Subject: Re: [PATCH v3 06/16] memcg: infrastructure to match an allocation to the right cache

Posted by Tejun Heo on Mon, 24 Sep 2012 17:56:19 GMT

View Forum Message <> Reply to Message

Hello, Glauber.

```
On Mon, Sep 24, 2012 at 12:46:35PM +0400, Glauber Costa wrote:

> >> +#ifdef CONFIG_MEMCG_KMEM

> >> + /* Slab accounting */
> >> + struct kmem_cache *slabs[MAX_KMEM_CACHE_TYPES];

> >> +#endif

> >

> > Bah, 400 entry array in struct mem_cgroup. Can't we do something a

> > bit more flexible?

> >

> I guess. I still would like it to be an array, so we can easily access

> its fields. There are two ways around this:

> 1) Do like the events mechanism and allocate this in a separate

> structure. Add a pointer chase in the access, and I don't think it helps

> much because it gets allocated anyway. But we could at least

> defer it to the time when we limit the cache.
```

Start at some reasonable size and then double it as usage grows? How many kmem_caches do we typically end up using?

```
>>> + if (memcg->slabs[idx] == NULL) {
>>> + memcg_create_cache_enqueue(memcg, cachep);
>>
> > Do we want to wait for the work item if @gfp allows?
>>
> I tried this once, and it got complicated enough that I deemed as "not > worth it". I honestly don't remember much of the details now, it was one > of the first things I tried, and a bunch of time has passed. If you > think it is absolutely worth it, I can try it again. But at the very > best, I view this as an optimization.
```

I don't know. It seems like a logical thing to try and depends on how complex it gets. I don't think it's a must. The whole thing is somewhat opportunistic after all.

Thanks.

-tejun