**cGroup使用调研**

CGroup是Control Groups 的缩写，是Linux内核提供的一种可以限制、记录、隔离进程组(process groups)所使用的物力资源(如cpu memory i/o 等等) 的机制。

CGroup是将任意进程进行分组化管理的 Linux 内核功能。CGroup本身是提供将进程进行分组化管理的功能和接口的基础结构，I/O 或内存的分配控制等具体的资源管理功能是通过这个功能来实现的。这些具体的资源管理功能称为CGroup子系统或控制器。CGroup 子系统有控制内存的Memory控制器、控制进程调度的CPU控制器等。运行中的内核可以使用的Cgroup子系统由/proc/cgroup 来确认。

CGroup 提供了一个 CGroup 虚拟文件系统，作为进行分组管理和各子系统设置的用户接口。要使用 CGroup，必须挂载 CGroup 文件系统。这时通过挂载选项指定使用哪个子系统。

**Cgroups提供的功能：**

1.限制进程组可以使用的资源数量（Resource limiting ）。比如：memory子系统可以为进程组设定一个memory使用上限，一旦进程组使用的内存达到限额再申请内存，就会出发OOM（out of memory）。

2.进程组的优先级控制（Prioritization ）。比如：可以使用cpu子系统为某个进程组分配特定cpu share。

3.记录进程组使用的资源数量（Accounting ）。比如：可以使用cpuacct子系统记录某个进程组使用的cpu时间

4.进程组隔离（Isolation）。比如：使用ns子系统可以使不同的进程组使用不同的namespace，以达到隔离的目的，不同的进程组有各自的进程、网络、文件系统挂载空间。

5.进程组控制（Control）。比如：使用freezer子系统可以将进程组挂起和恢复。

**Cgroup相关概念介绍：**

1.任务（task）。在cgroups中，任务就是系统的一个进程；

2.控制族群（control group）。控制族群就是一组按照某种标准划分的进程。Cgroups中的资源控制都是以控制族群为单位实现。一个进程可以加入到某个控制族群，也从一个进程组迁移到另一个控制族群。一个进程组的进程可以使用cgroups以控制族群为单位分配的资源，同时受到cgroups以控制族群为单位设定的限制；

3.层级（hierarchy）。控制族群可以组织成hierarchical的形式，既一颗控制族群树。控制族群树上的子节点控制族群是父节点控制族群的孩子，继承父控制族群的特定的属性；

4.子系统（subsystem）。一个子系统就是一个资源控制器，比如cpu子系统就是控制cpu时间分配的一个控制器。子系统必须附加（attach）到一个层级上才能起作用，一个子系统附加到某个层级以后，这个层级上的所有控制族群都受到这个子系统的控制。

**相关概念的相互关系：**

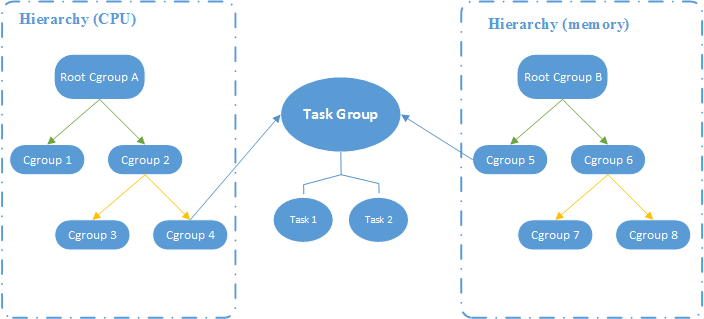
每次在系统中创建新层级时，该系统中的所有任务都是那个层级的默认cgroup（我们称之为root cgroup，此 cgroup 在创建层级时自动创建，后面在该层级中创建的cgrou 都是此cgroup的后代）的初始成员；

一个子系统最多只能附加到一个层级；

一个层级可以附加多个子系统；

一个任务可以是多个cgroup的成员，但是这些cgroup必须在不同的层级；

系统中的进程（任务）创建子进程（任务）时，该子任务自动成为其父进程所在cgroup的成员。然后可根据需要将该子任务移动到不同的cgroup中，但开始时它总是继承其父任务的cgroup。



由上图所示，CGroup层级关系显示，CPU和Memory两个子系统有自己独立的层级系统，而又通过Task Group取得关联关系。

**cGroup特点**

在cgroups中，任务就是系统的一个进程。

控制族群（control group）。控制族群就是一组按照某种标准划分的进程。Cgroups 中的资源控制都是以控制族群为单位实现。一个进程可以加入到某个控制族群，也从一个进程组迁移到另一个控制族群。一个进程组的进程可以使用cgroups以控制族群为单位分配的资源，同时受到cgroups以控制族群为单位设定的限制。

层级（hierarchy）。控制族群可以组织成hierarchical的形式，既一颗控制族群树。控制族群树上的子节点控制族群是父节点控制族群的孩子，继承父控制族群的特定的属性。

子系统（subsytem）。一个子系统就是一个资源控制器，比如cpu子系统就是控制cpu时间分配的一个控制器。子系统必须附加（attach）到一个层级上才能起作用，一个子系统附加到某个层级以后，这个层级上的所有控制族群都受到这个子系统的控制。

**子系统介绍**

**blkio**--这个子系统为块设备设定输入/输出限制，比如物理设备（磁盘，固态硬盘，USB）。

**cpu**--这个子系统使用调度程序提供对CPU的cgroup任务访问。

**cpuacct**--这个子系统自动生成 cgroup 中任务所使用的CPU报告。

**cpuset**--这个子系统为cgroup中的任务分配独立CPU（在多核系统）和内存节点。

**devices**--这个子系统可允许或者拒绝cgroup中的任务访问设备。

**freezer**--这个子系统挂起或者恢复cgroup中的任务。

**memory**--这个子系统设定cgroup中任务使用的内存限制，并自动生成由那些任务使用的内存资源报告。

**net\_cls**--这个子系统使用等级识别符（classid）标记网络数据包，可允许Linux流量控制程序（tc）识别从具体cgroup中生成的数据包。

**cGroup的用法**

cgroup操作的步骤大致如下：

1. 挂载cgroup根文件系统，类型为tmpfs。

mount -t tmpfs cgroup\_root /sys/fs/cgroup

1. 在cgroupfs根目录下创建子cgroup，名为cpuset。

mkdir /sys/fs/cgroup/cpuset

1. 将名为cpuset的cgroup关联到cpuset子系统。

mount -t cgroup -o cpuset cpuset /sys/fs/cgroup/cpuset

1. 在cpuset目录下创建目录，生成一个子cgroup，属性文件中写入相应内容，设置属性。
2. 启动需要限制的进程，查找其对应的进程ID，将其写入对应的task文件中。

例：

mount -t tmpfs cgroup\_root /sys/fs/cgroup

mkdir /sys/fs/cgroup/cpuset

mount -t cgroup cpuset -o cpuset /sys/fs/cgroup/cpuset

cd /sys/fs/cgroup/cpuset

mkdir Charlie

cd Charlie

echo 2-3 > cpuset.cpus

echo 1 > cpuset.mems

echo $$ > tasks