Subject: Re: [PATCH 0/5] Kernel memory accounting container (v5)
Posted by Pavel Emelianov on Tue, 02 Oct 2007 12:51:10 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> Hi Pavel,
>
> One question about the general design of this - have you tested an
> approach where rather than tagging each object within the cache with
> the cgroup that allocated it, you instead have (inside the cache code)
> a separate cache structure for each cgroup? So the space overheads
> would go from having a per-object overhead (one pointer per object?)
> to having a "wastage" overhead (on average half a slab per cgroup).
> And the time overhead would be the time required to lookup the
> relevant cache for a cgroup at the start of the allocation operation,
> and the relevant cache for an object (from its struct page) at
> deallocation, rather then the time required to update the per-object
> housekeeping pointer.

Such a lookup would require a hastable or something similar. We already
have such a bad experience (with OpenVZ RSS fractions accounting for
example). Hash lookups imply the CPU caches screwup and hurt the performance.
See also the comment below.

> Each cache would need to be assigned a unique ID, used as an index
> into a per-cgroup lookup table of localized caches. (This could almost
> be regarded as a form of kmem_cache namespace).
>
> It seems to me that this alternative approach would be a lower memory
> overhead for people who have the kernel memory controller compiled in
> but aren't using it, or are only using a few groups.

I thought the same some time ago and tried to make a per-beancounter kmem
caches. The result was awful - the memory waste was much larger than in the
case of pointer-per-object approach. Let alone the performance questions -
each kmalloc required a synchronized hash table lookup that was too bad.

If you insist I can try to repeat the experiment, but I'm afraid the result
would be the same.

> Paul
>