
Subject: [PATCH 2/3] slab: Fix imbalanced rcu locking
Posted by [Anton Vorontsov](#) on Mon, 30 Apr 2012 10:01:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Not sure why the code tries to unlock the rcu. The only case where slab grabs the lock is around mem_cgroup_get_kmem_cache() call, which won't result into calling cache_grow() (that tries to unlock the rcu).

```
=====
[ BUG: bad unlock balance detected! ]
3.4.0-rc4+ #33 Not tainted
-----

swapper/0/0 is trying to release lock (rcu_read_lock) at:
[<ffffff8134f0b4>] cache_grow.constprop.63+0xe8/0x371
but there are no more locks to release!
```

other info that might help us debug this:
no locks held by swapper/0/0.

stack backtrace:

Pid: 0, comm: swapper/0 Not tainted 3.4.0-rc4+ #33

Call Trace:

```
[<ffffff8134f0b4>] ? cache_grow.constprop.63+0xe8/0x371
[<ffffff8134cf09>] print_unlock_inbalance_bug.part.26+0xd1/0xd9
[<ffffff8134f0b4>] ? cache_grow.constprop.63+0xe8/0x371
[<ffffff8106865e>] print_unlock_inbalance_bug+0x4e/0x50
[<ffffff8134f0b4>] ? cache_grow.constprop.63+0xe8/0x371
[<ffffff8106cb26>] __lock_release+0xd6/0xe0
[<ffffff8106cb8c>] lock_release+0x5c/0x80
[<ffffff8134f0cc>] cache_grow.constprop.63+0x100/0x371
[<ffffff8134f5c6>] cache_alloc_refill+0x289/0x2dc
[<ffffff810bf682>] ? kmem_cache_alloc+0x92/0x260
[<ffffff81676a0f>] ? pidmap_init+0x79/0xb2
[<ffffff810bf842>] kmem_cache_alloc+0x252/0x260
[<ffffff810bf5f0>] ? kmem_freepages+0x180/0x180
[<ffffff81676a0f>] pidmap_init+0x79/0xb2
[<ffffff81667aa3>] start_kernel+0x297/0x2f8
[<ffffff8166769e>] ? repair_env_string+0x5a/0x5a
[<ffffff816672fd>] x86_64_start_reservations+0x101/0x105
[<ffffff816673f1>] x86_64_start_kernel+0xf0/0xf7
```

Signed-off-by: Anton Vorontsov <anton.vorontsov@linaro.org>

include/linux/slab_def.h | 2 --
1 file changed, 2 deletions(-)

diff --git a/include/linux/slab_def.h b/include/linux/slab_def.h

index c4f7e45..2d371ae 100644

--- a/include/linux/slab_def.h

+++ b/include/linux/slab_def.h

@@ -245,13 +245,11 @@ mem_cgroup_kmem_cache_prepare_sleep(struct kmem_cache
*cachep)

 * enabled.

 */

 kmem_cache_get_ref(cachep);

- rcu_read_unlock();

}

static inline void

mem_cgroup_kmem_cache_finish_sleep(struct kmem_cache *cachep)

{

- rcu_read_lock();

 kmem_cache_drop_ref(cachep);

}

--

1.7.9.2
