

Re: [PATCH] sched: smpnice work around for active\_load\_balance()

## Re: [PATCH] sched: smpnice work around for active\_load\_balance()

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-03/msg09750.html>

---

- *From:* Peter Williams <[pwil3058@xxxxxxxxxxxxxxxxx](mailto:pwil3058@xxxxxxxxxxxxxxxxx)>
  - *Date:* Wed, 29 Mar 2006 09:44:49 +1100
- 

Siddha, Suresh B wrote:

On Tue, Mar 28, 2006 at 05:00:50PM +1100, Peter Williams wrote:

Problem:

It is undesirable for HT/MC packages to have more than one of their CPUs busy if there are other packages that have all of their CPUs idle. This

We need to balance even if the other packages are not idle.. For example, consider a 4-core DP system, if we have 6 runnable (assume same priority) processes, we want to schedule 3 of them in each package..

Well I hope that when you do a proper implementation for this issue that it takes this into account. The current implementation doesn't.

Today's active load balance implementation is very simple and generic. And hence it works smoothly with dual and multi-core..

The application of active balancing to address your problem in the current implementation is essentially random.

Please read my OLS 2005 paper which talks about different scheduling scenarios and also how

A URL would be handy.

we were planning to implement Power savings policy in case of multi-core.. I had a prototype patch for doing this, which I held it up before going on vacation, as it needed some rework with your smpnice patch in place..

Re: [PATCH] sched: smpnice work around for active\_load\_balance()

Re: [PATCH] sched: smpnice work around for active\_load\_balance()

I will post a patch on top of current mainline for your reference.

```
+ } else if (!busiest_has_loaded_cpus && avg_load < max_load) {
```

I haven't fully digested the result of this patch but should this be  
avg\_load < max\_load or avg\_load > max\_load ?

Yes. Thanks for spotting that.

Either way, I can show scheduling scenarios which will fail...

I'd be interested to see the ones that would fail with the corrected code. I can show lots of examples where load balancing fails to do the right thing without the smpnice patches so it becomes a matter of which are more important.

```
- if (rqi->raw_weighted_load > max_load && rqi->nr_running > 1) {  
+ if (rqi->nr_running > 1) {  
+ if (rqi->raw_weighted_load > max_load || !busiest_is_loaded) {  
+ max_load = rqi->raw_weighted_load;  
+ busiest = rqi;  
+ busiest_is_loaded = 1;  
+ }  
+ } else if (!busiest_is_loaded && rqi->raw_weighted_load > max_load) {
```

Please note the point that same scheduling logic has to work for all the different levels of scheduler domains... I think these checks complicates the decisions as we go up in the scheduling hierarchy.. Please go through the HT/MC/MP/Numa combinations and with same/different priority processes for different scenarios..

Sometimes complexity is necessary. E.g. to handle the limitations of HT technology. In this case, the complexity is necessary to make "nice" work on SMP systems. The thing that broke "nice" on SMP systems was the adoption of separate run queues for each CPU and backing out that change in order to fix the problem is not an option so alternative solutions such as smpnice are required.

Even with no HT and MC, this patch has still has issues in the presence of different priority tasks... consider a simple DP system and run two instances of high priority tasks (simple infinite loop) and two normal priority

Re: [PATCH] sched: smpnice work around for active\_load\_balance()

Re: [PATCH] sched: smpnice work around for active\_load\_balance()

tasks. With "top" I observed that these normal priority tasks keep on jumping from one processor to another... Ideally with smpnice, we would assume that each processor should have two tasks (one high priority and another one with normal priority) ..

Yes, but you are failing to take into account the effect of the other tasks on your system (e.g. top) that run from time to time. If their burst of CPU use happens to coincide with some load balancing activity they will cause an imbalance to be detected (that is different to that which only considers your test tasks) and this will result in some tasks being moved. Beware the Heisenberg Uncertainty Principle :-).

Peter

--

Peter Williams pwil3058@xxxxxxxxxxxxxxxx

"Learning, n. The kind of ignorance distinguishing the studious."

-- Ambrose Bierce

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>