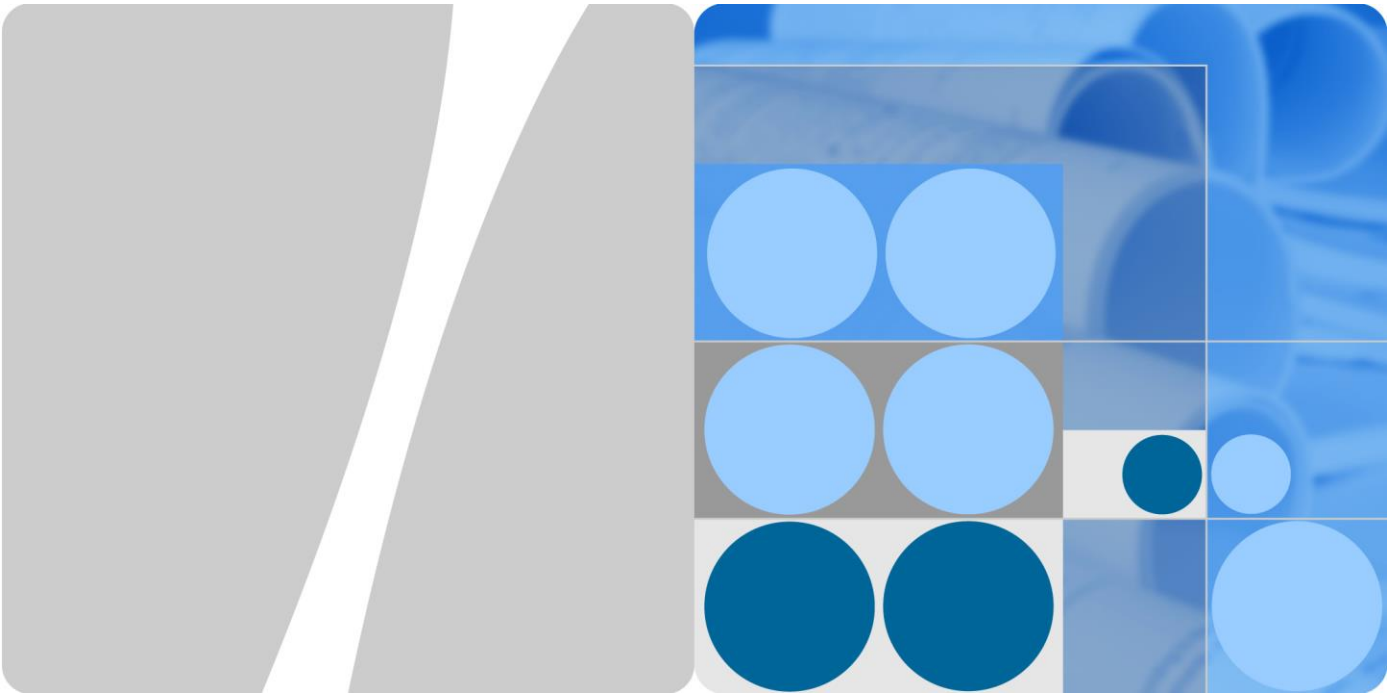

HiAI DDK V320

System Debug Tool Instructions

Issue 02

Date 2020-06-09

HUAWEI TECHNOLOGIES CO., LTD.

Huawei HiAI Application

Send an application email to developer@huawei.com.

Email subject: HUAWEI HiAI + Company name + Product name

Email body: Cooperation company + Contact person + Phone number + Email address

We will reply to you within 5 working days.

Official website: https://developer.huawei.com/consumer/en/

About This Document

Purpose

This document describes the functions and usage of the system debug tool: **model_run_tool** and **data_proc_tool**.

This tool is only supported in the NPU scenario of Kirin 990, Kirin 820, and Kirin 985.

Change History

Changes between document issues are cumulative. The latest document issue contains all the changes made in earlier issues.

Date	Version	Change Description
2020-06-09	02	Added the description of performance statistics files.
2020-03-19	01	This issue is the first official release.

Contents

1 Introduction

The system debug tool can collect statistics on the performance of model inference and obtain performance data.

This tool has two executable programs: **model_run_tool** and **data_proc_tool**.

model_run_tool: enables the underlying performance statistics function (Profiling), perform model inference, and generate performance statistics.

data_proc_tool: converts the generated performance statistics into a CSV file.

2 About This Document

2.1 Procedure

> **NOTICE**
>
> The operations of enabling Profiling and performing inference by **model_run_tool** take effect only in the current inference.

Step 1 Decompress the DDK. Push all the files in **ddk/tools/tools_sysdbg** to **/data/local/tmp** on the mobile phone environment. (The files must be placed in **/data/local/tmp** or its subdirectories.)

Step 2 Add the path of dynamic link libraries, which must be the same as the path specified in Step 1. Run the **export LD_LIBRARY_PATH=***{Path for storing the .so files in Step 1}*.

Step 3 Grant the execute permission on **model_run_tool** and **data_proc_tool**. Run the **chmod +x model_run_tool data_proc_tool** command.

Step 4 Execute **model_run_tool** to complete inference and generate intermediate data. For details, see Table 2-1.

Step 5 Execute **data_proc_tool** to generate analysis data. For details, see Table 2-2

Step 6 Analyze the data. For details, see Table 2-3 and Table 2-4.

