

edsionte's TechBlog

No Pains, No Gains

Cgroup简介-子系统与层级

2013年3月24日 由 edsionte

[留言 »](#)

Cgroup是一种资源控制机制，它将操作系统中的所有进程以组为单位划分，所有进程组以层级结构进行组织。cgroup为每个进程组都指定一组访问资源的行为，这些行为限制了该组进程对资源的访问。

1基本概念

子系统（subsystem）实际上是cgroup对进程组进行资源控制的具体体现。子系统具有多种类型，每个类型的子系统都代表一种系统资源，比如CPU、memory等。当创建一个cgroup实例时，必须至少指定一种子系统。也就是说，这个新建的进程组在访问子系统对应的系统资源时就有了一些限制。具体的限制项与子系统的类型有关。

Cgroup中进程组的层级关系与Linux中进程的层级关系比较类似。在Linux操作系统中，一个进程通过fork()系统调用创建了一个子进程，这两个进程之间存在父子这样的等级关系，并且子进程可以继承父进程的一些资源。系统中所有的进程形成一个树形的等级关系，每个进程都唯一的位于进程树中的某一个位置。

对于cgroup来说，cgroup实例之间也是具体级别关系的，但是它们层级关系是为了更细粒度的对进程组进行资源控制。同时，子cgroup会继承父cgroup的对资源的控制属性。

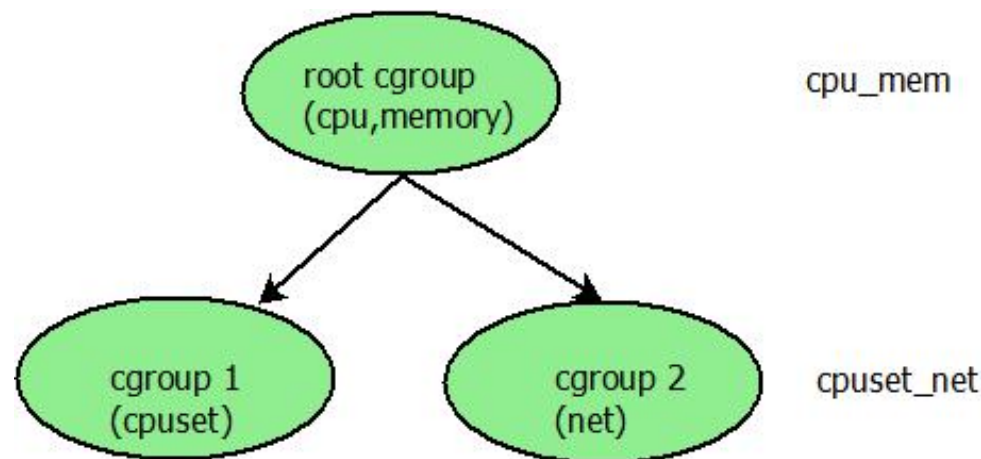
2子系统与层级的关系

如果要创建一个cgroup实例，通过下面的命令即可完成：

```
1 | mount -t cgroup -o cpu,memory cpu_memory /cgroup/cpu_memory
```

可以看到，通过Linux中挂载文件系统的方法就可以创建cgroup实例。其中-t选项指定文件系统类型为cgroup类型，-o指定本次创建的cgroup实例与cpu和memory子系统（或资源）关联，cpu_memory指定了当前cgroup实例在整个cgroup树中所处的层级名称，最后的路径为文件系统挂载点。关于cgroup文件系统本文稍后说明。

图1是一个系统中的cgroup结构图，它包含两个层级，即第一层级cpu_mem和第二层级cpuset_net。接下来以该图为例说明cgroup子系统和层级之间的设置规则。



[http://edsionte.com/techblog/wordpress/wp-content/uploads/2013/03/cgroup_example.jpeg]

图1

1. 系统中第一个被创建的cgroup被称为root cgroup，该cgroup的成员包含系统中所有的进程。

如图1所示，root cgroup位于cpu_mem层，它包含了系统中的所有进程；root cgroup又被分为cgroup1和cgroup2两个子cgroup，它们位于第二层级cpuset_net中。

2. 一个子系统只能位于一个层级中。

如图1中所示，cpu子系统位于第一层级cpu_mem中，那么这个子系统将不能再位于第二层级中。不过这种设置规则并不影响子cgroup对cpu子系统的使用，因为所有的子cgroup都将继承root cgroup所属的cpu子系统。

3. 每个层级中可以关联多个子系统。

图1中第一层级cpu_mem关联了cpu和memory两个子系统。

4. 一个进程可以位于不同层级的cgroup中。

由于root cgroup包含了系统中所有的进程，因此cgroup2中的进程P也位于root cgroup中。从资源控制角度来说，进程P所在的进程组在访问cpu、memory和net时会受到资源限制。

5. 一个进程创建了子进程后，该子进程默认为父进程所在cgroup的成员。

子进程被创建后，继承父进程的cgroup，但是后续可根据需求将子进程移动到其他cgroup中。

- [上一篇: Cgroup简介-概述](#)
- [下一篇: Cgroup简介-资源控制的实现](#)

发表在 性能隔离

Tags: cgroup 子系统 层次

若要跟踪这篇文章的任何更新，你可以使用 [RSS 2.0 Feed](#). 你可以直接转到文章底部进行评论，Pinging目前已关闭

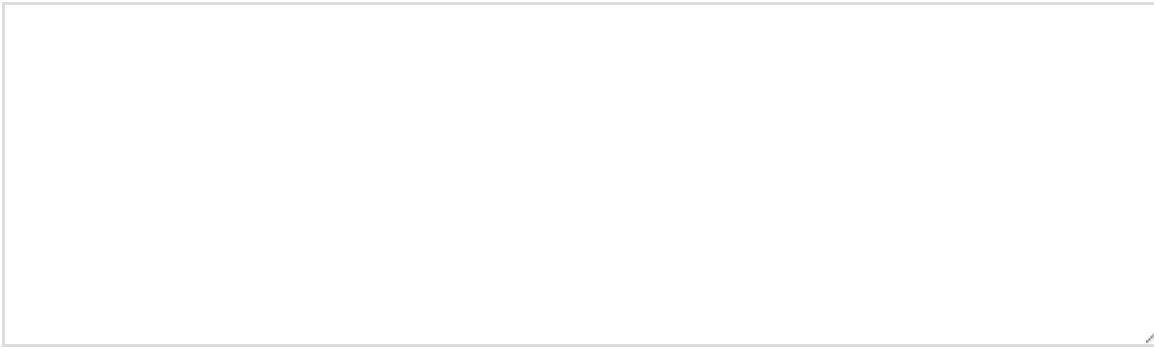
发表评论

姓名 (required)

电子邮件（不会被公开）(required)

站点

请回答问题: 3 + 5 = ?



发表评论

© 2015 edsionte's TechBlog · Proudly powered by [WordPress](#) & [Green Park 2](#) 由 [Cordobo](#).

[XHTML 1.0验证通过](#) | [CSS3验证通过](#)



[返回顶端](#)