Subject: bind() call in cgroup's css structure
Posted by Glauber Costa on Mon, 09 Apr 2012 13:59:56 GMT
View Forum Message <> Reply to Message

Hello Tejun,

During your cgroup refactor, I was wondering if you have any plans to
get rid of the bind() callback that is called when hierarchies are moved?

At least in tree, there seems to be no users for that.
I actually planned to use it myself, to start or remove a jump label
when cpuacct and cpu cgroups were comounted.

Problem is, because we have some calls in the cpuset cgroup from inside
the cpu hotplug handler, we end up taking the almighty cgroup_mutex from
inside the cpu_hotplug.lock.

jump labels take it in most arches through the get_online_cpus()
function call. This means we effectively can't apply jump labels with
the cgroup_mutex held, which is the case throughout the whole bind() call.

All that explained, I figured I might as well ask before I attempted a
solution to that myself: as much as populate(), bind seems to be one of
the overly complicated callbacks, designed for a scenario in which
everything can come and go at will, which is something we're trying to fix.

thanks in advance for your answer!

Subject: Re: bind() call in cgroup's css structure
Posted by Tejun Heo on Mon, 09 Apr 2012 18:09:32 GMT
View Forum Message <> Reply to Message

Hello, Glauber.

On Mon, Apr 09, 2012 at 10:59:56AM -0300, Glauber Costa wrote:
> During your cgroup refactor, I was wondering if you have any plans
> to get rid of the bind() callback that is called when hierarchies
> are moved?
>
> At least in tree, there seems to be no users for that.

I don't have any current plan for the callback but if it doesn't have
in-kernel user, I'd prefer to remove it.

> I actually planned to use it myself, to start or remove a jump label
> when cpuacct and cpu cgroups were comounted.

I see.

> Problem is, because we have some calls in the cpuset cgroup from
> inside the cpu hotplug handler, we end up taking the almighty
> cgroup_mutex from inside the cpu_hotplug.lock.

Yeah, those two are pretty big locks.

> jump labels take it in most arches through the get_online_cpus()
> function call. This means we effectively can't apply jump labels
> with the cgroup_mutex held, which is the case throughout the whole
> bind() call.
>
> All that explained, I figured I might as well ask before I attempted
> a solution to that myself: as much as populate(), bind seems to be
> one of the overly complicated callbacks, designed for a scenario in
> which everything can come and go at will, which is something we're
> trying to fix.

I haven't read the code so this could be completely off but if this is
jump label optimization which can be made to work w/o it immediately
applied, maybe just punt it to a work item from the callback? Note
that if cancellation is necessary for e.g. unbinding, it may
re-introduce locking dependency through flushing.

Thanks.

--
tejun

Subject: Re: bind() call in cgroup's css structure
Posted by Li Zefan on Tue, 10 Apr 2012 00:59:33 GMT
View Forum Message <> Reply to Message

Glauber Costa wrote:

> Hello Tejun,
>
> During your cgroup refactor, I was wondering if you have any plans to get rid of the bind() callback that is called when hierarchies are moved?

I planned to remove it long ago, and Paul M agreed. But after some time, I was
trying to make use of it in a patchset, which was used to fix the problem that
remount with different subsys bits will fail for !root cgroups.

>

> At least in tree, there seems to be no users for that.
> I actually planned to use it myself, to start or remove a jump label
> when cpuacct and cpu cgroups were comounted.
>
> Problem is, because we have some calls in the cpuset cgroup from inside the cpu hotplug handler, we end up taking the almighty cgroup_mutex from inside the cpu_hotplug.lock.
>
> jump labels take it in most arches through the get_online_cpus() function call. This means we effectively can't apply jump labels with the cgroup_mutex held, which is the case throughout the whole bind() call.
>
> All that explained, I figured I might as well ask before I attempted a solution to that myself: as much as populate(), bind seems to be one of the overly complicated callbacks, designed for a scenario in which everything can come and go at will, which is something we're trying to fix.
>

As we aim for single hierarchy, it defenitely should be removed.
