Subject: Re: [RFC][PATCH 2/7] RSS controller core
Posted by ebiederm on Fri, 16 Mar 2007 18:54:09 GMT
View Forum Message <> Reply to Message

Dave Hansen <hansendc@us.ibm.com> writes:

> On Thu, 2007-03-15 at 18:55 -0600, Eric W. Biederman wrote:
>> To create a DOS attack.
>>
>> - Allocate some memory you know your victim will want in the future,
>>   (shared libraries and the like).
>> - Wait until your victim is using the memory you allocated.
>> - Terminate your memory resource group.
>> - Victim is pushed over memory limits by your exiting.
>> - Victim can no longer allocate memory
>> - Victim dies
>>
>> It's not quite that easy unless your victim calls mlockall(MCL_FUTURE),
>> but the potential is clearly there.
>>
>> Am I missing something? Or is this fundamental to any first touch scenario?
>>
>> I just know I have problems with first touch because it is darn hard to
>> reason about.
>
> I think it's fundamental to any case where two containers share the use
> of the page, but either one _can_ be charged but does not receive a
> _full_ charge for it.

Reasonable.

> I don't think it's uniquely associated with first-touch schemes.
>
> The software zones approach where there would be a set of "shared" zones
> would not have this problem, because any sharing would have to occur on
> data on which neither one was being charged.

True.   The "shared" zones approach would simply have the problem that it
would make sharing hard and thus reduce the effectiveness of the page cache.

The "shared" zone approach also would seem to interact in very weird ways
with real NUMA and memory hotplug or process migration. The fact that we
actually have to care about the real memory size on the machine makes me
look at it strange.

Zones should definitely be penalized in some category for the reduction
in efficiency of the page cache. It took us decades to learn that the
most efficient page cache was one that could resize and reallocate memory

on demand based on the current usage. Zones and possibly anything else with the concept of page ownership seems to be trying to be ignoring that wisdom.

> http://linux-mm.org/SoftwareZones

Looking at your page, and I'm too lazy to figure out how to update it I have a couple of comments.

- Why do limits have to apply to the unmapped page cache?

- Could you mention proper multi process RSS limits.
  (I.e. we count the number of pages each group of processes have mapped and limit that).
  It is the same basic idea as partial page ownership, but instead of page ownership you just count how many pages each group is using and strictly limit that. There is no page owner ship or partial charges.
  The overhead is just walking the rmap list at map and unmap time to see if this is the first users in the container. No additional kernel data structures are needed.

Eric

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