Subject: Re: [PATCH v2 5/5] expose per-taskgroup schedstats in cgroup
Posted by Sha Zhengju on Thu, 19 Apr 2012 13:30:04 GMT
View Forum Message <> Reply to Message

On 04/19/2012 12:24 AM, Glauber Costa wrote:
>
>>
>> You define the idle time as the sum of task's sleeping time which i
>> think it needs to
>> discuss.
>
> Where is it done ?
>
> Idle time here is measured as the time between enqueue_sleeper() and
> the group being put back in the rq.
> But note it is enqueue sleeper for the group, not any tasks.
>

Sorry, I still do not catch the point. In enqueue_sleeper(), it sums up
the sleep
time to se.statistics since dequeue_sleeper(), and then put back to rq.
Here do
you mean idle time is measured as the time between dequeue_sleeper() and
enqueue_sleeper()? But it's still the sum of sleeping time of the
group's task?
Not cfs expert. If I've miss something, please feel free to point it
out. :-)

> cfs will call this callback until it finds anything that is running
> (task or not a task).
>
> Maybe I made some mistake in the code - and in this case, please point
> out - but that's the idea.
>
>> 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.
>
> No. The idea that idle time can only be true system-wide is wrong. As a
> matter of fact, that first series of mine is totally wrong wrt that
> (and then I changed).
>
> A cgroup is idle when none of its tasks are in the runqueue. What is
> the problem that you see with this?

Actually, both idle and steal are the time when the group don't work.
IMO, i'd like to contribute
the real cpu idle time to a group's idle, and let the time cpu servicing
for other group to be steal time.
For example, suppose that 2 tasks(groups) are sharing one cpu and #1
keep running while #2 keep sleeping,
in your way: #1(idle)=0, #1(steal)=0; #2(idle)=100%, #2(steal)=0;
in my way: #1(idle)=0, #1(steal)=0; #2(idle)=0,   #2(steal)=100%;
IMHO, our opinions diverge from the meaning of "idle". But both idle and
steal can be get from cpuacct
in my way without involving in cpu controller.

>
> As for iowait, that one seemed a bit trickier, so we decided to leave
> it out at least for now.
>
>>
>> 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.
>
> And that's exactly what I've done.
>
> Steal time is runqueue time, until you are chosen to run.
>
> In a summary: If you are not running, you can be either idle or stolen.
> if you are in the runqueue, you are stolen.
> If you are not, you are idle.