Subject: Re: [ckrm-tech] [PATCH 4/13] BC: context handling
Posted by Pavel Emelianov on Thu, 23 Nov 2006 08:35:10 GMT
View Forum Message <> Reply to Message

Paul Menage wrote:
> On 11/9/06, Kirill Korotaev <dev@sw.ru> wrote:
>> +
>> +int bc_task_move(int pid, struct beancounter *bc, int whole)
>> +{
>
> ...
>
>> +
>> +	down_write(&mm->mmap_sem);
>> +	err = stop_machine_run(do_set_bcid, &data, NR_CPUS);
>> +	up_write(&mm->mmap_sem);
>
> Isn't this a little heavyweight for moving a task into/between
> beancounters?

It's a main reason we were against moving arbitrary task.

We need to track the situation when we change beancounter on
task that is currently handles an interrupt and thus set a
temporary BC as exec one. I see no other way that keeps pair
set_exec_bc()/get_exec_bc() lock-less.

The problem is even larger than I've described. set_exec_bc()
is used widely in OpenVZ beancounters to set temporary context
e.g. for skb handling. Thus we need some safe way to "catch"
the task in a "safe" place. In OpenVZ we solve this by moving
only current into beancounter. In this patch set we have to
move arbitrary task and thus - such complication.

I repeat - we can do this w/o stop_machine, but this would
require locking in set_exec_bc()/get_exec_bc() but it's too
bad. Moving tasks happens rarely but setting context is a
very common operation (e.g. in each interrupt).

We can do the following:

```
 if (tsk == current)
    /* fast way */
    tsk->exec_bc = bc;
 else
    /* slow way */
    stop_machine_run(...);
```

What do you think?

> Paul
> -
> To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
> the body of a message to majordomo@vger.kernel.org
> More majordomo info at http://vger.kernel.org/majordomo-info.html
> Please read the FAQ at http://www.tux.org/lkml/
>