Subject: Re: [RFC] Default child of a cgroup
Posted by Paul Menage on Fri, 01 Feb 2008 02:39:56 GMT
View Forum Message <> Reply to Message

On Jan 30, 2008 6:40 PM, Srivatsa Vaddagiri <vatsa@linux.vnet.ibm.com> wrote:

```
>
> Here are some questions that arise in this picture:
>
> 1. What is the relationship of the task-group in A/tasks with the
>    task-group in A/a1/tasks? In otherwords do they form siblings
>    of the same parent A?
```

I'd argue the same as Balbir - tasks in A/tasks are are children of A and are siblings of a1, a2, etc.

```
>
> 2. Somewhat related to the above question, how much resource should the
>    task-group A/a1/tasks get in relation to A/tasks? Is it 1/2 of parent
>    A's share or 1/(1 + N) of parent A's share (where N = number of tasks
>    in A/tasks)?
```

Each process in A should have a scheduler weight that's derived from its static_prio field. Similarly each subgroup of A will have a scheduler weight that's determined by its cpu.shares value. So the cpu share of any child (be it a task or a subgroup) would be equal to its own weight divided by the sum of weights of all children.

So yes, if a task in A forks lots of children, those children could end up getting a disproportionate amount of the CPU compared to tasks in A/a1 - but that's the same as the situation without cgroups. If you want to control cpu usage between different sets of processes in A, they should be in sibling cgroups, not directly in A.

Is there a restriction in CFS that stops a given group from simultaneously holding tasks and sub-groups? If so, couldn't we change CFS to make it possible rather than enforcing awkward restructions on cgroups?

If we really can't change CFS in that way, then an alternative would be similar to Peter's suggestion - make cpu_cgroup_can_attach() fail if the cgroup has children, and make cpu_cgroup_create() fail if the cgroup has any tasks - that way you limit the restriction to just the hierarchy that has CFS attached to it, rather than generically for all cgroups

BTW, I noticed this code in cpu_cgroup_create():

```
        /* we support only 1-level deep hierarchical scheduler atm */
        if (cgrp->parent->parent)
                return ERR_PTR(-EINVAL);
```

Is anyone working on allowing more levels?

Paul

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