关于cgroups

一个cgroup是一组processes(tasks)的集合,可以绑定他们一起来运用一系列的约束来控制对系统资源的访问.你可以创建一个cgroups的层次接口,在那里孩子cgroups将会继承父cgroups的属性,你可以使用cgroups来通过下面的方式来管理进程以及tasks.

1. 限制group里面可以使用的CPU, I/O以及内存资源的量
2. 改变想对于其他的groups调整优先级
3. 测量group使用的资源,可以用于计费的目的
4. 隔离文件系统中跟他们文件系统的进程和网络
5. 冰冻一个小时,包含了很多文件以及前端的

你可以通过以下的方式来创建或者管理:

1. 编辑/etc/cgconfig.conf配置文件
2. 使用cgcreate cgclassify cgexec命令操作
3. 操作cgroup的虚拟文件系统,比如在/sys/fs/cgroup目录下添加进程ID到tasks目录下
4. /etc/cgrules.conf配置文件在添加规则
5. 使用其他的应用软件
6. 使用libvirt提供的APi

子系统

### blkio Parameters

|  |
| --- |
| 报告参数blkio.io\_merged 报告 cgroup 将 BIOS 请求合并到 I/O 操作请求的次数。条目有两个字段：number 和 operation。 blkio.io\_queued 报告 cgroup 为 I/O 操作排队的请求次数。条目有两个字段：number 和 operation。 blkio.io\_service\_bytes 报告 cgroup 对具体设备的读写字节数。条目有四个字段：major, minor, operation 和 bytes。其中 operation 表示操作类型，包括 **read, write, sync 和 async**。 blkio.io\_serviced 报告 cgroup 对具体设备的 I/O 操作数。条目有四个字段：major, minor, operation 和 number。 blkio.io\_service\_time 报告 cgroup 对具体设备的 I/O 操作请求发送和请求完成之间的时间。条目有四个字段：major, minor, operation 和 time，其中 time 的单位为**纳秒(ns)**。 blkio.io\_wait\_time 报告 cgroup 对具体设备的 I/O 操作在队列中等待的时间。条目有四个字段：major, minor, operation 和 time，其中 time 的单位为**纳秒(ns)**。 blkio.reset\_stats 向该文件中写入一个整数，可以重置该 cgroup 中的报告计数。 blkio.sectors 报告 cgroup 对具体设备的扇区读写数。条目有三个字段：major, minor 和 sectors。 blkio.throttle.io\_service\_bytes 报告 cgroup 限流对具体设备的读写字节数。**blkio**.**io**\_service\_bytes 与 **blkio**.throttle.**io**\_service\_bytes 的不同之处在于，CFQ 调度请求队列时，前者不会更新。条目有四个字段：major, minor, operation 和 bytes。 blkio.throttle.io\_serviced 报告 cgroup 限流对具体设备的读写操作数。条目有四个字段：major, minor, operation 和 number。 blkio.time 报告 cgroup 对具体设备的 I/O 访问时间。条目有三个字段：major, minor 和 time。time 的单位为**毫秒(ms)**  可设置参数 blkio.throttle.read\_bps\_device Specifies the maximum number of bytes per second that a cgroup may read from a device specified by its major and minor numbers. For example, the setting 8:1 4194304 specifies that a maximum of 4 MB per second may be read from /dev/sda1. blkio.throttle.write\_bps\_device Specifies the maximum number of bytes per second that a cgroup may write to a device specified by its major and minor numbers. For example, the setting 8:2 2097152 specifies a maximum of 2 MB per second may be written to /dev/sda2.  说明：指定 cgroup 中某设备每秒钟读/写数据的字节上限。其格式为 major:minor bytes\_per\_second。 示例：设置 cgroup lv0 在 /dev/sdc 上每秒最多读 10KiB 数据： cgset -r **blkio**.throttle.read\_bps\_device="8:32 10240" lv0 blkio.throttle.read\_iops\_device Specifies the maximum number of read operations per second that a cgroup may perform on a device specified by its major and minor numbers. For example, the setting 8:1 100 specifies that a maximum of 100 read operations per second may be performed on/dev/sda1. blkio.throttle.write\_iops\_device Specifies the maximum number of write operations per second that a cgroup may perform on a device specified by its major and minor numbers. For example, the setting 8:2 50 specifies that a maximum of 50 write operations per second may be performed on /dev/sda2.  说明：指定 cgroup 中某设备每秒钟可以执行的读/写请求数上限。其格式为 major:minor operations\_per\_second。 示例：设置 cgroup lv0 在 /dev/sdc 上每秒最多执行 1000 次读请求： cgset -r **blkio**.throttle.read\_iops\_device="8:32 1000" lv0 blkio.weight Specifies a bias value from 100 to 1000 that determines a cgroup's share of access to block I/O. The default value is 1000. The value is overridden by the setting for an individual device (see blkio.weight\_device).  说明：指定 cgroup 默认可用 **IO** 的比例（加权），值的范围为 100 至 1000。该值可由具体设备的 **blkio**.weight\_device 参数覆盖。 示例：设置 cgroup lv0 的默认加权为 500： cgset -r **blkio**.weight=100 lv0 blkio.weight\_device Specifies a bias value from 100 to 1000 that determines a cgroup's share of access to block I/O on a device specified by its major and minor numbers. For example, the setting 8:17 100 specifies a bias value of 100 for /dev/sdb1.  说明：指定 cgroup 中指定设备的可用 **IO** 比例（加权），范围是 100 至 1000。该值的的格式为 major:minor weight ，其中 major 和 minor 参考文档 [Linux 分配的设备](http://www.kernel.org/doc/Documentation/devices.txt)。 示例：设置 cgroup lv0 对 /dev/sda 的加权为 500： cgset -r **blkio**.weight\_device="8:0 500" lv0 |

