Subject: Re: [PATCH v5 12/14] execute the whole memcg freeing in free_worker
Posted by KAMEZAWA Hiroyuki on Wed, 17 Oct 2012 06:56:33 GMT
View Forum Message <> Reply to Message

(2012/10/16 19:16), Glauber Costa wrote:
> A lot of the initialization we do in mem_cgroup_create() is done with
> softirqs enabled. This include grabbing a css id, which holds
> &ss->id_lock->rlock, and the per-zone trees, which holds
> rtpz->lock->rlock. All of those signal to the lockdep mechanism that
> those locks can be used in SOFTIRQ-ON-W context. This means that the
> freeing of memcg structure must happen in a compatible context,
> otherwise we'll get a deadlock, like the one bellow, caught by lockdep:
>
> [<ffffffff81103095>] free_accounted_pages+0x47/0x4c
> [<ffffffff81047f90>] free_task+0x31/0x5c
> [<ffffffff8104807d>] __put_task_struct+0xc2/0xdb
> [<ffffffff8104dfc7>] put_task_struct+0x1e/0x22
> [<ffffffff8104e144>] delayed_put_task_struct+0x7a/0x98
> [<ffffffff810cf0e5>] __rcu_process_callbacks+0x269/0x3df
> [<ffffffff810cf28c>] rcu_process_callbacks+0x31/0x5b
> [<ffffffff8105266d>] __do_softirq+0x122/0x277
>
> This usage pattern could not be triggered before kmem came into play.
> With the introduction of kmem stack handling, it is possible that we
> call the last mem_cgroup_put() from the task destructor, which is run in
> an rcu callback. Such callbacks are run with softirqs disabled, leading
> to the offensive usage pattern.
>
> In general, we have little, if any, means to guarantee in which context
> the last memcg_put will happen. The best we can do is test it and try to
> make sure no invalid context releases are happening. But as we add more
> code to memcg, the possible interactions grow in number and expose more
> ways to get context conflicts. One thing to keep in mind, is that part
> of the freeing process is already deferred to a worker, such as vfree(),
> that can only be called from process context.
>
> For the moment, the only two functions we really need moved away are:
>
> * free_css_id(), and
> * mem_cgroup_remove_from_trees().
>
> But because the later accesses per-zone info,
> free_mem_cgroup_per_zone_info() needs to be moved as well. With that, we
> are left with the per_cpu stats only. Better move it all.
>
> Signed-off-by: Glauber Costa <glommer@parallels.com>
> Tested-by: Greg Thelen <gthelen@google.com>
> Acked-by: Michal Hocko <mhocko@suse.cz>

> CC: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>
> CC: Johannes Weiner <hannes@cmpxchg.org>
> CC: Tejun Heo <tj@kernel.org>

Acked-by: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>