
Subject: Re: [PATCH v3 04/13] kmem accounting basic infrastructure
Posted by [Michal Hocko](#) on Thu, 27 Sep 2012 12:08:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu 27-09-12 01:24:40, Glauber Costa wrote:

[...]

> About use cases, I've already responded: my containers use case is kmem
> limited. There are people like Michal that specifically asked for
> user-only semantics to be preserved.

Yes, because we have many users (basically almost all) who care only about the user memory because that's what occupies the vast majority of the memory. They usually want to isolate workload which would disrupt the global memory otherwise (e.g. backup process vs. database). You really do not want to pay an additional overhead for kmem accounting here.

> So your question for global vs local switch (that again, doesn't
> exist; only a local *limit* exists) should really be posed in the
> following way: "Can two different use cases with different needs be
> hosted in the same box?"

I think this is a good and a relevant question. I think this boils down to whether you want to have trusted and untrusted workloads at the same machine.

Trusted loads usually only need user memory accounting because kmem consumption should be really negligible (unless kernel is doing something really stupid and no kmem limit will help here).

On the other hand, untrusted workloads can do nasty things that administrator has hard time to mitigate and setting a kmem limit can help significantly.

IMHO such a different loads exist on a single machine quite often (Web server and a back up process as the most simplistic one). The per hierarchy accounting, therefore, sounds like a good idea without too much added complexity (actually the only added complexity is in the proper `kmem.limit_in_bytes` handling which is a single place).

So I would rather go with per-hierarchy thing.

> > Michal, Johannes, Kamezawa, what are your thoughts?
> >
> waiting! =)

Well, you guys generated a lot of discussion that one has to read through, didn't you :P

--

