2023.8.17 ANP性能优化：绑核调研

目录

一、背景

1. 中断

由于接收来自外围硬件(相对于CPU和内存)的异步信号或者来自软件的同步信号，而进行相应的硬件、软件处理；发出这样的信号称为进行中断请求(interrupt request, IRQ)

- 硬中断：外围硬件发给CPU或者内存的异步信号就称之为硬中断
- 软中断：由软件系统本身发给操作系统内核的中断信号，称之为软中断。通常是由硬中断处理程序或进程调度程序对操作系统 内核的中断，也就是我们常说的系统调用(System Call)

硬中断通过设置CPU的屏蔽位可进行屏蔽，软中断则由于是指令之方式给出，不能屏蔽。

硬中断发生后，通常会在硬中断处理程序中调用一个软中断来进行后续工作的处理

硬中断和软中断均会引起上下文切换(进程/线程之切换)，进程切换的过程是差不多的。

2. 绑核

中断、进程/线程均可以进行绑核。

进程/线程绑核，其实就是设定某个进程/线程与某个CPU核的亲和力（affinity）。设定以后，Linux调度器就会让这个进程/线程只在所绑定的核上面去运行。但并不是说该进程/线程就独占这个CPU的核，其他的进程/线程还是可以在这个核上面运行的。如果想要实现某个进程/线程独占某个核，就要使用cpuset命令去实现。其实，很多情况下，为了提高性能，Linux调度器会自动实现尽量让某个进程/线程在同样的CPU上去运行。所以，除非必须，我们没有必要显式的去进行进程绑核操作。

3. 查看各个cpu的中断情况：cat /proc/stat

参考https://man7.org/linux/man-pages/man5/proc.5.html

第一行代表自系统启动开始累计到当前时刻，所有 CPU 的总和，而第二行开始表示每个 CPU 核心的使用情况信息：user ， nice，system，idle，iowait， irq，softirq，steal，guest， guest_nice

所以，top 命令的 CPU 使用率计算公式如下：CPU总时间 = user + nice + system + idle + wait + irq + softirq + steal +guest

jiffies是内核中的一个全局变量，用来记录自系统启动一来产生的节拍数，在linux中，一个节拍大致可理解为操作系统进程调度的最小时间片，不同linux内核可能值有不同，通常在1ms到10ms之间。

%us = user / CPU总时间， 用户态的cpu时间（单位：jiffies）,不包含nice值为负的进程。1jiffies=0.01秒。

%ni = nice / CPU总时间，nice值为负的进程所占用的cpu时间（单位：jiffies）。

%sy = system / CPU总时间，系统态的cpu时间(单位：jiffies)。

%id = idel / CPU总时间，除硬盘io等待时间以外的其他等待时间（单位：jiffies）。（不准确）

%wa = wait / CPU总时间，硬盘io等待时间(单位：jiffies)。

%hi = irq / CPU总时间，硬中断时间(单位ie：jiffies)。

%si = softirq / CPU总时间，软中断时间(单位：jiffies)。

%st = steal / CPU总时间

intr： 中断的信息，第一个为自系统启动以来，发生的所有的中断的次数，后每个数对应一个特定的中断自系统启动以来发生的中断次数。

ctxt：cpu发生的上下文转换的次数。

btime： boot time, in **seconds** since the Epoch, 1970-01-01 00:00:00 +0000 (UTC).

processes：Number of forks since boot. 创建的进程总数。

procs_running：Number of processes in runnable state.

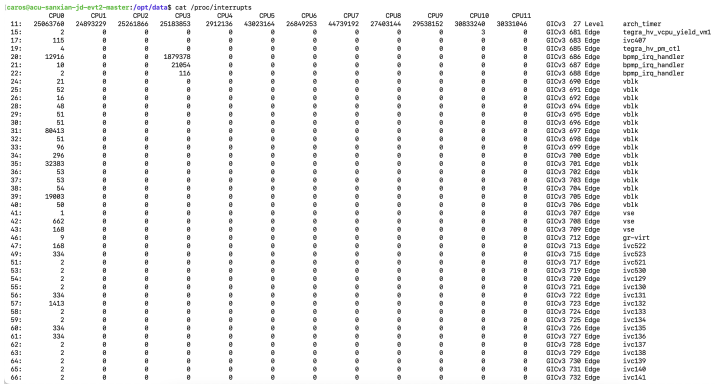
procs_blocked：Number of processes blocked waiting for I/O to complete.

softirq：This line shows the number of softirq for all CPUs. The first column is the total of all softirqs and each subsequent column is the total for particular softirq.

