**linux下查询进程占用的内存方法总结**

假设现在有一个「php-cgi」的进程 ，进程id为「25282」。现在想要查询该进程占用的内存大小。linux命令行  
下有很多的工具进行查看，现总结常见的几种方式：

**通过进程的 status**

[root@web3\_u ~]# cat /proc/25282/status

Name: php-cgi

State: S (sleeping)

Tgid: 25282

Pid: 25282

PPid: 27187

TracerPid: 0

Uid: 99 99 99 99

Gid: 99 99 99 99

Utrace: 0

FDSize: 256

Groups: 99

VmPeak: 496388 kB

VmSize: 438284 kB

VmLck: 0 kB

VmHWM: 125468 kB

VmRSS: 113612 kB

VmData: 92588 kB

VmStk: 100 kB

VmExe: 6736 kB

VmLib: 18760 kB

VmPTE: 528 kB

VmSwap: 0 kB

Threads: 1

SigQ: 0/46155

SigPnd: 0000000000000000

ShdPnd: 0000000000000000

SigBlk: 0000000000000000

SigIgn: 0000000000001000

SigCgt: 0000000184000004

CapInh: 0000000000000000

CapPrm: 0000000000000000

CapEff: 0000000000000000

CapBnd: ffffffffffffffff

Cpus\_allowed: f

Cpus\_allowed\_list: 0-3

Mems\_allowed: 00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000000,00000001

Mems\_allowed\_list: 0

voluntary\_ctxt\_switches: 68245

nonvoluntary\_ctxt\_switches: 15751

VmRSS: 113612 kB 表示占用的物理内存

**通过 pmap**

[root@web3\_u ~]# pmap -x 25282

25282: /usr/local/php/bin/php-cgi --fpm --fpm-config /usr/local/php/etc/php-fpm.conf

Address Kbytes RSS Dirty Mode Mapping

0000000000400000 6736 2692 0 r-x-- php-cgi

0000000000c93000 264 196 120 rw--- php-cgi

0000000000cd5000 60 48 48 rw--- [ anon ]

. . .

00007fd6226bc000 4 4 4 rw--- ld-2.12.so

00007fd6226bd000 4 4 4 rw--- [ anon ]

00007fff84b02000 96 96 96 rw--- [ stack ]

00007fff84bff000 4 4 0 r-x-- [ anon ]

ffffffffff600000 4 0 0 r-x-- [ anon ]

---------------- ------ ------ ------

total kB 438284 113612 107960

关键信息点

* 进程ID
* 启动命令「/usr/local/php/bin/php-cgi --fpm --fpm-config /usr/local/php/etc/php-fpm.conf」
* RSS :占用的物理内存 113612KB

**通过 smaps**

[root@web3\_u ~]# cat /proc/25282/smaps | grep '^Rss:' \

| awk '{sum +=$2} END{print sum}'

113612

求和得到实际占用物理内存为 113612

**通过 ps 命令**

[root@web3\_u ~]# ps -e -o 'pid,comm,args,pcpu,rsz,vsz,stime,user,uid' \

| awk '$1 ~ /25282/'

25282 php-cgi /usr/local/php/bin/php-cgi 0.0 113612 438284 Oct09 nobody 99

awk 过滤 25282 进程号，得到第5列「rsz」的内存大小为「113612」

输出php-cgi进程占用的物理内存，并从高到低进行排序

[root@web3\_u ~]# ps -e -o 'pid,comm,args,pcpu,rsz,vsz,stime,user,uid' \

| grep php-cgi | sort -k5nr

输出结果

23946 php-cgi /usr/local/php/bin/php-cgi 0.0 129540 440000 Oct06 nobody 99

24418 php-cgi /usr/local/php/bin/php-cgi 0.0 129336 437684 Oct06 nobody 99

18973 php-cgi /usr/local/php/bin/php-cgi 0.0 129268 440176 Oct06 nobody 99

17219 php-cgi /usr/local/php/bin/php-cgi 0.0 126588 439840 Oct06 nobody 99

6996 php-cgi /usr/local/php/bin/php-cgi 0.0 124876 438104 Oct09 nobody 99

23850 php-cgi /usr/local/php/bin/php-cgi 0.0 122984 440036 Oct09 nobody 99

28310 php-cgi /usr/local/php/bin/php-cgi 0.0 122920 436456 Oct09 nobody 99

其中rsz为实际内存，上例实现按内存排序，由大到小

TOP 命令输出的列

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

25282 nobody 20 0 428m 110m 93m S 0.0 1.9 0:34.42 php-cgi

输出列信息

* PID 25282
* 用户 nobody
* 虚拟内存 428M
* 物理内存 110M 110\*1024= 112640 「和前面计算出来的值基本一致」
* 共享内存 93M
* 进程使用的物理内存和总内存的百分比 1.9 %

PID：进程的ID

USER：进程所有者

PR：进程的优先级别，越小越优先被执行

NInice：值

VIRT：进程占用的虚拟内存

RES：进程占用的物理内存

SHR：进程使用的共享内存

S：进程的状态。S表示休眠，R表示正在运行，Z表示僵死状态，N表示该进程优先值为负数

%CPU：进程占用CPU的使用率

%MEM：进程使用的物理内存和总内存的百分比

TIME+：该进程启动后占用的总的CPU时间，即占用CPU使用时间的累加值。

COMMAND：进程启动命令名称

按P

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

16036 root 20 0 8902m 8.6g 480 R 100.0 36.6 0:33.15 redis-server

12934 root 20 0 8902m 8.6g 1072 S 5.5 36.6 285:37.81 redis-server

969 root 20 0 0 0 0 D 4.2 0.0 277:14.85 flush-252:16

1304 root 23 3 1689m 50m 3264 S 4.2 0.2 1445:03 xs-searchd

1294 root 20 0 14928 928 584 S 3.5 0.0 635:05.31 xs-indexd

1287 nobody 20 0 12884 772 576 S 2.8 0.0 833:11.42 dnsmasq

1302 root 23 3 1113m 39m 3244 S 0.7 0.2 1437:57 xs-searchd

4444 www 20 0 280m 43m 884 S 0.7 0.2 27:43.92 nginx

1 root 20 0 19232 1160 868 S 0.0 0.0 0:06.75 init

按 P .表示按cpu排序，默认也是按cpu排序

按M

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

12934 root 20 0 8902m 8.6g 1072 S 6.0 36.6 285:39.77 redis-server

16036 root 20 0 8902m 8.6g 480 R 100.0 36.6 1:11.42 redis-server

1236 www 20 0 1053m 209m 6556 S 0.0 0.9 4:40.70 php-cgi

1231 www 20 0 1034m 146m 6536 S 0.0 0.6 4:20.82 php-cgi

1184 www 20 0 1043m 119m 6584 S 0.0 0.5 4:21.85 php-cgi

按M 。 表示按占用内存排序。 第一列 redis服务器占用了8.6G的内存 。 这个内存和redis info

[root@img1\_u ~]# redis-cli info memory

# Memory

used\_memory\_human:8.32G

基本相同。

[root@img1\_u ~]# top -u www

top - 22:09:01 up 67 days, 14:16, 1 user, load average: 0.61, 0.90, 0.98

Tasks: 283 total, 2 running, 281 sleeping, 0 stopped, 0 zombie

Cpu(s): 3.9%us, 1.0%sy, 0.5%ni, 89.7%id, 4.6%wa, 0.0%hi, 0.3%si, 0.0%st

Mem: 24542176k total, 21130060k used, 3412116k free, 1750652k buffers

Swap: 524280k total, 0k used, 524280k free, 4039732k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

