Subject: Re: [ckrm-tech] [RFC] Resource Management - Infrastructure choices
Posted by Pavel Emelianov on Mon, 30 Oct 2006 14:38:33 GMT
View Forum Message <> Reply to Message

Paul Jackson wrote:
> Pavel wrote:
>> 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"!
>
> I had trouble understanding what you are saying here.
>
> What does the phrase "live as long as user want" mean?

What if if user creates a controller (configfs directory)
and doesn't remove it at all. Should controller stay in memory
even if nobody uses it?

>
>
>> 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".
>>
>> 3. Configfs may be easily implemented later as an additional
>>   interface. I propose the following solution:
>>   - First we make an interface via any common kernel
>>   facility (syscall, ioctl, etc);
>>   - Later we may extend this with configfs. This will
>>   alow one to have configfs interface build as a module.
>
> So you would add bloat to the kernel, with two interfaces
> to the same facility, because you don't want the resource
> controller to depend on configfs.
>
> I am familiar with what is wrong with kernel bloat.
>
> Can you explain to me what is wrong with having resource
> groups depend on configfs? Is there something wrong with

Resource controller has nothing common with confgifs.
That's the same as if we make netfilter depend on procfs.

> configfs that would be a significant problem for some systems
> needing resource groups?

Why do we need to make some dependency if we can avoid it?

> It is better where possible, I would think, to reuse common
> infrastructure and minimize redundancy. If there is something
> wrong with configfs that makes this a problem, perhaps we
> should fix that.

The same can be said about system calls interface, isn't it?
