Subject: Re: Re: containers and cgroups mini-summit @ Linux Plumbers
Posted by ebiederm on Thu, 26 Jul 2012 19:19:12 GMT
View Forum Message <> Reply to Message

Tejun Heo <tj@kernel.org> writes:

> Hello, Eric.
>
> On Thu, Jul 26, 2012 at 03:42:50AM -0700, Eric W. Biederman wrote:
>> - Create a new mount namespace.
>> - Create fresh mounts of all of the control groups like I would do at
>>   boot, with no consideration to any other control group state.
>> - Start forking processes.
>>
>> The expected semantics would be something like chroot for control
>> groups, where all of the control groups that are created by fresh mounts
>> are relative to whatever state the process of being in a control group
>> that the process that mounted them was in.
>
> No, any attempt to build namespace support into cgroup core code will
> be nacked with strong prejudice.

The cgroup code was only merged with the understanding that this support
was simple to add and it would be added. I am sorry that no one had
the sense to follow up and make certain that promise was not fullfilled.

> I still think it was a mistake to add that to sysfs.

sysfs fundamentally can not represent all of the network devices
in the hierarchy of objects that it chose.

sysfs does not have namespace hacks. Sysfs has hacks for the
limitations of the hiearchary of devices that was choosen
for a sysfs user space ABI.

> Thankfully, procfs is going the FUSE way.

No procfs is not going the FUSE way. Hacks for programs that misuse
information in procfs is going the FUSE way.

The best example is there currently is not a good method for programs
to figure out how parellel it is productive to be so the programs
read /proc/cpuinfo and get the count of cpus. Control groups can
limit you to fewer cpus but those programs have figured that out yet.

But ultimately fuse for procfs is about the rare case where people
want to lie to applications, because it is easier to lie to applications
then to disabuse the applications of their mistaken asumptions.

I have not seen a single suggest that any of the other procfs bits
can go away.

> and I hope in time we could convert sysfs to a similar mechanism and
> deprecate the in-kernel support.

I have nothing that even suggests there is a reasonable possibility of
using fuse to deprecate any of the proc or sysfs support.

> So, no, no, no, no, no, no, no, no, no, no, no, no. :P

Bahahahahahahaha! :P

I sort of wish I had the energy to tackle this. As it is control groups
hierarchies have very severe usablilty problems supporting one of their
core use cases.

We should have our interfaces designed such that it is possible to run
nested init's without hacks, and the only significant piece left on the
hacks pile is control groups.

Control group hiearchies are are really strange piece of work whose
design makes very little sense to me.

I think all I want from control groups is that a process that is bound
into a control group hiearchy when it mounts that hiearchy will get not
the normal control group root but instead the dentry of the directory
for that processes place in the control group hiearchy.

What is that maybe 15-30 lines of code to look up the right dentry?

Eric
