Subject: [PATCH 4/5] expose fine-grained per-cpu data for cpuacct stats
Posted by Glauber Costa on Thu, 02 Feb 2012 14:19:31 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.

Note that in this first version, I am using clock_t units, being quite
proc-centric. It made my testing easier, but I am happy to show any units
you guys would prefer.

Signed-off-by: Glauber Costa <glommer@parallels.com>
---
 kernel/sched/core.c | 28 ++++++++++++++++++++++++++++
 1 files changed, 28 insertions(+), 0 deletions(-)

diff --git a/kernel/sched/core.c b/kernel/sched/core.c
index 91ea913..013ca9c 100644
--- a/kernel/sched/core.c
+++ b/kernel/sched/core.c
@@ -8308,6 +8308,29 @@ static int cpuacct_stats_show(struct cgroup *cgrp, struct cftype *cft,
  return 0;
 }

+static int cpuacct_stats_percpu_show(struct cgroup *cgrp, struct cftype *cft,
+   struct seq_file *m)
+{
+ struct cpuacct *ca = cgroup_ca(cgrp);
+ int cpu;
+
+ for_each_online_cpu(cpu) {
+ struct kernel_cpustat *kcpustat = per_cpu_ptr(ca->cpustat, cpu);
+ seq_printf(m,
+ "cpu%d %llu %llu %llu %llu %llu %llu %llu\n", cpu,
+ (unsigned long long)cputime_to_clock_t(kcpustat->cpustat[CPUTIME_USER]),
+ (unsigned long long)cputime_to_clock_t(kcpustat->cpustat[CPUTIME_NICE]),
+ (unsigned long long)cputime_to_clock_t(kcpustat->cpustat[CPUTIME_SYSTEM]),
+ (unsigned long long)cputime_to_clock_t(kcpustat->cpustat[CPUTIME_IRQ]),
+ (unsigned long long)cputime_to_clock_t(kcpustat->cpustat[CPUTIME_SOFTIRQ]),
+ (unsigned long long)cputime_to_clock_t(kcpustat->cpustat[CPUTIME_GUEST]),
+ (unsigned long long)cputime_to_clock_t(kcpustat->cpustat[CPUTIME_GUEST_NICE])
+);

```
+ }
+
+ return 0;
+}
+
 static struct cftype files[] = {
  {
    .name = "usage",
@@ -8322,6 +8345,11 @@ static struct cftype files[] = {
    .name = "stat",
    .read_map = cpuacct_stats_show,
  },
+ {
+  .name = "stat_percpu",
+  .read_seq_string = cpuacct_stats_percpu_show,
+ },
+
 };

 static int cpuacct_populate(struct cgroup_subsys *ss, struct cgroup *cgrp)
--
1.7.7.4
```