

原 CGroup 介绍、应用实例及原理描述

2017年12月06日 14:53:27 DemonHunter211 阅读数：1454

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/kwame211/article/details/78730705>

CGroup 介绍

CGroup 是 Control Groups 的缩写，是 Linux 内核提供了一种可以限制、记录、隔离进程组 (process groups) 所使用的物力资源 (如 cpu memory i/o 等等) 的机制。2007 年进入 Linux 2.6.24 内核，CGROUPs 不是全新创造的，它将进程管理从 cpuset 中剥离出来，作者是 Google 的 Paul Menage。CGROUPs 也是 LXC 为实现虚拟化所使用的资源管理手段。

CGroup 功能及组成

CGroup 是将任意进程进行分组化管理的 Linux 内核功能。CGroup 本身是提供将进程进行分组化管理的功能和接口的基础结构，I/O 或内存的分配控制等具体的资源管理功能是通过这个功能来实现的。这些具体的资源管理功能称为 CGroup 子系统或控制器。CGroup 子系统有控制内存的 Memory 控制器、控制进程调度的 CPU 控制器等。运行中的内核可以使用的 Cgroup 子系统由 /proc/cgroup 来确认。

CGroup 提供了一个 CGroup 虚拟文件系统，作为进行分组管理和各子系统设置的用户接口。要使用 CGroup，必须挂载 CGroup 文件系统。这时通过挂载选项指定使用哪个子系统。

Cgroups提供了以下功能：

- 1.限制进程组可以使用的资源数量 (Resource limiting)。比如：memory子系统可以为进程组设定一个memory使用上限，一旦进程组使用的内存达到限额再申请内存，就会出发 OOM (out of memory)。

- 2.进程组的优先级控制（Prioritization）。比如：可以使用cpu子系统为某个进程组分配特定cpu share。
- 3.记录进程组使用的资源数量（Accounting）。比如：可以使用cpuacct子系统记录某个进程组使用的cpu时间
- 4.进程组隔离（Isolation）。比如：使用ns子系统可以使不同的进程组使用不同的namespace，以达到隔离的目的，不同的进程组有各自的进程、网络、文件系统挂载空间。
- 5.进程组控制（Control）。比如：使用freezer子系统可以将进程组挂起和恢复。

CGroup 支持的文件种类

表 1. CGroup 支持的文件种类

文件名	R/W	用途
Release_agent	RW	删除分组时执行的命令，这个文件只存在于根分组
Notify_on_release	RW	设置是否执行 release_agent。为 1 时执行
Tasks	RW	属于分组的线程 TID 列表
Cgroup.procs	R	属于分组的进程 PID 列表。仅包括多线程进程的线程 leader 的 TID，这点与 tasks 不同
Cgroup.event_control	RW	监视状态变化和分组删除事件的配置文件

CGroup 相关概念解释

1. 任务（task）。在 cgroups 中，任务就是系统的一个进程；
2. 控制族群（control group）。控制族群就是一组按照某种标准划分的进程。Cgroups 中的资源控制都是以控制族群为单位实现。一个进程可以加入到某个控制族群，也从一个进程组迁移到另一个控制族群。一个进程组的进程可以使用 cgroups 以控制族群为单位分配的资源，同时受

到 cgroups 以控制族群为单位设定的限制；

3. 层级 (hierarchy)。控制族群可以组织成 hierarchical 的形式，既一颗控制族群树。控制族群树上的子节点控制族群是父节点控制族群的孩子，继承父控制族群的特定的属性；
4. 子系统 (subsystem)。一个子系统就是一个资源控制器，比如 cpu 子系统就是控制 cpu 时间分配的一个控制器。子系统必须附加 (attach) 到一个层级上才能起作用，一个子系统附加到某个层级以后，这个层级上的所有控制族群都受到这个子系统的控制。

相互关系

1. 每次在系统中创建新层级时，该系统中的所有任务都是那个层级的默认 cgroup（我们称之为 root cgroup，此 cgroup 在创建层级时自动创建，后面在该层级中创建的 cgroup 都是此 cgroup 的后代）的初始成员；
2. 一个子系统最多只能附加到一个层级；
3. 一个层级可以附加多个子系统；
4. 一个任务可以是多个 cgroup 的成员，但是这些 cgroup 必须在不同的层级；
5. 系统中的进程（任务）创建子进程（任务）时，该子任务自动成为其父进程所在 cgroup 的成员。然后可根据需要将该子任务移动到不同的 cgroup 中，但开始时它总是继承其父任务的 cgroup。

图 1. CGroup 层级图



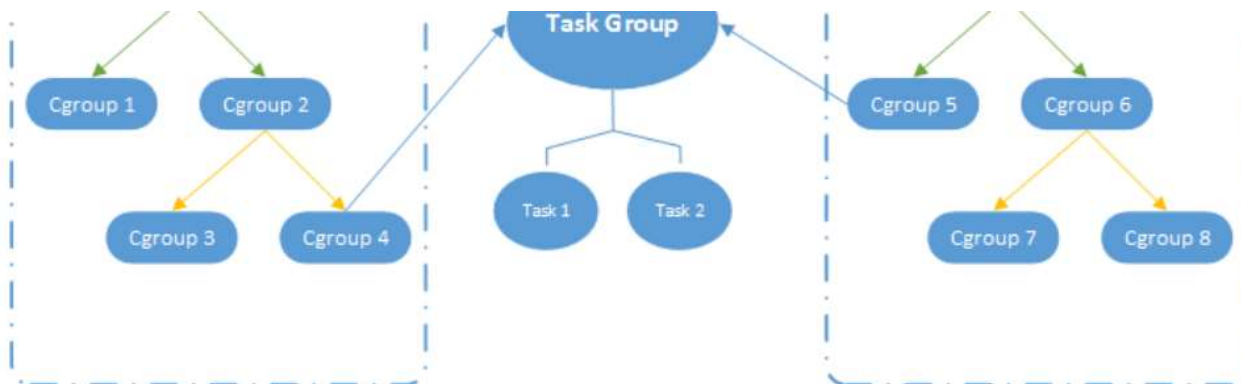


图 1 所示的 CGroup 层级关系显示，CPU 和 Memory 两个子系统有自己独立的层级系统，而又通过 Task Group 取得关联关系。

CGroup 特点

在 cgroups 中，任务就是系统的一个进程。

控制族群（control group）。控制族群就是一组按照某种标准划分的进程。Cgroups 中的资源控制都是以控制族群为单位实现。一个进程可以加入到某个控制族群，也从一个进程组迁移到另一个控制族群。一个进程组的进程可以使用 cgroups 以控制族群为单位分配的资源，同时受到 cgroups 以控制族群为单位设定的限制。

层级（hierarchy）。控制族群可以组织成 hierarchical 的形式，既一颗控制族群树。控制族群树上的子节点控制族群是父节点控制族群的孩子，继承父控制族群的特定的属性。

