Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by Balbir Singh on Mon, 18 Sep 2006 08:25:55 GMT
View Forum Message <> Reply to Message

Pavel Emelianov wrote:
> Kirill Korotaev wrote:
>
> [snip]
>>> 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.
>
> Look at http://wiki.openvz.org/Containers/Guarantees_for_resources
> I've described there how a guarantee can be get with limiting in details.
>
> [snip]
>
>>> 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.
>
> This is also on the page I gave link at.
>

This approach has the following disadvantages

1. Lets consider initialization - When we create 'n' groups initially, we need to spend O(n^2) time to assign guarantees.
2. Every time a limit or a guarantee changes, we need to recalculate guarantees and ensure that the change will not break any guarantees
3. The same thing as stated above, when a resource group is created or deleted

This can lead to some instability; a change in one group propagates to all other groups.

--

Balbir Singh,
Linux Technology Center,