本篇内容灰常少，都是和虚拟机线程相关，所以我给本篇分享起了个 title，叫做：

How libvirt tunes the domain threads

本篇只炒了 4 个栗子：(均基于 upstream 的 libvirt git head)

* vcpu threads 到 host cpu 的绑定方式
* emulator threads 到 host cpu 的绑定方式
* iothreads 到 host cpu 的绑定方式
* 各种 threads 的权重设置方式

先快速科普下大背景，libvirt 针对虚拟机的种种调优，绝大多数都是通过 cgroup 实现的。通常 libvirt 不会直接和 cgroup 说话，而是利用 systemd 与 cgroup 相配合这个特性，来'间接地'做出设置。

在仅启用 libvirt service，而没有任何虚拟机运行时，如下图，我们可以看到，libvirt 为用户创建了用于组控制的 machine.slice 和 system.slice 等 unit

```
root:~# systemctl -t slice
UNIT                                        LOAD   ACTIVE SUB    DESCRIPTION
-.slice                                     loaded active active Root Slice
machine.slice                               loaded active active Virtual Machine and Container Slice
system-getty.slice                          loaded active active system-getty.slice
system-systemd\x2dbacklight.slice           loaded active active system-systemd\x2dbacklight.slice
system-systemd\x2dhibernate\x2dresume.slice loaded active active system-systemd\x2dhibernate\x2dresume.slice
system.slice                                loaded active active System Slice
user-1000.slice                             loaded active active User Slice of suse
user.slice                                  loaded active active User and Session Slice

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

8 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
root:~#
```

没有任何虚拟运行时，对应的 scope unit 是不存在的

```
root:/sys/fs/cgroup# systemctl -t scope
UNIT            LOAD   ACTIVE SUB     DESCRIPTION
init.scope      loaded active running System and Service Manager
session-2.scope loaded active running Session 2 of user suse

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

2 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
root:/sys/fs/cgroup#
```

当用户启动某个虚拟机后，或虚拟机处于非’ shut off’状态时，我们就会看到对应的 scope unit

```
root:~# systemctl -t scope
UNIT                                 LOAD   ACTIVE SUB     DESCRIPTION
init.scope                           loaded active running System and Service Manager
machine-qemu\x2d1\x2dsles12sp3.scope loaded active running Virtual Machine qemu-1-sles12sp3
session-2.scope                      loaded active running Session 2 of user suse

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

3 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
root:~#
```

至于什么是 slice，什么是 scope，systemd 又是如何与 cgroup 一起玩耍的，不属于本篇范围，请大家自行脑补。
此外，关于虚拟机其他组件的调控手段，比如虚拟磁盘、虚拟网卡等的资源控制(限制、流控等)，请大家自己 man 吧,我以后也不打算写它们了，忒墨迹。

例 1 vcpu threads 到 host cpu 的绑定方式

下图中 host 有四个处理器可用， 虚拟机有两个 vcpu，现绑定虚拟机 sles12sp3 的 vcpu1 到 host 处理器 2 和处理器 3。如此， 代表 sles12sp3 vcpu1 的线程就不会被调度到其他 host 处理器，但针对 vcpu0 的线程未做设置，意味着可能被调度到任意 host 处理器去 run

```
root:~# virsh nodecpumap
CPUs present:   4
CPUs online:    4
CPU map:        yyyy

root:~#
root:~# virsh vcpuinfo --domain sles12sp3
VCPU:           0
CPU:            N/A
State:          N/A
CPU time        N/A
CPU Affinity:   yyyy

VCPU:           1
CPU:            N/A
State:          N/A
CPU time        N/A
CPU Affinity:   yyyy

root:~#
root:~# virsh vcpupin --domain sles12sp3 1 2,3 --config

root:~# virsh vcpuinfo --domain sles12sp3
VCPU:           0
CPU:            N/A
State:          N/A
CPU time        N/A
CPU Affinity:   yyyy

VCPU:           1
CPU:            N/A
State:          N/A
CPU time        N/A
CPU Affinity:   --yy

root:~#
root:~# virsh start sles12sp3
Domain sles12sp3 started
```

让我们可以换个角度，通过 libvirtd 日志看看它都做了哪些动作以实现上述效果：

```
root:~# cat /var/run/libvirt/qemu/sles12sp3.pid && echo
8989
```

```
root:~# virsh qemu-monitor-command sles12sp3 --hmp info cpus
* CPU #0: thread_id=9005
  CPU #1: thread_id=9006


root:~#
```

```
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupNew:1158 : pid=8989 path= parent=(nil) contro
llers=-1 group=0x7fe01c962638
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:648 : group=0x7fe01401b5d0 controllers
=-1 path= parent=(nil)
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:699 : Auto-detecting controllers
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'cpu' present=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'cpuacct' present=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'cpuset' present=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'memory' present=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'devices' present=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'freezer' present=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'blkio' present=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'net_cls' present=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'perf_event' present=
yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:704 : Controller 'name=systemd' presen
t=yes
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetectPlacement:553 : Detecting placement for
 pid 8989 path
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:747 : Detected mount/mapping 0:cpu at
/sys/fs/cgroup/cpu,cpuacct in /machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope for pid 8989
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:747 : Detected mount/mapping 1:cpuacct
 at /sys/fs/cgroup/cpu,cpuacct in /machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope for pid 8989
2018-05-13 03:07:10.421+0000: 8532: debug : virCgroupDetect:747 : Detected mount/mapping 2:cpuset
```

```
2018-05-13 03:07:10.520+0000: 8532: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/c
puset/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu1/cpuset.cpus' to '2-3'
2018-05-13 03:07:10.520+0000: 8532: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/c
pu,cpuacct/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu1/tasks' to '9006'
2018-05-13 03:07:10.520+0000: 8532: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/c
pu,cpuacct/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu1/tasks' to '9006'
2018-05-13 03:07:10.520+0000: 8532: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/c
puset/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu1/tasks' to '9006'
```

例 2 emulator threads 到 host cpu 的绑定方式

在 libvirt 眼中啊，除了 vcpu threads 和 iothreads 以外，其他的都属于 emulator threads :-)
比如主线程，事件监视线程，spice 或 vnc 等等等。让我们来看看目前这个虚拟机有哪些线程：

```
root:/proc/5355/task# for lwp in *; do echo -n "$lwp: " && cat $lwp/cpuset; done
5355: /machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator
5402: /machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator
5421: /machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator
5422: /machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu0
5423: /machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu1
5443: /machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator
root:/proc/5355/task#
```

默认时，emulator threads 到 host cpu 的亲和力设置是：

```
root:/# virsh emulatorpin --domain sles12sp3
emulator: CPU Affinity
----------------------------------
       *: 0-3

root:/#
```

即全部。

现在让我们把 emulator threads 绑定到 host cpu1 和 cpu2 上面

```
root:/proc/5355/task# virsh emulatorpin sles12sp3 1,2

root:/proc/5355/task# virsh emulatorpin sles12sp3
emulator: CPU Affinity
----------------------------------
       *: 1-2

root:/proc/5355/task#
```

再 check 一下 libvirtd 日志和虚拟机运行时配置，于是它干了些啥就一目了然了：

```
2018-05-13 06:55:43.287+0000: 3641: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpuset/machine.slice/m
achine-qemu\x2d1\x2dsles12sp3.scope/emulator/cpuset.cpus' to '1-2'
```

```
<domain type='kvm' id='1'>
  <name>sles12sp3</name>
  <uuid>0623d8ff-3ef3-482d-9a44-04548d722fc3</uuid>
  <memory unit='KiB'>2097152</memory>
  <currentMemory unit='KiB'>1048576</currentMemory>
  <vcpu placement='static' current='2'>4</vcpu>
  <cputune>
    <vcpupin vcpu='1' cpuset='2-3'/>
    <emulatorpin cpuset='1-2'/>
  </cputune>
  <resource>
    <partition>/machine</partition>
  </resource>
  <os>
```

```
root:/sys/fs/cgroup/cpuset/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator# cat cpuset.cpus
1-2
root:/sys/fs/cgroup/cpuset/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator#
root:/sys/fs/cgroup/cpuset/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator# cat tasks
5355
5402
5418
5421
5443
root:/sys/fs/cgroup/cpuset/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator#
root:/sys/fs/cgroup/cpuset/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator# ps aux | grep 5418 | grep -v grep
root      5418  0.0  0.0      0     0 ?        S    13:54   0:00 [vhost-5355]
root:/sys/fs/cgroup/cpuset/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/emulator#
```

注意到上面这个'奇怪'的线程 5418 了吗？它怎么也跑到 emulator 下面去了？有兴趣研究 vhost 的同学可以看看，就会找到答案的.

例 3 iothreads 到 host cpu 的绑定方式 (available since qemu 2.0.0 and libvirt 1.2.8)

首先，看下我的这个正在运行着的虚拟机，目前是没有任何的 iothreads.

```
root:/sys/fs/cgroup# virsh qemu-monitor-command sles12sp3 --hmp info iothreads


root:/sys/fs/cgroup#
```

现在让我们加 2 个 iothreads:

```
# virsh iothreadadd sles12sp3 --id 4
# virsh iothreadadd sles12sp3 --id 6
```

```
2018-05-13 08:07:52.761+0000: 3641: debug : virCgroupMakeGroup:1126 : Done making controllers for group
2018-05-13 08:07:52.761+0000: 3641: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpu,cpuacct/machine.sl
ice/machine-qemu\x2d6\x2dsles12sp3.scope/iothread6/tasks' to '26044'
2018-05-13 08:07:52.761+0000: 3641: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpu,cpuacct/machine.sl
ice/machine-qemu\x2d6\x2dsles12sp3.scope/iothread6/tasks' to '26044'
2018-05-13 08:07:52.761+0000: 3641: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpuset/machine.slice/m
achine-qemu\x2d6\x2dsles12sp3.scope/iothread6/tasks' to '26044'
```

```
root:/sys/fs/cgroup# virsh qemu-monitor-command sles12sp3 --hmp info iothreads
iothread4:
  thread_id=26013
  poll-max-ns=32768
  poll-grow=0
  poll-shrink=0
iothread6:
  thread_id=26044
  poll-max-ns=32768
  poll-grow=0
  poll-shrink=0


root:/sys/fs/cgroup#
```

默认情况 iothreads 到 host cpu 的亲和力是酱婶儿的:

```
root:/# virsh iothreadinfo sles12sp3
 IOThread ID     CPU Affinity
---------------------------------------------------
 4               0-3
 6               0-3

root:/#
```

现在我们可以把其中某个 iothread 与新添加的虚拟机块设备绑定，如此就可将设备的模拟放入单独的线程中来 run.

```
# virsh attach-disk sles12sp3 /opt/vms/sles12sp3/disk1.raw vdb \
--driver qemu --subdriver raw --targetbus virtio --iothread 4
```

接下来，我们可以将其与某些 host cpu 的关系变的很好:

```
root:/# virsh iothreadpin sles12sp3 4 1,2

root:/#
root:/# virsh iothreadinfo sles12sp3
 IOThread ID     CPU Affinity
---------------------------------------------------
 4               1-2
 6               0-3

root:/#
```

老规矩，check 一下 libvirtd 日志:

```
2018-05-13 08:15:38.360+0000: 3642: debug : virCgroupMakeGroup:1126 : Done making controllers for group
2018-05-13 08:15:38.360+0000: 3642: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpuset/machine.slice/m
achine-qemu\x2d6\x2dsles12sp3.scope/iothread4/cpuset.cpus' to '1-2'
```

See? :-)

例 4 各种 threads 的权重设置方式

权重控制的参数包括：
cpu_shares：对应 cgroup 中的 cpu.shares，是使用 host cpus 的权重时间比，无固定数值.这是一个相对设置,和其他 guest 相比较,谁的数值大,谁可以使用的主机 cpu 资源就多,如设置 2048 值的 guest,就比设 1024 值的 guest 可多使用 2 倍的 cpu,即它们使用 cpu 的理论比值就是 2 ： 1

*_period： 强制间隔的时间周期,单位微秒,范围[1000,1000000]， 比如可以通过 vcpu_period 设置 vcpu 不能使用超过 period 时间周期.

*_quota：允许带宽,即在此 period 内可使用的 cpu 时间,单位微秒,范围[1000,18446744073709551]，负值代表无限制

下面以调整 vcpu 权重为例，设置该虚拟机的 vcpus 只能使用 20%的主机 CPU 资源：
(即 20000/100000 = 0.2)

```
root:~# virsh schedinfo --domain sles12sp3 --set vcpu_period=100000 --set vcpu_quota=20000
Scheduler      : posix
cpu_shares     : 1024
vcpu_period    : 100000
vcpu_quota     : 20000
emulator_period: 100000
emulator_quota : -1
global_period  : 100000
global_quota   : -1
iothread_period: 100000
iothread_quota : -1

root:~#
```

check 一下 libvirtd 日志：

```
2018-05-13 10:01:00.624+0000: 6165: debug : virCgroupGetValueStr:832 : Get value /sys/fs/cgroup/cpu,cpuac
ct/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu0/cpu.cfs_period_us
2018-05-13 10:01:00.624+0000: 6165: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpu,cpua
cct/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu0/cpu.cfs_period_us' to '100000'
......
2018-05-13 10:01:00.625+0000: 6165: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpu,cpua
cct/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu1/cpu.cfs_period_us' to '100000'
......
2018-05-13 10:01:00.625+0000: 6165: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpu,cpua
cct/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu0/cpu.cfs_quota_us' to '20000'
......
2018-05-13 10:01:00.625+0000: 6165: debug : virCgroupSetValueStr:796 : Set value '/sys/fs/cgroup/cpu,cpua
cct/machine.slice/machine-qemu\x2d1\x2dsles12sp3.scope/vcpu1/cpu.cfs_quota_us' to '20000'
2018-05-13 10:01:00.625+0000: 6166: debug : virCgroupGetValueStr:832 : Get value /sys/fs/cgroup/cpu,cpuac
```

还有这么大空地儿，别浪费了，解释一下日志中 类似'machine.slice/machine-qemu\x2d3\x2dsles12sp3...'的意思吧.
Libvirt 所使用 scope 名字是直接映射到了 cgroup 的目录名,并要求转义任何 systemd 的保留字符，于是上述字符串被解释为：
'\x2d'是以\x 开头,是十六进制数，也就是 ASCII 值为 46 的减号'-'
machine-qemu\x2d3\x2dsles12sp3...'是按照此格式得出的:machine-${drivertype}-${DOMID}-${DOMNAME}
其中，drivetype 是 qemu，domain id 是 3，domain name 是 sles12sp3
最终就是'machine-qemu-3-sles12sp3...'