Subject: Re: [RFC] libcg: design and plans
Posted by Dhaval Giani on Wed, 05 Mar 2008 10:33:43 GMT
View Forum Message <> Reply to Message

On Tue, Mar 04, 2008 at 10:15:20PM -0800, Paul Menage wrote:
> Hi Dhaval,
>
> On Tue, Mar 4, 2008 at 7:23 AM, Dhaval Giani <dhaval@linux.vnet.ibm.com> wrote:
> > Hi,
> >
> > We have been working on a library for control groups which would provide
> > simple APIs for programmers to utilize from userspace and make use of
> > control groups.
> >
> > We are still designing the library and the APIs. I've attached the
> > design (as of now) to get some feedback from the community whether we
> > are heading in the correct direction and what else should be addressed.
>
> There are a few things that it would be nice to include in such a
> library, if you're going to develop one:
>
> - the ability to create abstract groups of processes, and resource
> groups, and have the ability to tie these together arbitrarily. E.g
> you might create abstract groups A, B and C, and be able to say that A
> and B share memory with each other but not with C, and all three
> groups are isolated from each other for CPU. Then libcg would mount
> different resource types in different cgroup hierarchies (you would
> probably tell it ahead of time which combinations of sharing you would
> want, in order that it could minimize the number of mounted
> hierarchies). When you tell libcg to move a process into abstract
> group A, it would move it into the appropriate resource group in each
> hierarchy.
>

I am not very clear about what you are asking for here, so let me try to
rephrase it, and if I have understood it correctly, we can move further
ahead from there.

So there are two different points, /mem and /cpu. /mem has A and C and
/cpu has A, B and C. A and B of /cpu correspond to A of /mem and the C's
are the same. With this is mind, if I say a task should move to B in
/cpu, it should also move to A in /mem?

> - an interface for gathering usage stats from cgroups.
>

Yes, that is a todo. We should get around to it as the functionality
gets implemented in kernel.

> - support for dynamically migrating processes between groups based on
> process connector events (i.e. a finished version of the daemon that
> you were working on last year)
>

libcg is at a lower level than this. The dynamic migration of processes
can be based on top of libcg, and exploit it (and be more powerful than
the daemon I posted last year) It would be able to utilize the
configuration and other capabilities of libcg.

Thanks,
--
regards,
Dhaval

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