

Re: Pthread scheduling in FreeBSD 7

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/stable/2007-11/msg00728.html>

- *From:* Daniel Eischen <deischen@xxxxxxxxxxx>
 - *Date:* Wed, 28 Nov 2007 04:08:39 -0500 (EST)
-

On Tue, 27 Nov 2007, Gregor Maier wrote:

-----BEGIN PGP SIGNED MESSAGE-----

Hash: SHA1

Hello,

I have a question concerning *pthread* scheduling with FreeBSD 7.

Did this get answered already? I missed it if it did.

My goal:

I have a multi-threaded application, in which I have a thread that should get higher priority than the other threads. Realtime-Priority for this thread would be nice but is not necessary.

My approach so far (which worked for FreeBSD 6.2):

Use the libpthread implementation, use `pthread_attr_setschedparam()` to increase the priority (scope is `PTHREAD_SCOPE_SYSTEM`, policy is `SCHED_RR`). For the libpthread implementation that worked fine and did just what I expected. I also tried it using `libthr`, but `libthr` didn't care about the priority.

My problem in FreeBSD 7:

The libpthread implementation seems to have gone and `libthr` is the default. However `libthr` still ignores the pthread scheduling parameters. (FreeBSD 7 has a different default priority and policy for `libthr` than FreeBSD 6 however). I also saw, that `libkse` is now available as a new M:N pthread implementation, so I tried using that. But `libkse` is significantly slower than `libthr` and `libkse` does not always schedule to all CPUs. As far as I can tell `libkse` only schedules to all CPUs if the

Re: Pthread scheduling in FreeBSD 7

load is small when the threads are created. I have never seen this behaviour with libpthread from FreeBSD 6.

And libkse is quite unpredictable: I used a test program with 10 threads (with equal priority) and recorded the wall-runtime of each of them. The runtime of them differs by a factor of up to 5(!).

I have to mention however, that I used different machines. The one running FreeBSD 6 is a dual-cpu AMD Athlon with 32bit OS, while the one running FreeBSD 7 is a dual-core Intel running a 64bit OS. In FreeBSD 6 I used SCHED_4BSD in the kernel, while I used SCHED_ULE in FreeBSD 7.

My questions:

- * Can anyone shed some light on this issue?
- * How can I tune thread scheduling in FreeBSD 7? What happened to the libpthread implementation from FreeBSD 6?

It is the same as libkse, just renamed. Other than setting the sysctl kern.threads.virtual_cpu (which defaults to the number of CPUs), there isn't much else. And if all threads are system scope, virtual_cpu > 1 isn't of any use.

- * What triggers the whether libkse schedules to only one or to all CPUs? Is this tuneable?

In both cases (system and process-scope threads) it is dependent on the kernel. For process scope threads, there is only one userland run-queue and threads are scheduled onto KSEs as they become available. When threads block in the kernel, an upcall is made and libkse runs another thread in that KSE. When threads unblock, the kernel puts them in the next KSE upcall – they are not bound to any specific upcall. What should be done is that threads should get bound to a specific upcall and they should stay there (N run queues to match N KSEs) and threads should only migrate to average the KSE loading.

For system scope threads it is totally dependent on the kernel scheduler.

- * Do rtprio() and/or nice() work for single threads?

I'm not sure about nice(), but rtprio only works if you have superuser privileges.

Re: Pthread scheduling in FreeBSD 7

* Any other ideas how I can assign a higher priority to a thread. Maybe even realtime priority?

For libthr, priorities are only really used if the process has superuser privileges. Try running (if it is safe) it as root. You should set the scheduling class to `SCHED_FIFO` or `SCHED_RR` (see `sched_setscheduler(2)`).

--
DE

freebsd-stable@xxxxxxxxxxx mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-stable>
To unsubscribe, send any mail to "freebsd-stable-unsubscribe@xxxxxxxxxxx"