

## [PATCH 1/2] lock\_cpu\_hotplug only if CONFIG\_CPU\_HOTPLUG

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-01/7611.html>

---

**From:** Rusty Russell ([rusty\\_at\\_rustcorp.com.au](mailto:rusty_at_rustcorp.com.au))

**Date:** 01/30/04

To: [akpm@osdl.org](mailto:akpm@osdl.org)

Date: Fri, 30 Jan 2004 16:33:34 +1100

The cpucontrol mutex is not required when no cpus can go up and down. Andrew wrote a wrapper for it to avoid #ifdefs, this expands that to only be defined for CONFIG\_HOTPLUG\_CPU, and uses it everywhere.

The only downside is that the cpucontrol lock was overloaded by my recent patch to net/core/flow.c to protect it from reentrance, so this reintroduces the local flow\_flush\_sem. This code isn't speed critical, so taking two locks when CONFIG\_HOTPLUG\_CPU=y is not really an issue.

Name: lock\_cpu\_hotplug/unlock\_cpu\_hotplug Macros only if CONFIG\_HOTPLUG\_CPU

Author: Andrew Morton, Rusty Russell

Status: Booted on 2.6.2-rc2-bk2

Depends:

D: The cpucontrol mutex is not required when no cpus can go up and down.

D: Andrew wrote a wrapper for it to avoid #ifdefs, this expands that to

D: only be defined for CONFIG\_HOTPLUG\_CPU, and uses it everywhere.

D:

D: The only downside is that the cpucontrol lock was overloaded by my

D: recent patch to net/core/flow.c to protect it from reentrance, so

D: this reintroduces the local flow\_flush\_sem. This code isn't speed

D: critical, so taking two locks when CONFIG\_HOTPLUG\_CPU=y is not really

D: an issue.

```
diff -urpN --exclude TAGS -X /home/rusty/devel/kernel/kernel-patches/current-dontdiff --minimal
.31749-linux-2.6.2-rc1-bk1/include/linux/cpu.h .31749-linux-2.6.2-rc1-bk1.updated/include/linux/cpu.h
--- .31749-linux-2.6.2-rc1-bk1/include/linux/cpu.h 2004-01-25 01:25:08.000000000 +1100
+++ .31749-linux-2.6.2-rc1-bk1.updated/include/linux/cpu.h 2004-01-25 01:25:46.000000000 +1100
@@ -38,9 +38,6 @@ extern void unregister_cpu_notifier(stru
```

```
int cpu_up(unsigned int cpu);
```

```
—#define lock_cpu_hotplug() down(&cpucontrol)
```

```
—#define unlock_cpu_hotplug() up(&cpucontrol)
```

Linux-Kernel: [PATCH 1/2] lock\_cpu\_hotplug only if CONFIG\_CPU\_HOTPLUG

```

-
#else

static inline int register_cpu_notifier(struct notifier_block *nb)
@@ -51,12 +48,17 @@ static inline void unregister_cpu_notifi
{
}

-#define lock_cpu_hotplug() do { } while (0)
-#define unlock_cpu_hotplug() do { } while (0)
-
#endif /* CONFIG_SMP */
extern struct sysdev_class cpu_sysdev_class;

+#ifdef CONFIG_HOTPLUG_CPU
/* Stop CPUs going up and down. */
extern struct semaphore cpucontrol;
+#define lock_cpu_hotplug() down(&cpucontrol)
+#define unlock_cpu_hotplug() up(&cpucontrol)
+#else
+#define lock_cpu_hotplug() do { } while (0)
+#define unlock_cpu_hotplug() do { } while (0)
+#endif
+
#endif /* _LINUX_CPU_H_ */
diff -urpN --exclude TAGS -X /home/rusty/devel/kernel/kernel-patches/current-dontdiff --minimal
.31749-linux-2.6.2-rc1-bk1/kernel/module.c .31749-linux-2.6.2-rc1-bk1.updated/kernel/module.c
--- .31749-linux-2.6.2-rc1-bk1/kernel/module.c 2004-01-21 16:19:08.000000000 +1100
+++ .31749-linux-2.6.2-rc1-bk1.updated/kernel/module.c 2004-01-25 01:25:14.000000000 +1100
@@ -554,7 +554,7 @@ static int stop_refcounts(void)
    stopref_state = STOPREF_WAIT;

    /* No CPUs can come up or down during this. */
- down(&cpucontrol);
+ lock_cpu_hotplug();

    for (i = 0; i < NR_CPUS; i++) {
        if (i == cpu || !cpu_online(i))
@@ -572,7 +572,7 @@ static int stop_refcounts(void)
    /* If some failed, kill them all. */
    if (ret < 0) {
        stopref_set_state(STOPREF_EXIT, 1);
- up(&cpucontrol);
+ unlock_cpu_hotplug();
        return ret;
    }

@@ -595,7 +595,7 @@ static void restart_refcounts(void)
    stopref_set_state(STOPREF_EXIT, 0);
    local_irq_enable();
    preempt_enable();

```

## Linux-Kernel: [PATCH 1/2] lock\_cpu\_hotplug only if CONFIG\_CPU\_HOTPLUG

```
- up(&cpucontrol);
+ unlock_cpu_hotplug();
}
#else /* ...!SMP */
static inline int stop_refcounts(void)
diff -urpN --exclude TAGS -X /home/rusty/devel/kernel/kernel-patches/current-dontdiff --minimal
.31749-linux-2.6.2-rc1-bk1/net/core/flow.c .31749-linux-2.6.2-rc1-bk1.updated/net/core/flow.c
--- .31749-linux-2.6.2-rc1-bk1/net/core/flow.c 2004-01-25 01:25:08.000000000 +1100
+++ .31749-linux-2.6.2-rc1-bk1.updated/net/core/flow.c 2004-01-25 01:25:14.000000000 +1100
@@ -283,10 +283,11 @@ static void flow_cache_flush_per_cpu(voi
void flow_cache_flush(void)
{
    struct flow_flush_info info;
+ static DECLARE_MUTEX(flow_flush_sem);

- /* Don't want cpus going down or up during this, also protects
- * against multiple callers. */
+ /* Don't want cpus going down or up during this. */
    lock_cpu_hotplug();
+ down(&flow_flush_sem);
    atomic_set(&info.cpuleft, num_online_cpus());
    init_completion(&info.completion);

@@ -296,6 +297,7 @@ void flow_cache_flush(void)
    local_bh_enable();

    wait_for_completion(&info.completion);
+ up(&flow_flush_sem);
    unlock_cpu_hotplug();
}

--
    Anyone who quotes me in their sig is an idiot. -- Rusty Russell.
-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at http://vger.kernel.org/majordomo-info.html
Please read the FAQ at http://www.tux.org/lkml/
```