
Subject: Re: [RFC][mm] [1/2] Simple stats for cpu resource controller

Posted by [Balaji Rao](#) on Sat, 05 Apr 2008 20:39:57 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Sunday 06 April 2008 01:10:41 am Dhaval Giani wrote:

<snip>

> >

> > +`#ifdef CONFIG_CGROUP_SCHED`

> > +`enum cpu_cgroup_stat_index {`

> > + `CPU_CGROUP_STAT_UTIME, /* Ustime of the task group */`

> > + `CPU_CGROUP_STAT_STIME, /* Kerntime of the task group */`

> > +

> > + `CPU_CGROUP_STAT_NSTATS,`

>

> why the extra space?

Just to keep things clearly separated. If you've not noticed,
`CPU_CGROUP_STAT_NSTATS` is not a stat.

>

> > +};

> > +

> > +`struct cpu_cgroup_stat_cpu {`

> > + `s64 count[CPU_CGROUP_STAT_NSTATS];`

>

> u64? time does not go negative :)

Right. But these stats are not only going to measure time. We need the same
variables for measuring other stats as well. I'm not sure if we would
encounter scheduler stats that would count negative.

Balbir, what do you say ?

> count also is not very clear? Can you give a more descriptive name?

>

ok. How does 'value' look ?

<snip>

> > +`static s64 cpu_cgroup_read_stat(struct cpu_cgroup_stat *stat,`

> > + `enum cpu_cgroup_stat_index idx)`

> > +{

> > + `int cpu;`

> > + `s64 ret = 0;`

> > + `unsigned long flags;`

>

> > +

> > + `local_irq_save(flags);`

>

> I am just wondering. Is `local_irq_save()` enough?

>

Hmmm.. You are right. This does not prevent concurrent updates on other CPUs from crossing a 32bit boundary. Am not sure how to do this in a safe way. I can only think of using atomic64_t now..

```
> > + for_each_possible_cpu(cpu)
> > + ret += stat->cpustat[cpu].count[idx];
> > + local_irq_restore(flags);
> > +
> > + return ret;
> > +}
> > +
```

Thanks for the review.

--

regards,
Balaji Rao
Dept. of Mechanical Engineering,
National Institute of Technology Karnataka, India

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