子系统（subsystem）。一个子系统就是一个资源控制器，比如 cpu 子系统就是控制 cpu 时间分配的一个控制器。子系统必须附加（attach）到一个层级上才能起作用，一个子系统附加到某个层级以后，这个层级上的所有控制族群都受到这个子系统的控制。

子系统的介绍

blkio -- 这个子系统为块设备设定输入/输出限制，比如物理设备（磁盘，固态硬盘，USB 等等）。

cpu -- 这个子系统使用调度程序提供对 CPU 的 cgroup 任务访问。

cpuacct -- 这个子系统自动生成 cgroup 中任务所使用的 CPU 报告。

cpuset -- 这个子系统为 cgroup 中的任务分配独立 CPU（在多核系统）和内存节点。

devices -- 这个子系统可允许或者拒绝 cgroup 中的任务访问设备。

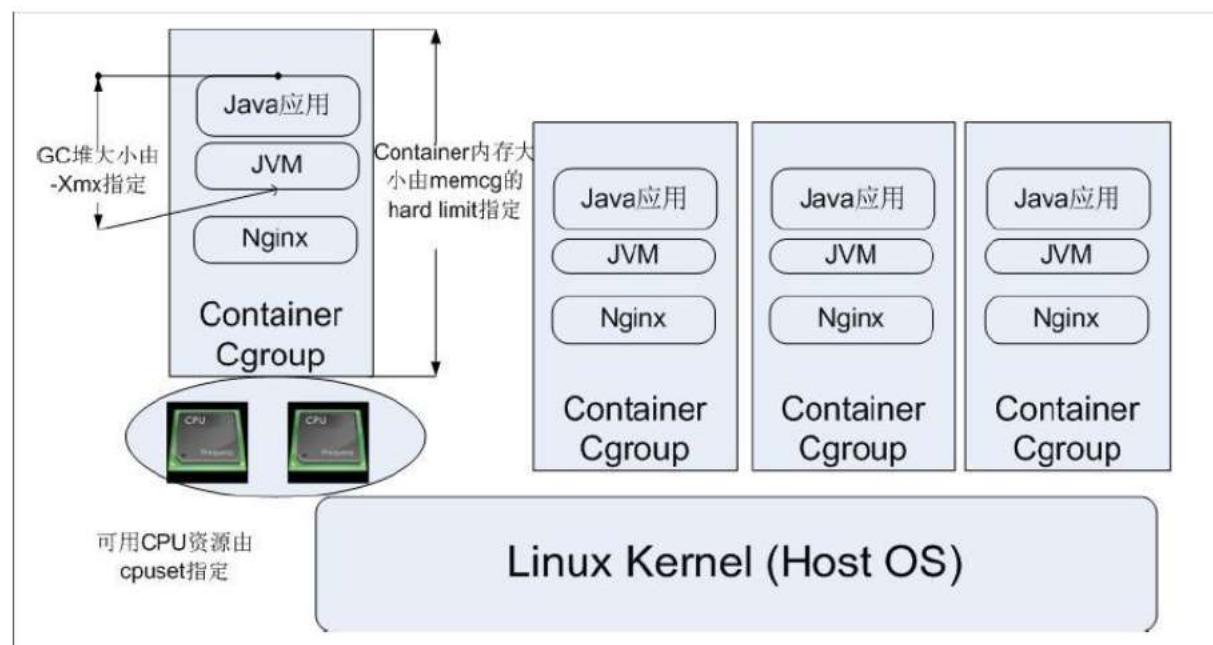
freezer -- 这个子系统挂起或者恢复 cgroup 中的任务。

memory -- 这个子系统设定 cgroup 中任务使用的内存限制，并自动生成由那些任务使用的内存资源报告。

net_cls -- 这个子系统使用等级识别符（classid）标记网络数据包，可允许 Linux 流量控制程序（tc）识别从具体 cgroup 中生成的数据包。

CGROUP 应用架构

图 2. CGroup 典型应用架构图



如图 2 所示，CGroup 技术可以被用来在操作系统底层限制物理资源，起到 Container 的作用。图中每一个 JVM 进程对应一个 Container Cgroup 层级，通过 CGroup 提供的各类子系统，可以对每一个 JVM 进程对应的线程级别进行物理限制，这些限制包括 CPU、内存等等许多种类的资源。下一部分会具体对应用程序进行 CPU 资源隔离进行演示。

cgroup的安装

其实安装很简单，最佳实践就是yum直接安装（centos下）

配置文件

/etc/cgconfig.conf

```
1. mount {
    cpuset = /cgroup/cpuset;
    cpu    = /cgroup/cpu;
    cpuacct = /cgroup/cpuacct;
    memory = /cgroup/memory;
    devices = /cgroup/devices;
    freezer = /cgroup/freezer;
    net_cls = /cgroup/net_cls;
    blkio   = /cgroup/blkio;
}
```

cgroup section的语法格式如下

```
1. group <name> {
2.     [<permissions>]
3.     <controller> {
4.         <param name> = <param value>;
5.     }
```

7. ...}

name: 指定cgroup的名称

permissions: 可选项, 指定cgroup对应的挂载点文件系统的权限, root用户拥有所有权限。

controller: 子系统的名称

param name 和 param value: 子系统的属性及其属性值

7.1 配置对mysql实例的资源限制

前提: **MySQL数据库**已在机器上安装

7.1.1 修改cgconfig.conf文件

```
[plain]
1.  mount {
2.      cpuset  = /cgroup/cpuset;
3.      cpu     = /cgroup/cpu;
4.      cpuacct = /cgroup/cpuacct;
5.      memory  = /cgroup/memory;
6.      blkio   = /cgroup/blkio;
7.
8.
9.  group mysql_g1 {
10.     cpu {
11.         cpu.cfs_quota_us = 50000;
12.     }
13.     cpuset {
14.         ,      cpuset.cpus = "3":
15.     }
16.     cpuacct{
```

```
19.
20.     }
21.     memory {
22.         memory.limit_in_bytes=104857600;
23.         memory.swappiness=0;
24.         # memory.max_usage_in_bytes=104857600;
25.         # memory.oom_control=0;
26.     }
27.     blkio {
28.         blkio.throttle.read_bps_device="8:0 524288";
29.         blkio.throttle.write_bps_device="8:0 524288";
30.     }
31. }
```

