Subject: Re: [RFC] Default child of a cgroup
Posted by Srivatsa Vaddagiri on Fri, 01 Feb 2008 08:17:18 GMT

View Forum Message <> Reply to Message

On Thu, Jan 31, 2008 at 06:39:56PM -0800, Paul Menage wrote:
> 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.

Assuming all tasks are of same prio, then what you are saying is that
A/a1/tasks should cumulatively recv 1/(1 + N) of parent's share.

After some thought, that seems like a reasonable expectation. The only issue
I have for that is it breaks current behavior in mainline. Assume this
structure:

```
   /
   |------<tasks>
   |------<cpuacct.usage>
    |------<cpu.shares>
   |
   |----[A]
   |     |----<tasks>
   |     |----<cpuacct.usage>
   |     |----<cpu.shares>
```

then, going by above argument, /A/tasks should recv 1/(1+M)% of system
resources (M -> number of tasks in /tasks), whereas it receives 1/2 of
system resources currently (assuming /cpu.shares and /A/cpu.shares are

same).

Balbir, is this behaviour same for memory controller as well?

So pick any option, we are talking of deviating from current
behavior, which perhaps is a non-issue if we want to DTRT.

> 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?

Should be possible, need to look closely at what will need to change
(load_balance routines for sure).

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