
Subject: Re: [ckrm-tech] [RFC][PATCH 2/7] UBC: core (structures, API)

Posted by [dev](#) on Thu, 17 Aug 2006 14:00:36 GMT

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

> On Wed, Aug 16, 2006 at 07:37:26PM +0400, Kirill Korotaev wrote:

>

>>+struct user_beancounter

>>+{

>>+ atomic_t ub_refcount;

>>+ spinlock_t ub_lock;

>>+ uid_t ub_uid;

>>+ struct hlist_node hash;

>>+

>>+ struct user_beancounter *parent;

>

>

> This seems to hint at some heirarchy of ubc? How would that heirarchy be

> used? I cant find anything in the patch which forms this heirarchy

> (basically I dont see any place where beancounter_findcreate() is called

> with non-NULL 2nd arg).

yes, it is possible to use hierarchical beancounters.

kernel memory, user memory and TCP/IP buffers are accounted hierarchically.

user interface for this is not provided yet as it would complicate patchset

and increase number of topics for discussion :)

> [snip]

>

>

>>+static void init_beancounter_syslimits(struct user_beancounter *ub)

>>+{

>>+ int k;

>>+

>>+ for (k = 0; k < UB_RESOURCES; k++)

>>+ ub->ub_parms[k].barrier = ub->ub_parms[k].limit;

>

>

> This sets barrier to 0. Is this value of 0 interpreted differently by

> different controllers? One way to interpret it is "dont allocate any

> resource", other way to interpret it is "don't care - give me what you

> can" (which makes sense for stuff like CPU and network bandwidth).

every patch which adds a resource modifies this function and sets

some default limit. Check: [PATCH 5/7] UBC: kernel memory accounting (core)

Thanks,

Kirill
