

## Re: Comments on the KSE option

---

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/current/2006-10/msg00583.html>

---

- *From:* Chuck Swiger <[cswiger@xxxxxxx](mailto:cswiger@xxxxxxx)>
  - *Date:* Sun, 29 Oct 2006 12:33:44 -0500
- 

Matthew Dillon wrote:

: I think the notion of fairness is orthogonal to M:N threading. M:N is about : efficiently representing user threading to kernel space, as well as avoiding : kernel involvement in user context switches when not needed. Fairness is : about how the kernel allocates time slices to user processes/threads. : Fairness can be implemented for both 1:1 and M:N, with the primary differences : being in bookkeeping.

Yes, this is precisely what I mean. Very well said.

Agreed.

What we are talking about here is primarily algorithmic complexity and physical resource limitations (e.g. like kernel memory). Having the kernel scheduler only deal with (N) threads, where N is limited by the number of physical cpus, is a far easier problem for the kernel to solve in all respects then having the kernel deal with (M\*N) individual threads.

Yes. A userspace program would like to assume that it can have as many parallel threads of execution as it wants to have, but the kernel only needs to schedule N of them at a time from the set of runnable tasks.

Most of the debate comes from whether userland threads are counted in M, or only userland processes.

I personally see no reason why a program couldn't have 10,000 threads, or 100,000 threads, or a million threads, but the kernel is the wrong place to try to manage them if your system only has N cpus (N=2,4,8,16,32, etc). You have to ask yourself, what exactly is the kernel accomplishing trying to manage all those threads for a single application when it only has N cpu contexts to work with anyhow?

Well, it is solving the same problem that the kernel needs to solve when someone sets up a web hosting machine with a hundred jails (or a thousand), times anywhere from ten to 200+ preforked httpd processes.

## Re: Comments on the KSE option

The answer is: The kernel should only have to worry about the N cpu contexts and kernel memory resources for