功耗分析工具介绍

文档版本	V1.1
发布日期	2020-09-03

紫光展锐（上海）科技有限公司

紫光展锐（上海）科技有限公司

前 言

概述

文本档主要介绍了 CPU、GPU、DDR 等常用功耗分析工具的功能及使用方法。

读者对象

文档适用于需要借助功耗程序分析功耗相关问题的开发人员。

缩略语

无

符号约定

在本文中可能出现下列标志，它所代表的含义如下。

符号	说明
说明	用于突出重要/关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。

变更信息

文档版本	发布日期	修改说明
V1.0	2019-10-15	第一次正式发布。
V1.1	2020-09-03	1、 添加文档模板，优化文档内容 2、 调整文档结构、分类描述内容 3、 补充部分工具使用方法，添加输出结果说明 4、 文档名称由《Unisoc_Power_Tools_Introduction》修改为《功耗分析工具介绍》

关键字

功耗、频点、GPU、CPU、DDR、使用率。

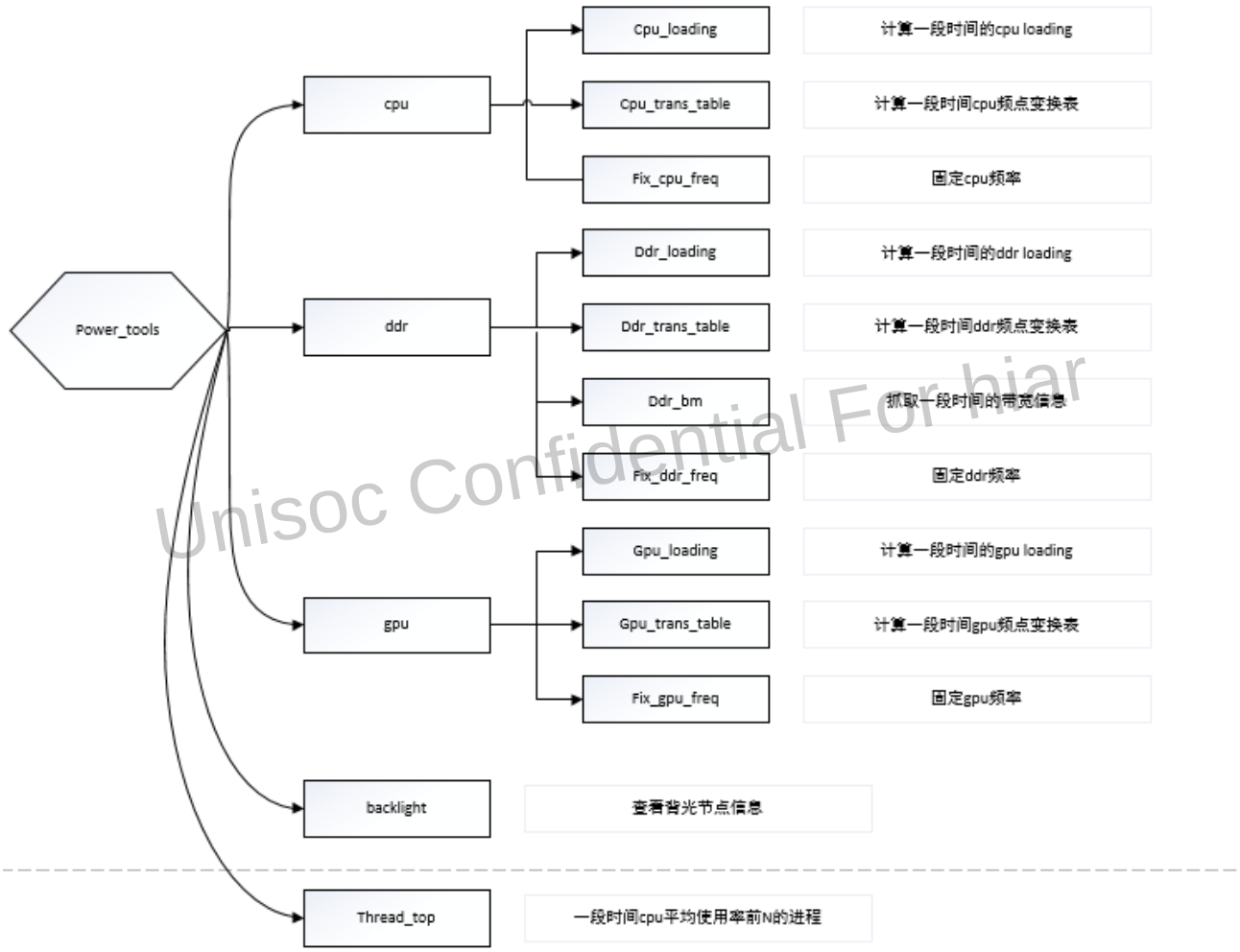
目 录

图目录

1 概览

展锐将常用的功耗分析涉及到的工具（脚本程序）集成到手机/vendor/bin/power 目录下，以便对常见功耗问题进行分析，功耗分析工具及功能简介详见图 1-1。