4. 中断的绑核情况：cat /proc/interrupts

https://blog.csdn.net/m0_57982541/article/details/124274815

字段依次是：逻辑中断号、中断在各CPU上发生的次数，**中断所属父设备名称、硬件中断号、中断触发方式(电平或边沿)、中断名称。**



```
(caros@acu-sanxian-jd-evt2-master:/opt/data$ cat /proc/interrupts
           CPU0       CPU1       CPU2       CPU3       CPU4       CPU5       CPU6       CPU7       CPU8       CPU9       CPU10      CPU11
  11:   2506376O   24893229   25261866   25183863    2912136   43023164   26849253   44739192   27403144   29538152   30833240   30331046     GICv3  27 Level     arch_timer
  15:          2          0          0          0          0          0          0          0          0          0          3          0     GICv3 681 Edge      tegra_hv_vcpu_yield_vm1
  17:        115          0          0          0          0          0          0          0          0          0          0          0     GICv3 683 Edge      ivc407
  19:          4          0          0          0          0          0          0          0          0          0          0          0     GICv3 685 Edge      tegra_hv_pm_ctl
  20:      12916          0          0    1879378          0          0          0          0          0          0          0          0     GICv3 686 Edge      bpmp_irq_handler
  21:         10          0          0      21054          0          0          0          0          0          0          0          0     GICv3 687 Edge      bpmp_irq_handler
  22:          2          0          0        116          0          0          0          0          0          0          0          0     GICv3 688 Edge      bpmp_irq_handler
  24:         21          0          0          0          0          0          0          0          0          0          0          0     GICv3 690 Edge      vblk
  25:         52          0          0          0          0          0          0          0          0          0          0          0     GICv3 691 Edge      vblk
  26:         16          0          0          0          0          0          0          0          0          0          0          0     GICv3 692 Edge      vblk
  28:         48          0          0          0          0          0          0          0          0          0          0          0     GICv3 694 Edge      vblk
  29:         51          0          0          0          0          0          0          0          0          0          0          0     GICv3 695 Edge      vblk
  30:         51          0          0          0          0          0          0          0          0          0          0          0     GICv3 696 Edge      vblk
  31:      80413          0          0          0          0          0          0          0          0          0          0          0     GICv3 697 Edge      vblk
  32:         51          0          0          0          0          0          0          0          0          0          0          0     GICv3 698 Edge      vblk
  33:         96          0          0          0          0          0          0          0          0          0          0          0     GICv3 699 Edge      vblk
  34:        296          0          0          0          0          0          0          0          0          0          0          0     GICv3 700 Edge      vblk
  35:      32383          0          0          0          0          0          0          0          0          0          0          0     GICv3 701 Edge      vblk
  36:         53          0          0          0          0          0          0          0          0          0          0          0     GICv3 702 Edge      vblk
  37:         53          0          0          0          0          0          0          0          0          0          0          0     GICv3 703 Edge      vblk
  38:         84          0          0          0          0          0          0          0          0          0          0          0     GICv3 704 Edge      vblk
  39:      19083          0          0          0          0          0          0          0          0          0          0          0     GICv3 705 Edge      vblk
  40:         50          0          0          0          0          0          0          0          0          0          0          0     GICv3 706 Edge      vblk
  41:          1          0          0          0          0          0          0          0          0          0          0          0     GICv3 707 Edge      vse
  42:        662          0          0          0          0          0          0          0          0          0          0          0     GICv3 708 Edge      vse
  43:        168          0          0          0          0          0          0          0          0          0          0          0     GICv3 709 Edge      vse
  46:          9          0          0          0          0          0          0          0          0          0          0          0     GICv3 712 Edge      gr-virt
  47:        168          0          0          0          0          0          0          0          0          0          0          0     GICv3 713 Edge      ivc522
  49:        334          0          0          0          0          0          0          0          0          0          0          0     GICv3 715 Edge      ivc523
  51:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 717 Edge      ivc521
  53:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 719 Edge      ivc530
  54:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 720 Edge      ivc129
  55:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 721 Edge      ivc130
  56:        334          0          0          0          0          0          0          0          0          0          0          0     GICv3 722 Edge      ivc131
  57:       1413          0          0          0          0          0          0          0          0          0          0          0     GICv3 723 Edge      ivc132
  58:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 724 Edge      ivc133
  59:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 725 Edge      ivc134
  60:        334          0          0          0          0          0          0          0          0          0          0          0     GICv3 726 Edge      ivc135
  61:        334          0          0          0          0          0          0          0          0          0          0          0     GICv3 727 Edge      ivc136
  62:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 728 Edge      ivc137
  63:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 729 Edge      ivc138
  64:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 730 Edge      ivc139
  65:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 731 Edge      ivc140
  66:          2          0          0          0          0          0          0          0          0          0          0          0     GICv3 732 Edge      ivc141
```

