Subject: [PATCH] move_task_off_dead_cpu() should be called with disabled ints
Posted by Kirill Korotaev on Tue, 14 Nov 2006 10:24:12 GMT
View Forum Message <> Reply to Message

move_task_off_dead_cpu() requires interrupts to be disabled,
while migrate_dead() calls it with enabled interrupts.
Added appropriate comments to functions and added
BUG_ON(!irqs_disabled()) into double_rq_lock() and
double_lock_balance() which are the origin sources of such bugs.

Signed-Off-By: Kirill Korotaev <dev@openvz.org>

```
--- ./kernel/sched.c.schedx 2006-11-08 17:44:15.000000000 +0300
+++ ./kernel/sched.c 2006-11-14 11:32:24.000000000 +0300
@@ -1942,6 +1942,7 @@ static void double_rq_lock(struct rq *rq
  __acquires(rq1->lock)
  __acquires(rq2->lock)
 {
+ BUG_ON(!irqs_disabled());
  if (rq1 == rq2) {
   spin_lock(&rq1->lock);
   __acquire(rq2->lock); /* Fake it out ;) */
@@ -1981,6 +1982,11 @@ static void double_lock_balance(struct r
  __acquires(busiest->lock)
  __acquires(this_rq->lock)
 {
+ if (unlikely(!irqs_disabled())) {
+  /* printk() doesn't work good under rq->lock */
+  spin_unlock(&this_rq->lock);
+  BUG_ON(1);
+ }
  if (unlikely(!spin_trylock(&busiest->lock))) {
   if (busiest < this_rq) {
    spin_unlock(&this_rq->lock);
@@ -5056,7 +5062,10 @@ wait_to_die:
 }

 #ifdef CONFIG_HOTPLUG_CPU
-/* Figure out where task on dead CPU should go, use force if neccessary. */
+/*
+ * Figure out where task on dead CPU should go, use force if neccessary.
+ * NOTE: interrupts should be disabled by the caller
+ */
 static void move_task_off_dead_cpu(int dead_cpu, struct task_struct *p)
 {
  unsigned long flags;
@@ -5176,6 +5185,7 @@ void idle_task_exit(void)
```

```
  mmdrop(mm);
 }

+/* called under rq->lock with disabled interrupts */
 static void migrate_dead(unsigned int dead_cpu, struct task_struct *p)
 {
  struct rq *rq = cpu_rq(dead_cpu);
@@ -5192,10 +5202,11 @@ static void migrate_dead(unsigned int de
   * Drop lock around migration; if someone else moves it,
   * that's OK.  No task can be added to this CPU, so iteration is
   * fine.
+  * NOTE: interrupts should be left disabled  --dev@
   */
- spin_unlock_irq(&rq->lock);
+ spin_unlock(&rq->lock);
  move_task_off_dead_cpu(dead_cpu, p);
- spin_lock_irq(&rq->lock);
+ spin_lock(&rq->lock);

  put_task_struct(p);
 }
```