Subject: Re: [PATCH] Hookup group-scheduler with task container infrastructure
Posted by Dmitry Adamushko on Wed, 12 Sep 2007 16:25:37 GMT
View Forum Message <> Reply to Message

Hi Srivatsa,

please find a few more minor comments below.

```
> [ ... ]
> +
> +/* destroy runqueue etc associated with a task group */
> +static void sched_destroy_group(struct container_subsys *ss,
> +                                struct container *cont)
> +{
> +       struct task_grp *tg = container_tg(cont);
> +       struct cfs_rq *cfs_rq;
> +       struct sched_entity *se;
> +       int i;
> +
> +       for_each_possible_cpu(i) {
> +               cfs_rq = tg->cfs_rq[i];
> +               list_del_rcu(&cfs_rq->leaf_cfs_rq_list);
> +       }
> +
> +       /* wait for possible concurrent references to cfs_rqs complete */
> +       synchronize_sched();
> +
> +       /* now it should be safe to free those cfs_rqs */
> +       for_each_possible_cpu(i) {
> +               cfs_rq = tg->cfs_rq[i];
> +               kfree(cfs_rq);
> +
> +               se = tg->se[i];
> +               kfree(se);
> +       }
> +
> +       kfree(tg);
> +}
```

kfree(tg->cfs_rq) && kfree(tg->se) ?

```
> +
> +/* change task's runqueue when it moves between groups */
> +static void sched_move_task(struct container_subsys *ss, struct container *cont,
> +                            struct container *old_cont, struct task_struct *tsk)
> +{
> +       int on_rq, running;
```

```
> +	unsigned long flags;
> +	struct rq *rq;
> +
> +	rq = task_rq_lock(tsk, &flags);
> +
> +	if (tsk->sched_class != &fair_sched_class)
> +		goto done;
```

this check should be redundant now with sched_can_attach() in place.

```
> +static void set_se_shares(struct sched_entity *se, unsigned long shares)
> +{
> +	struct cfs_rq *cfs_rq = se->cfs_rq;
> +	struct rq *rq = cfs_rq->rq;
> +	int on_rq;
> +
> +	spin_lock_irq(&rq->lock);
> +
> +	on_rq = se->on_rq;
> +	if (on_rq)
> +		__dequeue_entity(cfs_rq, se);
> +
> +	se->load.weight = shares;
> +	se->load.inv_weight = div64_64((1ULL<<32), shares);
```

A bit of nit-picking... are you sure, there is no need in non '__'
versions of dequeue/enqueu() here (at least, for the sake of
update_curr())? Although, I don't have -mm at hand at this very moment
and original -rc4 (that I have at hand) doesn't already have
'se->load' at all... so will look later.

```
>
> --
> Regards,
> vatsa
>
```

--
Best regards,
Dmitry Adamushko

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