


大连理工大学
信息检索研究室
Information Retrieval Laboratory of DUT

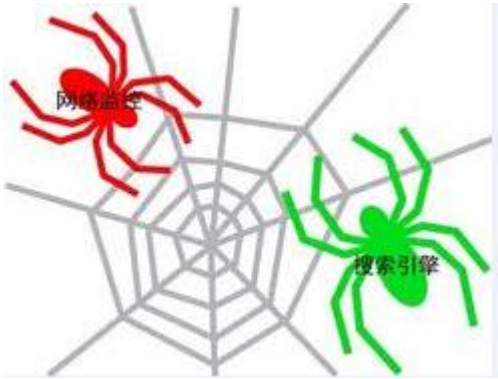
网络爬虫

刘文飞

2017-10-12

网络爬虫的定义

- 网络爬虫又称网络蜘蛛，网络机器人。
- 网络爬虫是一个自动提取网页的程序，它为搜索引擎从万维网上下载网页，是搜索引擎的重要组成。爬虫一般从一个或若干初始网页的URL开始，获得初始网页上的URL，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，直到满足系统的一定停止条件。



爬虫分类

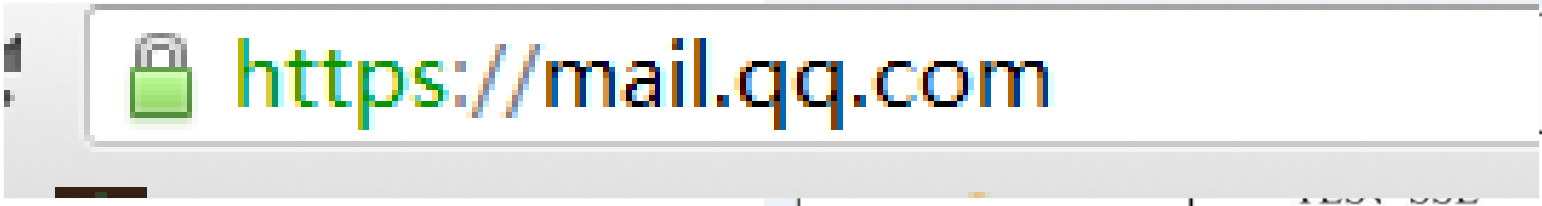
- **非定向爬虫**
  - 爬取互联网上任何基于Http协议的内容
  - 工具：Larbin、Ncrawler，Heritrix、Nutch、Scrapy…
- **定向爬虫**
  - 根据网站自身的属性采用特定的爬取策略
  - 工具包：HttpClient（Java和C#均已携带封装好的类库）

HTTP协议

- HTTP：Hyper Text Transfer Protocol（超文本传输协议）
- 万维网协会和Internet工作小组，1999年6月发布了RFC 2616，定义了今天普遍使用的HTTP/1.1
- HTTP协议是用于从WWW服务器传输超文本到本地浏览器的传送协议，属于应用层协议，由请求和响应构成，是一个标准的客户端服务器模型

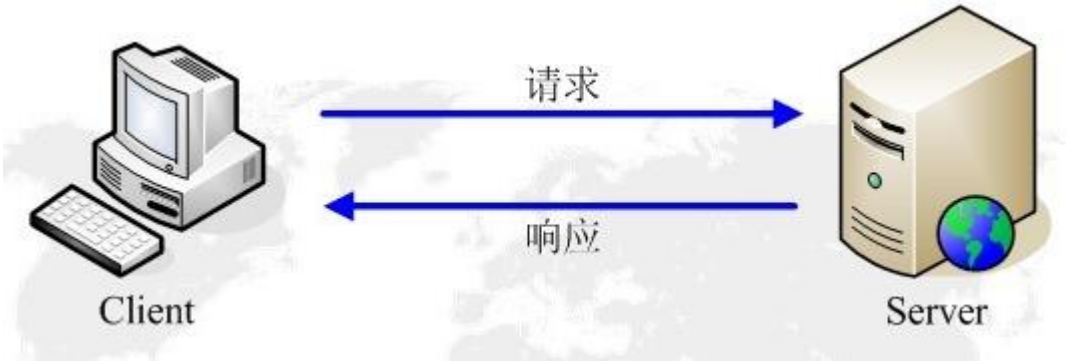
HTTP在TCP/IP协议栈中的位置

- HTTP协议通常承载于TCP协议之上，有时也承载于TLS或SSL协议层之上（这就是所说的HTTPS）
- 默认HTTP端口为80，HTTPS端口为443



HTTP的请求响应模型

- HTTP协议永远都是客户端发起请求，服务器回送响应（无法推送）
- HTTP协议是一个无状态的协议，同一个客户端的这次请求和上次请求没有对应关系（Cookie & Session）



