



ServerSpeeder Linux 部署手册

北京华夏创新科技有限公司

目录

一、	ServerSpeeder 简介	1
二、	ServerSpeeder 加速原理	2
三、	软件命名格式.....	3
四、	安装步骤.....	3
1.	自动安装	3
2.	采用手动安装方式进行安装	5
五、	使用方法.....	5
1.	目录及文件说明	5
2.	常用配置参数说明	6
3.	运行 ServerSpeeder.....	10
4.	确认 ServerSpeeder 已经运行.....	12

一、 ServerSpeeder 简介

ServerSpeeder 是一款内嵌 [ZetaTCP](#) 加速引擎的加速软件，可以安装运行在所有主流 Linux 和 Windows 操作系统服务器上，加速服务器上传服务的质

量。ServerSpeeder 尤其适合具有较分散互联网用户访问的服务，能够显著提高距离服务器远，跨运营商访问服务器，和通过无线网络访问服务器的用户的体验，甚至将服务从原先的不可用变为可用。ServerSpeeder 内嵌的 ZetaTCP 还可以增强连接的稳定性，显著降低访问失败率。

ServerSpeeder 通过加速用户访问连接，提高连接稳定性，全面提高应用服务质量，提升最终用户体验。

二、 ServerSpeeder 加速原理

ServerSpeeder 安装到服务器上，对于从服务器到访问服务器的用户的流量，会起到最好的加速效果；而对于从用户到服务器的流量，加速效果会不稳定或者没有加速效果（但不会比没安装 ServerSpeeder 之前慢）。

ServerSpeeder 本质是一个实现了 Zeta-TCP TCP 优化技术的软件，相对于传统的 TCP，Zeta-TCP 主要对以下三个方面进行改进：

第一， Zeta-TCP 引入了更精确的丢包判断及预测算法。传统 TCP 经常错误判断丢包。将未丢的数据包判断为丢失将导致错误的重传，从而造成带宽的浪费。反之，如果不能及时判断确实已丢失的数据包，将导致不必要的等待，从而导致带宽空置。两种误判都会降低连接的吞吐率和带宽的利用率。Zeta-TCP 加速引擎能够精确及时地判断丢包，从而保证了最佳的带宽利用率。

第二， Zeta-TCP 随时精确侦测连接路径带宽，并相应调整发送数据量。TCP 协议通过滑动窗口机制对带宽进行自适应。传统 TCP 的滑动窗口实现经常误判路径带宽容量。高估带宽容量将导致过量传输从而引发拥塞并导致大量丢包。低估带宽容量则导致闲置带宽容量。两个极端的出现都将导致带宽利用率的下降。传统 TCP 往往在两个极端之间震荡，很难有效利用带宽。Zeta-TCP 在主动精确侦测路径带宽的基础上，随时调整发送数据量，从而在防止引入拥塞的同时最大限度地利用路径带宽。

第三， Zeta-TCP 监控学习连接对端的传输行为，并根据其发送模式智能反馈引导对端正确判断丢包及准确估算路径带宽，从而大幅提升下载方向带宽利用率。对比于传统 TCP 被动确认及接受数据，Zeta-TCP 反向智能遥控对端的传输行为，实现了业界独一无二的单边下载加速。

除了以上三点主要突破性技术手段，Zeta-TCP 在很多其它方面也对传统 TCP 进行了细致的改进和完善。经过多年在各种真实网络环境中的不断调整改进，单边 TCP 加速具备了广泛的适用性和有效性。

三、 软件命名格式

ServerSpeeder 模块文件名：acce-模块版本号-[Linux 系统版本_内核版本]

ServerSpeeder 授权文件名：apx-过期时间.lic

四、 安装步骤

通过 ServerSpeeder 发布系统，我们为您提供两种安装方式，自动安装和手动安装；两种安装均需由具有 root 权限的用户进行操作；

1. 自动安装

自动安装时，安装程序需要通过互联网访问 ServerSpeeder 发布系统以便对您的安装帐号进行验证，并在验证通过后，在服务器上下载与您的安装账号绑定的 ServerSpeeder 已授权程序；所以采用自动安装方式进行安装时，需要您的服务器能够访问互联网。自动安装可采用交互式安装和非交互式安装两种方式进行操作，具体步骤如下：

交互式安装：

- 1) 下载自动安装文件，
- 2) 复制安装文件到您的服务器；或者在服务器上运行
`wget http://dl.serverspeeder.com/d/ls/serverSpeederInstaller.tar.gz` 下载自动安装文件；
- 3) 解压缩：`tar xzvf serverSpeederInstaller.tar.gz`；
- 4) 为解压出的 `serverSpeederInstaller.sh` 赋予可执行权限：`chmod +x serverSpeederInstaller.sh`；
- 5) 以 root 用户权限运行解压后得到的安装脚本：`./serverSpeederInstaller.sh`；

