Subject: Re: [PATCH 1/1, v6] cgroup/freezer: add per freezer duty ratio control
Posted by jacob.jun.pan on Thu, 10 Feb 2011 22:22:21 GMT
View Forum Message <> Reply to Message

On Thu, 10 Feb 2011 11:11:17 -0800
Matt Helsley <matthltc@us.ibm.com> wrote:

> On Wed, Feb 09, 2011 at 07:06:15PM -0800, Arjan van de Ven wrote:
> > On 2/9/2011 7:04 PM, Matt Helsley wrote:
> > >On Tue, Feb 08, 2011 at 05:05:41PM -0800,
> > >jacob.jun.pan@linux.intel.com wrote:
> > >>From: Jacob Pan<jacob.jun.pan@linux.intel.com>
> > >>
> > >>Freezer subsystem is used to manage batch jobs which can start
> > >>stop at the same time. However, sometime it is desirable to let
> > >>the kernel manage the freezer state automatically with a given
> > >>duty ratio.
> > >>For example, if we want to reduce the time that backgroup apps
> > >>are allowed to run we can put them into a freezer subsystem and
> > >>set the kernel to turn them THAWED/FROZEN at given duty ratio.
> > >>
> > >>This patch introduces two file nodes under cgroup
> > >>freezer.duty_ratio_pct and freezer.period_sec
> > >>
> > >>Usage example: set period to be 5 seconds and frozen duty ratio
> > >>90% [root@localhost aoa]# echo 90> freezer.duty_ratio_pct
> > >>[root@localhost aoa]# echo 5000> freezer.period_ms
> > >I kept wondering how this was useful when we've got the "cpu"
> > >subsystem because for some reason "duty cycle" made me think this
> > >was a scheduling policy knob. In fact, I'm pretty sure it is -- it
> > >just happens to sometimes reduce power consumption.
> > >
> > >Have you tried using the cpu cgroup subsystem's share to see if it
> > >can have a similar effect?
> >
> > does the cpu cgroup system work on a 20 to 30 second time window?
>
> I don't think so -- it works directly with the scheduler IIRC.
>
I played with cpu subsystem a little today, it is for real-time tasks
only. By data type of cpu.rt_period_us cpu.rt_runtime_us, it
actually has a pretty long time window (35 mins, int type at usec
resolution).
For some reason, I could not even get cpu subsystem to work with RT
task to work on 38-rc2 kernel. Here is what I did
- mount and create cpu cgroup fs
- launch task with SCHED_RR
- attach task to my newly created cgroup

- adjust cpu.rt_period_us cpu.rt_runtime_us
But it never changed percentage of runtime. The ask in the cpu cgroup always run at 100% or more than the runtime_us as I specified. I have tried both with system idle and background tasks.

I do agree that dealing with group scheduler directly might be more natural. but the hurdle might be changing cpu subsystem to support non-rt, and deal with scheduler heuristics.

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