

## 前言

在任何需要分析网站速度在每个阶段耗时的场景下，通过抓包分析报文的方式太繁杂，有这么一款工具，可视化将每个阶段耗时统计出来。

httpstat通过封装curl命令，将整个连接过程每个阶段耗时可视化统计出来，就如README所述："httpstat visualizes `curl(1)` statistics in a way of beauty and clarity."

## 一、安装httpstat

安装方式有多种，根据便捷度选取其中一种即可。

### 1.wget安装

```
wget https://raw.githubusercontent.com/reorx/httpstat/master/httpstat.py
mv httpstat.py /usr/bin/httpstat #移动到环境变量路径
chmod +x /usr/bin/httpstat #添加可执行权限
exec bash #重置当前bash进程，如果是其他shell解释器则替换为相应shell，这一步将刷新httpstat的命令补全
```

### 2.pip安装

如果没有安装pip，则可以先执行这两条命令：

```
curl -O https://bootstrap.pypa.io/get-pip.py
python get-pip.py
```

之后通过pip安装httpstat:

```
pip install httpstat
```

macOS用户可直接通过 `brew install httpstat` 安装。

Windows用户推荐使用 `golang`版本。


## 二、使用说明

前面说过，httpstat是基于curl命令封装的工具，因此支持所有curl命令选项，除了 `-w`、`-D`、`-o`、`-s`、`-S`，这五个参数已经被httpstat所使用。

## 模拟测试

最直接的方式，不加任何参数，测试耗时：

```
httpstat <URL>
```



```
⌘ 01:27:12 ~ httpstat https://blog.linux-code.com
Connected to 192.168.1.82:443 from 192.168.1.12:45300

HTTP/2 200
server: nginx
date: Sun, 02 Oct 2022 17:27:15 GMT
content-type: text/html; charset=UTF-8
vary: Accept-Encoding
x-powered-by: PHP/8.0.8
set-cookie: PHPSESSID=s6i043qbgaijm2p70vtb6snc0d; path=/
expires: Thu, 19 Nov 1981 08:52:00 GMT
cache-control: no-store, no-cache, must-revalidate
pragma: no-cache
link: <https://blog.linux-code.com/wp-json/>; rel="https://api.w.org/"
access-control-allow-origin: *

Body stored in: /tmp/tmp6_gv9xsq

DNS Lookup   TCP Connection   TLS Handshake   Server Processing   Content Transfer
[ 2ms        | 1ms              | 11ms            | 293ms              | 0ms              ]
  namelookup:2ms      connect:3ms      pretransfer:14ms  starttransfer:307ms  total:307ms
```

