Subject: [RFC] cgroup basic comounting
Posted by Glauber Costa on Fri, 16 Dec 2011 12:29:09 GMT
View Forum Message <> Reply to Message

Turns out that most of the infrastructure we need to put two controllers in the same hierarchy is by far already into place. All we need to do is not failing when we specify two of them.

With this, we can effectively guarantee that by comounting cpu and cpuacct, we'll have the same set of tasks, therefore allowing us to use cpu cgroup data to fill in the usage fields in cpuacct.

I decided not to stabilish any dependency between cgroups as Li previously did: cgroups may or may not be comounted, and any of them can be combined (I don't see a reason to prevent any combination).

After testing and some trials, I could verify that the current mount behavior plays well under the plans, so I didn't change it. That is:

 * If subsystems A and B aren't mounted, we can comount them.
 * If subsystem A is mounted, but B is not:
   * we can comount them if A has no children,
   * we fail otherwise
 * If subsystems A and B are comounted at a location, we can't
   mount any of them separately at another point. We do can mount
   them together.
 * If subsystems A and B are comounted at a location,
   * we can comount a third subsystem C, if they have no children
   * we fail otherwise

Paul,

Please let me know if this is tuned with the idea you had in mind.
If this is okay, I patch that extracts usage from cpu cgroup data
in case of comount would follow.

Signed-off-by: Glauber Costa <glommer@parallels.com>
CC: Paul Turner <pjt@google.com>
CC: Li Zefan <lizf@cn.fujitsu.com>
---
 kernel/cgroup.c | 4 ++--
 1 files changed, 2 insertions(+), 2 deletions(-)

diff --git a/kernel/cgroup.c b/kernel/cgroup.c
index 1fd7867..e894a4f 100644
--- a/kernel/cgroup.c
+++ b/kernel/cgroup.c
@@ -1211,9 +1211,9 @@ static int parse_cgroupfs_options(char *data, struct cgroup_sb_opts

```
*opts)
    set_bit(i, &opts->subsys_bits);
    one_ss = true;

-   break;
+   continue;
   }
-  if (i == CGROUP_SUBSYS_COUNT)
+  if (opts->subsys_bits == 0)
    return -ENOENT;
  }

--
1.7.6.4
```