Subject: Re: [PATCH v3 13/28] slub: create duplicate cache
Posted by Glauber Costa on Tue, 29 May 2012 20:08:38 GMT
View Forum Message <> Reply to Message

On 05/29/2012 11:55 PM, Christoph Lameter wrote:
>> NUMA just means what is the*best* node to put my memory.
>> > Now, if you actually say, through you syscalls "this is the node it should
>> > live in", then you have a constraint, that to the best of my knowledge is
>> > respected.
> Eith cpusets it means that memory needs to come from an assigned set of
> nodes.
>
>> > Now isolation here, is done in the container boundary. (cgroups, to be
>> > generic).
> Yes and with cpusets it is done at the cpuset boundary. Very much the
> same.

Well, I'd have to dive in the code a bit more, but that the impression
that the documentation gives me, by saying:

"Cpusets constrain the CPU and Memory placement of tasks to only
the resources within a task's current cpuset."

is that you can't allocate from a node outside that set. Is this correct?

So extrapolating this to memcg, the situation is as follows:

* You can't use more memory than what you are assigned to.
* In order to do that, you need to account the memory you are using
* and to account the memory you are using, all objects in the page
  must belong to you.

Please note the following:

Having two cgroups touching the same object is something. It tells
something about the relationship between them. This is shared memory.

Now having two cgroups putting objects in the same page, *does not mean
_anything_*. It just mean that one had the luck to allocate just after
the other.

With a predictable enough workload, this is a recipe for working around
the very protection we need to establish: one can DoS a physical box
full of containers, by always allocating in someone else's pages, and
pinning kernel memory down. Never releasing it, so the shrinkers are
useless.

So I still believe that if a page is allocated to a cgroup, all the

objects in there belong to it - unless of course the sharing actually means something - and identifying this is just too complicated.