Subject: Re: [PATCH 11/33] task containersv11 make cpusets a client of containers
Posted by Paul Jackson on Thu, 04 Oct 2007 09:53:55 GMT
View Forum Message <> Reply to Message

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:

```
    static inline struct cpuset *task_cs(struct task_struct *task)
    {
     return container_of(task_subsys_state(task, cpuset_subsys_id),
    struct cpuset, css);
    }

    static inline struct cgroup_subsys_state *task_subsys_state(
     struct task_struct *task, int subsys_id)
    {
     return rcu_dereference(task->cgroups->subsys[subsys_id]);
    }
```

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.

+ my_cpusets_mem_gen = task_cs(tsk)->mems_generation;

I guess the two, rather than one, invocations of task_cs() won't matter much, as they are on the same address, so the second invocation will hit cache lines just found on the first invocation.

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.

... Probably this is all lost in the noise of the other stuff that gets coded in the memory allocation hot path. It would be nice to think that it actually matters however.

--

I won't rest till it's the best ...
Programmer, Linux Scalability
Paul Jackson <pj@sgi.com> 1.925.600.0401

_____