图1-1 常用功耗分析程序



2 工具使用详解

本文以 SC9863A 为例，介绍每个工具（脚本程序）的使用方法和输出结果。

2.1 CPU

2.1.1 cpu_loading

功能

cpu_loading 程序体现 CPU 每个 core 的每个频点的闲忙程度。

程序运行时间按需设定，单位：s(秒)。

使用方法

```
adb shell
cd vendor/bin/power/cpu/loading
./cpu_loading -t 60
```

上述命令中 60 代表运行 60s，可自定义。

运行结果

cpu_loading 运行 60s 后结果如下：

```
accumulate_time is 60(s)
                                                    小核频率
       768000  884000  1000000 1100000 1200000 online  offline
cpu0   9.92%   0.16%   14.72%  15.18%  45.38%  85.36%  14.64%
cpu1   9.07%   0.14%   13.47%  13.88%  41.51%  78.07%  21.93%
cpu2   7.25%   0.11%   10.77%  11.11%  33.20%  62.45%  37.55%
cpu3   4.69%   0.07%   6.97%   7.18%   21.47%  40.39%  59.61%
freq   11.62%  0.18%   17.25%  17.78%  53.17%
sum:2920860.38kHZ
                                                    大核频率
       768000  1050000 1225000 1400000 1500000 1600000 online  offline
cpu4   0.00%   0.00%   0.00%   0.00%   0.00%   48.11%  48.11%  51.89%
cpu5   0.00%   0.00%   0.00%   0.00%   0.00%   0.45%   0.45%   99.55%
cpu6   0.00%   0.00%   0.00%   0.00%   0.00%   55.86%  55.86%  44.14%
cpu7   0.00%   0.00%   0.00%   0.00%   0.00%   0.03%   0.03%   99.97%
freq   0.00%   0.00%   0.00%   0.00%   0.00%   100.00%
sum:1671152.25kHZ

SUM:4592012.63kHZ
```

输出结果显示 CPU 的分组情况，输出是二维矩阵，其中：

- 第一个红框表示小核可用频点。
- 第二个红框表示大核可用频点。
- cpu0、cpu1、cpu2、cpu3 属于小核 cluster0。
- cpu4、cpu5、cpu6、cpu7 属于大核 cluster1。
- Online：未拔核时，每个 CPU 运行时间对应总时间的占比。
- Offline：每个 CPU 非 online 状态的时间对应总时间的占比。
- SUM：大小核对应 sum 的总和，和功耗正相关。

2.1.2 cpu_trans_table

功能

cpu_trans_table 程序是扫描 CPU 各个频点的跳变关系，脚本默认扫描 1min，20ms 采样一次。

cpu_trans_table 提供所有有关 CPU 频率的细粒度信息过渡，输出的是二维矩阵。

使用方法

```
adb shell
cd vendor/bin/power/cpu/trans_table
./cpu_trans_table -t 60
```

上述命令中 60 代表运行 60s，可自定义。

运行结果

cpu_trans_table 运行 60s 后结果如下：

```
accumulate_time is 60
        768000  884000  1000000 1100000 1200000
768000  0       0       0       0       0
884000  0       1380    0       0       26
1000000 0       0       0       0       0
1100000 0       0       0       0       0
1200000 0       26      0       0       1567

        768000  1050000 1225000 1400000 1500000 1570000
768000  0       0       0       0       0       0
1050000 0       0       0       0       0       0
1225000 0       0       2882    0       0       2
1400000 0       0       0       0       0       0
1500000 0       0       0       0       0       0
1570000 0       0       2       0       0       113
```

输出结果中展示了每行、每列的实际 freq 值，其中：

条目<i，j>（第 i 行，第 j 列）表示 freq_i 到 freq_j 的转换次数。

2.1.3 fix_freq

功能

fix_freq 的功能是固定 CPU 大小核频率，运行过程中会提示“input fix freq”，按照提示输入固定的频率即可。

注意：输入的固定频率须在可用频率中选取。

cur_freq 显示的频率和输入的频率一致，则代表固定成功；重启后失效，需重新设置。

分析问题时若想将 CPU 大小核频率固定在某一个频点，可使用该程序。

使用方法

```
adb shell
cd vendor/bin/power/cpu/fix_freq
./fix_cpu_freq
```

运行结果

运行 fix_freq，输入固定的 CPU 大小核频率后，结果如下：

```
s9863a1h10:/vendor/bin/power/cpu/fix_freq # ./fix_cpu_freq
lit core
768000  884000  1000000 1100000 1200000
input fix freq
884000
cur_freq:884000

big core
768000  1050000 1225000 1400000 1500000 1600000
input fix freq
1050000
cur_freq:1050000
```

