Subject: Re: [PATCH v2 5/5] expose per-taskgroup schedstats in cgroup
Posted by Sha Zhengju on Wed, 18 Apr 2012 14:44:25 GMT
View Forum Message <> Reply to Message

On Mon, Apr 9, 2012 at 6:25 PM, Glauber Costa <glommer@parallels.com> wrote:
> This patch aims at exposing stat information per-cgroup, such as:
> * idle time,
> * iowait time,
> * steal time,
> * # context switches
> and friends. The ultimate goal is to be able to present a per-container view of
> /proc/stat inside a container. With this patch, everything that is needed to do
> that is in place, except for number of tasks.
>
> For most of the data, I achieve that by hooking into the schedstats framework,
> so although the overhead of that is prone to discussion, I am not adding anything,
> but reusing what's already there instead. The exception being that the data is
> now computed and stored in non-task se's as well, instead of entity_is_task() branches.
> However, I expect this to be minimum comparing to the alternative of adding new
> hierarchy walks. Those are kept intact.
>
> The format of the new file added is the same as the one recently
> introduced for cpuacct:
>
> cpu0.idle X
> cpu0.steal Y
> ...
> cpu1.idle X1
> cpu1.steal Y1
> ...
>

You define the idle time as the sum of task's sleeping time which i
think it needs to
discuss. It changes the current meaning of idle time which might be confusing
and can not reflect the the actual cpu busy-idle situation. IMHO, idle
time can just
be the true system value. Personally I prefer to your last version in
the way of computing
idle time (http://thread.gmane.org/gmane.linux.kernel/1194838). And
iowait can be
computed in the similar way.

As to steal time, "Steal time is the percentage of time a virtual CPU
waits for a real
CPU while the hypervisor is servicing another virtual processor".
Speaking from the
point of view of resource controlling(isolation), cgroup is a

lightweight method towards
virtualiztion, so I think obeying its primitive meaning is more
appropriate: the time not
servicing me including time stolen by the tasks of other cgroup.