HTTP协议工作流程

- （1）首先客户端与服务器需要建立连接（只要单击某个超链接，HTTP的工作就开始了）
- （2）建立连接后，客户机向服务器发送请求
- （3）服务器接收到请求后，给予相应的相应信息
- （4）客户端接受服务器所返回的信息通过浏览器显示在用户显示屏上，然后客户端与服务器断开连接

HTTP协议之请求

- HTPP请求由三个部分组成：**请求行、消息报头、请求正文**

```
⊟ Hypertext Transfer Protocol
  ⊟ GET /pv/pv.gif?t=0 HTTP/1.1\r\n
    ⊞ [Expert Info (Chat/Sequence): GET /pv/pv.gif?t=0 HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /pv/pv.gif?t=0
      Request Version: HTTP/1.1
  Accept: */*\r\n
  Referer: http://image.baidu.com/\r\n
  Accept-Language: zh-cn\r\n
  Accept-Encoding: gzip, deflate\r\n
  If-Modified-Since: Wed, 19 Aug 2009 15:23:32 GMT\r\n
  If-None-Match: "557649757"\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.4506.2152; .NET CLR 3.5.21022)\r\n
  Host: image.baidu.com\r\n
  Connection: Keep-Alive\r\n
  Cookie: iCast_Rotator_1_1=1259581841765; iCast_Rotator_1_2=1259586044296; BAIDUID=50265E09E7592D1C415755687611D9F9:FG=1; BD_UTK_DVT=1\r\n
  \r\n
```

HTTP协议之请求 – 请求行

- 请求行：**Method Request-URI HTTP-Version CRLF**

  例：GET /index.jsp HTTP/1.1 (CRLF)

GET	请求获取Request-URI所标识的资源
POST	在Request-URI所标识的资源后附加新的数据
HEAD	请求获取由Request-URI所标识的资源的响应消息报头
PUT	请求服务器存储一个资源，并用Request-URI作为其标识
DELETE	请求服务器删除Request-URI所标识的资源
TRACE	请求服务器回送收到的请求信息，主要用于测试或诊断
CONNECT	保留将来使用
OPTIONS	请求查询服务器的性能，或者查询与资源相关的选项和需求

HTTP协议之请求 – 消息报头

- Accept：浏览
- Accept-Chars
- Accept-Encoc
- Accept-Langu
- Authorizatior
- Connection：
- Content-Leng
- Cookie：这是
- Host：初始UF
- Referer：跳转
- User-Agent：

按照内容类型排列的 Mime 类型列表

类型/子类型	扩展名
application/envoy	evy
application/fractals	fif
application/futuresplash	spl
application/hta	hta
application/internet-property-stream	acx
application/mac-binhex40	hqx
application/msword	doc
application/msword	dot
application/octet-stream	*
application/octet-stream	bin
application/octet-stream	class
application/octet-stream	dms
application/octet-stream	exe
application/octet-stream	lha
application/octet-stream	lzh
application/oda	oda
application/olescript	axs
application/pdf	pdf
application/pics-rules	prf
application/pkcs10	p10
application/pkix-crl	crl
application/postscript	ai

HTTP协议之响应

- HTPP响应由三个部分组成：**状态行、消息报头、响应正文**

```
⊟ Hypertext Transfer Protocol
  ⊟ HTTP/1.1 200 OK\r\n
    ⊞ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Request Version: HTTP/1.1
      Response Code: 200
    Content-Type: image/gif\r\n
    ETag: "567281165"\r\n
    Accept-Ranges: bytes\r\n
    Last-Modified: Wed, 19 Aug 2009 15:23:26 GMT\r\n
    Expires: Mon, 30 Nov 2009 13:15:39 GMT\r\n
    Cache-Control: max-age=0\r\n
  ⊟ Content-Length: 0\r\n
      [Content length: 0]
    Date: Mon, 30 Nov 2009 13:15:39 GMT\r\n
    Server: Apache\r\n
    \r\n
```