注意:对块设备的独显的限制,是针对组中所有的进程向块设备的读写速率之和不能超过1Mib/s

Iostat可以查看磁盘IO状态.

Example:

|  |  |  |
| --- | --- | --- |
| **blkio**子系统使用心得  1. 查看一个设备的 major 和 minor 可以使用：ls -l /dev/DEV 例如，要查看 /dev/dm-0 的 major 和 minor：  |  |  | | --- | --- | | 1 2 | ~]# ls -l /dev/dm-0 brw-rw---- 1 root disk 253, 0 Dec 3 01:02 /dev/dm-0 |   其中 253, 0 即为 /dev/dm-0 的 major, minor   1. **blkio**.throttle.\* 的限制是针对整个 cgroup 组的，而不是针对组中单个进程的。 例如，当设置 lv0 **blkio**.throttle.write\_bps\_device="253:0 1048576" 时，整个 lv0 组内所有进程向 /dev/dm-0 的写入速率之**和**不能超过 1 MiB/秒。 2. 经测试，对某个进程的 **blkio** 限制有可能影响 Cgroup 之外的进程。复现方法： /dev/sdb + /dev/sdc 做 flashcache，设置 lv0 **blkio**.throttle.write\_bps\_device="8:32 1048576" ，即限制 lv0 内的进程对 /dev/sdc 的写入速率为 1MiB/秒，然后使用 cgexec -g **blkio**:lv0 dd if=/dev/zero of=/data0/testfile1 bs=10240 count=100000 启动一个在 lv0 限制下的 dd 向 flashcache 写文件。dd 很快便完成，但使用 iostat 观察，发现 /dev/sdc 的写入受 cgroup 限制，只有 1MiB/秒。这时我们再直接执行一个 dd: dd if=/dev/zero of=/data0/testfile1 bs=10240 count=100000，再使用 iostat 查看磁盘 **IO** 状态，/dev/sdc 写入速率仍然是 1MiB/秒，也就是说后面一个 dd 虽不在 cgroup lv0 限制之内，却也受到了 lv0 限流的影响。 3. 进程对 pagecache 的读写操作不受 **blkio**.throttle.\* 限制。 |

cpu/cpuacct/cpuset Parameters

cpu parameters

|  |
| --- |
| cpu.rt\_period\_us Specifies how often in microseconds that a cgroup's access to a CPU should be rescheduled. The default value is 1000000 (1 second). cpu.rt\_runtime.us Specifies for how long in microseconds that a cgroup has access to a CPU between rescheduling operations. The default value is 950000 (0.95 seconds).  通过配置 cpu.rt\_period\_us 和 cpu.rt\_runtime\_us 就可以对 cgroup 中的进程组中的实时进程进行 CPU使用时间的控制*，平时可能不会有机会用到。*   cpu.shares 指定确定cgroup的CPU时间份额的偏置值。 默认值为1024。 |

Cpiacct parameter

|  |
| --- |
| cpuacct.stat 以cgroup中的所有任务报告以用户和系统模式花费的纳秒的总CPU时间。 cpuacct.usage 报告cgroup中所有任务的总CPU时间（以毫秒为单位）。 将此参数设置为0将重置其值，并重置cpuacct.usage\_percpu的值。 cpuacct.usage\_percpu 报告cgroup中所有任务的每个CPU内核的总CPU时间（以毫秒为单位）。 |

Cpuset parameter

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cpuset.cpu\_exclusive 指定由cpuset.cpus指定的CPU是否专门分配给该CPU集，不能与其他CPU集共享。 默认值为0表示CPU不是专门分配的。 值为1可以通过CPU设置独占使用CPU。 cpuset.cpus 指定cgroup可以访问的CPU内核列表。 例如，设置0,1,5-8允许访问内核0，1，5，6，7和8.默认设置包括所有可用的CPU内核。  注意:如果将cpuset子系统与cgroup相关联，则必须为cpuset.cpus参数指定一个值。 cpuset.mem\_exclusive 指定由cpuset.mems指定的内存节点是否专用于此CPU集，并且不能与其他CPU集共享。 默认值为0指定不专门分配内存节点。 值为1可以通过CPU设置独占使用存储器节点。 cpuset.mem\_hardwall 指定内核是否将页面和缓冲区分配给由cpuset.mems指定的内存节点专用于此CPU集，并且不能与其他CPU集共享。 默认值为0指定不专门分配内存节点。 值为1允许您分配分配给不同cgroup的内存节点。 cpuset.memory\_migrate 指定如果cpuset.mems的值更改，是否允许内存页面在内存节点之间迁移。 默认值为0指定不允许内存节点进行迁移。 值1允许页面在存储器节点之间迁移，在可能的情况下保持其在节点列表上的相对位置。 cpuset.memory\_pressure 如果cpuset.memory\_pressure\_enabled设置为1，则会报告内存压力，这表示进程回收使用中内存的每秒尝试次数。 报告的值将实际尝试次数缩小到1000。 cpuset.memory\_pressure\_enabled 指定是否收集内存压力统计信息。 默认值为0将禁用计数器。 值为1启用计数器。 cpuset.memory\_spread\_page 指定文件系统缓冲区是否分配在分配的内存节点之间。 默认值为0会导致缓冲区与拥有它们的进程放在同一个内存节点上。 值为1允许缓冲区分布在CPU集的存储器节点上。 cpuset.memory\_spread\_slab 指定I / O slab缓存是否在分配的内存节点之间分布。 默认值为0导致高速缓存放置在与拥有它们的进程相同的内存节点上。 值为1允许高速缓存分布在CPU集的存储器节点上。 cpuset.mems 指定cgroup可以访问的内存节点。 例如，设置0-2,4允许访问内存节点0,1,2和4.默认设置包括所有可用的内存节点。 在没有NUMA架构的系统上，该参数的值为0。  注意  如果将cpuset子系统与cgroup相关联，则必须为cpuset.mems参数指定一个值。 cpuset.sched\_load\_balance 指定内核是否应通过在分配给CPU集的CPU核心之间移动进程来尝试平衡CPU负载。 默认值为1，打开负载平衡。 值0将禁用负载平衡。 如果在父cgroup中启用负载平衡，则禁用cgroup的负载平衡将无效。 cpuset.sched\_relax\_domain\_level 如果cpuset.sched\_load\_balance设置为1，请指定以下负载平衡方案之一。   | **Setting** | **Description** | | --- | --- | | -1 | Use the system's default load balancing scheme. This is the default behavior. | | 0 | Perform periodic load balancing. Higher numeric values enable immediate load balancing. | | 1 | Perform load balancing for threads running on the same core. | | 2 | Perform load balancing for cores of the same CPU. | | 3 | Perform load balancing for all CPU cores on the same system. | | 4 | Perform load balancing for a subset of CPU cores on a system with a NUMA architecture. | | 5 | Perform load balancing for all CPU cores on a system with a NUMA architecture. | |

Devices Parameters

The following devices parameters are defined:

##### devices.allow

Specifies a device that a cgroup is allowed to access by its type (a for any, b for block, or c for character), its major and minor numbers, and its access modes (m for create permission, r for read access, and w for write access).

For example, b 8:17 rw would allow read and write access to the block device /dev/sdb1.

You can use the wildcard \* to represent any major or minor number. For example, b 8:\* rw would allow read and write access to any /dev/sd\* block device.

Each device that you specify is added to the list of allowed devices.

##### devices.deny

Specifies a device that a cgroup is not allowed to access.

Removes each device that you specify from the list of allowed devices.

##### devices.list

Reports those devices for which access control is set. If no devices are controlled, all devices are reported as being available in all access modes: a \*:\* rwm.

### freezer Parameter

The following freezer parameter is defined:

##### freezer.state

Specifies one of the following operations.

| **Setting** | **Description** |
| --- | --- |
| FROZEN | Suspends all the tasks in a cgroup. You cannot move a process into a frozen cgroup. |
| THAWED | Resumes all the tasks in a cgroup. |

Note

You cannot set the FREEZING state. If displayed, this state indicates that the system is currently suspending the tasks in the cgroup.

The freezer.state parameter is not available in the root cgroup.

### memory Parameters

The following memory parameters are defined:

##### memory.failcnt

报告内存达到 memory.limit\_in\_bytes 设定的限制值的次数。

##### memory.force\_empty

设定为 0 时，该 cgroup 中任务所用的所有页面内存都将被清空。这个接口只可在 cgroup 没有任务时使用。如果无法清空内存，请在可能的情况下将其移动到父 cgroup 中。移除 cgroup 前请使用 memory.force\_empty 参数以免将废弃的页面缓存移动到它的父 cgroup 中。

##### memory.limit\_in\_bytes

设定用户内存（包括文件缓存）的最大用量。如果没有指定单位，则该数值将被解读为字节。但是可以使用后缀代表更大的单位 —— k 或者 K 代表千字节，m 或者 M 代表兆字节 ，g 或者 G 代表千兆字节。

您不能使用 memory.limit\_in\_bytes 限制 root cgroup；您只能对层级中较低的群组应用这些值。 在 memory.limit\_in\_bytes 中写入 -1 可以移除全部已有限制

为了避免out-of-memory错误,memory.limit\_in\_bytes < memory.memsw.limit\_in\_bytes,设置memory.memsw.limit\_in\_bytes小于可以使用的swap空间的总数

##### memory.max\_usage\_in\_bytes

报告 cgroup 中进程所用的最大内存量（以字节为单位）。

##### memory.memsw.failcnt

报告内存和 swap 空间总和达到 memory.memsw.limit\_in\_bytes 设定的限制值的次数。

##### memory.memsw.limit\_in\_bytes

指定用户内存加交换空间允许的最大使用量。 默认单位为字节，但您也可以分别指定千字节，兆字节和千兆字节的k或K，m或M，g或G后缀。 值-1取消限制。

##### memory.memsw.max\_usage\_in\_bytes

报告cgroup中任务使用的最大用户内存和交换空间（以字节为单位）。

##### memory.memsw.usage\_in\_bytes

报告cgroup中任务使用的内存和交换空间的总大小（以字节为单位）。

##### memory.move\_charge\_at\_immigrate

指定在cgroups之间迁移任务时是否移动任务的费用。 您可以指定以下值。

| **Setting** | **Description** |
| --- | --- |
| 0 | Disable moving task charges. |
| 1 | Moves charges for an in-use or swapped-out anonymous page exclusively owned by the task. |
| 2 | Moves charges for file pages that are memory mapped by the task. |
| 3 | Equivalent to specifying both 1 and 2. |

