Re: −Os versus −O2

Source: http://linux.derkeiler.com/Mailing−Lists/Kernel/2007−06/msg10370.html

- *From*: Willy Tarreau <w@xxxxxx>
- *Date*: Mon, 25 Jun 2007 10:19:05 +0200

On Mon, Jun 25, 2007 at 09:08:23AM +0200, Segher Boessenkool wrote:

> > In my experience, −Os produced faster code on gcc−2.95 than −O2 or −O3.
>
> On what CPU? The effect of different optimisations varies
> hugely between different CPUs (and architectures).

x86

> > It was not only because of cache considerations, but because gcc used
> > different tricks to avoid poor optimizations, and at the end, the CPU
> > ended executing the alternative code faster.
>
> −Os is "as fast as you can without bloating the code size",
> so that is the expected result for CPUs that don't need
> special hand−holding around certain performance pitfalls.

> > With gcc−3.3, −Os show roughly the same performance as −O2 for me on
> > various programs. However, with gcc−3.4, I noticed a slow down with
> > −Os. And with gcc−4, using −Os optimizes only for size, even if the
> > output code is slow as hell. I've had programs whose speed dropped
> > by 70% using −Os on gcc−4.
>
> Well you better report those! <http://gcc.gnu.org/bugzilla>

No, −Os is for size only :

−Os Optimize for size. −Os enables all −O2 optimizations
that do not typically increase code size. It also
performs further optimizations designed to reduce code
size.

So it is expected that speed can be reduced using −Os. I won't report
a thing which is already documented !

> > But in some situtations, it's desirable to have the smallest possible
> > kernel whatever its performance. This goes for installation CDs for
> > instance.
>
> There are much better ways to achieve that.

Optimizing is not a matter of choosing *one* way, but cumulating
everything you have. For instance, on a smart boot loader, I have
a kernel which is about 300 kB, or 700 kB with the initramfs. Among
the tricks I used :
− −Os
− −march=i386
− align everything to 0
− replace gzip with p7zip

Even if each of them reduces overall size by 5%, the net result is
0.95^4 = 0.81 = 19% gain, for the same set of features. This is
something to consider.

Regards,
Willy

−