Subject: [PATCH v5 0/2] fix static_key disabling problem in memcg
Posted by Glauber Costa on Fri, 11 May 2012 20:11:15 GMT
View Forum Message <> Reply to Message

Hi, Tejun, Kame,

This series is composed of the two patches of the last fix, with no changes (only exception is the removal of x = false assignments that Tejun requested, that is done now). Note also that patch 1 of this series was reused by me in the slab accounting patches for memcg.

The first patch, that adds a mutex to memcg is dropped. I didn't posted it before so I could wait for Kame to get back from his vacations and properly review it.

Kame: Steven Rostedt pointed out that our analysis of the static branch updates were wrong, so the mutex is really not needed.

The key to understand that, is that atomic_inc_not_zero will only return right away if the value is not yet zero - as the name implies - but the update in the atomic variable only happens after the code is patched.

Therefore, if two callers enters with a key value of zero, both will be held at the jump_label_lock() call, effectively guaranteeing the behavior we need.

Glauber Costa (2):
  Always free struct memcg through schedule_work()
  decrement static keys on real destroy time

```
 include/net/sock.h        |    9 ++++++++
 mm/memcontrol.c           |   50 +++++++++++++++++++++++++++++++++-----------
 net/ipv4/tcp_memcontrol.c |   32 +++++++++++++++++++++------
 3 files changed, 71 insertions(+), 20 deletions(-)
```

--
1.7.7.6