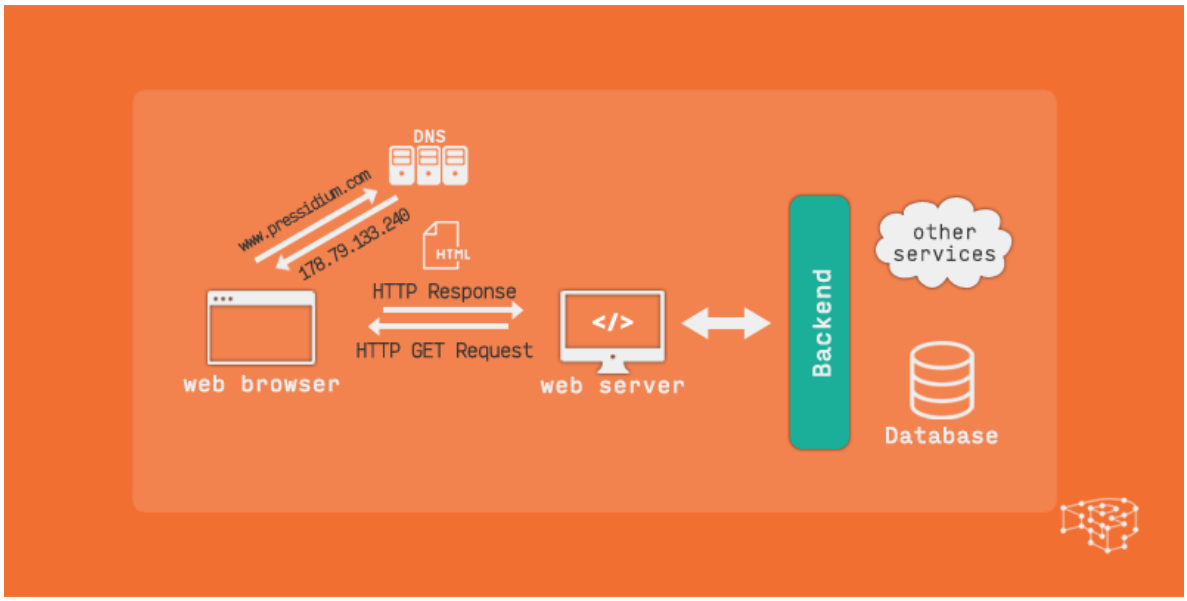
Python 爬虫入门

饶淙元 2020年4月8日

1. 网络请求

1.1 茫茫人海君何在

（浏览器视角下的网络请求）



- 域名解析：由 DNS (Domain Name Server) 完成，如果被劫持则会找到错误的结果
- 请求发送：将信息直接传到目标服务器上，由服务器决定处理与返回的结果
- 响应处理：浏览器渲染结果并呈现给你，随后用户为所欲为（例：浏览器页面修改）

1.2 终极哲学三大问

（服务器视角下的网络请求）

- 你是谁：User-Agent | Cookies
- 你从哪里来：IP
- 你来这里干什么：Method & Parameter

在这一过程中服务器可能面临一些攻击（爬虫可能无意间导致这样的问题）

- DoS (Denial of Service) attack：拒绝服务攻击，让服务器无法正常服务，可以通过构造大量的请求使服务器“忙不过来”完成
- DDoS (Distributed Denial of Service) attack：分布式拒绝服务攻击，在 DoS 攻击基础上改为多个客户端（用户）同时对目标网站发起大量请求

尽管谈论攻击有点遥远，但是不妨考虑以下两个情景：

1. 当你准时到达查询高考成绩时
2. 当你双十一午夜十二点下单时

写爬虫时应该全力避免这种情况！

1.3 网络抓包基本功

> 浏览网页的本质是基于 HTTP 协议的一次或多次网络请求，这些请求用户是可以直接查看的，比如按下 F12 再看看“网络”。
>
> ——某十八线主播

网络抓包通常有如下用处：

1. 查看真正的数据来源（常为 XHR）
2. 查看下载视频等数据
3. 模拟请求完成登录
4. ~~测试网站安全性~~

譬如，如果你想下载直播回放的话，只需要有一款现代浏览器与 ffmpeg就足以应付很多情况。比如你想下载一个装机教程的话……

友情链接：个人计算机主要硬件介绍(计算未来云讲坛：技术视界)

