

系统 DEBUG 与性能优化工具

Author **Darren**

Rivision **V0.1**

Date **2017/12/28**



Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.

Office 501, Building 13, No.99, Tianzhou Road, Shanghai, China, 200233

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local office. For more information, please visit:

<http://www.quectel.com/support/salesupport.aspx>

For technical support, or to report documentation errors, please visit:

<http://www.quectel.com/support/techsupport.aspx>

Or Email to: Support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Quectel Wireless Solutions Co., Ltd. 2017. All rights reserved.

About the Document

History

Revision	Date	Author	Description
0.1	December 28,2017	Darren	Initial

目 录

About The Document	2
Content	3
1 模块 DEBUG 工具使用	4
1.1 dmesg 工具.....	4
1.2 dump 工具	4
1.3 top 工具	4
1.4 mpstat cpu 使用率分析	6
1.5 进程文件描述符分析	6
1.6 内存分析.....	7

第 1 章 模块 DEBUG 工具使用

移远模块支持多种 debug 手段，与性能优化工具，具体如下

1.1 dmesg 工具

dmesg 是一种可以操作内核日志缓冲区的工具，它支持设置内核 log 等级，内核 log 缓冲区大小等功能

```
dmesg:
-c: 当打印完成后，直接清除内核缓冲区内容
-s: 设置内核缓冲区大小(dmesg -s 4096)
-n: 设置内核打印级别1-7, 1的级别最高，打印的内容也最少

使用示例:
root@mdm9607-perf:~# dmesg
```

1.2 dump 工具

dump 工具是当程序发生异常时，linux 会产生 core file 文件供开发者分析

```
core打开步骤如下:
1. ulimit -c 1024 解除 core file 的限制,此时 core file 最多可生成 1024 个
2. sysctl -w "kernel.core_pattern=/data/core_%e_%t.%p" (e:程序名 t:发生 dump 时间 p:pid)
3. kill -s SIGSEGV $$, 这个指令用来测试 core file 是否在指定目录生成

使用示例:
当打开core dump功能后，在程序发生异常退出时，可以去/data目录找到对应的core file文件，然后使用gdb进行分析
```

1.3 top 工具

top 工具可以观察进程 CPU 占用率，CPU 负载，进程的内存相关信息，在使用 top 对 CPU 的利用率的计算是建立在 proc stat 基础上的

```
/proc/${pid}/stat:
这个是进程状态信息，通过它就可以分析进程的系统资源占用情况（包括cpu, memory, io, interrupt等）

root@mdm9607-perf:/bin# cat /proc/1245/stat
1245 (atfwd_daemon) S 1 1243 1243 0 -1 1077936384 845 9460 34 2 136 541 2 2 20 0 10 0 1431
85893120 416 4294967295 32768 374084 3201240608 3201239984 1150533868 0 0 0 18951 4294967295
0 0 17 0 0 0 0 0 378184 439268 454656 3201240855 3201240877 3201240877 3201241062 0

参数解释:
pid=1245 tcomm='atfwd_daemon' state='S' ppid=1 pgid=1243 sid=1243 tty=0 tty_pgrp=-1
flag=1077936384(0x40400100) minflt=845 cminflt=9460 majflt=34 cmajflt=2
utime=136 stime=541 cutime=2 cstime=2 priority=20 nice=0 num_threads=10 0(固定)
start_time=1431 vss=85893120(字节) rss=416(page) rlim=4294967295(字节) scode=32768(0x8000)
ecode=374084(0x5b544) sstack=3201240608(0xbecf0e20) esp=3201239984(0xbecf0bb0) eip=1150533868
```

```
pendsig=0 blocksig=0 ignsig=0 catchsig=18951 wchan=4294967295 0(固定) 0(固定) esig=17 cpu=0
rt_pri=0 policy=0 0 0 0(cpu time) seg_data[378184, 439268] sbrk=454656 arg=[3201240855, 3201240877]
env=[3201240877, 3201241062]
```

tty: 进程终端
flag: fork进程时的flag参数
min_flt: 本进程缺页次数(内存分配)
cmin_flt: 本进程加所有子程序的缺页次数
maj_flt: 读磁盘产生的缺页次数
cmaj_flt: 本进程加所有子程序的读磁盘缺页次数
utime: 进程应用空间占用时间(以jiffies为单位)
stime: 进程内核空间占用时间(以jiffies为单位)

```
/proc/stat:
root@mdm9607-perf:/bin# cat /proc/stat
cpu 980 0 5343 1347896 5 0 25 0 0 0
cpu0 980 0 5343 1347896 5 0 25 0 0 0
ctxt 986078(上下文切换次数)
btime 315964802(系统bootime, 单位秒)
processes 1477(系统总共fork进程数)
procs_running 1(当前runing状态进程数量)
procs_blocked 0(当前阻塞进程)
softirq 1439782 8 1383450 1383 550 0 0 1232 0 0 53159
```

