Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by dev on Fri, 15 Sep 2006 08:51:24 GMT
View Forum Message <> Reply to Message

Chandra,

>>>>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.
>>>
>>
>>extrapolate that to a varying # of permutations and real time changes in
>>the system workload. Won't it be complex ?
>>
>
> I have a C program that computes limits to obtain desired guarantees
> in a single 'for (i = 0; i < n; n++)' loop for any given set of guarantees.
> With all error handling, beautifull output, nice formatting etc it weights
> only 60 lines.
>
>>Wouldn't it be a lot simpler if we have the guarantee support instead ?
the calculation above doesn't seem hard :)

>>Why you do not like guarantee ? :)

> I do not 'do not like guarantee'. I'm just sure that there are two ways
> for providing guarantee (for unreclaimable resorces):
> 1. reserving resource for group in advance
> 2. limit resource for others
> Reserving is worse as it is essentially limiting (you cut off 100Mb from
> 1Gb RAM thus limiting the other groups by 900Mb RAM), but this limiting
> is too strict - you _have_ to reserve less than RAM size. Limiting in
> run-time is more flexible (you may create an overcommited BC if you
> want to) and leads to the same result - guarantee.
I think this deserves putting on Wiki.
It is very good clear point.

Chanrda, do you propose some 3rd way (we are unaware of) of implementing guarantees?

Thanks,
Kirill