- 6) 根据屏幕提示输入 ServerSpeeder 安装授权帐号和密码;
- 7) 安装授权验证通过后, 安装程序会自动连接 ServerSpeeder 下载服务器并下载已授权的 ServerSpeeder 程序; 该过程视您的网络状况和服务器繁忙程度的不同, 将持续几秒或几十秒的时间;
- 8) 根据屏幕提示, 输入【加速接口】、【上行带宽】、【下行带宽】等(注: 此设置在安装 ServerSpeeder 后也可更改);
- 9) 根据屏幕提示选择是否立即启动 ServerSpeeder;

非交互式安装:

- 1) 下载自动安装文件;
- 2) 复制安装文件到您的服务器;
- 3) 解压缩: `tar xzvf serverSpeederInstaller.tar.gz`;
- 4) 为解压出的 `serverSpeederInstaller.sh` 赋予可执行权限: `chmod +x serverSpeederInstaller.sh`;
- 5) 以 root 用户权限运行解压后得到的安装脚本:

```
./serverSpeederInstaller.sh -e email -p password [-in inbound_bandwidth] [-out  
outbound_bandwidth] [-i interface] [-r] [-t shortRttMS] [-b] [-f]
```

各参数说明如下:

- e: 您的安装帐号, 一般为 email 地址格式;
- p: 安装帐号的密码;
- in: 入向带宽; 可省略, 默认为 1000 000kbps;
- out: 出向带宽; 可省略, 默认为 1000 000kbps;
- i: 加速接口; 可省略, 默认为 eth0;
- r: 安装完成后是否立即运行 ServerSpeeder; 可省略, 默认为不立即运行

ServerSpeeder:

- t: ShortRtt 数值; 可省略, 默认为 0ms, 不开启 ShortRtt 功能;
- b: 是否随机自动启动 ServerSpeeder; 可省略, 默认为开机不启动 ServerSpeeder;
- f: 是否强制安装 ServerSpeeder, 如果服务器已经安装了 ServerSpeeder, 会卸载后再

安装。

安装完成后, 安装程序会给出相应的安装结果提示, 一般有如下三种情况:

- 1) 安装成功: 安装成功后, 安装程序会在您的服务器文件系统的根目录下

面创建一个/serverSpeeder 目录，下面包含 bin、etc 和 log 三个子目录；其中 bin 目录中存放有 ServerSpeeder 模块文件及执行脚本，etc 目录中存放有包含 ServerSpeeder 授权文件和 ServerSpeeder 配置文件，log 目录中未存放任何文件；

- 2) ServerSpeeder 版本未找到：自动安装程序未找到与您的服务器相匹配的 ServerSpeeder 版本，程序会在给出相应的提示后退出；
- 3) 授权次数达到上限：您所使用的安装账户的授权次数已到，安装程序会在给出相应的提示后退出；

2. 采用手动安装方式进行安装

由于自动安装需要服务器直接连接互联网，如果需要安装 ServerSpeeder 的服务器无法连接互联网，则您可以选择手动安装的方式进行安装；具体安装步骤请参见：

<http://dl.serverspeeder.com/w.do?m=ls1m>

五、 使用方法

1. 目录及文件说明

名称	类型	说明
/serverSpeeder	目录	安装成功后，安装程序会自动在服务器文件系统的根目录下面创建该目录；该目录用于存放与 ServerSpeeder 相关的各文件；
/serverSpeeder/bin/	目录	用于存放 ServerSpeeder 模块文件及执行脚本；
/serverSpeeder/bin/accel-xxx	文件	ServerSpeeder 模块文件；
/serverSpeeder/bin/serverSpeeder.sh	文件	用于启动 ServerSpeeder 的脚本文件；

/serverSpeeder/etc/	目录	用于存放 ServerSpeeder 授权文件及其参数配置文件；
/serverSpeeder/etc/apx-xxx.lic	文件	ServerSpeeder 授权文件；
/serverSpeeder/etc/config	文件	ServerSpeeder 配置文件；
/serverSpeeder/log/	目录	用于存放 ServerSpeeder 运行过程中可能需要抓取的一些用于问题调试的 log 文件；

2. 常用配置参数说明

通过修改/serverSpeeder/etc/config 文件，可以为 ServerSpeeder 设定不同的配置参数；常用参数的说明如下：

1) accif="eth*"

