法律声明

- 本课件包括：演示文稿，示例，代码，题库，视频和声音等，小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：大数据分析挖掘
  - 新浪微博：ChinaHadoop

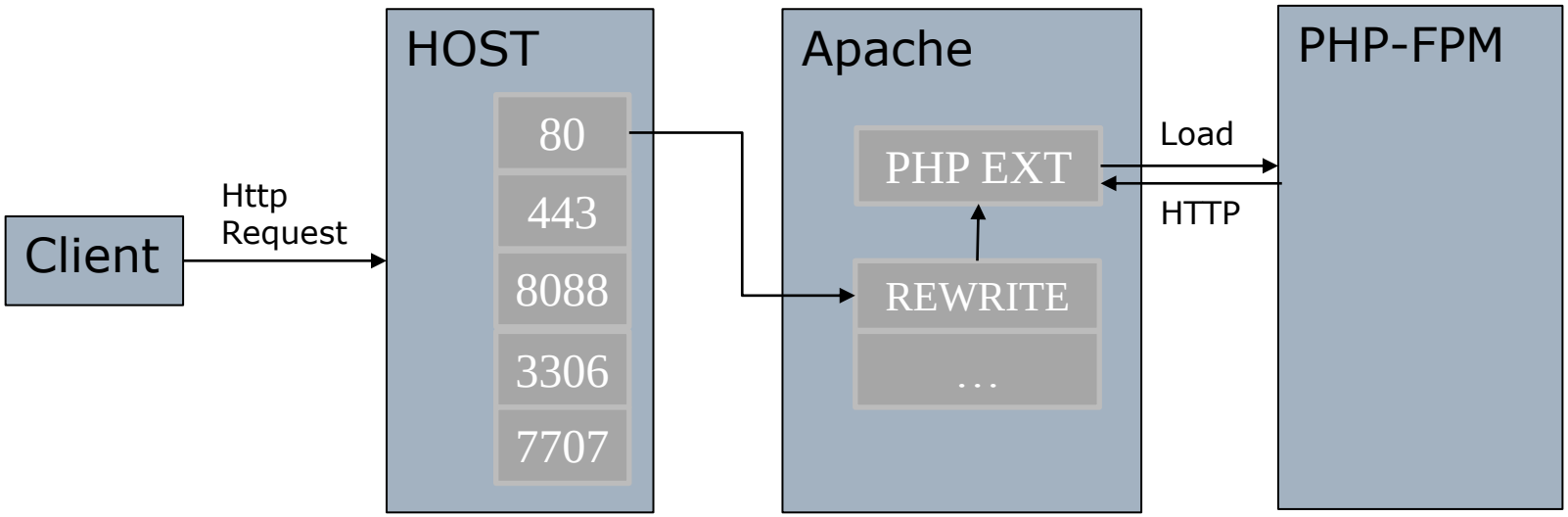
分布式爬虫

大纲

- Web 服务器
- 网站的反爬虫技术
- 应对网站的反爬策略
- 表单及登录

Web 服务器

服务器处理 Web 请求


HOST
80
443
8088
3306
7707
Apache
PHP EXT
REWRITE
...
PHP-FPM
Load
HTTP
Client
Http
Request

服务器处理Web请求流程

1. 到达防火墙，对访问频次进行检查
2. 根据端口映射，到达对应的服务，例如Apache
3. 到达 Apache，通过 virtual host 查找根目录
4. 查找 .htaccess 伪静态设置，映射实际目录及文件
5. 执行脚本或提取文件
6. 确认 cookie 信息，查找用户
7. 用户权限检查
8. 执行命令，返回数据

服务器处理 Web 请求

- 伪静态：实际的文件系统结构与网络请求路径分离
- 路由过程：通常第一步是身份及权限检查
- 设置 Cookie 及 SESSIONID，保存SESSION

```
AddDefaultCharset UTF-8

<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteCond %{SERVER_PORT} !^443$
    RewriteRule ^(.*) https://$(SERVER_NAME)/public/$1 [L,R]
    RewriteRule ^api/$ api/     [L]
    RewriteRule ^api/(.*) api/$1 [L]
    RewriteRule ^$ public/      [L]
    RewriteRule /img/(.*) upload/img/$1 [L]
    RewriteRule (.*) public/$1 [L]
</IfModule>
```

