
Subject: Re: [PATCH 4/4] get rid of populate for memcg
Posted by [Tejun Heo](#) on Wed, 21 Mar 2012 16:06:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello, Glauber.

On Wed, Mar 21, 2012 at 11:36:19AM +0400, Glauber Costa wrote:

> On 03/20/2012 10:31 PM, Tejun Heo wrote:
> >Hello, Glauber.
> >
> >On Tue, Mar 20, 2012 at 08:50:56PM +0400, Glauber Costa wrote:
> >>@@ -4929,7 +4929,9 @@ mem_cgroup_create(struct cgroup *cont)
> >> atomic_set(&memcg->refcnt, 1);
> >> memcg->move_charge_at_immigrate = 0;
> >> mutex_init(&memcg->thresholds_lock);
> >>- return&memcg->css;
> >>+
> >>+ if (!register_kmem_files(memcg,&mem_cgroup_subsys))
> >>+ return&memcg->css;
> >
> >After the change, I think register_kmem_files() is a quite misleading
> >name.
>
> how about init_kmem() ?
>
> Remember the slab bits will be likely to end up here as well in the end.

I don't know. Whatever which describes what's going on.
memcg_init_kmem()?

> >So, init_cgroup() is overloaded to do two things - one load time init
> >and per-cgroup init, depending on the args.
>
> Yes. I don't love it, but there is quite a bunch of precedents for this.
> Like the shrinkers in vmscan, for instance.
>
> a NULL argument is a probe, a valid argument should have action taken.

Please don't. Just add a new callback if necessary.

> >What I don't get is why you can't just keep this. Is it because the
> >files might appear before the protocol is registered? Wouldn't it be
> >much better to add ipv4_tcp_init_cgroup() or whatever call to
> >inet_init() instead of overloading init_cgroup() with mostly unrelated
> >stuff?
> >
>
> The reason is that this has to be kept generic for protocols that

> may want to implement this in the future - since the pressure
> controls themselves are generic, the per-cgroup versions should be
> as well.
>
> And in general, a protocol can live in a module, or not be registered
> despite being compiled in.

Hmmmm... yeah, CGROUP_SUBSYS_CFTYPES() would register the files on module load but won't unregister them on unload. Will fix that. However, the fact that files living in modules shouldn't be a problem in itself. If those file handlers can cope with protocol not being registered yet, everything should be fine.

> Now, what we do with the files, are our decision in the end. If you
> want, we can use CGROUP_SUBSYS_CFTYPES(mem_cgroup_subsys, tcp_files)
> as you suggested. tcp itself is always available if it is compiled in.
> Then in the future, if anyone cares about adding support for a
> protocol that may differ in that aspect, we can put the files
> nevertheless, and
> use ENOTSUPP as kame suggested for the swap accounting.

I don't quite get why a protocol module would be loaded but not registered. Do we actually have cases like that? I know it's mechanically possible but don't think there's any actual use case or existing code which does that, so no need to worry about them.

Thanks.

--

tejun