681 www 20 0 855m 25m 5796 S 0.0 0.1 0:47.00 php-cgi

1181 www 20 0 887m 57m 6484 S 0.0 0.2 4:41.66 php-cgi

1183 www 20 0 864m 34m 6320 S 0.0 0.1 3:52.39 php-cgi

1184 www 20 0 1043m 119m 6584 S 0.0 0.5 4:21.85 php-cgi

1185 www 20 0 869m 39m 6376 S 0.0 0.2 3:57.84 php-cgi

1186 www 20 0 886m 56m 6244 S 0.0 0.2 3:44.75 php-cgi

1187 www 20 0 926m 66m 6480 S 0.0 0.3 4:16.12 php-cgi

1188 www 20 0 890m 60m 6288 S 0.0 0.3 4:13.35 php-cgi

1189 www 20 0 892m 62m 6408 S 0.0 0.3 4:06.60 php-cgi

-u 指定用户。 php-cgi占用的内存在60M左右

**按进程消耗内存多少排序的方法**

**通过 ps 命令**

第一种方法

ps -e -o 'pid,comm,args,pcpu,rsz,vsz,stime,user,uid' | sort -k5nr

第二种方法

ps -e -o 'pid,comm,args,pcpu,rsz,vsz,stime,user,uid' --sort -rsz

输出结果

[root@web3\_u ~]# ps -e -o 'pid,comm,args,pcpu,rsz,vsz,stime,user' | sort -k5nr

23946 php-cgi /usr/local/php/bin/php-cgi 0.0 129540 440000 Oct06 nobody

24418 php-cgi /usr/local/php/bin/php-cgi 0.0 129336 437684 Oct06 nobody

18973 php-cgi /usr/local/php/bin/php-cgi 0.0 129268 440176 Oct06 nobody

17219 php-cgi /usr/local/php/bin/php-cgi 0.0 126588 439840 Oct06 nobody

6996 php-cgi /usr/local/php/bin/php-cgi 0.0 125056 438104 Oct09 nobody

23850 php-cgi /usr/local/php/bin/php-cgi 0.0 122984 440036 Oct09 nobody

参数解析:

* -e 显示所有进程
* -o 定制显示信息
* pid 进程ID
* comm 进程名
* args 启动命令
* pcpu 占用CPU 百分比
* rsz 占用物理内存大小
* vsz 占用虚拟内存大小
* stime 进程启动时间
* user 启动用户

以第一行为例

* 进程ID 23946
* 进程名 php-cgi
* 启动命令 /usr/local/php/bin/php-cgi
* 占用CPU 0
* 占用物理内存 129540
* 占用虚拟内存 440000
* 启动时间 Oct06
* 启动用户 nobody

ps 命令

**通过 top 命令**

top命令默认是以CPU排序输出的，按字母「M」，可以按内存占用大小进行排序显示

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

23946 nobody 20 0 429m 126m 107m S 0.0 2.2 1:15.01 php-cgi

24418 nobody 20 0 427m 126m 109m S 0.0 2.2 1:19.56 php-cgi

18973 nobody 20 0 429m 126m 107m S 0.0 2.2 1:20.18 php-cgi

17219 nobody 20 0 429m 123m 104m S 0.0 2.1 1:23.60 php-cgi

6996 nobody 20 0 427m 122m 105m S 0.0 2.1 1:05.27 php-cgi

23850 nobody 20 0 429m 120m 101m S 0.0 2.1 1:02.43 php-cgi

输出参数介绍

* PID：进程的ID
* USER：进程所有者
* VIRT：进程占用的虚拟内存
* RES：进程占用的物理内存
* SHR：进程使用的共享内存
* S：进程的状态。S表示休眠，R表示正在运行，Z表示僵死状态，N表示该进程优先值为负数
* %CPU：进程占用CPU的使用率
* %MEM：进程使用的物理内存和总内存的百分比
* TIME+：该进程启动后占用的总的CPU时间，即占用CPU使用时间的累加值。

通过比较进程ID 「23946」，top 命令和 ps 命令输出的结果基本保持一致