Virtual Host

```
<VirtualHost *:80>
    DocumentRoot "/Users/zhen/Sites/silkroad"
    ServerName silkroad.lo
    <Directory />
        Options FollowSymLinks Multiviews
        MultiviewsMatch Any
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog "/private/var/log/apache2/silkroad.com-error_log"
    CustomLog "/private/var/log/apache2/silkroad.com-access_log" common
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "/Users/zhen/Sites/"
    ServerName jc.lo
    <Directory />
        Options FollowSymLinks Multiviews
        MultiviewsMatch Any
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog "/private/var/log/apache2/jc.lo-error_log"
    CustomLog "/private/var/log/apache2/jc.lo-access_log" common
</VirtualHost>
```

Virtual Host

定义

- 一台服务器主机可以处理多个域名或IP
- 将不同的域名及Ip映射到不同的网站根目录

对爬虫的影响

- 不同的域名指向同一个服务器，更容易被服务器识别为爬虫（异常访问）而禁止
- 对于一个域名映射多台服务器的网站，在单机上不能并发抓取不同服务器的数据

网站如何发现爬虫？

- 单一IP非常规的访问频次
- 单一IP非常规的数据流量
- 大量重复简单的网站浏览行为
- 只下载网页，没有后续的js、css请求
- 通过一些陷阱来发现爬虫，例如一些通过CSS对用户隐藏的链接，只有爬虫才会访问

网站如何进行反爬 - UserAgent

假设我们已经确定了两类爬虫的UserAgent，它们分别是 Nutch 和 Scrapy

User-Agent: Nutch

User-Agent: Scrapy

对应反爬虫的 .htaccess 文件定义：

```
BrowserMatchNoCase Nutch bad_bot
BrowserMatchNoCase Scrapy bad_bot
Order Deny,Allow
Deny from env=bad_bot
```

Nutch Scrapy 为禁止的key，bad_bot 是定义的环境变量

网站如何进行反爬 - .htaccess

1. Check access_log file

Apache: /var/log/httpd/

```
52.78.[redacted] - - [13/Feb/2017:03:22:03 +0000] "GET http://[redacted]/ HTTP/1.1" 200 836
52.78.[redacted] - - [13/Feb/2017:03:22:03 +0000] "GET http://[redacted]/build/common-ef0ce496.js HTTP/1.1" 304 -
52.78.[redacted] - - [13/Feb/2017:03:22:05 +0000] "GET http://[redacted]/build/index-04b31db5.js HTTP/1.1" 304 -
52.78.[redacted] - - [13/Feb/2017:03:22:05 +0000] "GET http://[redacted]/favicon.ico HTTP/1.1" 200 836
52.78.[redacted] - - [13/Feb/2017:03:22:09 +0000] "POST http://[redacted]:9702/api/map/suggestion HTTP/1.1" 200 814
52.78.[redacted] - - [13/Feb/2017:03:22:09 +0000] "POST http://[redacted]:9702/api/map/suggestion HTTP/1.1" 200 85
52.78.[redacted] - - [13/Feb/2017:03:22:10 +0000] "POST http://[redacted]:9702/api/map/suggestion HTTP/1.1" 200 85
```

2. Order Deny, Allow

Deny from xxx.xxx.xxx.xxx

Deny from xxx.xxx.xxx.xxx

网站如何进行反爬 - UserAgent

假设我们已经确定了两类爬虫的UserAgent，它们分别是 Nutch 和 Scrapy

User-Agent: Nutch

User-Agent: Scrapy

对应反爬虫的 .htaccess 文件定义：

```
BrowserMatchNoCase Nutch bad_bot
BrowserMatchNoCase Scrapy bad_bot
Order Deny,Allow
Deny from env=bad_bot
```

