

Optimizing Linux TCP/IP Networking to increase Linux Servers Performance

Author : admin



Some time ago I thought of ways to optimize my Linux Servers network performance.

Even though there are plenty of nice articles on the topic on how to better optimize Linux server performance by tuning up the kernel sysctl (variables).

Many of the articles I found was not structured in enough understandable way so I decided to google around and found few interesting websites which gives a good overview on how one can speed up a bit and decrease overall server loads by simply tuning few basic kernel *sysctl* variables.

Below article is a product of my research on the topic on how to increase my GNU / Linux servers performance which are mostly running LAMP (Linux / Apache / MySQL / PHP) together with Qmail mail servers.

The article is focusing on Networking as networking is usual bottleneck for performance.

Below are the variables I found useful for optimizing the Linux kernel Network stack.

Implementing the variables might reduce your server load or if not decrease server load times and CPU utilization, they would at least increase throughput so more users will be able to access your servers with (hopefully) less interruptions.

That of course would [save you some Hardware costs and raise up your Servers efficiency.](#)

Here are the variables themselves and some good example:

```
# values.net.ipv4.ip_forward = 0 ( Turn off IP Forwarding )

net.ipv4.conf.default.rp_filter = 1

# ( Control Source route verification )
net.ipv4.conf.default.accept_redirects = 0

# ( Disable ICMP redirects )
net.ipv4.conf.all.accept_redirects = 0 ( same as above )
net.ipv4.conf.default.accept_source_route = 0

# ( Disable IP source routing )
net.ipv4.conf.all.accept_source_route = 0
```

(- || -)net.ipv4.tcp_fin_timeout = 40

(Decrease FIN timeout) - Useful on busy/high load

serversnet.ipv4.tcp_keepalive_time = 4000 (keepalive tcp timeout)

net.core.rmem_default = 786426 - Receive memory stack size (a good idea to increase it if your server receives big files)

net.ipv4.tcp_rmem = "4096 87380 4194304"

net.core.wmem_default = 8388608 (Reserved Memory per connection)

net.core.wmem_max = 8388608

net.core.optmem_max = 40960

(maximum amount of option memory buffers)

like a homework investigate by yourself what the variables below stand for :)

net.ipv4.tcp_max_tw_buckets = 360000

net.ipv4.tcp_reordering = 5

net.core.hot_list_length = 256

net.core.netdev_max_backlog = 1024

Below are newly added experimental

#net.core.rmem_max = 16777216

#net.core.wmem_max = 16777216

##kernel.msgmni = 1024

##kernel.sem = 250 256000 32 1024

##vm.swappiness=0

kernel.sched_migration_cost=5000000

Also a [good sysctl.conf file which one might want to substitute or use as a skele for some productive server is ready for download here](#)

Even if you can't reap out great CPU reduction benefits from integrating above values or similar ones, your overall LAMP performance to end customers should increase - at some occasions dramatically, at others little bit but still noticable.

If you're unsure on exact kernel variable values to use check yourself what should be the best values that fits you according to your server Hardware - usually this is done by experimenting and reading the kernel documentation as provided for each one of uplisted variables.

Above **sysctl.conf** is natively created to run on Debian and on other distributions like CentOS, Fedora Slackware some values might either require slight modifications.

Hope this helps and gives you some idea of how network optimization in Linux is usually done. Happy (hacking) tweakening !