加速接口；eth*一般为服务器上对外提供服务的网络接口，例如 eth0, eth1...，一般通过 ifconfig 命令可以看到；可以同时设定多个接口作为加速接口（接口之间用空格分开），如 accif="eth0 eth1"；默认为"eth0"；

2) acc="1"

TCP 加速开关；设为 1 表示开启 ServerSpeeder 的 TCP 加速功能，设为 0 表示关闭 ServerSpeeder 的 TCP 加速功能；默认为 1；

3) advacc="1"

高级加速开关；设为 1 表示开启，设为 0 表示关闭；开启此功能可以得到更好的加速效果，但有可能会造成有效数据率下降；默认为 1；

4) advinacc="0"

高级入向加速开关；设为 1 表示开启，设为 0 表示关闭；开启此功能可以得到更好的流入方向流量加速效果；

5) wankbps="1000000"

加速接口上行带宽；指的是从服务器流出到 Internet 的最大带宽，单位为 Kbps，例如服务器所连接的 Internet 带宽为 1G，则设置为 1000000；默认为 1000000；

6) waninkbps="1000000"

加速接口下行带宽；指的是从 Internet 流入服务器的最大带宽，单位为 Kbps，例如服务器所连接的 Internet 带宽为 1G，则设置为 1000000；默认为 1000000；

wankbps 和 waninkbps 的设置一般与实际带宽相同为最理想；如果无法确定实际带宽，一般设置为网卡的最大吞吐能力即可；例如网卡为千兆全双工网卡，则 wankbps 和 waninkbps 都设置为 1000000；

当设置过小时，例如实际带宽有 100Mbps，但 wankbps 和 waninkbps 设置为 10Mbps，则数据流量会被限制在 10Mbps 以内，此时会降低服务器的吞吐；

7) csvmode="0" highcsv="0"

csvmode 拥塞控制模式开关；设为 1 表示开启，设为 0 表示关闭；开启 tcp 加速后，在某些丢包较大的网络中可能会造成有效数据率下降的问题；此时开启拥塞控制模式，可以有效避免该问题；但开启此选项时，可能加速效果也会下降；默认为 0；

csvmode="1" highcsv="0"时，为普通拥塞控制模式，对拥塞具有一定控制的同时对加速效果的影响较小；

csvmode="1" highcsv="1"时，为高级拥塞控制模式，可以更好的控制拥塞但是相比于普通拥塞控制模式对加速效果的影响较大。

8) subnetAcc="0"

局域网加速开关；设为 1 表示开启；设为 0 表示关闭；不开启时对于同一局域网内的连接不会加速；开启后对同一网段的 TCP 连接也进行加速；

9) maxmode="0"

最大传输模式；设为 1 表示开启；设为 0 表示关闭；开启后会进一步提高加速效果，但是可能会降低有效数据率；

10) maxTxEffectiveMS=" 0"

最大传输模式生效时间；用于控制最大传输模式对每一个连接的生效时间；设为 0 表示从连接建立到连接结束，都采用最大传输模式传输；单位为 ms，1000 即 1s；

11) pcapEnable= "0"

抓包开关；设为 1 表示开启，设为 0 表示关闭；开启此功能后会抓取经过加速引擎处理的数据包，在/serverSpeeder/log 目录下生成抓包文件；

12) bypassOverFlows="1"

流量 bypass 开关；设为 1 表示开启，设为 0 表示关闭；开启此功能后当服务器的连接数超过引擎设定后会对超过的连接的流量 bypass，否则会无法新建连接；

13) initialCwndWan="22"

初始 TCP 发送窗口能够发送的数据包的数量；该值设置的高会获得更好的加速效果，但是可能会造成网络的拥塞；

14) l2wQLimit="256 2048"

从 LAN 到 WAN 加速引擎在缓冲池充满和空闲时分别能够缓存的数据包队列的长度的上限；该值设置的高会获得更好的加速效果，但是会消耗更多的内存；

15) w2lQLimit="256 2048"

从 WAN 到 LAN 加速引擎在缓冲池充满和空闲时分别能够缓存的数据包队列的长度的上限；该值设置的高会获得更好的加速效果，但是会消耗更多的内存；

16) shrinkPacket="0"

slab 缓存缩减开关；设为 1 表示开启，设为 0 表示关闭；当服务器内存消耗很高时可以将该值设定为 1，这样会降低内存的消耗；

17) retranWaitListMS="32"

