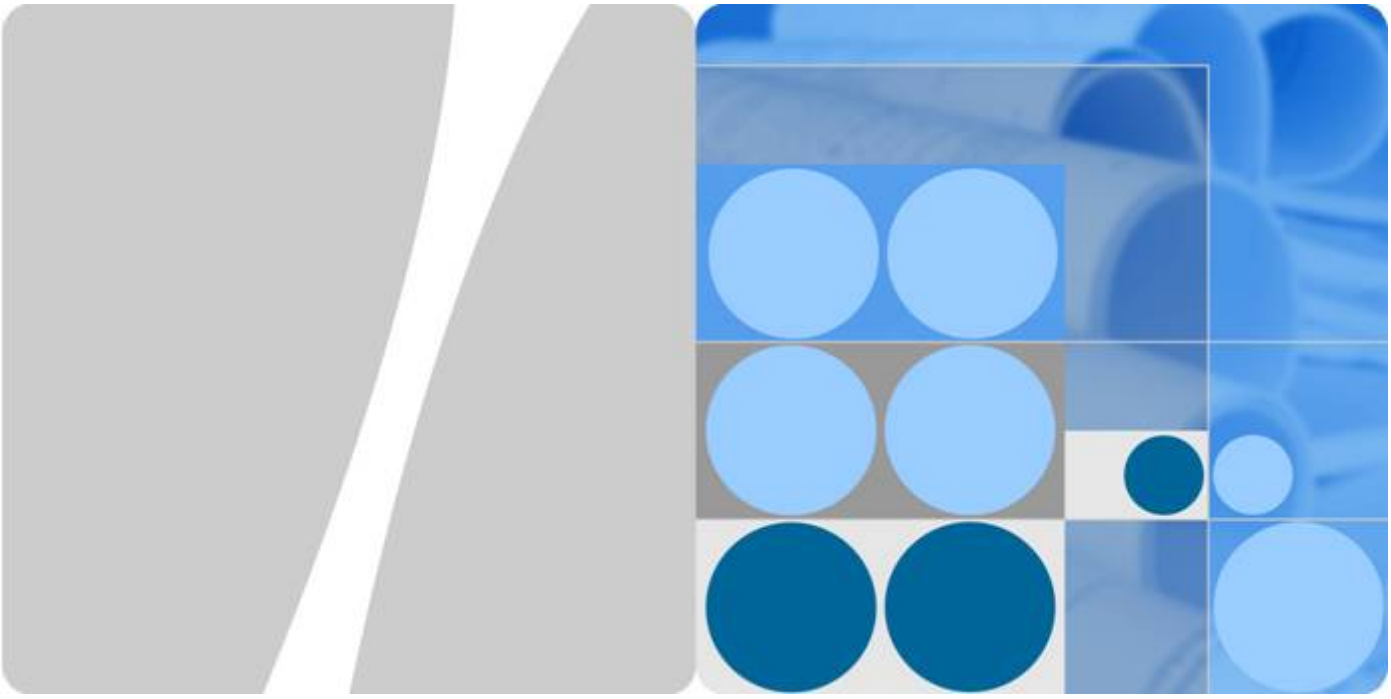


Ascend 310

V100R001

Mind Studio 开发辅助工具

文档版本　01

发布日期　2019-03-12

华为技术有限公司

HUAWEI

版权所有 © 华为技术有限公司 2019。 保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： http://www.huawei.com

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目 录

 版权所有 © 华为技术有限公司

 版权所有 © 华为技术有限公司

1 日志工具

1.1 简介

Mind Studio为NPU（Neural network Processing Unit）平台提供覆盖全系统的日志收集与日志分析解决方案，提升运行时算法问题定位效率。

Mind Studio提供全系统统一的日志格式，并以图形化的形式提供跨平台日志可视化分析能力及运行时诊断能力，提升日志分析系统的易用性。

说明

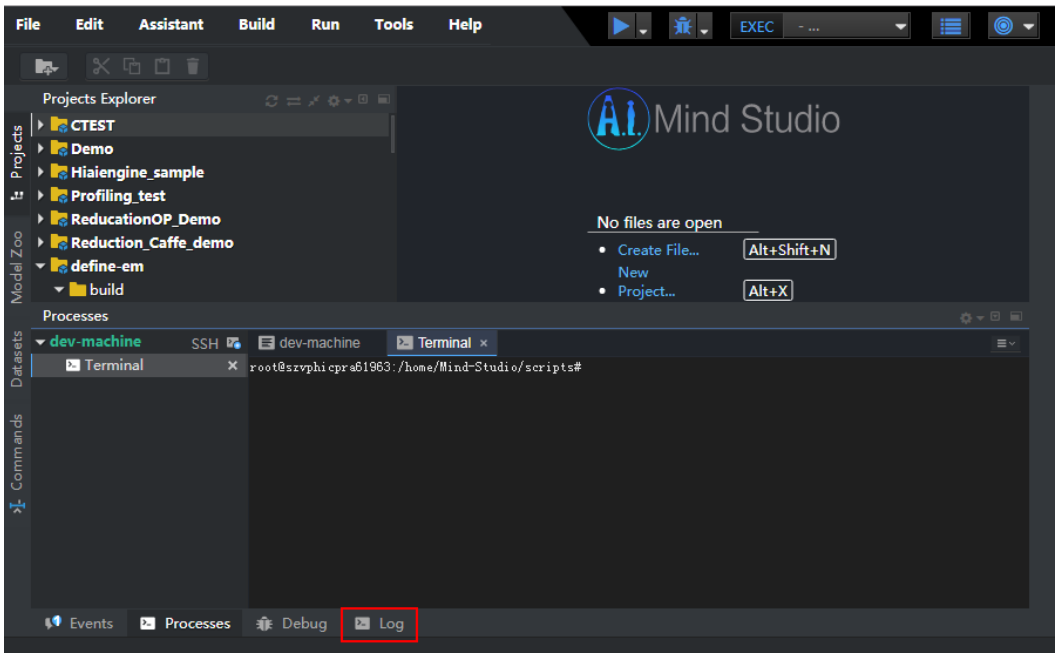
1、当Host侧为CentOS环境时，请确保CentOS防火墙是关闭的，否则Mind Studio可能无法连接Host侧，导致无法查看日志信息。

2、使用Log工具之前，建议先安装JDK1.8.0_171及以上的版本，否则动态传输日志时，会出现失败。

在Mind Studio窗口底部单击“Log”标签，显示“Log”窗口，您可以通过该窗口查看日志信息，如图1-1所示。

 版权所有 © 华为技术有限公司

图 1-1 日志查看入口



1.2 日志概述

日志主要用于记录系统的运行过程及异常信息，为快速定位系统运行中出现的问题及开发过程中程序调试问题提供详细信息。

1.2.1 日志处理机制介绍

图 1-2 日志系统架构

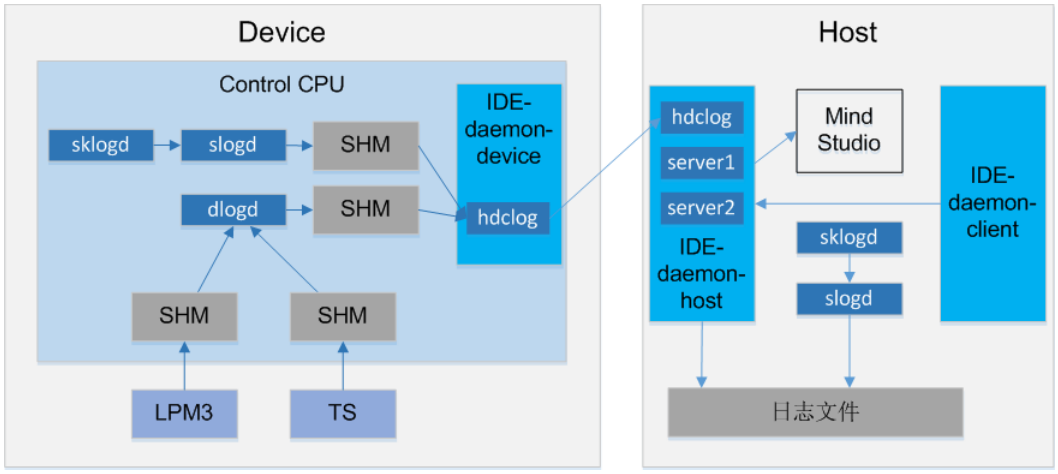


图1-2中的日志数据的流向解释如下：

1. 采集日志

   在Device侧，dlogd进程调用log驱动采集非Control CPU上的日志，sklogd和slogd进程采集Control CPU上的日志。

   在Host侧，sklogd和slogd进程采集用户态和内核态日志。

 版权所有 © 华为技术有限公司

2. 传输日志

   Device侧的日志可通过IDE-daemon-device、IDE-daemon-host模块传输给Host侧。

   Device侧的dlogd和slogd将采集到的日志分别写入各自的共享内存，被挂载在IDE-daemon-device中的日志线程发送给host侧的IDE-daemon-host。

3. 将日志记录到文件中

   Device侧的日志被IDE-daemon-host记录在文件名称以device-0开头的日志文件中，Host侧的日志被slogd记录在文件名称以host-0开头的日志文件中。IDE-daemon-host中拉起TCP/IP线程，将host侧日志发送给Mind Studio，即完成日志的传输。

   Host以自定义压缩的方式，将日志内容记录到日志文件中，您可以在Mind Studio工具上查看日志信息，但无法直接通过日志文件查看日志。若直接查看日志文件，日志信息以乱码显示。

   如果需要调整日志级别，可通过IDE-daemon-client实现。见2.3.5 。

1.2.2 日志文件介绍

在Mind Studio中，日志文件区分Device侧和Host侧，本节介绍各侧日志的存储路径、文件名称及记录的主要信息。

表 1-1 日志分类介绍

日志文件	存储路径	说明
device-0_*.log	“/var/dlog”	● Slog采集Device侧Control CPU上的日志，主要采集以下模块的日志： – Slog – Dlog – IDE-daemon-device – Matrix – Driver – HDC ● Dlog采集Device侧非Control CPU上的日志，主要采集以下模块的日志： – Task Scheduler CPU – LPM3

 版权所有 © 华为技术有限公司

日志文件	存储路径	说明
host-0_*.log	“/var/dlog”	Slog采集Host侧用户态和内核态日志，主要采集以下模块的日志： ● Slog ● Dlog ● Matrix ● Framework ● Runtime ● CCE ● IDE-daemon-host ● Driver ● HCCL ● DVPP ● HDC ● MDC ● MLL ● Dlog Memory Managent ● Kernel

1.2.3 日志级别介绍

本节介绍日志的级别，以及各级别日志的定义。

表 1-2 日志级别

日志级别	定义
ERROR	一般错误级别，该级别的日志记录了如下错误： ● 非预期的数据或者事件。 ● 影响面较大且模块内部能够处理的错误。 ● 限制在模块内的错误。 ● 对其他模块有轻微影响的错误，例如统计任务创建失败。 ● 引起调用失败的错误。 ● 在业务逻辑错误的情况下记录错误状态的信息及造成错误的可能原因。
WARNING	警告级别，系统和预期的状态不一致，但不影响整个系统的运行。
INFO	正常级别，系统正常运行的信息。
DEBUG	调试级别，该级别的日志记录了调试信息，便于开发人员或维护人员定位问题。

 版权所有 © 华为技术有限公司

日志级别	定义
EVENT	输出全系统最关键日志，例如：整网运算启动\完成\异常终止， 内存不够，单板温度过高等。

1.2.4 日志记录格式介绍

本节介绍系统记录日志的格式以及各字段的含义，便于理解日志所记录的信息。

日志样例如下：

```
[INFO] KERNEL(1080,sklogd):1970-01-01-08:00:04.495.604 sklogd started
```

日志格式如下：

```
[Level] ModuleName(PID,PName):DateTimeMS LogContent
```

表 1-3 日志字段说明

字段	说明
Level	日志级别。运行日志存在5种日志级别：ERROR、WARNING、INFO、DEBUG、EVENT。
ModuleName	产生日志的模块的名称。
PID	进程ID。
PName	进程名称。
DateTimeMS	日志打印时间，格式为：yyyy-mm-dd hh:mm:ss.SSS。
LogContent	各模块具体的日志内容。

⚠注意

日志格式化字符串中，务必正确使用日志格式化字符串%。若要打印%符号需使用%%的形式，与printf同。

未正确使用不能正确打印出日志。

1.2.5 日志配置介绍

系统运行过程中需要记录的日志级别、日志输出路径、日志文件名、单个日志文件大小等信息可以配置，本节对控制日志生成的配置进行介绍。

slog.conf

/etc/slog.conf文件用于控制**Slog**采集日志时的配置，该文件中的配置段样例，如图1-3所示。

 版权所有 © 华为技术有限公司

图 1-3 配置样例

```
# Global log level
global_level=6

# Share memory size of node 512k * 32 biggest support
maxNodeSize=524272

# Share memory count of queue
maxQueueCount=32

# log-agent-host #
logAgentMaxFileNum=1
# set host one log file max size
logAgentMaxFileSize=10485760
# set host log dir
logAgentFileDir=/var/dlog

#log server send log to IDE
#IP_address to bind: if you want to set ip_address manually, please open the item and add your ip.
#log_server_ip = 127.0.0.1

# Port: Please select a port number between 18000 and 18080([18000, 18080]).
daemon_socket_port=18080

#enable print event log, 0:disable, 1:enable
enableEvent=1
```

相关配置项说明详情见表1-4。

表 1-4 相关配置项说明

配置项	说明
Global_level	配置slog日志级别，device与host侧是单独配置的。 ● 0：表示DEBUG级别。 ● 1：表示INFO级别。 ● 2：表示WARNING级别。 ● 3：表示ERROR级别。
MaxNodeSize	共享内存（SHM）节点大小，最大支持为524272bytes。
MaxQueueCount	共享内存（SHM）节点数目。
LogAgentMaxFileNum	Host侧“/var/dlog”下保存文件的数量，Host侧日志文件数目大于该数目时发生滚动，新日志覆盖最早的日志。 该参数在Device侧无效。
logAgentMaxFileSize	单个日志文件大小，该数值为最大值，如果日志文件大小超过该最大值，则生成新的日志文件。当前默认值是10M，您可以根据实际情况调整大小。最大不超过100M。 该参数在Device侧无效。
logAgentFileDir	日志文件路径。 该参数在Device侧无效。
daemon_socket_port	发送日志给IDE（Mind studio）的端口号。 该参数在Device侧无效。

 版权所有 © 华为技术有限公司

dlog.conf

/etc/dlog.conf文件用于控制**Dlog**采集日志时的配置，该文件中的配置段样例，如**图1-4**所示。

图 1-4 配置样例

```
{
    "method": 0,
    "channel":
    [
        {
            "name": "ts",
            "type": 0,
            "id": 0,
            "level": 1,
            "buffer_size": 1048576
        },
        {
            "name": "ts-dump",
            "type": 0,
            "id": 1,
            "level": 1,
            "buffer_size": 1048576
        },
        {
            "name": "lpm3",
            "type": 2,
            "id": 30,
            "level": 3,
            "buffer_addr": "0xa00000"
        }
    ]
```

相关配置项说明详情见**表1-5**。

表 1-5 相关配置项说明

配置项	说明
Method	获取方式，暂未使用。
Channel	通道，当前包括4种类型的通道。
Name	默认通道名，不可配置。
Type	通道类型，不可配置。
id	通道模块id，包括id_max，id_min。不可配置。
Level	用于配置该类型通道的日志级别。
Buffer_size	用于配置buffer大小。
Buffer_addr	buffer地址，不可配置。

 版权所有 © 华为技术有限公司

1.3 基本操作

本节介绍在Mind Studio工具上一些基本的日志操作，如查看、导出、删除、上传日志。

1.3.1 日志查看

步骤1 在Chrome浏览器的地址栏输入访问Mind Studio界面的地址。

访问地址：***https://安装机器的IP地址:8888***

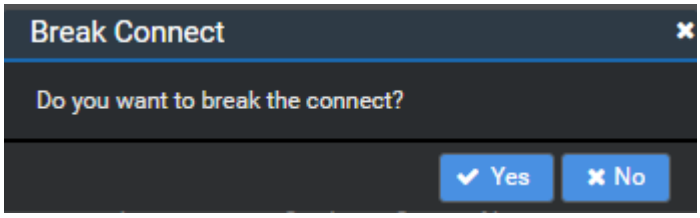
步骤2 在窗口底部单击“Log”标签，显示“Log”窗口。

步骤3 单击左上角“Log List”上的连接配置图标，输入Host侧主机的IP地址（“HostAddress”参数）和端口号（“Port”参数，当前必须配置为“18080”），如图1-5所示。

图 1-5 连接配置

如果您想断开连接，可以直接单击“Log List”上的图标，系统会弹出图1-6中的对话框，单击“Yes”。

图 1-6 断开连接



步骤4 单击“OK”后，在界面右侧系统提示“Connect success!”，则表示连接成功。

步骤5 在“Log”窗口，展开Log List，选中一个device或者host，查看该device或者host对应的所有日志。

每条日志信息包含如下字段：

- **Type**：日志级别
- **Time**：日志生成时间戳

 版权所有 © 华为技术有限公司

- **Module**：上报日志的模块名
- **Content**：日志内容，如果需要查看日志信息的上下文内容，可双击该列下的日志信息，在系统显示的页面中查看日志信息的上下文内容。若**Time**列内容为空，则无法查看对应日志信息的上下文内容。

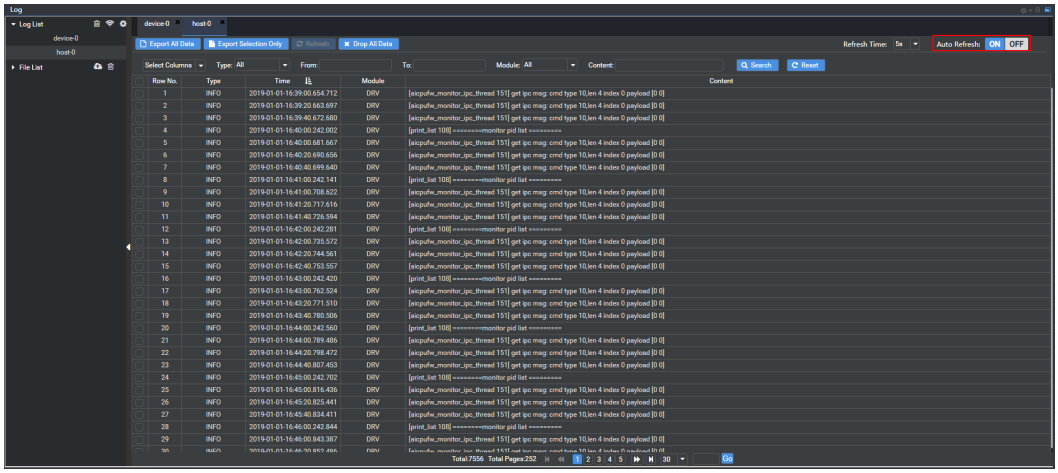
您可以单击界面左边的**Select Columns**筛选日志想要包含的字段，例如还可以筛选出如下字段：

- **PID**：上报日志的进程号
- **PName**：上报日志的进程名
- **SubModule**：上报日志的子模块名

您可以根据日志级别、时间范围、进程号、进程名、模块名、子模块名以及日志内容中的关键字，精确查询符合条件的日志信息。

步骤6 打开device或者host日志窗口后，Auto Refresh默认处于ON状态，动态从host侧获取日志，且每5秒刷新一次，每次刷新都会获取最新的1w条日志，如图1-7所示。

图 1-7 刷新日志信息

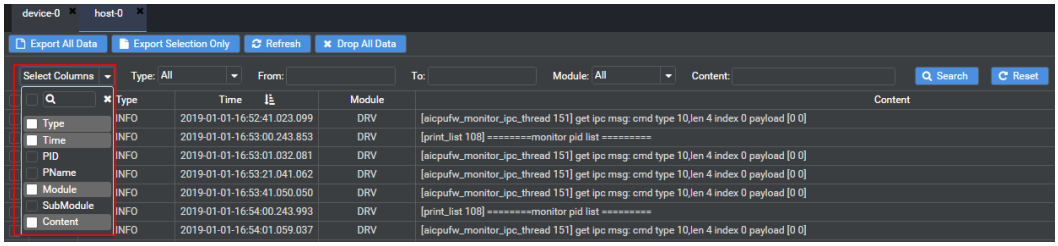


当您单击查询按钮或者跳转页时，“Auto Refresh”会自动关闭，并停止从host侧动态获取日志。只有当您重新打开“Auto Refresh”后，才能动态更新日志。对于不同的device或者host，如果同时打开了“Auto Refresh”，每次也仅传输一个device或者host的日志，Log窗口显示的为当前device或者host。

另外，您也可以通过单击“Refresh”按钮，从host侧动态获取一次最新的日志，只有当“Auto Refresh”处于OFF状态时，单击该按钮才能有效 。

步骤7 如果您需要调整日志显示列，可以单击“Log”窗口顶部中央的“Select Columns”下拉框进行配置，如图1-8所示。

图 1-8 设置日志显示列



 版权所有 © 华为技术有限公司

说明

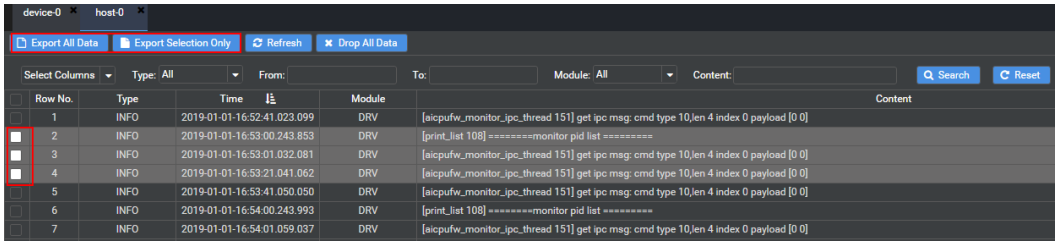
Content下的搜索只针对当前打开的文件有效。

----结束

1.3.2 日志导出

在日志查看窗口，您可以单击“Export All Data”导出全部当前显示的日志信息，也可以在“index”列勾选部分日志信息后单击“Export Selection Only”导出选中的日志信息，如图1-9所示。

