Subject: Re: [ckrm-tech] [RFC][PATCH] UBC: user resource beancounters
Posted by dev on Mon, 21 Aug 2006 13:21:48 GMT
View Forum Message <> Reply to Message

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.

> - 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.

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
forth, we believe that more clear and correct interface is
something like fork()/exec()-required-application.

So you can always execute new applications in desired UB and
NO application modification are required.

> - UBC is deleted when the last task (in that UBC) exits. For resource
>   management purposes, UBC should be deleted only when the administrator
>   deletes it.

1. UBCs are freed when last _resource_ using it puts the last reference.
 not the task. And it is a BIG error IMHO to think that resource
 management should group tasks. No, it should group _objects_. Tasks
 are just the same objects like say sockets.
2. this is easily changeable. You are the only who requested it so far.
3. kernel does so for many other objects like users and no one complains :)

> - No ability to set resource specific configuration information.
UBC model allows to _limit_ users. It is _core_.
We want to do resource management step by step and send it patch by patch,
while you are trying to solve everything at once.

sys_open() for example doesn't allow to open sockets, does it?
the same for UBC. They do what they are supposed to do.

> - 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.

> - 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?
do you want to have interfaces to find kernel structures and even pages
which belong to the container? tasks are just one type of objects...

> - Doesn't inform the resource controllers when limits(shares) change.
As was answered and noted by Alan Cox:
1. no one defined what type of action should be done when limits change
2. it is extendable _when_ needed. Do you want to introduce hooks just
   to have them?
3. is it so BIG obstacle for UBC patch? These 3-lines hooks code which
   is not used?

> - Doesn't inform the resource controllers when a task's UBC has changed.
the same as above. we don't add functionality which is not used YET
(and no one even knows HOW).

> - Doesn't recalculate the resource usage when a task's UBC has changed.
>   i.e doesn't uncharge the old UBC and charge new UBC.
You probably missed my explanation, that most
resources (except for the simplest one - numproc) can't be recharged
easily. And nothing in UBC code prevents such recharge to be added later
if requested.

> - 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?
It is a question of interface. I don't mind in changing UBC interface even
to configfs if someone really wants it.

Thanks,
Kirill