----End

2.2 Parameter Description

Table 2-1 Parameters of model_run_tool

Parameter	Description	Required (Y/N)	Default Value
--model	Offline model path. Note: The path must be **/data/local/tmp** or its subdirectory. For example: `--model=/data/local/tmp/hiai_model.om`	Yes	N/A

--input	Path of the input data file(s) for inference.
	Note: The path must be **/data/local/tmp** or its subdirectory.
	If there are multiple input files, separate them with commas (,). Spaces are allowed before and after the commas. For example:
	`--input=/data/local/tmp/input1.dat`
	`--input="./data/local/tmp/input1.dat, ./data/local/tmp/input2.dat, ./data/local/tmp/input2.dat"`
		No	Generated random input data

--input_type	Data type of the input tensor. Supported options: FP32, FP16, INT32 and UINT8, INT8, INT16, BOOL, INT64, UINT32, DOUBLE If there are multiple input tensors, separate them using commas (,). Note: If the input tensor is of type FP32/INT32/UINT32, you can leave this parameter unspecified.	No	N/A
--output_type	Data type of the output tensor. Supported options: FP32, FP16, INT32 and UINT8, INT8, INT16, BOOL, INT64, UINT32, DOUBLE If there are multiple output tensors, separate them using commas (,). Note: If the output tensor is of type FP32/INT32/UINT32, you can leave this parameter unspecified.	No	N/A

Parameter	Description	Required (Y/N)	Default Value
--output_dir	Directory for storing the inference result. Note: The directory must be **/data/local/tmp** or its subdirectory. *Note: The directory must exist. Dynamic directory creation is not supported.* The inference result is saved as a binary file named **output_${index}**.	No	Current directory
--enable_item	Enables or disables Profiling. **0** or **off**: disabled **1** or **profiling**: enabled If it is set to other values, an error is returned.	No	0/off
--dev_perf	Computing power level required for model inference. **1**: LOW level **2**: NORMAL level **3**: HIGH level **4**: EXTREME level	No	3
--aipp_format	Format of the input image for AIPP. *Note: This parameter needs to be explicitly specified if the model is generated in compatible format (old AIPP format).* Supported formats: YUV420SP_U8, XRGB8888_U8, YUV400_U8, ARGB8888_U8, YUYV_U8, YUV422SP_U8, AYUV444_U8. When there are multiple input files of different formats, separate the formats with commas (,). Example: **--aipp_format="YUV420SP_U8, XRGB8888_U8, YUV400_U8"**	No	N/A
--version	Queries the version.	No	N/A
--help	Displays help and usage information.	No	N/A

Table 2-2 Parameters of data_proc_tool

Name	Description	Mandatory (Y/N)	Default Value
--result_path	Directory for storing performance profiling results, that is, converted CVS files. Note: The directory must be **/data/local/tmp** or its subdirectory. Note: The directory must exist. Creating a directory dynamically is not supported. NOTE • The directory must exist. Creating a directory dynamically is not supported. • Two CSV files are generated, including the model execution process table (***_model.csv**) and operator execution record table (***_op.csv**).	No	Current directory
--version	Queries the version.	No	N/A
--help	Displays help and usage information.	No	N/A

Table 2-3 Description of the model execution process table

Item	Description
item event	Event type. The options are: • **load begin**: Model loading starts. • **load end**: Model loading ends. • **inference begin**: Model inference starts. • **inference end**: Model inference ends. • **unload begin**: Model unloading starts. • **unload end**: Model unloading ends. NOTE Model inference is performed twice in the tool execution process. Therefore, both **inference begin** and **inference end** have two records.
enter time	Event reporting time.
total time	Execution time of each event, in μs. The **begin** column is fixed at **0**, and the total time is recorded in the **end** column.

Table 2-4 Description of the operator execution record table

Item	Description
op name	Operator name. It depends on the model structure. Example: **transdata_for_nd_0**
start time	Start time of operator execution.
total time	Total time of operator execution, in μs. NOTE During one inference, if there are NPU operators with the same name, execution time of these operators needs to be added up to obtain the total time.
device type	Device type. • NPU • CPU

2.3 Example Demonstration

Query the tool version.

```
./model_run_tool --version
./data_proc_tool --version
```

Print the help information.

```
./model_run_tool --help
./data_proc_tool --help
```

Perform regular model inference with multiple inputs.

```
./model_run_tool --model=./offline.om --input ="./input1.dat, ./input2.dat, ./input3.dat" --output_dir=./output
```

Perform AIPP model inference with a single input.

```
./model_run_tool --model=./aipp_offline.om --input=image.bin --output_dir=./output --aipp_format=YUV420SP_U8
```

Perform AIPP model inference with multiple inputs.

```
./model_run_tool --model=./aipp_offline.om --input="./image1.bin, ./image2.bin, ./image3.bin" --output_dir=./output --aipp_format="YUV420SP_U8,XRGB8888_U8,YUV400_U8"
```


Enable Profiling inference. The performance profiling result (CSV file) is generated in the directory specified by result_path.

```
./model_run_tool --model=./offline.om --input_files="./input1.dat" --output_dir=./output --
enable_item=1
./data_proc_tool --result_path=./prof_data
```

