Subject: Re: [ckrm-tech] [PATCH 4/13] BC: context handling
Posted by Pavel Emelianov on Thu, 23 Nov 2006 09:56:00 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 11/23/06, Pavel Emelianov <xemul@openvz.org> wrote:
>> You mean moving is like this:
>>
>> old_bc = task->real_bc;
>> task->real_bc = new_bc;
>> cmpxchg(&tsk->exec_bc, old_bc, new_bc);
>>
>> ? Then this won't work:
>>
>> Initialisation:
>> current->exec_bc = init_bc;
>> current->real_bc = init_bc;
>> ...
>> IRQ:
>> current->exec_bc = init_bc;
>> ...
>> old_bc = tsk->real_bc; /* init_bc */
>> tsk->real_bc = bc1;
>> cx(tsk->exec_bc, init_bc, bc1); /* ok */
>> ...
>> Here at the middle of an interrupt
>> we have bc1 set as exec_bc on task
>> which IS wrong!
>
> You could get round that by having a separate "irq_bc" that's never
> valid for a task not in an interrupt.

No no no. This is not what is needed. You see, we do have to
set exec_bc as temporary (and atomic) context. Having temporary
context is 1. flexible 2. needed by beancounters' network accountig.
We have to track this particular scenario.

Moreover making get_exec_bc() as
if (in_interrupt())
 return &irq_bc;
else
 return current->exec_bc;
is awful. It must me simple and stupid to allow us making temporary
contexts in any place of code.

Maybe we can make smth similar to wait_task_inactive and change
it's beancounter before unlocking the runqueue?

> Paul
>