

---

Subject: Re: [ckrm-tech] [patch00/05]: Containers(V2)- Introduction  
Posted by [Chandra Seetharaman](#) on Fri, 22 Sep 2006 00:06:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 2006-09-21 at 15:09 -0700, Paul Menage wrote:

> 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,

Think about what will be available to customer through a distro.

There are two (competing) memory controllers in the kernel. But, distro  
can turn only one ON. Which in turn mean

- there will be a debate from the two controller users/advocates with  
the distro (headache to distro) about which one to turn ON
- one party will not get what they want and hence no point in them  
getting the feature into the mainline in the first place  
(dissatisfaction of the users/original implementors of one solution).

So, IMHO, it is better to sort out the differences before we get things  
in mainline kernel.

>

> 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 Fri, 22 Sep 2006 00:13:31 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

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

> Think about what will be available to customer through a distro.

>

> There are two (competing) memory controllers in the kernel. But, distro

> can turn only one ON. Which in turn mean

Why's that? I don't see why cpuset memory nodemasks can't coexist with, say, the RG memory controller. They're attempting to solve different problems, and I can see situations where you might want to use both at once.

>

> So, IMHO, it is better to sort out the differences before we get things

> in mainline kernel.

Agreed, if we can come up with a definition of e.g. memory controller that everyone agrees is suitable for their needs. You're assuming that's so a priori, I'm not yet convinced.

And I'm not trying to get another memory controller into the kernel, I'm just trying to get a standard process aggregation into the kernel (or rather, take the one that's already in the kernel and make it possible to hook other controller frameworks into it), so that the various memory controllers can become less intrusive patches in their own right.

Paul

---

---

Subject: Re: [ckrm-tech] [patch00/05]: Containers(V2)- Introduction  
Posted by [Paul Jackson](#) on Fri, 22 Sep 2006 00:24:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Chandra wrote:

> There are two (competing) memory controllers in the kernel. But, distro

> can turn only one ON.

Huh - time for me to play the dummy again ...

My (fog shrouded) vision of the future has:

- 1) mempolicy - provides fine grained memory placement for task on self
- 2) cpuset - provides system wide cpu and memory placement for unrelated tasks
- 3) some form of resource groups - measures and limits proportion of various resources used, including cpu cycles, memory pages and network bandwidth,

by collections of tasks.k

Both (2) and (3) need to group tasks in flexible ways distinct from the existing task groupings supported by the kernel.

I thought that Paul M suggested (2) and (3) use common underlying grouping or 'bucket' technology - the infrastructure that separates tasks into buckets and can be used to associate various resource metrics and limits with each bucket.

I can't quite figure out whether you have in mind above:

- \* a conflict between two competing memory controllers for (3),
- \* or a conflict between cpusets and one memory controller for (3).

And either way, I don't see what that has to do with the underling bucket technology - how we group tasks generically.

Guess I am missing something ...

--

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

---

Subject: Re: [ckrm-tech] [patch00/05]: Containers(V2)- Introduction  
Posted by [Chandra Seetharaman](#) on Fri, 22 Sep 2006 00:57:22 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 2006-09-21 at 17:24 -0700, Paul Jackson wrote:

> Chandra wrote:

> > There are two (competing) memory controllers in the kernel. But, distro  
> > can turn only one ON.

>

> Huh - time for me to play the dummy again ...

>

> My (fog shrouded) vision of the future has:

- > 1) mempolicy - provides fine grained memory placement for task on self
- > 2) cpuset - provides system wide cpu and memory placement for unrelated tasks
- > 3) some form of resource groups - measures and limits proportion of various  
> resources used, including cpu cycles, memory pages and network bandwidth,  
> by collections of tasks.k

>

> Both (2) and (3) need to group tasks in flexible ways distinct from the  
> existing task groupings supported by the kernel.

>

> I thought that Paul M suggested (2) and (3) use common underlying  
> grouping or 'bucket' technology - the infrastructure that separates

> tasks into buckets and can be used to associate various resource  
> metrics and limits with each bucket.  
>  
> I can't quite figure out whether you have in mind above:  
> \* a conflict between two competing memory controllers for (3),

Yes.

> \* or a conflict between cpusets and one memory controller for (3).

No.

>  
> And either way, I don't see what that has to do with the underling  
> bucket technology - how we group tasks generically.

True. I clarified it in the reply to Paul M.

>  
> Guess I am missing something ...  
>  
--

-----  
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 Jackson](#) on Fri, 22 Sep 2006 01:11:41 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Chandra wrote:

> I clarified it in the reply to Paul M.

Yup - thanks.

--

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

---