7.1.2 配置文件的部分解释。

cpu: cpu使用时间限额。

cpu.cfs_period_us和cpu.cfs_quota_us来限制该组中的所有进程在单位时间里可以使用的cpu时间。这里的cfs是完全公平调度器的缩写。cpu.cfs_period_us就是时间周期(微秒)，默认为100000，即百毫秒。cpu.cfs_quota_us就是在这期间内可使用的cpu时间(微秒)，默认-1，即无限制。(cfs_quota_us是cfs_period_us的两倍即可限定在双核上完全使用)。

cpuset: cpu绑定

我们限制该组只能在0一共1个超线程上运行。cpuset.mems是用来设置内存节点的。

本例限制使用超线程0上的第四个cpu线程。

其实cgconfig也就是帮你把配置文件中的配置整理到/cgroup/cpuset这个目录里面，比如你需要动态设置mysql_group1/ cpuset.cpus的CPU超线程号，可以采用如下的办法。


```
[plain] 1. [root@localhost ~]# echo "0" > mysql_group1/ cpuset.cpus
```

cpuacct: cpu资源报告

memory: 内存限制

内存限制我们主要限制了MySQL可以使用的内存最大大小memory.limit_in_bytes=256M。而设置swappiness为0是为了让操作系统不会将MySQL的内存匿名页交换出去。

blkio: BLOCK IO限额

blkio.throttle.read_bps_device="8:0 524288"; #每秒读数据上限

blkio.throttle.write_bps_device="8:0 524288"; #每秒写数据上限

其中8:0对应主设备号和副设备号，可以通过ls -l /dev/sda查看

```
[plain] 1. [root@localhost /]# ls -l /dev/sda
2. brw-rw----. 1 root disk 8, 0 Sep 15 04:19 /dev/sda
```

7.1.3 拓展知识

现在较新的服务器CPU都是numa结构<非一致内存访问结构（NUMA: Non-Uniform Memory Access）>，使用numactl --hardware可以看到numa各个节点的CPU超线程号，以及对应的节点号。

本例结果如下：

```
[plain] 1. [root@localhost /]# numactl --hardware
```

```
2. available: 1 nodes (0)
3. node 0 cpus: 0 1 2 3
4. node 0 size: 1023 MB
5. node 0 free: 68 MB
6. node distances:
7. node    0
8.    0:   10
```

以下是较高端服务器的numa信息，仅作参考。

```
[plain]

1. [root@localhost ~]# numactl --hardware
2. available: 4 nodes (0-3)
3. node 0 cpus: 0 4 8 12 16 20 24 28 32 36 40 44 48 52 56 60
4. node 0 size: 16338 MB
5. node 0 free: 391 MB
6. node 1 cpus: 1 5 9 13 17 21 25 29 33 37 41 45 49 53 57 61
7. node 1 size: 16384 MB
8. node 1 free: 133 MB
9. node 2 cpus: 2 6 10 14 18 22 26 30 34 38 42 46 50 54 58 62
10. node 2 size: 16384 MB
11. node 2 free: 137 MB
12. node 3 cpus: 3 7 11 15 19 23 27 31 35 39 43 47 51 55 59 63
13. node 3 size: 16384 MB
14. node 3 free: 186 MB
15. node distances:
16. node    0    1    2    3
17.    0:   10   20   30   20
18.    1:   20   10   20   30
19.    2:   30   20   10   20
20.    3:   20   30   20   10
```

7.1.4 修改cgrules.conf文件

```
[plain]
```

```
1. [root@localhost ~]# vi /etc/cgrouprules.conf
2. # /etc/cgrouprules.conf
3. #The format of this file is described in cgrouprules.conf(5)
4. #manual page.
5. #
6. # Example:
7. #<user>          <controllers>   <destination>
8. #@student        cpu,memory      usergroup/student/
9. #peter           cpu             test1/
10. #%               memory          test2/
11. */usr/local/mysql/bin/mysqld * mysql_g1
```

注：共分为3个部分，分别为需要限制的实例，限制的内容（如cpu，memory），挂载目标。

7.2 使配置生效

```
[plain]
1. [root@localhost ~]# /etc/init.d/cgconfig restart
2. Stopping cgconfig service: [ OK ]
3. Starting cgconfig service: [ OK ]
4. [root@localhost ~]# /etc/init.d/cgred restart
5. Stopping CGroup Rules Engine Daemon... [ OK ]
6. Starting CGroup Rules Engine Daemon: [ OK ]
```

注：重启顺序为cgconfig -> cgred，更改配置文件后两个服务需要重启，且顺序不能错。

7.3 启动MySQL，查看MySQL是否处于cgroup的限制中

```
[plain]
1. [root@localhost ~]# ps -eo pid,cgroup,cmd | grep -i mysqld
2. 29871 blkio:;/net_cls:;/freezer:;/devices:;/memory:;/cpuacct:;/cpu:;/cpuset:/ /bin/sh ./bin/mys
   qld_safe --defaults-file=/etc/my.cnf --basedir=/usr/local/mysql/ --
   datadir=/usr/local/mysql/data/
```


3. 30219 blkio:;/net_cls:;/freezer:;/devices:;/memory:;/cpuacct:;/cpu:;/cpuset:/mysql_g1 /usr/local/mysql/bin/mysqld --defaults-file=/etc/my.cnf --basedir=/usr/local/mysql/ --datadir=/usr/local/mysql/data/ --plugin-dir=/usr/local/mysql/lib/plugin --user=mysql --log-error=/usr/local/mysql/data//localhost.localdomain.err --pid-file=/usr/local/mysql/data//localhost.localdomain.pid --socket=/tmp/mysql.sock --port=3306
4. 30311 blkio:;/net_cls:;/freezer:;/devices:;/memory:;/cpuacct:;/cpu:;/cpuset:/ grep -i mysqld