Nutch Scrapy 为禁止的key，bad_bot 是定义的环境变量

网站如何进行反爬 - Javascript

大量使用动态网页，使得爬虫的爬取难度增加，常规手段只能拿到一个基本的HTML页面，重要的数据等信息都拿不到

即使爬虫采用了 Web 环境来渲染网页，也会大大增加爬虫的负担与爬取的时间

同时，采用动态加载数据，对服务器的负担也会大大减轻

基于流量的拒绝

开启带宽限制模块

LoadModule bw_module /usr/lib/apache/mod_bw.so

设置访问的最大带宽，每个Ip最多3个连接，最大1MB/s

```
BandWidthModule On
ForceBandWidthModule On
BandWidth all 1024000
MinBandwidth all -1
MaxConnection all 3
#<Location /modbw>
#    SetHandler modbw-handler
#</Location>
```

基于IP连接的拒绝

some_folder 目录 每个Ip最多1个同时的连接，不限制 image/* 目录的文件

```
[codesyntax lang=”apache”]
<Location /some_folder>
MaxConnPerIP 1
NoIPLimit image/*
</Location>
```

/home/*/public_html 目录限制仅限1个Ip访问，但是仅针对 audio/mpeg 格式文件

```
[codesyntax lang=”apache”]
<Directory /home/*/public_html>
MaxConnPerIP 1
OnlyIPLimit audio/mpeg video
</Directory>
```

iptables 的控制

Syntax:

/sbin/iptables -A INPUT -p tcp --syn --dport $port -m connlimit --connlimit-above N -j REJECT --reject-with tcp-reset

Example: Limit HTTP Connections Per IP / Host

/sbin/iptables -A INPUT -p tcp --syn --dport 80 -m connlimit --connlimit-above 20 -j REJECT --reject-with tcp-reset

Example: Limit HTTP Connections Per IP / Host Per Second

iptables -A INPUT -m state --state RELATED,ESTABLISHED -m limit --limit 10/second --limit-burst 20 -j ACCEPT

当收到20个数据包后，触发访问频次限制，此时每秒最多10次连接请求，即单位时间为 100ms；如果100ms内没有收到请求，系统触发的条件就+1，也就是说如果1s钟内停止发送数据，则再次建立10次连接之内都不会激活单位计数限制

如何发现自己可能被网站识别了？

- CAPTCHA pages
- Unusual content delivery delay
- Frequent response with HTTP 404, 301 or 50x errors