图 1-9 日志导出



1.3.3 日志上传

步骤1 在“Log”窗口的“File List”区域，单击。

步骤2 在弹出的“New Log”窗口中，单击“Select”选择日志文件，如图1-10。

说明

目前只支持选择*.log的日志文件，如果日志文件中包含其他格式文件，则导入时会提示格式错误，如图1-11所示，单击Upload，则只会上传*.log的日志文件。

图 1-10 上传日志文件

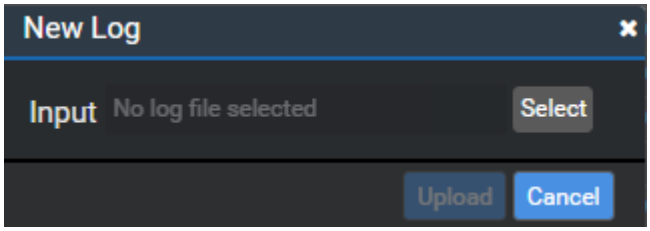
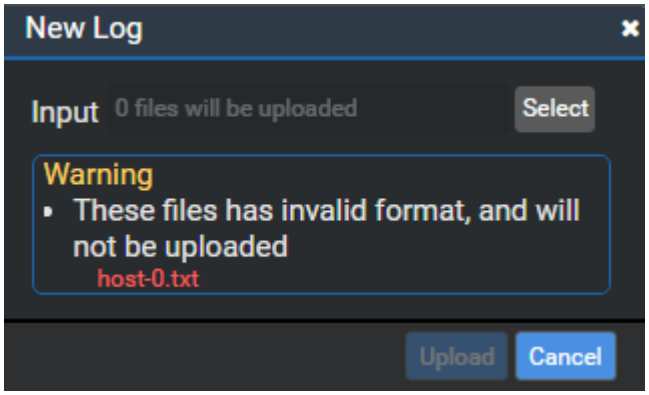


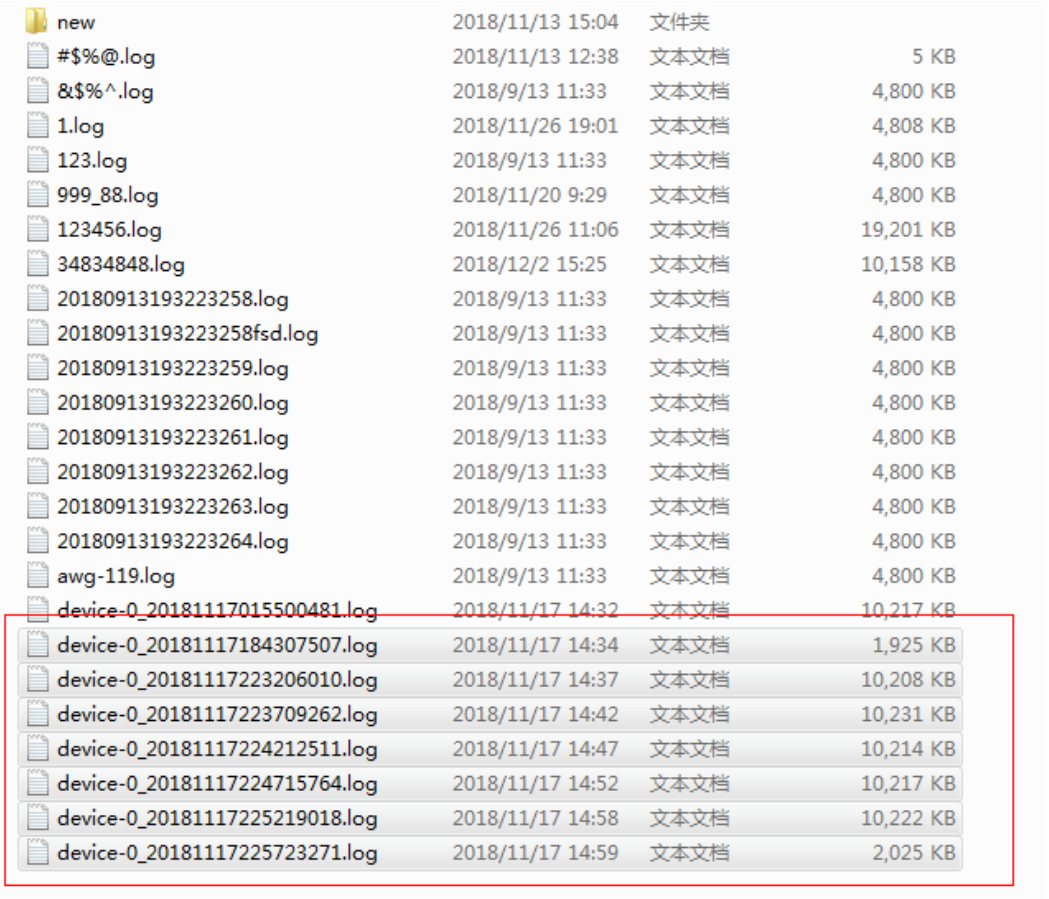
图 1-11 导入其他格式日志文件



 版权所有 © 华为技术有限公司

步骤3 您若要选择多个文件，直接选择多个文件即可，每次导入最大可选择50个文件，如图1-12所示。

图 1-12 选择多个文件



步骤4 选好需要上传的日志文件后，单击“Upload”。

日志文件上传成功后，在“File List”区域可查看上传的日志文件。

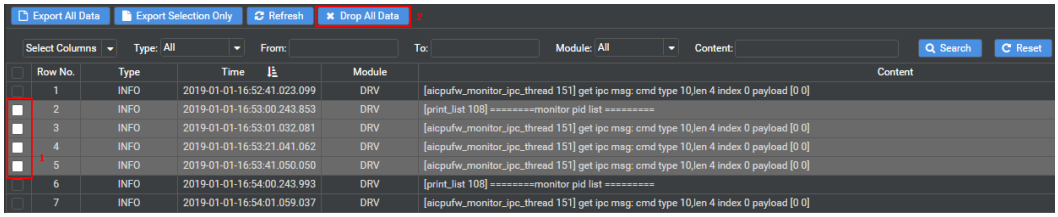
----结束

1.3.4 日志删除

步骤1 删除“Log list”下的日志文件。

若您不想保留日志文件，则可以对日志进行清空，通过单击“Log list”处的  ，将“Log List”下的所有日志都删除。或者通过单击“Drop All Data”删除当前Device ID对应的所有日志，如图1-13所示。

图 1-13 批量删除

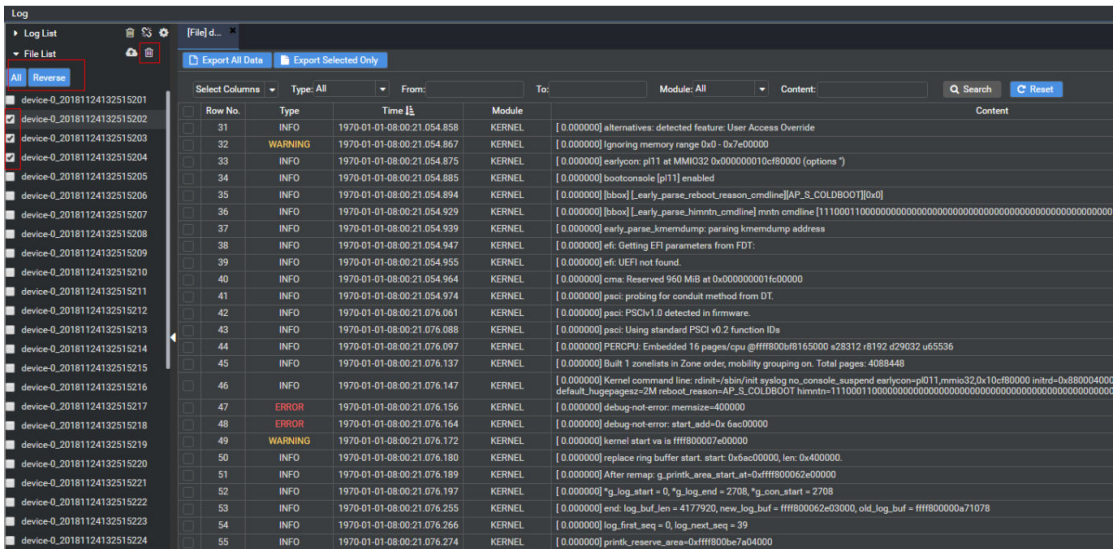


 版权所有 © 华为技术有限公司

步骤2 删除“File List”下的日志文件。

每个日志文件名前都会有一个勾选框，如图2 批量删除所示。您可以勾选多个要删除的日志文件，或者点击All选择所有的日志文件，也可以点击Reverse，对所有文件的勾选状态进行反转。之后再点击“File List”处的  ，并在弹出的“Delete Conformation”对话框中单击“Yes”，即可删除所有选中的日志文件。

图 1-14 批量删除



----结束

1.3.5 日志级别设置

目前，支持通过以下三种方式来设置日志级别：

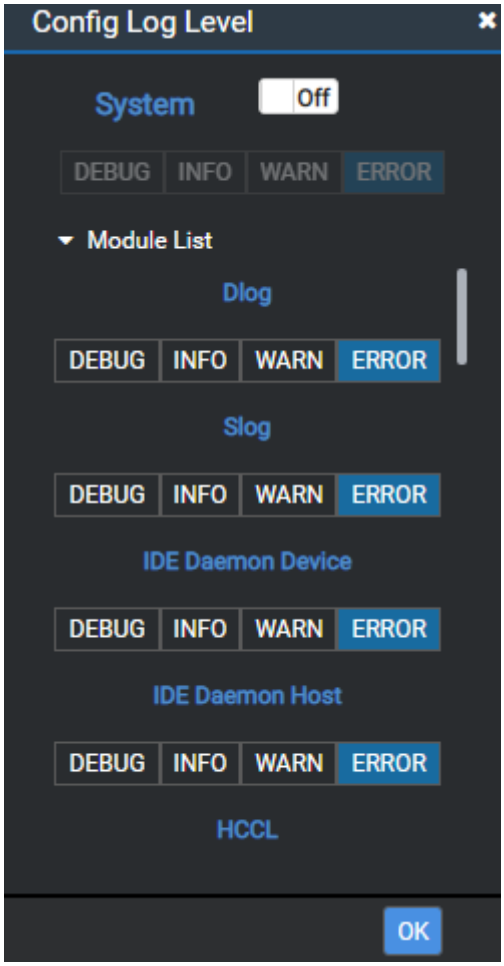
- 在Mind Studio界面设置日志级别。

  在Mind Studio界面的“Log”窗口，单击“Log List”处的  ，在弹出的“Config Log Level”窗口中设置日志级别后，单击“OK”。

  “System”区域的日志级别设置是全局的，“ModuleList”区域的日志级别设置是针对每个模块。如果设置了全局级的日志级别就不能再设置模块级的日志级别。如果需要设置模块级的日志级别，需要先关闭System处的日志级别设置，如图1-15所示。

 版权所有 © 华为技术有限公司

图 1-15 关闭全局级日志级别的设置



- 在服务器后台通过执行命令来设置日志级别。

  您可以用Mind Studio安装用户登录服务器后，依次执行**export LD_LIBRARY_PATH=~/tools/che/ddk/ddk/uihost/lib**和**export PATH=$PATH:~/tools/che/ddk/ddk/uihost/bin**设置环境变量后，执行全局级或模块级的日志级别设置命令。

  您也可以HwHiAiUser用户登录Host侧服务器（目前支持PCIE形态），执行全局级或模块级的日志级别设置命令。

  - 设置全局级的日志级别的命令如下：
    ```
    IDE-daemon-client --host 服务器的IP地址:端口号 --log 'SetLogLevel(0)[level]'
    ```
    “level”可设置为error、info、warning、debug中的一个。
  - 设置模块级的日志级别的命令如下：
    ```
    IDE-daemon-client --host 服务器的IP地址:端口号 --log 'SetLogLevel(1)[moduleName:level]'
    ```
    “level”可设置为error、info、warning、debug中的一个。

    “moduleName”可设置为slog、dlog、cce等模块中的一个。
  - 设置event的日志级别的命令如下：
    ```
    IDE-daemon-client --host 服务器的IP地址:端口号 --log 'SetLogLevel(2)[enable/disable]'
    ```
    “enable”：开启event日志级别。

    “disable”：不开启event日志级别。

 版权所有 © 华为技术有限公司

说明

“~/tools”是默认的toolpath路径，该路径可在安装Mind Studio时由用户自定义，您可以在“scripts/env.conf”文件通过toolpath参数查看实际路径。您可以使用**find / -name 'env.conf'**命令查看script目录下的“env.conf”文件的位置。

- 在服务器后台通过修改配置文件来设置slogd、dlogd日志级别。

  配置文件的说明请参见1.2.5 日志配置介绍。

  修改完配置文件后，以**HwHiAiUser**用户登录Host侧的服务器，以ssh方式登录到Device侧的服务器，重启slogd、dlogd、IDE-daemon-host进程：

  - 登录Host侧的服务器，进入目录“/usr/local/HiAi/driver/tools”，重启slogd进程：pkill slogd；./slogd &
  - 登录Host侧的服务器，进入目录“/usr/local/HiAi/driver/tools”，重启IDE-daemon-host进程：pkill IDE-daemon； ./IDE-daemon-host &
  - 登录Device侧的服务器，重启slogd进程：pkill slogd；/var/slogd >/dev/null &
  - 登录Device侧的服务器，重启dlogd进程：pkill dlogd；/var/dlogd >/dev/null &

  说明

  对于开发者板，登录服务器后，可执行**pkill *进程名*；/var/*进程名* >/dev/null &**重启进程。

1.4 常见问题

本节介绍通过日志分析一些常见问题，便于帮助您快速定位问题。

1.4.1 日志目录没有生成日志文件，怎么办？

如果日志目录下没有生成日志文件，您需要检查Host侧、Device侧的对应进程是否正常运行。

如果进程不存在，可以手工启动进程，启动命令为**/var/进程名>/dev/null &**。

步骤1 查看Host侧的slogd进程是否存在，如图1-16所示。

执行如下命令：

```
ps -elf | grep slogd
```

图 1-16 检查 slogd 进程

```
root@huawei:~# ps -elf |grep slogd
4 S syslog     831     1  0  80   0 - 64097 poll_s Nov14 ?        00:00:07 /usr/sbin/rsyslogd -n
1 S root      1046     1  0  80   0 - 24982 skb_re Nov14 ?        00:00:23 /usr/local/HiAI/driver/tools/slogd
0 S root     12370 12356  0  80   0 -  3236 pipe_w 18:22 pts/0    00:00:00 grep --color=auto slogd
root@huawei:~#
```

步骤2 查看Host侧的IDE-daemon-host进程是否存在，如图1-17所示。

执行如下命令：

```
ps -elf | grep IDE
```

图 1-17 检查 IDE-daemon-host 进程

```
0 S root     12370 12356  0  80   0 -  3236 pipe_w 18:22 pts/0    00:00:00 grep --color=auto slogd
root@huawei:~# ps -elf |grep IDE
1 S root      1073     1  0  80   0 - 250011 futex_ Nov14 ?        00:01:02 /usr/local/HiAI/driver/tools/IDE-daemon-host
```

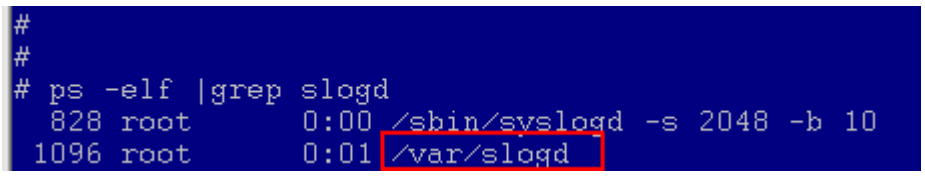
步骤3 查看Device侧的slogd进程是否存在，如图1-18所示。

执行如下命令：

 版权所有 © 华为技术有限公司

```
ps -elf | grep slogd
```

图 1-18 查看 slogd 进程

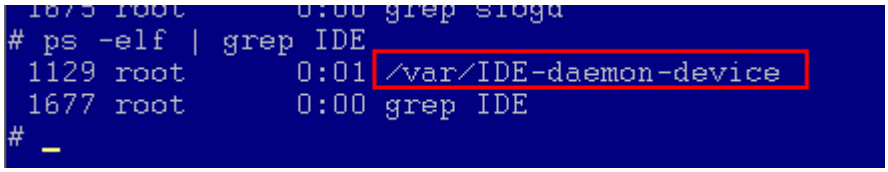


步骤4 查看Device侧的IDE-daemon-device进程是否存在，如图1-19所示。

执行如下命令：

```
ps -elf | grep IDE-daemon-device
```

图 1-19 查看 IDE-deamon-device 进程

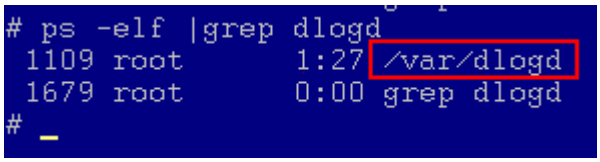


步骤5 查看Device侧的dlogd进程是否存在，如图1-20所示。

执行如下命令：

```
ps -elf | grep dlogd
```

图 1-20 查看 dlogd 进程



步骤6 查看Device侧的驱动是否存在，如图1-21所示。

执行如下命令：

```
lsmod
```

 版权所有 © 华为技术有限公司

图 1-21 查看驱动

```
# lsmod
Module                  Size  Used by
optee_drv              19290  1
tee_drv                24417  2 optee_drv
drv_pwm                 5974  0
drv_fan                 3478  0
drv_dbg                24798  0
drv_prof               28011  2
drv_gmac               68932  1 drv_prof
drv_mdio               13914  0
drv_log                19711  2
drv_dvpp              404203  1 drv_prof
drv_devdrv             88786  3 drv_dbg,drv_log,drv_prof
drv_pcie_hdc           85396  8
drv_devmm              85428  3 drv_devdrv
drv_devmng            171881  10 drv_pcie_hdc,drv_dbg,drv_log,drv_devdrv,drv_dvp
p,drv_prof,drv_devmm
drv_platform           14683  1 drv_devmng
ipc_drv                37045  2 drv_log,drv_devmng
drv_pcie_vnic          21241  0
drv_pcie              161110  5 drv_pcie_hdc,drv_pcie_vnic,drv_devdrv,drv_devmng
,drv_devmm
drv_pca6416             8767  0
drv_e2prom             12131  0
drv_nor_flash          22787  1
drv_dfm                 9401  4 drv_devmng,drv_dvpp,drv_gmac,drv_pcie
#
```

----结束

 版权所有 © 华为技术有限公司

2 Profiling

2.1 简介

Mind Studio提供针对Host和Device多节点、多模块异构体系的高效、易用、可灵活扩展的系统化性能分析的工具Profiling（目前已经支持多卡多芯片场景），便于快速识别产品的关键性能瓶颈并提出针对性能优化的建议，最终实现产品的极致性能。

Mind Studio支持通过图形化、命令行两种方式，实现硬件和软件的性能数据的采集、分析、汇总展示，总体流程如下：

1. 配置需要采集的数据，包括硬件和软件的性能数据。

   硬件的性能数据包括：Control CPU上的PMU事件、Task Schedule CPU上的PMU事件、AI CPU上的PMU事件、外围设备的性能数据等。

   软件的性能数据包括：HIAI Engine、OME、RTS等模块的性能数据。

2. 采集性能数据。

   配置Mind Studio与Host侧的连接后，可开始采集性能数据。

3. 汇总展示性能分析的结果。

   在Mind Studio界面上以Summary、Timeline、AI CPU Function&Control CPU Function三个维度展示性能分析的结果。

   说明

   请在运行profiling工具前确认:

   1.在运行profiling侧的非root用户文件夹具备750权限。

   2.和device相连的Host侧，有权限运行编译好的用例。

   - 有访问并执行依赖库的权限（以当前用户可访问依赖库所在路径，并查询依赖库权限可被当前用户执行）
   - 查看编译用例的执行权限（在运行过流程编排用例后，可切换至~/HIAI_PROJECTS/workspace_mind_studio相应的工程路径下，查看是否可以执行用例）

2.2 全流程 Profiling（图形化方式）

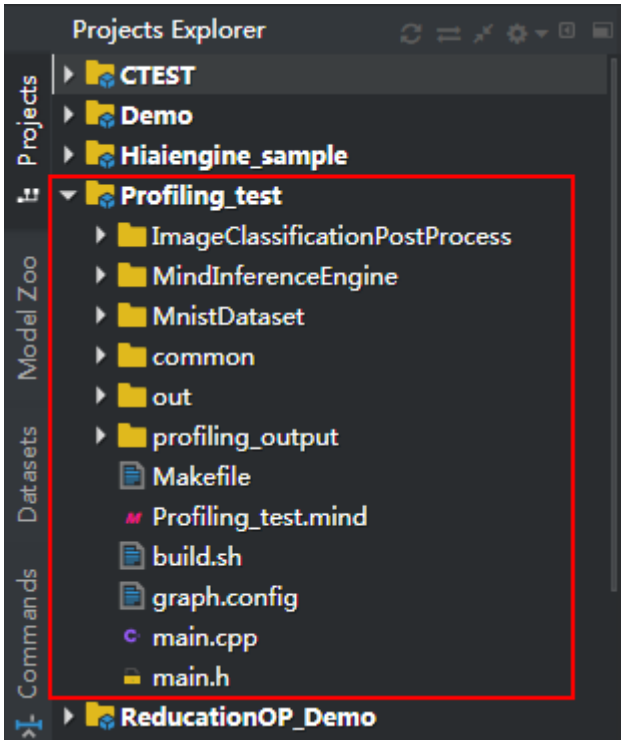
Mind Studio在流程编排的基础上，同时提供针对系统进行Profiling的能力。当需要对整网进行性能分析时，可以使用全流程Profiling功能。

 版权所有 © 华为技术有限公司

前提条件

已通过Mind Studio配置Mind工程，导入数据集和模型，并完成工程的编译运行，如图2-1所示。

图 2-1 Mind 工程

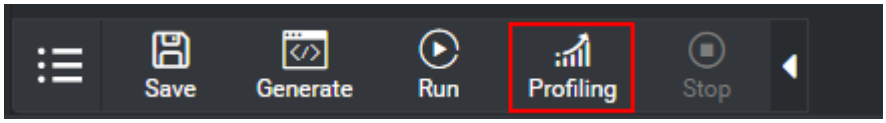


2.2.1 配置需采集的数据

步骤1 双击Mind工程下的*.mind文件。

步骤2 在右侧出现的流程编排界面中，单击“Profiling”图标，如图2-2所示。

图 2-2 Profiling 图标



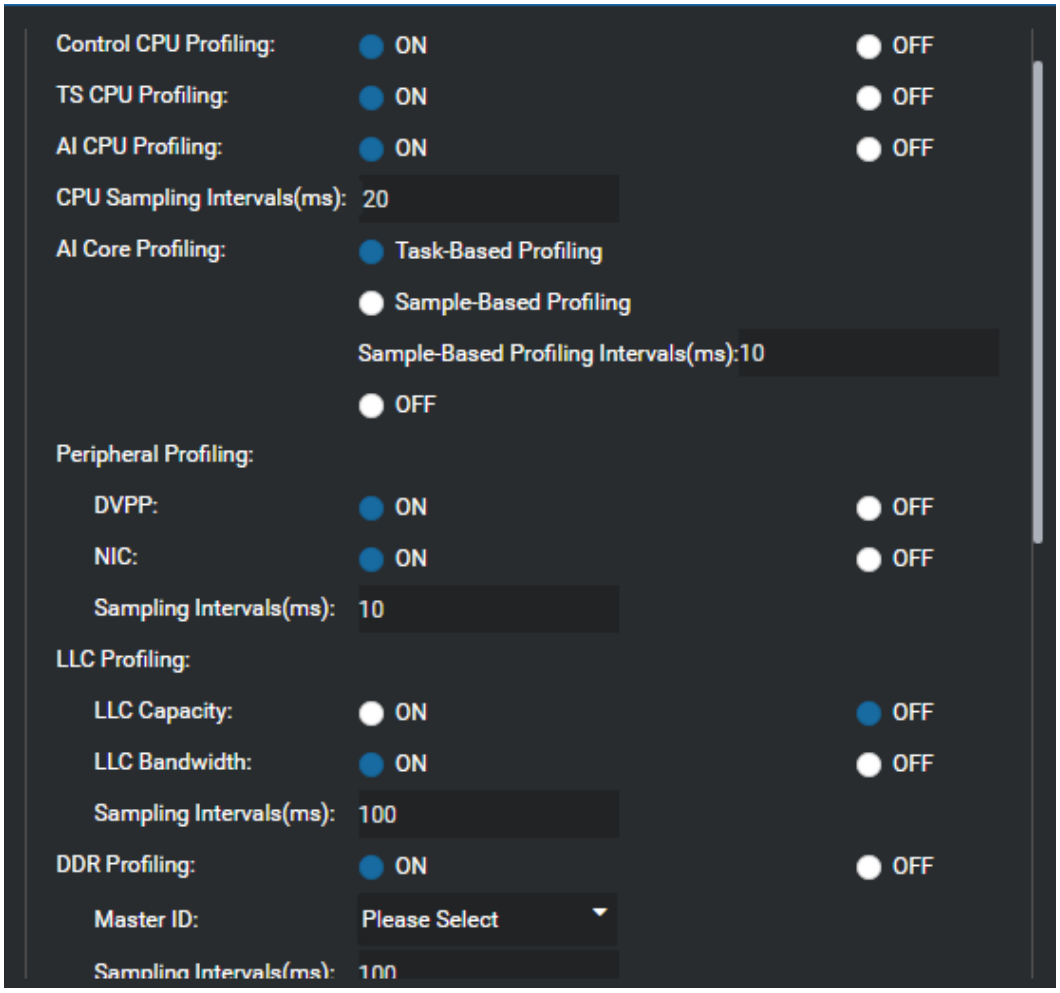
步骤3 在弹出的“Profiling”窗口中，配置Hardware、Software和Connection三个区域的配置项，包括需采集哪些模块的性能数据、采集周期等。

