
Subject: Re: [RFC][PATCH 5/7] UBC: kernel memory accounting (core)
Posted by [Rohit Seth](#) on Fri, 18 Aug 2006 16:55:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 2006-08-18 at 13:38 +0400, Kirill Korotaev wrote:

> Rohit Seth wrote:

> > On Thu, 2006-08-17 at 17:27 +0400, Kirill Korotaev wrote:

> >

> >>> If I'm reading this patch right then seems like you are making page
> >>> allocations to fail w/o (for example) trying to purge some pages from
> >>> the page cache belonging to this container. Or is that reclaim going to
> >>> come later?

> >>

> >> charged kernel objects can't be `_reclaimed_`. how do you propose
> >> to reclaim tasks page tables or files or task struct or vma or etc.?

> >

> >

> >

> > I agree that kernel objects can't be reclaimed easily. But what you
> > are proposing is also not right. Returning failure w/o doing any
> > reclaim on pages (that are reclaimable) is not useful. And this is why
> > I asked, is this change going to be part of next set of patches (as
> > current set of patches are only tracking kernel usage).

> 1. reclaiming user resources is not that good idea as it looks to you.

> such solutions end up with lots of resources spent on reclaim.

> for user memory reclaims mean consumption of expensive disk I/O bandwidth

> which reduces overall system throughput and influences other users.

>

May be I'm overlooking something very obvious. Please tell me, what happens when a user hits a page fault and the page allocator is easily able to give a page from its pcp list. But container is over its limit of physical memory. In your patch there is no attempt by container support to see if some of the user pages are easily reclaimable. What options a user will have to make sure some room is created.

> 2. kernel memory is mostly not reclaimable. can you reclaim vma structs or ipc ids?

I'm not arguing about that at all. If people want to talk about reclaiming kernel pages then that should be done independent of this subject.

-rohit
