Subject: [PATCH 8/9] task_cpu(p) needs to be correct always
Posted by Srivatsa Vaddagiri on Thu, 12 Apr 2007 17:59:37 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>

---

 1 file changed, 8 insertions(+), 2 deletions(-)

```
diff -puN kernel/sched.c~task_cpu kernel/sched.c
--- linux-2.6.20/kernel/sched.c~task_cpu 2007-04-12 09:10:39.000000000 +0530
+++ linux-2.6.20-vatsa/kernel/sched.c 2007-04-12 11:07:14.000000000 +0530
@@ -5400,6 +5400,7 @@ static int __migrate_task(struct task_st
 {
  struct rq *rq_dest, *rq_src;
  int ret = 0;
+ struct prio_array *array;

  if (unlikely(cpu_is_offline(dest_cpu)))
   return ret;
@@ -5415,8 +5416,8 @@ static int __migrate_task(struct task_st
  if (!cpu_isset(dest_cpu, p->cpus_allowed))
   goto out;

- set_task_cpu(p, dest_cpu);
- if (p->array) {
+ array = p->array;
+ if (array) {
   /*
    * Sync timestamp with rq_dest's before activating.
    * The same thing could be achieved by doing this step
@@ -5426,6 +5427,11 @@ static int __migrate_task(struct task_st
```

```
   p->timestamp = p->timestamp - rq_src->most_recent_timestamp
     + rq_dest->most_recent_timestamp;
   deactivate_task(p, rq_src);
+ }
+
+ set_task_cpu(p, dest_cpu);
+
+ if (array) {
   __activate_task(p, rq_dest);
   if (TASK_PREEMPTS_CURR(p, rq_dest))
    resched_task(rq_dest->curr);

_
--
Regards,
vatsa
```