Subject: Re: [ckrm-tech] [patch00/05]: Containers(V2)- Introduction
Posted by Chandra Seetharaman on Thu, 21 Sep 2006 21:44:22 GMT
View Forum Message <> Reply to Message

On Thu, 2006-09-21 at 13:10 -0700, Paul Menage wrote:
> On 9/21/06, Chandra Seetharaman <sekharan@us.ibm.com> wrote:
> > > The current fake numa support requires you to choose your node layout
> > > at boot time - I've been working with 64 fake nodes of 128M each,
> > > which gives a reasonable granularity for dividing a machine between
> > > multiple different sized jobs.
> >
> > It still will not satisfy what OpenVZ/Container folks are looking for:
> > 100s of containers.
>
> Right - so fake-numa is not the right solution for everyone, and I
> never suggested that it is. (Having said that, there are discussions
> underway to make the zone-based approach more practical - if you could
> have dynamically-resizable nodes, this would be more applicable to
> openvz).

It would still have the other issue you pointed, i.e the userspace being
able to cope up with memory allocators dynamics.

>
> But, there's no reason that the OpenVZ resource control mechanisms
> couldn't be hooked into a generic process container mechanism along
> with cpusets and RG.

Isn't that one of the things we are trying to avoid (each one having
their own solution, especially when we _can_ have a common solution).

>
> Paul
--

--- ----------
Chandra Seetharaman | Be careful what you choose....
- sekharan@us.ibm.com | .....you may get it.
--- ----------

Subject: Re: [ckrm-tech] [patch00/05]: Containers(V2)- Introduction
Posted by Paul Menage on Thu, 21 Sep 2006 22:09:47 GMT
View Forum Message <> Reply to Message

On 9/21/06, Chandra Seetharaman <sekharan@us.ibm.com> wrote:
>
> >

> > But, there's no reason that the OpenVZ resource control mechanisms
> > couldn't be hooked into a generic process container mechanism along
> > with cpusets and RG.
>
> Isn't that one of the things we are trying to avoid (each one having
> their own solution, especially when we _can_ have a common solution).

Can we actually have a single common solution that works for everyone, no matter what their needs? It's already apparent that there are multiple different and subtly incompatible definitions of what "memory controller" means and needs to do. Maybe these can be resolved - but maybe it's better to have, say, two simple but very different memory controllers that the user can pick between, rather than one big and complicated one that tries to please everyone.

Paul
