Subject: Re: [RFC] [-mm PATCH] Memory controller fix swap charging context in unuse_pte()
Posted by Balbir Singh on Fri, 26 Oct 2007 06:14:44 GMT
View Forum Message <> Reply to Message

Hugh Dickins wrote:
> Gosh, it's nothing special. Appended below, but please don't shame
> me by taking it too seriously. Defaults to working on a 600M mmap
> because I'm in the habit of booting mem=512M. You probably have
> something better yourself that you'd rather use.
>

Thanks for sending it. I do have something more generic that I got
from my colleague.

>> In the use case you've mentioned/tested, having these mods to
>> control swapcache is actually useful, right?
>
> No idea what you mean by "these mods to control swapcache"?
>

Yes

> With your mem_cgroup mods in mm/swap_state.c, swapoff assigns
> the pages read in from swap to whoever's running swapoff and your
> unuse_pte mem_cgroup_charge never does anything useful: swap pages
> should get assigned to the appropriate cgroups at that point.
>
> Without your mem_cgroup mods in mm/swap_state.c, unuse_pte makes
> the right assignments (I believe). But I find that swapout (using
> 600M in a 512M machine) from a 200M cgroup quickly OOMs, whereas
> it behaves correctly with your mm/swap_state.c.
>

I'll try this test and play with your test

> Thought little yet about what happens to shmem swapped pages,
> and swap readahead pages; but still suspect that they and the
> above issue will need a "limbo" cgroup, for pages which are
> expected to belong to a not-yet-identified mem cgroup.
>

This is something I am yet to experiment with. I suspect this
should be easy to do if we decide to go this route.

>> Could you share your major objections at this point with the memory
>> controller at this point. I hope to be able to look into/resolve them

>> as my first priority in my list of items to work on.
>
> The things I've noticed so far, as mentioned before and above.
>
> But it does worry me that I only came here through finding swapoff
> broken by that unuse_mm return value, and then found one issue
> after another. It feels like the mem cgroup people haven't really
> thought through or tested swap at all, and that if I looked further
> I'd uncover more.
>

I thought so far that you've found a couple of bugs and one issue
with the way we account for swapcache. Other users, KAMEZAWA,
YAMAMOTO have been using and enhancing the memory controller.
I can point you to a set of links where I posted all the test
results. Swap was tested mostly through swapout/swapin when the
cgroup goes over limit. Please do help uncover as many bugs
as possible, please look more closely as you find more time.

> That's simply FUD, and I apologize if I'm being unfair: but that
> is how it feels, and I expect we all know that phase in a project
> when solving one problem uncovers three - suggests it's not ready.
>

I disagree, all projects/code do have bugs, which we are trying to
resolve, but I don't think there are any major design drawbacks
that *cannot* be fixed. We discussed the design at VM-Summit and
everyone agreed it was the way to go forward (even though Double
LRU has its complexity).

> Hugh

[snip]

Thanks for the review and your valuable feedback!

--
Warm Regards,
Balbir Singh
Linux Technology Center
IBM, ISTL

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers