

Re: repeatedly opening the same .so(s) is slow?

Re: repeatedly opening the same .so(s) is slow?

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2006-03/msg00197.html>

- *From:* Eric Schuele <e.schuele@xxxxxxxxxxxxx>
 - *Date:* Mon, 20 Mar 2006 13:06:28 -0600
-

Peter Jeremy wrote:

On Mon, 2006-Mar-20 10:49:57 -0600, Eric Schuele wrote:

I have a port (gnucash) which takes 3-4 minutes to open on a 2.6GHz machine. It used to take 15-20 seconds till all of the libtool changes.

It takes 15 minutes on a -current Athlon XP-1800 and about 2 minutes on a 2.2GHz AMD64 running -stable.

I have no idea if the symptom is related to libtool or not.

I initially thought it was libtool related but now I'm uncertain. I didn't just upgrade libtool, I upgraded quite a few other ports at the same time. On the not-libtool side, ade@ says that he hasn't seen this behaviour with other libtool/libltdl ports and I've found that guile include it's own libltdl code (based on libtool). I'm not sure if this is gnucash specific or affects other guile applications.

FWIW... I have removed my symlink to libguile-ltdl.so and recreated it to point at libltdl.so.1. So that guile is using my stock libltdl.so. I get the same results. And gnucash seems to run fine.

Using truss, I can see that gnucash/guile is trying to open a dozen or two files, repeatedly. It fails attempting to open it the first few times everytime it tries to access it, because it is traversing the LD_LIBRARY_PATH:

Worse than that, it's expanding LD_LIBRARY_PATH using additional paths embedded in the .la files that it's opening.

Now I said a dozen or two files repeatedly. It is 12-20 files maybe... but it is

Re: repeatedly opening the same .so(s) is slow?

attempting to open them *hundreds of thousands of times*! It goes on and on and on...

I took a complete ktrace of the startup and there are 24e6 NAMI events with the top files tested 2,000,000 times.

I have thought of placing symlinks in the folder(s) where it first looks for any given file, to make sure it finds the file... but this does not seem quite right either.

It's definitely a hack. I tried something like this and it didn't help much. The code still wants to open libraries multiple times.

I've been looking at adding caching to `lt_dlopenext()` and my first attempt went much faster but blew up because I wasn't correctly handling open/close/open sequences (`libm` is opened and then closed 42,000 times). I think this is the way forward but need to find the time to understand `ltdl.[ch]` (~4800 lines).

What I'm wondering is.... what is the lists opinion on how to best fix this type of a problem. Is this even the cause of my long startup?

Any system calls involving opening pathnames are expensive, even with the namei cache. Having 4 orders of magnitude too many is a distinct problem.

I have spoken with one or two of the gnucash devs, they seem to think this is unique to FreeBSD, meaning they have not seen this problem on any other platform. They said it might have to do with how FreeBSD handles how files are opened up many times recursively!?

Possibly Linux can more efficiently handle opening a non-existent file but the underlying problem is that there are far too many system calls being executed during the gnucash startup. It would be interesting to get a truss of gnucash starting on another OS (unfortunately, I don't have access to any Linux systems) and/or some other guile applications.

hmm... I have a Gentoo system somewhere. It was just an experiment. No idea what shape its in. But maybe I can try installing gnucash on it.

If there is a more appropriate list, please let me know.

Re: repeatedly opening the same .so(s) is slow?

Re: repeatedly opening the same .so(s) is slow?

-ports may be better.

--

Regards,
Eric

freebsd-hackers@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-hackers>

To unsubscribe, send any mail to "freebsd-hackers-unsubscribe@xxxxxxxxxxx"