##### memory.numa\_stat

报告每个内存节点（N0，N1，...）的NUMA内存使用量（以字节为单位）以及以下统计信息。

| **Statistic** | **Description** |
| --- | --- |
| anon | The size in bytes of anonymous and swap cache. |
| file | The size in bytes of file-backed memory. |
| total | The sum of the anon, file and unevictable values. |
| unevictable | The size in bytes of unreclaimable memory. |

##### memory.oom\_control

显示内存不足（OOM）通知控制功能的值。

| **Setting** | **Description** |
| --- | --- |
| oom\_kill\_disable | Whether the OOM killer is enabled (0) or disabled (1). |
| under\_oom | Whether the cgroup is under OOM control (1) allowing tasks to be stopped, or not under OOM control (0). |

##### memory.soft\_limit\_in\_bytes

指定用户内存的软上限，包括文件缓存。 默认单位为字节，但您也可以分别指定千字节，兆字节和千兆字节的k或K，m或M，g或G后缀。 值-1取消限制。  
软限制应低于memory.limit\_in\_bytes的硬限制，因为硬限制始终优先。

##### Memory.stat

报告以下内存统计信息。

| **Statistic** | **Description** |
| --- | --- |
| active\_anon | The size in bytes of anonymous and swap cache on active least-recently-used (LRU) list (includes tmpfs). |
| active\_file | The size in bytes of file-backed memory on active LRU list. |
| cache | The size in bytes of page cache (includes tmpfs). |
| hierarchical\_memory\_limit | The size in bytes of the limit of memory for the cgroup hierarchy. |
| hierarchical\_memsw\_limit | The size in bytes of the limit of memory plus swap for the cgroup hierarchy. |
| inactive\_anon | The size in bytes of anonymous and swap cache on inactive LRU list (includes tmpfs). |
| inactive\_file | The size in bytes of file-backed memory on inactive LRU list. |
| mapped\_file | The size in bytes of memory-mapped files (includes tmpfs). |
| pgfault | The number of page faults, where the kernel has to allocate and initialize physical memory for use in the virtual address space of a process. |
| pgmajfault | The number of major page faults, where the kernel has to actively free physical memory before allocation and initialization. |
| pgpgin | The number of paged-in pages of memory. |
| pgpgout | The number of paged-out pages of memory. |
| rss | The size in bytes of anonymous and swap cache (does not include tmpfs). The actual resident set size is given by the sum of rss and mapped\_file. |
| swap | The size in bytes of used swap space. |
| total\_\* | The value of the appended statistic for the cgroup and all of its children. |
| unevictable | The size in bytes of memory that in not reclaimable. |

##### memory.swappiness

指定内核的偏置值，以交换cgroup中进程所使用的内存页面，而不是从页面缓存中回收页面。 小于默认值60的值会降低内核对交换的偏好。 大于60的值增加了交换的偏好。 大于100的值允许系统交换落在cgroup任务的地址空间内的页面。

##### memory.usage\_in\_bytes

报告cgroup中所有任务使用的内存的总大小（以字节为单位）。

##### memory.use\_hierarchy

指定内核是否应尝试从cgroup的层次结构回收内存。 默认值为0可防止从层次结构中的其他任务回收内存。 值为1允许从层次结构中的其他任务回收内存。

[escore@hz-36kv72shtt-0-meezv53v5rjs-kube-master-r2rfes6js65r ~]$ cat /run/flannel/docker

DOCKER\_NETWORK\_OPTIONS="--bip=10.100.28.1/24 --mtu=1500"

"bip": "10.100.45.1/24",

"mtu": 1500

DOCKER\_NETWORK\_OPTIONS="--bip=10.100.56.1/24 --mtu=1500"