### 爬虫入门知识整理

### 语言选择

都可以用什么语言写爬虫呢？

1. C++

优点：高效率，快速，适合通用搜索引擎做全网爬取。

缺点：开发慢，写起来又臭又长，例如：天网搜索源代码。

搜索引擎无一例外使用C\C++ 开发爬虫，猜想搜索引擎爬虫采集的网站数量巨大，对页面的解析要求不高，部分支持javascript。

脚本语言：Perl、Python、Java、Ruby

优点：简单，易学，良好的文本处理能方便网页内容的细致提取。

缺点：但效率往往不高，适合对少量网站的聚焦爬取。

node.js

随着Node.js的出现，JavaScript已经发展成为一种非常强大的网页抓取语言。Node.js是无需浏览器即可运行JavaScript代码的引擎。使用JavaScript和Node.js进行网页抓取不仅简单，而且速度很快，而且对于已经熟悉JavaScript的人来说，学习曲线非常低。

自带V8 引擎，乍一看可以推平所有带js 的页面，node本身是为了js 实现后台来设计的，对于信息采集，node和python 和java 没有本质区别，js 引擎和浏览器引擎并不是一个东西，倒是有牛人用开源浏览器实现了一套浏览器采集程序。hrome的无头浏览器出来了，遇到复杂的js 页面可以少折腾。

爬虫目前主要开发语言为java、python、c++

有些公司也用go语言（杭州某互联网金融公司）

对于一般的信息采集需要，各种语言差别不大。

### java与python比较

1. 优劣

Python

网络功能强大，模拟登陆、解析javascript，短处是网页解析。

python写起程序来真的很便捷，著名的python爬虫有**scrapy**等。

python开发爬虫非常灵活，最适合一次性采集数据，写几行代码就可以把初步结构化的数据入库。urllib、requests、beautifulsoup等模块可以轻松构建一个基本的采集程序。

抓取的网页通常需要处理，比如过滤html标签，提取文本等。python的beautifulsoap提供了简洁的文档处理功能，能用极短的代码完成大部分文档的处理。其实以上功能很多语言和工具都能做，但是用python能够干得最快、最干净。

使用python时会遇到中文乱码问题，window开发 Linux 部署乱码问题会让你感觉更棘手，最好统一开发部署环境。

python遇到复杂的js时表现和java差不多，你都会想自己实现一个浏览器，phantomjs 不够稳定，中介再加上一层selenium 写写demo还可以。

Java

java有很多解析器，对网页的解析支持很好，缺点是网络部分

java开源爬虫非常多，著名的如 **nutch** 国内有**webmagic。**

java优秀的解析器有htmlparser、jsoup。

总结

python的requests库比java的jsoup简单；

python的scrapy框架比较强大；

python对excel的支持性比java好；

java没有pip这样的包管理工具。

对于一般性的需求无论java还是python都可以胜任。

如需要模拟登陆、对抗防采集选择python更方便些。抓取网页有时候需要模拟浏览器的行为，很多网站对于生硬的爬虫抓取都是封杀的。这是我们需要模拟user agent的行为构造合适的请求，譬如模拟用户登陆、模拟session/cookie的存储和设置。在python里都有非常优秀的第三方包帮你搞定，如Requests，mechanize。

如果需要处理复杂的网页，解析网页内容生成结构化数据或者对网页内容精细的解析则可以选择java。

Java实现网络爬虫的代码要比Python多很多，Java的语法比Python严格，所以代码会更加复杂。Java对于爬虫的相关库也有，但是没有Python那么多。

2、常用框架

Java：webmagic

<https://gitee.com/flashsword20/webmagic?utm_source=alading&utm_campaign=list>

Python：scrapy

[https://scrapy.org](https://scrapy.org/)

<http://blog.csdn.net/sunnyxiaohu/article/details/50787430>

### 爬虫系统

网络爬虫按照系统结构和实现技术，大致可以分为以下几种类型：通用网络爬虫、聚焦网络爬虫、增量式网络爬虫、深层网络爬虫。 实际的网络爬虫系统通常是几种爬虫技术相结合实现的。

