
Subject: Re: [RFC][PATCH 5/7] UBC: kernel memory accounting (core)

Posted by [Rohit Seth](#) on Wed, 16 Aug 2006 18:24:53 GMT

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

On Wed, 2006-08-16 at 19:40 +0400, Kirill Korotaev wrote:

- > Introduce UB_KMEMSIZE resource which accounts kernel
- > objects allocated by task's request.
- >
- > Reference to UB is kept on struct page or slab object.
- > For slabs each struct slab contains a set of pointers
- > corresponding objects are charged to.
- >
- > Allocation charge rules:
- > 1. Pages - if allocation is performed with __GFP_UBC flag - page
- > is charged to current's exec_ub.
- > 2. Slabs - kmem_cache may be created with SLAB_UBC flag - in this
- > case each allocation is charged. Caches used by kmalloc are
- > created with SLAB_UBC | SLAB_UBC_NOCHARGE flags. In this case
- > only __GFP_UBC allocations are charged.

<snip>

```
> --- ./mm/page_alloc.c.kmemcore 2006-08-16 19:10:38.000000000 +0400
> +++ ./mm/page_alloc.c 2006-08-16 19:10:51.000000000 +0400
> @@ -38,6 +38,8 @@
> #include <linux/mempolicy.h>
> #include <linux/stop_machine.h>
>
> +#include <ub/kmem.h>
> +
> #include <asm/tlbflush.h>
> #include <asm/div64.h>
> #include "internal.h"
> @@ -484,6 +486,8 @@ static void __free_pages_ok(struct page
> if (reserved)
> return;
>
> + ub_page_uncharge(page, order);
> +
> kernel_map_pages(page, 1 << order, 0);
> local_irq_save(flags);
> __count_vm_events(PGFREE, 1 << order);
> @@ -764,6 +768,8 @@ static void fastcall free_hot_cold_page(
> if (free_pages_check(page))
> return;
>
> + ub_page_uncharge(page, 0);
> +
```

```
> kernel_map_pages(page, 1, 0);
>
> pcp = &zone_pcp(zone, get_cpu())->pcp[cold];
> @@ -1153,6 +1159,11 @@ nopage:
>   show_mem();
>   }
> got_pg:
> + if ((gfp_mask & __GFP_UBC) &&
> +   ub_page_charge(page, order, gfp_mask)) {
> +   __free_pages(page, order);
> +   page = NULL;
> + }
> #ifdef CONFIG_PAGE_OWNER
>   if (page)
>     set_page_owner(page, order, gfp_mask);
```

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?

-rohit
