Subject: Re: [ckrm-tech] [PATCH 4/13] BC: context handling
Posted by Pavel Emelianov on Thu, 23 Nov 2006 09:20:29 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
```
> On 11/23/06, Pavel Emelianov <xemul@openvz.org> wrote:
>>
>> We can do the following:
>>
>>   if (tsk == current)
>>       /* fast way */
>>       tsk->exec_bc = bc;
>>   else
>>       /* slow way */
>>       stop_machine_run(...);
>>
>> What do you think?
>
> How about having two pointers per task:
>
> - exec_bc, which is the one used for charging
> - real_bc, which is the task's actual beancounter
>
> at the start of irq, do
>
> current->exec_bc = &init_bc;
>
> at the end of irq, do
>
> current->exec_bc = current->real_bc;
>
> When moving a task to a different bc do:
>
> task->real_bc = new_bc;
> atomic_cmpxchg(&task->exec_bc, old_bc, new_bc);
```

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!
...
current->exec_bc =
     current->real_bc;
```

We need some way to be sure that task isn't running at
the moment we change it's beancounter. Otherwise we're
risking that we'll spoil some temporary context.

> (with appropriate memory barriers). So if the task is in an irq with a
> modified exec_bc pointer, we do nothing, otherwise we update exec_bc
> to point to the new real_bc.
>
> Paul