Subject: Re: [ckrm-tech] containers development plans
Posted by Paul Jackson on Tue, 10 Jul 2007 07:25:20 GMT
View Forum Message <> Reply to Message

Kirill, Serge, et al,

Is it fair to say then that Paul Menage's containers are primarily
for the purposes of managing resources, while namespaces are for the
purposes of managing identifiers?

We've got some resources, like cpu cycles, memory bytes, network
bandwidth, that we want to allocate and account for differentially
by groups of tasks -- that's Menage's containers.

We've got some system wide namespaces, like process id's, that we
want to virtualize, for more flexible uses -- these are the name-
space containers.

In Serge's opening post to this thread, he wrote:
 1. namespaces
 2. process containers
 3. checkpoint/restart

Are the 'process containers' of item (2) the containers of Paul Menage?

If so, then I propose that this thread is misnamed. It should not be
"containers development plans", but rather "namespace, container and
c/r development plans." And if so, there is really no conflict over
the use of the word 'container' -- that applies just to the resource
virtualization efforts, of which my cpusets is the granddaddy example,
being generalized by Paul Menage with his container patches. The other
work is, as Serge actually termed it in the body of his post, better
called 'namespaces'.

Perhaps the confusion arose from looking for a single word to encompass
all three parts, listed above, of this work. The efforts have some strong
dependencies, but taking the name of one of the efforts, containers, and
trying to make it serve double duty as the umbrella term, might be an
unnecessary confusion.

Perhaps also, on a separate point, the word 'process' in item (2) is
not the right focus. I think that the essential purpose of (2) is
resource management. While most of those resource management is done
per-process, it might also be per-file or per-virtual-address-range.
For example, disk i/o backing pages in a particular range of an
applications address space might have certain bandwidth limits, and the
memory backing the pages in that range might have certain memory node
placement restrictions, or the i/o to and from a particular disk file

might have certain bandwidth or placement constraints. We see a bit of
this in present day kernels, with the mbind(2) system call.

So, I suggest, we have three efforts:
 1. namespaces,
 2. resource containers, and
 3. checkpoint/restart.

And they are very much in this order. Names, such as pathnames, task pids, user uids, and other system wide identifiers, are essential to the implementation of all else. Resource containers depend on some naming scheme, and reach out to manage the use of resources outside the operating system, such as disk, network, memory and processor. Checkpoint/restart is a particular feature of interest, that requires that both names and resources be virtualized to some degree.

Are there any Python programmers in the namespace work? The use of namespaces in Python might serve as a informative example for the work we need in Linux namespaces. See further page 418, section "A.2 Namespaces and Binds" of David Mertz's "Text Processing in Python" for a clear and concise exposition of the central role of namespaces in Python.

--
                  I won't rest till it's the best ...
                  Programmer, Linux Scalability
                  Paul Jackson <pj@sgi.com> 1.925.600.0401