Subject: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by Pavel Emelianov on Tue, 31 Oct 2006 08:31:28 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 10/30/06, Pavel Emelianov <xemul@openvz.org> wrote:
>> > Debated:
>> > - syscall vs configfs interface
>>
>> 1. One of the major configfs ideas is that lifetime of
>> the objects is completely driven by userspace.
>> Resource controller shouldn't live as long as user
>> want. It "may", but not "must"! As you have seen from
>> our (beancounters) patches beancounters disapeared
>> as soon as the last reference was dropped.
>
> Why is this an important feature for beancounters? All the other
> resource control approaches seem to prefer having userspace handle
> removing empty/dead groups/containers.

That's functionality user may want. I agree that some users
may want to create some kind of "persistent" beancounters, but
this must not be the only way to control them. I like the way
TUN devices are done. Each has TUN_PERSIST flag controlling
whether or not to destroy device right on closing. I think that
we may have something similar - a flag BC_PERSISTENT to keep
beancounters with zero refcounter in memory to reuse them.

Objections?

>> 2. Having configfs as the only interface doesn't alow
>> people having resource controll facility w/o configfs.
>> Resource controller must not depend on any "feature".
>
> Why is depending on a feature like configfs worse than depending on a
> feature of being able to extend the system call interface?

Because configfs is a _feature_, while system calls interface is
a mandatory part of a kernel. Since "resource beancounters" is a
core thing it shouldn't depend on "optional" kernel stuff. E.g.
procfs is the way userspace gets information about running tasks,
but disabling procfs doesn't disable such core functionality
as fork-ing and execve-ing.

Moreover, I hope you agree that beancounters can't be made as
module. If so user will have to built-in configfs, and thus
CONFIG_CONFIGFS_FS essentially becomes "bool", not a "tristate".

I have nothing against using configfs as additional, optional
interface, but I do object using it as the only window inside
BC world.

>> > - Interaction of resource controllers, containers and cpusets
>> > - Should we support, for instance, creation of resource
>> > groups/containers under a cpuset?
>> > - Should we have different groupings for different resources?
>>
>> This breaks the idea of groups isolation.
>
> That's fine - some people don't want total isolation. If we're looking
> for a solution that fits all the different requirements, then we need
> that flexibility. I agree that the default would probably want to be
> that the groupings be the same for all resource controllers /
> subsystems.

Hm... OK, I don't mind although don't see any reasonable use of it.
Thus we add one more point to our "agreement" list
http://wiki.openvz.org/Containers/UBC_discussion

 - all resource groups are independent