

---

Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)  
Posted by [Rohit Seth](#) on Thu, 17 Aug 2006 17:36:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 2006-08-17 at 10:23 -0700, Dave Hansen wrote:

> On Thu, 2006-08-17 at 10:16 -0700, Rohit Seth wrote:

> > > That said, it sure is simpler to implement, so I'm all for it!

> >

> > hmm, not sure why it is simpler.

>

> When you ask the question, "which container owns this page?", you don't

> have to look far,

as in page->mapping->container for user land?

> nor is it ambiguous in any way. It is very strict,

> and very straightforward.

What additional ambiguity you have when inode or task structures have  
the required information.

-rohit

---

---

Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)  
Posted by [Dave Hansen](#) on Thu, 17 Aug 2006 17:53:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, 2006-08-17 at 10:36 -0700, Rohit Seth wrote:

> On Thu, 2006-08-17 at 10:23 -0700, Dave Hansen wrote:

> > nor is it ambiguous in any way. It is very strict,

> > and very straightforward.

>

> What additional ambiguity you have when inode or task structures have

> the required information.

I think `_l_` was being too ambiguous. ;)

When you uniquely assign a kernel object, say mapping->container, there  
is no ambiguity.

-- Dave

---

---

Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)  
Posted by [dev](#) on Fri, 18 Aug 2006 08:52:46 GMT

Rohit Seth wrote:

> On Thu, 2006-08-17 at 10:23 -0700, Dave Hansen wrote:

>

>>On Thu, 2006-08-17 at 10:16 -0700, Rohit Seth wrote:

>>

>>>>That said, it sure is simpler to implement, so I'm all for it!

>>>

>>>hmm, not sure why it is simpler.

>>

>>When you ask the question, "which container owns this page?", you don't

>>have to look far,

>

>

> as in page->mapping->container for user land?

in case of anon\_vma, page->mapping can be the same  
for 2 pages belonging to different containers.

>>nor is it ambiguous in any way. It is very strict,

>>and very straightforward.

>

> What additional ambiguity you have when inode or task structures have  
> the required information.

inodes can belong to multiple containers and so do the pages.

Kirill

---

---

Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)  
Posted by [Dave Hansen](#) on Fri, 18 Aug 2006 14:52:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 2006-08-18 at 12:54 +0400, Kirill Korotaev wrote:

> > as in page->mapping->container for user land?

> in case of anon\_vma, page->mapping can be the same

> for 2 pages belonging to different containers.

page->mapping->container is the logical way to think about it, but it is quite easy to get from a mapping, using the VMA list, to the mms mapping a page. It wouldn't be a horrible stretch to get back to the tasks (or directly to the container) from that mm.

Has anyone ever thought of keeping a list of tasks using an mm as a list hanging off an mm?

-- Dave

---

---

Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)  
Posted by [Rohit Seth](#) on Fri, 18 Aug 2006 17:38:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, 2006-08-18 at 12:54 +0400, Kirill Korotaev wrote:

> Rohit Seth wrote:

> > On Thu, 2006-08-17 at 10:23 -0700, Dave Hansen wrote:

> >

> >> On Thu, 2006-08-17 at 10:16 -0700, Rohit Seth wrote:

> >>

> >>> That said, it sure is simpler to implement, so I'm all for it!

> >>>

> >>> hmmm, not sure why it is simpler.

> >>

> >> When you ask the question, "which container owns this page?", you don't

> >> have to look far,

> >

> >

> > as in page->mapping->container for user land?

> in case of anon\_vma, page->mapping can be the same

> for 2 pages belonging to different containers.

>

In your experience, have you seen processes belonging to different containers sharing the same anon\_vma? On a more general note, could you please point me to a place that has the list of requirements for which we are designing this solution.

> >> nor is it ambiguous in any way. It is very strict,

> >> and very straightforward.

> >

> > What additional ambiguity you have when inode or task structures have

> > the required information.

> inodes can belong to multiple containers and so do the pages.

>

I'm still thinking that inodes should belong to one container (or may be have it configurable based on some flag).

-rohit

---

Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)  
Posted by [dev](#) on Mon, 21 Aug 2006 11:27:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

>> in case of anon\_vma, page->mapping can be the same

>>for 2 pages belonging to different containers.  
>>  
>  
>  
> In your experience, have you seen processes belonging to different  
> containers sharing the same anon\_vma? On a more general note, could you  
> please point me to a place that has the list of requirements for which  
> we are designing this solution.  
>  
>  
>>>>nor is it ambiguous in any way. It is very strict,  
>>>>and very straightforward.  
>>>  
>>>What additional ambiguity you have when inode or task structures have  
>>>the required information.  
>>  
>>inodes can belong to multiple containers and so do the pages.  
>>  
>  
>  
> I'm still thinking that inodes should belong to one container (or may be  
> have it configurable based on some flag).  
this is not true for OpenVZ nor Linux-VServer.

Thanks,  
Kirill

---

Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)  
Posted by [Rohit Seth](#) on Tue, 22 Aug 2006 01:48:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Mon, 2006-08-21 at 15:29 +0400, Kirill Korotaev wrote:  
> >>in case of anon\_vma, page->mapping can be the same  
> >>for 2 pages belonging to different containers.  
> >>  
> >  
> >  
> > In your experience, have you seen processes belonging to different  
> > containers sharing the same anon\_vma? On a more general note, could you  
> > please point me to a place that has the list of requirements for which  
> > we are designing this solution.  
> >  
> >  
> >>>>nor is it ambiguous in any way. It is very strict,  
> >>>>and very straightforward.  
> >>>  
> >>>What additional ambiguity you have when inode or task structures have

> >>>the required information.  
> >>  
> >>>inodes can belong to multiple containers and so do the pages.  
> >>  
> >  
> >  
> > I'm still thinking that inodes should belong to one container (or may be  
> > have it configurable based on some flag).  
> this is not true for OpenVZ nor Linux-VServer.

Well, it is still useful. Just like an anonymous page get charged to container where the object (task) belong to, file page seems appropriate to belong to container where the object (inode) belongs to.

-rohit

---

Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)  
Posted by [xemul](#) on Tue, 22 Aug 2006 07:43:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Rohit Seth wrote:

[snip]

>>>> inodes can belong to multiple containers and so do the pages.  
>>>>  
>>>  
>>> I'm still thinking that inodes should belong to one container (or may be  
>>> have it configurable based on some flag).  
>> this is not true for OpenVZ nor Linux-VServer.  
>  
>  
> Well, it is still useful. Just like an anonymous page get charged to  
> container where the object (task) belong to, file page seems appropriate  
> to belong to container where the object (inode) belongs to.  
>  
> -rohit

Making inodes belong to one container only significantly shrinks container density. Sharing of glibc only saves up to 4Mb per container. For the case of 100 containers it is 400Mb of booth - the disk space and (what is more important) - RAM.

Pavel.