

moon http-stress 使用教程

一见 2010 年 10 月 16

1. http-stress 是什么？

http-stress 是一个和 ab (Apache Benchmark) 类似的 HTTP 压力测试工具，是作为 dispatcher 和 http_parser 等测试用的附属应用，具备如下特性：

- 1) 多线程
- 2) 同时支持 IPV4 和 IPV6
- 3) 模拟多用户并发测试
- 4) 可以配置多个不同或相同的 URLs
- 5) 可以同时对同一个域名的不同 IP 进行测试
- 6) 每秒定时输出测试进度：成功和失败个数
- 7) 长连接和短连接测试
- 8) 暂只支持 GET 方法测试

2. 测试结果样本

```
[2010-10-15 21:11:29][0xf35bc6f0][STATE]time: 52 seconds
[2010-10-15 21:11:29][0xf35bc6f0][STATE]total number: 100000
[2010-10-15 21:11:29][0xf35bc6f0][STATE]failure number: 0
[2010-10-15 21:11:29][0xf35bc6f0][STATE]success number: 98040
[2010-10-15 21:11:29][0xf35bc6f0][STATE]percent number: 1923
[2010-10-15 21:11:29][0xf35bc6f0][STATE]bytes sent: 8600000
[2010-10-15 21:11:29][0xf35bc6f0][STATE]bytes received: 32894278
```

以上信息分别为：

耗时（暂只精确到秒，因此建议请求数不要地少）

总请求个数（每个用户的请求数 * 用户数）

成功请求个数

失败请求个数

发送出去的字节数

接收到的字节数

3. 配置文件

需要配置两个文件，分别为 tress.xml 和 route.table，tress.xml 用来配置测试参数，route.table 是 dispatcher 模块需要的路由表。

3.1. route.table

指定消息应当发往哪，为文本文件，由两部分组成：首行必须为路由项个数，其余行的格式必须满足：路由 ID\t 目标 IP\t 目标端口，其中 IP 可以为 IPV4 或 IPV6 或者域名，路由 ID 不可以重复，其它可以重复。

另外，建议尽量避免配置域名，因为域名解析的系统调用是阻塞的，会影响测试的真实性，如果实在需要使用域名，建议在/etc/hosts 中配置和 IP 的映射关系。

route.table 文件示例，注意是以分隔：

10		
1	127.0.0.1	9999
2	127.0.0.1	9999
3	127.0.0.1	9999
4	127.0.0.1	9999
5	127.0.0.1	9999
6	127.0.0.1	9999
7	127.0.0.1	9999
8	127.0.0.1	9999
9	127.0.0.1	9999
10	127.0.0.1	9999

route.table 中的每一行路由项，代表一个用户，所以用户数需要通过它来配置，而不能通过 tress.xml 配置。

路由 ID 取值范围为从 0 到 65535，可以不连续。

mk_table.sh 脚本可以用来自动生成 route.table，运行参数为：

并发数 目标 IP 目标端口

如：sh mk_table.sh 10 127.0.0.1 9999。

3.2. tress.xml

定义各测试参数，示例如下：

```
<?xml version="1.0" encoding="gb2312"?>
<stress>
  <log level="state" />
  <thread number="1" />
  <connect keep_alive="true" />
  <request number="1000" domain_name="127.0.0.1" error_number="100" />
  <urls>
    <url value="/" />
  </urls>
</stress>
```

<log level="state" />：日志级别，除调试外，因为设置为 state 级别

<thread number="1" />: 线程个数

<connect keep_alive="true" />: 是否为长连接

<request number="1000" domain_name="127.0.0.1" error_number="100" />: 分别为每个用户发送的请求个数（注意不是总请求个数）、测试的域名、允许的最大出错个数（超出这个数进程将退出）。

urls: 需要测试的 url 列表

<url value="/" />: 每行一个 URL，要求以 “/” 打头，可以重复。

4. 测试方法

直接运行 http_stress 即可，不带任何参数，但 http_stress 依赖 libutil.so、libsys.so、libnet.so、libdispatcher.so 和 libhttp_parser.so 这几个库文件，因此运行之前需要先设置好 LD_LIBRARY_PATH 环境变量，如：export LD_LIBRARY_PATH=\$MOOON_HOME/lib。

5. 编译步骤

在编译 http-stress 之前，需要依次先编译好基础类库（common-library）和公共组件（common-component）。

请使用 SVN 从 <http://mooon.googlecode.com/svn/trunk/> 上下载源代码。

5.1. 编译基础类库

基础类库（common-library）包含 util、sys、net 和 plugin 等几个库，编译步骤如下：

- 1) 进入基础类库的 src 目录
- 2) 运行 first_once.sh 脚本，如：sh first_once.sh（建议这样用，因为从 SVN 取下的 first_once.sh 没有可执行权限，如果不为它增加可执行权限，直接 ./first_once.sh 将报错）。下面的操作遵循 automake；
- 3) 运行 configure，生成 Makefile，如：./configure --prefix=/usr/local/mooon，其中 --prefix 后面为 mooon 的安装目录，必须指定
- 4) 编译源文件，运行 make 即可
- 5) 安装，运行 make install 即可。基础类库（common-library）编译成功后，就可以开始编译公共组件了。

5.2. 编译公共组件

公共组件（common-component）依赖于基础类库（common-library），在这里只需要编译 dispatcher 和 http-parser 两个公共组件，不要尝试去编译其它组件，因为可能编译通不过。公共组件编译步骤如下：

- 1) 进入公共组件的 src 目录

- 2) 运行 first_once.sh 脚本
- 3) 运行 configure, 注意指定--prefix 参数为 mooon 安装目录
- 4) 进入 dispatcher 目录, 运行 make 编译源文件
- 5) dispatcher 编译成功后, 运行 make install 安装
- 6) 进入 http_parser 目录, 运行 make 编译源文件
- 7) http_parser 编译成功后, 运行 make install 安装。下面可以开始编译 http-stress 了。

5.3. 编译 http-stress

http-stress 隶属 web application, 编译方法相同, 分别如下:

- 1) 进入 web application 的 src 目录
- 2) 运行 first_once.sh 脚本
- 3) 运行 configure, 注意指定--prefix 参数为 mooon 安装目录
- 4) 进入 http_stress 目录, 运行 make 编译源文件
- 5) http_stress 编译成功后, 运行 make install 安装。

假设 MOOON_HOME 代表 mooon 的安装目录, 则\$MOOON_HOME/lib 库文件存放目录, \$MOOON_HOME/bin 为 http_stress 等可执行文件存放目录, 如下所示:

```
mooon/
|-- bin
|   |-- http_stress
|   |-- mk_table.sh
|-- conf
|   |-- route.table
|   |-- stress.xml
|-- include
|   |-- sys
|   |-- net
|   |-- plugin
|   |-- util
|   |-- dispatcher
|   |-- http_parser
|-- lib
|   |-- libdispatcher.so
|   |-- libhttp_parser.so
|   |-- libnet.so
|   |-- libsys.so
|   |-- libutil.so
|   |-- libtinyxml.so
|-- log
```

请注意, 要将\$MOOON_HOME/lib 所在路径加入到 LD_LIBRARY_PATH 中, 才可以运行 http_stress, 否则运行时会报.so 找不到错误。

6. 后序

在 moon 中，http-stress 是作为 dispatcher 和 http-parser 的测试工具而诞生的，因而比较粗鲁简单，在易用性和测试输出的数据方面还有很大改进空间，欢迎动手对它进行改进并分享。

如需要任何帮助、或有任何建议、或发现任何问题，请发邮件给 eyjian@qq.com，谢谢！