
Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)

Posted by [Pavel Emelianov](#) on Thu, 07 Sep 2006 07:29:17 GMT

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

Chandra Seetharaman wrote:

[snip]

>>> Will we need new user/kernel interfaces for cpu, i/o bandwidth, etc...?

>>>

>> no. no new interfaces are required.

>>

>

> Good to know that.

>

> Your CPU controller supports guarantee ?

>

It does, but CPU controller is not so simple as memory one.

> Do you have a i/o controller ?

>

>

>> BUT: I remind you the talks at OKS/OLS and in previous UBC discussions.

>> It was noted that having a separate interfaces for CPU, I/O bandwidth

>>

>

> But, it will be lot simpler for the user to configure/use if they are

> together. We should discuss this also.

>

IMHO such unification may only imply that one syscall is used to pass configuration info into kernel.

Each controller has specific configuring parameters different from the other ones. E.g. CPU controller must assign a "weight" to each group to share CPU time accordingly, but what is a "weight" for memory controller? IO may operate on "bandwidth" and it's not clear what is a "bandwidth" in Kb/sec for CPU controller and so on.

[snip]

>> The question is - whether web server is multithreaded or not...

>> If it is not - then no problem here, you can change current

>> context and new resources will be charged accordingly.

>>

>> And current BC code is `_able_` to handle it with `_minor_` changes.

>> (One just need to save bc not on mm struct, but rather on vma struct

>> and change mm->bc on `set_bc_id()`).

>>

>> However, no one (can some one from CKRM team please?) explained so far

>> what to do with threads. Consider the following example.

>>

>> 1. Threaded web server spawns a child to serve a client.
>> 2. child thread touches some pages and they are charged to child BC
>> (which differs from parent's one)
>> 3. child exits, but since its mm is shared with parent, these pages
>> stay mapped and charged to child BC.
>>
>> So the question is: what to do with these pages?
>> - should we recharge them to another BC?
>> - leave them charged?
>>
>
> Leave them charged. It will be charged to the appropriate UBC when they
> touch it again.
>
Do you mean that page must be re-charged each time someone touches it?
