Subject: Re: Re: [RFC][PATCH 2/7] RSS controller core
Posted by dev on Tue, 13 Mar 2007 15:30:40 GMT
View Forum Message <> Reply to Message

Eric,

>>>And misses every resource sharing opportunity in sight.
>>
>>that was my point too.
>>
>>
>>>Except for
>>>filtering the which pages are eligible for reclaim an RSS limit should
>>>not need to change the existing reclaim logic, and with things like the
>>>memory zones we have had that kind of restriction in the reclaim logic
>>>for a long time. So filtering out ineligible pages isn't anything new.
>>
>>exactly this is implemented in the current patches from Pavel.
>>the only difference is that filtering is not done in general LRU list,
>>which is not effective, but via per-container LRU list.
>>So the pointer on the page structure does 2 things:
>>- fast reclamation
>
> Better than the rmap list?
>
>>- correct uncharging of page from where it was charged
>> (e.g. shared pages can be mapped first in one container, but the last unmap
>> done from another one).
>
> We should charge/uncharge all of them, not just one.
>
>
>>>>We need to work out what the requirements are before we can settle on an
>>>>implementation.
>>>
>>>
>>>If you are talking about RSS limits the term is well defined. The
>>>number of pages you can have mapped into your set of address space at
>>>any given time.
>>>
>>>Unless I'm totally blind that isn't what the patchset implements.
>>
>>Ouch, what makes you think so?
>>The fact that a page mapped into 2 different processes is charged only once?
>>Imho it is much more correct then sum of process' RSS within container, due to:
>>1. it is clear how much container uses physical pages, not abstract items
>>2. shared pages are charged only once, so the sum of containers RSS is still
>> about physical RAM.

>
>
> No the fact that a page mapped into 2 separate mm_structs in two
> separate accounting domains is counted only once. This is very likely
> to happen with things like glibc if you have a read-only shared copy
> of your distro. There appears to be no technical reason for such a
> restriction.
>
> A page should not be owned.

I would be happy to propose OVZ approach then, where a page is tracked
with page_beancounter data structure, which ties together
a page with beancounters which use it like this:

page -> page_beancounter -> list of beanocunters which has the page mapped

This gives a number of advantages:
- the page is accounted to all the VEs which actually use it.
- allows almost accurate tracking of page fractions used by VEs
  depending on how many VEs mapped the page.
- allows to track dirty pages, i.e. which VE dirtied the page
  and implement correct disk I/O accounting and CFQ write scheduling
  based on VE priorities.

> Going further unless the limits are draconian I don't expect users to
> hit the rss limits often or frequently. So in 99% of all cases page
> reclaim should continue to be global. Which makes me question messing
> with the general page reclaim lists.

It is not that rare when containers hit their limits, believe me :/
In trusted environments - probably you are right, in hosting - no.

Thanks,
Kirill