2. 爬虫初步

2.1 何为爬

> 网络爬虫（又称为网页蜘蛛，网络机器人，在FOAF社区中间，更经常的称为网页追逐者），是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。
>
> ——百度百科《爬虫》

> A **Web crawler**, sometimes called a **spider** or **spiderbot** and often shortened to **crawler**, is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web indexing (*web spidering*).
>
> ——wikipedia: Web crawler

- 初衷：为搜索引擎服务
- 现状：自动化或批量的数据获取

2.2 为何爬

> 社会主义的根本任务是解放和发展生产力。

考虑以下场景：

1. 你非常关注某网站的信息（比如优惠信息、新通知等），你愿意每小时去点开一次吗？
2. 你想从某网站保存几百页、几千条信息，你愿意手动全部保存一遍吗？

2.3 孰可爬

> 不以规矩，不能成方圆
>
> ——《孟子·离娄章句上》

爬之前请务必关注robots.txt，这是网站与爬虫间的“君子协议”，此处规定了什么可爬或什么不可爬，下面是几个例子。

2.3.1 百度

https://www.baidu.com/robots.txt

```
User-agent: Baiduspider
Disallow: /baidu
Disallow: /s?
Disallow: /ulink?
Disallow: /link?
Disallow: /home/news/data/
Disallow: /bh

User-agent: Googlebot
Disallow: /baidu
Disallow: /s?
Disallow: /shifen/
Disallow: /homepage/
Disallow: /cpro
```

```
Disallow: /ulink?
Disallow: /link?
Disallow: /home/news/data/
Disallow: /bh
----------------------------此处省略若干行以节省版面----------------------------------

User-agent: *
Disallow: /
```

解读：“除了我钦点的爬虫之外，其他拒不接待”

2.3.2 淘宝 / 天猫

https://market.m.taobao.com/robots.txt

https://list.tmall.com/robots.txt

> ```
> User-agent: *
> Disallow: /
> ```

解读：“莫挨老子”

2.3.3 虎扑

https://www.hupu.com/robots.txt

```
User-agent: *
Allow: /

Sitemap: https://bbs.hupu.com/sitemap_index.xml
Sitemap: https://bbs.hupu.com/sitemap/sitemap_boards.xml
Sitemap: https://voice.hupu.com/sitemap_index.xml
Sitemap: https://nba.hupu.com/players/index.xml
```

解读：www.hupu.com 域名下的都不能爬

https://bbs.hupu.com/robots.txt

```
User-agent: *
Request-rate: 50/1
Disallow: /api/
Disallow: /ajax/
Disallow: /profile.php?*
Disallow: /hack/
Disallow: /template/
Disallow: /attachment/
Disallow: /gearfeedback/
Disallow: /*_*.html$

Sitemap: https://bbs.hupu.com/sitemap_index.xml
Sitemap: https://bbs.hupu.com/sitemap/sitemap_boards.xml
```