重传等待时间；当服务器没有收到 ack 或者丢包产生之后客户端重复 ack 时间达到 32ms，服务器重传数据；默认为 32ms；

18) halfCwndMinSRtt="500"

halfCwndLossRateShift="3"

以上两个值用于判断网络拥塞，分别为延时和丢包率，丢包率默认值为 3，即 $1/2^3$ ，当拥塞产生时，退出第三代 Learning-based TCP 算法，采用类似传统 TCP 的算法。

20) shaperEnable="1"

流量整形开关，配合上行和下行带宽设置开启；设为 1 表示开启，设为 0 表示关闭；开启此功能，会采用 waninkbps 以及 waninkbps 设置的带宽大小运行；关闭此功能，会根据实时传输自动测算带宽大小；

21) tcpOnly="1"

设为 1 表示开启，设为 0 表示关闭；开启此功能，即只处理 TCP 流量；关闭此功能，也处理除 TCP 以外的流量（不会对非 TCP 流量加速），一般用于配置策略对非 TCP 数据丢弃时；

22) SmBurstMS="16"

引擎允许的最大突发时间，设为 0，则关闭；该值越大，数据包发送量越大；当前推荐默认为 16；

23) rsc="0"

网卡接收端合并开关；设为 1 表示开启，设为 0 表示关闭；在有些较新的网卡驱动中，带有 RSC 算法的，需要打开该功能；

24) gso="0"

GSO 开关；设为 1 表示开启，设为 0 表示关闭；当网卡需要开启 gso 功能时，开启此功能，ServerSpeeder 变为支持 gso 的模式；关闭此功能，如果网卡开启 gso，ServerSpeeder 会关闭网卡的 gso 功能；

25) lanSegment=""

当 ServerSpeeder 不是 TCP 连接的最终节点并且采用单臂模式部署时，如果 ServerSpeeder 和服务端之间网络环境很好，不需要加速，将此参数的值设为 Lan 一侧的子网网段（参数的值为子网网段的十六进制表示加子网掩码，例：ip 为 172.33.0.0/16，lanSegment="AC210000/16"），则 ServerSpeeder 和服务端之间不进行加速，可以节省 ServerSpeeder 使用的资源并获得更好的加速效果。如果不设置此参数，则 ServerSpeeder 会对所有经过的 TCP 连接进行加速。（注：如果参数转换为十六进制后，第一位数字为 0 时，需要省略，不然无法正常写入配置。例如：参数十六进制值为：0C210000/16，那么 lanSegment="C210000/16"。）

26) configTimeout="30"

设置开启 ServerSpeeder 时，写入配置所需要的时间。当引擎的数目很多时需要写入配置的时间较长，默认为 30s。

27) engineNum="0"

ServerSpeeder 启动的加速引擎的个数；在多处理器(核)的系统环境下，启用多个加速引擎可以使网络流量的负载在多个处理器(核)之间做均衡分配，从而帮助优化系统 CPU 资源的利用；启用多个加速引擎时，引擎序号从 0 开始，分别为 engine0, engine1, engine2, 等等；每个引擎的负载对应交给相同序号的处理器来处理，即，engine0 使用 cpu0, engine1 使用 cpu1 等；engineNum 默认为

0，表示启用的加速引擎个数与系统的处理器个数相同，64bit 架构下，当实际的 engineNum 参数大于 4 时，检查内存：

当总内存减去 engine 占用内存剩余数小于 2G 时，提示用户一个 warning：

ServerSpeeder Warning: \$CPUNUM engines will be launched according to the config file. Your system's total RAM is \$memTotal(KB), which might be insufficient to run all the engines without performance penalty under extreme network conditions. ；

28) shortRttMS="0"

白名单开关；打开后 ServerSpeeder 将不对 RTT 小于 shortRttMS 的连接加速，节约系统资源，提高性能；设置为 0 时表示关闭此功能，ServerSpeeder 会加速所有的连接；设置为其他值时，例如 10，则表示当某个 24 位子网网段 IP，测量如果 RTT 小于 10ms，则不对其所在的整个 24 位子网进行加速；如果 RTT 大于 10ms，ServerSpeeder 对其所在 24 位子网进行加速，如果该网段的某个 ip 发送的连接小于 10ms，ServerSpeeder 则一直不会再对这个网段进行加速。默认值为 0。

重启 ServerSpeeder 后，原来记录的每个 24 位子网网段的 RTT 会被清除，重新开始记录。

29) pmtu=""

