Subject: Re: [RFC] libcg: design and plans
Posted by Balbir Singh on Wed, 05 Mar 2008 04:48:50 GMT
View Forum Message <> Reply to Message

Xpl++ wrote:
> Hi,
>
> I was wonder if creating such library makes any sense at all,
> considering the nature of cgroups, the way they work and their possible
> application?
> It seems to me that any attempt to create a 'simple' API would actualy
> result in something that will be much harder to use that just making raw
> mkdir/open/read/write/close operations. Another thing is suggested
> config for this lib would be more appropriate for a daemon rather than a
> library.

The configuration file is important, since it allows us two levels of control.
At one level the system administrator and at the other level applications. Each
application can maintain it's own hierarchy across reboots.

We thought of a daemon, but there were several overheads and disadvantages. For
one, we needed an IPC mechanism to communicate every client request to the
daemon, the client being the application. The daemon also becomes the single
point of failure for all applications. Moreover, we want to provide the ability
to programmatically update the configuration. A daemon, if desired can be built
on top of the library we propose.

> In general - cgroup is a very flexible subsystem that can be used in a
> wide variety of ways and modes and trying to create a universal simple
> API would more likely result in something hard to manage and work with.

I agree that cgroups is flexible, but why do you think abstracting common tasks
amongst applications will be hard to manage and work with? We want to provide
API that will allow us to fill in parameters and say -- go create this group or
delete this group.

> The idea of having multiple container managers (applications that use
> libcg) creates a lots of questions and possible issues. Containers are
> supposed provide a flexible resource management and task grouping
> ability, which somewhat implies that there cannot be two different
> resource managers per node without one knowing well the
> desires/goals/methods of the other and vice versa.

We don't assume that there cannot be two different resource managers per node.
We don't enforce any policy, we just allow for easy creation and manipulation of
control groups hierarchially.

And should there be
> only one manager per node - probably it will be easier for it to use
> cgroup subsystem directly rather than using a wrapper library?
>

Could you please elaborate, why is it probably easier?

--
Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers