Subject: Re: [ckrm-tech] [PATCH 00/10] Containers(V10): Generic Process Containers
Posted by Paul Jackson on Thu, 07 Jun 2007 00:46:09 GMT
View Forum Message <> Reply to Message

> I suppose as a cleaner alternative we could
> add a container_subsys->inherit_defaults() handler, to be called at
> container_clone(), and for cpusets this would set cpus and mems to
> the parent values - sibling exclusive values. If that comes to nothing,
> then the attach_task() is still refused, and the unshare() or clone()
> fails, but this time with good reason.

Unfortunately, I haven't spent the time I should thinking about
container cloning, namespaces and such.

I don't know, for the workloads that matter to me, when, how or
if this container cloning will be used.

I'm tempted to suggest the following.

First, I am assuming that the classic method of creating cpuset
children will still work, such as the following (which can fail
for certain combinations of exclusive cpus or mems):

```
 cd /dev/cpuset/foobar
 mkdir foochild
 cp cpus foochild
 cp mems foochild
 echo $$ > foochild/tasks
```

Second, given that, how about you fail the unshare() or clone()
anytime that the instance to be cloned has any sibling cpusets
with any exclusive flags set.

The exclusive property is not really on friendly terms with cloning.

Now if the above classic code must be encoded using cloning under
the covers, then we've got problems, probably more problems than
just this.

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