Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by Pavel Emelianov on Thu, 14 Sep 2006 07:53:18 GMT
View Forum Message <> Reply to Message

Chandra Seetharaman wrote:
> On Wed, 2006-09-13 at 12:06 +0400, Pavel Emelianov wrote:
>
>> Chandra Seetharaman wrote:
>>
>>> On Tue, 2006-09-12 at 14:48 +0400, Pavel Emelianov wrote:
>>> <snip>
>>>
>>>
>>>>> I do not think it is that simple since
>>>>> - there is typically more than one class I want to set guarantee to
>>>>> - I will not able to use both limit and guarantee
>>>>> - Implementation will not be work-conserving.
>>>>>
>>>>> Also, How would you configure the following in your model ?
>>>>>
>>>>> 5 classes: Class A(10, 40), Class B(20, 100), Class C (30, 100), Class D
>>>>> (5, 100), Class E(15, 50); (class_name(guarantee, limit))
>>>>>
>>>>>
>>>>>
>>>> What's the total memory amount on the node? Without it it's hard to make
>>>> any
>>>> guarantee.
>>>>
>>>>
>>> I wrote the example treating them as %, so 100 would be the total amount
>>> of memory.
>>>
>>>
>> OK. Then limiting must be done this way (unreclaimable limit/total limit)
>> A (15/40)
>> B (25/100)
>> C (35/100)
>> D (10/100)
>> E (20/50)
>> In this case each group will receive it's guarantee for sure.
>>
>> E.g. even if A, B, E and D will eat all it's unreclaimable memory then
>> we'll have
>> 100 - 15 - 25 - 20 - 10 = 30% of memory left (maybe after reclaiming) which
>> is perfectly enough for C's guarantee.
>>

>
> How did you arrive at the +5 number ?
>
I've solved a linear equations set :)
> What if I have 40 containers each with 2% guarantee ? what do we do
> then ? and many other different combinations (what I gave was not the
> _only_ scenario).
>
Then you need to solve a set of 40 equations. This sounds weird, but
don't afraid - sets like these are solved lightly.
>
>>>
>>>
>>>>> "Limit only" approach works for DoS prevention. But for providing QoS
>>>>> you would need guarantee.
>>>>>
>>>>>
>>>>>
>>>> You may not provide guarantee on physycal resource for a particular group
>>>> without limiting its usage by other groups. That's my major idea.
>>>>
>>>>
>>> I agree with that, but the other way around (i.e provide guarantee for
>>> everyone by imposing limits on everyone) is what I am saying is not
>>> possible.
>>>
>> Then how do you make sure that memory WILL be available when the group needs
>> it without limiting the others in a proper way?
>>
>
> You could limit others only if you _know_ somebody is not getting what
> they are supposed to get (based on guarantee).
>
I don't understand your idea. Limit does _not_ imply anything - it's
just a limit.
You may limit anything to anyone w/o bothering the consequences.
Guarantee implies that the resource you guarantee will be available and
this "will be" is something not that easy.

So I repeat my question - how can you be sure that these X megabytes you
guarantee to some group won't be used by others so that you won't be able
to reclaim them?