/proc/irq/{IRQ}/smp_affinity和/proc/irq/{IRQ}/smp_affinity_list指定了哪些CPU能够关联到一个给定的IRQ源. 这两个文件包含了这些指定cpu的cpu位掩码(smp_affinity)和cpu列表(smp_affinity_list)

- smp_affinity：修改该文件中的值可以改变CPU和某中断的亲和性
- **smp_affinity和smp_affinity_list修改其一即可**

每个IRQ的默认的smp affinity在这里：/proc/irq/default_smp_affinity (fff)，

如何把中断号绑定到CPU的某个核上?

需要设置多个CPU核心时可使用：

//设置1-6core，把中断号23分配给

echo 1-6 >/proc/irq/23/smp_affinity_list

echo 7e >/proc/irq/23/smp_affinity # 01111110

//设置core1,core6

echo 1,6> /proc/irq/23/smp_affinity_list

echo 42> /proc/irq/23/smp_affini # 01000010

不绑10核，二进制1011, 1111, 1111 ，十六进制BFF

echo 0,1,2,3,4,5,6,7,8,9,11 > /proc/irq/?/smp_affinity_list

echo BFF> /proc/irq/?/smp_affini

5. 进程的绑核情况

通过 taskset 命令可将某个进程与某个CPU核心绑定，使得其仅在与之绑定的CPU核心上运行。部署位置/usr/bin/taskset

```
root@acu-sanxian-jd-evt2-slave:/opt/data/hph# taskset --help
Usage: taskset [options] [mask | cpu-list] [pid|cmd [args...]]


Show or change the CPU affinity of a process.

Options:
 -a, --all-tasks         operate on all the tasks (threads) for a given pid
 -p, --pid               operate on existing given pid
 -c, --cpu-list          display and specify cpus in list format
 -h, --help              display this help
 -V, --version           display version

The default behavior is to run a new command:
    taskset 03 sshd -b 1024
You can retrieve the mask of an existing task:
    taskset -p 700
Or set it:
    taskset -p 03 700
List format uses a comma-separated list instead of a mask:
    taskset -pc 0,3,7-11 700
Ranges in list format can take a stride argument:
    e.g. 0-31:2 is equivalent to mask 0x55555555
```

查看绑核情况：taskset -pc 8357 以列表形式显示进程在哪几块cpu上运行，taskset -p 8357 以二进制形式显示进程在哪几块cpu上运行，

```
caros@acu-sanxian-jd-evt2-slave:/opt/data/hph$ ps -aux|grep mainboard
caros     460109  0.0  0.0   4728   616 pts/5    S+   06:39   0:00 grep --color=auto mainboard
caros    3756819 55.6 14.5 36598888 4200012 pts/0 Sl+ 04:22  76:42 mainboard -p compute2d_pnc_sched -d dynamic_hdmap_master.dag -d orin_imu_gps.dag -d orin_rtcm_parser.dag -d dag_streaming_local_anp_fusion.dag -d planning.dag -d relative_map.dag -d prediction_navi.dag -d control_navi.dag -d link-radar.dag -d adas_orin_exclusive.dag -d status_machine.dag -d perfanp_slave_orinx.dag -d computron.dag -d scene_recognizer.dag -d metric.dag
caros    3756824 24.0 12.4 36390116 3609936 pts/0 Sl+ 04:22  33:10 mainboard -p compute2d_perception_sched -d dynamic_hdmap_slave_tl.dag -d camera_bev_orinx_slave.dag -d perception_avp_slave.dag -d perception_bev_orinx_slave.dag -d tl_perception_trafficlight_split_orinx_slave.dag -d scene_recognizer.dag -d metric.dag
caros    3756826 14.6  0.7 1401052 220420 pts/0  Sl+  04:22  20:13 mainboard -p patrol_slave_orinx_sched -d patrol_slave_orinx.dag -d metric.dag
caros@acu-sanxian-jd-evt2-slave:/opt/data/hph$ taskset -pc 3756819
pid 3756819's current affinity list: 0-3,5-11
caros@acu-sanxian-jd-evt2-slave:/opt/data/hph$ taskset -pc 3756824
pid 3756824's current affinity list: 0-3,5-11
caros@acu-sanxian-jd-evt2-slave:/opt/data/hph$ taskset -pc 3756826
pid 3756826's current affinity list: 0-3,5-11
```

