

---

Subject: Re: [PATCH v4 10/14] memcg: use static branches when code not in use  
Posted by [KAMEZAWA Hiroyuki](#) on Tue, 16 Oct 2012 08:48:29 GMT

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

---

(2012/10/12 16:47), Glauber Costa wrote:

> On 10/11/2012 05:40 PM, Michal Hocko wrote:

>> On Mon 08-10-12 14:06:16, Glauber Costa wrote:

>>> We can use static branches to patch the code in or out when not used.

>>>

>>> Because the `_ACTIVE` bit on `kmem_accounted` is only set after the  
>>> increment is done, we guarantee that the root memcg will always be  
>>> selected for `kmem` charges until all call sites are patched (see  
>>> `memcg_kmem_enabled`). This guarantees that no mischarges are applied.

>>>

>>> static branch decrement happens when the last reference count from the  
>>> `kmem` accounting in memcg dies. This will only happen when the charges  
>>> drop down to 0.

>>>

>>> When that happen, we need to disable the static branch only on those  
>>> memcgs that enabled it. To achieve this, we would be forced to  
>>> complicate the code by keeping track of which memcgs were the ones  
>>> that actually enabled limits, and which ones got it from its parents.

>>>

>>> It is a lot simpler just to do `static_key_slow_inc()` on every child  
>>> that is accounted.

>>>

>>> [ v4: adapted this patch to the changes in `kmem_accounted` ]

>>>

>>> Signed-off-by: Glauber Costa <[glommer@parallels.com](mailto:glommer@parallels.com)>

>>> CC: Kamezawa Hiroyuki <[kamezawa.hiroyu@jp.fujitsu.com](mailto:kamezawa.hiroyu@jp.fujitsu.com)>

>>> CC: Christoph Lameter <[cl@linux.com](mailto:cl@linux.com)>

>>> CC: Pekka Enberg <[penberg@cs.helsinki.fi](mailto:penberg@cs.helsinki.fi)>

>>> CC: Michal Hocko <[mhocko@suse.cz](mailto:mhocko@suse.cz)>

>>> CC: Johannes Weiner <[hannes@cmpxchg.org](mailto:hannes@cmpxchg.org)>

>>> CC: Suleiman Souhlal <[suleiman@google.com](mailto:suleiman@google.com)>

>>

>> Looks reasonable to me

>> Acked-by: Michal Hocko <[mhocko@suse.cz](mailto:mhocko@suse.cz)>

>>

>> Just a little nit.

>>

>> [...]

>>

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

>>> index 634c7b5..724a08b 100644

>>> --- a/mm/memcontrol.c

>>> +++ b/mm/memcontrol.c

>>> @@ -344,11 +344,15 @@ struct mem\_cgroup {

```
>>> /* internal only representation about the status of kmem accounting. */
>>> enum {
>>>  KMEM_ACCOUNTED_ACTIVE = 0, /* accounted by this cgroup itself */
>>> + KMEM_ACCOUNTED_ACTIVATED, /* static key enabled. */
>>>  KMEM_ACCOUNTED_DEAD, /* dead memcg, pending kmem charges */
>>> };
>>>
>>> -/* first bit */
>>> -#define KMEM_ACCOUNTED_MASK 0x1
>>> +/*
>>> + * first two bits. We account when limit is on, but only after
>>> + * call sites are patched
>>> + */
>>> +#define KMEM_ACCOUNTED_MASK 0x3
>>
>> The names are long but why not use KMEM_ACCOUNTED_ACTIVE*
>> #define KMEM_ACCOUNTED_MASK 1<<KMEM_ACCOUNTED_ACTIVE |
>> 1<<KMEM_ACCOUNTED_ACTIVATED
>>
> Because the names are long! =>
>
```

please use "long" macros ;) it's not bad.

Anyway,

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

---