HTTP协议之响应 – 状态行

- 状态行： **HTTP-Version Status-Code Reason-Phrase CRLF**

  例： HTTP/1.1 200 OK （CRLF）

  状态代码有三位数字组成，第一个数字定义了响应的类别，且有五种可能取值：

  1xx：指示信息--表示请求已接收，继续处理

  2xx：成功--表示请求已被成功接收、理解、接受

  3xx：重定向--要完成请求必须进行更进一步的操作

  4xx：客户端错误--请求有语法错误或请求无法实现

  5xx：服务器端错误--服务器未能实现合法的请求

HTTP协议之响应 – 状态行

- 常见状态代码、状态描述、说明：
  - 200 OK　　//客户端请求成功

    400 Bad Request　//客户端请求有语法错误，不能被服务器所理解

    401 Unauthorized //请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用

    403 Forbidden　//服务器收到请求，但是拒绝提供服务

    404 Not Found　//请求资源不存在，eg：输入了错误的URL

    500 Internal Server Error //服务器发生不可预期的错误

    503 Server Unavailable　//服务器当前不能处理客户端的请求，一段时间后可能恢复正常

HTTP协议之响应 – 消息报头

- Location：用于重定向接受者到一个新的位置
- Server：服务器用来处理请求的软件信息

HTTP相关知识点 - Session

- Session机制是一种服务器端保存用户状态的机制，服务器使用一种类似于散列表的结构来保存信息。(比如未登录状态下购物车的实现)
- 客户端维护Session ID的方式
  - Cookie
  - URL重写
  - 表单隐藏字段

HTTP相关知识点 - Cookies

- Cookies是客户端保存状态的一种方案
  - 会话性质的cookie，存放在浏览器内存
  - 持久化的cookie，存放在硬盘上
- Cookies可以记录你的用户ID、密码、浏览过的网页、停留的时间等信息。当你再次来到该网站时，网站通过读取Cookies，得知你的相关信息，就可以做出相应的动作（如在页面显示欢迎你的标语，或者让你不用输入ID、密码就直接登录等等）

HTTP相关知识点 – 压缩

- HTTP压缩
- HTTP压缩
- HTTP压缩文件。
- 网页压缩

请输入要查询网址： 113.10.161.28/Manage/MainFrame.aspx 查 询

网址 113.10.161.28/Manage/MainFrame.aspx 检测结果如下：	
是否压缩	是
压缩类型	gzip
原始文件大小	2547 字节
压缩后文件大小	925 字节
压缩率（估计值）	63.68%

Header信息	
Cache-Control	private
Date	Wed, 20 Mar 2013 11:26:49 GMT
Content-Type	text/html; charset=utf-8
Server	Microsoft-IIS/6.0
X-Powered-By	ASP.NET
X-AspNet-Version	2.0.50727
Content-Encoding	gzip
Vary	Accept-Encoding
Transfer-Encoding	chunked

