

TopSharedMemoryBug

When I look at 'top', the spamd processes are not sharing enough memory

Here's a sample output from "top" viewing two spamd processes:

PID	USER	PR	NI	CODE	VIRT	RES	SHR	nDRT	%CPU	%MEM	TIME	COMMAND
7158	jm	16	0	964	34300	31m	2280	0	1.8	3.1	0:02	perl
7409	jm	17	0	964	34300	31m	2280	0	0.0	3.1	0:00	perl

Looking at the above 'top' output, it appears that the two processes listed are sharing a pitiful 2280KB of memory pages – 3.4% of their total memory usage. However, all's not as it seems.

Are you running Linux? If so, this is an annoying bug somewhere between 'top' and Linux memory management.

What's happening is that, sometime during kernel 2.5 development, the memory management subsystem was changed to no longer track data pages shared between multiple processes through copy-on-write, as 'shared'. Instead, only text pages, and pages loaded from shared libraries, are considered part of the 'shared' section, in contradiction to what the top(1) manual page states.

This is generally observable in all 2.6.x kernels and Red Hat's patched versions of the 2.4.1x and 2.4.2x kernels. Reportedly, RHEL 2.1 ES and AS kernels are the same, too.

This is hard to spot for most typical heavyweight Linux desktop apps written in C/C+: since they use lots of large shared libraries (GNOME, GTK, X11, etc.), this underreporting is not as noticeable, and most people are unaware of it. But spamd makes it quite obvious, since most of its memory usage is compiled perl code, in the 'data' section.

In reality, each spamd process is probably sharing somewhere between 50% and 90% of its pages with the other spamds (depending on its age), but 'top' is unable to report this correctly.

More details, and a demonstration script, [can be found here](#).

Workarounds

If you want a reliable idea of your memory sharing, one way to determine this seems to be to downgrade the kernel to vanilla 2.4.[12]* or test on a non-Linux platform. There's also patches that can be applied to 2.6 to get useful figures via the /proc/\$\$/smaps interface; [lkml thread on the topic](#).