1. 配置Hardware区域的配置项。

   硬件可配置采集项如图2-3所示。

 版权所有 © 华为技术有限公司

图 2-3 Hardware 配置界面



界面参数说明详情见表2-1。

表 2-1 Hardware 配置参数

参数	说明
Control CPU Profiling	表示NPU的主控CPU，支持ON/OFF开关，设置为ON后，采集Control CPU上的PMU事件，当前支持的默认事件包括0x11和0x8，其中0x11表示CPU执行的Cycles数，0x8表示已执行完的Instructions数。
TS CPU Profiling	表示AI task调度CPU，支持ON/OFF开关，设置为ON后，采集TS CPU上的PMU事件，当前支持的默认事件包括0x11和0x8，其中0x11表示CPU执行的Cycles数，0x8表示已执行完的Instructions数。
AI CPU Profiling	表示处理ai算子（非ai core）的CPU，支持ON/OFF开关，设置为ON后，采集AI CPU上的PMU事件，当前支持的默认事件包括0x11和0x8，其中0x11表示CPU执行的Cycles数，0x8表示已执行完的Instructions数。

 版权所有 © 华为技术有限公司

参数	说明
CPU Sampling Intervals(ms)	配置Control CPU、TS CPU和AI CPU的PMU事件采集周期，默认值为20ms。
AI Core Profiling	负责scalar、cube、fp等ai计算的处理单元，配置AI Core采集方式，当前支持Task-Based和Sample-Based采集，当前支持的默认事件包括0x3、0x8、0x9、0xe、0x3a、0x3b、0x4a和0x49，0x3表示执行的cube指令数，0x8表示执行的vector指令数，0x9表示执行的scalar指令数，0xe指的是总指令cycle数，0x3a表示scalar访问UB读请求数，0x3b表示scalar访问UB写请求数，0x4a表示cube int类型指令数，0x49表示cube FP类型指令数，Task-Based是以task为粒度进行性能数据采集，Sample-Based是以固定的时间周期进行性能数据采集。
Peripheral Profiling	配置外设采集，当前支持的外设包括DVPP和NIC，默认采集周期是10ms。
LLC capacity	支持ON/OFF开关，表示采集LLC （Last Level Cache）的使用容量信息。与**LLC bandwidth**开关互斥，只能同时采集一种LLC数据，“Sampling Intervals”表示LLC采集周期，若有输入需确认输入是否在[100, 1000] ms之间，默认值为100ms。
LLC bandwidth	支持ON/OFF开关，表示采集LLC （Last Level Cache）的读写带宽信息。与**LLC capacity**开关互斥，只能同时采集一种LLC数据，“Sampling Intervals”表示LLC采集周期，若有输入需确认输入是否在[100, 1000] ms之间，默认值为100ms。
DDR Profiling	支持ON/OFF开关，表示采集DDR的读写带宽。配置为on为开启采集，配置为off为不采集，默认为off。 “Master ID”表示具体采集哪个Core的读写带宽，当前可选0,1,2,3,4,5,6,7。0,1,2,3对应Control CPU core ID；4,5,6,7对应AI CPU core ID。 “Sampling Intervals”表示DDR采集周期，若有输入需确认输入是否在[100, 1000] ms之间，默认值为100ms。

说明

- PMU是指Performance Monitor Unit，性能监视单元，CPU提供的一个单元，属于硬件的范畴。通过访问相关的寄存器能读取到CPU的一些性能数据。
- DVPP: Digital Vision Pre-Process
- NIC: Network Interface Controller
- 建议DDR和LLC的采样周期不大于程序执行所需的时间，否则可能会出现采集不到数据的问题。

2. 配置Software区域的配置项。

   软件可配置采集项，如图2-4所示。

 版权所有 © 华为技术有限公司

图 2-4 Software 配置界面

界面参数说明详情见表2-2。

表 2-2 Software 配置参数

参数	说明
HIAI Engine Profiling	支持ON/OFF开关，设置为ON后，支持采集Engine性能数据。
OME Profiling	支持ON/OFF开关，设置为ON后，支持采集OME性能数据。
RTS Profiling	支持ON/OFF开关，设置为ON后，采集RTS性能数据。

3. 配置Connection区域的配置项
   连接信息如图2-5，界面参数说明如表2-3所示。

图 2-5 Connection 配置界面

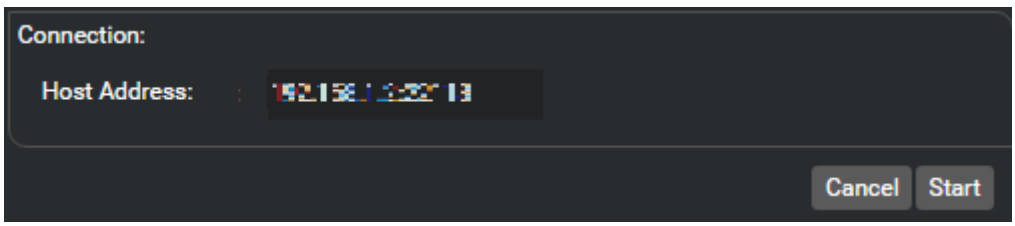


表 2-3 Connection 配置参数

参数	说明
Host Address	Host侧服务器的IP地址。

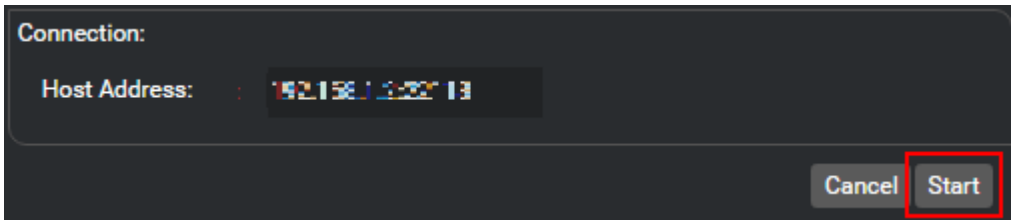
----结束

2.2.2 采集性能数据

在“Profiling”窗口，单击“Start”，采集性能数据，如图2-6所示。

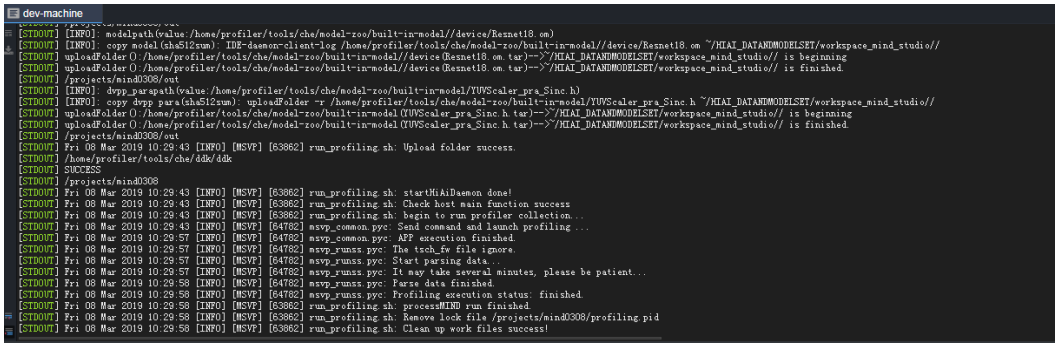
 版权所有 © 华为技术有限公司

图 2-6 采集性能数据



在dev-machine窗口查看运行结果，如图2 运行结果。在“dev-machine”窗口，如果显示**send command and launch profiling**和**Parse data finish**，且两行日志之间无任何报错，则表明采集性能数据成功。

图 2-7 运行结果



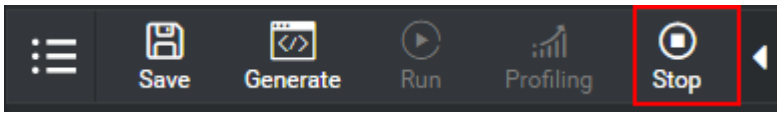
说明

1、如果miniRC场景采集所有硬件、软件数据，您需要先在Device侧的/bin目录下安装perf工具（采集Control CPU数据）、dmidecode工具（采集Host Info处的Frequency数据）、mpstat工具（采集Cpuusage数据），并赋予可执行权限。perf工具、mpstat工具、dmidecode工具是Linux内核自带的工具，如果您的Linux系统是Ubuntu 16.04及以上版本，则可以直接使用这些工具；如果您的Linux版本不是Ubuntu 16.04及以上版本，则您需要参见《开发板使用指导（Atlas 200 DK）》中的*SD卡制作*章节重新制卡。

2、需要保证AI Host上的IDE-daemon-host和Device上的IDE-daemon-device程序都已经启动，IDE-daemon-host启动的时候可以配置端口号，默认端口号是22118。

点击“Start”后可点击流程编排页面的“Stop”按钮停止采集。

图 2-8 Stop



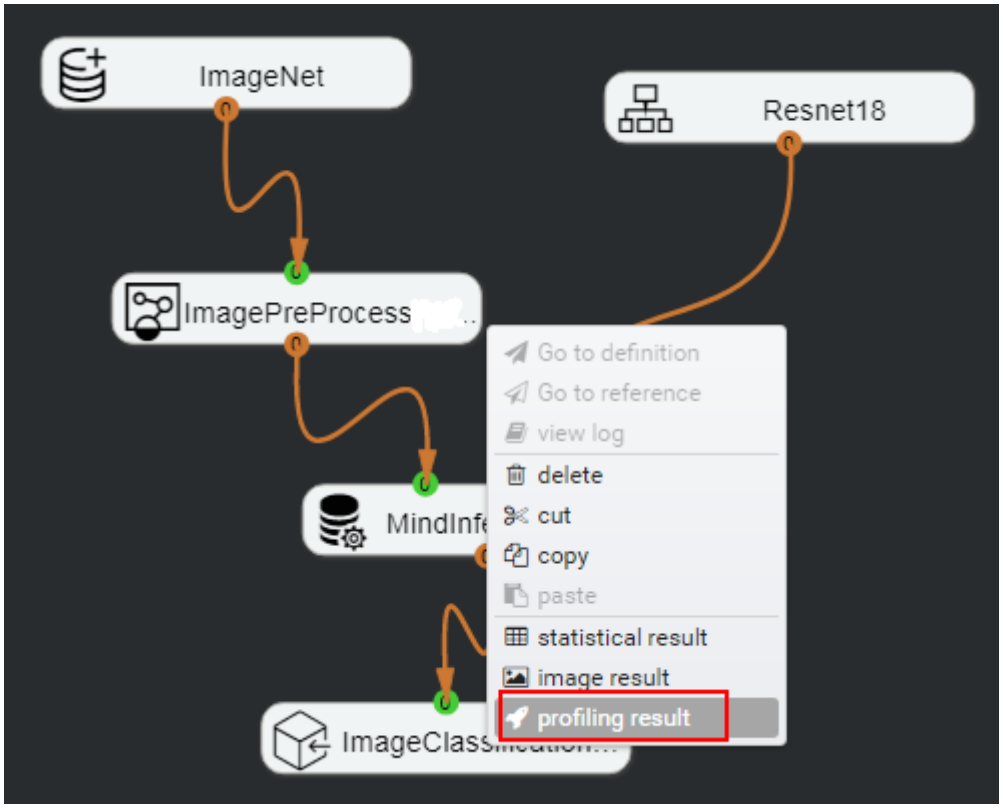
2.2.3 展示性能分析数据

本节以resnet18网络为例，说明如何展示性能分析的结果。

采集成功后，选中“ImageClassificationPostProcess”节点，单击右键，选择“profiling result”，如图2-9所示，会自动跳转进入数据分析界面，在数据分析界面，您需要输入用户名、密码登录。

 版权所有 © 华为技术有限公司

图 2-9 选择 profiling result



 版权所有 © 华为技术有限公司

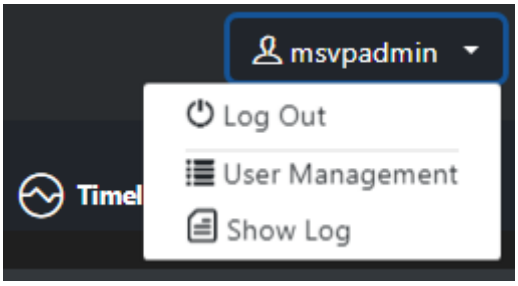
说明

1. 若是第一次登录数据分析界面，则在单击“profiling result”之后，在浏览器右上角会提示“已拦截弹出式窗口”，单击 图标，选择“始终允许”后单击“完成”。再次单击“profiling result”，系统会正常展示数据分析界面。
2. 不建议用户在浏览器页面开启两个mind studio窗口后同时对同一个工程发起profiling采集，若如此操作可能造成profiling功能异常。
3. 登录数据分析界面的默认账号为管理员，用户名为：msvpadmin，初始密码为：Admin12#$。该管理员可以查看性能分析结果，还可以创建普通用户。
4. 为了确保您的账户安全，首次登录会有修改初始密码提示，请用户首先修改初始密码，并至少90天更新一次密码，如果用户密码连续输入错误三次，页面会锁频锁住10分钟。

   修改方式如下：

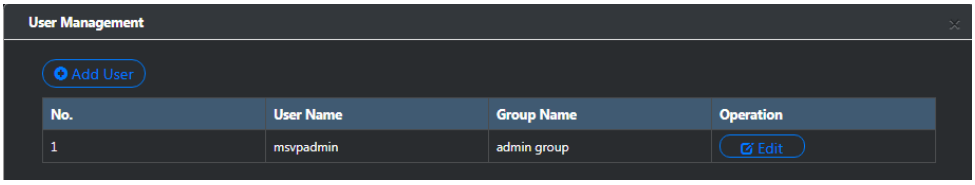
   进入结果显示页面，点击右上角所对应的User Management，如图2-10所示。

图 2-10 结果显示页面



1. 若用户名为msvpadmin，则可在弹出界面对所有用户进行修改操作，如图2-11所示。

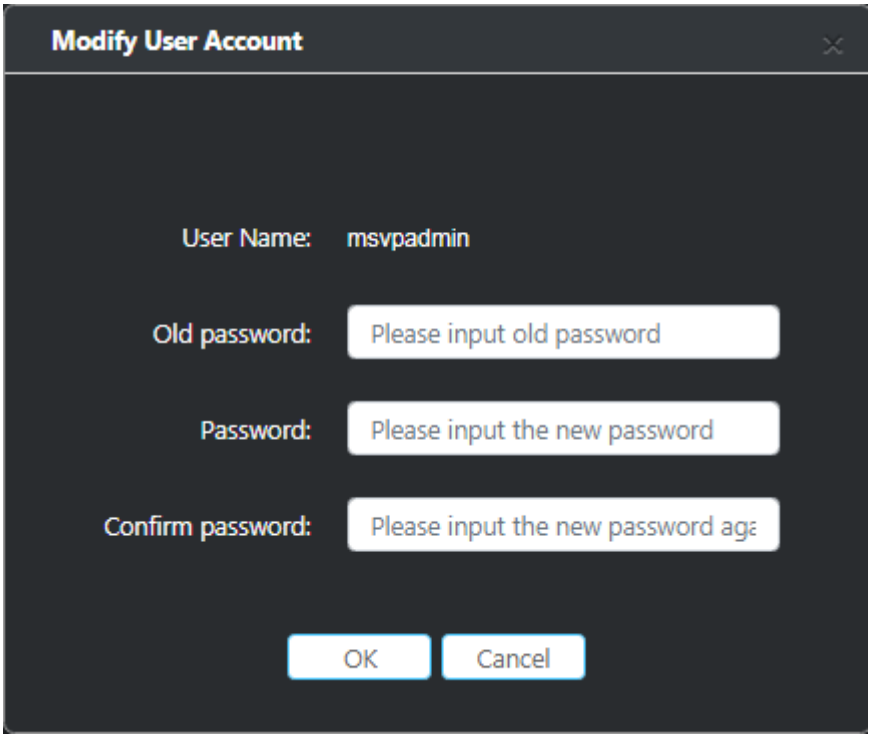
图 2-11 账户修改弹出页面



2. 若用户为普通用户则可在弹出页面修改当前用户密码，如图2-12所示。

 版权所有 © 华为技术有限公司

图 2-12 密码修改页面

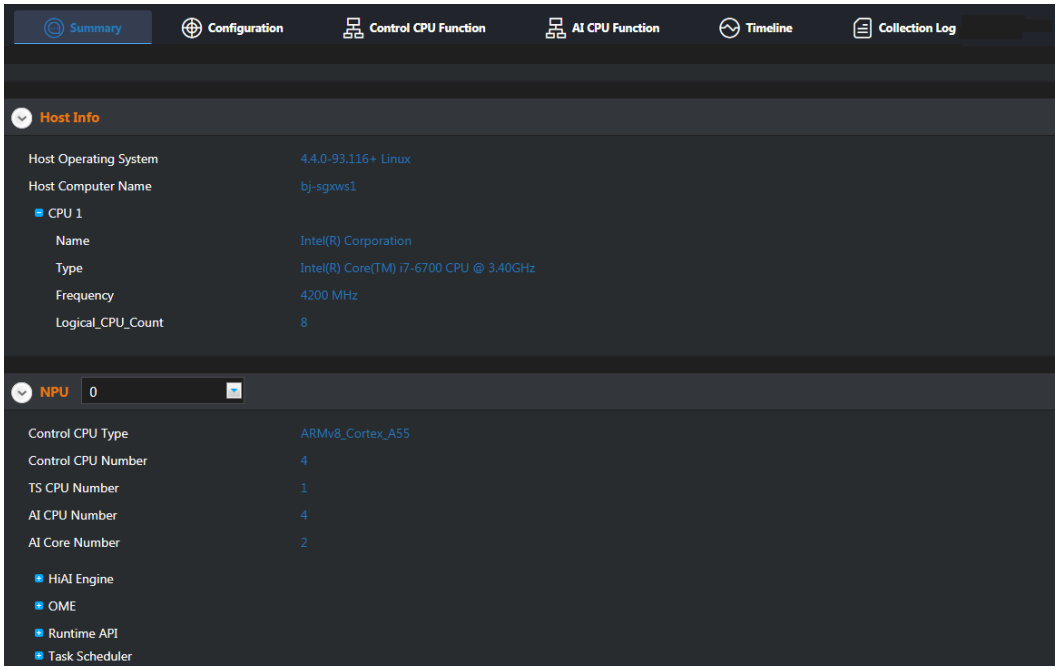


5. 以msvpadmin账号登录数据分析界面后，可以执行以下操作：
   - 在User Management界面，单击“add user”，按照界面提示添加普通用户。如果新建“group”为“user group”的用户，则该用户可以看自己的分析数据、修改自己的分析，同时也可以看其他用户的分析但不可修改；如果新建“group”为“guest group”的用户，则该用户只可以看其他用户的分析，且无法导入新的分析。
   - 在User Management界面，单击“edit”，按照界面提示修改用户密码。
   - 在User Management界面，单击“delete”，按照界面提示删除普通用户。
   - 在Show Log界面，查看所有用户的操作日志。

分析界面主要包括Summary、Timeline、AI CPU Function&Control CPU Function，如图2-13所示。

 版权所有 © 华为技术有限公司

图 2-13 分析界面



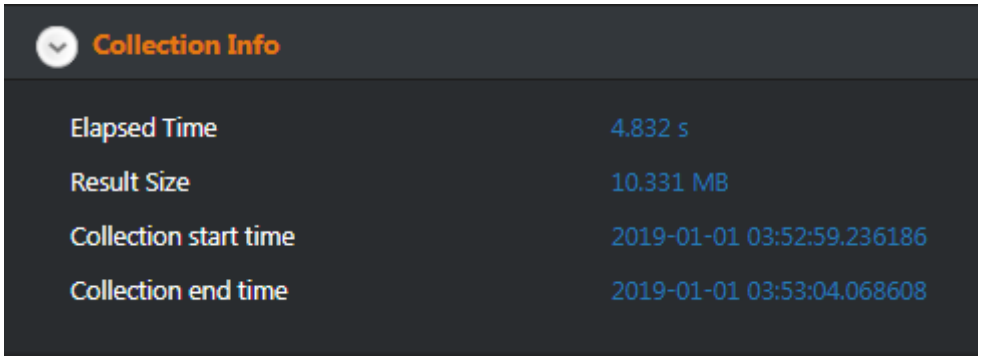
2.2.3.1 Summary 分析

Summary分析对采集的性能数据以表格的形式进行汇总呈现，主要包括如下内容。

您可以单击性能数据表格旁边的图标，将性能数据导出到本机的Excel表格中，以便查看。

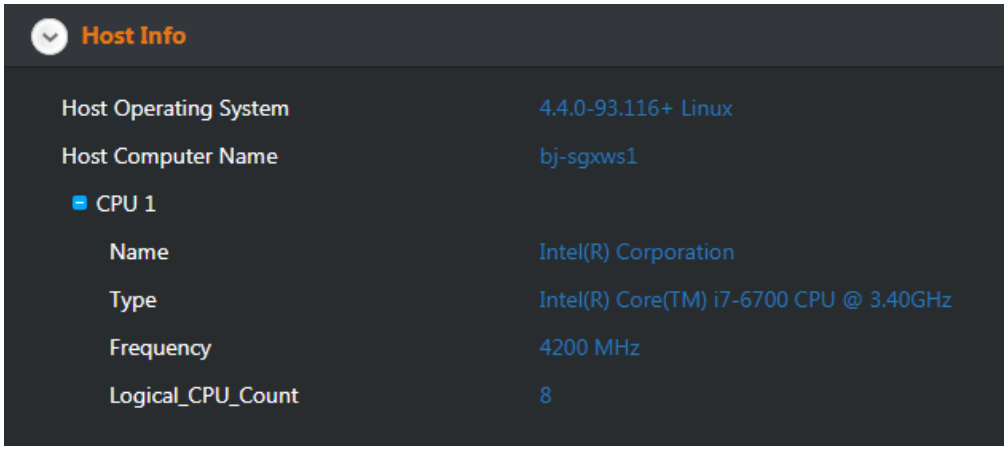
1. Collection Info：呈现本次性能数据采集的起止时间及本次采集的性能数据大小，如图2-14所示。