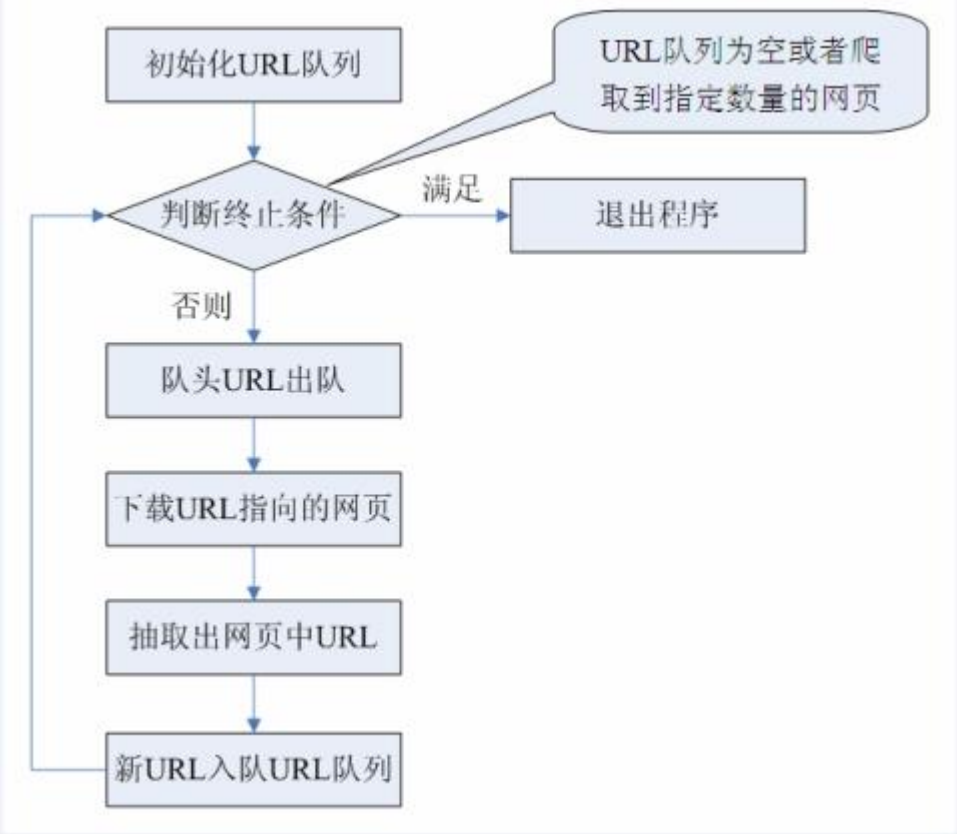
HTTP相关知

- JSON 即 Java
，非常适合于
- JSON 是基于
的，因此，JS
String，Nun
Object 对象。

JSON示例

```
{
    "statuses": [
        {
            "created_at": "Tue May 31 17:46:55 +0800 2011",
            "id": 11488058246,
            "text": "求关注。",
            "source": "<a href="http://weibo.com" rel="nofollow">新浪微博</a>",
            "favorited": false,
            "truncated": false,
            "in_reply_to_status_id": "",
            "in_reply_to_user_id": "",
            "in_reply_to_screen_name": "",
            "geo": null,
            "mid": "5612814510546515491",
            "reposts_count": 8,
            "comments_count": 9,
            "annotations": [],
            "user": {
                "id": 1404376560,
                "screen_name": "zaku",
                "name": "zaku",
                "province": "11",
                "city": "5",
                "location": "北京 朝阳区",
                "description": "人生五十年，乃如梦如幻；有生斯有死，壮士复何憾。",
                "url": "http://blog.sina.com.cn/zaku",
                "profile_image_url": "http://tp1.sinaimg.cn/1404376560/50/0/1",
                "domain": "zaku",
                "gender": "m",
                "followers_count": 1204,
                "friends_count": 447,
                "statuses_count": 2908,
                "favourites_count": 0,
                "created_at": "Fri Aug 28 00:00:00 +0800 2009",
                "following": false,
                "allow_all_act_msg": false,
                "remark": "",
                "geo_enabled": true,
                "verified": false,
                "allow_all_comment": true,
                "avatar_large": "http://tp1.sinaimg.cn/1404376560/180/0/1",
                "verified_reason": "",
                "follow_me": false,
                "online_status": 0,
                "bi_followers_count": 215
            }
        },
        ...
    ],
    "previous_cursor": 0,                    // 暂未支持
    "next_cursor": 11488013766,      // 暂未支持
    "total_number": 81655
}
```

