Subject: [RFC] Default child of a cgroup
Posted by Srivatsa Vaddagiri on Thu, 31 Jan 2008 02:40:49 GMT
View Forum Message <> Reply to Message

Hi,
As we were implementing multiple-hierarchy support for CPU controller, we hit some oddities in its implementation, partly related to current cgroups implementation. Peter and I have been debating on the exact solution and I thought of bringing that discussion to lkml.

Consider the cgroup filesystem structure for managing cpu resource.

```
# mount -t cgroup -ocpu,cpuacct none /cgroup
# mkdir /cgroup/A
# mkdir /cgroup/B
# mkdir /cgroup/A/a1
```

will result in:

```
/cgroup
   |------<tasks>
   |------<cpuacct.usage>
    |------<cpu.shares>
   |
   |----[A]
   |     |----<tasks>
   |     |----<cpuacct.usage>
   |     |----<cpu.shares>
   |     |
   |     |---[a1]
   |           |----<tasks>
   |           |----<cpuacct.usage>
   |           |----<cpu.shares>
   |           |
   |
   |----[B]
   |     |----<tasks>
   |     |----<cpuacct.usage>
   |     |----<cpu.shares>
   |
```

Here are some questions that arise in this picture:

1. What is the relationship of the task-group in A/tasks with the task-group in A/a1/tasks? In otherwords do they form siblings of the same parent A?

2. Somewhat related to the above question, how much resource should the
   task-group A/a1/tasks get in relation to A/tasks? Is it 1/2 of parent
   A's share or 1/(1 + N) of parent A's share (where N = number of tasks
   in A/tasks)?

3. What should A/cpuacct.usage reflect? CPU usage of A/tasks? Or CPU usage
   of all siblings put together? It can reflect only one, in which case
   user has to manually derive the other component of the statistics.

It seems to me that tasks in A/tasks form what can be called the
"default" child group of A, in which case:

4. Modifications to A/cpu.shares should affect the parent or its default
   child group (A/tasks)?

To avoid these ambiguities, it may be good if cgroup create this
"default child group" automatically whenever a cgroup is created?
Something like below (not the absence of tasks file in some directories
now):

```
 /cgroup
    |
    |------<cpuacct.usage>
     |------<cpu.shares>
    |
      |---[def_child]
    |     |----<tasks>
    |     |----<cpuacct.usage>
    |     |----<cpu.shares>
    |     |
    |
    |----[A]
    |     |
    |     |----<cpuacct.usage>
    |     |----<cpu.shares>
    |     |
    |     |---[def_child]
    |     |     |----<tasks>
    |     |     |----<cpuacct.usage>
    |     |     |----<cpu.shares>
    |     |     |
    |     |
    |     |---[a1]
    |           |
    |           |----<cpuacct.usage>
    |           |----<cpu.shares>
    |           |
```

```
|        |---[def_child]
|        |       |---<tasks>
|        |       |---<cpuacct.usage>
|        |       |---<cpu.shares>
|        |       |
|
|----[B]
|     |
|     |----<cpuacct.usage>
|     |----<cpu.shares>
|     |
|     |---[def_child]
|     |     |----<tasks>
|     |     |----<cpuacct.usage>
|     |     |----<cpu.shares>
|     |     |
```

Note that user cannot create subdirectories under def_child with this scheme! I am also not sure what impact this will have on other resources like cpusets ..

Thoughts?

--
Regards,
vatsa

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