Subject: Re: [RFC] Prefixing cgroup generic control filenames with "cgroup."
Posted by Paul Jackson on Thu, 28 Feb 2008 23:36:18 GMT
View Forum Message <> Reply to Message

Paul M wrote:
> But control files provided by
> cgroups itself have no prefix. Currently that set of files is just
> "tasks", "notify_on_release" and "release_agent", but that set is
> likely to expand in the future. To reduce the risk of clashes, it
> would make sense to prefix these files and any future ones with the
> "cgroup." prefix.

I don't see the mixing of kernel generated filenames with user generated names to be a practical problem. There just aren't that many names in play here.

I'd think that renaming even the few cgroup files that were published in 2.6.24 would be a fairly unacceptable incompatibility.

Paul M wrote:
> The point would be to avoid situations where a user has code that
> creates a group directory called "foo", and then in a future kernel
> release cgroups introduces a control file called "foo".

We could accomplish that much by decreeing that future new kernel generated names that we might add follow some stronger convention, such as the cgroup_ or appropriate subsystem prefix. No need to change the existing well known names for this reason.

Serge wrote:
> To me it seems quite logical that files belonging to the cgroup
> subsystem would have no prefix, and I don't see any good reason to
> do so.

Agreed.

Paul M wrote:
> It wouldn't be hard to throw that away and move all the user-created
> group directories into their own subdirectory, i.e. change the
> existing directory layout from something like:

Yuck. You're complicating this more than necessary, to solve a problem that exists only in your imagination ;). Simple, overlapping, namespaces really are ok, so long as the number of distinct names and the rate of their growth are sufficiently low.

> But I don't know what other users might want to do.

Just set some convention for future added names; that's enough to enable
others adding user space names to avoid collisions. Such as using only
lower case letters and underscores, or the cgroup_ or other prefix
on -future- names. That's sufficient, if not more than sufficient.

> But I still don't see any argument *against*
> doing the prefixing automatically (rather than an argument that it's
> no better or worse) other than wanting 2.6.24 compatibility.

That's sufficient reason. Actually, in terms of 'common names used
by humans' some of these names, "tasks" and "notify_on_release", date
back much earlier than that. Please don't rename these two files in
cgroups; and of course absolutely don't rename them in cpusets.

Please don't end up with different names of these files, depending on
whether you're in cgroups or cpusets, either.

Andrew, looking at a tree that Paul M drew, wrote:
> That looks nice.

Not to me ;) Yuck.

Andrew wrote:
> That doesn't. It sounds like cpusets legacy has mucked us up here?
>
> Could we do something like auto-prefixing user-created directories with a
> fixed string so that there is no way in which the user can cause a
> collision with kernel-created files?

Lordy lordy -- a bunch of intrusive, complicating crap to solve a
non-existent problem (sorry for the indelicate choice of words ;).

> So yet another option would be to promise to prefix all _future_
> kernel-generated files with "kern_", ...

that could work

> ... and to change the implementation now
> to reject any user-created files which start with "kern_". hm.

Unnecessary complication. No need to burden our children and grand children
with this special case, forever after, in order to solve some empty set of
special cases currently facing us.

> > It wouldn't be hard to throw that away and move all the user-created
> > group directories into their own subdirectory, i.e. change the
> > existing directory layout from something like:
> ...

> I'll put a patch together for consideration.

I'll keep a bucket of ice water handy, for that patch ;). One nice property of the cpuset file system is that there is exactly one directory per cpuset. The kernel creates regular files; the user creates directories; no exceptions. I encourage us to keep that state of affairs, for both cgroups and cpusets.

> > If so, we could do the prefixing only for non-cpusets directories,
> > but that's getting a bit weird.
>
> We already have something like that in place, actually.

I probably won't insist on a full force NAK so long as cpusets are unchanged, thanks to such compatibility measures; though I would be tempted to ... if I could think of a sufficient reason.

Human beings really don't have problems with overlapping name spaces like this; to them it seems simpler in many cases, such as this one.

This has been, in my (biased and less than humble) view, one of the successes of the cpuset interface. It's just enough to comfortably do what's needed; not some overly formalized and unnecessarily robust complication beyond what's practically needed. More people can comfortably understand it this way.

Design interfaces, and write code, for humans. Resists making concessions to the limitations of computers (or our fellow geeks) as best you can, whenever you can.

--

I won't rest till it's the best ...
Programmer, Linux Scalability
Paul Jackson <pj@sgi.com> 1.940.382.4214