可以清晰直观看到DNS解析耗时、TCP建联耗时、TCL握手耗时、服务端处理耗时、内容传递以及整体耗时等。

又或者是POST请求：

```
httpstat <URL> -X POST -d 'xxx' -v
```



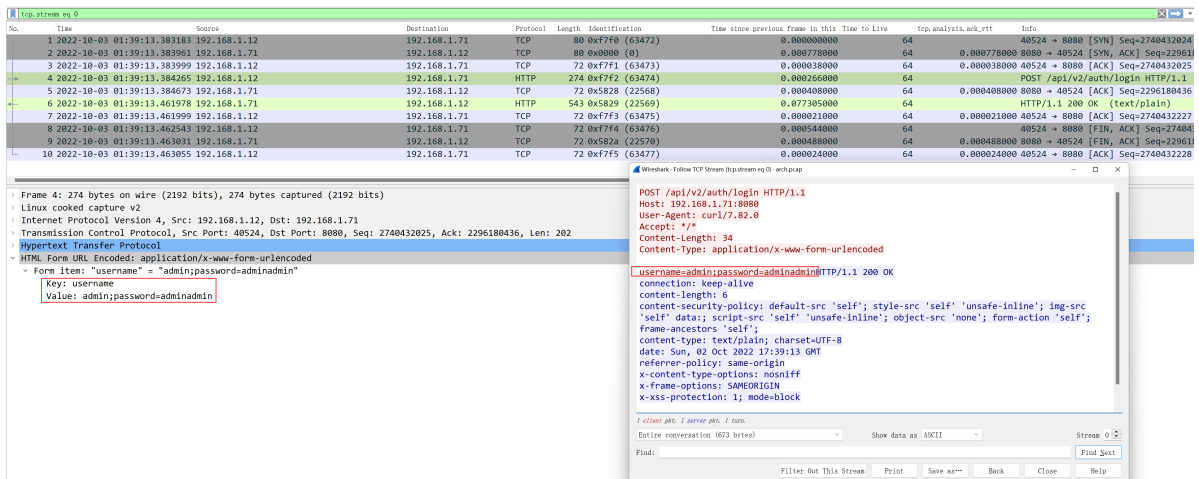
```
⌘ 01:37:05 ~ httpstat http://192.168.1.71:8080/api/v2/auth/login -X POST -d 'username=admin;password=adminadmin' -v
Trying 192.168.1.71:8080...
* Connected to 192.168.1.71 (192.168.1.71) port 8080 (#0)
POST /api/v2/auth/login HTTP/1.1
Host: 192.168.1.71:8080
User-Agent: curl/7.82.0
Accept: */*
Content-Length: 34
Content-Type: application/x-www-form-urlencoded
[34 bytes data]
Mark bundle as not supporting multiuse
HTTP/1.1 200 OK
connection: keep-alive
content-length: 6
content-security-policy: default-src 'self'; style-src 'self' 'unsafe-inline'; img-src 'self' data; script-src 'self' 'unsafe-inline'; object-src 'none'; form-action 'self'; frame-ancestors 'self';
date: Sun, 02 Oct 2022 17:39:13 GMT
referrer-policy: same-origin
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block
[6 bytes data]
Connection #0 to host 192.168.1.71 left intact
Connected to 192.168.1.71:8080 from 192.168.1.12:40524

HTTP/1.1 200 OK
connection: keep-alive
content-length: 6
content-security-policy: default-src 'self'; style-src 'self' 'unsafe-inline'; img-src 'self' data; script-src 'self' 'unsafe-inline'; object-src 'none'; form-action 'self'; frame-ancestors 'self';
content-type: text/plain; charset=UTF-8
date: Sun, 02 Oct 2022 17:39:13 GMT
referrer-policy: same-origin
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block

Body stored in: /tmp/tmp9dzmkzby

DNS Lookup   TCP Connection   Server Processing   Content Transfer
[ 0ms        | 1ms              | 78ms              | 0ms              ]
  nameLookup:0ms      connect:1ms      starttransfer:79ms  total:79ms
```

通过抓包可以清晰看到，-d参数默认将POST字段的内容以application/x-www-form-urlencoded类型传递给服务端。



如果是application/json格式，则需要使用 `-H "Content-Type: application/json"` 指定类型。

再或者，通过 `-k` 参数不校验证书，`-H` 参数指定HOST：

```
httpstat <URL> -Ik -H 'Host:domain'
```

```
01:46:07 httpstat https://192.168.1.1 -Ik -H 'Host:openwrt.linux-code.com'
Connected to 192.168.1.1:443 from 192.168.1.12:44064
HTTP/1.1 302 Found
Connection: Keep-Alive
Keep-Alive: timeout=20
Location: /cgi-bin/luci/
Body stored in: /tmp/tmpqvv73xc
[ DNS Lookup 0ms | TCP Connection 0ms | TLS Handshake 3ms | Server Processing 6ms | Content Transfer 1ms ]
[ namelookup:0ms | connect:0ms | pretransfer:3ms | starttransfer:9ms | total:10ms ]
```

能否灵活运用取决于你对curl命令的熟悉度，同时需要注意的是，选项参数跟随在URL后面，加在前面会报错。

### 三、变量说明

httpstat提供了一些环境变量来控制一些其它功能。

#### 如何设置变量

如果只是单次生效，直接在httpstat前面加变量声明即可，shell会将此变量解析，只在这条命令中单次生效，如：

```
HTTPSTAT_SHOW_BODY=true httpstat https://cloud.tencent.com
```

如果需要在当前终端生效，则需要用到export来申明变量，如：

```
export HTTPSTAT_SHOW_BODY=true
httpstat https://cloud.tencent.com
```

如需取消，通过unset命令来重置：

```
HTTPSTAT_SHOW_BODY
```

如需要永久生效，则将 `export` 的变量赋值写入到 `.bashrc` 或 `.zshrc`，根据shell解释器配置文件走，如：

```
export HTTPSTAT_SHOW_IP=false
export HTTPSTAT_SHOW_SPEED=true
export HTTPSTAT_SAVE_BODY=false
```