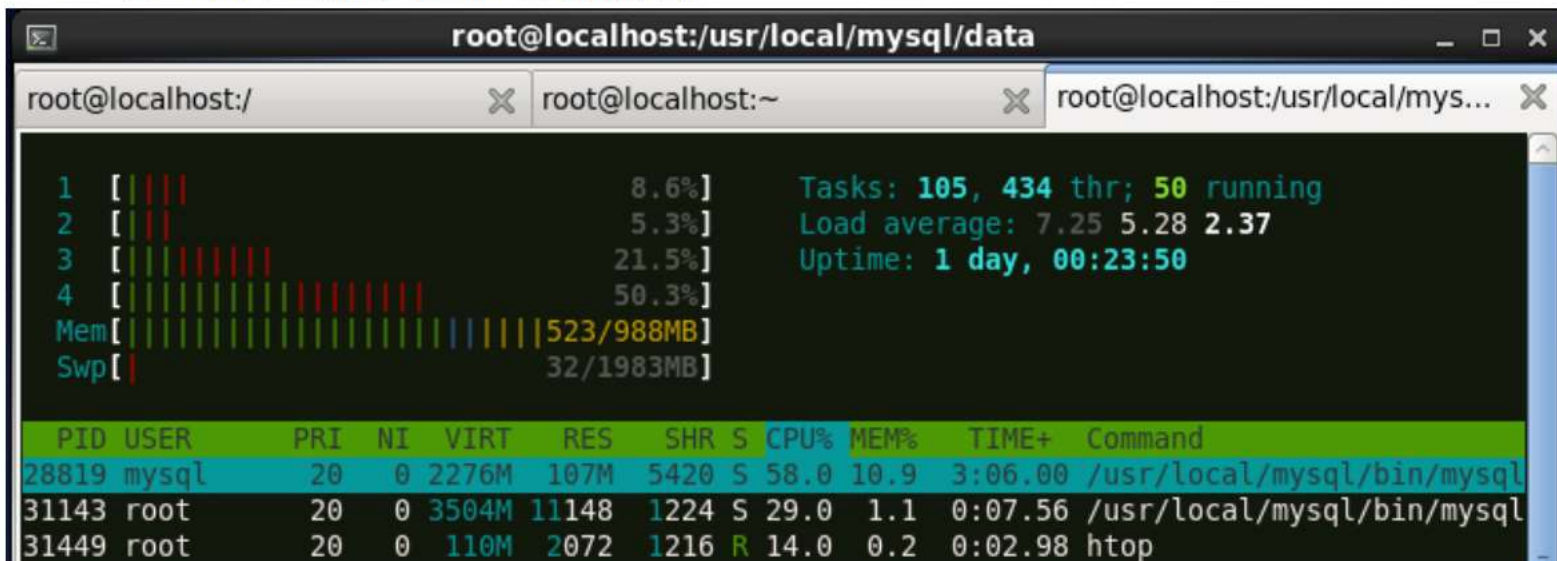
7.4 资源限制验证

使用mysqlslap对mysql进行压力测试，看mysql使用资源是否超过限制。

7.4.1 在shell窗口1用mysqlslap对mysql进行压力测试。

- ```
[plain]
1. [root@localhost ~]# /usr/local/mysql/bin/mysqlslap --defaults-file=/etc/my.cnf --concurrency=150 --iterations=1 --number-int-cols=8 --auto-generate-sql --auto-generate-sql-load-type=mixed --engine=innodb --number-of-queries=100000 -ujesse -pjesse --number-char-cols=35 --auto-generate-sql-add-autoincrement --debug-info -P3306 -h127.0.0.1
```

7.4.2 在shell窗口2查看mysql对cpu，内存的使用



|       |       |    |   |       |       |      |   |     |      |         |                            |
|-------|-------|----|---|-------|-------|------|---|-----|------|---------|----------------------------|
| 28871 | mysql | 20 | 0 | 2276M | 107M  | 5420 | S | 5.0 | 10.9 | 0:12.70 | /usr/local/mysql/bin/mysql |
| 31233 | root  | 20 | 0 | 3504M | 11148 | 1224 | S | 2.0 | 1.1  | 0:00.08 | /usr/local/mysql/bin/mysql |
| 1510  | root  | 20 | 0 | 184M  | 2244  | 1908 | S | 1.0 | 0.2  | 5:44.55 | /usr/sbin/vmtoolsd         |
| 31328 | mysql | 20 | 0 | 2276M | 107M  | 5420 | R | 1.0 | 10.9 | 0:00.09 | /usr/local/mysql/bin/mysql |
| 3158  | root  | 20 | 0 | 201M  | 33488 | 6936 | S | 1.0 | 3.3  | 5:43.63 | /usr/bin/Xorg :0 -nr -verb |
| 3579  | root  | 20 | 0 | 398M  | 45568 | 4884 | S | 1.0 | 4.5  | 6:27.36 | /usr/lib/vmware-tools/sbin |
| 31356 | mysql | 20 | 0 | 2276M | 107M  | 5420 | S | 1.0 | 10.9 | 0:00.09 | /usr/local/mysql/bin/mysql |
| 31360 | mysql | 20 | 0 | 2276M | 107M  | 5420 | R | 1.0 | 10.9 | 0:00.09 | /usr/local/mysql/bin/mysql |
| 31446 | mysql | 20 | 0 | 2276M | 107M  | 5420 | R | 1.0 | 10.9 | 0:00.09 | /usr/local/mysql/bin/mysql |
| 31340 | mysql | 20 | 0 | 2276M | 107M  | 5420 | R | 1.0 | 10.9 | 0:00.09 | /usr/local/mysql/bin/mysql |
| 31244 | root  | 20 | 0 | 3504M | 11148 | 1224 | S | 1.0 | 1.1  | 0:00.06 | /usr/local/mysql/bin/mysql |

可见：cpu限制在了第四个核心上，且对第四个核心的使用限制在50%。

#### 7.4.3 在shell窗口3查看io的消耗

| root@localhost:~                                       |      |       |           |             |        |         |                               |  |  |  |
|--------------------------------------------------------|------|-------|-----------|-------------|--------|---------|-------------------------------|--|--|--|
| root@localhost:/                                       |      |       |           |             |        |         |                               |  |  |  |
| Total DISK READ: 0.00 B/s   Total DISK WRITE: 3.46 M/s |      |       |           |             |        |         |                               |  |  |  |
| TID                                                    | PRI  | USER  | DISK READ | DISK WRITE  | SWAPIN | IO>     | COMMAND                       |  |  |  |
| 31337                                                  | be/4 | mysql | 0.00 B/s  | 801.77 K/s  | 0.00 % | 36.11 % | mysqld --based~ck --port=3306 |  |  |  |
| 31389                                                  | be/4 | mysql | 0.00 B/s  | 138.24 K/s  | 0.00 % | 27.40 % | mysqld --based~ck --port=3306 |  |  |  |
| 31446                                                  | be/4 | mysql | 0.00 B/s  | 49.15 K/s   | 0.00 % | 13.06 % | mysqld --based~ck --port=3306 |  |  |  |
| 31306                                                  | be/4 | mysql | 0.00 B/s  | 12.29 K/s   | 0.00 % | 5.47 %  | mysqld --based~ck --port=3306 |  |  |  |
| 377                                                    | be/3 | root  | 0.00 B/s  | 0.00 B/s    | 0.00 % | 2.15 %  | [jbd2/dm-0-8]                 |  |  |  |
| 28832                                                  | be/4 | mysql | 0.00 B/s  | 0.00 B/s    | 0.00 % | 1.34 %  | mysqld --based~ck --port=3306 |  |  |  |
| 28871                                                  | be/4 | mysql | 0.00 B/s  | 1738.70 K/s | 0.00 % | 0.83 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31352                                                  | be/4 | mysql | 0.00 B/s  | 12.29 K/s   | 0.00 % | 0.47 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31420                                                  | be/4 | mysql | 0.00 B/s  | 3.07 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31439                                                  | be/4 | mysql | 0.00 B/s  | 6.14 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31297                                                  | be/4 | mysql | 0.00 B/s  | 3.07 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31298                                                  | be/4 | mysql | 0.00 B/s  | 6.14 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31299                                                  | be/4 | mysql | 0.00 B/s  | 3.07 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31300                                                  | be/4 | mysql | 0.00 B/s  | 3.07 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31301                                                  | be/4 | mysql | 0.00 B/s  | 3.07 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31302                                                  | be/4 | mysql | 0.00 B/s  | 3.07 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |
| 31303                                                  | be/4 | mysql | 0.00 B/s  | 3.07 K/s    | 0.00 % | 0.00 %  | mysqld --based~ck --port=3306 |  |  |  |



```

31304 be/4 mysql 0.00 B/s 3.07 K/s 0.00 % 0.00 % mysqld --based~ck --port=3306
31305 be/4 mysql 0.00 B/s 6.14 K/s 0.00 % 0.00 % mysqld --based~ck --port=3306
31307 be/4 mysql 0.00 B/s 3.07 K/s 0.00 % 0.00 % mysqld --based~ck --port=3306
31308 be/4 mysql 0.00 B/s 6.14 K/s 0.00 % 0.00 % mysqld --based~ck --port=3306
31309 be/4 mysql 0.00 B/s 6.14 K/s 0.00 % 0.00 % mysqld --based~ck --port=3306

