

Performance Tweaks & Tips

By *srlinuxx*

Created *10/04/2005 - 4:36am*

Submitted by srlinuxx on Sunday 10th of April 2005 04:36:52 AM Filed under [Howtos](#) [1]

Has your system seemed to have slowed down lately or perhaps it never performed the way you thought it should. Do you ever exclaim, seems my friend's computer is much faster than mine... or the dreaded, my XP is faster than linux? Bite your tongue and check out a few things on your gentoo install.

I don my asbestos house robe and share a few things I've learned from my time with gentoo. Actually these principals can be applied to any linux installation, but I had gentoo in mind when writing them.

- Filesystem type can make a difference. You should have researched this before install the only conversion that works is ext2 to ext3. If possible for you, you may want to test different filesystems to see which works for you best.
 - I prefer reiserfs. I find it stable and have never lost an install using it. Every lost partition was on ext2 and ext3. I find reiserfs the fastest for me as it's probably best for a bunch of small files as opposed to a few large ones. YMMV.
- Optimize your build flags.
 - You should have chosen your CFLAGS very carefully at install time for your system, but it doesn't hurt to consult the documentation and recheck them from time to time, maybe every major version gcc upgrade. Some get deprecated while new ones introduced.
 - You'll hate to hear this one, but rebuild at least major packages if you change them dramatically (glibc, gcc, binutils, qt, gtk, X, mod-utils, kernel, kde, etc). In fact some you may have to build twice, like gcc and glibc.
- USE flags.
 - Some use flags can help speed things up. One of which is nptl if you're running a 2.6 kernel.
 - Read the documentation and select your USE flags carefully. Building support into apps that you'll

never need can and will slow some things down.

- I hesitate to post my CFLAGS and USE flags as they are so system and goal dependent. Please check the documentation [here](#) [2] and [here](#) [3].

- Less info stored the better.
 - Keep all your big movie files and such on another partition. Preferably one that's not mounted all the time.
 - Remove unnecessary files such as in /tmp, /usr/portage/distfiles and /var/tmp/portage. These can build up fast. Check them often, or as I do, I put the following in my /etc/conf.d/local.start file:

```
rm /usr/portage/distfiles/* -rf
rm /var/tmp/portage/* -rf
```
 - Remove all old kernels. I usually keep the last two sources, kernels and modules around. If you run gentoo, you know it don't take long before you have a lot of kernel sources.
 - Check that home directory. In my case, it don't take long at all before I have all kinds of jpegs, gifs, and mpegs laying around. Clear out those application sources, crash dumps, and dot files/directories (you no longer need).
 - Check your /var/log/ directory. Perhaps change your logrotate schedule to keep only a couple rotations instead of the default 5.
- Your kernel is the brain of your os, don't let it get all scrambled. (little joke there)
 - Upgrade to a 2.6 kernel. I found a noticeable speed increase when I made the switch.
 - Turn on pre-empt.
 - Build support for all filesystems you use daily into the kernel as opposed to modules.
 - Same goes for ethernet support.
 - Don't enable support for all this cool stuff you plan on getting someday.
 - Don't load every module at boot time unless you're gonna use it regularly.
 - Disable debugging.
- ccache
 - ccache can speed up those compile times somewhat in if you need to restart a compile, it'll take up where it left off. This is especially handy in the cases of mozilla, openoffice.org or kde-meta for example. All of which are huuuge and take forever to compile - and quite frequently bomb out before it completes.
 - May not be installed by default. `emerge ccache` and put an entry in your /etc/make.conf in the FEATURES.
- hdparm - Tweak the harddrive.
 - You'll need to install hdparm, `emerge hdparm`, as it's not installed by default.

- I check that dma is enabled (also make sure you build support for your chips in the kernel). set with the -d flag - 1 on, 0 off. (test first, may already be enabled if running 2.6).
- 32-bit IO - use -c1 to set on.
- Multi-sector count - you'll have to experiment with this one. I usually find 16 works best for me.

- You can check your drive's set up with something like this

example: `hdparm -Ii /dev/hda`

- Check your harddrives' speed, for example: `hdparm`

`-Tt /dev/hd[ab]`

- so the command that I put in my `/etc/conf.d/local.start` is: `hdparm -d1 -c1 -m16 /dev/hd[ab]`

- Video/Graphics

- Use the correct kernel and X support for your graphics card - don't rely on basic vesa or fb-vesa if possible.
- If you have an nvidia or newer ati, I suggest using the proprietary drivers they provide. (I know, know...)
- Do you really need "composite" or "render" acceleration enabled?
- Use the lowest resolution, bit depth, and refresh you can stand. I like mine at 1280x1024x16@85, (although I had to recently begin using 24 bit depth so DOOM 3 would play).

- Misc

- Temp - excessive temperatures can slow things down. In fact they can cause crashes and even system damage. Perhaps your fans and heatsink(s) need blowing out.
- Do you have services starting up that you don't need? xinetd, apache, sendmail, or cups just for example?
- Prelinking - ok, I really don't mess with this much anymore as I didn't see much difference and it was a pain to remember to re-run every upgrade. But some people like it, so I mention it as something you may want to research.
- Cli - this is one of the most underrated tips one can get. Gui's are nice and I like my eyecandy as much as the next person, but so many times a cli application or built-in command can do a task much more quickly and efficiently than a gui application. See what you can substitute.

- Can you think of others?

[Howtos](#)

Source URL: <http://www.tuxmachines.org/node/542>

Links:

[1] <http://www.tuxmachines.org/taxonomy/term/58>

[2] <http://www.gentoo.org/dyn/use-index.xml>

[3] <http://gcc.gnu.org/onlinedocs/>