Subject: Re: Re: [RFC][PATCH 1/7] Resource counters
Posted by dev on Tue, 13 Mar 2007 09:36:08 GMT
View Forum Message <> Reply to Message

>> - doesn't store the accounted value but
>>   limit - accounted (i.e. the free resource)
>> - uses atomic_add_return()
>> - when negative, an error is returned and
>>   the resource amount is added back
>>
>>changes to the limit have to adjust the 'current'
>>value too, but that is again simple and atomic
>>
>>best,
>>Herbert
>>
>>PS: atomic_add_unless() didn't exist back then
>>(at least I think so) but that might be an option
>>too ...
>
>
> I think as far as having this discussion if you can remove that race
> people will be more willing to talk about what vserver does.
>
> That said anything that uses locks or atomic operations (finer grained locks)
> because of the cache line ping pong is going to have scaling issues on large
> boxes.

> So in that sense anything short of per cpu variables sucks at scale. That said
> I would much rather get a simple correct version without the complexity of
> per cpu counters, before we optimize the counters that much.
fully agree with it. We need to get a working version first.

FYI, in OVZ we recently added such optimizations: reserves like in TCP/IP,
e.g. for kmemsize, numfile these reserves are done on task-basis for
fast charges/uncharges w/o involving lock operations.
On task exit reserves are returned back to the beancounter.

As it demonstrated atomic counters can be replaced with
task-reserves on the next step.

Thanks,
Kirill