Subject: Re: [PATCH] introduce task cgroup v2
Posted by KOSAKI Motohiro on Sat, 21 Jun 2008 17:01:12 GMT
View Forum Message <> Reply to Message

>> I am going to convert spinlock in task limit cgroup to atomic_t.
>> task limit cgroup has following caractatics.
>>        - many write (fork, exit)
>>        - few read
>>        - fork() is performance sensitive systemcall.
>
> This is true, but I don't see how it can be more performance-sensitive
> than the overhead of allocating/freeing a page.
>
> What kinds of performance regressions did you see?

I ran spawn test of unix bench, thus

implement way              performance degression
-----------------------------------------------------------------------
use res_counter            15-20%
use spin_lock()             nealy 10%
use atomic_t               nealy 5%

Yes, this is really roughly number.
Of cource, I'll post more detail result at next week.


>>        if increase fork overhead, system total performance cause degression.
>
> What kind of overhead were you seeing? How about if you delay doing
> any task accounting until the task_limit subsystem is bound to a
> hierarchy? That way there's no noticeable overhead for people who
> aren't using your subsystem.

honestly, I am seeing it on micro-benchmark only.
but, I'm afraid to performance degression because many people check
performance regression periodically.
So if my implement cause performance regression, they never used mine.

Or, if you strongly want to task_limit subsystem use res_counter,
I can be working on improve to res_counter performance instead.
_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers