Linux进程、线程和调度(5)

麦当劳喜欢您来，喜欢您再来

第五次课大纲

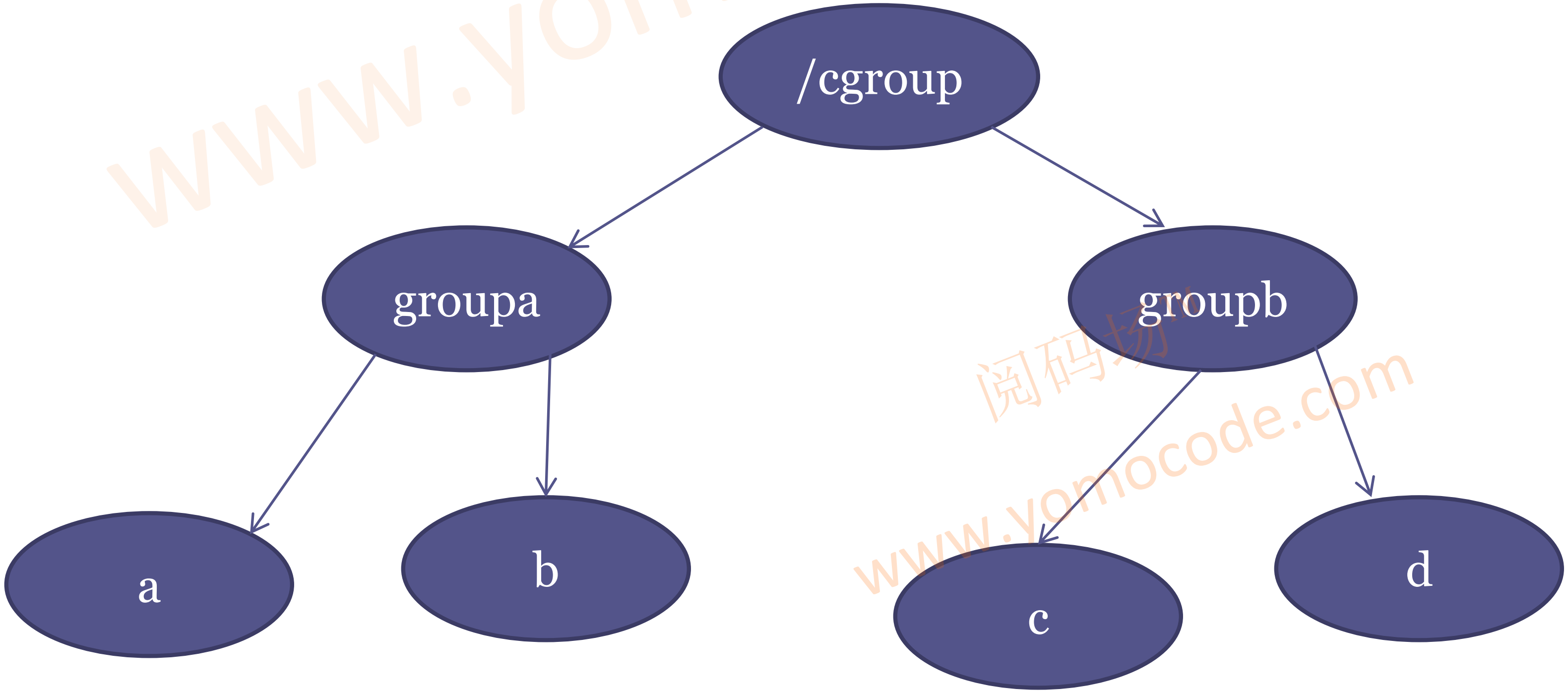
- 基于cpu cgroups进行CPU资源分配
- Linux的sched_autogroup（全新内容）
- 基于cpuset cgroups进行进程CPU绑定（全新内容）
- Docker和cgroups
- Systemd和cgroups（全新内容）
- Android对cgroups的利用

练习题

- 开启和关闭sched_autogroup，观察CPU利用率情况；
- 创建和分群CPU的cgroups，调整权重和quota；
- 利用cpuset cgroups，进行进程的CPU绑定

cgroup

- 定义不同cgroup CPU分享的share
- 定义某个cgroup在某个周期里面最多跑多久



Cgroups之前的权重

```
/sys/fs/cgroup/cpu/A$ cat cpu.shares
1024

/sys/fs/cgroup/cpu/B$ cat cpu.shares
2048
```

Cgroups的资源控制

```
/sys/fs/cgroup/cpu/A

        cpu.cfs_period_us
        cpu.cfs_quota_us
```

sched_autogroup: 200行的wonder patch

Yeah. And I have to say that I'm (very happily) surprised by just how small that patch really ends up being, and how it's not intrusive or ugly either.

I'm also very happy with just what it does to interactive performance. Admittedly, my "testcase" is really trivial (reading email in a web-browser, scrolling around a bit, while doing a "make -j64" on the kernel at the same time), but it's a test-case that is very relevant for me. And it is a _huge_ improvement.

It's an improvement for things like smooth scrolling around, but what I found more interesting was how it seems to really make web pages load a lot faster. Maybe it shouldn't have been surprising, but I always associated that with network performance. But there's clearly enough of a CPU load when loading a new web page that if you have a load average of 50+ at the same time, you _will_ be starved for CPU in the loading process, and probably won't get all the http requests out quickly enough.

So I think this is firmly one of those "real improvement" patches. Good job. Group scheduling goes from "useful for some specific server loads" to "that's a killer feature".

Linus

sched_autogroup: per-session cgroups

```
$ ps -C a.out o pid,ppid,pgid,sid,comm
  PID  PPID  PGID   SID COMMAND
 3832  3778  3832  3778 a.out
 3852  3835  3852  3835 a.out

$ cat  /proc/3832/autogroup
/autogroup-485 nice 0
$ cat  /proc/3852/autogroup
/autogroup-487 nice 0

$ cat  /proc/3778/autogroup
/autogroup-485 nice 0
$ cat  /proc/3835/autogroup
/autogroup-487 nice 0
```

Android和cgroup

- apps, bg_non_interactive

Shares:

apps: cpu.shares = 1024

bg_non_interactive: cpu.shares = 52

Quota:

apps:

cpu.rt_period_us: 1000000 cpu.rt_runtime_us: 800000

bg_non_interactive:

cpu.rt_period_us: 1000000 cpu.rt_runtime_us: 700000

Docker和cgroup

- Docker使用cgroup调配容器的CPU资源

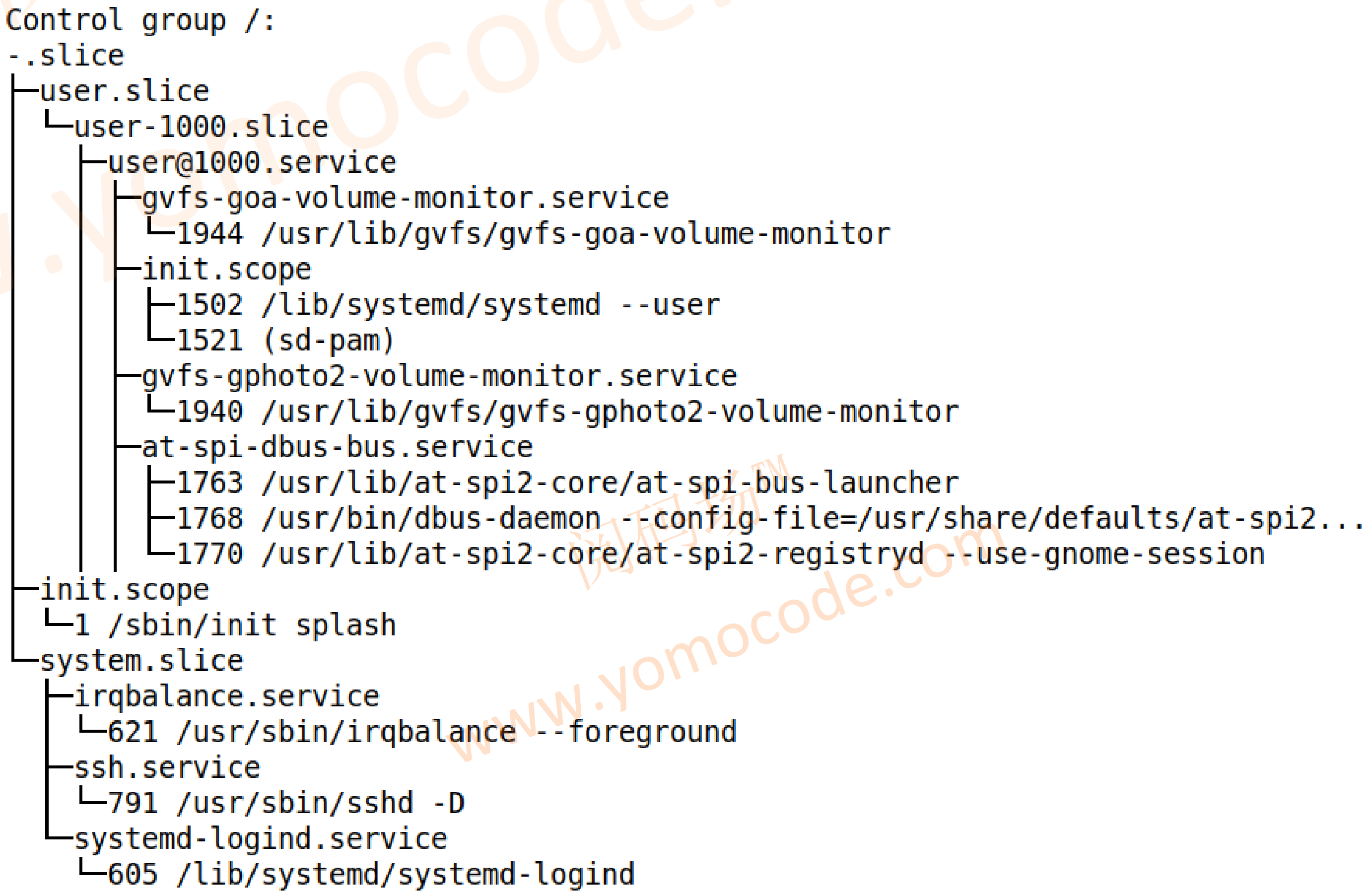
```
$ docker run --cpu-quota 25000 --cpu-period 10000 --cpu-shares 30
linuxep/lepv0.1

baohua@ubuntu:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS              PORTS               NAMES
3f39ca25d14d

baohua@ubuntu:/sys/fs/cgroup/cpu/docker$ cd 3f39c...
baohua@ubuntu:/sys/fs/cgroup/cpu/docker/3f39c...$ ls
cgroup.clone_children  cgroup.procs  cpuacct.stat  cpuacct.usage
cpuacct.usage_percpu  cpu.cfs_period_us  cpu.cfs_quota_us  cpu.shares  cpu.stat
notify_on_release  tasks
baohua@ubuntu:/sys/fs/cgroup/cpu/docker/3f39c...$ cat cpu.cfs_quota_us
25000
baohua@ubuntu:/sys/fs/cgroup/cpu/docker/3f39c...$ cat cpu.cfs_period_us
10000
baohua@ubuntu:/sys/fs/cgroup/cpu/docker/3f39c...$ cat cpu.shares
30
```

