

---

Subject: Re: linux-next: lockdep whinge in cgroup\_rmdir  
Posted by [Nick Piggin](#) on Fri, 14 Jan 2011 03:35:17 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Fri, Jan 14, 2011 at 2:34 AM, <Valdis.Kletnieks@vt.edu> wrote:  
> Seen booting yesterday's linux-next, was not present in 2.6.37-rc7-mmotm1202.  
>  
> Not sure if it's an selinux or cgroup issue, so I'm throwing it at every  
> address I can find for either. This is easily replicatable and happens at  
> every boot, so I can test patches if needed. Am willing to bisect it down if  
> nobody knows right off the bat what the problem is.  
>  
> The 'W' taint is from the already-reported kernel/workqueue.c worker\_enter\_idle issue.  
>  
> [ 85.100795] systemd[1]: readahead-replay.service: main process exited, code=exited,  
status=1  
> [ 85.101530]  
> [ 85.101531] =====  
> [ 85.101796] [ INFO: possible recursive locking detected ]  
> [ 85.102002] 2.6.37-next-20110111 #1  
> [ 85.102009] -----  
> [ 85.102009] systemd/1 is trying to acquire lock:  
> [ 85.102009] (&(&dentry->d\_lock)->rlock){+...}, at: [<ffffffff8107ca5c>]  
cgroup\_rmdir+0x339/0x479  
> [ 85.102009]  
> [ 85.102009] but task is already holding lock:  
> [ 85.102009] (&(&dentry->d\_lock)->rlock){+...}, at: [<ffffffff8107ca54>]  
cgroup\_rmdir+0x331/0x479  
> [ 85.102009]  
> [ 85.102009] other info that might help us debug this:  
> [ 85.102009] 4 locks held by systemd/1:  
> [ 85.102009] #0: (&sb->s\_type->i\_mutex\_key#14/1){+...}, at: [<ffffffff810fea4d>]  
do\_rmdir+0x7d/0x121  
> [ 85.102009] #1: (&sb->s\_type->i\_mutex\_key#14){+...}, at: [<ffffffff810fd4bc>]  
vfs\_rmdir+0x4a/0xbe  
> [ 85.102009] #2: (cgroup\_mutex){+...}, at: [<ffffffff8107cb84>]  
cgroup\_rmdir+0x461/0x479  
> [ 85.102009] #3: (&(&dentry->d\_lock)->rlock){+...}, at: [<ffffffff8107ca54>]  
cgroup\_rmdir+0x331/0x479  
> [ 85.102009]  
> [ 85.102009] stack backtrace:  
> [ 85.102009] Pid: 1, comm: systemd Tainted: G W 2.6.37-next-20110111 #1  
> [ 85.102009] Call Trace:  
> [ 85.102009] [<ffffffff81069f22>] ? \_\_lock\_acquire+0x929/0xd4e  
> [ 85.102009] [<ffffffff8107c6f1>] ? cgroup\_clear\_directory+0xff/0x131  
> [ 85.102009] [<ffffffff8107c6f1>] ? cgroup\_clear\_directory+0xff/0x131  
> [ 85.102009] [<ffffffff8107ca5c>] ? cgroup\_rmdir+0x339/0x479  
> [ 85.102009] [<ffffffff8106a859>] ? lock\_acquire+0x100/0x126

```
> [ 85.102009] [<ffffff8107ca5c>] ? cgroup_rmdir+0x339/0x479
> [ 85.102009] [<ffffff815521ef>] ? sub_preempt_count+0x35/0x48
> [ 85.102009] [<ffffff8154e401>] ? _raw_spin_lock+0x36/0x45
> [ 85.102009] [<ffffff8107ca5c>] ? cgroup_rmdir+0x339/0x479
> [ 85.102009] [<ffffff8107ca5c>] ? cgroup_rmdir+0x339/0x479
> [ 85.102009] [<ffffff810579cd>] ? autoremove_wake_function+0x0/0x34
> [ 85.102009] [<ffffff811e1839>] ? selinux_inode_rmdir+0x15/0x17
> [ 85.102009] [<ffffff810fd4eb>] ? vfs_rmdir+0x79/0xbe
> [ 85.102009] [<ffffff810feaa0>] ? do_rmdir+0xd0/0x121
> [ 85.102009] [<ffffff8100256c>] ? sysret_check+0x27/0x62
> [ 85.102009] [<ffffff8106ac79>] ? trace_hardirqs_on_caller+0x117/0x13b
> [ 85.102009] [<ffffff8154e201>] ? trace_hardirqs_on_thunk+0x3a/0x3f
> [ 85.102009] [<ffffff8110040b>] ? sys_rmdir+0x11/0x13
> [ 85.102009] [<ffffff8100253b>] ? system_call_fastpath+0x16/0x1b
> [ 85.268272] systemd[1]: readahead-collect.service: main process exited, code=exited,
status=1
>
> Any ideas?
```

It looks like it is just a missing parent->child lock order annotation, but mainline cgroupfs code looks to be OK there. What does cgroup\_clear\_directory() look like in mmotm?

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---