```

可见：mysql对io的读及写消耗均限制在2M每秒以内。

## 8 cgroup实例分析（手工动态验证）

还原配置文件/etc/cgconfig.conf及/etc/cgrules.conf 为默认配置。测试实例依然为mysql，测试工具为mysqlslap。

开启cgconfig及cgrules 服务。

```

[plain]
1. [root@localhost ~]# /etc/init.d/cgconfig restart
2. Stopping cgconfig service: [OK]
3. Starting cgconfig service: [OK]
4. [root@localhost ~]# /etc/init.d/cgred restart
5. Stopping CGroup Rules Engine Daemon... [OK]
6. Starting CGroup Rules Engine Daemon: [OK]

```

开启mysqlslap压力测试程序。

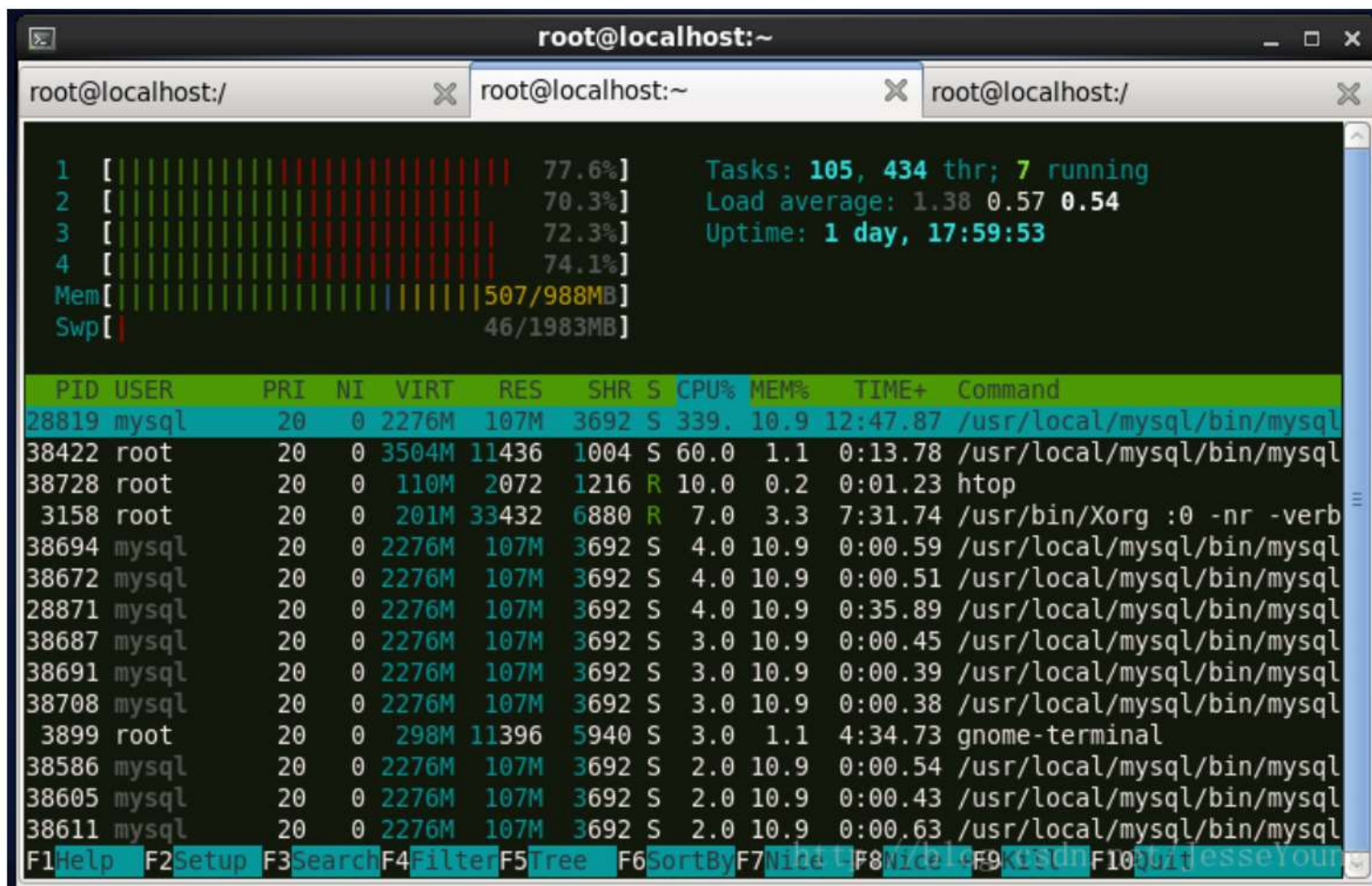
```

[plain]
1. [root@localhost ~]# /usr/local/mysql/bin/mysqlslap --defaults-file=/etc/my.cnf --
concurrency=150 --iterations=1 --number-int-cols=8 --auto-generate-sql --auto-generate-sql-
load-type=mixed --engine=innodb --number-of-queries=100000 -ujesse -pjesse --number-char-
cols=35 --auto-generate-sql-add-autoincrement --debug-info -P3306 -h127.0.0.1