## 1.HTTPSTAT\_SHOW\_BODY

默认 `false`。设置为 `true` 将显示响应的body信息，body长度最大1023字节，超过则会被截断。

```
01:55:13 ~ HTTPSTAT_SHOW_BODY=true Wtptat@ https://cloud.tencent.com
Connected to 113.105.154.57:443 from 192.168.1.12:33672

HTTP/2 200
server: nginx
date: Sun, 02 Oct 2022 17:55:45 GMT
content-type: text/html; charset=utf-8
vary: Accept-Encoding
x-request-id: Bk-FsgBwfi
set-cookie: qomainCSRFToken=ryfYjeUDfj; Domain=.cloud.tencent.com; Path=/
set-cookie: intl=; Domain=.cloud.tencent.com; Path=/; Secure
set-cookie: language=zh; Max-Age=15552000; Domain=.cloud.tencent.com; Path=/; Expires=Fri, 31 Mar 2023 17:55:45 GMT
content-length: 173362
x-nws-log-uuid: 13002327699449971836
x-cache-lookup: Cache Miss

<!DOCTYPE html>
<html lang="zh">
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, maximum-scale=1">
  <script>var timeStamp = [new Date()];</script>
  <meta charset="utf-8">
  <title>腾讯云 - 产业智变 云启未来</title>
  <meta name="keywords" content="腾讯云,云服务器,云主机,CDN,对象存储,域名注册备案,云存储,云数据库,互联网+解决方案,QQ云">
  <meta name="description" content="腾讯云为数百万的企业和开发者提供安全稳定的云计算服务，涵盖云服务器、云数据库、云存储、视频与CDN、域名注册等全方位云服务和各行业解决方案。">
  <meta property="og:title" content="腾讯云 - 产业智变 云启未来" />
  <meta property="og:description" content="腾讯云为数百万的企业和开发者提供安全稳定的云计算服务，涵盖云服务器、云数据库、云存储、视频与CDN、域名注册等全方位云服务和各行业解决方案。" />
  <meta property="og:image" content="https://cloudcache.tencent-cloud.com/open_proj/proj_qcloud_v2/gateway/shareicons/cloud.png" />
  <meta property="og:url" content="https://cloud.tencent.com" />
  <meta name="format-detection" content="telephone=no">
  <meta name="baidu-site-verification" content="Yqh6LFzxHs" />
  <meta name="google-site-verification" content="FLzxwYFG80Hk_XLVvz6GpBdsY-f7[... 超过1023字节的部分被截断

Body is truncated (1024 out of 154844), stored in: /tmp/tpaok797pt

DNS Lookup      TCP Connection  TLS Handshake   Server Processing  Content Transfer
[ 15ms | 4ms | 23ms | 24ms | 24ms ]
namelookup:15ms connect:19ms pretransfer:42ms starttransfer:66ms total:90ms
```

## 2.HTTPSTAT\_SHOW\_IP

默认 `true`。显示本地即服务端的IP/PORT信息，设置为 `false` 则不显示：

```

02:05:51 HTTPSTAT_SHOW_IP=false httpstat https://blog.linux-code.com
HTTP/2 200
server: nginx
date: Sun, 02 Oct 2022 18:05:53 GMT
content-type: text/html; charset=UTF-8
vary: Accept-Encoding
x-powered-by: PHP/8.0.8
set-cookie: PHPSESSID=dcjhmihgmhivgofv5ttvhnkv91; path=/
expires: Thu, 19 Nov 1981 08:52:00 GMT
cache-control: no-store, no-cache, must-revalidate
pragma: no-cache
link: <https://blog.linux-code.com/wp-json/>; rel="https://api.w.org/"
access-control-allow-origin: *

Body stored in: /tmp/tmpv5zzwaeu

DNS Lookup | TCP Connection | TLS Handshake | Server Processing | Content Transfer
[ 1ms | 1ms | 10ms | 286ms | 0ms ]
  namelookup: 1ms | connect: 2ms | pretransfer: 12ms | starttransfer: 298ms | total: 298ms

02:05:52 httpstat https://blog.linux-code.com
Connected to 192.168.1.82:443 from 192.168.1.12:45316

HTTP/2 200
server: nginx
date: Sun, 02 Oct 2022 18:06:01 GMT
content-type: text/html; charset=UTF-8
vary: Accept-Encoding
x-powered-by: PHP/8.0.8
set-cookie: PHPSESSID=7qmlrco6bfp0pk2e6qd2ti18u9; path=/
expires: Thu, 19 Nov 1981 08:52:00 GMT
cache-control: no-store, no-cache, must-revalidate
pragma: no-cache
link: <https://blog.linux-code.com/wp-json/>; rel="https://api.w.org/"
access-control-allow-origin: *

Body stored in: /tmp/tmpely95iew

DNS Lookup | TCP Connection | TLS Handshake | Server Processing | Content Transfer
[ 2ms | 0ms | 8ms | 273ms | 1ms ]
  namelookup: 2ms | connect: 2ms | pretransfer: 10ms | starttransfer: 283ms | total: 284ms

02:06:01

