

Re: concurrency, threads and objects

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2006-11/msg01890.html>

- *From:* Chris Smith <cdsmith@xxxxxxx>
 - *Date:* Wed, 15 Nov 2006 19:59:49 -0700
-

Tom Forsmo <spam@xxxxxxxxxxx> wrote:

If I use 100 threads, I would create 100 ClassA objects, which means I would have 100 ClassA objects and 100 Thread objects.

So what ?

I don't believe in code bloat and I see it as unnecessary runtime resource consumption. I don't subscribe to the idea that you should not worry about resources (cpu, memory etc.), because its so cheap. The reason is simple, bloated code runs slower and is more difficult to maintain. Think of a program that takes up 300 MB of memory and compare it to a program that only requires say, 150MB. The smaller program requires less bus bandwidth between the cpu, memory and disk and less processing cycles (barring algorithm efficiency).

You seem to see things in black and white. The world doesn't work that way. Practically everything is an object in Java. Objects are cheap. The entire runtime system, memory management, etc. is designed that way, and people have put lots of effort into making it so. Anything else you do that tries to minimize creating objects is likely to not be a noticeable improvement, and often hurts the performance of your code.

On the other hand, creating 100 threads is certainly not cheap, and almost certainly harmful if you care about performance in this application... unless it will be running on some kind of supercomputer that has at least 50 processors or so. Sometimes creating 100 threads can make your development life easier by helping you separate various tasks in your application design; but if that cost is okay with you, you are certainly misplacing your priorities when you worry about creating that extra 100 objects. This isn't about whether you should be happy with a sub-optimal program. It's about whether you should worry about polishing the deck when the Titanic is sinking.

Re: concurrency, threads and objects

—
Chris Smith