Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by dev on Wed, 06 Sep 2006 13:54:13 GMT
View Forum Message <> Reply to Message

> On Tue, 2006-09-05 at 19:02 +0400, Kirill Korotaev wrote:
>
>>Core Resource Beancounters (BC) + kernel/user memory control.
>>
>>BC allows to account and control consumption
>>of kernel resources used by group of processes.
>
>
> Hi Kirill,
>
> I've honestly lost track of these discussions along the way, so I hope
> you don't mind summarizing a bit.
I think we need to create wiki to summarize it once and forever.
http://wiki.openvz.org/UBC_discussion

> Do these patches help with accounting for anything other than memory?
this patch set - no, but the complete one - does:
* numfile
* numptys
* numsocks (TCP, other, etc.)
* numtasks
* numflocks
...
this list of resources was chosen to make sure that no DoS from the container
is possible.
This list is extensible easily and if resource is out of interest than
its limits can be set to unlimited.

> Will we need new user/kernel interfaces for cpu, i/o bandwidth, etc...?
no. no new interfaces are required.

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
and memory maybe worthwhile. BTW, I/O bandwidth already has a separate
interface :/

> Have you given any thought to the possibility that a task might need to
> move between accounting contexts? That has certainly been a
> "requirement" pushed on to CKRM for a long time, and the need goes
> something like this:
Yes we thought about this and this is no more problematic for BC
than for CKRM. See my explanation below.

> 1. A system runs a web server, which services several virtual domains
> 2. that web server receives a request for foo.com
> 3. the web server switches into foo.com's accounting context
> 4. the web server reads things from disk, allocates some memory, and
>   makes a database request.
> 5. the database receives the request, and switches into foo.com's
>   accounting context, and charges foo.com for its resource use
> etc...

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?

> So, the goal is to run _one_ copy of an application on a system, but
> account for its resources in a much more fine-grained way than at the
> application level.

Yes.

> I think we can probably use beancounters for this, if we do not worry
> about migrating _existing_ charges when we change accounting context.
> Does that make sense?

exactly. thats what I'm saying. we can use beancounters for this
if charges are kept for creator.

Thanks,
Kirill