Subject: Re: [PATCH 6/7] BC: kernel memory (core)
Posted by Balbir Singh on Mon, 04 Sep 2006 15:45:47 GMT
View Forum Message <> Reply to Message

Kirill Korotaev wrote:
> Balbir Singh wrote:
>> Kirill Korotaev wrote:

>>> +#ifdef CONFIG_BEANCOUNTERS
>>> +   union {
>>> +   struct beancounter   *page_bc;
>>> +   } bc;
>>> +#endif
>>> };
>>>
>>> +#define page_bc(page)   ((page)->bc.page_bc)
>>
>>
>> Minor comment - page->(bc).page_bc has too many repititions of page
>> and bc - see
>> the Practice of Programming by Kernighan and Pike
>>
>> I missed the part of why you wanted to have a union (in struct page
>> for bc)?
> because this union is used both for kernel memory accounting and user
> memeory tracking.

Ok.. that's good. I remember seeing a user_bc sometime back in the code.
I had some idea about allowing tasks to migrate across resources (bean
counters), which I think can be easily done for user space pages, if the
user limits are tracked separately.

>
>>> const char *bc_rnames[] = {
>>> +   "kmemsize",   /* 0 */
>>> };
>>>
>>> static struct hlist_head bc_hash[BC_HASH_SIZE];
>>> @@ -221,6 +222,8 @@ static void init_beancounter_syslimits(s
>>> {   int k;
>>>
>>> +   bc->bc_parms[BC_KMEMSIZE].limit = 32 * 1024 * 1024;
>>> +
>>
>>
>> Can't this be configurable CONFIG_XXX or a #defined constant?
> This is some arbitraty limited container, just to make sure it is not
> created unlimited. User space should initialize limits properly after

> creation
> anyway. So I don't see reasons to make it configurable, do you?

May be its not very important now but configurable limits will help a confused user. Even if we decide to use this number for now, a constant like BC_DEFAULT_MEM_LIMIT is easier to read.

>> I wonder if bc_page_charge() should be called bc_page_charge_failed()?
>> Does it make sense to atleast partially start reclamation here? I know
>> with
>> bean counters we cannot reclaim from a particular container, but for now
>> we could kick off kswapd() or call shrink_all_memory() inline (Dave's
>> patches do this to shrink memory from the particular cpuset). Or do
>> you want to leave this
>> slot open for later?
> yes. my intention is to account correctly all needed information first.
> After we agree on accounting, we can agree on how to do reclamaition.
>

That sounds like a good plan.

--

Balbir Singh,
Linux Technology Center,
IBM Software Labs