   图 2-14 Collection Info 界面

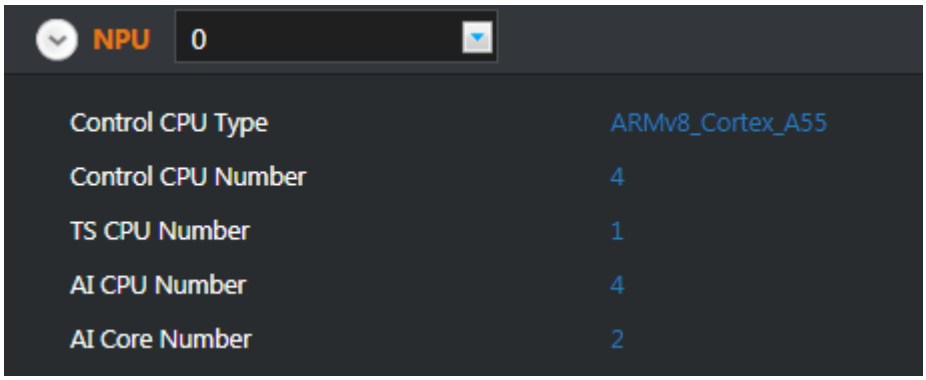
2. Host Info：呈现本次性能数据采集的Host侧基本信息，包括Host侧的操作系统和CPU信息。在开发者板形态，由于Host和Device皆位于miniRC上，所以此处采集到的数据即为miniRC的基本信息，如图2-15所示。

图 2-15 Host Info 界面



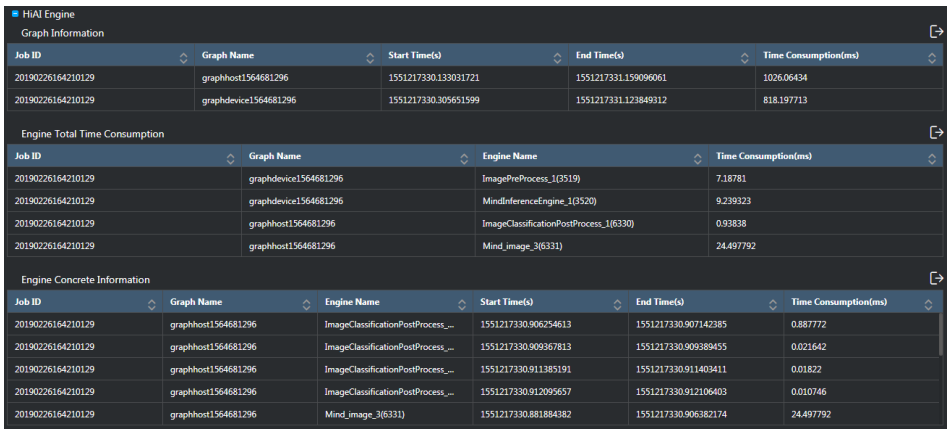
3. Device：呈现本次性能数据采集的Device侧信息，包括内容如图2-16所示。
   - Device设备基本信息：

图 2-16 Device 设备基本信息



   - HiAI Engine信息：显示每个Engine运行的起止时间，如图2-17所示。

图 2-17 HiAI Engine 信息



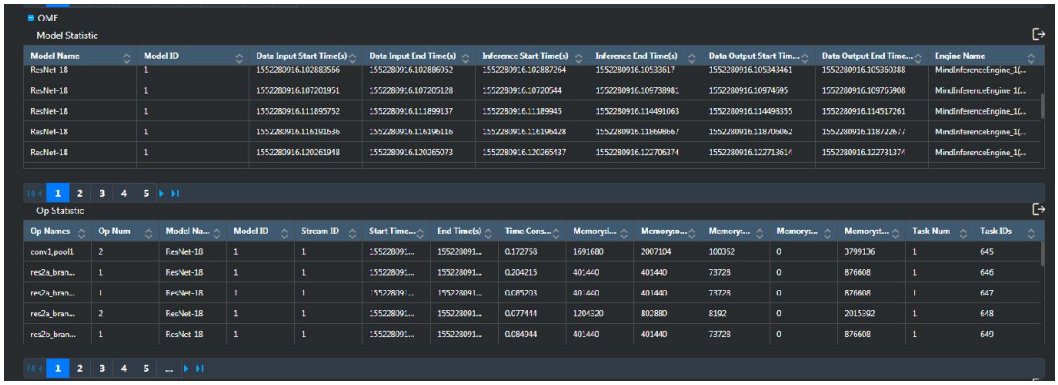
说明

HiAI Engine数据中graph name是由graph+device/host+graphID组成，Engine Name是由Engine Name（threadID）组成。

 版权所有 © 华为技术有限公司

- OME信息：模型的数据输入、推理、数据输出耗时；每个op运行的详细信息，包括op name、op type、运行时间和内存使用情况等，如图2-18所示。

图 2-18 OME 信息

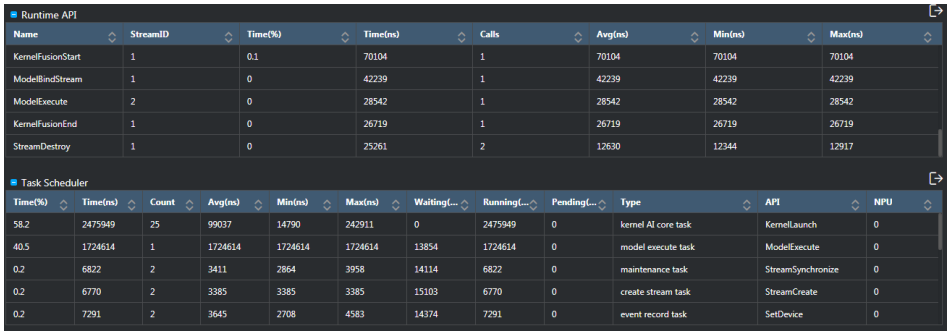


说明

当用户在进行HiAI Engine代码开发时，如果在代码中新建了用于调用推理Engine的线程，那么在编译运行工程后，再执行Profiling采集OME数据时，由于系统内部的线程与Engine无法匹配，会导致采集结果中的Model Statistic表格无法展示Engine Name数据。

- RTS信息：显示Runtime的API调用信息和Task Scheduler的任务调度信息，如图2-19所示。

图 2-19 RTS 信息



- NPU子项LLC信息：显示所有core下LLC的读写带宽及命中率，如图2-20。

图 2-20 NPU 子项 LLC 信息

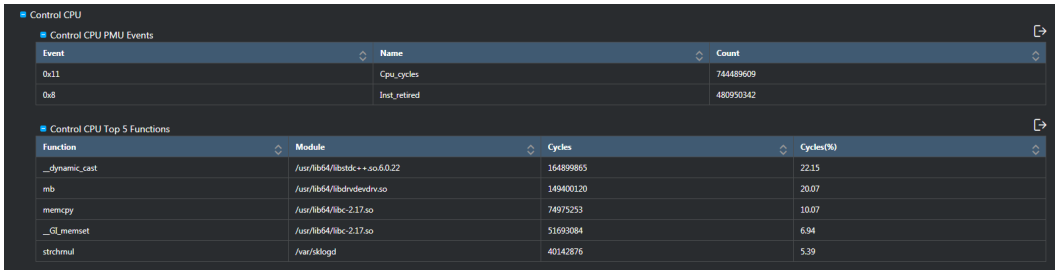
- DDR信息：显示DDR信息，如图2-21所示

图 2-21 DDR 数据

- Control CPU信息：显示采集的Control CPU的PMU事件和热点函数，如图2-22所示。

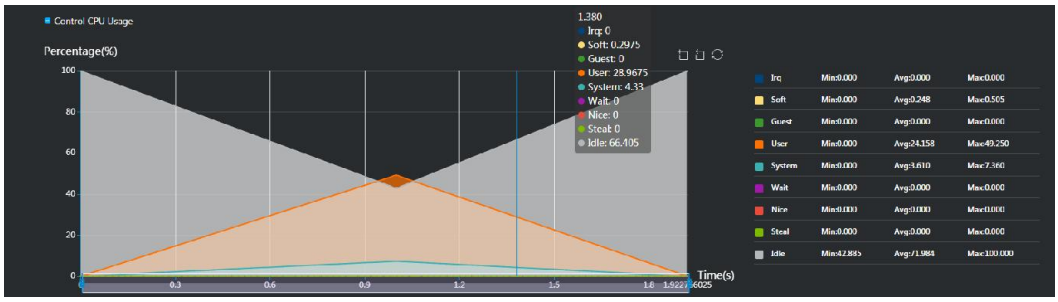
 版权所有 © 华为技术有限公司

图 2-22 Control CPU 信息



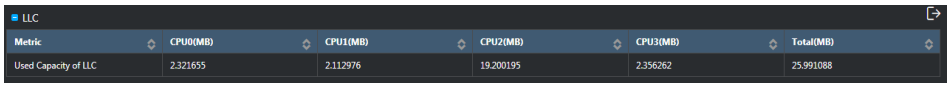
- Control CPU Usage信息：显示CPU Usage信息，如图2-23所示。
  如果采集时间少于1s，则不会展示CPU Usage图。

图 2-23 Control CPU Usage 信息



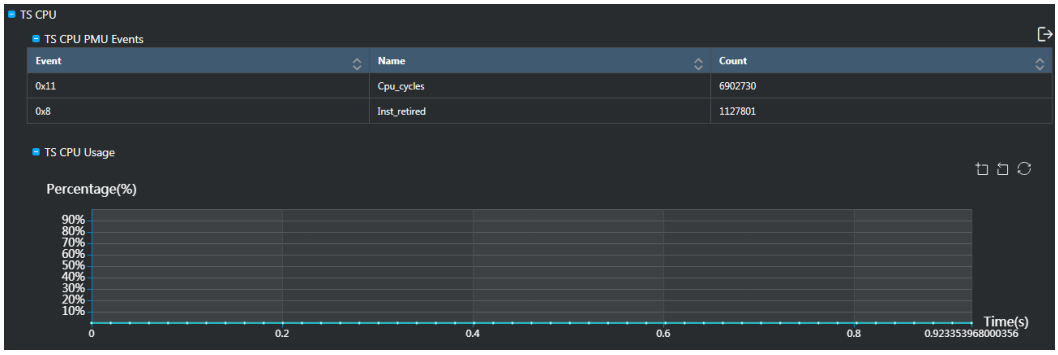
- Control CPU子项LLC信息：显示Control CPU LLC的使用总容量，如图2-24所示。

图 2-24 Control CPU 子项 LLC 信息



- TS CPU信息：显示TS CPU usage 如图2-25所示。
  如果采集时间少于1s，则不会展示TS CPU Usage图。

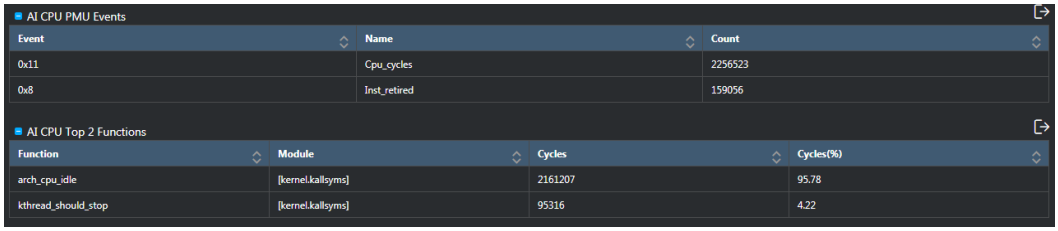
图 2-25 TS CPU 信息



- AI CPU信息：显示采集的AI CPU的PMU事件和热点函数，如图2-26所示。

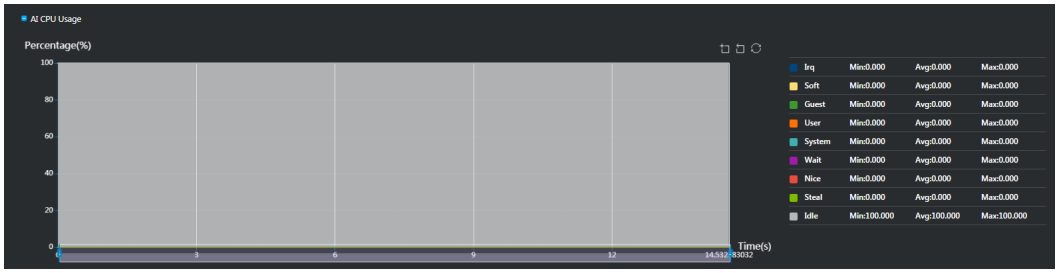
 版权所有 © 华为技术有限公司

图 2-26 AI CPU 信息



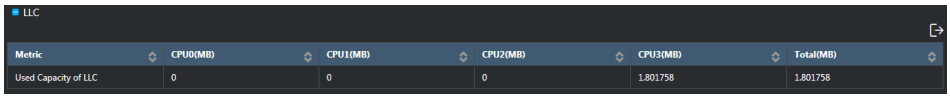
- AI CPU Usage信息，如图2-27所示。
  如果采集时间少于1s，则不会展示AI CPU Usage图。

图 2-27 AI CPU Usage 信息



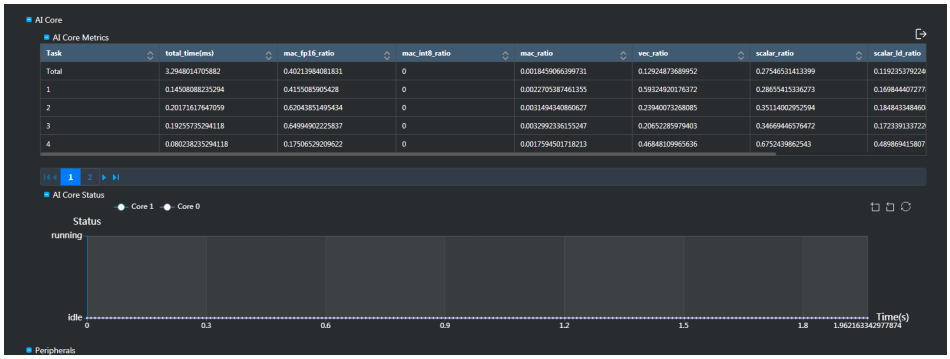
- AI CPU子项LLC信息：显示AI CPU LLC的使用总容量，如图2-28所示。

图 2-28 AI CPU 子项 LLC 信息



- AI Core信息：显示采集AI Core PMU事件计算出的指标数据以及AI Core状态信息，如图2-29所示。

图 2-29 AI Core 信息



补充说明如下：

 版权所有 © 华为技术有限公司

说明

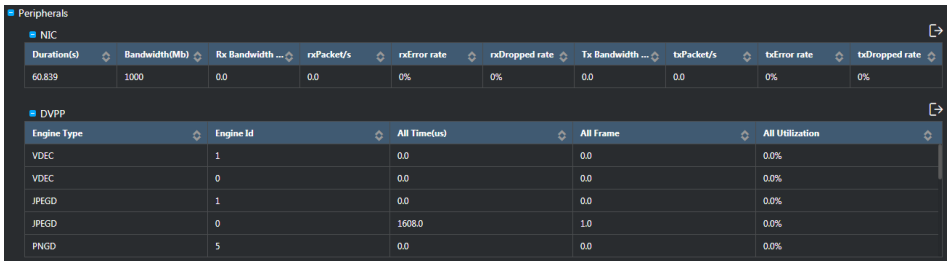
当前支持的默认事件包括0x3、0x8、0x9、0xe、0x3a、0x3b、0x4a和0x49，0x3表示执行的cube指令数，0x8表示执行的vector指令数，0x9表示执行的scalar指令数，0xe表示指的是总指令cycle数，0x3a表示scalar访问UB读请求数，0x3b表示scalar访问UB写请求数，0x4a表示cube int类型指令数，0x49表示cube FP类型指令数。

AI Core metrics计算公式如下，由于下列所述ratio存在包含关系，所以相加之和不为1。

1. “total_time”通过从PMU_TASK_CYC_CNT寄存器读取task cycles数，并在task-based场景运用公式total_time=PMU_TASK_CYC_CNT/(680*1000*((block_num +core_num-1)/core_num))，在sample-based场景运用公式total_time=PMU_TASK_CYC_CNT/(680*1000)获取。
2. “mac_fp16_ratio”代表cube fp类型指令在所有指令中占用比，通过公式mac_fp16_ratio=1.0*SUM(0x49)/SUM(PMU_TASK_CYC_CNT)获取。
3. “mac_int8_ratio”代表cube int指令在所有指令占用比，通过公式mac_int8_ratio=1.0*SUM(0x4a)/SUM(PMU_TASK_CYC_CNT)获取。
4. “vec_ratio”代表vec类型指令在所有指令占用比，通过公式vec_ratio=1.0*SUM(0x8)/SUM(PMU_TASK_CYC_CNT)获取。
5. “mac_ratio”代表cube类型指令（未包含特殊寄存器写请求）在所有指令占用比，通过公式mac_ratio=1.0*SUM(0x3)/SUM(PMU_TASK_CYC_CNT)获取。
6. “scalar_ratio”代表scalar类型指令在所有指令占用比，通过公式scalar_ratio=1.0*SUM(0x9)/SUM(PMU_TASK_CYC_CNT)获取。
7. “scalar_ld_ratio”代表scalar访问UB读请求指令数在所有指令中的占比，通过公式scalar_ld_ratio=1.0*SUM(0x3a)/SUM(PMU_TASK_CYC_CNT)获取。
8. “scalar_st_ratio”代表scalar访问UB写请求指令数在所有指令中的占比，通过公式scalar_st_ratio=1.0*SUM(0x3b)/SUM(PMU_TASK_CYC_CNT)获取。

- Peripherals信息：显示DVPP各个engine的任务时间，任务帧数及利用率信息和NIC的网络发送和接受的状态信息，如图2-30所示。

图 2-30 Peripherals 信息

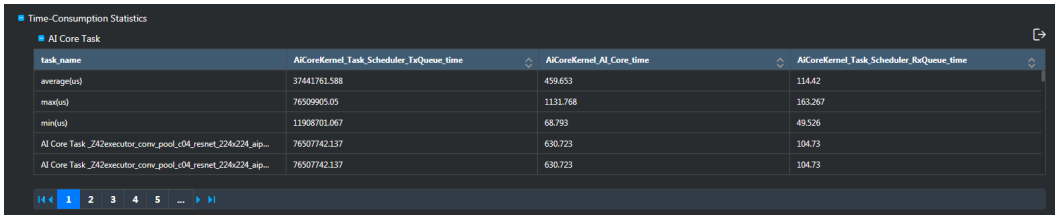


说明

pcie卡形态Device无NIC数据，采集时请关闭NIC采集开关。

- 每个AICore Task上下行数据，如图2-31所示。

图 2-31 上下行数据

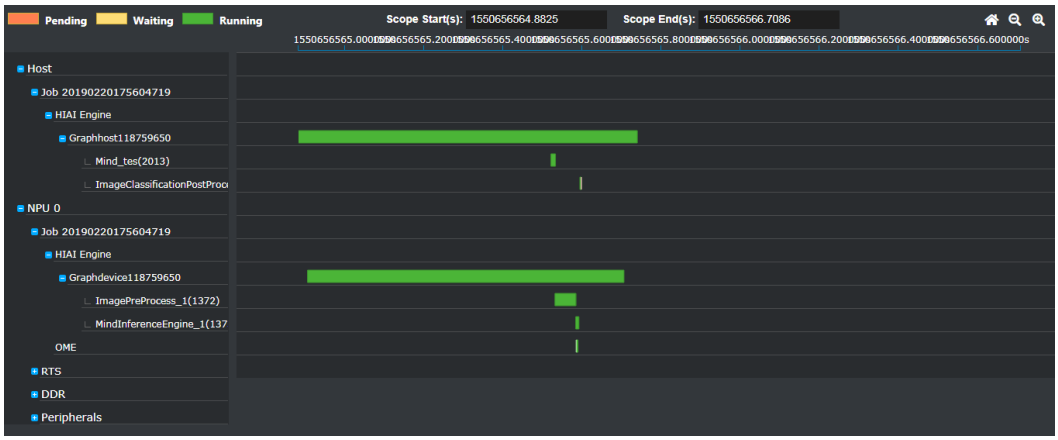


 版权所有 © 华为技术有限公司

2.2.3.2 Timeline 分析

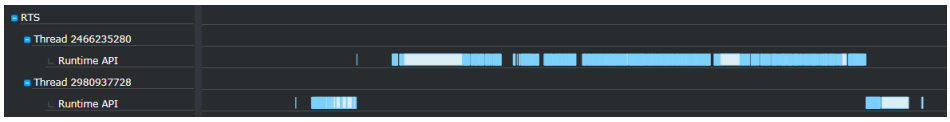
支持分析RTS、HiAI Engine、OME、CCE及Peripherals性能时序信息，如图2-32所示。

图 2-32 性能时序信息分析



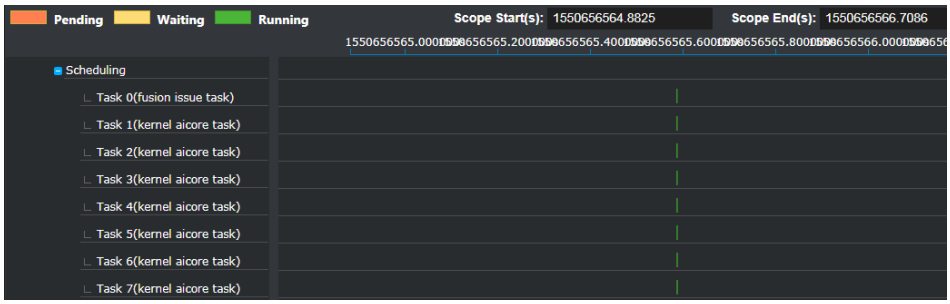
1. RTS包括如下部分：
   - Thread：每个线程调用runtime API的时序信息，如图2-33所示。

