
Subject: [PATCH v4 2/4] expose fine-grained per-cpu data for cpuacct stats
Posted by [Glauber Costa](#) on Tue, 05 Jun 2012 14:49:39 GMT

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

The cpuacct cgroup already exposes user and system numbers in a per-cgroup fashion. But they are a summation along the whole group, not a per-cpu figure. Also, they are coarse-grained version of the stats usually shown at places like /proc/stat.

I want to have enough cgroup data to emulate the /proc/stat interface. To achieve that, I am creating a new file "stat_percpu" that displays the fine-grained per-cpu data. The original data is left alone.

The format of this file is as follows:

```
1line header
cpux val1 val2 ... valn.
```

It is the same format used for the cpu part /proc/stat, except for the header, that may allow us to add fields in the future if they prove themselves needed.

Signed-off-by: Glauber Costa <glommer@parallels.com>

CC: Peter Zijlstra <a.p.zijlstra@chello.nl>

CC: Paul Turner <pjt@google.com>

```
kernel/sched/core.c | 37 ++++++
1 file changed, 37 insertions(+)
```

```
diff --git a/kernel/sched/core.c b/kernel/sched/core.c
```

```
index 220d416..c058189 100644
```

```
--- a/kernel/sched/core.c
```

```
+++ b/kernel/sched/core.c
```

```
@@ -8178,6 +8178,39 @@ static int cpuacct_stats_show(struct cgroup *cgrp, struct cftype *cft,
    return 0;
}
```

```
+static inline void do_fill_seq(struct seq_file *m, struct cpuacct *ca,
+    int cpu, int index)
```

```
+{
+    struct kernel_cpustat *kcpustat = per_cpu_ptr(ca->cpustat, cpu);
+    u64 val;
+
+    val = cputime64_to_clock_t(kcpustat->cpustat[index]);
+    seq_put_decimal_ull(m, ' ', val);
+}
```

```
+
+static int cpuacct_stats_percpu_show(struct cgroup *cgrp, struct cftype *cft,
+    struct seq_file *m)
```

```

+{
+ struct cpuacct *ca = cgroup_ca(cgrp);
+ int cpu;
+
+ seq_printf(m, "user nice system irq softirq guest guest_nice\n");
+
+ for_each_online_cpu(cpu) {
+ seq_printf(m, "cpu%d", cpu);
+ do_fill_seq(m, ca, cpu, CPUTIME_USER);
+ do_fill_seq(m, ca, cpu, CPUTIME_NICE);
+ do_fill_seq(m, ca, cpu, CPUTIME_SYSTEM);
+ do_fill_seq(m, ca, cpu, CPUTIME_IRQ);
+ do_fill_seq(m, ca, cpu, CPUTIME_SOFTIRQ);
+ do_fill_seq(m, ca, cpu, CPUTIME_GUEST);
+ do_fill_seq(m, ca, cpu, CPUTIME_GUEST_NICE);
+ seq_putc(m, '\n');
+ }
+
+ return 0;
+}
+
static struct cftype files[] = {
{
.name = "usage",
@@ -8192,6 +8225,10 @@ static struct cftype files[] = {
.name = "stat",
.read_map = cpuacct_stats_show,
},
+ {
+ .name = "stat_percpu",
+ .read_seq_string = cpuacct_stats_percpu_show,
+ },
{ } /* terminate */
};

```

--

1.7.10.2