上图表示：CPU 小核固定在 884000Hz，大核固定在 1050000Hz，重启将失效。

注意：如果测试场景中没有要求拔核，则需在拔核后重启手机。

2.2 GPU

2.2.1 gpu_loading

功能

gpu_loading 统计一段时间内 GPU 在各个频点的占比和使用情况。

程序运行时间按需设定，单位：s(秒)。

使用方法

```
adb shell
cd vendor/bin/power/gpu/loading
./gpu_loading -t 60
```

上述命令中 60 代表运行 60s，可自定义。

运行结果

gpu_loading 运行 60s 后结果如下：

```
accummulate_time is 60(s)
256: 58.08%       utilisation:28.88%
384: 30.92%       utilisation:21.09%
550: 11.00%       utilisation:23.70%
Averag GPU Utilisation : 25.90%
```


上述输出结果表示 60s 内：

- GPU 运行在 256MHz、384MHz、550MHz 频点上的占比分别为 58.08%、30.92%、11.00%，
- 频点 256MHz、384MHz、550MHz 的利用率依次为 28.88%、21.09%、23.7%。
- GPU 平均占用率是 25.90%。

一般占比越高，则功耗越高。

2.2.2 gpu_trans_table

功能

gpu_trans_table 是统计一段时间 GPU 频点的跳变关系，提供所有有关 GPU 频率的细粒度信息过渡，输出的是二维矩阵。

程序运行时间按需设定，单位：s(秒)。

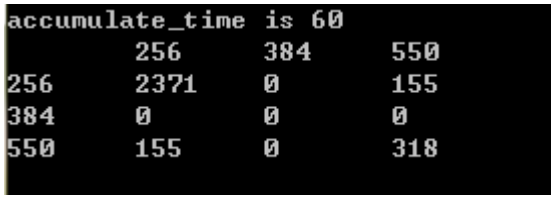
使用方法

```
adb shell
cd vendor/bin/power/gpu/trans_table
./gpu_trans_table -t 60
```

上述命令中 60 代表运行 60s，可自定义。

运行结果

gpu_trans_table 运行 60s 后结果如下：



上图中展示了每行、每列的实际 freq 值。其中：

条目<i，j>（第 i 行，第 j 列）表示 freq_i 到 freq_j 的转换次数。

2.2.3 fix_gpu_freq

功能

fix_gpu_freq 的功能是固定 GPU 频率，运行过程中会提示“input fix freq”，按照提示输入固定的频率即可。

注意：输入的固定频率须在可用频率中选取。

cur_freq 显示的频率和输入的频率一致，则代表固定成功；重启后失效，需重新设置。

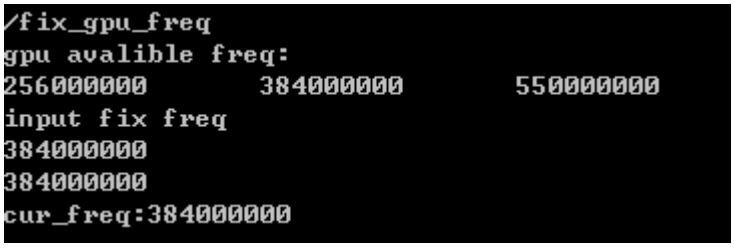
分析问题时若想将 GPU 固定在某一个频点，可使用该程序。

使用方法

```
adb shell
cd vendor/bin/power/gpu/fix_freq
./fix_gpu_freq
```

运行结果

运行 fix_gpu_freq，输入固定的 GPU 频率后，结果如下：



上图表示将 GPU 的频率固定在 384MHz。

2.3 DDR

2.3.1 ddr_loading

功能

ddr_loading 是统计一段时间内 DDR 在各个频点的使用率。

程序运行时间按需设定，单位：s(秒)。

使用方法

```
adb shell
cd vendor/bin/power/ddr/loading
./ddr_loading –t 60
```

上述命令中 60 代表运行 60s，可自定义。

运行结果

ddr_loading 运行 60s 后结果如下：

```
s9863a1h10:/vendor/bin/power/ddr/loading # ./ddr_loading -t 60
./ddr_loading -t 60
accummulate_time is 60(s)
time_use is 55463.82ms
160:    0.00%
233:    0.00%
311:    98.84%
400:    0.92%
533:    0.00%
622:    0.20%
800:    0.03%
933:    0.00%
unit:10ms
theory_bw = 1280000     1864000 2488000 3200000 4264000 4976000 6400000 7464000
overflow  = 448000 652400 870800 1120000 1492400 1741600 2240000 4294967295
underflow = 0 348000 552400 770800 1020000 1020000 1641600 2140000
```