爬虫流程



- 爬虫抓取策略
- 网页地址过滤
- 网页更新去重
- 网页解析
- 多线程并发爬取

爬虫抓取策略

- **深度优先搜索策略**
- **广度优先搜索策略**
- **最佳优先搜索策略**
  - 可能根据主题相似度、反向链接数、PR值等策略

网页地址过滤

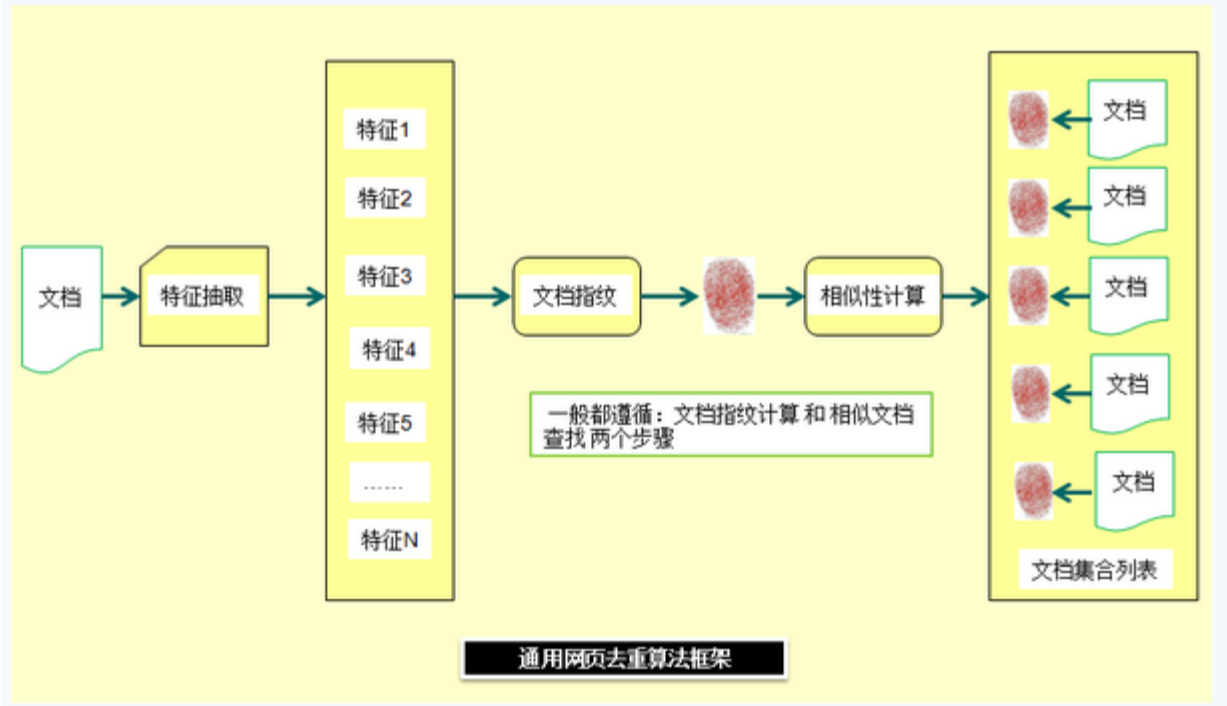
- **正则表达式**
  - 可以过滤非正规的网址、无需下载的文件（后缀名）或特定域名下的网页
- **建立IP规则库**
  - 如若建立校内搜索引擎，则在爬取时将所有非校内IP过滤掉

网页更新策略

- **历史参考策略**
  - 据页面以往的历史更新数据，预测该页面未来何时会发生变化。
- **用户体验策略**
  - 根据用户点击信息优先爬取质量较高/关注度高的页面
- **聚类抽样策略**
  - 无需保存历史信息，解决冷启动问题（无历史信息的网页）

