Subject: Re: containers (was Re: -mm merge plans for 2.6.23)
Posted by Peter Zijlstra on Wed, 11 Jul 2007 12:06:33 GMT
View Forum Message <> Reply to Message

On Wed, 2007-07-11 at 04:42 -0700, Paul Jackson wrote:
> Srivatsa wrote:
> > The fact that we will have two interface for group scheduler in 2.6.24
> > is what worries me a bit (one user-id based and other container based).
>
> Yeah.
>
> One -could- take linear combinations, as Peter drew in his ascii art,
> but would one -want- to do that?

I'd very much like to have it, but that is just me. We could take a
weight of 0 to mean disabling of that grouping and default to that. That
way it would not complicate regular behaviour.

It could be implemented with a simple hashing scheme where
sched_group_hash(tsk) and sched_group_cmp(tsk, group->some_task) could
be used to identify a schedule group.

pseudo code:

```
u64 sched_group_hash(struct task_struct *tsk)
{
 u64 hash = 0;

 if (tsk->pid->weight)
  hash_add(&hash, tsk->pid);

 if (tsk->pgrp->weight)
  hash_add(&hash, tsk->pgrp);

 if (tsk->uid->weight)
  hash_add(&hash, tsk->uid);

 if (tsk->container->weight)
  hash_add(&hash, tsk->container);

 ...

 return hash;
}

s64 sched_group_cmp(struct task_struct *t1, struct task_struct *t2)
{
 s64 cmp;
```

```
 if (t1->pid->weight || t2->pid->weight) {
  cmp = t1->pid->weight - t2->pid->weight;
  if (cmp)
   return cmp;
 }

 ...

 return 0;
}

u64 sched_group_weight(struct task_struct *tsk)
{
 u64 weight = 1024; /* 1 fixed point 10 bits */

 if (tsk->pid->weight) {
  weight *= tsk->pid->weight;
  weight /= 1024;
 }

 ....

 return weight;
}
```

_____