Subject: Re: [RFC][PATCH] introduce task cgroup (#task restrictioon for prevent fork bomb by cgroup)
Posted by KOSAKI Motohiro on Sat, 07 Jun 2008 06:46:56 GMT
View Forum Message <> Reply to Message

Hi

> Hi Kosaki,
>
> The basic idea of a task-limiting subsystem is good, thanks.

Thanks.

```
> > -void cgroup_fork(struct task_struct *child)
> > +int cgroup_fork(struct task_struct *child)
> >  {
> > +       int i;
> > +       int ret;
> > +
> > +       for (i = 0; i < CGROUP_SUBSYS_COUNT; i++) {
> > +               struct cgroup_subsys *ss = subsys[i];
> > +               if (ss->can_fork) {
> > +                       ret = ss->can_fork(ss, child);
> > +                       if (ret)
> > +                               return ret;
> > +               }
> > +       }
> > +
> >         task_lock(current);
> >         child->cgroups = current->cgroups;
> >         get_css_set(child->cgroups);
> >         task_unlock(current);
> >         INIT_LIST_HEAD(&child->cg_list);
> > +
> > +       return 0;
> >  }
```
>
> I don't think this is the right way to handle this check. This isn't a
> generic control groups callback, it's one that specific for a
> particular subsystem. So the right way to handle it is to call
> task_cgroup_can_fork() from the same place that the RLIM_NPROC limit
> is checked.
>
> If it later turned out that multiple cgroup subsystems wanted to be
> able to prevent forking, then it might make sense to have a generic
> cgroup callback, but for just one subsystem it's cleaner to call
> directly.

OK.

```
> > +static int task_cgroup_populate(struct cgroup_subsys *ss,
> > +                               struct cgroup *cgrp)
> > +{
> > +       if (task_cgroup_subsys.disabled)
> > +               return 0;
>
```

> I don't think you should need this check - if the subsystem is
> disabled, it'll never be mounted in the first place.

to be honest, I did copy&past it from memcontrol.c ;)
Thanks good opinion.