
Subject: Re: [PATCH v3 16/28] memcg: kmem controller charge/uncharge infrastructure

Posted by [Frederic Weisbecker](#) on Wed, 30 May 2012 13:11:01 GMT

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

On Wed, May 30, 2012 at 04:38:39PM +0400, Glauber Costa wrote:

> On 05/30/2012 04:34 PM, Frederic Weisbecker wrote:

> > On Fri, May 25, 2012 at 05:03:36PM +0400, Glauber Costa wrote:

> > > +bool __mem_cgroup_new_kmem_page(struct page *page, gfp_t gfp)

> > > +{

> > > + struct mem_cgroup *memcg;

> > > + struct page_cgroup *pc;

> > > + bool ret = true;

> > > + size_t size;

> > > + struct task_struct *p;

> > > +

> > > + if (!current->mm || in_interrupt())

> > > + return true;

> > > +

> > > + rcu_read_lock();

> > > + p = rcu_dereference(current->mm->owner);

> > > + memcg = mem_cgroup_from_task(p);

> >

> > So this takes the memcg of the group owner rather than the

> > task? I understand why we want this for user memory, but for

> > kernel?

>

> That was already discussed when this first came up in my last submission

> If I recall correctly, Kame pointed out that this would be needed

> for proper OOM-scoring and killing.

Can we have at least a comment in the code that explain the reasons of taking the owner rather than the task? It's not going to be very obvious to future reviewers.

> Now of course we won't oom kernel threads or anything like that.

Seems we are not even accounting them anyway.

> But since this is also accounted towards memcg, it should at least be

> consistent with each memcg it accounts to.

>

> We can't account kmem for the thread's memcg, and mem to the process'.

Don't know. This goes a bit against cgroups semantics which group at the task level and not process. But I personally don't mind much, as long as it's documented.