```

### 3.HTTPSTAT\_SHOW\_SPEED

默认 `false` 。设置为 `true` 将显示上行和下行的速度:

```

02:11:18 export HTTPSTAT_SHOW_SPEED=true
02:11:25 httpstat http://192.168.1.71:8080/api/v2/auth/login -X POST -d 'username=admin;password=adminadmin'
Connected to 192.168.1.71:8080 from 192.168.1.12:40528

HTTP/1.1 200 OK
connection: keep-alive
content-length: 0
content-security-policy: default-src 'self'; style-src 'self' 'unsafe-inline'; img-src 'self' data; script-src 'self' 'unsafe-inline'; object-src 'none'; form-action 'self'; frame-ancestors 'self';
content-type: text/plain; charset=UTF-8
date: Sun, 02 Oct 2022 18:11:30 GMT
referrer-policy: same-origin
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block

Body stored in: /tmp/tmpsau047v8

DNS Lookup | TCP Connection | Server Processing | Content Transfer
[ 0ms | 0ms | 79ms | 0ms ]
  namelookup: 0ms | connect: 0ms | starttransfer: 79ms | total: 79ms

speed_download: 0.1 KiB/s, speed_upload: 0.4 KiB/s
02:11:50

```

## 4.HTTPSTAT\_SAVE\_BODY

默认 `true`。保存body信息到/tmp目录下的临时文件(系统重启后会被清理)，设置为 `false` 将不保存临时文件。

```
^ 02:14:46 * ~ unset HTTPSTAT_SHOW_SPEED
^ 02:14:58 * ~ HTTPSTAT_SAVE_BODY=false httpstat https://console.linux-code.com
Connected to 183.131.200.172:443 from 192.168.1.12:49770

HTTP/2 200
server: marco/2.17
date: Sun, 02 Oct 2022 18:15:47 GMT
content-type: text/html
content-length: 5254
vary: Accept-Encoding
x-source: C/200
last-modified: Wed, 23 Feb 2022 19:05:35 GMT
etag: "6216857f-1486"
accept-ranges: bytes
expires: Sat, 03 Mar 1990 23:33:33 GMT
cache-control: no-cache, no-store, must-revalidate
pragma: no-cache
x-request-id: 5fd6c3a23b6e5dba9840141a653bf8a7
via: S.mix-js-czx1-109, T.109.D, V.mix-js-czx1-113, T.76.D, M.ctn-zj-jha2-075

DNS Lookup | TCP Connection | TLS Handshake | Server Processing | Content Transfer
[ 65ms | 26ms | 49ms | 101ms | 0ms ]
  namelookup:65ms | connect:91ms | pretransfer:140ms | starttransfer:241ms | total:241ms
```

## 5.HTTPSTAT\_CURL\_BIN

指定curl所在的二进制路径，用于指定curl版本的测试，通常因为某些需求场景需要指定curl版本的特定参数用到。

```
^ 02:19:28 * ~ HTTPSTAT_CURL_BIN=/usr/bin/curl httpstat https://vcenter.linux-code.com --http2
Connected to 192.168.1.25:443 from 192.168.1.12:55722

HTTP/2 200
date: Sun, 2 Oct 2022 18:19:44 GMT
content-security-policy: upgrade-insecure-requests
content-type: text/html
strict-transport-security: max-age=31536000
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1
content-length: 3618
x-envoy-upstream-service-time: 1
server: envoy

Body stored in: /tmp/tmp6z6kq7vk

DNS Lookup | TCP Connection | TLS Handshake | Server Processing | Content Transfer
[ 2ms | 1ms | 11ms | 2ms | 0ms ]
  namelookup:2ms | connect:3ms | pretransfer:14ms | starttransfer:16ms | total:16ms
```