  301 Moved Temporarily

  401 Unauthorized

  403 Forbidden

  404 Not Found

  408 Request Timeout

  429 Too Many Requests

  503 Service Unavailable

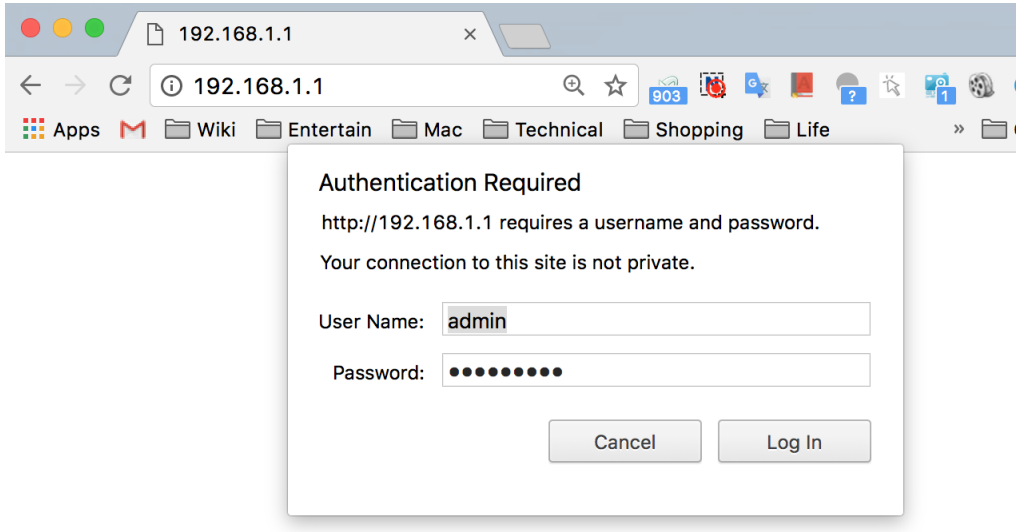
动态 IP 切换技术

1. 模拟路由器登录
2. 通过POSTMAN查找路由器的断线、重连接口
3. 当发现IP被主机识别后，强制路由器断线、重连

路由器登录

1. 普通表单：直接通过表单发送FORM请求
2. 401 Unauthorized：

没有登录的情况下，服务器返回401，并在Header里设置 WWW-Authenticate: Basic realm="User Visible Realm"，客户端收到 401 同时检查到Header里包含了以上信息的情况下，会自动弹出登录框要求用户登录



路由器登录

2种实现自动登录的方法：

1. url: usename:password@ip

   e,g: admin:rootadmin@192.168.1.1

2. set **Authorization** field in Header:

   Key: Authorization

   Value: ‘Basic ‘ + base64.b64encode(‘username:password’)

   Authorization: Basic YWRtaW46cm9vdGFkbWlu

路由器重连


路由器信息
NETGEAR Router R7000
192.168.1.1/RST_st_poe.htm
连接状态
连接时间 05:34:11
连接状态 已连接
协商 成功连接
认证 成功连接
IP地址 115.171.230.156
子网掩码
连接
断开连接
关闭窗口
因特网端口
MAC 地址 C4:04:15:21:7E:C1
IP地址 115.171.230.156
连接 PPPoE
子网掩码 255.255.255.255
域名服务器 219.141.140.10
219.141.136.10
显示统计信息
连接状态
1. 点击链接状态
2. 在弹出的小窗口邮件点击
4. 点击断开连接
Forward
Reload
Save As...
Print...
Cast...
Translate to English
AdBlock
Nimbus Screenshot Screen Video Recorder
View Page Source
Inspect
3. 选择 Inspect
5. 在右侧检查窗口的Network选项下，查看断开连接的API接口及参数
无线接入点 开
广播名称 开
Wi-Fi保护设置(WPS) 已设置
访客网络 (2.4 GHz)
无线网络标识(SSID) NETGEAR-Guest
无线接入点 关

nofollow 属性

"Nofollow" provides a way for webmasters to tell search engines "Don't follow links on this page" or "Don't follow this specific link."
Originally, the nofollow attribute appeared in the page-level meta tag, and instructed search engines not to follow (i.e., crawl) any outgoing links on the page. For example:

```
<meta name="robots" content="nofollow" />
```

css 的 display 属性

The **display** CSS property specifies the type of rendering box used for an element. In HTML, default display property values are taken from behaviors described in the HTML specifications or from the browser/user default stylesheet. The default value in XML is inline.

display:none 隐藏当前块

检测display属性：

document.getElementById('id').style.display

document.getElementsByClassName('classname')[0].style.display

好的规避反爬虫检查的方法

1. 多主机策略
2. 爬的慢一点，不要攻击主机，如果发现被阻止，立即降低访问频次，设计得smart一点，来找到访问频次限制的临界点
3. 通过变换 IP 或者 代理服务器 来掩饰
4. 把爬虫放到访问频繁的主站IP的子网下，例如教育网
5. 频繁改变自己的 User-Agent
6. 探测陷阱，比如 nofollow 的 tag，或者 display:none 的 CSS
7. 如果使用了规则(pattern)来批量爬取，需要对规则进行组合
8. 如果可能，按照 Robots.txt 定义的行为去文明抓取

表单及登录

HTML 提交数据

- form 表单

```
<form>
        <input type="text" name=”username">
</form>
```

HTML 的标签，由浏览器实现POST方法

- ajax 请求

```
$.(ajax){
}
```

基于ajax技术，异步发送http请求并获得返回数据，然后利用JavaScript对网页进行处理

HTML 表单

<form></form> tag 包围

里面包含了表单的元素：

<select> <textarea> <input> 等

<input> type 包含了多种输入类型，例如：

- <input type="text" name="name"> 文本输入框
- <input type="password" name="password"> 密码输入框
- <input type="submit" value="Submit"> 提交按钮

