
Subject: Re: [RFC PATCH 5/5] refresh VM committed space after a task migration
Posted by [Andrea Righi](#) on Wed, 11 Jun 2008 10:37:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dave Hansen wrote:

> On Tue, 2008-06-10 at 01:33 +0200, Andrea Righi wrote:

```
>> + preempt_disable();
>> + committed = atomic_long_read(&p->vm_committed_space);
>> + atomic_long_sub(committed, &old_mem->vmacct.vm_committed_space);
>> + atomic_long_add(committed, &mem->vmacct.vm_committed_space);
>> + preempt_enable();
```

>> out:

```
>> mmpout(mm);
```

```
>> }
```

>

> Why bother with the preempt stuff here? What does the actually protect
> against? I assume that you're trying to keep other tasks that might run
> on this CPU from seeing weird, inconsistent numbers in here. Is there
> some other looks that keeps *other* cpus from seeing this?

>

> In any case, I think it needs a big, fat comment.

Yes, true, `mem_cgroup_move_task()` is called after the `task->cgroups` pointer has been changed. So, even if task changes its committed space between the `atomic_long_sub()` and `atomic_long_add()` it will be correctly accounted in the new cgroup.

-Andrea

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
