Subject: Re: [PATCH, v3 2/2] cgroups: introduce timer slack subsystem
Posted by Matt Helsley on Sun, 06 Feb 2011 02:49:51 GMT
View Forum Message <> Reply to Message

On Thu, Feb 03, 2011 at 11:41:38AM +0200, Kirill A. Shutemov wrote:
> On Wed, Feb 02, 2011 at 09:46:16PM -0800, Matt Helsley wrote:
> > On Wed, Feb 02, 2011 at 10:47:36PM +0200, Kirill A. Shutsemov wrote:
> > > From: Kirill A. Shutemov <kirill@shutemov.name>

<snip>

> > > diff --git a/kernel/cgroup_timer_slack.c b/kernel/cgroup_timer_slack.c
> > > new file mode 100644
> > > index 0000000..a343a50
> > > --- /dev/null
> > > +++ b/kernel/cgroup_timer_slack.c

<snip>

> > > +static int tslack_write_set_slack_ns(struct cgroup *cgroup, struct cftype *cft,
> > > + u64 val)
> > > +{
> > > + struct timer_slack_cgroup *tslack_cgroup;
> > > + struct cgroup_iter it;
> > > + struct task_struct *task;
> > > +
> > > + tslack_cgroup = cgroup_to_tslack_cgroup(cgroup);
> > > + if (!val || val < tslack_cgroup->min_slack_ns ||
> >
> > Why is a val of 0 disallowed? I know having slack is good, but for
> > an administrator or tool that doesn't care about number of wakeups
> > and cares more about wringing out performance a slack of
> > 0 seems acceptable. Is this just here to be consistent with the
> > values passed in via prctl?
>
> Yes, it's to consistent with the prctl(). I don't think that it's good
> idea to allow to set timer_slack outside of range prctl() allows. It may
> lead to interface abuse.

Hmm, I was just thinking that 0 timer slack might be useful. But I
suppose you could just as easily set it to 1 and nobody would notice.

> > > +   val > tslack_cgroup->max_slack_ns)
> > > + return -EINVAL;
> >
> > Shouldn't it be EPERM and not EINVAL?
> >
> > The write(2) man page says: "Other errors may occur, depending on the

> > object connected to fd." So I think EPERM is fine and more descriptive.
>
> What do you think about -EINVAL for (val == 0) and -EPERM for rest?

OK, that makes sense to me given both of our points above.

Cheers,
-Matt Helsley

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

---

Subject: Re: [PATCH, v3 2/2] cgroups: introduce timer slack subsystem
Posted by Kirill A. Shutsemov on Mon, 07 Feb 2011 09:48:51 GMT
View Forum Message <> Reply to Message

On Sat, Feb 05, 2011 at 06:49:51PM -0800, Matt Helsley wrote:
> On Thu, Feb 03, 2011 at 11:41:38AM +0200, Kirill A. Shutemov wrote:
> > On Wed, Feb 02, 2011 at 09:46:16PM -0800, Matt Helsley wrote:
> > > On Wed, Feb 02, 2011 at 10:47:36PM +0200, Kirill A. Shutsemov wrote:
> > > > From: Kirill A. Shutemov <kirill@shutemov.name>
>
> <snip>
>
> > > > diff --git a/kernel/cgroup_timer_slack.c b/kernel/cgroup_timer_slack.c
> > > > new file mode 100644
> > > > index 0000000..a343a50
> > > > --- /dev/null
> > > > +++ b/kernel/cgroup_timer_slack.c
>
> <snip>
>
> > > > +static int tslack_write_set_slack_ns(struct cgroup *cgroup, struct cftype *cft,
> > > > + u64 val)
> > > > +{
> > > > + struct timer_slack_cgroup *tslack_cgroup;
> > > > + struct cgroup_iter it;
> > > > + struct task_struct *task;
> > > > +
> > > > + tslack_cgroup = cgroup_to_tslack_cgroup(cgroup);
> > > > + if (!val || val < tslack_cgroup->min_slack_ns ||
> > >
> > > Why is a val of 0 disallowed? I know having slack is good, but for
> > > an administrator or tool that doesn't care about number of wakeups
> > > and cares more about wringing out performance a slack of
> > > 0 seems acceptable. Is this just here to be consistent with the

> > > values passed in via prctl?
> >
> > Yes, it's to consistent with the prctl(). I don't think that it's good
> > idea to allow to set timer_slack outside of range prctl() allows. It may
> > lead to interface abuse.
>
> Hmm, I was just thinking that 0 timer slack might be useful. But I
> suppose you could just as easily set it to 1 and nobody would notice.

I've rechecked once again. it lookes cleaner to allow 0 as timer slack value.
I allowed it in version 4 of the patchset.

--
Kirill A. Shutemov

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