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。

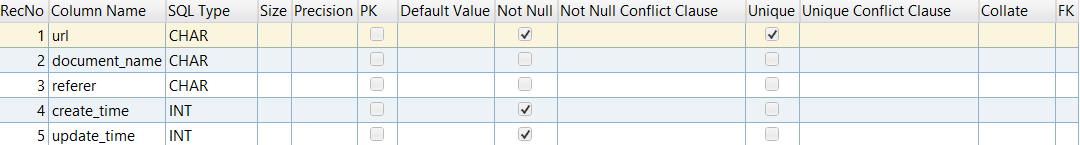
当文档被下载下来以后，将被写入文件，以便于多线程或者分布式工作单元进行下一步处理。在写入文件的过程中，文件的命名建议用对URL或者文档内容的HASH进行，在数据表中，将URL与文件名关联以便于查找和更新。

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。然后查询数据库判断该URL是否已被请求，为了避免对资源的浪费，更重要的是避免URL形成环，仅近期未被请求过的URL才会被录入数据库并且加入URL队列。

上述步骤就是一个基础的通用爬虫所遵循的，当然，其中省略了很多实现细节。设计一个能够有效处理大量条目的数据结构和数据表并不是一个简单的工作。其目标在于平衡时空开销。下一节将对数据表进行介绍。

## 3.2.数据表

对于处理过的URL和已下载的文档，需要进行高效的管理，也就是查询和修改。虽然可以自己设计一个数据结构用以进行该目的，但是使用数据库无疑是更方便高效的选择。既然要使用数据库，设计一个合理简约的数据表就是至关重要的。这张数据表必须能够将URL和根据URL下载的HTML文档关联起来，存储引用地址，文档创建时间和更新时间。也就是如下所示。



表中存储的是HTML文档名而不是直接存储HTML文档内容，以免HTML文档内容过多而对数据库造成不必要的负担。而因为URL中有部分字符不符合文件系统的命名规范，不能直接用URL作为文件名，这里的处理方法是对URL进行MD5加密，得到的是一个固定长度的，符合文件系统命名规范的哈希字符串，而因为该字符串并不包含文档内容的信息，无法直接通过该字符串进行检索，因为要将URL与文档名也就是该字符串进行关联。

有些网站会对请求头做REFERER检查，只有通过某些页面的引用才能访问该页面，因此就需要伪造REFERER请求头，因此存储REFERER是必不可少的。

使用数据库来存储爬虫爬取的文档信息有另一个得天独厚的优势，因为字段名可以设置为唯一的，因此URL去重在试图存储的过程中就自然而然的完成了。因为网站的维护者可能会在一定时间后更新页面内容，因此需要保存更新时间，以便在一定时间以后重新下载该URL对应的HTML文档。

# 4.实现

本爬虫由依赖注入和控制反转的模式实现，以降低各模块之间的耦合度。这里使用一个开源库Pimple作为依赖注入的容器。整个爬虫由文档抓取模块，链接提取模块，文档处理模块，数据库连接模块，数据访问对象模块，配置读取模块，调度模块和文档抓取模块组成。

## 4.1.链接提取模块

链接提取模块可以说是爬虫最基础也是最重要的部分，链接提取模块就仿佛用来点击网页上各个链接的鼠标，没有链接提取模块便无法获取当前HTML文档中指向其它页面的URL，爬行也便无从谈起。然而这一最基础最重要的模块却并不复杂。

链接提取的核心功能，也就是寻找a标签中的href属性，由正则匹配完成，如果使用的语言支持正则匹配，只需要两行代码，一行定义正则表达式，一行调用正则匹配函数。在本爬虫中使用一个久经考验的正则表达式

<a\s.\*?href\s\*=\s\*(\"[.#]+?\"|\'[.#]+?\'|[^\s]+?)(>|\s.\*?>)(.\*?)<[\/ ]?a>

该正则表达式匹配带有href属性的a标签，然而会将a标签完整的匹配下来，因此就需要一个剔除href属性之外所有字符的方法来将链接清洗干净。例如<a href=”http://www.a.com” hidden></a>将会被处理为http://www.a.com。这里有一个需要注意的地方，在HTML中单引号和双引号都可以表示字符串，href=后面既可能是单引号包裹的URL，也可能是双引号包裹的URL。

在对URL进行清洗以后，将得到一个包含该HTML文档中所有URL的列表，在该列表中会含有相对链接，而相对链接在脱离了上下文的情况下是毫无意义的，因此要根据RFC中对相对URL的规定，根据相对链接的类型对其进行处理。以‘/’开头的以该网站的根目录为前缀，而不以‘/’开头的相对链接则遵循目录访问的规则，其中‘.’代表当前目录，‘..’代表上级目录。

至此链接提取就完成了，将处理过的列表返回即可。该模块的输入是HTML文档和该文档对应的URL。对外没有任何依赖。

## 4.2.数据库连接模块和数据访问对象模块

数据访问对象模块是依赖数据库连接而模块的，而数据库连接模块的功能相对简单。数据库连接模块是为了适配不同的数据库而设计的，在应用过程中根据用户配置的不同可以返回不同的数据库连接对象。由于PHP语言PDO（PHP Date Object）扩展的强大功能，获取数据库连接对象的获取十分轻松，这也是PHP这门语言在内容处理上的优势。

数据访问对象模块对数据库连接模块进行了封装，数据访问对象模块是面向数据表的，其自动组装SQL语句，并将之和数据库连接对象封装在一起，这样在外部对数据表的访问不需要进行任何SQL拼写，仅仅需要直接调用数据访问对象中的CURD方法操作数据表。通过这样的封装，不但简化了对于数据库连接对象的调用，也限制了对数据表的访问，不会出现对设计之外数据表的访问，同时数据访问对象还会对SQL语句进行预处理，以防止可能出现的SQL注入等情况。

## 4.3.调度模块

如同第三节中所描述的，从文档中抓取到的链接将被放入URL队列中。URL队列是一种存放了所有待下载URL的数据结构。大多数爬虫都是根据广度优先搜索的顺序，从种子URL集合开始遍历网络的。这种遍历方式可以由一个先入先出队列实现。

## 4.4.文档抓取模块

# 5.