
Subject: Re: [RFC][PATCH 5/6] core changes for group fairness
Posted by [Srivatsa Vaddagiri](#) on Thu, 14 Jun 2007 12:06:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, Jun 13, 2007 at 10:56:06PM +0200, Dmitry Adamushko wrote:

```
> >+static int balance_tasks(struct rq *this_rq, int this_cpu, struct rq
> >*busiest,
> >+      unsigned long max_nr_move, unsigned long
> >max_load_move,
> >+      struct sched_domain *sd, enum idle_type idle,
> >+      int *all_pinned, unsigned long *load_moved,
> >+      int this_best_prio, int best_prio, int
> >best_prio_seen,
> >+      void *iterator_arg,
> >+      struct task_struct *(*iterator_start)(void *arg),
> >+      struct task_struct *(*iterator_next)(void *arg));
>
> IMHO, it looks a bit frightening :)
```

I agree :) It is taking (ooops) 15 args (8 perhaps was the previous record in sched.c (move_tasks)!

```
> maybe it would be possible to
> create a structure that combines some relevant arguments .. at least,
> the last 3 ones.
```

How does this look?

```
struct balance_tasks_args {
    struct rq *this_rq, struct rq *busiest;
    unsigned long max_nr_move, unsigned long max_load_move;
    struct sched_domain *sd, enum idle_type idle;
    int this_best_prio, best_prio, best_prio_seen;
    int *all_pinned;
    unsigned long *load_moved;
    void *iterator_arg;
    struct task_struct *(*iterator_start)(void *arg);
    struct task_struct *(*iterator_next)(void *arg));
};
```

```
static int balance_tasks(struct balance_tasks_args *arg);
```

[down to one argument now!]

?

I will try this in my next iteration ..

```
> >-static int move_tasks(struct rq *this_rq, int this_cpu, struct rq
> >*busiest,
> >+static int balance_tasks(struct rq *this_rq, int this_cpu, struct rq
> >*busiest,
> >
> >         unsigned long max_nr_move, unsigned long
> >         max_load_move,
> >         struct sched_domain *sd, enum idle_type idle,
> >-         int *all_pinned)
> >+         int *all_pinned, unsigned long *load_moved,
> >+         int this_best_prio, int best_prio, int
> >best_prio_seen,
> >+         void *iterator_arg,
> >+         struct task_struct *(*iterator_start)(void *arg),
> >+         struct task_struct *(*iterator_next)(void *arg))
>
> I think, there is a possible problem here. If I'm not complete wrong,
> this function (move_tasks()) in the current mainline) can move more
> 'load' than specified by the 'max_load_move'..
```

Yes I think you are right. I will tackle this in next iteration.

Thanks for all your review so far!

--

Regards,
vatsa

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