```

通过htop查看资源消耗。





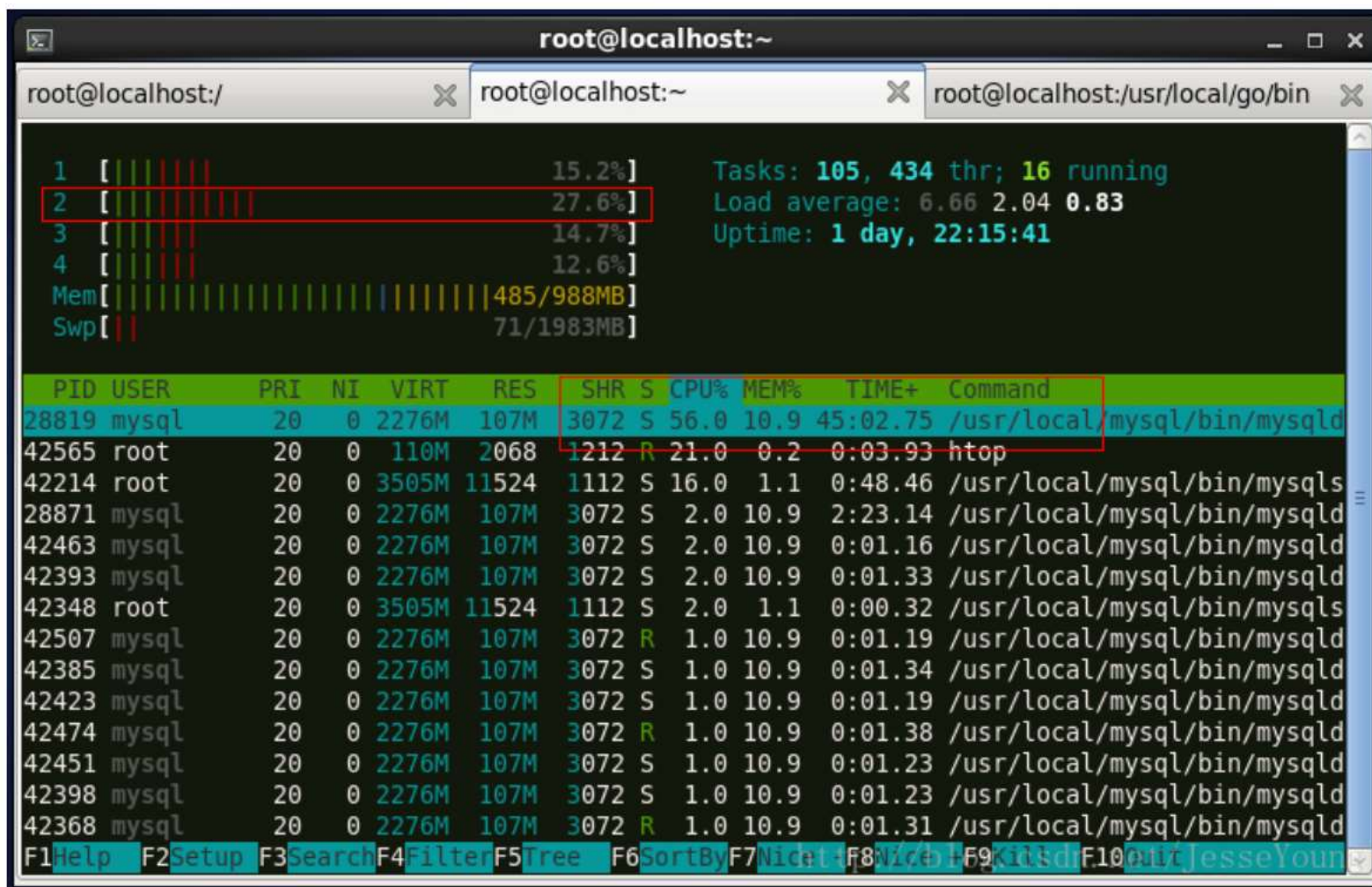
## 8.1 cpu限制实例

限制mysql使用一个核，如第2个核，且对该核的使用不超过50%

```
[plain]
1. [root@localhost /]# mkdir -p /cgroup/cpu/foo/
2. [root@localhost /]# mkdir -p /cgroup/cpuset/foo/
3. [root@localhost /]# echo 50000 > /cgroup/cpu/foo/cpu.cfs_quota_us
4. [root@localhost /]# echo 100000 > /cgroup/cpu/foo/cpu.cfs_period_us
5. [root@localhost /]# echo "0" > /cgroup/cpuset/foo/cpuset.mems
```

6. [root@localhost /]# echo "1" > /cgroup/cpuset/foo/cpuset.cpus
7. [root@localhost /]# echo 28819 > /cgroup/cpu/foo/tasks