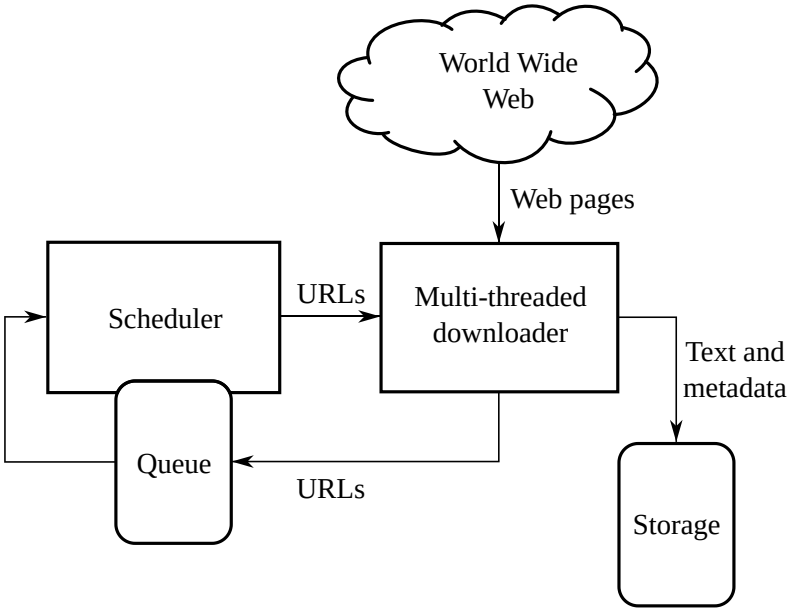
解读：法无禁止皆可为，但是一秒钟最多 50 次

注：`/*_*.html$` 只限制了 `.html` 带 `_` 的，而对一般的 bbs.hupu.com 的具体 bbs 页面这部分是纯数字的，因此是可以爬取的。

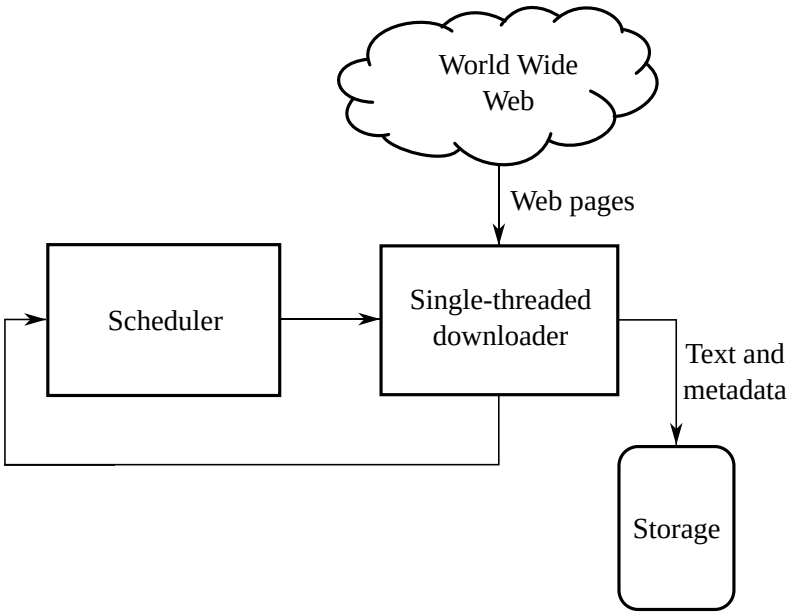
2.4 如何爬

下面给出爬虫的理论架构：

理想中的高性能爬虫架构



直播用的入门版爬虫架构



3. 爬虫实战

3.1 准备

> 工欲善其事，必先利其器。
>
> ——《论语·卫灵公》

1. Python3：编程语言，推荐使用 3.7 或 3.8（当前的稳定新版）
2. Pycharm：强大的Python IDE ，或者使用其他替代品也可，但不建议用 Windows 自带的 idle
3. requests：一个用于发起请求的 Python 库
4. BeautifulSoup4：一个用于解析 html 的 Python 库

3.2 自学内容

上面只说到了比较重要的一些部分，下面还有些可能会用到的自学内容：

- Python、Pycharm 下载安装与环境配置
- 娴熟的搜索引擎使用技巧
- 文件编码处理
- `js2py`：在 Python 中执行 Javascript 脚本
- `pyppeteer`：通过 headless 的方法直接渲染网页
- `scrapy`：真正的高性能爬虫架构

3.3 爬取演示

不失一般性，爬取下面几个省的卫健委新闻作为示范：

- 陕西省：http://sxwjw.shaanxi.gov.cn/（直接用 BeautifulSoup 解析）
- 北京市：http://wjw.beijing.gov.cn/ (需要设置 User-Agent，否则会被封 IP)
- 贵州省：http://www.gzhfpc.gov.cn/ （正则匹配 JavsScript 内容）

~~夹带私货：~~一个两个月前能用的卫健委爬虫集合 https://github.com/rcy17/wjw_scrawler ~~（现在有多少能用不好说了）~~