

---

Subject: Re: [PATCH v5 2/2] decrement static keys on real destroy time  
Posted by [Glauber Costa](#) on Thu, 17 May 2012 09:52:13 GMT

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

---

On 05/17/2012 09:37 AM, Andrew Morton wrote:

>> > If that happens, locking in static\_key\_slow\_inc will prevent any damage.  
>> > My previous version had explicit code to prevent that, but we were  
>> > pointed out that this is already part of the static\_key expectations, so  
>> > that was dropped.  
> This makes no sense. If two threads run that code concurrently,  
> key->enabled gets incremented twice. Nobody anywhere has a record that  
> this happened so it cannot be undone. key->enabled is now in an  
> unknown state.

Kame, Tejun,

Andrew is right. It seems we will need that mutex after all. Just this is not a race, and neither something that should belong in the static\_branch interface.

We want to make sure that enabled is not updated before the jump label update, because we need a specific ordering guarantee at the patched sites. And *that*, the interface guarantees, and we were wrong to believe it did not. That is a correction issue for the accounting, and that part is right.

But when we disarm it, we'll need to make sure that happened only once, otherwise we may never unpatch it. That, or we'd need that to be a counter. The jump label interface does not - and should not - keep track of how many updates happened to a key. That's the role of whoever is using it.

If you agree with the above, I'll send this patch again with the correction.

Andrew, thank you very much. Do you spot anything else here?

---