Subject: Re: [ckrm-tech] [PATCH 00/10] Containers(V10): Generic Process Containers
Posted by Paul Jackson on Thu, 07 Jun 2007 22:01:13 GMT
View Forum Message <> Reply to Message

> > The set of people using exclusive cpusets is roughly some subset of
> > those running multiple, cpuset isolated, non-cooperating jobs on big
> > iron, usually with the aid of a batch scheduler.
>
> Unfortunately I would imagine these users to be very intereseted in
> providing checkpoint/restart/migrate functionality.

Yup - such customers are very interested in checkpoint, restart, and
migrate functionality.

> Surely if the admin wants to give cpus 5-6 exclusively to /cpusets/set0/set4
> later, those cpus can just be taken away from set3?

Yeah - that works, so far as I know (which isn't all that far ..')

But both:
 1) that (using whatever cpus are still available) and
 2) my other idea, of not allowing any cloning of cpusets with
    exclusive siblings at all,

looked a little ugly to me.

For example, such customers as above would not appreciate having their
checkpoint/restart/migrate fail in any case where there weren't spare
non-exclusive cpus, which for users of the exclusive flag, is often the
more common case.

My rule of thumb when doing ugly stuff is to constrain it as best
I can -- minimize what it allows. This led me to prefer (2) above
over (1) above.

Perhaps there's a better way to think of this ... When we clone
like this for checkpoint/restart/migrate functionality, perhaps
we are not really starting up a new, separate, competing job that
should have its resources isolated and separated from the original.

Perhaps instead we are firing up a convenient alter-ego of the
original job, which will be co-operatively using the same resources
by default. If that's the normal case, then it seems wrong to
force the clone onto disjoint CPUs, or fail for lack thereof.

So perhaps we should refine the meaning of 'exclusive', from:
 - no overlapping siblings

to:
 - no overlapping siblings other than clones of ones self.

Then default to cloning right on the same CPU resources as the
original, possibly with both original and clone marked exclusive.

--

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