     图 2-33 Thread 分析

   - Scheduling：Task Scheduler调度的task的运行时序信息，如图2-34所示。

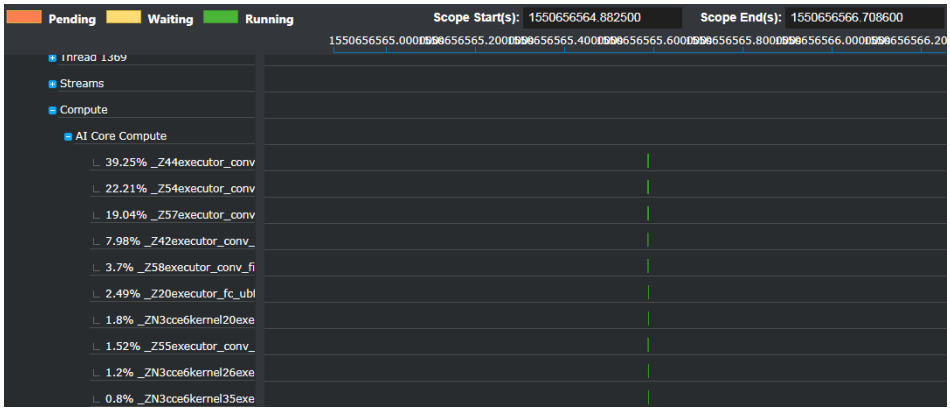
     图 2-34 Scheduling 分析

   - compute：AI Core上kernel函数的运行时序信息，单击每个kernel函数运行的矩形，可以显示kernel函数运行的起止时间和task-based模式采集的AI Core PMU值，如图2-35所示。

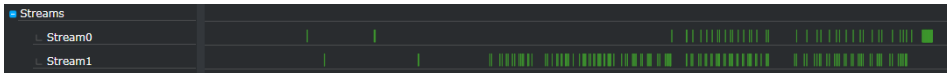
 版权所有 © 华为技术有限公司

图 2-35 compute 分析



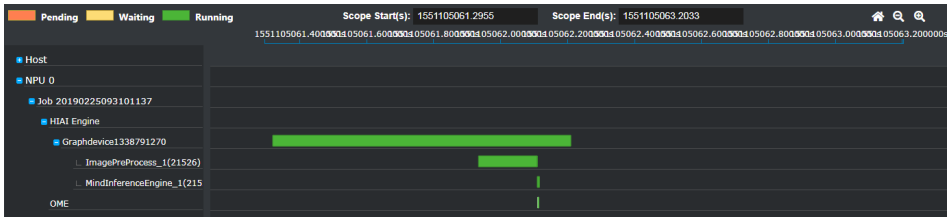
- Streams：Stream内的task运行时序信息，包括流间同步信息，如图2-36所示。

图 2-36 Streams 分析



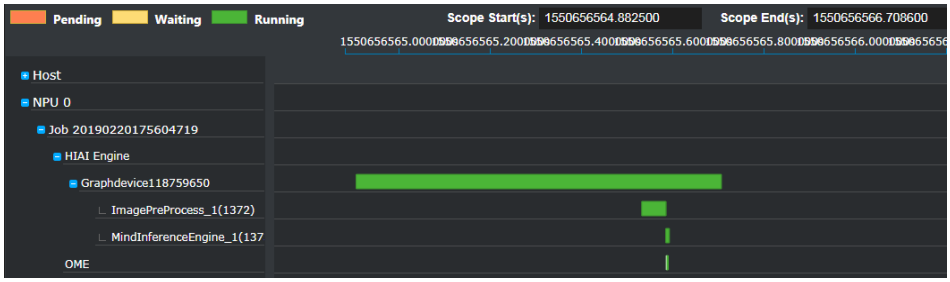
2. HiAIEngine：每个Engine运行的起止时间，如图2-37所示。

图 2-37 HiAIEngine 分析



3. OME：每个op运行的起止时间和相关信息，如图2-38所示。

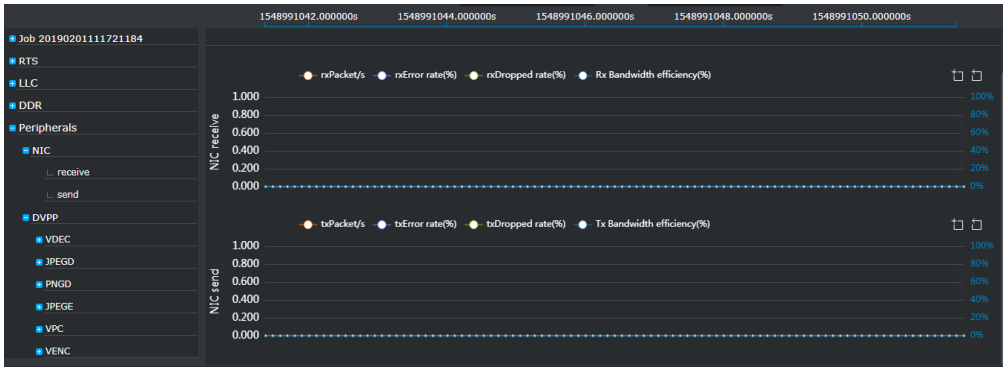
图 2-38 OME 界面



4. Peripherals：外设信息，包括DVPP各个engine的任务时间及任务帧数信息。NIC代表每个时间节点上接受和发送网络信息，如图2-39所示。

 版权所有 © 华为技术有限公司

图 2-39 Peripherals 分析界面



NIC图表各指标代表含义如下：

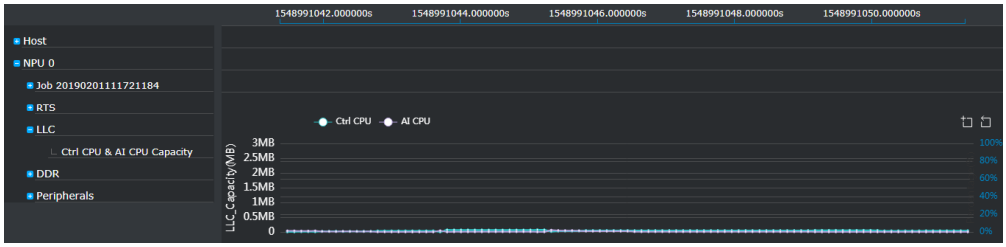
- rxPacket/s：代表每秒收包速率
- rxError rate：代表接收包错误率
- rxDropped rate：代表接收包丢包率
- Rx Bandwidth efficiency(%)：代表接收包带宽利用率
- txPacket/s：代表每秒发送包速率
- txError rate：代表发送包错误率
- txDropped rate：代表发送包丢包率
- Tx Bandwidth efficiency(%)：代表发送包带宽利用率

DVPP图表各指标代表含义如下：

- proc_time ：代表每次采样间隔中引擎使用的时间
- last_time：代表最后一次任务用的时间
- proc_frame：代表每次采样间隔中引擎处理的帧数
- last_frame：代表最后一次任务的帧数
- proc_utilization：代表时间窗利用率，即时间段累加的处理时间/运行时间
- all_utilization：代表平均利用率，即累加的处理时间/运行时间

5. LLC：显示LLC每一时刻的读写带宽以及命中率，包含Control CPU和AI CPU每一时刻使用LLC的容量，如图2-40所示。

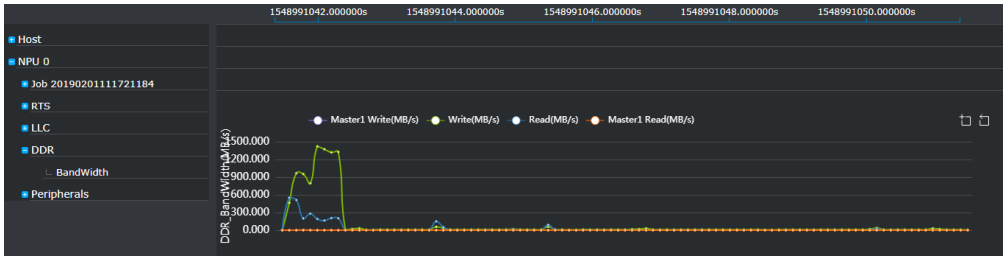
图 2-40 LLC 分析界面



6. DDR信息

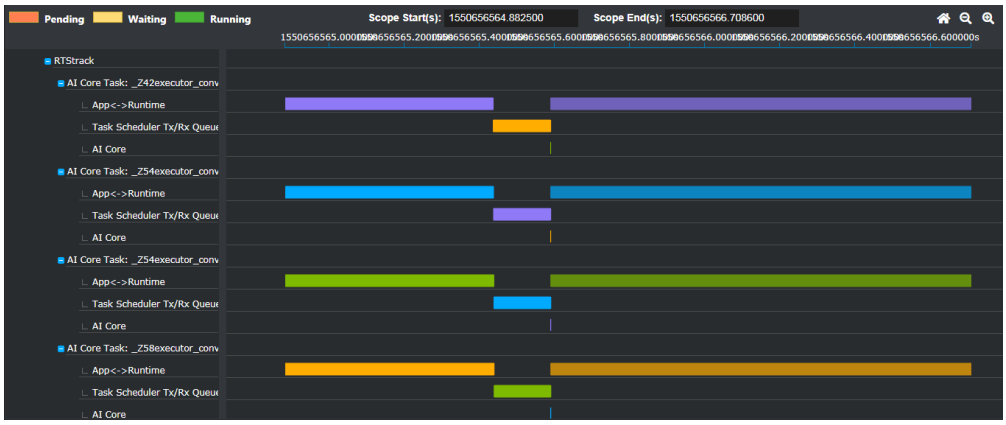
 版权所有 © 华为技术有限公司

图 2-41 DDR 分析界面



7. RTStrack：显示每个AI Core Task的上下行数据，如图2-42所示。

图 2-42 RTStrack 分析界面



说明

1. Timeline 所有图表中的时间块均可通过滚轮等比例放大缩小，也可通过图表区右上角的按钮进行放大、缩小、还原。
2. Timeline中的时间块，放大之后需鼠标左键点住拖拉，查看具体信息。

2.2.3.3 Control CPU Function 分析

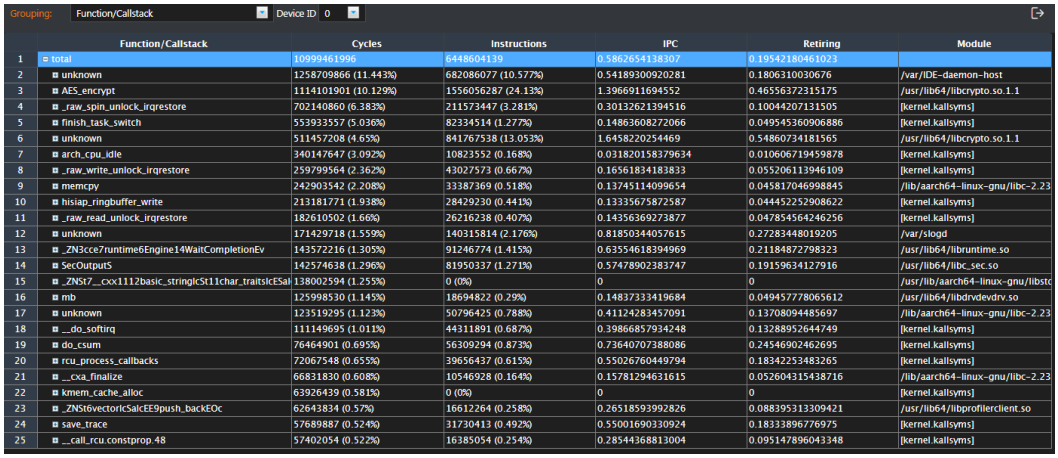
在硬件采集项配置采集Control CPU Profiling，在分析界面会生成对应的Control CPU Function页签，

默认按照cycles排序显示采集过程中Control CPU的热点函数，如图2-43所示。

您可以单击表格旁边的图标，将数据导出到本机的Excel表格中，以便查看。

 版权所有 © 华为技术有限公司

图 2-43 Control CPU Function 分析界面



Grouping: Function/Callstack Device ID 0

	Function/Callstack	Cycles	Instructions	IPC	Retiring	Module
1	total	10999461996	6448604139	0.5862654138307	0.19542180461023	
2	unknown	1258709866 (11.443%)	682086077 (10.577%)	0.54189300920281	0.1806310030676	/var/IDE-daemon-host
3	AES_encrypt	1114101901 (10.129%)	1556056287 (24.13%)	1.3966911694552	0.46556372315175	/usr/lib64/libcrypto.so.1.1
4	_raw_spin_unlock_irqrestore	702140860 (6.383%)	211573447 (3.281%)	0.30132621394516	0.10044207131505	[kernel.kallsyms]
5	finish_task_switch	553933557 (5.036%)	82334514 (1.277%)	0.14863608272066	0.0495453609068 86	[kernel.kallsyms]
6	unknown	511457208 (4.65%)	841767538 (13.053%)	1.6458220254469	0.54860734181565	/usr/lib64/libcrypto.so.1.1
7	arch_cpu_idle	340147647 (3.092%)	10823552 (0.168%)	0.031820158379634	0.010606719459878	[kernel.kallsyms]
8	_raw_write_unlock_irqrestore	259799564 (2.362%)	43027573 (0.667%)	0.16561834183833	0.055206113946109	[kernel.kallsyms]
9	memcpy	242903542 (2.208%)	33387369 (0.518%)	0.13745114099654	0.045817046998845	/lib/aarch64-linux-gnu/libc-2.23
10	hisiap_ringbuffer_write	213181771 (1.938%)	28429230 (0.441%)	0.13335675827587	0.044452252908622	[kernel.kallsyms]
11	_raw_read_unlock_irqrestore	182610502 (1.66%)	26216238 (0.407%)	0.14356369273877	0.047854564246256	[kernel.kallsyms]
12	unknown	171429718 (1.559%)	140315814 (2.176%)	0.81850344057615	0.27283448019205	/var/slogd
13	_ZN3cce7runtime6Engine14WaitCompletionEv	143572216 (1.305%)	91246774 (1.415%)	0.63554618394969	0.21184872798323	/usr/lib64/libruntime.so
14	SecOutputS	142574638 (1.296%)	81950337 (1.271%)	0.57478902383747	0.19159634127916	/usr/lib64/libc_sec.so
15	_ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESal	138002594 (1.255%)	0 (0%)	0	0	/usr/lib/aarch64-linux-gnu/libstc
16	mb	125998530 (1.145%)	18694822 (0.29%)	0.14837333419684	0.049457778065612	/usr/lib64/libdrvdevdrv.so
17	unknown	123519295 (1.123%)	50796425 (0.788%)	0.41124283457091	0.13708094485697	/lib/aarch64-linux-gnu/libc-2.23
18	__do_softirq	111149695 (1.011%)	44311891 (0.687%)	0.39866857934248	0.13288952644749	[kernel.kallsyms]
19	do_csum	76464901 (0.695%)	56309294 (0.873%)	0.73640707388086	0.24546902462695	[kernel.kallsyms]
20	rcu_process_callbacks	72067548 (0.655%)	39656437 (0.615%)	0.55026760449794	0.18342253483265	[kernel.kallsyms]
21	__cxa_finalize	66831830 (0.608%)	10546928 (0.164%)	0.15781294631615	0.052604315438716	/lib/aarch64-linux-gnu/libc-2.23
22	kmem_cache_alloc	63926439 (0.581%)	0 (0%)	0	0	[kernel.kallsyms]
23	_ZNSt6vectorIcSaIcEE9push_backEOc	62643834 (0.57%)	16612264 (0.258%)	0.26518593992826	0.088395313309421	/usr/lib64/libprofilerclient.so
24	save_trace	57689887 (0.524%)	31730413 (0.492%)	0.55001690330924	0.18333896776975	[kernel.kallsyms]
25	__call_rcu.constprop.48	57402054 (0.522%)	16385054 (0.254%)	0.28544368813004	0.095147896043348	[kernel.kallsyms]

可以通过配置“Configuration”页签下的“Control CPU Binary/Symbol Search”参数处的路径和“Control CPU C/C++ Source Search”参数处的路径；在“Control CPU Function”页签下，在“Grouping”参数处选择“Module/Fucntion/Callstack”，双击函数名称，可以跳转到对应函数的源码。

说明

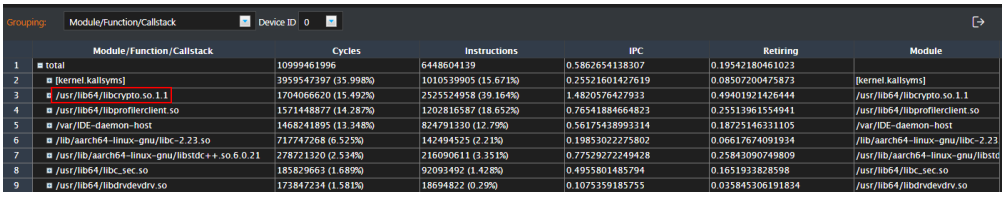
“~/tools”是默认的toolpath路径，该路径可在安装Mind Studio时由用户自定义，您可以在“scripts/env.conf”文件通过toolpath参数查看实际路径。您可以使用**find / -name 'env.conf'**命令查看script目录下的“env.conf”文件的位置。

具体步骤如下：

1. 点击分析界面的“Control CPU Function”页签，在参数处选择“Module/Fucntion/Callstack”，查看函数归属的二进制应用文件在Device侧的路径，并记录该路径。

   例如，在图2-44的示例图中，二进制应用文件“libcrypto.so.1.1”在Device侧的“/usr/lib64”目录下。

   图 2-44 查看二进制应用文件



Grouping: Module/Function/Callstack Device ID 0

	Module/Function/Callstack	Cycles	Instructions	IPC	Retiring	Module
1	total	10999461996	6448604139	0.5862654138307	0.19542180461023	
2	[kernel.kallsyms]	3959547397 (35.998%)	1010539905 (15.671%)	0.25521601427619	0.08507200475873	[kernel.kallsyms]
3	/usr/lib64/libcrypto.so.1.1	1704066620 (15.492%)	2525524958 (39.164%)	1.4820576427933	0.49401921426444	/usr/lib64/libcrypto.so.1.1
4	/usr/lib64/libprofilerclient.so	1571448877 (14.287%)	1202816587 (18.652%)	0.76541884664823	0.25513961554941	/usr/lib64/libprofilerclient.so
5	/var/IDE-daemon-host	1468241895 (13.348%)	824791330 (12.79%)	0.56175438993314	0.18725146331105	/var/IDE-daemon-host
6	/lib/aarch64-linux-gnu/libc-2.23.so	717747268 (6.525%)	142494525 (2.21%)	0.19853022275802	0.06617674091934	/lib/aarch64-linux-gnu/libc-2.23
7	/usr/lib/aarch64-linux-gnu/libstdc++.so.6.0.21	278721320 (2.534%)	216090611 (3.351%)	0.77529272249428	0.2584309074980 9	/usr/lib/aarch64-linux-gnu/libstc
8	/usr/lib64/libc_sec.so	185829663 (1.689%)	92093492 (1.428%)	0.49558014857 94	0.16519338285 98	/usr/lib64/libc_sec.so
9	/usr/lib64/libdrvdevdrv.so	173847234 (1.581%)	18694822 (0.29%)	0.10753591857 55	0.03584530619 1834	/usr/lib64/libdrvdevdrv.so

2. 以Mind Studio安装用户登录Mind Studio所在服务器，在“~/tools/che/ddk/ddk/toolchains/aarch64-linux-gcc6.3/bin”路径下可获取aarch64-linux-gnu-objdump、aarch64-linux-gnu-addr2line工具。执行如下命令。

   获取汇编指令地址：./aarch64-linux-gnu-objdump -S *二进制文件路径*

   获取源码路径：./aarch64-linux-gnu-addr2line -ie libcrypto.so.1.1 *具体汇编指令地址*

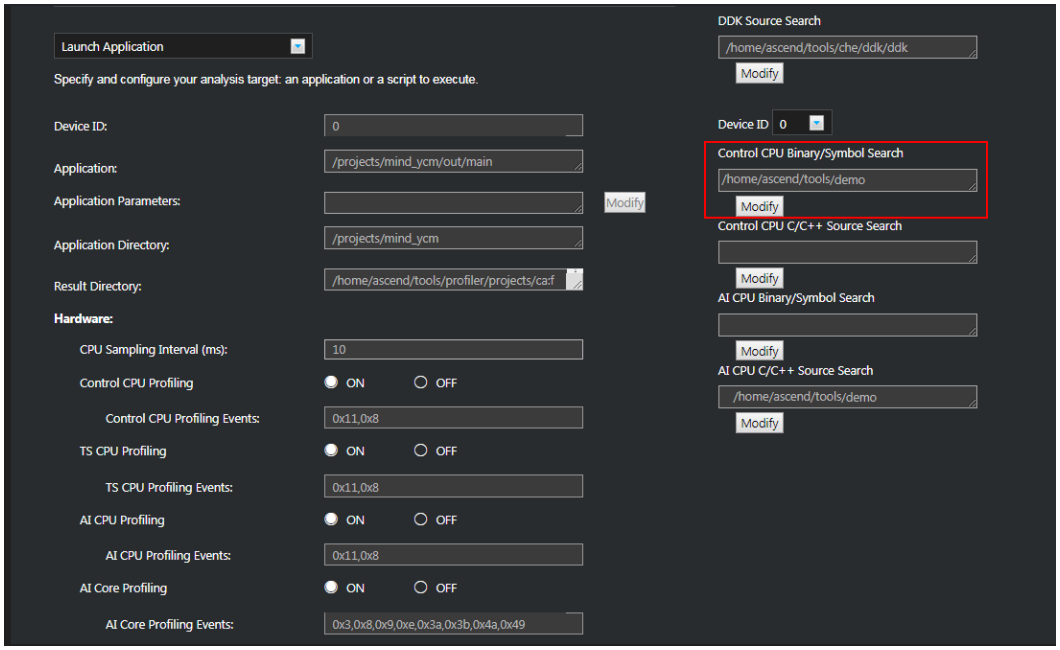
 版权所有 © 华为技术有限公司

说明

- 二进制文件需要包含符号信息（符号信息未被strip掉），即使用-g选项编译出的debug版本，否则执行addr2line命令后，无法获取源码路径。
- objdump工具和addr2line工具是第三方开源工具，您可以自行查询开源工具的其它参数说明。
- 对于开发者板上的Mind Studio，您可以在Mind Studio服务器的“/usr/bin”目录下获取aarch64-linux-gnu-objdump、aarch64-linux-gnu-addr2line工具。

3. 点击分析界面的“Configuration”页签，单击“Control CPU Binary/Symbol Search”参数处的“Modify”，此路径必须为Mind Studio安装用户家目录下的tools目录下的路径，例如“/home/ascend/tools/demo”，demo是自定义目录。

图 2-45 配置路径



4. 同理，单击“Control CPU C/C++ Source Search”参数处的“Modify”，此处路径需配置为Mind Studio所在服务器上的路径，此路径必须为Mind Studio安装用户家目录下的tools目录下的路径，例如“/home/ascend/tools/demo”，demo是自定义目录。
5. 在3配置的路径下新建1中记录的二进制文件路径；在4配置的路径下新建2中记录的源码路径。

