

Amlogic

Perlayer Visual 工具介绍


Revision: 0.1

Release Date: 2022-02-10

Copyright

© 2022 Amlogic. All rights reserved. No part of this document may be reproduced, transmitted, transcribed, or translated into any language in any form or by any means without the written permission of Amlogic.

Trademarks

 , and other Amlogic icons are trademarks of Amlogic companies. All other trademarks and registered trademarks are property of their respective holders.

Disclaimer

Amlogic may make improvements and/or changes in this document or in the product described in this document at any time.

This product is not intended for use in medical, life saving, or life sustaining applications.

Circuit diagrams and other information relating to products of Amlogic are included as a means of illustrating typical applications. Consequently, complete information sufficient for production design is not necessarily given. Amlogic makes no representations or warranties with respect to the accuracy or completeness of the contents presented in this document.

Contact Information

- Website: www.amlogic.com
- Pre-sales consultation: contact@amlogic.com
- Technical support: support@amlogic.com

更改记录

版本 0.1 (2022-02-10)

第一版。

Confidential!
nick@khadas.com
2022-03-17 19:26:34

目录

更改记录.....	ii
1. Perlayer visual 简介.....	1
2. Perlayer visual 使用方式介绍.....	2
2.1 获取板端结果.....	2
2.2 PC 可视化.....	2

Confidential!
nick@khadas.com
2022-03-17 19:26:34

1. Perlayer visual 简介

Perlayer visual 是 Amlogic 开发的基于硬件信息打印的模型各层信息可视化工具。方便用户评估模型的运行时间和带宽占用，以及定位模型耗时和带宽占用过大的问题。

工具获取方式：

https://github.com/Amlogic-NN/AML_NN_SDK/tree/master/Tools/get_layer_info

Confidential!
nick@khadas.com
2022-03-17 19:26:34

2. Perlayer visual 使用方式介绍

2.1 获取板端结果

1. 导入支持模型运行的动态库文件

Github 路径下找到对应指定平台的动态库文件，注意 DDK 版本和平台信息要保持一致。主要包括 A311D_0x88, C305X_0xbe, C308X_0xa1, S905D3_0x99 四种平台，common 保存各平台通用的动态库文件。

https://github.com/Amlogic-NN/AML_NN_SDK/tree/master/Linux/runtime_sharelib

将对应的动态库文件 push 到板端，例如/data/test/drivers，设置环境变量：

```
export LD_LIBRARY_PATH=/data/test/drivers
```

2. 板端运行 export.data 模型前设置环境变量

```
export VIV_VX_PROFILE=1
```

```
export VIV_VX_DEBUG_LEVEL=1
```

```
export CNN_PERF=1
```

```
export VIV_VX_SAVE_PATH=inceptionv1.txt #设置保存打印信息路径
```

3. 运行 export.data case

```
./inceptionv1 inceptionv1.export.data input.tensor
```

每层信息打印会保存在之前设置的路径 inceptionv1.txt 文件中。

2.2 PC 可视化

1. 将板端运行时保存的 inceptionv1.txt 和可视化脚本放置在 python3 环境下
2. 执行脚本

```
python3 show_layer_info.py -i inceptionv1.txt -c 0xa1
```

其中-i 为输入的文本文件路径，-c 为对应的平台序号

```
A311D --- 0x88
```

```
S905D3 --- 0x99
```

```
C305X --- 0xbe
```

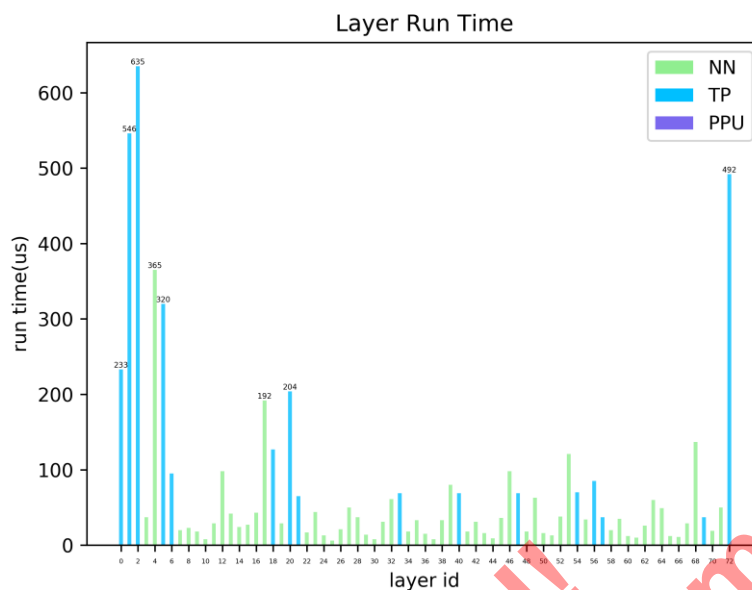
```
C308X --- 0xa1
```

3. 保存图片和对结果文件

执行脚本后保存 DDR_BandWidth.png, Runtime.png 分别描述带宽、运行时间的图像，而且会生成 results.txt 保存相关信息。

4. 示例

基于 DDK6487+C305X+inceptionv1 模型的执行结果



Runtime.png 获取模型各层运行在 NN/TP/PPU 具体硬件上，获取每层的运行时间。DDR_BandWidth.png 获取模型各层占用的 DDK 带宽。可参考上述指标针对性的优化算子，提升模型的运行效率。每层标号对应具体算子信息可见 results.txt。