PMTU 开关，如果设为 0，则关闭 ServerSpeeder 的 PMTU 功能，不设值为开启 PMTU 功能。默认为开启。

30) apxexe="/serverSpeeder/bin/acce-*"**

ServerSpeeder 模块文件的路径；

31) apxlic="/serverSpeeder/etc/apx-*.lic"**

ServerSpeeder 授权文件的路径；

3. 运行 ServerSpeeder

使用 /serverSpeeder/bin/ 目录下的 serverSpeeder.sh 脚本进行 ServerSpeeder 的启动，停止，以及重新加载配置等操作；可通过 ./serverSpeeder.sh --help 查看脚本的帮助信息；各参数说明如下：

start: 启动 ServerSpeeder，加载加速模块；使用/serverSpeeder/etc/config 文件中的配置作为模块加载时的初始化参数；

stop: 停止 ServerSpeeder，卸载加速模块；停止 ServerSpeeder 前请确认没有其它进程在访问/proc/net/appex/目录，例如确认控制台当前目录是否是/proc/net/appex/；

reload: 在不停止 ServerSpeeder 运行的情况下实时修改 ServerSpeeder 参数配置，修改/serverSpeeder/etc/config 文件的配置后运行此命令，此时加速模块不退出，参数被实时修改；

restart: 重启 ServerSpeeder；

status: 查看当前 ServerSpeeder 的实时运行状态；License 的信息；连接的信息和 ServerSpeeder 当前的配置；

stats: 实时显示每个加速引擎的连接数、流量，以及所有引擎的总的连接数和流量；其中连接数统计包括网络连接数(sessions)，tcp 连接数(tcp sessions)，已加速的 tcp 连接数(accelerated sessions)以及活动的 tcp 连接数(active tcp sessions)；被 ShortRtt 功能忽略的包的个数（short rtt bypassed packets）；流量统计包括流入(in)流量统计和流出(out)流量统计，单位均为 kbit/s；一般情况下，所有引擎的总连接数及流量即为服务器的互联网连接数及流量；

renewLic: 当有新的 License 文件可用时，联网更新 License 文件并重启 ServerSpeeder；

renew -grace: ServerSpeeder 支持更新 license 文件，不需要重启，程序自动将新 License 读入；

update: 当有新的 ServerSpeeder 版本可用时，联网更新 ServerSpeeder 并重启 ServerSpeeder；

注：

运行 ServerSpeeder 前请务必确认加速接口的 tso (tcp segmentation offload)、gso (generic segmentation offload)、gro(generic receive offload)、lro(large receive offload)和 sg(scatter gather)功能是否处于关闭状态！！！如果在 tso、gso、gro、lro 或 sg 其中之一处于开启状态而 ServerSpeeder 中未开启该功能，此时运行 ServerSpeeder，会关闭这些功能。如果在 ServerSpeeder 运行过程中有其他软件等开启这些功能，会导致

加速模块无法取到数据包，进而导致网络中断。您可通过 `ethtool -k eth*` 来查看加速接口的 `tso`、`gso`、`gro`、`lro` 和 `sg` 的当前状态。

如果您的加速接口的 `tso`、`gso`、`gro`、`lro` 和 `sg` 功能是关闭的，对于 `bonding` 性质的接口，需组成该接口的所有 `eth` 口的 `tso`、`gso`、`gro`、`lro` 和 `sg` 功能是关闭的，那么您无需更改 `ServerSpeeder` 的配置。

如果您的加速接口的 `tso`、`gso`、`gro`、`lro` 和 `sg` 功能其中之一是打开的，可以通过打开 `ServerSpeeder` 的 `gso` 功能(`gso="1"`)，将 `ServerSpeeder` 转换为支持 `tso`、`gso` 和 `sg` 的形式，打开 `ServerSpeeder` 的 `rsc` 功能 (`rsc="1"`)，则将 `ServerSpeeder` 转换为支持 `gro` 和 `lro` 的形式，修改 `ServerSpeeder` 中的 `gso` 和 `rsc` 功能后，需要 `serverSpeeder.sh restart` 让配置生效。

4. 确认 `ServerSpeeder` 已经运行

通过 `serverSpeeder.sh status` 命令来查看 `ServerSpeeder` 当前的运行状态；
通过 `lsmod` 查看是否有 `appex0` 的模块来确认 `ServerSpeeder` 是否在正常运行；

通过查看 `/proc/net/appex/stats` 文件来确认 `ServerSpeeder` 是否在正常运行。

应用流畅 · 网络如飞



北京华夏创新科技有限公司

AppEx Networks Corporation

北京市海淀区北清路 68 号用友软件园北区 16 号 C 座 6 层

邮编：100094

电话：400-00-27739 010-62973737

www.appexnetworks.com.cn