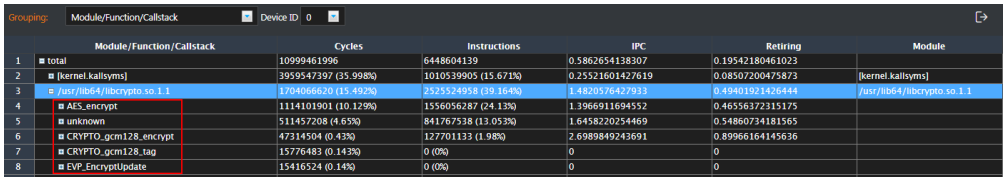
   例如，在“~/tools/demo”目录下新建“usr/lib64”目录。

   说明

   - 如果二进制文件路径和源码路径不是标准linux绝对路径格式，直接将二进制文件和源码放置在3配置路径的根目录。
   - 如果二进制文件路径和源码路径中含有非法字符（[\';*?~`!@#$%^&+=)(<>{}]| "等），可能会导致该功能无法正常使用。
6. 将Device侧的二进制文件复制到5中新建的目录下，例如“~/tools/demo/usr/lib64”。
7. 点击分析界面的“Control CPU Function”页签，双击红框中的任一函数，则可以直接跳转到对应函数的汇编代码和源码。

 版权所有 © 华为技术有限公司

图 2-46 函数跳转



Grouping: Module/Function/Callstack Device ID 0

	Module/Function/Callstack	Cycles	Instructions	IPC	Retiring	Module
1	total	10999461996	6448604139	0.5862654138307	0.19542180461023	
2	[kernel.kallsyms]	3959547397 (35.998%)	1010539905 (15.671%)	0.25521601427619	0.08507200475873	[kernel.kallsyms]
3	/usr/lib64/libcrypto.so.1.1	1704066620 (15.492%)	2525524958 (39.164%)	1.4820576427933	0.49401921426444	/usr/lib64/libcrypto.so.1.1
4	AES_encrypt	1114101901 (10.129%)	1556056287 (24.13%)	1.3966911694552	0.46556372315175	
5	unknown	511457208 (4.65%)	841767538 (13.053%)	1.6458220254469	0.54860734181565	
6	CRYPTO_gcm128_encrypt	47314504 (0.43%)	127701133 (1.98%)	2.6989849243691	0.89966164145636	
7	CRYPTO_gcm128_tag	15776483 (0.143%)	0 (0%)	0	0	
8	EVP_EncryptUpdate	15416524 (0.14%)	0 (0%)	0	0	

说明

请确认配置的路径必须为Mind Studio安装用户家目录下的tools目录下的路径。

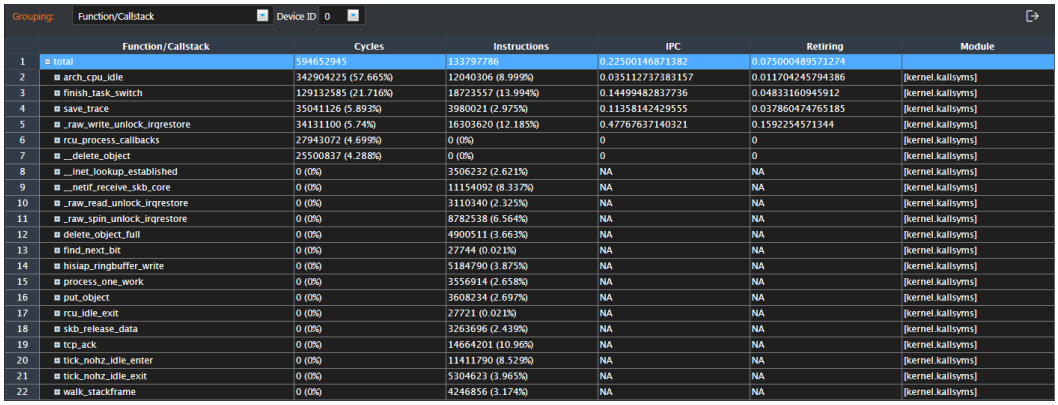
2.2.3.4 AI CPU Function 分析

在硬件采集项配置采集Al CPU Profiling，在分析界面会生成对应的Al CPU Function页签，

默认按照cycles排序显示采集过程中Al CPU的热点函数，如图2-47所示。

您可以单击表格旁边的图标，将数据导出到本机的Excel表格中，以便查看。

图 2-47 Al CPU Function 分析界面



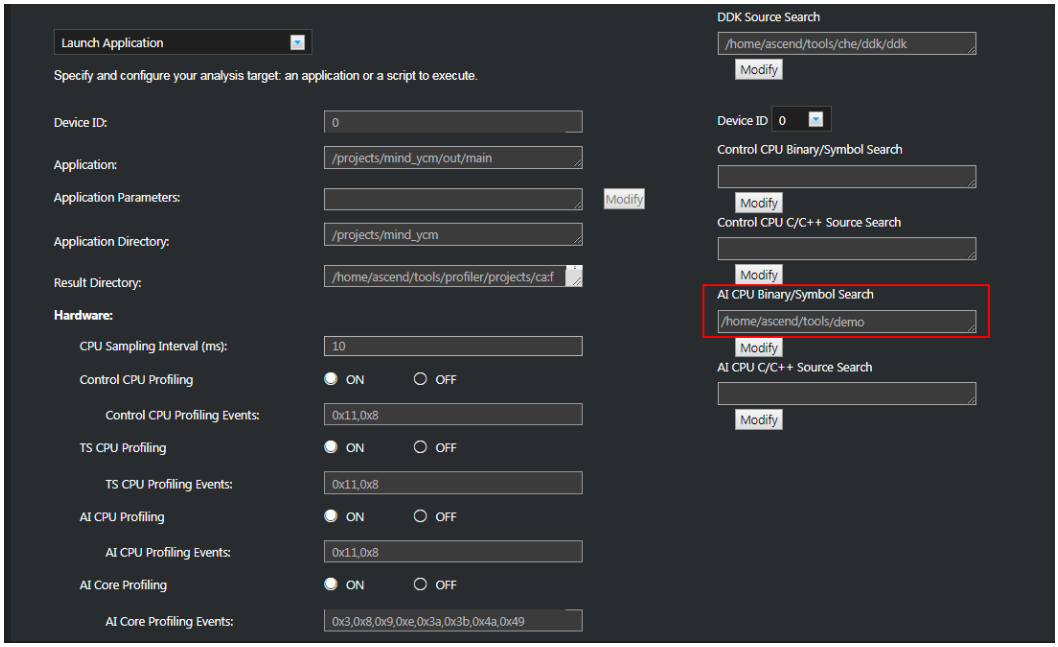
Grouping: Function/Callstack Device ID 0

	Function/Callstack	Cycles	Instructions	IPC	Retiring	Module
1	total	594652945	133797786	0.22500146871382	0.075000489571274	
2	arch_cpu_idle	342904225 (57.665%)	12040306 (8.999%)	0.035112737383157	0.011704245794386	[kernel.kallsyms]
3	finish_task_switch	129132585 (21.716%)	18723557 (13.994%)	0.14499482837736	0.04833160945912	[kernel.kallsyms]
4	save_trace	35041126 (5.893%)	3980021 (2.975%)	0.11358142429555	0.037860474765185	[kernel.kallsyms]
5	_raw_write_unlock_irqrestore	34131100 (5.74%)	16303620 (12.185%)	0.47767637140321	0.1592254571344	[kernel.kallsyms]
6	rcu_process_callbacks	27943072 (4.699%)	0 (0%)	0	0	[kernel.kallsyms]
7	__delete_object	25500837 (4.288%)	0 (0%)	0	0	[kernel.kallsyms]
8	__inet_lookup_established	0 (0%)	3506232 (2.621%)	NA	NA	[kernel.kallsyms]
9	__netif_receive_skb_core	0 (0%)	11154092 (8.337%)	NA	NA	[kernel.kallsyms]
10	_raw_read_unlock_irqrestore	0 (0%)	3110340 (2.325%)	NA	NA	[kernel.kallsyms]
11	_raw_spin_unlock_irqrestore	0 (0%)	8782538 (6.564%)	NA	NA	[kernel.kallsyms]
12	delete_object_full	0 (0%)	4900511 (3.663%)	NA	NA	[kernel.kallsyms]
13	find_next_bit	0 (0%)	27744 (0.021%)	NA	NA	[kernel.kallsyms]
14	hisiap_ringbuffer_write	0 (0%)	5184790 (3.875%)	NA	NA	[kernel.kallsyms]
15	process_one_work	0 (0%)	3556914 (2.658%)	NA	NA	[kernel.kallsyms]
16	put_object	0 (0%)	3608234 (2.697%)	NA	NA	[kernel.kallsyms]
17	rcu_idle_exit	0 (0%)	27721 (0.021%)	NA	NA	[kernel.kallsyms]
18	skb_release_data	0 (0%)	3263696 (2.439%)	NA	NA	[kernel.kallsyms]
19	tcp_ack	0 (0%)	14664201 (10.96%)	NA	NA	[kernel.kallsyms]
20	tick_nohz_idle_enter	0 (0%)	11411790 (8.529%)	NA	NA	[kernel.kallsyms]
21	tick_nohz_idle_exit	0 (0%)	5304623 (3.965%)	NA	NA	[kernel.kallsyms]
22	walk_stackframe	0 (0%)	4246856 (3.174%)	NA	NA	[kernel.kallsyms]

通过配置“Configuration”页签下的“Al CPU Binary/Symbol Search”参数处的路径，然后在“Al CPU Function”页签下，在“Grouping”参数处选择“Module/Fucntion/Callstack”，双击函数名称，可以跳转到对应函数的源码。具体配置方法请参见**2.2.3.3 Control CPU Function分析**。

 版权所有 © 华为技术有限公司

图 2-48 Configuration 页签配置页面



说明

- 请确认配置的路径必须为Mind Studio安装用户家目录下的tools目录下的路径。
- “Al CPU C/C++ Source Search”参数暂无使用场景。

2.3 全流程 Profiling（命令行方式）

若使用命令行方式分析性能数据，您可以Mind Studio安装用户登录Mind Studio服务器，执行“~/tools/profiler/analysis/msvp/host/bin64/hiprof.pyc”脚本中的命令采集性能数据，采集成功后，可直接在命令行窗口查看性能分析结果。

说明

“~/tools”是默认的toolpath路径，该路径可在安装Mind Studio时由用户自定义，您可以在“scripts/env.conf”文件通过toolpath参数查看实际路径。您可以使用**find / -name 'env.conf'**命令查看script目录下的“env.conf”文件的位置。

采集性能数据

以下采集数据的命令，均以流程编排类型的性能数据采集为例。

步骤1 以Mind Studio安装用户登录到Mind Studio的后台服务器。

步骤2 切换至“hiprof.pyc”脚本所在目录，如“~/tools/profiler/analysis/msvp/host/bin64”。

步骤3 执行以下命令，分别在sample-based模式下、task-based 模式下采集软件、硬件模块数据。

在以下命令中，“*_dir”参数值需要根据实际情况修改目录且Mind Studio安装用户可访问该目录，“--ai_core_profiling_mode=sample-based”表示sample-based 模式。“x.x.x.x”需替换为Host所在服务器的IP地址。

执行命令后，命令中的配置参数默认保存在“sample.ini”文件中，该文件保存在result_dir参数设置的目录下，关于配置参数的详细解释请参见**sample.ini文件**。

- 在sample-based 模式下，采集所有软件、硬件模块数据。

  python hiprof.pyc --ip_address=x.x.x.x --ddk_dir=~/tools/che/ddk/ddk --app=/projects/MIND/out/main --app_dir=/projects/MIND/ --umode=MIND --result_dir=/home/ascend/tools/out --peripheral_profiling=nic,dvpp --ai_cpu_profiling=on --RTS_Profiling=on --ai_core_profiling_mode=sample-based --ai_core_profiling=on --HIAI_Engine_Profiling=on --OME_Profiling=on --ctrl_cpu_profiling=on --profiling_mode=online --llc_bandwidth=on --ddr_profiling=on

- 在task-based 模式下，采集所有软件、硬件模块数据。

  python hiprof.pyc --ip_address=x.x.x.x --ddk_dir=~/tools/che/ddk/ddk --app=/projects/MIND/out/main --app_dir=/projects/MIND/ --umode=MIND --result_dir=/home/ascend/tools/out --peripheral_profiling=nic,dvpp --ai_cpu_profiling=on --RTS_Profiling=on --ai_core_profiling_mode=task-based --ai_core_profiling=on --HIAI_Engine_Profiling=on --OME_Profiling=on --ctrl_cpu_profiling=on --profiling_mode=online --llc_bandwidth=on --ddr_profiling=on

- 在task-based 模式下，执行带参数的命令采集profiling数据。

  python hiprof.pyc --ip_address=x.x.x.x --ddk_dir=~/tools/che/ddk/ddk --app_dir=/projects/MIND --umode=MIND --result_dir=/home/ascend/tools/out --ai_cpu_profiling=on --RTS_Profiling=on --ai_core_profiling=on --ctrl_cpu_profiling=on -- /projects/MIND/perfor/main Performace_Freespace_Road_FP16_1B_AIPP.json

说明

在“result_dir”处配置的路径用于存放采集结果，需确保Mind Studio安装用户对此路径有可读可写的权限，建议将该路径配置为“~/tools/”下的目录或者“/projects/”。

----结束

展示性能分析结果

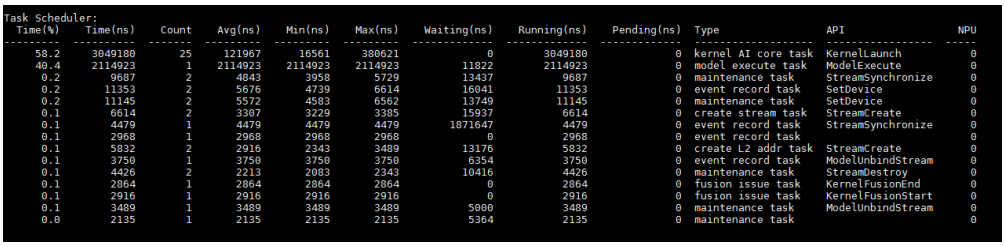
1. 显示Runtime的API调用信息和Task Scheduler的任务调度信息。

图 2-49 Runtime 数据采集结果

```
Profiling Results:
RunTime API:
Name              StreamID      Time(%)    Time(ns)    Calls    Avg(ns)    Min(ns)    Max(ns)
----------------  ----------  ---------  ----------  -------  ---------  ---------  ---------
MemCpy                            47.7    68580814      120     571506       2813   13772757
KernelLaunch      1               25.7    36899107       25    1475964     405625   25644577
DevMalloc                          8.6    12391140       52     238291     171146     685834
HostMalloc                         7.1    10276142      191      53801      32292     168907
DevFree                            3.5     5099423       52      98065      74739     520832
StreamSynchronize 2                1.9     2744115        2    1372057     490365    2253750
DeviceReset       0                1.8     2659062        2    1329531      34114    2624948
StreamCreate      1                1.4     2030938        2    1015469     860730    1170208
ManagedMemAlloc                    1.0     1477760       25      59110      52813      83228
HostFree                           0.3      404371      191       2117       1667       4375
ModelUnbindStream 1                0.2      316093        1     316093     316093     316093
ModelDestroy                       0.2      236563        1     236563     236563     236563
KernelFusionStart 1                0.2      226354        1     226354     226354     226354
SetDevice                          0.1      130989        2      65494      46875      84114
ModelCreate                        0.1       90000        1      90000      90000      90000
ManagedMemFree                     0.0       71510       25       2860       2500       5000
ModelBindStream   1                0.0       52500        1      52500      52500      52500
KernelFusionEnd   1                0.0       27552        1      27552      27552      27552
StreamDestroy     1                0.0       26353        2      13176      12239      14114
ModelExecute      2                0.0       21302        1      21302      21302      21302
```

 版权所有 © 华为技术有限公司

图 2-50 Task Scheduler 任务调度信息

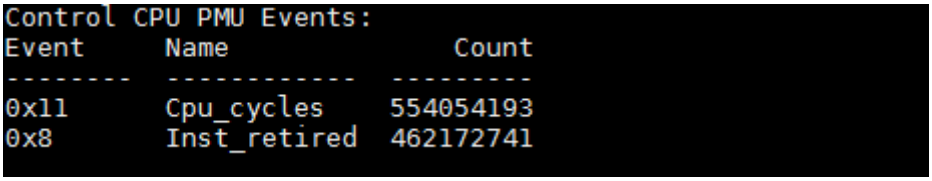


Task Scheduler:

Time(%)	Time(ns)	Count	Avg(ns)	Min(ns)	Max(ns)	Waiting(ns)	Running(ns)	Pending(ns)	Type	API	NPU
58.2	3049180	25	121967	16561	380621	0	3049180	0	kernel AI core task	KernelLaunch	0
40.4	2114923	1	2114923	2114923	2114923	11822	2114923	0	model execute task	ModelExecute	0
0.2	9687	2	4843	3958	5729	13437	9687	0	maintenance task	StreamSynchronize	0
0.2	11353	2	5676	4739	6614	16041	11353	0	event record task	SetDevice	0
0.2	11145	2	5572	4583	6562	13749	11145	0	maintenance task	SetDevice	0
0.1	6614	2	3307	3229	3385	15937	6614	0	create stream task	StreamCreate	0
0.1	4479	1	4479	4479	4479	1871647	4479	0	event record task	StreamSynchronize	0
0.1	2968	1	2968	2968	2968	0	2968	0	event record task		0
0.1	5822	2	2916	2343	3489	13176	5822	0	create L2 addr task	StreamCreate	0
0.1	3750	1	3750	3750	3750	6354	3750	0	event record task	ModelUnbindStream	0
0.1	4426	2	2213	2083	2343	10416	4426	0	maintenance task	StreamDestroy	0
0.1	2864	1	2864	2864	2864	0	2864	0	fusion issue task	KernelFusionEnd	0
0.1	2916	1	2916	2916	2916	0	2916	0	fusion issue task	KernelFusionStart	0
0.1	3489	1	3489	3489	3489	5000	3489	0	maintenance task	ModelUnbindStream	0
0.0	2135	1	2135	2135	2135	5364	2135	0	maintenance task		0

2. Control CPU pmu events

图 2-51 Control CPU 采集事件结果

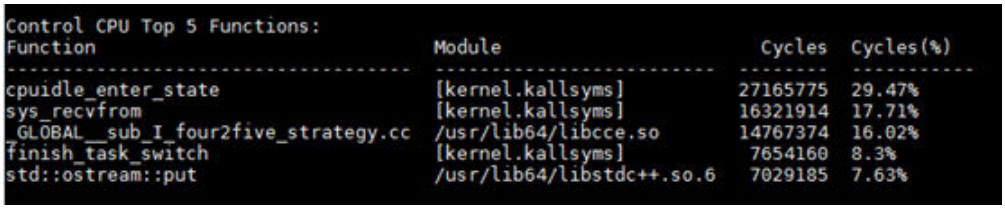


Control CPU PMU Events:

Event	Name	Count
0x11	Cpu_cycles	554054193
0x8	Inst_retired	462172741

3. Control CPU Top Functions

图 2-52 Control CPU 热点函数

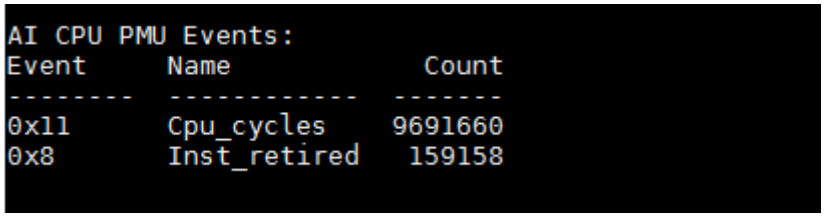


Control CPU Top 5 Functions:

Function	Module	Cycles	Cycles(%)
cpuidle_enter_state	[kernel.kallsyms]	27165775	29.47%
sys_recvfrom	[kernel.kallsyms]	16321914	17.71%
_GLOBAL__sub_I_four2five_strategy.cc	/usr/lib64/libcce.so	14767374	16.02%
finish_task_switch	[kernel.kallsyms]	7654160	8.3%
std::ostream::put	/usr/lib64/libstdc++.so.6	7029185	7.63%

4. AI CPU PMU Events

图 2-53 AI CPU 采集事件结果统计

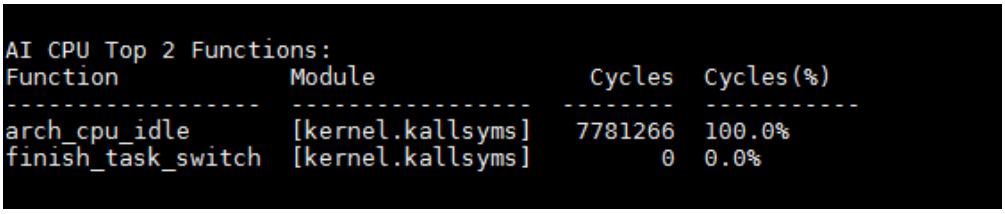


AI CPU PMU Events:

Event	Name	Count
0x11	Cpu_cycles	9691660
0x8	Inst_retired	159158

5. AI CPU Top Function

图 2-54 AI CPU 热点函数



AI CPU Top 2 Functions:

Function	Module	Cycles	Cycles(%)
arch_cpu_idle	[kernel.kallsyms]	7781266	100.0%
finish_task_switch	[kernel.kallsyms]	0	0.0%

6. TS CPU PMU Events

 版权所有 © 华为技术有限公司

图 2-55 TS CPU 采集事件结果统计

```
TS CPU PMU Events:
Event     Name              Count
--------  ------------    --------
0x11      Cpu_cycles      4234079
0x8       Inst_retired     959300
```

