

On 10/23/07, Srivatsa Vaddagiri <vatsa@linux.vnet.ibm.com> wrote:

> > Suppose you have two cgroups that would each want to use, say, 55% of
> > a CPU - technically they should each be regarded as having 45% idle
> > time, but if they run on a the same CPU the chances are that they will
> > both always have some processes on their runqueue due to contention
> > with the other group. So how would you measure the difference between
> > this and a cgroup that really is trying to use 100%?

>
> Good point. I think we need to subtract out the time it was waiting on runqueue
> when calculating idle time.

>
> |----- -----ZZZZZZZZZZZZ.....-----|
> t0 t1 t2 t3 t4 t5 t6

>
>
> ---- -> Running time
> -> Waiting time (to get on the cpu)
> zzzz -> Sleeping time (when it didnt want to run because of
> lack of tasks)

> So, in this case,

>
> idle time = (t4 - t3) / [(t6 - t1) - (t2-t1) - (t5-t4)
>

Do you mean (t6 - t0) where you have (t6 - t1)?

> ?

>
> This idle time will be a per-cpu stat for every cgroup and needs to be
> consolidated across cpus into a single idle-stat number, just like how
> top does it.

This would be an idle fraction, not an idle time. (seconds divided by seconds)

It doesn't seem quite right to me that a cgroup's idle time metric be
affected by the activity of other cgroups on the machine, but it's
hard to come up with a way of measuring it that doesn't have this
behaviour - which is why, in the absence of hard CPU partitioning,
it's not clear to me how much use this would be.

What would people be planning to use it for?

Paul

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