systemd cg层级

- slice
- scope
- service



```
Control group /:
-.slice
├─user.slice
│ └─user-1000.slice
│   ├─user@1000.service
│   │ ├─gvfs-goa-volume-monitor.service
│   │ │ └─1944 /usr/lib/gvfs/gvfs-goa-volume-monitor
│   │ ├─init.scope
│   │ │ ├─1502 /lib/systemd/systemd --user
│   │ │ └─1521 (sd-pam)
│   │ ├─gvfs-gphoto2-volume-monitor.service
│   │ │ └─1940 /usr/lib/gvfs/gvfs-gphoto2-volume-monitor
│   │ ├─at-spi-dbus-bus.service
│   │ │ ├─1763 /usr/lib/at-spi2-core/at-spi-bus-launcher
│   │ │ ├─1768 /usr/bin/dbus-daemon --config-file=/usr/share/defaults/at-spi2...
│   │ │ └─1770 /usr/lib/at-spi2-core/at-spi2-registryd --use-gnome-session
├─init.scope
│ └─1 /sbin/init splash
└─system.slice
  ├─irqbalance.service
  │ └─621 /usr/sbin/irqbalance --foreground
  ├─ssh.service
  │ └─791 /usr/sbin/sshd -D
  └─systemd-logind.service
    └─605 /lib/systemd/systemd-logind
```

systemd cg相关命令

- systemd-cgls
- systemd-cgtop
- systemd-run
- systemctl

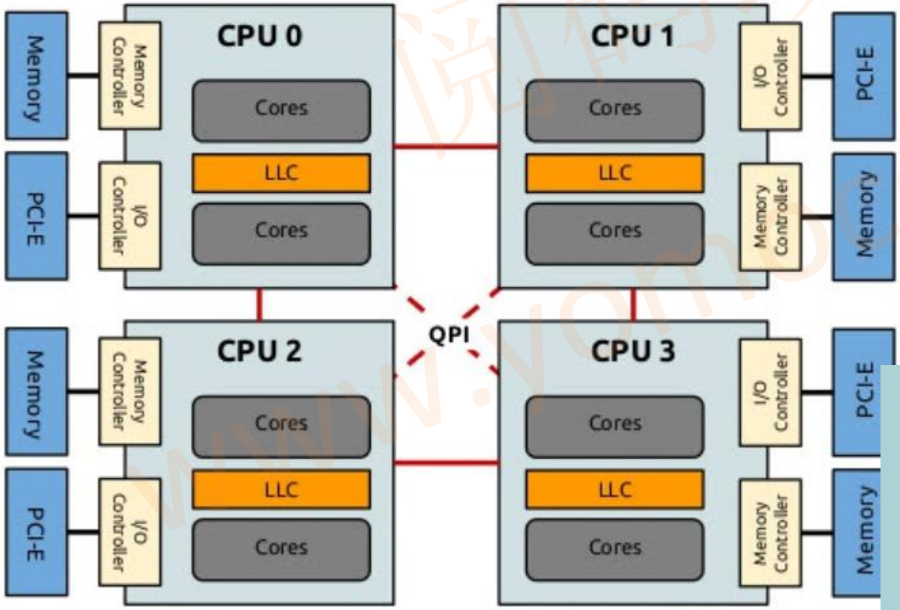
    set-property 限制cgroup的资源

cpuset

- Cpusets provide a Linux kernel mechanism to constrain which CPUs and Memory Nodes are used by a process or set of processes.
- The root cpuset contains all the systems CPUs and Memory Nodes.
- cpuset.cpus: list of CPUs in that cpuset
- cpuset.mems: list of Memory Nodes in that cpuset

NUMA

CPU architecture (Intel Sandy Bridge)



```
# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 12 13 14 15 16 17
node 0 size: 130677 MB
node 0 free: 1453 MB
node 1 cpus: 6 7 8 9 10 11 18 19 20 21 22 23
node 1 size: 131056 MB
node 1 free: 614 MB
node distances:
node   0   1
  0:  10  21
  1:  21  10
```

谢谢！

阅码场
YOMOCODE

阅码场出品

www.yomocode.com