Subject: Re: [PATCH 11/33] task containersv11 make cpusets a client of containers
Posted by Paul Menage on Thu, 04 Oct 2007 15:16:34 GMT
View Forum Message <> Reply to Message

On 10/4/07, Paul Jackson <pj@sgi.com> wrote:
> Paul M,
>
> This snippet from the memory allocation hot path worries me a bit.
>
> Once per memory page allocation, we go through here, needing to peak inside
> the current tasks cpuset to see if it has changed (it's 'mems_generation'
> value doesn't match the last seen value we have stashed in the task struct.)
>
> @@ -653,20 +379,19 @@ void cpuset_update_task_memory_state(voi
> struct task_struct *tsk = current;
> struct cpuset *cs;
>
> - if (tsk->cpuset == &top_cpuset) {
> + if (task_cs(tsk) == &top_cpuset) {
> /* Don't need rcu for top_cpuset. It's never freed. */
> my_cpusets_mem_gen = top_cpuset.mems_generation;
> } else {
> rcu_read_lock();
> - cs = rcu_dereference(tsk->cpuset);
> - my_cpusets_mem_gen = cs->mems_generation;
> + my_cpusets_mem_gen = task_cs(current)->mems_generation;
> rcu_read_unlock();
> }
>
> With this new cgroup code, the task_cs macro was added, -twice-,
> which deals with the fact that what used to be a single pointer
> in the task struct directly to the tasks cpuset is now roughly
> two more dereferences and an indexing away:

It's two constant-indexed dereferences *in total*, compared to a
single constant-indexed dereference in the pre-cgroup case.

The cpuset pointer is found at
task->cgroups->subsys[cpuset_subsys_id], where cpuset_subsys_id is a
compile-time constant.

>
> At a minimum, could you change that last added line to use 'tsk'
> instead of 'current'? This should save one instruction, as 'tsk'
> will likely already be in a register.

Sounds reasonable.

>
> I wonder if we can save any cache line hits on this, or if there is
> someway to measure whether or not this has noticeable performance
> impact.

I didn't notice any performance hit on a pure allocate/free memory
benchmark relative to non-cgroup cpusets. (There was a small
performance hit relative to not using cpusets at all, but that was to
be expected).

Paul

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers