Subject: Re: [PATCH v2 5/5] expose per-taskgroup schedstats in cgroup
Posted by Glauber Costa on Thu, 19 Apr 2012 15:00:37 GMT
View Forum Message <> Reply to Message

On 04/19/2012 10:30 AM, Sha Zhengju wrote:
> 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.
Yes.

> Here do
> you mean idle time is measured as the time between dequeue_sleeper() and
> enqueue_sleeper()?

In general, yes. In practice, when we read the field, a
dequeue_sleeper() may not yet have happened. So we need to sum whatever
is in the rq at the moment to it.

But it's still the sum of sleeping time of the
> group's task?
No.

cfs will walk the hierarchy up calling enqueue_sleeper until it finds a
group that is not sleeping. This means that enqueue_sleeper() will be
called when no tasks in the group are running. (actually, no subgroups,
since it is hierarchical).

Take a look at sched/core/fair.c

>>> 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.
The difference is why it doesn't work. If he doesn't want to work,
that's idle. If it can't work, that's steal time.

> 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.

"contribute the real idle time to a group's idle" doesn't make any
sense. If all groups are idle, they all passed through
enqueue_sleeper(), and that time is already counted as idle. For all of
them.

About steal time: That's *exactly* what I am doing! When a group enters
the runqueue, it should run. If it doesn't run, that's because someone
else is running. Therefore, runqueue time == 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%;

Have you actually tested this?
It depends on what you mean by "keep sleeping". If you mean sleeping as
not having any work to do, of course it is idle time.

Believing this is steal time just because another group exists in the
system is just wrong.

> IMHO, our opinions diverge from the meaning of "idle".

Yes, I believe idle time is the time during which you are idle.

> But both idle and
> steal can be get from cpuacct
> in my way without involving in cpu controller.

How so? If you wait for the idle tick to happen, that will mean *ALL*

your groups are idle. And that is *not* how you measure idle time.

Idle time of a group of tasks, is the time during which none of the tasks are running.