Subject: Re: [PATCH v3 04/13] kmem accounting basic infrastructure
Posted by Glauber Costa on Wed, 26 Sep 2012 22:54:11 GMT
View Forum Message <> Reply to Message

On 09/27/2012 02:42 AM, Tejun Heo wrote:
> Hello, Glauber.
>
> On Thu, Sep 27, 2012 at 02:29:06AM +0400, Glauber Costa wrote:
>> And then what? If you want a different behavior you need to go kill all
>> your services that are using memcg so you can get the behavior you want?
>> And if they happen to be making a specific flag choice by design, you
>> just say "you really can't run A + B together" ?
>>
>> I myself think global switches are an unnecessary complication. And let
>> us not talk about use_hierarchy, please. If it becomes global, it is
>> going to be as part of a phase out plan anyway. The problem with that is
>> not that it is global, is that it shouldn't even exist.
>
> I would consider it more of a compatibility thing which is set during
> boot and configurable by sysadmin. Let the newer systems enable it by
> default on boot and old configs / special ones disable it as
> necessary.
>

I don't. Much has been said in the past about the problem of sharing. A
lot of the kernel objects are shared by nature, this is pretty much
unavoidable. The answer we have been giving to this inquiry, is that the
workloads (us) interested in kmem accounted tend to be quite local in
their file accesses (and other kernel objects as well).

It should be obvious that not all workloads are like this, and some of
them would actually prefer to have their umem limited only.

There is nothing unreasonable in tracking user memory only.

If we have a global switch for "tracking all kernel memory", who would
you account the objects that are heavily shared to? I solve this by not
tracking kernel memory for cgroups in such workloads. What do you propose?

>>> Backward compatibility is covered with single switch and I really
>>> don't think "you can enable limits for kernel memory anytime but we
>>> don't keep track of whatever happened before it was flipped the first
>>> time because the first time is always special" is a sane thing to
>>> expose to userland. Or am I misunderstanding the proposed behavior
>>> again?
>>
>> You do keep track. Before you switch it for the first time, it all
>> belongs to the root memcg.

>
> Well, that's really playing with words. Limit is per cgroup and
> before the limit is set for the first time, everything is accounted to
> something else. How is that keeping track?
>

Even after the limit is set, it is set only by workloads that want kmem
to be tracked. If you want to track it during the whole lifetime of the
cgroup, you switch it before you put tasks to it. What is so crazy about it?

> The proposed behavior seems really crazy to me. Do people really
> think this is a good idea?
>

It is really sad that you lost the opportunity to say that in a room
full of mm developers that could add to this discussion in real time,
when after an explanation about this was given, Mel asked if anyone
would have any objections to this.