
Subject: Re: [RFC] [PATCH] cgroup: add "procs" control file

Posted by [Li Zefan](#) on Sat, 21 Jun 2008 06:20:16 GMT

[View Forum Message](#) <> [Reply to Message](#)

Paul Menage wrote:

> On Wed, Jun 18, 2008 at 1:02 AM, Li Zefan <lizf@cn.fujitsu.com> wrote:

>> - What to do if the attaching of a thread failed? continue to attach

>> other threads, or stop and return error?

>

> I think this is something that will have to be handled in the design

> of transactional cgroup attach.

>

Is the following proposal feasible?

- call `can_attach()` for each thread before attaching them into the new group.

This works for `cpuset`, doesn't it?

- the above may not always be reasonable, for example for Kosaki-san's task cgroup.

in this case, we require the subsystem to provide a `can_attach_thread_group()`,

like:

```
static int task_cgroup_can_attach_group(struct cgroup_subsys *ss,
    struct cgroup *cgrp, struct task_struct *tsk)
```

```
{
    struct task_cgroup *taskcg = task_cgroup_from_cgrp(cgrp);
```

```
    struct task_struct *t;
```

```
    int ret = 0;
```

```
    int nr_threads = 1;
```

```
    for (t = next_thread(tsk); t != tsk; t = next_thread(t)
```

```
        nr_threads++;
```

```
    spin_lock(&taskcg->lock);
```

```
    if (taskcg->nr_tasks + nr_threads > taskcg->max_tasks)
```

```
        ret = -EBUSY;
```

```
    spin_unlock(&taskcg->lock);
```

```
    return ret;
```

```
}
```

>> - When a sub-thread of a process is in the cgroup, but not its thread

>> cgroup leader, what to do when 'cat procs'? just skip those threads?

>

> Sounds reasonable. I think that in general the procs file is more

> useful as a write API than a read API anyway, for the reasons you

> indicate there.

>

>

>> + tsk = attach_get_task(cgrp, pidbuf);

```
>> +    if (IS_ERR(tsk))
>> +        return PTR_ERR(tsk);
>> +
>> +    /* attach thread group leader */
>
> Should we check that this is in fact a thread group leader?
>
```

No need actually, I added this check originally and then removed it, but forgot to remove the comment.

```
>> +
>> +    /* attach all sub-threads */
>> +    rcu_read_lock();
>
> cgroup_attach_task() calls synchronize_rcu(), so it doesn't seem
> likely that rcu_read_lock() is useful here, and might even deadlock?
>
> What are you trying to protect against with the RCU lock?
>
```

Ah yes, bad here. I am trying to protect the thread list.

```
>>    {
>> +        .name = "procs",
>
> Maybe call it "cgroup.procs" to avoid name clashes in future? We had a
> debate a while back where I tried to get the cgroup files like "tasks"
> and "notify_on_release" prefixed with "cgroup." , which were argued
> against on grounds of backwards compatibility. But there's no
> compatibility issue here. The only question is whether it's too ugly
> to have the legacy filenames without a prefix and the new ones with a
> prefix.
>
```

Yes it's ugly.. Is possible name clash of "procs" a kind of breaking compatibility that should be avoid in any case?

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