Subject: Re: [ckrm-tech] [RFC][PATCH 5/7] UBC: kernel memory accounting (core)
Posted by dev on Mon, 21 Aug 2006 10:48:55 GMT
View Forum Message <> Reply to Message

Chandra Seetharaman wrote:
> Kirill,
>
> IMO, a UBC with resource constraint(limit in this case) should behave no
> different than a kernel with limited memory. i.e it should do
> reclamation before it starts failing allocation requests. It could even
> do it preemptively.
first, please notice, that this thread is not about user memory.
we can discuss it later when about to control user memory. And
I still need to notice, that different models of user memory control
can exist. With and without reclamation.

> There is no guarantee support which is required for providing QoS.
where? in UBC? in UBC _there_ are guarentees, even in regard to OOM killer.

> Each controller modifying the infrastructure code doesn't look good. We
> can have proper interfaces to add a new resource controller.
controllers do not modify interfaces nor core. They just add
themself to the list of resources and setup default limits.
do you think it is worth creating infrastructure for these
2 one-line-changes?

> chandra
> On Wed, 2006-08-16 at 19:40 +0400, Kirill Korotaev wrote:
>
>>Introduce UB_KMEMSIZE resource which accounts kernel
>>objects allocated by task's request.
>>
>>Reference to UB is kept on struct page or slab object.
>>For slabs each struct slab contains a set of pointers
>>corresponding objects are charged to.
>>
>>Allocation charge rules:
>> define1. Pages - if allocation is performed with __GFP_UBC flag - page
>>   is charged to current's exec_ub.
>> 2. Slabs - kmem_cache may be created with SLAB_UBC flag - in this
>>   case each allocation is charged. Caches used by kmalloc are
>>   created with SLAB_UBC | SLAB_UBC_NOCHARGE flags. In this case
>>   only __GFP_UBC allocations are charged.
>>
>>Signed-Off-By: Pavel Emelianov <xemul@sw.ru>
>>Signed-Off-By: Kirill Korotaev <dev@sw.ru>
>>
> <snip>