
Subject: Re: [PATCH v5 09/14] memcg: kmem accounting lifecycle management
Posted by [Glauber Costa](#) on Thu, 18 Oct 2012 09:42:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 10/18/2012 03:28 AM, David Rientjes wrote:

> On Tue, 16 Oct 2012, Glauber Costa wrote:

>

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

>> index 1182188..e24b388 100644

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

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

>> @@ -344,6 +344,7 @@ struct mem_cgroup {

>> /* internal only representation about the status of kmem accounting. */

>> enum {

>> KMEM_ACCOUNTED_ACTIVE = 0, /* accounted by this cgroup itself */

>> + KMEM_ACCOUNTED_DEAD, /* dead memcg, pending kmem charges */

>

> "dead memcg with pending kmem charges" seems better.

>

ok.

>> };

>>

>> #define KMEM_ACCOUNTED_MASK (1 << KMEM_ACCOUNTED_ACTIVE)

>> @@ -353,6 +354,22 @@ static void memcg_kmem_set_active(struct mem_cgroup *memcg)

>> {

>> set_bit(KMEM_ACCOUNTED_ACTIVE, &memcg->kmem_accounted);

>> }

>> +

>> +static bool memcg_kmem_is_active(struct mem_cgroup *memcg)

>> +{

>> + return test_bit(KMEM_ACCOUNTED_ACTIVE, &memcg->kmem_accounted);

>> +}

>

> I think all of these should be inline.

>

They end up being, to be best of my knowledge the compiler can and will inline such simple functions regardless of their marking, unless you explicitly mark them noinline.

>> +

>> +static void memcg_kmem_mark_dead(struct mem_cgroup *memcg)

>> +{

>> + if (test_bit(KMEM_ACCOUNTED_ACTIVE, &memcg->kmem_accounted))

>> + set_bit(KMEM_ACCOUNTED_DEAD, &memcg->kmem_accounted);

>> +}

>

> The set_bit() doesn't happen atomically with the test_bit(), what
> synchronization is required for this?
>

I believe the explanation Michal gave in answer to this is comprehensive.
