
Subject: Re: [PATCH v2 03/13] memcg: Uncharge all kmem when deleting a cgroup.

Posted by [Glauber Costa](#) on Sun, 11 Mar 2012 08:19:55 GMT

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

On 03/10/2012 12:39 AM, Suleiman Souhlal wrote:

> Signed-off-by: Suleiman Souhlal<suleiman@google.com>

> ---

> mm/memcontrol.c | 31 ++++++

> 1 files changed, 30 insertions(+), 1 deletions(-)

>

> diff --git a/mm/memcontrol.c b/mm/memcontrol.c

> index e6fd558..6fbb438 100644

> --- a/mm/memcontrol.c

> +++ b/mm/memcontrol.c

> @@ -382,6 +382,7 @@ static void mem_cgroup_get(struct mem_cgroup *memcg);

> static void mem_cgroup_put(struct mem_cgroup *memcg);

> static void memcg_kmem_init(struct mem_cgroup *memcg,

> struct mem_cgroup *parent);

> +static void memcg_kmem_move(struct mem_cgroup *memcg);

>

> static inline bool

> mem_cgroup_test_flag(const struct mem_cgroup *memcg, enum memcg_flags flag)

> @@ -3700,6 +3701,7 @@ static int mem_cgroup_force_empty(struct mem_cgroup *memcg,
> bool free_all)

> int ret;

> int node, zid, shrink;

> int nr_retries = MEM_CGROUP_RECLAIM_RETRIES;

> + unsigned long usage;

> struct cgroup *cgrp = memcg->css.cgroup;

>

> css_get(&memcg->css);

> @@ -3719,6 +3721,8 @@ move_account:

> /* This is for making all *used* pages to be on LRU. */

> lru_add_drain_all();

> drain_all_stock_sync(memcg);

> + if (!free_all)

> + memcg_kmem_move(memcg);

Any reason we're not moving kmem charges when free_all is set as well?

Subject: Re: [PATCH v2 03/13] memcg: Uncharge all kmem when deleting a cgroup.

Posted by [Suleiman Souhlal](#) on Tue, 13 Mar 2012 23:16:18 GMT

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

On Sun, Mar 11, 2012 at 12:19 AM, Glauber Costa <glommer@parallels.com> wrote:

> On 03/10/2012 12:39 AM, Suleiman Souhlal wrote:

```

>>
>> Signed-off-by: Suleiman Souhlal<suleiman@google.com>
>> ---
>> mm/memcontrol.c | 31 ++++++++++++++++++++++++++++++++++++++
>> 1 files changed, 30 insertions(+), 1 deletions(-)
>>
>> diff --git a/mm/memcontrol.c b/mm/memcontrol.c
>> index e6fd558..6fbb438 100644
>> --- a/mm/memcontrol.c
>> +++ b/mm/memcontrol.c
>> @@ -382,6 +382,7 @@ static void mem_cgroup_get(struct mem_cgroup *memcg);
>> static void mem_cgroup_put(struct mem_cgroup *memcg);
>> static void memcg_kmem_init(struct mem_cgroup *memcg,
>>     struct mem_cgroup *parent);
>> +static void memcg_kmem_move(struct mem_cgroup *memcg);
>>
>> static inline bool
>> mem_cgroup_test_flag(const struct mem_cgroup *memcg, enum memcg_flags
>> flag)
>> @@ -3700,6 +3701,7 @@ static int mem_cgroup_force_empty(struct mem_cgroup
>> *memcg, bool free_all)
>>     int ret;
>>     int node, zid, shrink;
>>     int nr_retries = MEM_CGROUP_RECLAIM_RETRIES;
>> +     unsigned long usage;
>>     struct cgroup *cgrp = memcg->css.cgroup;
>>
>>     css_get(&memcg->css);
>> @@ -3719,6 +3721,8 @@ move_account:
>>     /* This is for making all *used* pages to be on LRU. */
>>     lru_add_drain_all();
>>     drain_all_stock_sync(memcg);
>> +     if (!free_all)
>> +         memcg_kmem_move(memcg);
>>
>
> Any reason we're not moving kmem charges when free_all is set as well?

```

Because the slab moving code expects to be synchronized with allocations (and itself). We can't call it when there are still tasks in the cgroup.

-- Suleiman