
Subject: Re: [PATCH v5 2/2] decrement static keys on real destroy time
Posted by [akpm](#) on Wed, 16 May 2012 20:57:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 16 May 2012 11:03:47 +0400
Glauber Costa <glommer@parallels.com> wrote:

```
> On 05/14/2012 05:38 AM, Li Zefan wrote:
> > +static void disarm_static_keys(struct mem_cgroup *memcg)
> >
> > +{
> > +#ifdef CONFIG_INET
> > + if (memcg->tcp_mem.cg_proto.activated)
> > +  static_key_slow_dec(&memcg_socket_limit_enabled);
> > +#endif
> > +}
> >
> >
> > Move this inside the ifdef/endif below ?
> >
> > Otherwise I think you'll get compile error if !CONFIG_INET...
>
> I don't fully get it.
>
> We are supposed to provide a version of it for
> CONFIG_CGROUP_MEM_RES_CTLR_KMEM and an empty version for
> !CONFIG_CGROUP_MEM_RES_CTLR_KMEM
>
> Inside the first, we take an action for CONFIG_INET, and no action for
> !CONFIG_INET.
>
> Bear in mind that the slab patches will add another test to that place,
> and that's why I am doing it this way from the beginning.
>
> Well, that said, I not only can be wrong, I very frequently am.
>
> But I just compiled this one with and without CONFIG_INET, and it seems
> to be going alright.
>
```

Yes, the ifdeffings in that area are rather nasty.

I wonder if it would be simpler to do away with the ifdef nesting.
At the top-level, just do

```
#if defined(CONFIG_CGROUP_MEM_RES_CTLR_KMEM) && defined(CONFIG_INET)
static void disarm_static_keys(struct mem_cgroup *memcg)
{
```

```
if (memcg->tcp_mem.cg_proto.activated)
    static_key_slow_dec(&memcg_socket_limit_enabled);
}
#else
static inline void disarm_static_keys(struct mem_cgroup *memcg)
{
}
#endif
```

The tcp_proto_cgroup() definition could go inside that ifdef as well.
