
Subject: Re: [PATCH 5/6] BC: kernel memory accounting (core)
Posted by [Chandra Seetharaman](#) on Thu, 24 Aug 2006 00:36:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 2006-08-23 at 15:06 +0400, Kirill Korotaev wrote:

<snip>

```
> --- ./include/bc/beancounter.h.bckmem 2006-07-28 18:43:52.000000000 +0400
> +++ ./include/bc/beancounter.h 2006-08-03 16:03:01.000000000 +0400
> @@ -14,7 +14,9 @@
>  * Resource list.
>  */
>
> -#define BC_RESOURCES 0
> +#define BC_KMEMSIZE 0
> +
> +#define BC_RESOURCES 1
```

As suggested before, a clean interface to define these would be better

```
>
> struct resource_parm {
>  /*
> --- ./include/bc/kmem.h.bckmem 2006-07-28 18:43:52.000000000 +0400
> +++ ./include/bc/kmem.h 2006-07-31 17:37:05.000000000 +0400
> @@ -0,0 +1,33 @@
> +/*
> + * include/bc/kmem.h
> + *
> + * Copyright (C) 2006 OpenVZ. SWsoft Inc
> + *
> + */
> +
> +#ifndef __BC_KMEM_H_
> +#define __BC_KMEM_H_
> +
> +#include <linux/config.h>
> +
> +/*
> + * BC_KMEMSIZE accounting
> + */
> +
> +struct mm_struct;
> +struct page;
> +struct beancounter;
> +
> +#ifdef CONFIG_BEANCOUNTERS
> +int bc_page_charge(struct page *page, int order, gfp_t flags);
```

```

> +void bc_page_uncharge(struct page *page, int order);
> +
> +int bc_slab_charge(kmem_cache_t *cachep, void *obj, gfp_t flags);
> +void bc_slab_uncharge(kmem_cache_t *cachep, void *obj);
> +#else
> +#define bc_page_charge(pg, o, mask) (0)
> +#define bc_page_uncharge(pg, o) do { } while (0)
> +#define bc_slab_charge(cachep, o, f) (0)
> +#define bc_slab_uncharge(cachep, o) do { } while (0)
> +#endif
> +#endif /* __BC_SLAB_H_ */
> --- ./kernel/bc/Makefile.bcsys 2006-07-28 14:08:37.000000000 +0400
> +++ ./kernel/bc/Makefile 2006-08-01 11:08:39.000000000 +0400
> @@ -7,3 +7,4 @@
> obj-$(CONFIG_BEANCOUNTERS) += beancounter.o
> obj-$(CONFIG_BEANCOUNTERS) += misc.o
> obj-$(CONFIG_BEANCOUNTERS) += sys.o
> +obj-$(CONFIG_BEANCOUNTERS) += kmem.o
> --- ./kernel/bc/beancounter.c.bckmem 2006-07-28 18:43:52.000000000 +0400
> +++ ./kernel/bc/beancounter.c 2006-08-03 16:14:17.000000000 +0400
> @@ -19,6 +19,7 @@ static void init_beancounter_struct(stru
> struct beancounter init_bc;
>
> const char *bc_rnames[] = {
> + "kmemsize", /* 0 */
> };

```

As suggested before, a clean interface would be a better idea.

```

>
> #define bc_hash_fun(x) (((x) >> 8) ^ (x)) & (BC_HASH_SIZE - 1)
> @@ -348,6 +378,8 @@ static void init_beancounter_syslimits(s
> {
> int k;
>
> + bc->bc_parms[BC_KMEMSIZE].limit = 32 * 1024 * 1024;
> +

```

This could be either be a configurable value by/for the controller or can be set by the controller through a callout.

```

> for (k = 0; k < BC_RESOURCES; k++)
> bc->bc_parms[k].barrier = bc->bc_parms[k].limit;
> }

```

<snip>

--

Chandra Seetharaman | Be careful what you choose....
- sekharan@us.ibm.com |you may get it.