其中：28819为mysqld的进程号。



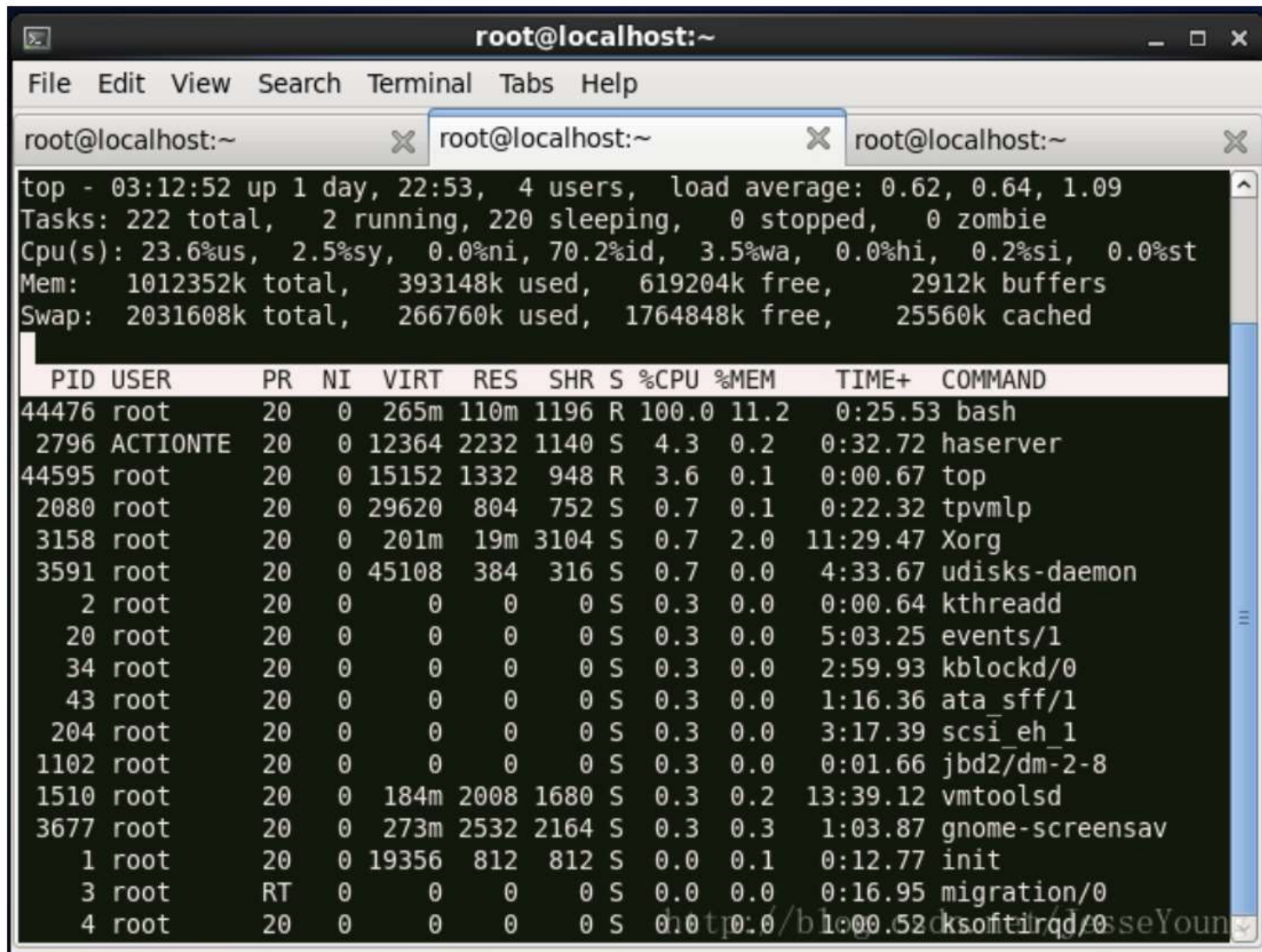
## 8.2 内存限制实例

限制mysql使用内存为不超过512M



## 跑一个消耗内存脚本

```
[plain]
1. x='a'
2. while [True];do
3. x=xx
4. done;
```



The terminal window shows the output of the 'top' command, displaying system statistics and a list of running processes. The system statistics include: 4 users, load average of 0.62, 0.64, 1.09, 222 total tasks (2 running, 220 sleeping), CPU usage of 23.6% user, 2.5% system, 70.2% idle, and memory usage of 393148k used out of 1012352k total.

| PID   | USER     | PR | NI | VIRT  | RES  | SHR  | S | %CPU  | %MEM | TIME+    | COMMAND         |
|-------|----------|----|----|-------|------|------|---|-------|------|----------|-----------------|
| 44476 | root     | 20 | 0  | 265m  | 110m | 1196 | R | 100.0 | 11.2 | 0:25.53  | bash            |
| 2796  | ACTIONTE | 20 | 0  | 12364 | 2232 | 1140 | S | 4.3   | 0.2  | 0:32.72  | haserver        |
| 44595 | root     | 20 | 0  | 15152 | 1332 | 948  | R | 3.6   | 0.1  | 0:00.67  | top             |
| 2080  | root     | 20 | 0  | 29620 | 804  | 752  | S | 0.7   | 0.1  | 0:22.32  | tpvmlp          |
| 3158  | root     | 20 | 0  | 201m  | 19m  | 3104 | S | 0.7   | 2.0  | 11:29.47 | Xorg            |
| 3591  | root     | 20 | 0  | 45108 | 384  | 316  | S | 0.7   | 0.0  | 4:33.67  | udisks-daemon   |
| 2     | root     | 20 | 0  | 0     | 0    | 0    | S | 0.3   | 0.0  | 0:00.64  | kthreadd        |
| 20    | root     | 20 | 0  | 0     | 0    | 0    | S | 0.3   | 0.0  | 5:03.25  | events/1        |
| 34    | root     | 20 | 0  | 0     | 0    | 0    | S | 0.3   | 0.0  | 2:59.93  | kblockd/0       |
| 43    | root     | 20 | 0  | 0     | 0    | 0    | S | 0.3   | 0.0  | 1:16.36  | ata_sff/1       |
| 204   | root     | 20 | 0  | 0     | 0    | 0    | S | 0.3   | 0.0  | 3:17.39  | scsi_eh_1       |
| 1102  | root     | 20 | 0  | 0     | 0    | 0    | S | 0.3   | 0.0  | 0:01.66  | jbd2/dm-2-8     |
| 1510  | root     | 20 | 0  | 184m  | 2008 | 1680 | S | 0.3   | 0.2  | 13:39.12 | vmtoolsd        |
| 3677  | root     | 20 | 0  | 273m  | 2532 | 2164 | S | 0.3   | 0.3  | 1:03.87  | gnome-screensav |
| 1     | root     | 20 | 0  | 19356 | 812  | 812  | S | 0.0   | 0.1  | 0:12.77  | init            |
| 3     | root     | RT | 0  | 0     | 0    | 0    | S | 0.0   | 0.0  | 0:16.95  | migration/0     |
| 4     | root     | 20 | 0  | 0     | 0    | 0    | S | 0.0   | 0.0  | 0:00.52  | ksoftirqd/0     |



内存的消耗在不断增加，对其进行限制，使其使用内存存在500M以内

- ```
[plain]
```
1. [root@localhost ~]# mkdir -p /cgroup/memory/foo
 2. [root@localhost ~]# echo 524288000 > /cgroup/memory/foo/memory.limit_in_bytes
 3. [root@localhost ~]# echo 44476 > /cgroup/memory/foo/tasks

```
root@localhost:~
File Edit View Search Terminal Tabs Help
root@localhost:~ root@localhost:~ root@localhost:~
top - 03:14:14 up 1 day, 22:54, 4 users, load average: 1.12, 0.77, 1.09
Tasks: 222 total, 3 running, 219 sleeping, 0 stopped, 0 zombie
Cpu(s): 3.7%us, 20.5%sy, 0.0%ni, 74.6%id, 0.1%wa, 0.2%hi, 1.0%si, 0.0%st
Mem: 1012352k total, 792592k used, 219760k free, 3004k buffers
Swap: 2031608k total, 373608k used, 1658000k free, 25804k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 44476 root        20   0   745m 498m 1196  R   93.4   50.5   1:47.93  bash
    34 root        20   0     0    0     0   S    4.2    0.0   3:00.06  kblockd/0
    35 root        20   0     0    0     0   R    2.6    0.0   0:12.95  kblockd/1
 44336 mysql      20   0   663m 7548  900   S    2.6    0.7   0:28.90  mysqld
     4 root        20   0     0    0     0   S    2.3    0.0   1:00.63  ksoftirqd/0
 44595 root        20   0   15152 1332  948   R    2.3    0.1   0:02.17  top
    37 root        20   0     0    0     0   S    1.3    0.0   0:08.95  kblockd/3
 1510 root        20   0   184m 2008 1680   S    0.6    0.2  13:39.61  vmtoolsd
 3158 root        20   0   201m   19m 3168   S    0.6    2.0  11:32.17  Xorg
    20 root        20   0     0    0     0   S    0.3    0.0   5:03.48  events/1
    36 root        20   0     0    0     0   S    0.3    0.0   0:08.36  kblockd/2
    42 root        20   0     0    0     0   S    0.3    0.0   1:40.16  ata_sff/0
   204 root        20   0     0    0     0   S    0.3    0.0   3:17.49  scsi_eh_1
   377 root        20   0     0    0     0   S    0.3    0.0   5:15.53  jbd2/dm-0-8
  1104 root        20   0     0    0     0   S    0.3    0.0   0:49.56  flush-253:0
  3591 root        20   0  45108  384  316   S    0.3    0.0   4:33.83  udisks-daemon
     1 root        20   0  19356  812  812   S    0.0    0.0   0:01.12  init
```

