、杨攀、骆凯等 同学的指导与帮助，在此表示感谢！在课题研究与项目开发过程中，我们协作互 助，顺利完成了本文的任务。

感谢张浩东、万莉、袁媛、胡泽林等老师在项目工作中的指导与帮助，与他 们一起在实验室学习生活的点点滴滴都是愉快而难忘！

感谢中科院合肥智能机械研究所科教处的葛运建、刘善文、陈峰老师在我攻 读硕士学位期间所给予的关怀与帮助！

感谢审稿和答辩委员会各位老师对我的耐心指导！

感谢所有曾帮助过我、关心过我却无法在此一一提及的善良的人们，他们给 予我的关心和帮助，我将永远铭记在心！

最后，谨以此文献给我的家人和朋友，他们无私的关爱和殷切的期望是我前 行的最大动力！

在读期间发表的学术论文与取得的研究成果

已发表论文：

1. 曾伟辉，李淼.基于JavaScript切片的AJAX框架网络爬虫技术研究. 计算机系统应用.已录用.
2. 曾伟辉，李淼.深层网络爬虫研究综述.计算机系统应用.2008.05, 1 22-126.
3. 李淼，张建，曾伟辉，张伟，陈曦，万莉.基于AJAX框架的网络爬虫系 *统.*软件登记：2009SR016090.

参与项目：

中国科学院知识创新工程重要方向项目（编号：KGCX2-SW-5U）