utime=980(用户空间占用的时间, jiffies为单位)
nice=0(rt进程占用时间)
sys=5343(内核空间占用的时间)
idle=1347896(idle进程占用时间)
iowait=5(io wait时间)
irq=0(硬中断占用时间)
sirq=25(软中断占用时间)
后面与虚拟机相关, 再下面一行cpu0是具体到核心, 是给SMP使用的

top 工具在计算每个进程的 CPU 利用率时, 会去采样/proc/\$pid/stat 里的 utime 与 stime, 然后利用在一段固定的时间段内, 计算它占用 CPU 的总时间与固定时间的比值, 就得出 CPU 的利用率。CPU 总体利用率是使用 IDLE 进程与其它进程的比值决定的。

```
Mem: 107680K used, 48428K free, 144K shrd, 40K buff, 54292K cached
CPU:  0.0% usr  0.2% sys  0.0% nic 99.7% idle  0.0% io  0.0% irq  0.0% sirq
Load average: 1.00 1.01 1.05 1/305 1531
  PID  PPID  USER      STAT  VSZ %VSZ CPU %CPU COMMAND
   893     1 www-data  S    4440  2.8   0  0.0 /usr/sbin/lighttpd -f /etc/lighttpd
  1531  1427  root      R    3088  1.9   0  0.0 top
   623     1 root      S    90280 57.5   0  0.0 /usr/bin/quectel_daemon
  1245     1 root      S    83880 53.5   0  0.0 /usr/bin/atfwd_daemon
   747     1 root      S    14996  9.5   0  0.0 alsaucm_test
     7     2 root      SW          0  0.0   0  0.0 [rcu_preempt]
   569     1 root      S    200m131.0  0  0.0 /usr/bin/netmgrd
   565     1 root      S <    192m125.8  0  0.0 /usr/bin/thermal-engine
   400     1 root      S    114m 74.9   0  0.0 QCMAP_ConnectionManager /etc/mobil
  1302     1 root      S    114m 74.5   0  0.0 /usr/bin/ql_manager_server
   288     1 root      S    63576 40.5   0  0.0 psmd
   910     1 root      S    63300 40.3   0  0.0 /usr/bin/qmi_ip_multiclient /etc/q
```

在top的显示中, 第二行的CPU是指的CPU总体的利用率, usr表示用户空间占用率, sys表示内核空间占用率, io表示磁盘读写时占用cpu率, irq指的中断占用cpu率。Load average指的是最近1分钟, 5分钟与15分钟的CPU的负载率。%CPU是指的单个进程的CPU利用率。

1.4 mpstat cpu 使用率分析

这是与 top 类似的工具，它是针对系统整体进行分析：

```
root@mdm9607-perf:/proc/1245# mpstat 5 5
Linux 3.18.20 (mdm9607-perf) 01/06/80 _armv7l_ (1 CPU)

06:08:50   CPU   %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
06:08:55   all    0.00    0.00    0.40    0.00    0.00    0.00    0.00    0.00   99.60

06:08:55   CPU   %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
06:09:00   all    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00

06:09:00   CPU   %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
06:09:05   all    0.00    0.00    0.61    0.00    0.00    0.00    0.00    0.00   99.39

06:09:05   CPU   %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
06:09:10   all    0.00    0.00    0.21    0.00    0.00    0.00    0.00    0.00   99.79

06:09:10   CPU   %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
06:09:15   all    0.00    0.00    0.41    0.00    0.00    0.00    0.00    0.00   99.59

Average:   CPU   %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
Average:   all    0.00    0.00    0.33    0.00    0.00    0.00    0.00    0.00   99.67

mpstat显示的所有CPU的整体利用率，其中包括用户空间，内核空间，中断，io等，对系统CPU性能上能一目了然
```