构建网络爬虫首先要了解选择的编程语言的知识，还需要了解网页的工作原理。为了提取所需的数据，我们还需要对CSS选择器有一个很好的了解。有些库可以使用XPATH选择器，但是对初学者来说，CSS选择器更容易学习。

对于一个爬虫系统来说，下载和内文解析只是基本的两个功能。真正好的系统还包括完善的任务调度、监控、存储、页面数据保存和更新逻辑、排重等等。爬虫是一个耗费带宽的应用，好的设计会节约大量的带宽和服务器资源，并且好坏差距很大。

传统爬虫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。

python抓取数据分四步

确定目标：确定抓取哪个网站的哪些页面的哪部分数据。比如是：抓取百度百科词条标题及链接。

分析目标：分析要抓取的url的格式，限定抓取范围，分析要抓取的数据的格式。要分析标题和简介这两个数据所在的标签的格式。分析要抓取的页面编码的格式，在网页解析器部分，要指定网页编码，然后才能进行正确的解析。

编写代码：在网页解析器部分，要使用到分析目标得到的结果。

执行爬虫：进行数据抓取。

网络爬虫的基本工作流程如下：

1. 首先选取一部分精心挑选的种子URL；
2. 将这些URL放入待抓取URL队列；
3. 从待抓取URL队列中取出待抓取在URL，解析DNS，并且得到主机的ip，并将URL对应的网页下载下来，存储进已下载网页库中。此外，将这些URL放进已抓取URL队列。
4. 分析已抓取URL队列中的URL，分析其中的其他URL，并且将URL放入待抓取URL队列，从而进入下一个循环。

如果需要大规模网页抓取，你需要学习分布式爬虫的概念，主要学会怎样维护一个所有集群机器能够有效分享的分布式队列就好。最简单的实现是python-rq: <https://github.com/nvie/rq>

rq和Scrapy的结合：darkrho/scrapy-redis · GitHub

后续处理，网页析取(grangier/python-goose · GitHub)，存储(Mongodb)

### 爬虫开源项目

## **[spider-flow](https://gitee.com/ssssssss-team/spider-flow" \o "spider-flow" \t "https://gitee.com/ssssssss-team/spider-flow?utm_source=alading&utm_campaign=list" \l "https://gitee.com/ssssssss-team/)** 平台以流程图的方式定义爬虫,是一个高度灵活可配置的爬虫平台，不写代码即可完成爬虫。

[https://www.spiderflow.org](https://www.spiderflow.org/)

[https://gitee.com/ssssssss-team/spider-flow?utm\_source=alading&utm\_campaign=list#https://gitee.com/ssssssss-team/spider-flow-oss](https://gitee.com/ssssssss-team/spider-flow?utm_source=alading&utm_campaign=list" \l "https://gitee.com/ssssssss-team/spider-flow-oss)

## **[Spiderman2](https://gitee.com/l-weiwei/Spiderman2" \o "Spiderman2" \t "https://gitee.com/l-weiwei/Spiderman2/tree/)**

<https://gitee.com/l-weiwei/Spiderman2/tree/master>

### 其它

大家最主要使用的方法，包括：第三方的爬虫客户端、第三方的爬虫平台、开源的爬虫以及自己使用Python编写的爬虫。

爬取的网站也在改变，需要随时更新爬虫规则。

一般网站都会设置反爬机制，因此一般会使用**代理ip**绕过反爬机制。

爬虫工具，如八爪鱼采集器，后羿采集器，webscraper，迷你派采集器，简数云采集器等。

搜索引擎，都使用爬虫提供最新的数据，它主要用于提供它访问过页面的一个副本，然后，搜索引擎就可以对得到的页面进行索引，以提供快速的访问。蜘蛛也可以在web上用来自动执行一些任务，例如检查链接，确认html代码；也可以用来抓取网页上某种特定类型信息，例如抓取电子邮件地址(通常用于垃圾邮件)。

Selenium是一个用于Web应用程序测试的工具。Selenium测试直接运行在浏览器中，就像真正的用户在操作一样。用selenium自动化框架来操作浏览器的时候，需要浏览器驱动，如谷歌浏览器chromedriver.exe。