Hewlett Packard Enterprise

HPE University ——Docker专业课程

Hewlett Packard Enterprise

Docker培训2

理解Docker的原理及组成

- 容器的核心原理
- Docker的架构与组成
- Docker的网络原理

课堂练习

cGroups限制进程的CPU占用

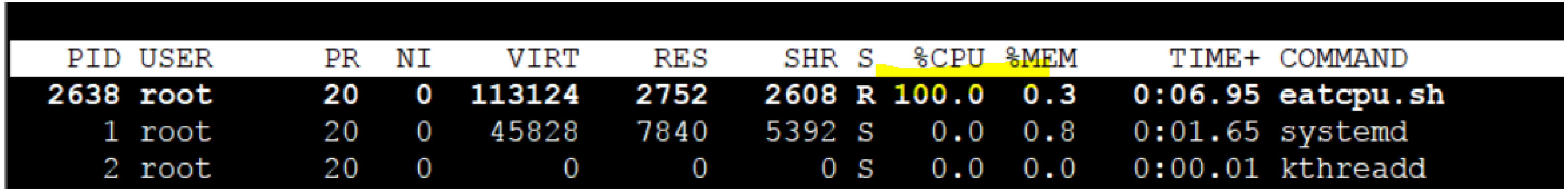
先模拟一个“吃”CPU的脚本程序eatcpu.sh

```
 root@docker_node ~]# vi eatcpu.sh
chmod +x eatcpu.sh
./eatcpu.sh
```

```
#! /bin/bash
i=0;
while true
do
i=i+1;
done
```

新开窗口执行top命令观察

```
root@docker_node ~]# top
```



```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 2638 root      20   0  113124   2752   2608 R 100.0  0.3   0:06.95 eatcpu.sh
    1 root      20   0   45828   7840   5392 S   0.0  0.8   0:01.65 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.01 kthreadd
```

创建一个限制CPU的cGroups——mylimit

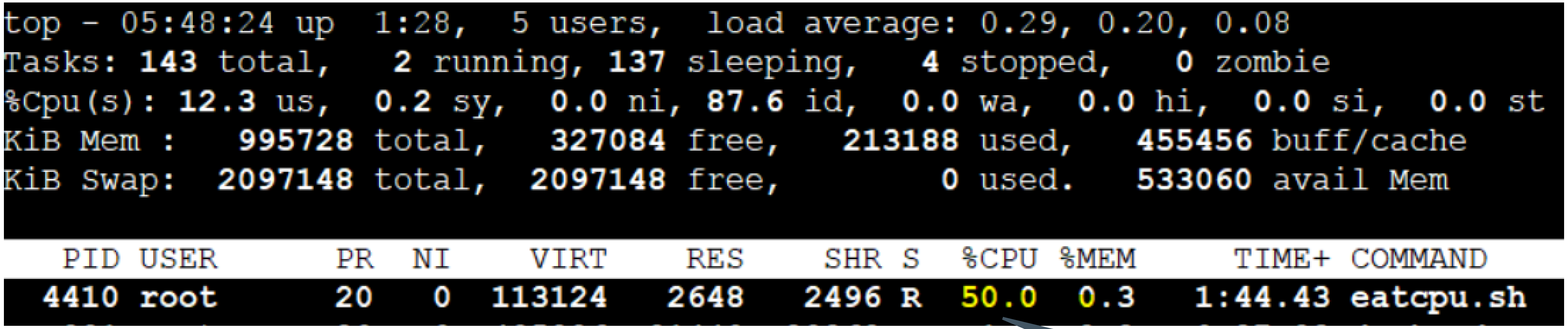
```
# cgcreate -g cpu:/mylimit
# ls /sys/fs/cgroup/cpu/mylimit/
cgroup.clone_children  cpuacct.usage         cpuacct.usage_percpu_sys
cpuacct.usage_user  cpu.rt_period_us   cpu.stat
cgroup.procs          cpuacct.usage_all     cpuacct.usage_percpu_user
cpu.cfs_period_us   cpu.rt_runtime_us  notify_on_release
cpuacct.stat          cpuacct.usage_percpu  cpuacct.usage_sys         cpu.cfs_quota_us
cpu.shares        tasks
# cat /sys/fs/cgroup/cpu/mylimit/cpu.cfs_period_us
100000
# cat /sys/fs/cgroup/cpu/mylimit/cpu.cfs_quota_us
-1
# echo 50000 > /sys/fs/cgroup/cpu/mylimit/cpu.cfs_quota_us
```

-1表示没限制，设置为50000，则为50000/100000=50%

见证奇迹的时刻

将消耗CPU的进程ID写入控制组

```
echo 4410 >/sys/fs/cgroup/cpu/mylimit/tasks
```



```
top - 05:48:24 up  1:28,  5 users,  load average: 0.29, 0.20, 0.08
Tasks: 143 total,   2 running, 137 sleeping,   4 stopped,   0 zombie
%Cpu(s): 12.3 us,  0.2 sy,  0.0 ni, 87.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   995728 total,   327084 free,   213188 used,   455456 buff/cache
KiB Swap:  2097148 total,  2097148 free,        0 used.   533060 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 4410 root      20   0  113124   2648   2496 R  50.0  0.3   1:44.43 eatcpu.sh
```

CPU使用率只占到50%的核心

继续实践

1. 启动两个脚本进程，将两个脚本进程的ID都写入同一个cgroup的tasks里会是什么结果？

2. 容器的CPU使用被限制为100m（ millicores千分之一）=100/1000=0.1核心

3. 限制容器只能使用128Mi内存

查看启动容器的相关Docker进程

系统中有2个运行中的容器时，可以看到有2个containerd-shim-runc-v2进程

```
# ps -ef | grep docker
root       938     1  0 10:49 ?        00:00:18 /usr/bin/dockerd -H fd://
root      3080  3026  0 16:14 pts/3    00:00:00 docker run -it busybox sh
root      3187  1869  0 16:18 pts/1    00:00:00 docker run -it busybox

# ps aux | grep shim
root      3103  0.0  0.3 111952 12004 ?        Sl   16:14   0:00 /usr/bin/containerd-shim-runc-v2 -namespace moby -id
f01cd084e721cea1291b3b12477462363c152a0c30b5128fb1014dc573a94c8f -address
/run/containerd/containerd.sock
root      3211  0.0  0.2 113232 10216 ?        Sl   16:18   0:00 /usr/bin/containerd-shim-runc-v2 -namespace moby -id
d6c9c3c52dc8b2908ab9ea0efd908efbce295907943e3ab025b4cf18eb8d3e5b -address
/run/containerd/containerd.sock
```

None网络模式

容器有自己的Network Namespace，但没有eth0网卡，这种模式可以让用户（程序）手动生成容器的网卡并实现固定IP这样的特殊功能

```
 root@localhost ~]# docker run --net=none -it busybox sh
/ # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
```

Hewlett Packard
Enterprise

Hewlett Packard Enterprise

HPE University

地址：北京市 朝阳区 广顺南大街8号院 1号楼 利星行中心A座

邮箱：hpeu@hpe.com