类似于中断绑核，绑核有两种方法：

- 掩码形式：十六进制。将掩码转换为二进制形式，从最低位到最高位代表物理CPU的#0、#1、……、#n号核。某位的值为0表示不绑该核，1表示绑。比如：0x00000001的二进制为0000...0001，只有第0号核的位置是1，所以表示只绑0号核；0x00000003的二进制为0000...0011，第0和1号核的位置是1，所以表示绑CPU的0号和1号核；再比如0xFFFFFFFF的二进制为1111...1111，所有32个核的位置都为1，所以表示绑CPU的0~31核。需要注意的是，并非掩码中给出的CPU核就一定会存在，比如0x00000400理论上代表CPU的第10号核，但是该核在真正的计算机上面并不一定是存在的。而且，如果我们试图将物理上并不存的核绑定给某个进程时，会返回错误。掩码形式的绑核命令为 taskset -p mask pid
- 列表形式：列表形式指直接指定要绑的CPU核的列表，列表中可以有一个或多个核。具体语法如下：taskset -cp cpu-list pid，其中cpu-list是数字化的cpu列表，从0开始。多个不连续的cpu可用逗号连接，连续的可用短现连接，比如0,2,5-11等。比如taskset -cp 0,2,5-11 9865命令表示将进程9865绑定到#0、#2、#5~#11号核上面。

二、

DriveOS	driveos/orin_baidu_sanxian/pdk_6.0.5.0/baidu-acu-s0-ap_signed/642-63663-0001-001_TS2/flash-images/A_1_6_gos0-fs_targetfs.img	
Gaia	gaia/linux-aarch64/etc/rc_board_id_3007.local gaia/linux-aarch64/etc/rc_board_id_3003.local gaia/linux-aarch64/etc/rc_board_id_3009.local gaia/linux-aarch64/etc/rc_board_id_3005.local	</> Bash #!/bin/bash ## Pull down FOTA OE GPIO. index = 492 echo 492 > /sys/class/gpio/export echo 0 > "/sys/class/gpio/PAC.06/value" function irq_cpu_bound() { irq_name=$1 cpu_list=$2

```
    irq_number=$(cat /proc/interrupts | grep ${irq_name} | cut -d: -f1 | sed s/[[:space:]]//g)
    for i in $irq_number; do
        echo ${cpu_list} > /proc/irq/${i}/smp_affinity_list
    done
}

modprobe nvpps
modprobe acu_timesync
modprobe scha634x_spi
modprobe scha634x
modprobe anp_imu
modprobe nct1008

# Do not pin delay sensitive tasks to VCPU 0,8,9,10
irq_cpu_bound b950000.tegra-hsp 4
irq_cpu_bound host_syncpt 2
irq_cpu_bound ttyS2 3
irq_cpu_bound bpmp_irq_handler 3
irq_cpu_bound timesync 1
irq_cpu_bound mgbe2_0.vm0 5
irq_cpu_bound mgbe1_0.vm0 6
irq_cpu_bound mgbe0_0.vm0 7
irq_cpu_bound 3210000.spi 11
irq_cpu_bound 3230000.spi 11
taskset -pc 11 `pidof spi0`
taskset -pc 11 `pidof spi2`
taskset -pc 11 `pidof imu-sampling`

chrt -p -r 51 `pgrep "\-timesyn"`
```

