
Subject: Re: [PATCH v5 2/2] decrement static keys on real destroy time

Posted by [Tejun Heo](#) on Mon, 14 May 2012 18:12:50 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Fri, May 11, 2012 at 05:11:17PM -0300, Glauber Costa wrote:

> We call the destroy function when a cgroup starts to be removed,
> such as by a rmdir event.
>
> However, because of our reference counters, some objects are still
> inflight. Right now, we are decrementing the static_keys at destroy()
> time, meaning that if we get rid of the last static_key reference,
> some objects will still have charges, but the code to properly
> uncharge them won't be run.
>
> This becomes a problem specially if it is ever enabled again, because
> now new charges will be added to the staled charges making keeping
> it pretty much impossible.
>
> We just need to be careful with the static branch activation:
> since there is no particular preferred order of their activation,
> we need to make sure that we only start using it after all
> call sites are active. This is achieved by having a per-memcg
> flag that is only updated after static_key_slow_inc() returns.
> At this time, we are sure all sites are active.
>
> This is made per-memcg, not global, for a reason:
> it also has the effect of making socket accounting more
> consistent. The first memcg to be limited will trigger static_key()
> activation, therefore, accounting. But all the others will then be
> accounted no matter what. After this patch, only limited memcgs
> will have its sockets accounted.
>
> [v2: changed a tcp limited flag for a generic proto limited flag]
> [v3: update the current active flag only after the static_key update]
> [v4: disarm_static_keys() inside free_work]
> [v5: got rid of tcp_limit_mutex, now in the static_key interface]
>
> Signed-off-by: Glauber Costa <glommer@parallels.com>
> CC: Tejun Heo <tj@kernel.org>
> CC: Li Zefan <lizefan@huawei.com>
> CC: Kamezawa Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>
> CC: Johannes Weiner <hannes@cmpxchg.org>
> CC: Michal Hocko <mhocko@suse.cz>

Generally looks sane to me. Please feel free to addmy Reviewed-by.

> + if (val == RESOURCE_MAX)
> + cg_proto->active = false;

```
> + else if (val != RESOURCE_MAX) {
```

Minor nitpick: CodingStyle says not to omit { } if other branches need them.

Thanks.

--
tejun
