Subject: Re: [PATCH v2 3/5] change number_of_cpusets to an atomic
Posted by Glauber Costa on Tue, 24 Apr 2012 16:30:05 GMT

View Forum Message <> Reply to Message

On 04/24/2012 01:24 PM, Christoph Lameter wrote:
> On Tue, 24 Apr 2012, Glauber Costa wrote:
>
>>> Would this not also be a good case to introduce static branching?
>>>
>>> number_of_cpusets is used to avoid going through unnecessary processing
>>> should there be no cpusets in use.
>>
>> static branches comes with a set of problems themselves, so I usually prefer
>> to use them only in places where we don't want to pay even a cache miss if we
>> can avoid, or a function call, or anything like that - like the slub cache
>> alloc as you may have seen in my kmem memcg series.
>>
>> It doesn't seem to be the case here.
>
> How did you figure that? number_of_cpusets was introduced exactly because
> the functions are used in places where we do not pay the cost of calling
> __cpuset_node_allowed_soft/hardwall. Have a look at these. They may take
> locks etc etc in critical allocation paths
I am not arguing that.

You want to avoid the cost of processing a function, that's fair.
(Note that by "function call cost" I don't mean the cost of processing a
function, but the cost of a (potentially empty) function call.)
The real question is: Are you okay with the cost of a branch + a global
variable (which is almost read only) fetch?

The test of a global variable can - and do as of right now - avoid all
the expensive operations like locking, sleeping, etc, and if you don't
need to squeeze every nanosecond you can, they are often simpler - and
therefore better - than static branching.

Just to mention one point I am coming across these days - that initiated
all this: static patching holds the cpu_hotplug.lock. So it can't be
called if you hold any lock that has been already held under the
cpu_hotplug.lock. This will probably mean any lock the cpuset cgroup
needs to take, because it is called - and to do a lot of things - from
the cpu hotplug handler, that holds the cpu_hotplug.lock.

So if if were a case of simple static branch usage, I am not opposed to
it. But I foresee it getting so complicated, that a global variable
seems to do the job we need just fine.