Subject: Re: [ckrm-tech] [PATCH] BC: resource beancounters (v4) (added user memory)
Posted by Pavel Emelianov on Thu, 14 Sep 2006 13:02:48 GMT
View Forum Message <> Reply to Message

Balbir Singh wrote:
> Pavel Emelianov wrote:
>
>> 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?
>>
>>
>
> May be we can treat a guarantee as a soft guarantee. A soft
> guarantee would imply that when a group needs its guaranteed
> resources, the
> system makes its best effort to make it available.
>
> In soft guarantees, resources not actively used by a group can be
> shared with
> other groups.
>
> Hard guarantees would probably require reserving the resource in
> advance and
> sharing of the resources not used, with other groups, might not be
> possible.
>
> Comments?
>
Reserving in advance means that sometimes you won't be able to start a
new group without taking back some of reserved pages. This is ... strange.

I think that a satisfactory solution now would be:
 - limit unreclaimable memory during mmap() against soft limit to prevent
   potential rejects during page faults;
 - reclaim memory in case of hitting hard limit;
 - guarantees are done via setting soft and hard limits as I've shown
before.

The question still open is wether or not to account fractions.

I propose to skip fractions for a while and try to charge the page to
it's first user.

So final BC design is:
1. three resources:
    - kernel memory
    - user unreclaimable memory
    - user reclaimable memory
2. unreclaimable memory is charged "in advance", reclaimable
   is charged "on demand" with reclamation if needed
3. each object (kernel one or user page) is charged to the
   first user
4. each resource controller declares it's own
    - meaning of "limit" parameter (percent/size/bandwidth/etc)
    - behaviour on changing limit (e.g. reclamation)
    - behaviour on hitting the limit (e.g. reclamation)
5. BC can be assigned to any task by pid (not just current)
   without recharging currently charged resources.

---