

测试机器：CentOS release 6.4 (Final)

启动Cgroups

```
service cgconfig start    #开启cgroups服务
chkconfig cgconfig on     #开启启动
```

在/cgroup，有如下文件夹，默认是多挂载点的形式，即各个子系统的配置在不同的子文件夹下

```
[root@localhost /]# ls /cgroup/
blkio  cpu  cpuacct  cpuset  devices  freezer  memory  net_cls
```

cgroups管理进程cpu资源

跑一个耗cpu的脚本

```
x=0
while [ True ];do
    x=$((x+1))
done;
```

top可以看到这个脚本基本占了100%的cpu资源

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
30142	root	20	0	104m	2520	1024	R	99.7	0.1	14:38.97	sh

下面用cgroups控制这个进程的cpu资源

```
mkdir -p /cgroup/cpu/foo/    #新建一个控制组foo
echo 50000 > /cgroup/cpu/foo/cpu.cfs_quota_us  #将cpu.cfs_quota_us设为50000，相对于
cpu.cfs_period_us的100000是50%
echo 30142 > /cgroup/cpu/foo/tasks
```

然后top的实时统计数据如下，cpu占用率将近50%，看来cgroups关于cpu的控制起了效果

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
30142	root	20	0	105m	2884	1024	R	49.4	0.2	23:32.53	sh

cpu控制组foo下面还有其他的控制，还可以做更多其他的关于cpu的控制

```
[root@localhost ~]# ls /cgroup/cpu/foo/
cgroup.event_control  cgroup.procs  cpu.cfs_period_us  cpu.cfs_quota_us  cpu.rt_period_us
cpu.rt_runtime_us  cpu.shares  cpu.stat  notify_on_release  tasks
```

cgroups管理进程内存资源

跑一个耗内存的脚本，内存不断增长

```
x="a"
while [ True ];do
    x=$x$x
done;
```

top看内存占用稳步上升

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
30215	root	20	0	871m	501m	1036	R	99.8	26.7	0:38.69	sh
30215	root	20	0	1639m	721m	1036	R	98.7	38.4	1:03.99	sh
30215	root	20	0	1639m	929m	1036	R	98.6	49.5	1:13.73	sh

下面用cgroups控制这个进程的内存资源

```
mkdir -p /cgroup/memory/foo
echo 1048576 > /cgroup/memory/foo/memory.limit_in_bytes #分配1MB的内存给这个控制组
echo 30215 > /cgroup/memory/foo/tasks
```

发现之前的脚本被kill掉

```
[root@localhost ~]# sh /home/test.sh
已杀死
```

因为这是强硬的限制内存，当进程试图占用的内存超过了cgroups的限制，会触发out of memory，导致进程被kill掉。

实际情况中对进程的内存使用会有一个预估，然后会给这个进程的限制超配50%比如，除非发生内存泄露等异常情况，才会因为cgroups的限制被kill掉。

也可以通过配置关掉cgroups oom kill进程，通过memory.oom_control来实现（oom_kill_disable 1），但是尽管进程不会被直接杀死，但进程也进入了休眠状态，无法继续执行，仍让无法服务。

关于内存的控制，还有以下配置文件，关于虚拟内存的控制，以及权值比重式的内存控制等

```
[root@localhost ~]# ls /cgroup/memory/foo/
cgroup.event_control  memory.force_empty      memory.memsw.failcnt
memory.memsw.usage_in_bytes  memory.soft_limit_in_bytes  memory.usage_in_bytes  tasks
cgroup.procs          memory.limit_in_bytes      memory.memsw.limit_in_bytes
memory.move_charge_at_immigrate  memory.stat                memory.use_hierarchy
memory.failcnt         memory.max_usage_in_bytes  memory.memsw.max_usage_in_bytes
memory.oom_control      memory.swappiness          notify_on_release
```

cgroups管理进程io资源

跑一个耗io的脚本

```
dd if=/dev/sda of=/dev/null &
```

通过iotop看io占用情况，磁盘速度到了284M/s

```
30252 be/4 root      284.71 M/s    0.00 B/s   0.00 %   0.00 % dd if=/dev/sda of=/dev/null
```

下面用cgroups控制这个进程的io资源

```
mkdir -p /cgroup/blkio/foo

echo '8:0 1048576' > /cgroup/blkio/foo/blkio.throttle.read_bps_device
#8:0对应主设备号和副设备号，可以通过ls -l /dev/sda查看
echo 30252 > /cgroup/blkio/foo/tasks
```

再通过iotop看，确实将读速度降到了1M/s

```
30252 be/4 root      993.36 K/s    0.00 B/s  0.00 %  0.00 % dd if=/dev/sda of=/dev/null
```

对于io还有很多其他可以控制层面和方式，如下

```
[root@localhost ~]# ls /cgroup/blkio/foo/
blkio.io_merged      blkio.io_serviced    blkio.reset_stats
blkio.throttle.io_serviced  blkio.throttle.write_bps_device  blkio.weight
cgroup.procs
blkio.io_queued      blkio.io_service_time  blkio.sectors
blkio.throttle.read_bps_device  blkio.throttle.write_iops_device  blkio.weight_device
notify_on_release
blkio.io_service_bytes  blkio.io_wait_time      blkio.throttle.io_service_bytes
blkio.throttle.read_iops_device  blkio.time                cgroup.event_control  tasks
```