目录

[修订记录 2](#_Toc422426660)

[SPEEDTEST应用程序设计文档 3](#_Toc422426661)

[功能简介 3](#_Toc422426662)

[设计原理 3](#_Toc422426663)

[1、 主程序流程。 3](#_Toc422426664)

[2、 关键的数据结构 3](#_Toc422426665)

[3、 URL接口 4](#_Toc422426666)

[其他说明 5](#_Toc422426667)

## **修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本V0.1 | 首版 | 时间2015.06.18 | 周涛 |
| 版本 | 修改点 | 修改时间 | 修改人 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## SPEEDTEST应用程序设计文档

(V0.1)

## 功能简介

1. 将github上利用speedtest测速的开源python脚本转化为C语言版本。

<https://github.com/sivel/speedtest-cli>

## 设计原理

1. [www.speedtest.net](http://www.speedtest.net)是全球著名的宽带网络速率测试网站，在全球有数百个测试节点，国内目前有大约70个测试点。github上的python开源项目，就是抓取到speedtest所使用的网络接口，采用多线程下载或上传来得出测试速率。
2. 依据测试客户端IP地址获取经纬度地理位置，然后获取测试服务器列表，依据经纬度计算出最近的5个服务器，从5个服务器中测试出延迟最低的那1个作为最终的测试服务器。

### 主程序流程。

up\_speed()通过多线程往服务器上传数据，监视本地的广域网网口总上行速率

主函数main()，初始化curl\_global\_init(CURL\_GLOBAL\_ALL)

get\_best\_server()通过延迟测试接口找出延迟最低的服务器作为最终的服务器。

down\_speed()通过多线程下载服务器上的图片，监视本地的广域网网口总下载速率

get\_config()依据测试客户端的IP地址获取经纬度地理位置

get\_server\_list()获取服务器列表，只选取country为China的服务器

curl\_global\_cleanup()测试结束。

具体的一键加速消息的处理，在最后的do\_custom（）函数中。注意到那个结构体数组type\_lists[]，是一种典型的时间函数编程模式。每次消息到来，遍历匹配类型，然后调用相应的回调函数来处理。这种处理方式是模仿http服务器的。

### 关键的数据结构

放在smartqos.h文件中。

struct client **{**

char ip**[**16**];**

double lat**;**

double lon**;**

void **\***other**;**

**}** cl**;**

记录客户端的ip地址和经纬度信息。

struct server **{**

struct list\_head list**;**

char url**[**256**];**

double lat**;**

double lon**;**

double dis**;** //到测试点的距离

void **\***other**;**

**};**

记录每个服务器节点的测试URL，经纬度信息和到客户端的距离，保存到链表中。

struct best\_server **{**

char url**[**256**];**

char p\_url**[**256**];**

struct timeval start**[**3**];**

struct timeval end**[**3**];**

long int interval**[**3**];** //时间间隔，单位us

long int avg**;** //平均延迟

**};**

最终所用测试服务器的所用到的一些变量结构。

### URL接口

1. 获取客户端端信息

<https://www.speedtest.net/speedtest-config.php>

得到的是一个xml文件。

1. 获取服务器列表

<http://c.speedtest.net/speedtest-servers-static.php>

得到的是一个xml文件。

1. 延迟测试

依据xml文件中url值生成。比如原URL为

<http://speed.westidc.com.cn/speedtest/upload.php>

那么延迟测试URL为

<http://speed.westidc.com.cn/speedtest/latency.txt>

1. 上传测试

xml文件中url值。

1. 下载测试

依据xml文件中的url值生成。比如原URL为

<http://speed.westidc.com.cn/speedtest/upload.php>

那么延迟测试URL为

<http://speed.westidc.com.cn/speedtest/random2000x2000.jpg>

其中2000可替换成[350, 500, 750, 1000, 1500, 2000, 2500, 3000, 3500, 4000]中的值。

## 其他说明

1. 广域网接口速率的监视，依赖于ip\_rate.ko内核模块的proc接口。其具体用法参考ip\_rate.ko模块的设计文档。
2. 延迟测试采用3次访问的平均延迟时间来计算。
3. http客户端采用了libcurl库，其具体用法有一些技巧，具体参考

<http://curl.haxx.se/libcurl/>