上述运行结果代表 60 秒时间内，DDR 运行在 160MHz、233MHz、311MHz、400MHz、533MHz、622MHz、800MHz、933MHz 的占比依次为 0.00%、0.00%、98.84%、0.92%、0.00%、0.20%、0.03%、0.00%；高频占比越高，则功耗越高

2.3.2 ddr_trans_table

功能

ddr_trans_table 统计一段时间 DDR 频点的跳变关系，提供所有有关 DDR 频率的细粒度信息过渡。输出的是二维矩阵。

程序运行时间按需设定，单位：s(秒)

使用方法

```
adb shell
cd vendor/bin/power/ddr/trans_table
./ddr_trans_table –t 60
```

上述命令中 60 代表运行 60s，可自定义。

运行结果

ddr_trans_table 运行 60s 后，结果如下：

```
accumulate_time is 60
        160     233     311     400     533     622     800     933
160     0       0       0       0       0       0       0       0
233     0       0       0       0       0       0       0       0
311     0       0       246     389     0       539     109     7
400     0       0       238     396     0       751     251     18
533     0       0       0       0       0       0       0       0
622     0       0       608     560     0       741     307     42
800     0       0       186     288     0       187     43      12
933     0       0       12      21      0       40      6       2
```

上述输出结果中展示了每行、每列的实际 freq 值，其中：

条目<i，j>（第 i 行，第 j 列）表示 freq_i 到 freq_j 的转换次数。

2.3.3 ddr_bm

功能

ddr_bm 是抓取一段时间内的带宽数据，输入抓取带宽指令，程序运行到达指定时间将在 vendor/bin/power/ddr/bm 目录下生成 ddr_bm.csv 文件，即为带宽数据。

注意：每次抓取带宽之前需要重启设备。

使用方法

```
adb shell
cd vendor/bin/power/ddr/bm
./tool/ddr_bm –t 240
```

上述命令中 240 代表运行 240s，可自定义。

运行结果

ddr_bm 运行 240s 后结果如下：


```
s9863a1h10:/vendor/bin/power/ddr/bm # ./tool/ddr_bm -t 240
./ddr_bm -t 240
accummulate_time is 240(s)
s9863a1h10:/vendor/bin/power/ddr/bm # ls
ls
tool readme sharkl3_ddr_bm.csv
```

2.3.4 fix_ddr_freq

功能

fix_ddr_freq 的功能是固定 DDR 频率，行过程中会提示“input fix freq”，按照提示输入固定的频率即可。

注意：输入的固定频率须在可用频率中选取。

cur_freq 显示的频率和输入的频率一致，则代表固定成功；重启后失效，需重新设置。

分析问题时若想将 DDR 固定在某一个频率，可使用该脚本。

使用方法

```
adb shell
cd vendor/bin/power/ddr/fix_freq
./fix_ddr_freq
```


运行结果

运行 fix_ddr_freq，输入固定的 DDR 频率后，结果如下：

```
/fix_ddr_freq
ddr avalible freq:
160     233     311     400     533     622     800     933
input fix freq
233
233
cur_freq:233
```

上图是将 DDR 频率固定在 233MHz。

2.4 Backlight

功能

Backlight 是查看背光节点（Brightness）信息，背光节点的取值范围为[0，255]。

通常情况该值越高，亮度越高，功耗则也会越大。

使用方法

```
adb shell
cd vendor/bin/power/backlight
./brightness
```

运行结果

运行 Backlight 脚本，执行结果如下：

```
s9863a1h10:/vendor/bin/power/backlight # ./brightness
./brightness
brightness: 0
brightness: 0
brightness: 0
brightness: 92
brightness: 92
brightness: 92
brightness: 92
brightness: 170
brightness: 204
brightness: 204
brightness: 204
brightness: 204
```

2.5 thread_top

功能

thread_top 程序可以获取一段时间内 CPU 使用率前 n 的线程信息。

建议 n 的值不超过 20。

使用方法

```
adb shell
cd vendor/bin/power/tops
./tops –t 10 –n 10
```

其中：

- 参数-t 表示运行的时间，单位 s（秒）。
- 参数-n 表示显示的进程的个数。

运行结果

thread_top 运行 10s，显示 CPU 使用率前 10 的运行结果如下：


```
s9863a1h10:/vendor/bin/power/tops # ./tops -t 10 -n 10
%[CPU]  ARGS
41.9    top -n 1 -d 3
 35.0   com.android.deskclock
 22.1   surfaceflinger
 7.6    android.hardware.graphics.composer@2.1-service
 3.4    [kworker/u16:0]
 3.3    [kworker/u16:1]
 3.0    [sprd-adf-dev]
 2.3    logd
 2.2    [kworker/u16:3]
 2.2    [rcu_preempt]
```

第一列显示为 CPU 的占比，第二列为线程的名称。

