Subject: Re: [PATCH 09/11] memcg: propagate kmem limiting information to children
Posted by Tejun Heo on Mon, 25 Jun 2012 22:49:42 GMT
View Forum Message <> Reply to Message

Hello, Glauber.

On Tue, Jun 26, 2012 at 02:36:27AM +0400, Glauber Costa wrote:
> >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)

Hmmm?  Are you sure gcc includes volatile in type check?  There are a
lot of bitops users in the kernel but most of them don't use volatile
decl on the variable.

```
> >>+	 */
> >>+	parent = parent_mem_cgroup(iter);
> >>+	while (parent && (parent != memcg)) {
> >>+		if (test_bit(KMEM_ACCOUNTED_THIS, &parent->kmem_accounted))
> >>+			goto noclear;
> >>+
> >>+		parent = parent_mem_cgroup(parent);
> >>+	}
> >
```
> >Better written in for (;;)?  Also, if we're breaking on parent ==
> >memcg, can we ever hit NULL parent in the above loop?
>
> I can simplify to test parent != memcg only, indeed it is not
> expected to be NULL (but if it happens to be due to any kind of bug,
> we protect against NULL-dereference, that is why I like to write
> this way)

I personally don't really like that.  It doesn't really add meaningful
protection (if that happens the tree walking is already severely
broken) while causes confusion to future readers of the code (when can
parent be NULL?).

Thanks.

--
tejun