中断绑核统计

进度名称 & 中断名称	CPU Core号	备注说明
b950000.tegra-hsp	4	RCE和A核交互中断
host_syncpt	2	NV模块间同步
bpmp_irq_handler	3	BPMP通信中断
timesync irq	1	时间同步驱动中断
mgbe2_0.vm0	5	网卡相关中断
mgbe1_0.vm0	6	网卡相关中断
mgbe0_0.vm0	7	网卡相关中断
ttyS2	3	串口中断
3210000.spi	11	IMU驱动中断
3230000.spi	11	IMU驱动中断
spi0	11	SPI驱动线程
spi2	11	SPI驱动线程
imu-sampling	11	IMU驱动线程

进程绑cpu， 隔离核上迁移负载

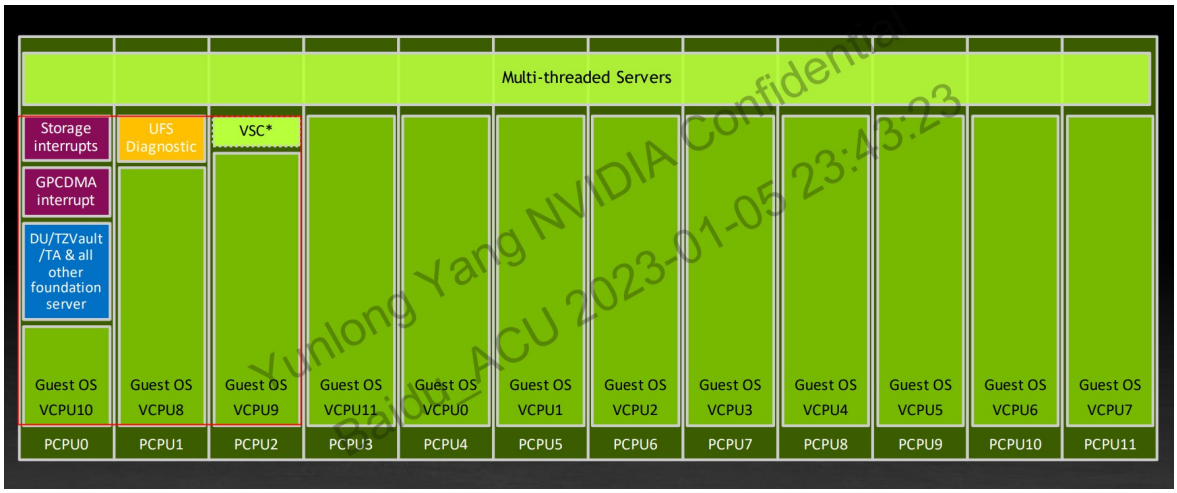
qa测试： cpu中断加压方法-复现camera绑定1核心后再次复现问

压测： Camera掉帧

Gaia	taskset -c 1-7,11 $BIN_PATH/gaia_timesync > /dev/null 2>&1 &	
ANP	anp3/cybertron/bin/carp_partition.sh	

二、ANP的绑定关系调整

咨询底软同学江帅，了解到0核默认有很多中断处理函数，8 9 10核上有跑nv的一些服、 实时性要求比较高的nv不建议在上面跑。



当前mainboard的绑核list是0-3,5-11，可以使用taskset命令改为1-7,11。需要root权限执行，

```bash
source setup.bash
# 原本启动命令：mainboard -p patrol_slave_orinx_sched -d patrol_slave_orinx.dag -d metric.dag
taskset -c 1-7,11 /opt/anp/bin/mainboard -p patrol_slave_orinx_sched -d patrol_slave_orinx.dag -d metric.dag
```

```
caros@acu-sanxian-jd-evt2-slave:~$ taskset -pc 483873
pid 483873's current affinity list: 1-7,11
```

因为anp启动时start_orinx.bash做了封装，需要在/opt/anp/bin/cyber_launch中做适配

```
    def start(self):
        """
        start a manager in process name
        """
        if self.process_type == 'binary':
            args_list = self.prefix.split() + self.name.split()
        else:
            dag_list = ['-d'] + ' -d '.join(self.dag_list).split()
            args_list = self.prefix.split() + [self.binary_path]
            if len(self.name) != 0:
                args_list.append('-p')
                args_list.append(self.name)

            if len(self.cpuprofile) != 0:
                args_list.append('-c')
                args_list.append('-o')
                args_list.append(self.cpuprofile)
            args_list += dag_list
```