Subject: Re: [PATCH 3/7] Containers (V8): Add generic multi-subsystem API to containers
Posted by Srivatsa Vaddagiri on Tue, 10 Apr 2007 14:44:54 GMT
View Forum Message <> Reply to Message

On Fri, Apr 06, 2007 at 04:32:24PM -0700, menage@google.com wrote:
> +struct container_subsys {
> + int (*create)(struct container_subsys *ss,
> + struct container *cont);
> + void (*destroy)(struct container_subsys *ss, struct container *cont);
> + int (*can_attach)(struct container_subsys *ss,
> + struct container *cont, struct task_struct *tsk);
> + void (*attach)(struct container_subsys *ss, struct container *cont,
> + struct container *old_cont, struct task_struct *tsk);
> + void (*fork)(struct container_subsys *ss, struct task_struct *task);
> + void (*exit)(struct container_subsys *ss, struct task_struct *task);
> + int (*populate)(struct container_subsys *ss,
> + struct container *cont);
> + void (*bind)(struct container_subsys *ss, struct container *root);

Is the first argument into all the callbacks, struct container_subsys *ss,
necessary?

> +struct container_subsys cpuset_subsys = {
> + .name = "cpuset",
> + .create = cpuset_create,
> + .destroy = cpuset_destroy,
> + .can_attach = cpuset_can_attach,
> + .attach = cpuset_attach,
> + .populate = cpuset_populate,
> + .subsys_id = cpuset_subsys_id,
> + .early_init = 1,
> +};

We seem to be missing a register call here as well ..

--
Regards,
vatsa

Subject: Re: [ckrm-tech] [PATCH 3/7] Containers (V8): Add generic multi-subsystem API to containers
Posted by Paul Menage on Tue, 10 Apr 2007 15:45:13 GMT
View Forum Message <> Reply to Message

On 4/10/07, Srivatsa Vaddagiri <vatsa@in.ibm.com> wrote:
>

> Is the first argument into all the callbacks, struct container_subsys *ss,
> necessary?

I added it to support library-like abstractions - where one subsystem
can have its container callbacks and file accesses all handled by a
library which converts to a higher-level abstraction. See the
res_group example patch to see how it can be used.

>
> > +struct container_subsys cpuset_subsys = {
> > +	.name = "cpuset",
> > +	.create = cpuset_create,
> > +	.destroy = cpuset_destroy,
> > +	.can_attach = cpuset_can_attach,
> > +	.attach = cpuset_attach,
> > +	.populate = cpuset_populate,
> > +	.subsys_id = cpuset_subsys_id,
> > +	.early_init = 1,
> > +};
>
> We seem to be missing a register call here as well ..

Registration is done at compile time in the new patch set, thanks to
the static declaration of subsystems and subsystem ids.