网页去重策略

- **MD5值比较法**
  - 缺点：精确匹配才算重复
- **网页指纹法**



网页解析

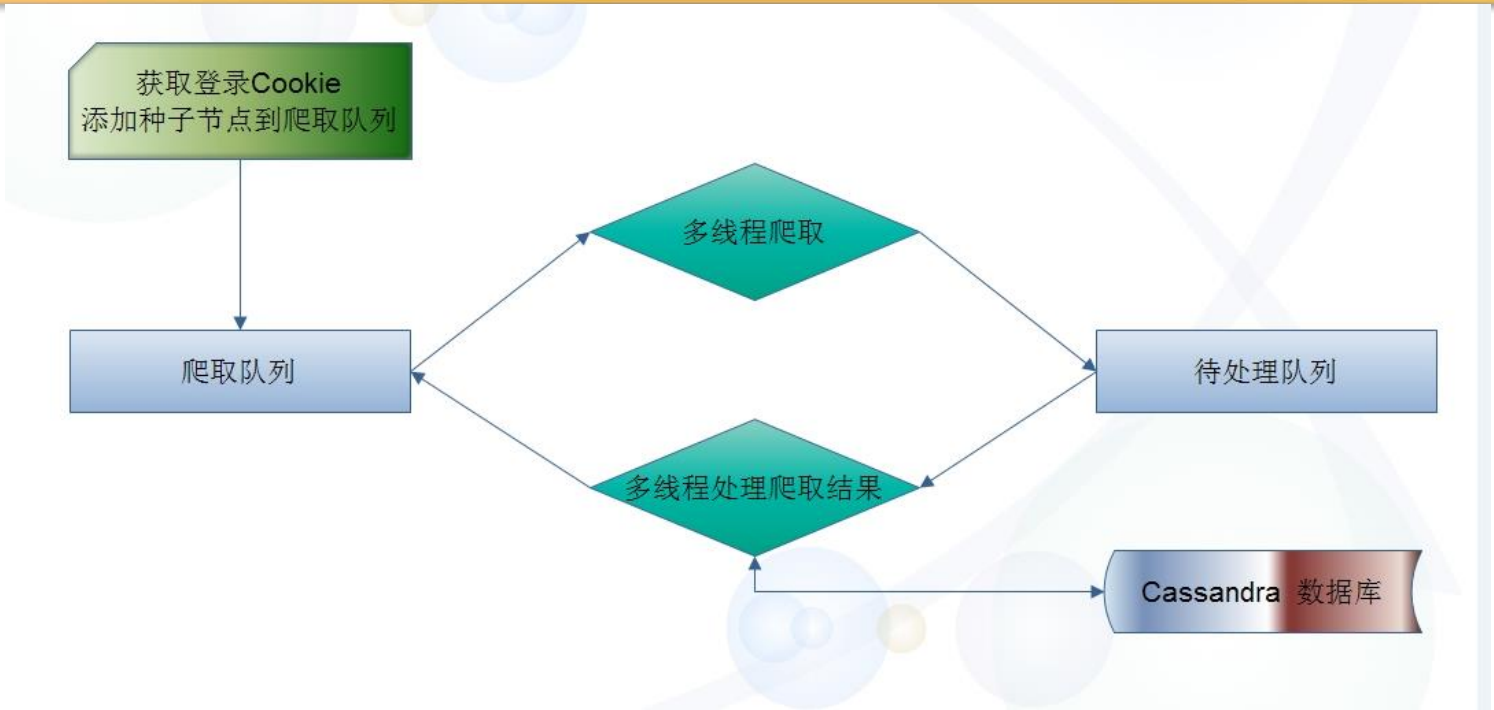
- **主要内容抽取**
  - TIKA，可抽取HTML, PDF, MS-*, Image(元数据), XML等
  - Lucene提供工具包抽取HTML（较粗糙，容易出错）
  - cx-extractor，基于行块分布函数的通用网页正文抽取算法（哈工大）
    http://code.google.com/p/cx-extractor/
- **特定内容抽取**

```
Lexer lexer = new Lexer(html);
Parser parser = new Parser(lexer);

NodeFilter divFilter = new AndFilter(new TagNameFilter("div"), new HasAttributeFilter("id", "photo"));

NodeList divNodes = parser.Parse(divFilter);
```

多线程并发爬取



- **多线程中主要问题**
  - 网络带宽
  - 服务器对爬虫请求频率的限制
  - 异常处理（多次爬取、日志记录）

爬虫相关知识点 – robots.txt

- robots.txt（统一小写）是一种存放于网站根目录下的ASCII编码的文本文件，它通常告诉网络蜘蛛，此网站中的哪些内容是不应被搜索引擎的漫游器获取的，哪些是可以被获取的。


```
User-agent: *
Disallow: /App_Code/
Disallow: /App_Data/
Disallow: /SiteManager/
Disallow: /UserControls/
```

谢谢！