面向动态网络的爬虫的设计与实现

摘要

# 1.简介

# 2.相关工作

爬虫

动态网页，即是指通过AJAX加载，或者通过JS懒加载的文档内容。这一技术在电商平台，如淘宝，京东，和通讯软件如微信朋友圈广泛应用。电商平台和通讯软件包含着很多非常有价值的数据，而通过爬虫去批量的获取它们是十分高效并且值得的。

# 3.设计

本论文致力于面向动态网页的爬虫的设计和实现。如同一般的通用爬虫，动态网页爬虫也必须要有URL队列，页面爬行，链接提取，内容入库，文档处理等功能。这些功能相互独立，因此，模块化将是一个明智的选择。

动态网络爬虫有别于普通爬虫的地方就是页面爬行部分，普通爬虫抓取静态页面后即进行后续步骤，而通过JS事件触发的改变文档内容的部分则会被忽略。动态网页爬虫将会模拟诸如下拉页面等操作，尽可能加载JS事件，以获取尽量全面的静态文档。那么，在模块化的基础上，只需要修改页面爬行模块即可，这样大大降低了各功能实现间的耦合度，也更便于实现和调试。

然而，无论如何优化，爬取动态页面的开销仍然是巨大的。下载一个静态页面的平均时间开销是10ms级，然而下载一个动态页面的平均时间开销是10^3ms级。因此，对于动态网页的爬取必须是定向的。试想对一个明显是静态页面的网页，比如百度首页，仍然去进行DOM建模，JS事件模拟等操作，无疑是对资源的一种极大的浪费。

## 3.1.架构

网络爬虫最基础的算法是，输入一个URL种子列表，然后重复以下步骤，从URL列表中取出一个URL，对该URL发起请求，下载对应的HTML文档，从下载的文档中提取URL链接，对每个链接进行绝对化，然后将其插入待下载的URL列表，并对URL列表进行去重，以确保待下载URL列表不会形成一个环。如果有需要的话，也可以对下载的HTML文档进行更深一步的处理，比如对其建立索引，搜索引擎会需要这一步。为了完成上述算法，需要以下几个模块：

1. 用以存储待下载的URL的容器，这里使用队列，以下称URL队列；
2. 根据协议下载URL所定位的文档的模块；
3. 从下载的文档中提取其中所包含的URL链接的模块；
4. 处理下载文档的模块，可以从下载的文档中提取所需的数据；
5. 将文档信息保存入数据库的模块，以便于管理和去重；

首先，需要从待下载URL队列中取出一个绝对URL，绝对URL是由策略（如“http”，“https”）或者“//”开头的，策略表明了下载该文档所需使用的网络协议。在PHP和phantomjs中，策略的选取是由运行环境自动检测的，虽然可以自行指定，但是对于大多数的页面，http协议都可以满足需求，即便是https网页，如百度，依然可以通过http访问，然后会收到一个带有Location: <https://www.baidu.com>的返回头。

在选择了合适的网络协议后，抓取方法将会被调用，其会向取出的绝对URL发出一个请求，然后根据响应头中的状态码进行对应操作，如果是200，也就是响应成功，就会将响应体下载下来，而如果是302，则会根据响应头中的Location进行跳转，如果是40X，则说明请求失败，服务器无法响应或者拒绝响应。

构造请求头不是一个轻松的任务，现今的大量网站都有一定的反爬虫策略，而绕过反爬虫这一课题，本身就可以展开为一个单独的课题，本文将不予详细描述。构造请求头对于绕过反爬虫有着举足轻重的作用，其对于合适的User-Agent，Referer，Connection，Cookie等的选择都可以对反爬虫策略造成干扰。在这里，因为本课题的重点并不是反反爬虫，仅仅使用一种简单的，模拟用户浏览器的请求头就可以满足需求。

当文档被下载下来以后，将被写入文件，以便于多线程或者分布式工作单元进行下一步处理。在写入文件的过程中，文件的命名建议用对URL或者文档内容的HASH进行，在数据表中，将URL与文件名关联以便于查找和更新。

在本爬虫中，仅对a标签中的href属性中的URL进行提取，即可以认为仅对HTML文档进行抓取，而不抓取多媒体资源。对于下载下来的文档，在写入文件的同时进行链接提取，a标签中的href属性中的URL有几种特征。根据RFC标准，URL由<scheme>://<user>:<password>@<host>:<port>/<url-path>构成，根据URL命名规则使用正则匹配进行链接提取。但是要注意，由于大量可以省略的部分，URL可以是省略scheme的，由//开头的，也可以是只有url-path的，省略域名的相对链接。相对链接是不可以脱离上下文独立存在的，因此对于相对链接要进行去相对化。然后才能存入数据库并插入URL队列。

3.2.数据表

# 4.实现

# 5.