
Subject: Re: [ckrm-tech] [RFC][PATCH] UBC: user resource beancounters
Posted by [dev](#) on Mon, 21 Aug 2006 10:53:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chandra Seetharaman wrote:

> On Fri, 2006-08-18 at 14:36 +0400, Kirill Korotaev wrote:

>

>>Chandra Seetharaman wrote:

>>

>>>On Thu, 2006-08-17 at 17:55 +0400, Kirill Korotaev wrote:

>>>

>>>

>>>>On Wed, Aug 16, 2006 at 07:24:03PM +0400, Kirill Korotaev wrote:

>>>>

>>>>

>>>>

>>>>>As the first step we want to propose for discussion

>>>>>the most complicated parts of resource management:

>>>>>kernel memory and virtual memory.

>>>>

>>>>>Do you have any plans to post a CPU controller? Is that tied to UBC

>>>>>interface as well?

>>>>

>>>>Not everything at once :) To tell the truth I think CPU controller

>>>>is even more complicated than user memory accounting/limiting.

>>>>

>>>>No, fair CPU scheduler is not tied to UBC in any regard.

>>>

>>>

>>>Not having the CPU controller on UBC doesn't sound good for the

>>>infrastructure. IMHO, the infrastructure (for resource management) we

>>>are going to have should be able to support different resource

>>>controllers, without each controllers needing to have their own

>>>infrastructure/interface etc.,

>>

>>1. nothing prevents fair cpu scheduler from using UBC infrastructure.

>

>

> ok.

>

>

>> but currently we didn't start discussing it.

>>

>>2. as was discussed with a number of people on summit we agreed that

>> it maybe more flexible to not merge all resource types into one set.

>> CPU scheduler is usefull by itself w/o memory management.

>> the same for disk I/O bandwidht which is controlled in CFQ by

>> a separate system call.

>>
>> it is also more logical to have them separate since they
>> operate in different terms. For example, for CPU it is
>> shares which are relative units, while for memory it is
>> absolute units in bytes.
>
>
> We don't have to tie the units with the number. We can leave it to be
> sorted out between the user and the controller writer.
>
> Current implementation of resource groups does that.
>
>
>>>>As we discussed before, it is valuable to have an ability to limit
>>>>different resources separately (CPU, disk I/O, memory, etc.).
>>>
>>>Having ability to limit/control different resources separately not
>>>necessarily mean we should have different infrastructure for each.
>>
>>I'm not advocating to have a different infrastructure.
>>It is not the topic I raise with this patch set.
>>
>>
>>>>For example, it can be possible to place some mission critical
>>>>kernel threads (like kjournald) in a separate container.
>>>
>>>I don't understand the comment above (in this context).
>>
>>If you have a single container controlling all the resources, then
>>placing kjournald into CPU container would require setting
>>it's memory limits etc. And kjournald will start to be accounted separately,
>
>
> Not necessarily. You could just set the CPU shares of the group and
> leave the other resources as don't care.
don't care IMHO doesn't mean "accounted and limited as container X".
it sounds like "no limits" for me.

>>while my intention is kjournald to be accounted as the host system.
>>I only want to _guarantee_ some CPU to it.
> I do not see any _guarantee_ support, only barrier(soft limit) and
> limit. May be I overlooked. Can you tell me how guarantee is achieved
> with UBC.
we just provide additional parameters like oomguarpages, where barrier
is a guarantee.

Kirill