Subject: Re: [RFC][PATCH] introduce task cgroup (#task restrictioon for prevent fork bomb by cgroup)
Posted by Paul Menage on Sat, 07 Jun 2008 09:12:49 GMT
View Forum Message <> Reply to Message

On Thu, Jun 5, 2008 at 2:52 PM, Daniel Hokka Zakrisson <daniel@hozac.com> wrote:
> Would it make more sense to implement this as part of an rlimit subsystem,
> which also supports limiting e.g. address space, CPU time, number of open
> files, etc.? If we create one subsystem per resource, I'm afraid we're
> going to see quite some time spent in all those loops, and the options for
> cgroupfs is going to become pretty long if you want to exclude just one or
> two of the subsystems for one particular mount point.
>

In general, most of the cgroup loops that depend on the number of
available subsystems are control path (fork/exit/task move) rather
than data path.

If these loops ever did become an issue it wouldn't be hard to cache
which subsystems had callbacks of a given type, and thus make the loop
overheap be linear in the number of subsystems actually using a given
callback, rather than the number of registered subsystems. I've not
done this yet since it didn't seem worth the complexity, but if the
number of subsystems grows from what we have now it might be
worthwhile.

Having a single rlimit subsystem seems a bit inflexible, since it
would require anyone that wanted to track one resource to have to pay
the overhead of tracking all resources. By splitting into separate
subsystems, it would be possible to just pay the overhead for the
resource tracking that you actually care about.

You mention that you think it makes the set of mount arguments more
complex, but if you have middleware that can limit resource X, adding
X to the list of subsystems at the point when a cgroupfs is mounted
seems simple compared to the rest of the structure required to actual
set limits and monitor usage for X.

Paul

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