
Subject: Re: [RFC][mm] Simple stats for cpu resource controller v4

Posted by [Balaji Rao](#) on Mon, 12 May 2008 05:41:24 GMT

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

On Monday 12 May 2008 08:24:45 am Li Zefan wrote:

> Balaji Rao wrote:

> > /*

> > @@ -3892,8 +3923,17 @@ void account_system_time(struct task_struct *p,

> > int hardirq_offset, cpustat->irq = cputime64_add(cpustat->irq, tmp);

> > else if (softirq_count())

> > cpustat->softirq = cputime64_add(cpustat->softirq, tmp);

> > - else if (p != rq->idle)

> > + else if (p != rq->idle) {

> > cpustat->system = cputime64_add(cpustat->system, tmp);

> > + #ifdef CONFIG_CGROUP_SCHED

> > + {

> > + struct task_group *tg;

> > + tg = task_group(p);

> > + __cpu_cgroup_stat_add(tg->stat, CPU_CGROUP_STAT_STIME,

> > + cputime_to_msecs(cputime));

> > + }

> > + }

>

> You should put this '}' after '#endif'

>

Yes, my mistake. Will fix it.

> > + #endif

> > else if (atomic_read(&rq->nr_iowait) > 0)

> > cpustat->iowait = cputime64_add(cpustat->iowait, tmp);

> > else

> > @@ -8179,10 +8219,26 @@ static inline struct task_group *cgroup_tg(struct

> > cgroup *cgrp) struct task_group, css);

> > }

> >

> > + static void cpu_cgroup_initialize(int early)

> > + {

> > + int i;

> > + struct cpu_cgroup_stat *stat;

> > +

> > + if (!early) {

> > + stat = kmalloc(sizeof(struct cpu_cgroup_stat)

> > + , GFP_KERNEL);

> > + for (i = 0; i < CPU_CGROUP_STAT_NSTATS; i++)

> > + percpu_counter_init(

> > + &stat->cpustat[i], 0);

> > + init_task_group.stat = stat;

> > + }

> > + }

```

> > +
> > static struct cgroup_subsys_state *
> > cpu_cgroup_create(struct cgroup_subsys *ss, struct cgroup *cgrp)
> > {
> >     struct task_group *tg;
> >     int i;
> >
> >     if (!cgrp->parent) {
> >         /* This is early initialization for the top cgroup */
> >         @@ -8198,6 +8254,10 @@ cpu_cgroup_create(struct cgroup_subsys *ss, struct
> > cgroup *cgrp) if (IS_ERR(tg))
> >         return ERR_PTR(-ENOMEM);
> >
> > + tg->stat = kmalloc(sizeof(struct cpu_cgroup_stat), GFP_KERNEL);
> > + for (i = 0; i < CPU_CGROUP_STAT_NSTATS; i++)
> > +     percpu_counter_init(&tg->stat->cpustat[i], 0);
> > +
>
> I guess you forgot to free those things in cpu_cgroup_destroy().
Oh. Sorry, I totally missed it.
>
> > /* Bind the cgroup to task_group object we just created */

```

--

Warm Regards,

Balaji Rao
 Dept. of Mechanical Engineering
 NITK

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