
Subject: [PATCH] sched: CPU hotplug race vs. set_cpus_allowed()

Posted by [dev](#) on Mon, 26 Jun 2006 07:58:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

There is a race between set_cpus_allowed()
and move_task_off_dead_cpu().
__migrate_task() doesn't report any err code, so
task can be left on its runqueue if its cpus_allowed mask
changed so that dest_cpu is not longer a possible target.
Also, changing cpus_allowed mask requires rq->lock being held.

Changes from original post according to Con Colivas notes:

- added comment
- cleanup code a bit

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

Acked-By: Ingo Molnar <mingo@elte.hu>

Kirill

P.S. against 2.6.17-mm1

--- linux-2.6.17-mm1s.orig/kernel/sched.c 2006-06-21 18:53:17.000000000 +0400

+++ linux-2.6.17-mm1.dev/kernel/sched.c 2006-06-26 11:54:19.000000000 +0400

@@ -4857,13 +4857,16 @@ EXPORT_SYMBOL_GPL(set_cpus_allowed);

*

* So we race with normal scheduler movements, but that's OK, as long
* as the task is no longer on this CPU.

+ *

+ * Returns non-zero if task was successfully migrated.

*/

-static void __migrate_task(struct task_struct *p, int src_cpu, int dest_cpu)

+static int __migrate_task(struct task_struct *p, int src_cpu, int dest_cpu)

{
 runqueue_t *rq_dest, *rq_src;

+ int ret = 0;

 if (unlikely(cpu_is_offline(dest_cpu)))

- return;

+ return ret;

 rq_src = cpu_rq(src_cpu);

 rq_dest = cpu_rq(dest_cpu);

@@ -4891,9 +4894,10 @@ static void __migrate_task(struct task_s

 if (TASK_PREEMPTS_CURR(p, rq_dest))

 resched_task(rq_dest->curr);

 }

-

+ ret = 1;

```

out:
    double_rq_unlock(rq_src, rq_dest);
+ return ret;
}

/*
@@ -4964,10 +4968,13 @@ int sigstop_on_cpu_lost;
/* Figure out where task on dead CPU should go, use force if neccessary. */
static void move_task_off_dead_cpu(int dead_cpu, struct task_struct *tsk)
{
+ runqueue_t *rq;
+ unsigned long flags;
    int dest_cpu;
    cpumask_t mask;
    int force = 0;

+restart:
    /* On same node? */
    mask = node_to_cpumask(cpu_to_node(dead_cpu));
    cpus_and(mask, mask, tsk->cpus_allowed);
@@ -4979,8 +4986,10 @@ static void move_task_off_dead_cpu(int d

    /* No more Mr. Nice Guy. */
    if (dest_cpu == NR_CPUS) {
+ rq = task_rq_lock(tsk, &flags);
        cpus_setall(tsk->cpus_allowed);
        dest_cpu = any_online_cpu(tsk->cpus_allowed);
+ task_rq_unlock(rq, &flags);

    /*
     * Don't tell them about moving exiting tasks or
@@ -5000,7 +5009,8 @@ static void move_task_off_dead_cpu(int d
        if (tsk->mm && sigstop_on_cpu_lost)
            force = 1;
    }
- __migrate_task(tsk, dead_cpu, dest_cpu);
+ if (!__migrate_task(tsk, dead_cpu, dest_cpu))
+ goto restart;

    if (force)
        force_sig_specific(SIGSTOP, tsk);

```