表单类型: form-data

POST /start.htm HTTP/1.1

Host: 192.168.1.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Language: zh-CN,en-US;q=0.8,en;q=0.6

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

Cache-Control: no-cache

------WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="name”

chacha

------WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="password”

chinahadoop

------WebKitFormBoundary7MA4YWxkTrZu0gW--

表单类型

- **form-data**

  http请求中的**multipart/form-data**，它会将表单的数据处理为一条消息，以标签为单元，用分隔符分开。既可以上传键值对，也可以上传文件。当上传的字段是文件时，会有Content-Type来表名文件类型。 由于有boundary隔离，所以multipart/form-data既可以上传文件，也可以上传键值对，它采用了键值对的方式，所以可以上传多个文件。

- **x-www-form-urlencoded**

  application/x-www-from-urlencoded，会将表单内的数据转换为键值对

表单类型: form-data

POST /start.htm HTTP/1.1

Host: 192.168.1.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Accept-Language: zh-CN,en-US;q=0.8,en;q=0.6

Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW

Cache-Control: no-cache

------WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="name”

chacha

------WebKitFormBoundary7MA4YWxkTrZu0gW

Content-Disposition: form-data; name="password”

chinahadoop

------WebKitFormBoundary7MA4YWxkTrZu0gW--

表单类型: x-www-form-urlencoded

POST /start.htm HTTP/1.1

Host: 192.168.1.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

'content-type': "application/x-www-form-urlencoded",

Cache-Control: no-cache

name=chacha&password=chinahadoop

ajax 登录

一般以 json 的方式提交数据

POST /start.htm HTTP/1.1

Host: 192.168.1.1

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Cache-Control: no-cache

{"name":"chacha", "password":"chinahadoop"}

登录

1. 根据 chrome inspector 里检查到的参数，来设置登录方式
   - 最常用的是 x-www-form-urlencoded 和 json 方式，两种方式只是body编码不同
   - 如果是 form-data，按照form-data的方式组合body
   - body 部分经常需要按照特定的方式编码，比如 base64，或者 md5
   - 对于非常复杂的登录协议，利用第5章的动态网页方式登录
2. request = urllib2.Request(url, data, headers = loginheaders)
   - url 是访问的地址
   - data 是 body 部分
   - headers = loginheaders 是HTTP请求的 HEADER 数据

表单登录

```
import urllib2

url = http://www.newsmth.net/nForum/user/ajax_login.json
headers = {
'accept': "application/json, text/javascript, */*; q=0.01",
'origin': "http://www.newsmth.net",
'content-type': "application/x-www-form-urlencoded",
'referer': "http://www.newsmth.net/nForum/",
'accept-language': "zh-CN,en-US;q=0.8,en;q=0.6",
'cache-control': "no-cache”
}

data = 'id=abc&passwd=skdjzix7sla’
request = urllib2.Request(url, data, headers=headers)
response = urllib2.urlopen(request)
```

获取并设置Cookie

登录的核心是为了获得 Cookie

登录成功后，HEADER 会有设置cookie的相关信息

```
Set-Cookie:main[UTMPKEY]=1381959; path=/; domain=.newsmth.net
Set-Cookie:main[UTMPNUM]=14820; path=/; domain=.newsmth.net
Set-Cookie:main[UTMPUSERID]=abc; path=/; domain=.newsmth.net
```

此时我们需要把服务器返回的 Cookie 信息，写入到我们后续请求的HEADER 的 Cookie 里

疑问

- 问题答疑： http://www.xxwenda.com/
  - 可邀请老师或者其他人回答问题

联系我们

小象学院：互联网新技术在线教育领航者

- 微信公众号：大数据分析挖掘
- 新浪微博：ChinaHadoop