Subject: Re: [PATCH 09/11] memcg: propagate kmem limiting information to children
Posted by David Rientjes on Tue, 26 Jun 2012 05:23:59 GMT
View Forum Message <> Reply to Message

On Mon, 25 Jun 2012, Andrew Morton wrote:

> > >> */
> > >> 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!
>

No version of gcc would complain about this, even with 4.6 and later with -fstrict-volatile-bitfields, it's a qualifier that determines whether or not the access to memory is the exact size of the bitfield and aligned to its natural boundary. If the type isn't qualified as such then it's simply going to compile to access the native word size of the architecture. No special consideration is needed for a member of struct mem_cgroup.