

---

Subject: Re: [PATCH 1/4] Virtualization/containers: introduction

Posted by [Sam Vilain](#) on Thu, 09 Feb 2006 22:09:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Jeff Dike wrote:

> On Thu, Feb 09, 2006 at 11:38:31AM -0500, Hubertus Franke wrote:

>>Jeff, interesting, but won't that post some serious scalability issue?

>>Imaging 100s of container/namespace ?

> In terms of memory?

> Running size on sched.o gives me this on x86\_64:

> text data bss dec hex filename

> 35685 6880 28800 71365 116c5 sched.o

>

> and on i386 (actually UML/i386)

>

> text data bss dec hex filename

> 10010 36 2504 12550 3106 obj/kernel/sched.o

>

> I'm not sure why there's such a big difference, but 100 instances adds

> a meg or two (or three) to the kernel. This overstates things a bit

> because there are things in sched.c which wouldn't be duplicated, like

> the system calls.

>

> How big a deal is that on a system which you plan to have 100s of

> containers on anyway?

Quite a big deal. You might have 2Gigs of main memory, but your CPU is unlikely to be more than a Megabyte in close reach. A meg or two of scheduler data and code means that your L1 and L2 cache will be cycling every scheduler round; which is OK if you have very short runqueues but as you get more and more processes it will really start to hurt.

Remember, systems today are memory bound and anything you can do to reduce the amount of time the system sits around waiting for memory to fetch, the better.

Compare that to the Token Bucket Scheduler of Linux-VServer; a tiny struct for each process umbrella, that will generally fit in one or two cachelines, to which the scheduling support adds four ints and a spinlock. With this it achieves fair CPU scheduling between vservers.

Sam.

---