

Re: Massive performance loss from OS::sleep hack

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2007-09/msg00043.html>

- *From:* "Kip Macy" <kip.macy@xxxxxxxxxx>
 - *Date:* Sun, 16 Sep 2007 12:56:49 -0700
-

On 9/16/07, Daniel Eischen <deischen@xxxxxxxxxx> wrote:

On Sat, 15 Sep 2007, Kip Macy wrote:

Or more likely they'll continue to maintain a sched_yield that isn't posix compliant. We may just want to add some sort of interface so the jvm can tell the kernel that sched_yield should be non-compliant for the current process.

I don't think that is a good idea, it seems like too much of a hack. The scheduler(s) should schedule threads the way they are designed to, either obeying a threads priority, using it as a hint, or totally ignoring it.

If the JVM kept track of the thread priorities in use, I suppose Thread.yield() could first lower the current thread's priority to the next used priority and then yield, raising the priority back after the yield. This isn't perfect, there are race conditions, the next highest priority thread(s) could be blocked and not runnable, etc. Maybe just lowering the priority to min or default priority would work well enough.

Yes, if we could hack the jvm to work around this without calling sleep that would be better yet. However, making java work well is more important than keeping the interface clean.

This test would fail even on Solaris if you use SCHED_RR or SCHED_FIFO since it is POSIX compliant for those scheduling classes.

I honestly don't think it matters. The JCK using it implies that there are probably java apps that rely on this defective behavior. I would be willing to bet that Solaris and Java developers at Sun already had this argument and the Java developers won :-(.

Re: Massive performance loss from OS::sleep hack

-Kip

freebsd-performance@xxxxxxxxxxx mailing list

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

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