
Subject: Re: [RFC] [PATCH 0/3] Add group fairness to CFS
Posted by [Srivatsa Vaddagiri](#) on Fri, 25 May 2007 07:45:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Thu, May 24, 2007 at 12:26:16AM +0200, Guillaume Chazarain wrote:
> As a sidenote, while in CFS-v13 a nice=0 tasks seems to get 10x more CPU
> than a nice=10 one, with the group fairness patch, the ratio drops to
> less than 2x (for tasks with the same UID).

gah ..silly me.

Can you repeat your tests with this patch pls? With the patch applied, I am now getting the same split between nice 0 and nice 10 task as CFS-v13 provides (90:10 as reported by top)

```
5418 guest  20  0 2464 304 236 R  90 0.0  5:41.40 3 hog
5419 guest  30 10 2460 304 236 R  10 0.0  0:43.62 3 nice10hog
```

Fix a stupid bug, where I was not calling `__check_preempt_curr_fair()` at task level during `task_tick` ..

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

```
diff -puN kernel/sched_fair.c~fix kernel/sched_fair.c
--- linux-2.6.22-rc1-cfs-group/kernel/sched_fair.c~fix 2007-05-25 12:28:52.000000000 +0530
+++ linux-2.6.22-rc1-cfs-group-vatsa/kernel/sched_fair.c 2007-05-25 12:30:06.000000000 +0530
@@ -577,11 +577,12 @@ static void entity_tick(struct lrq *lrq,
     *n = task_entity(next);

     if ((c == lrq->rq->idle) || (rt_prio(n->prio) &&
-    (n->prio < c->prio)))
+    (n->prio < c->prio))) {
         resched_task(c);
-    } else
-    __check_preempt_curr_fair(lrq, next, curr,
-    *(lrq->sched_granularity));
+    return;
+    }
+    }
+    __check_preempt_curr_fair(lrq, next, curr, *(lrq->sched_granularity));
 }
```

```
static void _update_load(struct Irq *this_rq)
```

—

--

Regards,
vatsa

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