

## Re: single thread per connection

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2008-07/msg00933.html>

---

- *From:* "Peter Duniho" <NpOeStPeAdM@xxxxxxxxxxxxxxxxxxxx>
  - *Date:* Sat, 12 Jul 2008 14:04:55 -0700
- 

On Sat, 12 Jul 2008 10:25:52 -0700, Neil Coffey <neil.coffey@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

Peter Duniho wrote:

On 32-bit Windows, the theoretical maximum number of threads per process is about 2000, with the practical maximum somewhat lower, and performance suffering significantly before that. Unix/Linux would be different...threads are much lighter-weight constructs on those OSs.

Hmm that's interesting -- I wasn't aware of this 2000 limit. FWIW, in a quick test I can start up about 5,600 threads in XP before I get a "Cannot start new native thread" error, but I'd concede that's more or less a limit within the order you say.

Is that in Java? The limit comes from the size of the stack allocated for a thread (1MB by default) and the maximum virtual address space for a process (2GB). If you use a different stack size than the default, or don't actually allocate one OS thread per Java thread, then the actual limit would be different.

Other than that it's a hard limit, and in practice you can't even reach that maximum because the process's virtual address space will include other things that prevent the entire space from being used for thread stacks.

Although it's a bit bonkers to want so many threads, it's also kind of disappointing that you absolutely can't. I guess Windows uses a bit more memory than you'd think for each thread control/environment block.

As I mention above, the main limit is the stack for each thread. They are also relatively "heavyweight" as compared to Unix threads for other reasons, but the main limit with respect to the maximum number of threads is simply address space.

I'm also intrigued by another thing: if I understand you correctly, you're saying that there's a performance difference between having

Re: single thread per connection

X threads all from one process and having X threads distributed across various processes.

No, that's not