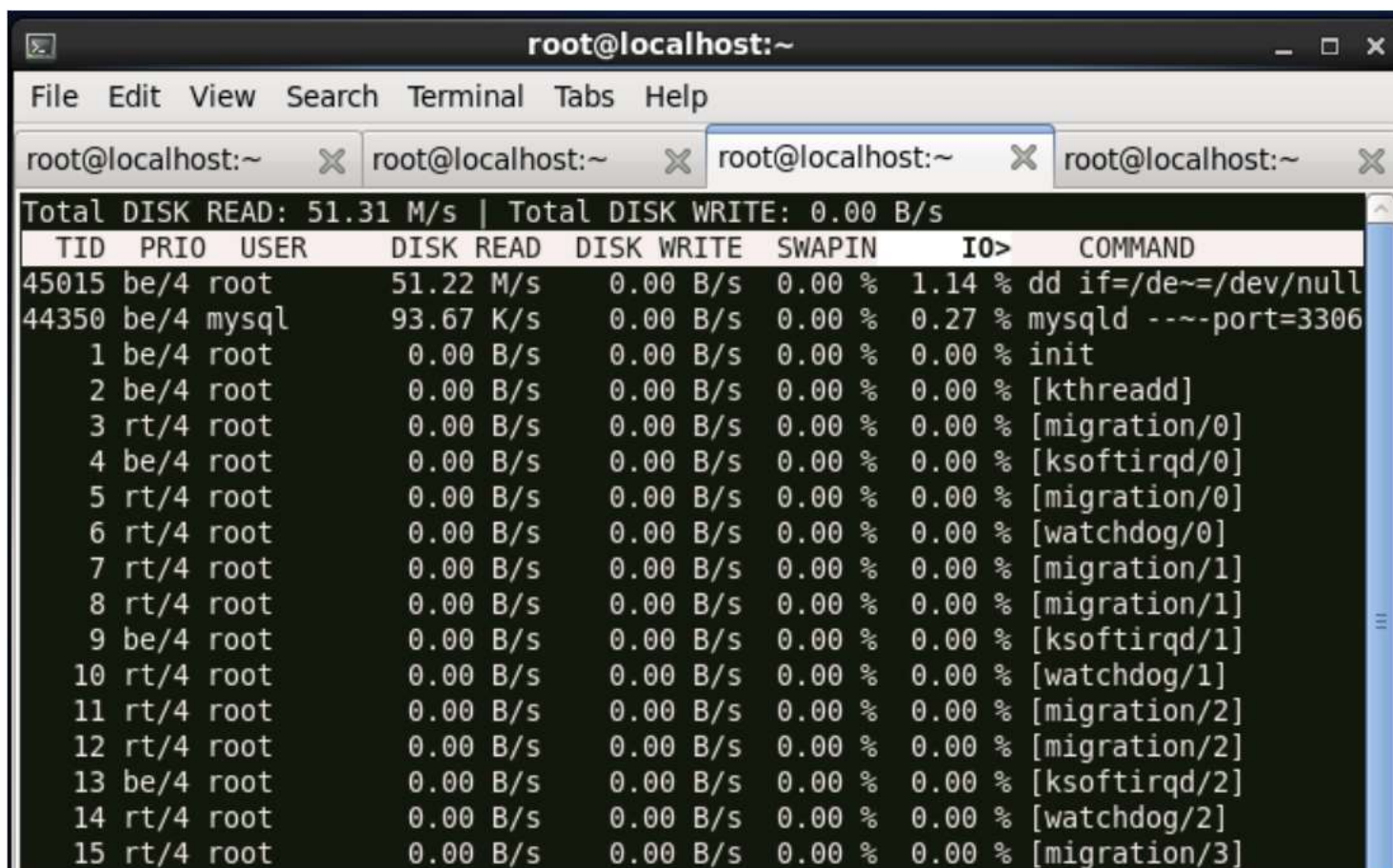
内存使用得到了有效控制。

8.3 IO限制实例

跑一个消耗IO的测试

```
1. [root@localhost ~]# dd if=/dev/sda of=/dev/null
```

通过iotop看io占用情况，磁盘读取速度到了50M/s



The screenshot shows a terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help) and four tabs. The main content displays iotop output. At the top, it shows 'Total DISK READ: 51.31 M/s | Total DISK WRITE: 0.00 B/s'. Below this is a table with columns: TID, PRIO, USER, DISK READ, DISK WRITE, SWAPIN, IO>, and COMMAND. The first row shows a process with TID 45015, PRIO be/4, USER root, DISK READ 51.22 M/s, DISK WRITE 0.00 B/s, SWAPIN 0.00 %, IO> 1.14 %, and COMMAND 'dd if=/de~/dev/null'. The second row shows a process with TID 44350, PRIO be/4, USER mysql, DISK READ 93.67 K/s, DISK WRITE 0.00 B/s, SWAPIN 0.00 %, IO> 0.27 %, and COMMAND 'mysqld ---port=3306'. The remaining rows show system processes with zero disk activity.

TID	PRIO	USER	DISK READ	DISK WRITE	SWAPIN	IO>	COMMAND
45015	be/4	root	51.22 M/s	0.00 B/s	0.00 %	1.14 %	dd if=/de~/dev/null
44350	be/4	mysql	93.67 K/s	0.00 B/s	0.00 %	0.27 %	mysqld ---port=3306
1	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	init
2	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[kthreadd]
3	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/0]
4	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/0]
5	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/0]
6	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[watchdog/0]
7	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/1]
8	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/1]
9	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/1]
10	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[watchdog/1]
11	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/2]
12	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/2]
13	be/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[ksoftirqd/2]
14	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[watchdog/2]
15	rt/4	root	0.00 B/s	0.00 B/s	0.00 %	0.00 %	[migration/3]

16	rt/4	root	0.00	B/s	0.00	B/s	0.00	%	0.00	%	[migration/3]
17	be/4	root	0.00	B/s	0.00	B/s	0.00	%	0.00	%	[ksoftirqd/3]
18	rt/4	root	0.00	B/s	0.00	B/s	0.00	%	0.00	%	[watchdog/3]
19	be/4	root	0.00	B/s	0.00	B/s	0.00	%	0.00	%	[events/0]
20	be/4	root	0.00	B/s	0.00	B/s	0.00	%	0.00	%	[events/1]

限制读取速度为10M/S

```
[plain]
1. [root@localhost ~]# mkdir -p /cgroup/blkio/foo
2. [root@localhost ~]# echo '8:0 10485760' > /cgroup/blkio/foo/blkio.throttle.read_bps_device
3. [root@localhost ~]# echo 45033 > /cgroup/blkio/foo/tasks
```

注1:45033为dd的进程号

注2: 8:0对应主设备号和副设备号，可以通过ls -l /dev/sda查看

```
[plain]
1. [root@localhost ~]# ls -l /dev/sda
2. brw-rw----. 1 root disk 8, 0 Sep 15 04:19 /dev/sda
```

9 cgroup小结

使用cgroup临时对进程进行调整，直接通过命令即可，如果要持久化对进程进行控制，即重启后依然有效，需要写进配置文件/etc/cgconfig.conf及/etc/cgrules.conf