1.5 进程文件描述符分析

进程文件描述符分析有两个重要的量，一是系统对进程的文件描述符限制，二是进程自己对自己的限制。分别可以通过 ulimit -a 查看与 cat /proc/\$pid/limits 查看，取最小值。

```
root@mdm9607-perf:/proc/1245# ulimit -a
-f: file size (blocks)             unlimited
-t: cpu time (seconds)             unlimited
-d: data seg size (kb)             unlimited
-s: stack size (kb)               8192
-c: core file size (blocks)        0
-m: resident set size (kb)         unlimited
-l: locked memory (kb)            64
-p: processes                     1148
-n: file descriptors              1024
-v: address space (kb)            unlimited
-w: locks                         unlimited
-e: scheduling priority            0
-r: real-time priority            0
上面的-n就是系统对进程的文件描述符的限制

root@mdm9607-perf:/proc/1245# cat limits
Limit                Soft Limit             Hard Limit             Units
Max open files       1024                   4096                   files
上面的Max open files就是进程本身的文件描述符限制

root@mdm9607-perf:/proc/1245# ls fd/ | wc -w
20
通过上面的命令计算每个进程的文件描述符使用数量
```

内存分析有总共内存分析与局部内存分析，下面是单个进程的内存分析：

```
root@mcmdm9607-perf:/proc/880# cat status
Name: eMBMs_Tunneling //程序名称
State: S (sleeping) //程序当前状态
Tgid: 880 //进程组ID，与组长进程的PID相同
Pid: 880 //进程id，系统内唯一
PPid: 1 //父进程id
Uid: 0 0 0 0 //RUID(实际) EUID(有效) SUID(Saved ID) FSUID(文件系统的用户ID)
Gid: 0 0 0 0 //RGUID EGUID SGUID FSGUID
FDSize: 32 //当前分配的描述符（当使用的文件描述符超过32时，FDSize将会以32为单位递增）
Groups: //进程所在的用组
VmPeak: 13756 kB //进程占用的虚拟内存大小最大值
VmSize: 13756 kB //进程当前占用的虚拟内存大小
Vmlck: 0 kB //进程锁住的物理内存大小，不能释放，不能交换
VmPin: 0 kB
VmHWM: 1304 kB //进程占用物理内存的峰值
VmRSS: 1304 kB //进程当前占用的物理内存（这里共享内存不平均，所以会偏大，PSS才是准确的）
VmData: 9516 kB //进程数据段占用虚拟内存大小
VmStk: 136 kB //用户栈占用虚拟内存大小
VmExe: 16 kB //进程代码段占用虚拟内存大小，不计算动态库
VmLib: 3908 kB //被映射到进程虚拟地址空间的动态库大小
VmPTE: 12 kB //进程所需页表占用物理内存大小
VmSwap: 0 kB
Threads: 2 //进程拥有的线程数
SigQ: 5/1175 //当前待处理信息个数
SigPnd: 0000000000000000 //该线程的待处理信号
ShdPnd: 0000000000000000 //该线程组的待处理信号
SigBlk: 0000000000000006 //mask位，被阻塞的信号，信号2，3被阻塞
SigIgn: 0000000000000010 //mask位，被设置为忽略的信号，信号16被忽略
SigCgt: 00000000280b2603 //存放被捕捉的信号，具体标识如下
00000000280b2603 ==> 101000000010110010011000000011
      | |         | || |   |    |\`-> 1 = SIGHUP
      | |         | || |   |    |--> 2 = SIGINT
      | |         | || |   |\`-----> 10 = SIGUSR1
      | |         | || |   \`-----> 11 = SIGSEGV
      | |         | || `-----> 14 = SIGALRM
      | |         | \|`-----> 17 = SIGCHLD
      | |         | `-----> 18 = SIGCONT
      | |         `-----> 20 = SIGTSTP
      | \|`-----> 28 = SIGWINCH
      `-----> 30 = SIGPWR

Cpus_allowed: 1 //CPU亲和性掩码
Cpus_allowed_list: 0 //进程可以使用CPU的列表，0表示只能使用CPU0，0-2表示可以使用0，1，2CP
voluntary_ctxt_switches: 4 //表示进程主动切换的次数（用于分析系统性能）
nonvoluntary ctxt switches: 1 //表示进程被动切换的次数
```

内存总体分析如下:

```
root@mdm9607-perf:~# cat /proc/meminfo
MemTotal:        43952 kB //伙伴系统总共可用内存
MemFree:          5280 kB //伙伴系统中未被分配的内存
MemAvailable:     8356 kB //当前系统可用内存
Buffers:          0 kB //block buffer
Cached:           2504 kB //系统缓冲(文件与共享内存)
```