7. AI Core数据信息

图 2-56 AI Core 数据信息

```
AI Core Metrics:
Task      total_time(ms)    mac_fp16_ratio    mac_int8_ratio          mac_ratio          vec_ratio     scalar_ratio    scalar_ld_ratio    scalar_st_ratio
------  ----------------  ----------------  ----------------  -----------------  -----------------  --------------  -----------------  -----------------
Total   3.92373529412      0.329821087087               0  0.00145847967693   0.107064303300  0.216119147395  0.0911957015035  0.0257077136
1       0.144963235294     0.415845802688               0  0.00227238143546   0.279604362161  0.118224702004  0.0339335531321  0.00708090286584
2       0.0850602941176    0                            0  0                  0.444840165281  0.103940111685  0.0730450718349  0.0122404522743
3       0.222305882353     0.562974429197               0  0.00285773821927   0.217227795473  0.31684180236   0.165927425356   0.0368727611903
4       0.218229411765     0.573490614671               0  0.00291111987143   0.182228017507  0.303949217131  0.150256239197   0.0375615327855
5       0.111608823529     0.125227290695               0  0.000632461064116  0.196062929876  0.262925922998  0.189817376868   0.0920626136453
6       0.0296602941176    0                            0  0                  0.341315880807  0.421339679706  0.358917150082   0.0173037830334
7       0.224166176471     0.558304304186               0  0.00283403200094   0.215425793627  0.314124894216  0.164708429277   0.0365668851233
8       0.233560294118     0.535846846932               0  0.002720034756     0.170266620673  0.283846960267  0.140506795365   0.0350960040045
9       0.0293852941176    0                            0  0                  0.344510059053  0.426083475128  0.363176859173   0.0174657191472
10      0.119229411765     0.262418826664               0  0.00133207526225   0.0572792362768 0.118924719247  0.0525429686777  0.0148748404285
11      0.142307352941     0.439725326678               0  0.00223210825725   0.0338226403981 0.1553175329    0.0621270131601  0.0121680901524
12      0.0822897058824    0.0422463476746              0  0.000214448465353  0.0584908189251 0.0802305321003 0.0475539471921  0.0211767859536
13      0.01785            0                            0  0                  0.284396111386  0.300378975119  0.241308288021   0.0183720547042
14      0.141454411765     0.442379066213               0  0.00224557901631   0.0482799488507 0.15623927892   0.0625643264822  0.0124286560833
15      0.13348235294      0.468799576943               0  0.00237969328398   0.036058963514  0.165471311475  0.066308989952   0.0129991591775
16      0.0184029411765    0                            0  0                  0.275851046828  0.293511267381  0.218315486655   0.0544190506633
17      0.113955882353     0.290592334495               0  0.000696864111498  0.0292682926829 0.125687185443  0.0440572977158  0.0154277971351
18      0.126464705882     0.523695731803               0  0.00125586506428   0.0210357398266 0.265161954266  0.0639793479967  0.0317803631543
19      0.0490161764706    0.0750641285271              0  0.000180009900545  0.0542729850142 0.146753071419  0.0794293686153  0.0361819900095
20      0.00992647058824   0                            0  0                  0.256592592593  0.208740740741  0.147555555556   0.0223703703704
21      0.134576470588     0.492132179386               0  0.00118017309205   0.0247836349331 0.247115132442  0.0579268292683  0.0298649357461
22      0.125960117647     0.525759114629               0  0.00126081322453   0.0211186215109 0.266206703324  0.0641263615032  0.0319055790985
23      0.0122764705882    0                            0  0                  0.207474844274  0.20471969334   0.155366554863   0.0180881648299
24      0.188964705882     0.216006723945               0  0.000840493089279  0.0111832274935 0.176877101233  0.041487672768   0.0212691445648
25      0.333839705882     0.244533867227               0  0.000951493646798  0.00515392392015 0.168639033705 0.029086632175   0.0120390376699
26      0.0552308823529    0.0821151849189              0  0.000319514338206  0.0311526479751 0.256250499241  0.131719785925   0.0727693905264
27      0.00790735294118   0                            0  0                  0.161242328436  0.22038311326   0.171099125907   0.0213873907383
28      0.325888235294     0.250500893486               0  0.000974711647804  0.00648453999025 0.171082200682 0.0278875832566  0.0123328098771
29      0.327608823529     0.249185272967               0  0.000969592501818  0.00525195938485 0.171846804385 0.0296264375555  0.0122680384605
30      0.00674264705882   0                            0  0                  0.189904874591  0.263685932388  0.204362050164   0.0741548527008
31      0.00691617647059   0                            0  0                  0.338932596215  0.0748458430789 0.0527322985329  0.00191367212418
32      0.127275           0.0360844396686              0  0.000589275191514  0.0161530728968 0.365870567437  0.190318555236   0.0715796041457
33      0.0137926470588    0                            0  0                  0.501759249387  0.442477876106  0.291715534705   0.0393432135622
34      0.00342794117647   0                            0  0                  0.011583011583  0.151866151866  0.0978120978121  0.019305019305
```

8. Nic数据

图 2-57 Nic 结果数据展示

```
Nic Data:
  Duration(s)    Bandwidth(Mb)    Rx Bandwidth efficiency(%)    rxPacket/s  rxError rate    rxDropped rate      Tx Bandwidth efficiency(%)    txPacket/s  txErr
or rate    txDropped rate
-----------  --------------  ----------------------------  ------------  -------------  ----------------  ----------------------------  ------------  -----
---------  --------------
       1.012            1000                             0             0  0%             0%                                          0             0  0%
          0%
```

9. Dvpp数据

图 2-58 Dvpp 数据展示

```
Dvpp Data:
Engine type      Engine id    All Time(s)    All Frame  All Utilization
-------------  -----------  -------------  -----------  -----------------
VDEC                     1              0            0  0.0%
VDEC                     0              0            0  0.0%
JPEGD                    1              0            0  0.0%
JPEGD                    0              0            0  0.0%
PNGD                     5              0            0  0.0%
PNGD                     4              0            0  0.0%
PNGD                     3              0            0  0.0%
PNGD                     2              0            0  0.0%
PNGD                     1              0            0  0.0%
PNGD                     0              0            0  0.0%
JPEGE                    0              0            0  0.0%
VPC                      3              0            0  0.0%
VPC                      2              0            0  0.0%
VPC                      1              0            0  0.0%
VPC                      0              0            0  0.0%
VENC                     0              0            0  0.0%
```

 版权所有 © 华为技术有限公司

10. HiAI Engine 数据信息

图 2-59 HiAI Engine 数据展示

HiAI Engine Concrete Information:

Job ID	Graph Name	Engine Name	Start Time(s)	End Time(s)	Time Consumption(ms)
20190226113457556	graphhost1029073110	ImageClassificationPostProcess_1(3570)	1551152098.565894847	1551152098.568289413	2.394566
20190226113457556	graphhost1029073110	ImageClassificationPostProcess_1(3570)	1551152098.568299844	1551152098.568318248	0.018404
20190226113457556	graphhost1029073110	Mind_T(3571)	1551152098.367547371	1551152098.379430081	11.88271
20190226113457556	graphdevice1029073110	ImagePreProcess_1(2069)	1551152098.386952252	1551152098.56603502	179.082768
20190226113457556	graphdevice1029073110	ImagePreProcess_1(2069)	1551152098.566057676	1551152098.566223249	0.165573
20190226113457556	graphdevice1029073110	MindInferenceEngine_1(2068)	1551152098.565795749	1551152098.572310226	6.514477
20190226113457556	graphdevice1029073110	MindInferenceEngine_1(2068)	1551152098.572365486	1551152098.57277408	0.408594

HiAI Graph Information:

Job ID	Graph Name	Start Time(s)	End Time(s)	Time Consumption(ms)
20190226113457556	graphhost1029073110	1551152097.55212487	1551152098.786911295	1234.786425
20190226113457556	graphdevice1029073110	1551152097.751495431	1551152098.763095648	1011.600217

HiAI Engine Total Time Consumption:

Job ID	Graph Name	Engine Name	Time Consumption(ms)
20190226113457556	graphdevice1029073110	ImagePreProcess_1(2069)	179.248341
20190226113457556	graphdevice1029073110	MindInferenceEngine_1(2068)	6.923071
20190226113457556	graphhost1029073110	ImageClassificationPostProcess_1(3570)	2.41297
20190226113457556	graphhost1029073110	Mind_T(3571)	11.88271

11. OME 数据信息

图 2-60 OME 结果数据展示

Model Statistic:

Model Name	Model ID	Data Input Start Time(s)	Data Input End Time(s)	Inference Start Time(s)	Inference End Time(s)	Data Output Start Time(s)	Data Output End Time(s)
ResNet-18	1	1551152098.56704270	1551152098.567738873	1551152098.56773955	1551152098.570538144	1551152098.57054356	1551152098.571550299

Op Statistic:

Op Names	Op Num	Model Name	Model ID	Stream ID	Start Time(s)	End Time(s)	Time Consumption(us)	Memory:input	Memory:output	Memory:weight	Memory:workspace	Memory:total	Task Num	Task IDs
conv1,pool1	2	ResNet-18	1	1	1551152098.568085761	1551152098.568282475	0.196714	1691680	2007104	100352		3799136	1	365
res2a_branch2a	1	ResNet-18	1	1	1551152098.568109457	1551152098.568360493	0.251036	401440	401440	73728		876608	1	366
res2a_branch2b	1	ResNet-18	1	1	1551152098.568331015	1551152098.56841773	0.086715	401440	401440	73728		876608	1	367
res2a_branch1,res2a	2	ResNet-18	1	1	1551152098.568386325	1551152098.568470227	0.083902	1204320	802880	8192		2015392	1	368
res2b_branch2a	1	ResNet-18	1	1	1551152098.56845153	1551152098.568531318	0.079788	401440	401440	73728		876608	1	369
res2b_branch2b,res2b	2	ResNet-18	1	1	1551152098.5684946	1551152098.568600638	0.106038	1204320	802880	73728		2080928	1	370
res3a_branch2a	1	ResNet-18	1	1	1551152098.568553712	1551152098.568649489	0.095777	401440	200736	147456		749632	1	371
res3a_branch2b	1	ResNet-18	1	1	1551152098.568627512	1551152098.568688341	0.060829	200736	200736	294912		696384	1	372
res3a_branch1,res3a	2	ResNet-18	1	1	1551152098.568671676	1551152098.568699694	0.028018	802912	401472	16384		1220768	1	373
res3b_branch2a	1	ResNet-18	1	1	1551152098.568721255	1551152098.568786982	0.065727	200736	200736	294912		696384	1	374
res3b_branch2b,res3b	2	ResNet-18	1	1	1551152098.568742504	1551152098.568845782	0.103278	602208	401472	294912		1298592	1	375
res4a_branch2a	1	ResNet-18	1	1	1551152098.568827449	1551152098.568925571	0.098122	200736	100384	589824		890944	1	376
res4a_branch2b	1	ResNet-18	1	1	1551152098.568868385	1551152098.56893885	0.070465	100384	100384	1179648		1380416	1	377
res4a_branch1,res4a	2	ResNet-18	1	1	1551152098.56895838	1551152098.569019418	0.061038	401504	200768	65536		667008	1	378
res4b_branch2a	1	ResNet-18	1	1	1551152098.568973743	1551152098.569087747	0.114004	100384	100384	1179648		1380416	1	379
res4b_branch2b,res4b	2	ResNet-18	1	1	1551152098.569049312	1551152098.569226964	0.177652	301152	200768	1179648		1681568	1	380
res5a_branch2a	1	ResNet-18	1	1	1551152098.569124413	1551152098.569236911	0.112498	100384	50208	2359296		2509888	1	381
res5a_branch2b	1	ResNet-18	1	1	1551152098.569258004	1551152098.56947998	0.221976	50208	50208	4718592		4819008	1	382
res5a_branch1,res5a	2	ResNet-18	1	1	1551152098.569292429	1551152098.569489353	0.196924	200800	100416	262144		563360	1	383
res5b_branch2a	1	ResNet-18	1	1	1551152098.569511123	1551152098.569804558	0.293435	50208	50208	4718592		4819008	1	384
res5b_branch2b,res5b	2	ResNet-18	1	1	1551152098.569544976	1551152098.569925597	0.380621	150624	100416	4718592		4969632	1	385
pool5	1	ResNet-18	1	1	1551152098.569908932	1551152098.569936741	0.027809	50208	1056	0		51264	1	386
fc1000	1	ResNet-18	1	1	1551152098.569954864	1551152098.570030798	0.075934	1056	2048	1034240		1037344	1	387
prob	1	ResNet-18	1	1	1551152098.569995644	1551152098.570039963	0.044319	2048	2048	0	6	72224	1	388

12. LLC信息

图 2-61 LLC 数据

LLC Ctrl CPU Capacity:

Metric	CPU0(MB)	CPU1(MB)	CPU2(MB)	CPU3(MB)	Total(MB)
Used Capacity of LLC	0.015137	0.015259	0.061584	0.01239	0.10437

LLC AI CPU Capacity:

Metric	CPU0(MB)	CPU1(MB)	CPU2(MB)	CPU3(MB)	Total(MB)
Used Capacity of LLC	0.0	0.0	0.0	0.628296	0.628296

13. DDR信息

 版权所有 © 华为技术有限公司

图 2-62 DDR 数据

```
DDR:
Metric                Read       Write
--------------  ----------  ----------
BandWidth(MB/s)  159.636322  444.054293
```

后续处理

采集成功的结果数据也可导入到Profiling UI页面上展示，操作步骤如下：

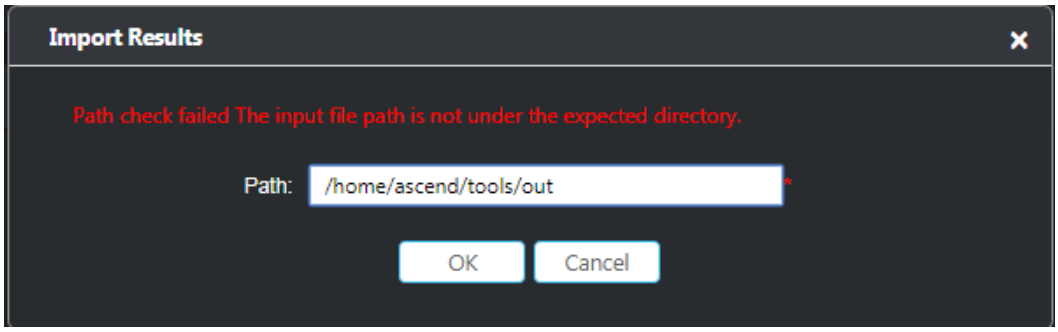
步骤1 在浏览器地址栏输入https://profiling安装服务器ip地址:8099，登录Profiling UI展示界面，默认账号密码是msvpadmin/Admin12#$，登录后点击页面上方Analysis图标，点击下拉菜单Import Results，如图2-63所示。

图 2-63 选择菜单

步骤2 在弹出的对话框Path栏中输入命令行采集结果所在的路径，然后按Tab键会自动补齐Device ID，单击OK，如图2-64所示。

“Path”参数处的路径需要与命令行中“result_dir”参数处的路径保持一致。

图 2-64 导入采集数据



----结束

sample.ini 文件

sample.ini文件的内容样例如下：

```
[GENERAL]
analysis_type=ai
analysis_target=Launch Application
result_dir=/home/ascend/tools/out
umode=MIND
profiling_mode=online
HIAI_Engine_Profiling=on
Framework_Profiling=on
RTS_Profiling=on
```

 版权所有 © 华为技术有限公司

```
local_app_dir=/projects/MIND0126_02
local_app=/projects/MIND0126_02/out/main
app_parameters=
cpu_profiling_interval=20
ctrl_cpu_profiling=on
ts_cpu_profiling=off
ai_cpu_profiling=on
ai_core_profiling=on
ai_core_profiling_interval=10
ai_core_profiling_mode=task-based
peripheral_profiling=nic
peripheral_profiling_interval=10
llc_capacity=on
llc_bandwidth=off
llc_interval=100
ddr_profiling=on
ddr_interval=100
ddr_master_id=3

app_dir=/home/HwHiAiUser/HIAI_PROJECTS/66adc0ebcb506226f153262feb4f7c7c//MIND0126_02/out
app=hiai_66adc0ebcb506226f153262feb4f7c7c_MIND0126_02_main
ctrl_cpu_profiling_events=0x11,0x8
ai_core_profiling_events=0x3,0x8,0x9,0xe,0x3a,0x3b,0x4a,0x49
ai_cpu_profiling_events=0x11,0x8
ts_cpu_profiling=off
llc_profiling_events=hisi_l3c0_1/dsid0/,hisi_l3c0_1/dsid1/,hisi_l3c0_1/dsid2/,hisi_l3c0_1/
dsid3/,hisi_l3c0_1/dsid4/,hisi_l3c0_1/dsid5/,hisi_l3c0_1/dsid6/,hisi_l3c0_1/dsid7/
ddr_profiling_events=read,write
stream_enabled=on
cleanup_host_results=on
cleanup_device_results=on
ddk_dir=/home/ascend/tools/che/ddk/ddk
llc_profiling=on
job_id=b8380eea-236c-11e9-8354-286ed488e3a7
devices=0
```

关键参数的含义如表2-4所示。

表 2-4 关键参数说明

参数	说明	是否必选
profiling_mode	表示profiling采集模式，配置为online为在host上运行app，配置为offline为在device上运行app。	是
ddk_dir	表示ddk路径（此路径须存在）。	是
umode	表示工程模式，配置为“MIND”表示通过Mind Studio工具创建的工程，使用Profiling系统采集数据，此时profiling_mode须为online。	是

 版权所有 © 华为技术有限公司

参数	说明	是否必选
app_dir	在命令行中，app_dir参数值需配置为app可执行文件在UI host（指Mind Studio所在的服务器）存储根目录，此选项用户必须输入而且此路径需确实在UI host 上存在，例如“--app_dir /projects/MIND0126_02 ”。 在sample.ini 中，app_dir参数值会被修改为对应Host 上的 app可执行文件的根路径，例如“~/HIAI_PROJECTS/workspace_mind_studio/MIND0126_02/out”。同时，在sample.ini 中，local_app_dir参数值与命令行中的app_dir参数值一致。	是
app	在命令行中，app参数需配置为app可执行文件在UI host存储目录（至app名字），此选项用户必须输入而且此路径需确实在 UI host 上存在，例如“--app /projects/MIND0126_02/out/main ”。 在sample.ini 中，app参数值会被修改为对应Host上的app名称，例如“hiai_workspace_mind_studio_MIND0126_02_main ”。同时，在sample.ini 中，local_app参数值与命令行中的app参数值一致。 **说明** 此处举例中的main函数的用于数据分析的代码入口，“/projects/profiling_test/app/main”是main函数的绝对路径。	是
devices	表示device列表，默认为0。 只在CCE场景需要输入该参数值。	否
OME_Profiling	表示OME模块开关。配置on为开启采集，配置off为不采集，默认值为off。 在sample.ini 中，该采集项的名称为Framework_Profiling。	否
HIAI_Engine_Profiling	表示HiAI Engine模块开关，配置on为开启采集，配置off为不采集，默认值为off。	否
peripheral_profiling_interval	为外设采样时间间隔，须位于 [1,1000] 之内，默认值为10。	否
peripheral_profiling	代表外设采集方式，可为空。dvpp或nic或者二者皆选，需用逗号分隔，默认值为空。	否
ctrl_cpu_profiling	代表是否采集ctrl cpu信息。配置on为开启采集，配置off为不采集。默认值为off。	否
ts_cpu_profiling	代表是否采集ts cpu信息。配置on为开启采集，配置off为不采集。默认值为off。	否

 版权所有 © 华为技术有限公司

参数	说明	是否必选
ai_cpu_profiling	代表是否采集ai cpu信息。配置on为开启采集，配置off为不采集。默认值为off。	否
ai_core_profiling	代表是否采集ai core信息。配置on为开启采集，配置off为不采集。默认值为off。	否
ai_core_profiling_mode	代表ai core采集模式。可选项为task-based 和 sample-based。其中task-based为以task粒度采集，sample-based为以sample粒度采集。默认值为task-based。	否
ai_core_profiling_interval	表示ai core采样模式为sample based时，ai core的采样周期长度。取值范围是[1, 1000]，默认值为10。	否
RTS_Profiling	表示rts模块开关。配置on为开启采集，配置off为不采集。默认值为off。	否
result_dir	表示结果文件存储地址，如果不存在则创建，若存在则将以前文件夹改为.old。此处路径应输入绝对路径。 在“result_dir”处配置的路径用于存放采集结果，需确保Mind Studio安装用户对此路径有可读可写的权限，建议将该路径配置为“~/tools/”下的目录或者“/projects/”。	否
cpu_profiling_interval	表示cpu 采样周期，若有输入需确认输入是否在[20, 1000] 之间。默认值为20。	否
import	代表调用mvsp_import.pyc将采集结果文件导入数据库，需要配置文件路径。采集结果文件必须存放在“Mind Studio安装用户家目录/tools”下的目录或子目录中，否则采集结果文件无法导入。此参数未保存于 sample.ini 中。 **python hiprof.pyc**命令行中可以只包含import参数。	否
report	表示打印profiling采集结果。 **python hiprof.pyc**命令行中可以只包含report参数，并在后面输入profiling采集后的结果绝对路径，即可打印结果。例 python hiprof.pyc --report /projects/test/profiling_output/	否
llc_bandwidth	代表采集LLC 的读写带宽信息。配置为on为开启采集，配置为off为不采集，默认为off。与llc_capacity开关互斥，只能同时采集一种llc数据。	否
llc_capacity	代表采集LLC 的使用容量信息。配置为on为开启采集，配置为off为不采集，默认为off。与llc_bandwidth开关互斥，只能同时采集一种llc数据。	否

 版权所有 © 华为技术有限公司

参数	说明	是否必选
llc_interval	表示LLC采样周期，若有输入需确认输入是否在[100, 1000] ms之间。默认值为100ms。	否
ddr_profiling	代表是否采集ddr信息。配置on为开启采集，配置off为不采集。默认值为off。	否
ddr_master_id	表示采集AI CPU和Control CPU Core的读写带宽。默认为空，当前可选0,1,2,3,4,5,6,7。0,1,2,3对应Control CPU core ID；4,5,6,7对应AI CPU core ID。	否
ddr_interval	表示DDR采集周期，若有输入需确认输入是否在[100, 1000] ms之间。默认值为100ms。	否
all	表示打开所有性能数据采集开关，取值有on，off两项，默认为off。	否
help	表示查询帮助信息。	否

说明

- 请确认创建的result_dir的各级路径皆具备750权限。
- 带参数算子输入格式为（其中para1 para2是app的参数） -- *app文件在UI host存储目录（至app名字）* para1 para2。例如：-- /projects/CCE/out/main para1 para2。如果在性能数据采集的命令行中增加“-- *app文件在UI host存储目录（至app名字）* para1 para2”参数，则需要将该参数放在命令行尾部，同时需删除命令行中的“--app”参数，否则会出现解析错误。
- 建议DDR和LLC的采样周期不大于程序执行所需的时间，否则可能会出现采集不到数据的问题。
- 如果采集时间少于1s，则不会展示Control CPU Usage、AI CPU Usage、TS CPU Usage图。

2.4 参考

2.4.1 重置连接 redis 服务的密码

为保证安全，建议您定期（例如，90天）重置连接redis服务的密码，连接redis服务的默认密码是“Huawei12#$”。

步骤1 以Mind Studio安装用户登录Mind Studio所在的服务器。

步骤2 切换至“tools/profiler”目录下，执行以下命令重置密码。

其中，“dir”表示Profiling的安装路径，“~”需替换为Mind Studio安装用户的家目录。

```
./install_profiling.sh --reset_redis_pwd --dir=~/tools/profiler
```

说明

“~/tools”是默认的toolpath路径，该路径可在安装Mind Studio时由用户自定义，您可以在“scripts/env.conf”文件通过toolpath参数查看实际路径。您可以使用**find / -name 'env.conf'**命令查看script目录下的“env.conf”文件的位置。

 版权所有 © 华为技术有限公司

根据系统回显提示输入两次新密码，如图2-65所示。

密码需满足以下要求：

- 口令长度至少6个字符；
- 口令必须包含如下至少两种字符的组合：
  - 至少一个小写字母；
  - 至少一个大写字母；
  - 至少一个数字；
  - 至少一个特殊字符：`~!@#$%^&*()-_=+\|[{}];:'",<.>/?和空格

图 2-65 输入新密码

```
Wed 20 Feb 2019 23:20:48 [INFO] Begin to reset redis password ...
Wed 20 Feb 2019 23:20:48 [INFO] Enter new redis password:
Wed 20 Feb 2019 23:20:52 [INFO] Enter new redis password again:
```

