Subject: Re: [PATCH 09/11] memcg: propagate kmem limiting information to children
Posted by akpm on Mon, 25 Jun 2012 23:21:58 GMT
View Forum Message <> Reply to Message

On Tue, 26 Jun 2012 02:36:27 +0400
Glauber Costa <glommer@parallels.com> wrote:

```
> On 06/25/2012 10:29 PM, Tejun Heo wrote:
> > Feeling like a nit pervert but..
> >
> > On Mon, Jun 25, 2012 at 06:15:26PM +0400, Glauber Costa wrote:
> >> @@ -287,7 +287,11 @@ struct mem_cgroup {
> >>     * Should the accounting and control be hierarchical, per subtree?
> >>     */
> >>    bool use_hierarchy;
> >> - bool kmem_accounted;
> >> + /*
> >> +  * bit0: accounted by this cgroup
> >> +  * bit1: accounted by a parent.
> >> +  */
> >> + volatile unsigned long kmem_accounted;
> >
> > Is the volatile declaration really necessary?  Why is it necessary?
> > Why no comment explaining it?
>
> Seems to be required by set_bit and friends. gcc will complain if it is
> not volatile (take a look at the bit function headers)
```

That would be a broken gcc. We run test_bit()/set_bit() and friends
against plain old `unsigned long' in thousands of places. There's
nothing special about this one!