Subject: Re: [RFC][PATCH 2/7] RSS controller core
Posted by mel on Wed, 14 Mar 2007 16:47:39 GMT
View Forum Message <> Reply to Message

On (13/03/07 10:26), Dave Hansen didst pronounce:
> On Mon, 2007-03-12 at 22:04 -0800, Andrew Morton wrote:
> > So these mmapped pages will contiue to be shared across all guests. The
> > problem boils down to "which guest(s) get charged for each shared page".
> >
> > A simple and obvious and easy-to-implement answer is "the guest which paged
> > it in". I think we should firstly explain why that is insufficient.
>
> My first worry was that this approach is unfair to the poor bastard that
> happened to get started up first. If we have a bunch of containerized
> web servers, the poor guy who starts Apache first will pay the price for
> keeping it in memory for everybody else.
>

I think it would be very difficult in practice to exploit a situation where
an evil guy forces another container to hold shared pages that the container
is not using themselves.

> That said, I think this is naturally worked around. The guy charged
> unfairly will get reclaim started on himself sooner. This will tend to
> page out those pages that he was being unfairly charged for.

Exactly. That said, the "poor bastard" will have to be pretty determined
to page out because the pages will appear active but it should happen
eventually especially if the container is under pressure.

> Hopefully,
> they will eventually get pretty randomly (eventually evenly) spread
> among all users. We just might want to make sure that we don't allow
> ptes (or other new references) to be re-established to pages like this
> when we're trying to reclaim them.

I don't think anything like that currently exists. It's almost the opposite
of what the current reclaim algorithm would be trying to do because it has no
notion of containers. Currently, the idea of paging out something in active
use is a mad plan.

Maybe what would be needed is something where the shared page is unmapped from
page tables and the next faulter must copy the page instead of reestablishing
the PTE. The data copy is less than ideal but it'd be cheaper than reclaim
and help the accounting. However, it would require a counter to track "how
many processes in this container have mapped the page".

> Either that, or force the next

> toucher to take ownership of the thing. But, that kind of arbitrary
> ownership transfer can't happen if we have rigidly defined boundaries
> for the containers.
>

Right, charging the next toucher would not work in the zones case. The next
toucher would establish a PTE to the page which is still in the zone of the
container being unfairly charged. It would need to be paged out or copied.

> The other concern is that the memory load on the system doesn't come
> from the first user ("the guy who paged it in"). The long-term load
> comes from "the guy who keeps using it." The best way to exemplify this
> is somebody who read()s a page in, followed by another guy mmap()ing the
> same page. The guy who did the read will get charged, and the mmap()er
> will get a free ride. We could probably get an idea when this kind of
> stuff is happening by comparing page->count and page->_mapcount, but it
> certainly wouldn't be conclusive. But, does this kind of nonsense even
> happen in practice?
>

I think this problem would happen with other accounting mechanisms as
well. However, it's more pronounced with zones because there are harder
limits on memory usage.

If the counter existed to track "how many processes in this container have
mapped the page", the problem of free-riders could be investigated by comparing
_mapcount to the container count. That would determine if additional steps
are required or not to force another container to assume the accounting cost.

--
Mel Gorman
Part-time Phd Student Linux Technology Center
University of Limerick IBM Dublin Software Lab

_____