## 6.HTTPSTAT\_METRICS\_ONLY

默认 `false`。设置为 `true` 将以json格式输出结果：

```
02:24:36 ~ HTTPSTAT_METRICS_ONLY=true httpstat https://esxi.linux-code.com
{
  "time_namelookup": 2,
  "time_connect": 3,
  "time_appconnect": 15,
  "time_pretransfer": 15,
  "time_redirect": 0,
  "time_starttransfer": 23,
  "time_total": 23,
  "speed_download": 10934,
  "speed_upload": 0,
  "remote_ip": "192.168.1.82",
  "remote_port": "443",
  "local_ip": "192.168.1.12",
  "local_port": "45320",
  "range_dns": 2,
  "range_connection": 1,
  "range_ssl": 12,
  "range_server": 8,
  "range_transfer": 0
}

02:24:44 ~
```

## 7.HTTPSTAT\_DEBUG

默认 `false` 。设置为 `true` 则开启debug模式:

```
02:27:57 ~ HTTPSTAT_SAVE_BODY=false HTTPSTAT_DEBUG=true httpstat https://esxi.linux-code.com
DEBUG:HttpstatEnv:
HTTPSTAT_SHOW_BODY=
HTTPSTAT_SHOW_IP=
HTTPSTAT_SHOW_SPEED=
HTTPSTAT_SAVE_BODY=false
HTTPSTAT_CURL_BIN=
HTTPSTAT_METRICS_ONLY=
HTTPSTAT_DEBUG=true
DEBUG:HttpstatFlags: { 'show.body': false, 'show.ip': true, 'show.speed': false, 'save.body': false, 'curl.bin': 'curl', 'is.debug': true }
DEBUG:HttpstatCmd: [ 'curl', '-w', '{\n"time_namelookup": %time_namelookup,\n"time_connect": %time_connect,\n"time_appconnect": %time_appconnect,\n"time_pretransfer": %time_pretransfer,\n"time_redirect": %time_redirect,\n"time_starttransfer": %time_starttransfer,\n"time_total": %time_total,\n"speed_download": %speed_download,\n"speed_upload": %speed_upload,\n"remote_ip": %remote_ip,\n"remote_port": %remote_port,\n"local_ip": %local_ip,\n"local_port": %local_port,\n"range_dns": %range_dns,\n"range_connection": %range_connection,\n"range_ssl": %range_ssl,\n"range_server": %range_server,\n"range_transfer": %range_transfer\n}', '-o', '/tmp/impch3y6n7t', '-s', 'https://esxi.linux-code.com' ]
DEBUG:HttpstatOut: {
  "time_namelookup": 0.001560,
  "time_connect": 0.002843,
  "time_appconnect": 0.009388,
  "time_pretransfer": 0.009396,
  "time_redirect": 0.000000,
  "time_starttransfer": 0.022950,
  "time_total": 0.023010,
  "speed_download": 11212,
  "speed_upload": 0,
  "remote_ip": "192.168.1.82",
  "remote_port": "443",
  "local_ip": "192.168.1.12",
  "local_port": "45320"
}
connected to 192.168.1.82:443 from 192.168.1.12:45320
DEBUG:Httpstat:rm header file /tmp/impch3y6n7t
HTTP/2 200
server: nginx
date: Sun, 02 Oct 2022 15:27:24 GMT
content-type: text/html
content-length: 250
content-security-policy: upgrade-insecure-requests
strict-transport-security: max-age=31536000
x-content-type-options: nosniff
x-frame-options: DENY
x-xxss-protection: 1
DEBUG:Httpstat:rm body file /tmp/impch3y6n7t
DNS Lookup      | TCP Connection | TLS Handshake | Server Processing | Content Transfer |
1ms             | 2ms            | 7ms           | 13ms              | 1ms              |
namelookup:1ms  | connect:2ms   | pretransfer:9ms | starttransfer:22ms | total:23ms
```

同时也可以清晰看到，httpstat通过curl的-w参数来拿到每个阶段的耗时，最终格式化输出结果。

## 四、总结

httpstat工具用于测试分析HTTP请求各阶段耗时非常直观好用，优化了curl -w的繁琐输出，提升了可读性，对于网络测试场景如虎添翼，一定程度上节省了很大一部分工作量。