Subject: [RFC] [PATCH 1/3] task_cpu(p) needs to be correct always
Posted by Srivatsa Vaddagiri on Wed, 23 May 2007 16:51:26 GMT
View Forum Message <> Reply to Message

We rely very much on task_cpu(p) to be correct at all times, so that we can
correctly find the task_grp_rq from which the task has to be removed or added
to.

There is however one place in the scheduler where this assumption of
task_cpu(p) being correct is broken. This patch fixes that piece of code.

(Thanks to Balbir Singh for pointing this out to me)

Signed-off-by : Srivatsa Vaddagiri <vatsa@in.ibm.com>

---
 kernel/sched.c | 8 +++++---
 1 file changed, 5 insertions(+), 3 deletions(-)

Index: linux-2.6.21-rc7/kernel/sched.c
===
--- linux-2.6.21-rc7.orig/kernel/sched.c 2007-05-23 20:46:41.000000000 +0530
+++ linux-2.6.21-rc7/kernel/sched.c 2007-05-23 20:48:26.000000000 +0530
@@ -4571,7 +4571,7 @@
 static int __migrate_task(struct task_struct *p, int src_cpu, int dest_cpu)
 {
  struct rq *rq_dest, *rq_src;
- int ret = 0;
+ int ret = 0, on_rq;

  if (unlikely(cpu_is_offline(dest_cpu)))
   return ret;
@@ -4587,9 +4587,11 @@
  if (!cpu_isset(dest_cpu, p->cpus_allowed))
   goto out;

- set_task_cpu(p, dest_cpu);
- if (p->on_rq) {
+ on_rq = p->on_rq;
+ if (on_rq)
   deactivate_task(rq_src, p, 0);
+ set_task_cpu(p, dest_cpu);
+ if (on_rq) {
   activate_task(rq_dest, p, 0);
   check_preempt_curr(rq_dest, p);
  }

--

Regards,
vatsa

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers