Subject: Re: [ckrm-tech] [RFC][PATCH] UBC: user resource beancounters
Posted by Rohit Seth on Tue, 22 Aug 2006 01:45:28 GMT
View Forum Message <> Reply to Message

On Mon, 2006-08-21 at 14:45 -0700, Chandra Seetharaman wrote:
> On Mon, 2006-08-21 at 17:24 +0400, Kirill Korotaev wrote:
> > Chandra Seetharaman wrote:
> > > Kirill,
> > >
> > > Here are some concerns I have (as of now) w.r.t using UBC for resource
> > > management (in the context of resource groups).
> > >
> > > - guarantee support is missing. I do not see any code to provide the
> > >   minimum amount of resource a group can get. It is important for
> > >   providing QoS. (In a different email you did mention guarantee, i am
> > >   referring it here for completeness).
> > I mentioned a couple of times that this is a limited core functionality
> > in this patch set.
> > guarantees are implementable as a separate UBC parameters.
>
> I will wait for oomguarpages patches :)
>
> >
> > > - Creation of a UBC and assignment of task to a UBC always happen in
> > >   the context of the task that is affected. I can understand it works in
> > >   OpenVZ environment, but IMO has issues if one wants it to be used for
> > >   basic resource management
> > >   - application needs to be changed to use this feature.
> > >   - System administrator does not have the control to assign tasks to a
> > >   UBC. Application does by itself.
> > >   - Assignment of task to a UBC need to be transparent to the
> > >   application.

I agree with the above points. Just want to add that assignment of a
task to a container may not be transparent to the application. For
example it may hit some limits and some reclaim may happen...

> > this is not 100% true.
> > UBC itself doesn't prevent from changing context on the fly.
> > But since this leads to part of resources to be charged to
> > one UBC and another part to another UBC and so long and so
>
> Let the controllers and the users worry about that part.
>

I think as the tasks move around, it becomes very heavy to move all the
pages belonging to previous container to a new container.

> As I mentioned UBC might be perfect for container resource management,
> but what I am talking for is resource management _without_ a container.
>

Can you explain that part a bit more?

> >
> > > - No ability to maintain resource specific data in the controller.
> > it's false. fields can be added to user_beancounter struct easily.
> > and that's what our controllers do.
>
> With the model of static array for resources (struct ubparm ub_parms
> [UB_RESOURCES] in struct user_beancounter), it is not be possible to
> attach _different_ "controller specific" information to each of the
> entries.
>
> I do not think it is good idea to add controller specific information of
> _different_ controllers to the user_beancounter. Think of all the fields
> it will have when all the numproc controller needs is just the basic 3-4
> fields.
>

IMO it is okay to add the fields whenever necessary as Kirill
suggested. I think once the container subject gets baked a little more,
the controllers will also get tightly coupled.

> >
> > > - No ability to get the list of tasks belonging to a UBC.
> > it is not true. it can be read from /proc or system calls interface,
> > just like the way one finds all tasks belonging to one user :)
> >
> > BTW, what is so valueable in this feature?
>
> Again, it may not be useful for container type usages (you can probably
> get the list from somewhere else, but for resource management it is
> useful for sysadmins).
>

I'm also debating about whether printing task information is really any
useful. If a sysadm wants to get information about any particular task
then that can come from /proc/<pid>/container Though container list
will be one place where one can easily get the list of all the contained
tasks (and other resources like files).

> >
> > > - For a system administrator name for identification of a UBC is
> > >   better than a number (uid).

> > Have you any problems with pids, uids, gids and signals?
>
> Again, in container land each UB is attached with a container hence no
> issue.
>
> In a non-container situation IMO it will be easier to manage/associate
> "gold", "silver", "bronze", "plastic" groups than 0, 11, 83 and 113.
>
>
> > It is a question of interface. I don't mind in changing UBC interface even
> > to configfs if someone really wants it.
> >

Yes please. Thanks.
-rohit
