

Hi Paul,

In [patch 3/7] Containers (V8): Add generic multi-subsystem API to containers, you have forcefully enabled interrupt in container\_init\_subsys() with spin\_unlock\_irq() which breaks on PPC64.

```
> +static void container_init_subsys(struct container_subsys *ss) {
> + int retval;
> + struct list_head *l;
> + printk(KERN_ERR "Initializing container subsys %s\n",
> + ss->name);
> +
> + /* Create the top container state for this subsystem */
> + ss->root = &rootnode;
> + retval = ss->create(ss, dummytop);
> + BUG_ON(retval);
> + init_container_css(ss, dummytop);
> +
> + /* Update all container groups to contain a subsys
> + * pointer to this state - since the subsystem is
> + * newly registered, all tasks and hence all container
> + * groups are in the subsystem's top container. */
> + spin_lock_irq(&container_group_lock);
> + l = &init_container_group.list;
> + do {
> + struct container_group *cg =
> + list_entry(l, struct container_group, list);
> + cg->subsys[ss->subsys_id] =
> + dummytop->subsys[ss->subsys_id];
> + l = l->next;
> + } while (l != &init_container_group.list);
> + spin_unlock_irq(&container_group_lock);
```

Interrupt gets enabled here and on PPC64, the kernel takes a pending decremter and crashes because it is too early to handle them.

Use of irqsave and restore routines would fix the problem.  
I have included the fix along with minor Kconfig correction.

Also your 3/7 patch did not showup on LKML.

--Vaidy

```
> +  
> + need_forkexit_callback |= ss->fork || ss->exit;
```

---