Subject: Re: [RFC][PATCH 4/6] Fix (bad?) interactions between SCHED_RT and SCHED_NORMAL tasks
Posted by Dmitry Adamushko on Tue, 12 Jun 2007 09:03:36 GMT
View Forum Message <> Reply to Message

On 11/06/07, Srivatsa Vaddagiri <vatsa@linux.vnet.ibm.com> wrote:
> Currently nr_running and raw_weighted_load fields in runqueue affect
> some CFS calculations (like distribute_fair_add, enqueue_sleeper etc).

[briefly looked.. a few comments so far]

(1)

I had an idea of per-sched-class 'load balance' calculator. So that update_load() (as in your patch) would look smth like :

```
...
struct sched_class *class = sched_class_highest;
unsigned long total = 0;

do {
        total += class->update_load(..., now);
        class = class->next;
    } while (class);
...
```

and e.g. update_load_fair() would become a fair_sched_class :: update_load().

That said, all the sched_classes would report a load created by their entities (tasks) over the last sampling period. Ideally, the calculation should not be merely based on the 'raw_weighted_load' but rather done in a similar way to update_load_fair() as in v17.

I'll take a look at how it can be mapped on the current v17 codebase (including your patches #1-3) and come up with some real code so we would have a base for discussion.

(2)

```
>  static void entity_tick(struct lrq *lrq, struct sched_entity *curr)
>  {
>         struct sched_entity *next;
>         struct rq *rq = lrq_rq(lrq);
>         u64 now = __rq_clock(rq);
>
> +       /* replay load smoothening for all ticks we lost */
```

```
> +       while (time_after_eq64(now, lrq->last_tick)) {
> +               update_load_fair(lrq);
> +               lrq->last_tick += TICK_NSEC;
> +       }
```

I think, it won't work properly this way. The first call returns a
load for last TICK_NSEC and all the consequent ones report zero load
('this_load = 0' internally).. as a result, we will get a lower load
than it likely was.

I guess, update_load_fair() (as it's in v17) could be slightly changed
to report the load for an interval of time over which the load
statistics have been accumulated (delta_exec_time and fair_exec_time):

update_load_fair(lrq, now - lrq->last_tick)

This new (second) argument would be used instead of TICK_NSEC
(internally in update_load_fair()) ... but again, I'll come up with
some code for further discussion.

> --
> Regards,
> vatsa
>

--
Best regards,
Dmitry Adamushko