Subject: Re: [PATCH, v6 3/3] cgroups: introduce timer slack controller
Posted by Matt Helsley on Tue, 15 Feb 2011 00:00:55 GMT
View Forum Message <> Reply to Message

On Tue, Feb 15, 2011 at 12:59:40AM +0200, Kirill A. Shutemov wrote:
> On Mon, Feb 14, 2011 at 05:59:26AM -0800, Matt Helsley wrote:
> > On Mon, Feb 14, 2011 at 03:06:27PM +0200, Kirill A. Shutsemov wrote:
> > > From: Kirill A. Shutemov <kirill@shutemov.name>

<snip>

> > > + list_for_each_entry(cur, &cgroup->children, sibling) {
> > > + child = cgroup_to_tslack_cgroup(cur);
> > > + if (type == TIMER_SLACK_MIN && val > child->min_slack_ns)
> > > +   return -EBUSY;
> > > + if (type == TIMER_SLACK_MAX && val < child->max_slack_ns)
> > > +   return -EBUSY;
> > > + }
> >
> > This doesn't look right. Child cgroups should not constrain their
> > parents. Instead you should allow the change and propagate the
> > constraint to the children.
>
> See discussion with Thomas.

<OK, shifting this topic to that thread>
<snip>

> > > +static struct cftype files[] = {
> > > + {
> > > + .name = "set_slack_ns",
> > > + .write_u64 = tslack_write_set_slack_ns,
> > > + },
> > > + {
> > > + .name = "min_slack_ns",
> > > + .private = TIMER_SLACK_MIN,
> > > + .read_u64 = tslack_read_range,
> > > + .write_u64 = tslack_write_range,
> > > + },
> > > + {
> > > + .name = "max_slack_ns",
> > > + .private = TIMER_SLACK_MAX,
> > > + .read_u64 = tslack_read_range,
> > > + .write_u64 = tslack_write_range,
> > > + },
> >
> > I didn't get a reply on how a max_slack_ns is useful. It seems
> > prudent to add as little interface as possible and only when

> > we clearly see the utility of it.
>
> For example, you can create two groups (excluding root cgroup):
>
> default - timer slack range 50000-50000
> relaxed - timer slack range 500000-unlimited.
>
> Now you can drag tasks between these group without need to reset value on
> relaxed -> default transition.

Perhaps you misunderstood my point.

Yes, I can see that a maximum allows you to do counter-productive/pointless
little tricks like "setting" the timer slack when you move the task. I
just don't get the point of it. Why is setting a maximum timer slack useful?
If anything it seems like it would be quite counterproductive or pointless
at best because limiting the amount of timer slack would not improve
the wakeup situation -- it could easily make it worse. Are there
any negative consequences to allowing timer slacks as large as
userspace requests -- perhaps even up to ULLONG_MAX? If there are none then
why should we bother providing userspace a knob to set and enforce such a
limit?

Cheers,
-Matt Helsley

_____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containe rs