输入新密码后，系统后台执行密码重置操作，若系统回显图2-66中的信息，则表示重置密码成功。

图 2-66 重置密码成功

```
Wed 20 Feb 2019 23:21:20 [INFO] Start profiler success
Wed 20 Feb 2019 23:21:20 [INFO] Redis password successfully changed.
```

----结束

2.4.2 脚本清单

表2-5中脚本的执行方法如下：

1. 以Mind Studio安装用户登录Mind Studio服务器。
2. 切换到“~/tools/profiler/analysis/msvp/host/bin64”路径下。

   说明

   “~/tools”是默认的toolpath路径，该路径可在安装Mind Studio时由用户自定义，您可以在“scripts/env.conf”文件通过toolpath参数查看实际路径。您可以使用**find / -name 'env.conf'**命令查看script目录下的“env.conf”文件的位置。
3. 执行表2-5中的命令调用工具。

表 2-5 脚本列表

脚本	作用	参数及示例
get_env_info.pyc 校验文件夹或删除文件/文件夹	校验指定的文件夹。是否符合导入规范。	--verify/-v <*指定要校验的文件夹*> **示例：**python get_env_info.pyc -v /home/msvptest/tools/projects_name
	删除指定的文件或文件夹。	--remove/-r <*指定要删除的文件*> **示例：**python get_env_info.pyc -r /home/whl/info.xml
	查看帮助信息	-h/--help

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
get_msvp_function.pyc 根据指定的project、Module名称、Function名称获取Function页签的数据	指定工程，获取Function页签下所有数据类型的数据	● --project <*指定工程文件夹*>：必选参数 ● --target <*指定数据类型*>：可选值包括total、core、thread、module、function、callstack、export、class、classmethod、exportfunc、exportmodule、exportcore、exporttid ● --deviceid=<*指定设备id*> ● --type <*指定cpu类型*>：可选值包括ctrlcpu、aicpu ● --order/-o <*指定按照什么指标来排序*>：可选值是cycles ● --core/-c <*指定cpu核数*>：通过core来过滤结果 ● --pid/-p <*指定pid*>：通过进程ID进行过滤结果 ● --tid/-t <*指定tid*>：通过线程ID进行过滤结果 ● --sort/-s <*指定排序*>：可选值包括升序ASC、降序DESC ● --limit/-l <*指定限制结果条数*> ● --export/-e：添加此选项表示要导出结果 **示例：**python get_msvp_function.pyc --project /home/msvptest/tools/projects_name --target total --deviceid=0 --type=ctrlcpu -o cycles -c 1 -p 1 -t 1 -s DESC -l 1 -e
	指定工程和Module名称，获取Function页签下数据类型为function的数据	--module/-m <*指定Module名称*> **示例：**/home/whl/analysis/msvp/host/bin64# python get_msvp_function.pyc --project /home/msvptest/tools/projects_name --target function -m /usr/lib64/libc-2.17.so --type ctrlcpu --deviceid=0
	指定工程和Function名称，获取Function页签下数据类型为callstack的数据	--function/-f <*指定Function名称*> **示例：**python get_msvp_function.pyc --project /home/msvptest/tools/projects_name --target callstack -f _dl_relocate_object --type ctrlcpu --deviceid=0
	查看帮助信息	-h/--help

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
get_msvp_info.pyc 获取指定工程的Summary页签的数据	获取Summary页签指定CPU的CPU Usage信息	● --project<*指定工程文件夹*>：必选参数 ● --deviceid=<*指定设备id*> ● --cpuusage：获取特定类型的CPU的CPU Usage数据，与“--type”一起使用 ● --type <*指定CPU类型*>：可选值包括aicpu、ctrlcpu ● --startTime=<*指定开始时间*>：缩放时指定开始时间，若采集时间为10s，指定“--startTime=0”，表示从0s开始。 ● --endTime=<*指定结束时间*>：缩放时指定结束时间，若采集时间为10s，指定“--endTime=6”，表示到6s结束。 ● --number <*指定页面最多显示的数据条数*> **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --cpuusage --type ctrlcpu --startTime=0 --endTime=6 --number 1000
	获取Summary页签Collection Info信息	--collection_info **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --type aicpu --collection_info
	获取Summary页签Host Info信息	--host_info **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --type aicpu --host_info
	获取设备id	--msvp_device **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --msvp_device
	获取指定设备的设备信息	--msvp_deviceinfo **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --msvp_deviceinfo --deviceid=0
	获取Runtime API数据	--runtime_api **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --runtime_api --deviceid=0

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
	获取HiAI Engine数据	--hiai_engine **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --hiai_engine
	获取Framework的算子数据	● --framework_op ● --limitPage <*指定每页显示多少条数据*> ● --page <*指定要获取第几页的数据*> ● --sortColName <*指定要排序的列名*>：可选参数，列包括op_names、fusion_op_nums、modelname、model_id、stream_id、op_start、op_end、op_en、memory_input、memory_output、memory_weight、memory_workspace、memory_total、task_num、task_ids ● --sort <*指定按照什么方式来排序*>：可选参数，排查方式包括desc（降序）和asc（升序） **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --framework_op --limitPage 50 --page 1 --sortColName op_names --sort desc
	获取Framework的模型数据	● --framework_model ● --limitPage <*指定每页显示多少条数据*> ● --page <*指定要获取第几页的数据*> ● --sortColName <指定要排序的列名>：可选参数，列包括modelname、model_id、input_start、input_end、infer_start、infer_end、output_start、output_end、thread_id ● --sort <*指定按照什么方式来排序*>：可选参数，排查方式包括desc（降序）和asc（升序） **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --framework_model --limitPage 50 --page 1 --sortColName modelname --sort desc

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
	获取CCE数据	--cce **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --cce
	获取Task Scheduler数据	● --task_scheduler ● --limitPage <*指定每页显示多少条数据*> ● --page <*指定要获取第几页的数据*> ● --sortColName <指定要排序的列名>：可选参数，列包括TimeRatio、Time、Count、Avg、Min、Max、Waiting、Running、Pending、Type、API、taskID、streamID ● --sort <*指定按照什么方式来排序*>：可选参数，排查方式包括desc（降序）和asc（升序） **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --type aicpu --task_scheduler --limitPage 100 --page 1 --sortColName TimeRatio --sort desc
	获取Control CPU PMU Events数据	--control_cpu_pmu_events **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --control_cpu_pmu_events
	获取Control CPU Top 5 Functions	--control_cpu_top_functions **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --control_cpu_top_functions
	获取TS CPU PMU Events数据	--ts_cpu_pmu_events **示例：** python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --ts_cpu_pmu_events
	获取TS CPU Top 5 Functions数据	--ts_cpu_top_functions **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --ts_cpu_top_functions
	获取AI CPU PMU Events数据	--ai_cpu_pmu_events **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --ai_cpu_pmu_events

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
	获取AI CPU Top 5 Functions	--ai_cpu_functions **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --ai_cpu_functions
	获取AI core CPU PMU Events数据	● --ai_core_pmu_events ● --limitPage <*指定每页显示多少条数据*> ● --page <*指定要获取第几页的数据*> ● --sortColName <指定要排序的列名>：可选参数，列包括total_time、mac_fp16_ratio、mac_int8_ratio、mac_ratio、vec_ratio、scalar_ratio、scalar_ld_ratio、scalar_st_ratio ● --sort <*指定按照什么方式来排序*>：可选参数，排查方式包括desc（降序）和asc（升序） **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --ai_core_pmu_events --limitPage 50 --page 1 --sortColName total_time --sort desc
	获取指定事件类型的AI core CPU PMU Events数据	--event_type <指定事件类型>：可选值包括“0”和“5”，“0”表示“AI Core”，“5”表示“DMA” **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --type aicpu --event_type=0 --ai_core_pmu_events
	获取nic外设数据	--nic **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --nic
	获取dvpp外设信息	--dvpp **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --dvpp
	获取llc数据	● --llc ● --type <*指定CPU类型*>：可选参数，可选值包括aicpu、ctrlcpu **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --llc --type ctrlcpu

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
	获取ddr数据	--ddr **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --ddr
	获取TS CPU Usage数据	● --ts_cpu_usage ● --startTime=<*指定开始时间*>：可选参数，缩放时指定开始时间，若采集时间为10s，指定“--startTime=0”，表示从0s开始。 ● --endTime=<*指定结束时间*>：可选参数，缩放时指定结束时间，若采集时间为10s，指定“--endTime=6”，表示到6s结束。 ● --number <*指定页面最多显示的数据条数*>：必选参数 **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --ts_cpu_usage --number 100
	获取AI Core Status数据	● --ai_core_status ● --number <*指定页面最多显示的数据条数*>：必选参数 **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --ai_core_status -number 100

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
	获取RTS track数据	● --autotuning：同时需指定“--event_type”参数，“--event_type”参数可选值包括“0”和“5”，“0”表示“AI Core”，“5”表示“DMA” ● --limitPage <*指定每页显示多少条数据*> ● --page <*指定要获取第几页的数据*> ● --sortColName <指定要排序的列名>：可选参数，列包括app_runtime、task_duration、run_tx_time、run_rx_time、runtime_app、sched_tx_time、core_time、sched_rx_time ● --sort <*指定按照什么方式来排序*>：可选参数，排查方式包括desc（降序）和asc（升序） **示例：**python get_msvp_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --autotuning --event_type=0 --limitPage 100 --page 1 --sortColName app_runtime --sort desc
	查看帮助信息	-h/--help
get_msvp_instruction.pyc 实现代码跳转的功能，获取code页签的数据	指定源码路径，获取code页签的数据	● --project <*指定工程文件夹*>：必选参数 ● --ddk_dir <*指定ddk目录*>：必选参数 ● --module/-m <*指定Module名称*> ● --function/-f <*指定Function名称*> ● --source <*指定要配置的源码路径*> ● --type <*指定cpu类型*>：可选值包括ctrlcpu、aicpu ● --deviceid=<*指定设备id*> ● --field/-fld <*指定code页签的filed字段*>：可选值包括cycles(default)、r8，其中cycles是默认值 **示例：**python get_msvp_instruction.pyc --project /home/msvptest/tools/projects_name --ddk_dir /home/tools/che/ddk/ddk --source /home/whl/MIND0214001 -m /usr/lib64/libpthread-2.17.so -f __errno_location --deviceid=0 --type=ctrlcpu --field cycles

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
	指定二进制符号表文件，获取code页签的数据	--symtab <*指定要配置的二进制符号表文件*> **示例：**python get_msvp_instruction.pyc --project /home/msvptest/tools/projects_name --ddk_dir /home/tools/che/ddk/ddk --symtab /home/whl/MIND0214001 -m /usr/lib64/libpthread-2.17.so -f __errno_location --deviceid=0 --type=ctrlcpu
	查看帮助信息	-h/--help
get_msvp_timeline.pyc 获取Timeline页签的数据	获取时序图数据	● --project <*指定工程文件夹*>：必选参数 ● --startTime=<*指定起始时间*> ● --endTime=<*指定结束时间*> ● --deviceid=<*指定设备id*> ● --timeline_diagram：指定获取时序图的数据，不包含左侧数据的树形结构。 ● --replayid=<*指定replayid*>：预留参数 **示例：**python get_msvp_timeline.pyc --project /home/msvptest/tools/projects_name --startTime=1550759017.8505 --endTime=1550759018.9249 --deviceid=0 --timeline_diagram
	获取llc的timeline数据	● --type <*指定CPU类型* >：可选值包括aicpu、ctrlcpu ● --timeline_llc：获取llc的timeline数据 **示例：**python get_msvp_timeline.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --startTime=1550759017.8505 --endTime=1550759018.3505 --type aicpu --timeline_llc
	获取ddr的timeline数据	● --master_id <*指定DDR通道*>：可选值包括0、1、2、3、4、5、6、7 ● --timeline_ddr：获取ddr的timeline数据 **示例：**python get_msvp_timeline.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --startTime=1550759017.8505 --endTime=1550759018.3505 --master_id 0 --timeline_ddr

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
	获取时序图左侧树形结构的数据列表	--timeline_list **示例：**python get_msvp_timeline.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --timeline_list
	获取左侧外设数据列表	--timeline_list_peri **示例：**python get_msvp_timeline.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --timeline_list_peri
	获取Timeline的开始时间、结束时间、Duration参数值（大数据量情况下用于展示一次接受多长时间的数据）	● --timeline_maxtime ● --deviceid=<*指定设备id*>：必选参数 **示例：**python get_msvp_timeline.pyc --project /home/msvptest/tools/projects_name --timeline_maxtime --deviceid=0
	获取dvpp的timeline数据	● --timeline_dvpp ● --enginetype <*指定enginetype*>：可选值VDEC、JPEGD、PNGD、JPEGE、VPC、VENC。 ● --engineid=<*指定engineid*> ● --number <*指定页面最多显示的数据条数*> **示例：**python get_msvp_timeline.pyc --project /home/msvptest/tools/projects_name --startTime=1550813088.0822 --endTime=1550813089.9895 --deviceid=0 --timeline_dvpp --enginetype VDEC --engineid=1 --number 10000
	获取timeline_apitask数据	● --timeline_apitask：获取timeline_apitask数据（必须指定rowId） ● --rowId=<*指定rowid*> **示例：**python get_msvp_timeline.pyc --project /home/msvptest/tools/projects_name --deviceid=0 -- rowId=0 --timeline_apitask
	查看帮助信息	-h/--help

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
get_rstrack_info.pyc 获取Timeline页签的RTStrack数据	获取rtstrack数据	● --project <*指定工程文件夹*>：必选参数 ● --startTime=<*指定开始时间*> ● --endTime=<*指定结束时间*> ● --deviceid=<*指定设备id*> ● --rtstrack **示例：**python get_rstrack_info.pyc --project /home/msvptest/tools/projects_name --startTime=1550759017.8505 --endTime=1550759018.9249 --deviceid=0 --rtstrack
	获取开始时间和结束时间	--timerange **示例：**python get_rstrack_info.pyc --project /home/msvptest/tools/projects_name --deviceid=0 --timerange
	查看帮助信息	-h/--help
msvp_import.pyc	将采集结果导入数据库	● --target <*指定要导入的目标文件夹*>：必选参数 ● --ddk_dir <*指定ddk目录*> **示例：**python msvp_import.pyc --target /home/msvptest/tools/profiler/projects/68:05:ca:83:ad:57/projects_20190221/rts_test_2019022109234475 --ddk_dir /home/msvptest/tools/che/ddk/ddk **说明** 在执行该脚本前，需要先清理“--target”指定目录下的“sqlite”目录下的所有*.db文件以及“data/log”目录下的*.db文件。
	查看帮助信息	-h/--help

 版权所有 © 华为技术有限公司

脚本	作用	参数及示例
msvp_runss.pyc	运行采集数据的主函数	● -c <*指定sample.ini 文件的路径*> -ddk <*指定ddk目录*> -ip <*指定要采集环境的ip地址*>：表示Host侧服务器的IP地址 **示例：**python msvp_runss.pyc -c /home/whl/r000ai/sample.ini -ddk /home/msvptest/tools/che/ddk/ddk -ip *xx.xx.xx.xx*:22118 **说明** ● *xx.xx.xx.xx*需要根据实际情况替换为Host侧服务器的IP地址。 ● 执行该脚本前，需要提前将编译后的工程的相关文件复制到Host侧的对应目录（该目录与*sample.ini 文件*中记录的app_dir参数保持一致）下，同时删除*sample.ini* 文件中app参数下的其它内容。
	停止采集	-stop **示例：**python msvp_runss.pyc -stop -c /home/whl/r000ai/sample.ini -ip *xx.xx.xx.xx*:22118 **说明** *xx.xx.xx.xx*需要根据实际情况替换为Host侧服务器的IP地址。
	查看帮助信息	-h/--help

 版权所有 © 华为技术有限公司

3 黑匣子异常信息获取

3.1 简介

Mind Studio 黑匣子异常获取工具提供了获取设备异常信息的功能，用户可以通过界面获取不同设备产生的异常信息，用于异常原因分析（部分异常信息的解析需要借助第三方异常分析工具）。

界面信息如图3-1所示，详细说明请参见表3-1。

图 3-1 黑匣子异常信息获取工具界面

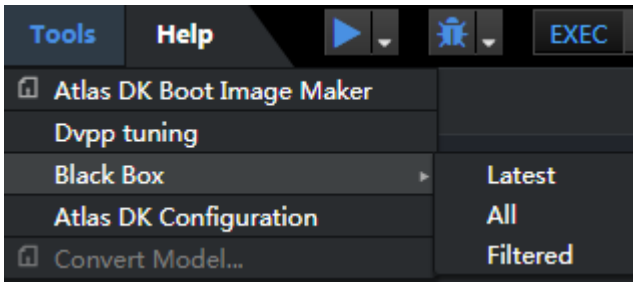


表 3-1 黑匣子异常信息获取工具功能说明表

序号	界面信息	说明
1	Latest	获取所有设备中最新产生的1个异常信息。
2	All	获取所有设备的异常信息。
3	Filtered	获取自定义选择设备最新的N个异常信息。

3.2 基本操作

根据表3-1的说明，下面分三个场景分别介绍异常信息的获取方式。

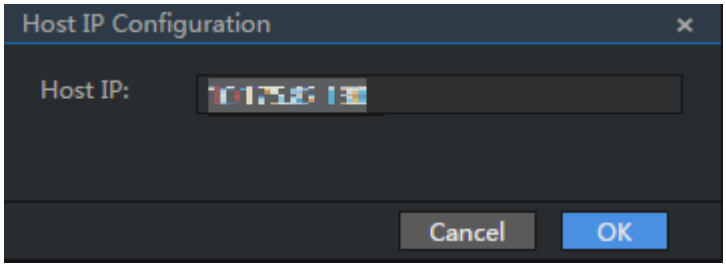
 版权所有 © 华为技术有限公司

获取所有设备中最新产生的 1 个异常信息

步骤1 在Mind Studio界面依次选择“Tools > Black Box > Latest”。

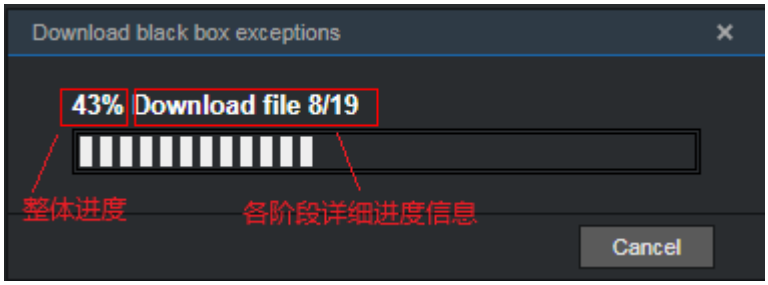
步骤2 输入目标主机的IP地址，如图3-2所示。

图 3-2 黑匣子异常获取工具主机 IP 配置图



步骤3 之后会弹出异常信息下载进度对话框，显示当前下载进度，如图3-3所示。

图 3-3 异常信息下载进度对话框



步骤4 等待异常信息传输完毕，会自动打开下载异常信息zip包的提示信息（zip包以时间戳命名），将异常信息zip包下载后，传给华为工程师进行分析。

----结束

获取所有设备的异常信息

步骤1 在Mind Studio界面依次选择“Tools > Black Box > All”。

步骤2 输入目标主机的IP地址，如图3-2所示。

步骤3 之后会弹出异常信息下载进度对话框，显示当前下载进度，如图3-3所示。

步骤4 等待异常信息传输完毕，会自动打开下载异常信息zip包的提示信息（zip包以时间戳命名），将异常信息zip包下载后，传给华为工程师进行分析。

----结束

获取自定义选择设备最新的 N 个异常信息

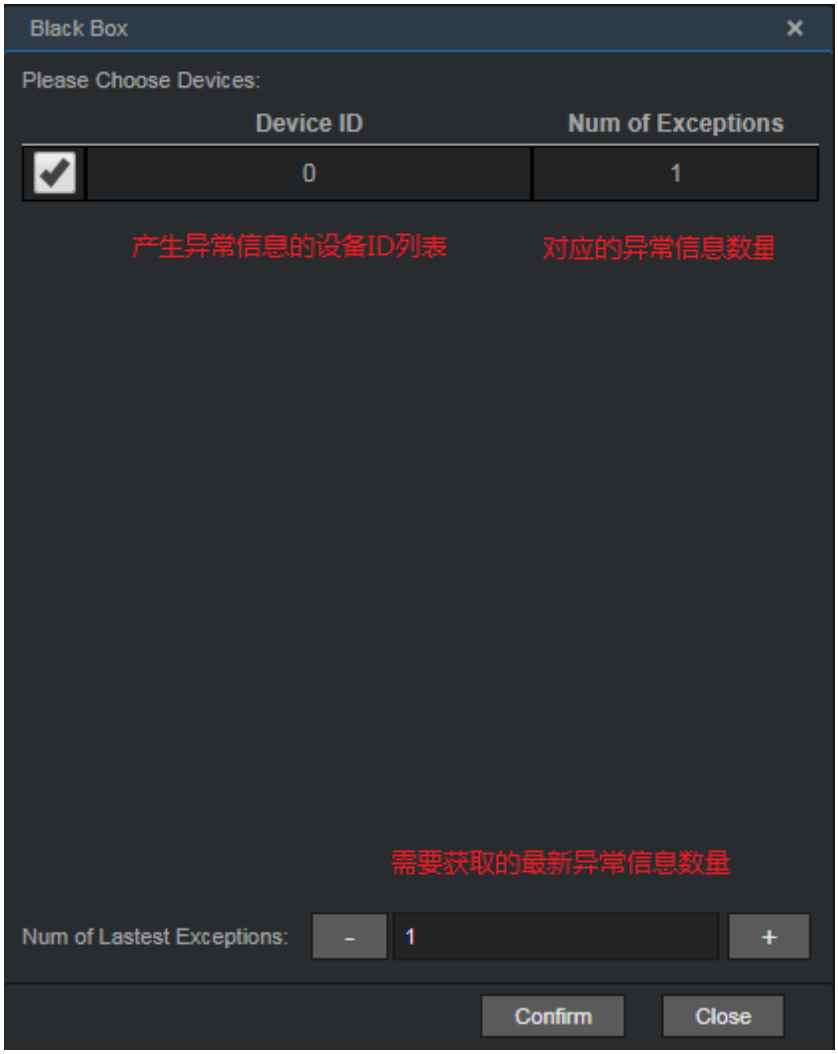
步骤1 在Mind Studio界面依次选择“Tools > Black Box > Filtered”。

步骤2 输入目标主机的IP地址，如图3-2所示。

步骤3 输入主机IP地址后，会弹出如图3-4所示界面，勾选关注的异常设备ID，并输入需要获取的最新异常信息数量。

 版权所有 © 华为技术有限公司

图 3-4 黑匣子异常获取工具自定义选择界面



步骤4 之后会弹出异常信息下载进度对话框，显示当前下载进度，如图3-3所示。

步骤5 等待异常信息传输完毕，会自动打开下载异常信息zip包的提示信息（zip包以时间戳命名），将异常信息zip包下载后，传给华为工程师进行分析。

----结束

 版权所有 © 华为技术有限公司