

---

Subject: Re: [PATCH 10/11] memcg: allow a memcg with kmem charges to be destructed.

Posted by [KAMEZAWA Hiroyuki](#) on Tue, 26 Jun 2012 05:59:56 GMT

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

---

(2012/06/25 23:15), Glauber Costa wrote:

> Because the ultimate goal of the kmem tracking in memcg is to  
> track slab pages as well, we can't guarantee that we'll always  
> be able to point a page to a particular process, and migrate  
> the charges along with it - since in the common case, a page  
> will contain data belonging to multiple processes.  
>  
> Because of that, when we destroy a memcg, we only make sure  
> the destruction will succeed by discounting the kmem charges  
> from the user charges when we try to empty the cgroup.  
>  
> Signed-off-by: Glauber Costa <glommer@parallels.com>  
> CC: Christoph Lameter <cl@linux.com>  
> CC: Pekka Enberg <penberg@cs.helsinki.fi>  
> CC: Michal Hocko <mhocko@suse.cz>  
> CC: Kamezawa Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>  
> CC: Johannes Weiner <hannes@cmpxchg.org>  
> CC: Suleiman Souhlal <suleiman@google.com>  
> ---  
> mm/memcontrol.c | 10 ++++++++-  
> 1 file changed, 9 insertions(+), 1 deletion(-)  
>  
> diff --git a/mm/memcontrol.c b/mm/memcontrol.c  
> index a6a440b..bb9b6fe 100644  
> --- a/mm/memcontrol.c  
> +++ b/mm/memcontrol.c  
> @@ -598,6 +598,11 @@ static void disarm\_kmem\_keys(struct mem\_cgroup \*memcg)  
> {  
> if (test\_bit(KMEM\_ACCOUNTED\_THIS, &memcg->kmem\_accounted))  
> static\_key\_slow\_dec(&mem\_cgroup\_kmem\_enabled\_key);  
> + /\*  
> + \* This check can't live in kmem destruction function,  
> + \* since the charges will outlive the cgroup  
> + \*/  
> + BUG\_ON(res\_counter\_read\_u64(&memcg->kmem, RES\_USAGE) != 0);  
> }  
> #else  
> static void disarm\_kmem\_keys(struct mem\_cgroup \*memcg)  
> @@ -3838,6 +3843,7 @@ static int mem\_cgroup\_force\_empty(struct mem\_cgroup \*memcg,  
bool free\_all)  
> int node, zid, shrink;  
> int nr\_retries = MEM\_CGROUP\_RECLAIM\_RETRIES;  
> struct cgroup \*cgrp = memcg->css.cgroup;

```
> + u64 usage;
>
>   css_get(&memcg->css);
>
> @@ -3877,8 +3883,10 @@ move_account:
>     if (ret == -ENOMEM)
>       goto try_to_free;
>     cond_resched();
> + usage = res_counter_read_u64(&memcg->res, RES_USAGE) -
> + res_counter_read_u64(&memcg->kmem, RES_USAGE);
>   /* "ret" should also be checked to ensure all lists are empty. */
> - } while (res_counter_read_u64(&memcg->res, RES_USAGE) > 0 || ret);
> + } while (usage > 0 || ret);
> out:
>   css_put(&memcg->css);
>   return ret;
>
```

Hm....maybe work enough. Could you add more comments on the code ?

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

---