NPU validation & debug

May 2024

CCG CPE CCE

Phoebe Hsu

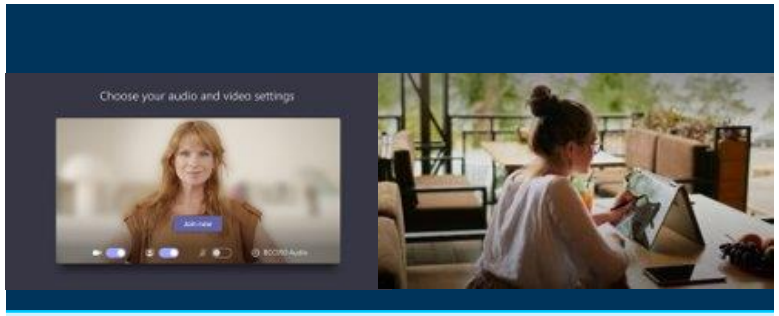
Outline

- Client AI and Roadmap
- Validation
  - Workshop validation
  - NPU benchmarking
  - Procyon (3rd party app)
  - PTAT
  - MSFT WSE Tool

- Logs/Debug
  - Intel STC v2.3: NPU/MEP Camera trace/UMD/KMD, Xperf
  - GPUView
  - Windbg-Live kernel dump/BSOD memory dump
  - Traceview
  - WPR
  - PTAT

Client AI and Roadmap

Transforming the PC Experience



AI Today
Enhancements

Elevated video collaboration & streaming
Enhanced Audio effects
Creator and Gaming effects

AI Tomorrow
Everything

AI Assistants know your daily context
More creative, productive, & collaborative
Across everything you do

Cloud
Massive scalable compute
High Latency
Privacy Concerns
Expensive

Client
Massive distributed scale
Low Latency
Improved Privacy
Lower Cost (to ISV)

8Q Client AI Roadmap

CPU	iGPU	NPU
Fast Response, Low latency	AI integrated with 3D/render/ media pipelines; high batch size	Sustained Workloads, dedicated offload

	Meteor Lake		Lunar Lake (MX 17W)	Arrow Lake		Panther Lake	
CPU	6 (45W)		5	9 (45W)	14	5	11 (45W)
	U-series	H-series		H-series	S-series	U-series	H-series
iGPU	8	19	59	72	9	41	123 (45W)
NPU	11		45	11	13	48	48
Platform TOPS	25	36	109	Up to 92	Up to 36	98	182

8Q Client AI Roadmap (Cont.)

Client AI Workloads are Diverse
No Single Compute Unit Meets All Key Needs

 Bursty, Latency sensitive

 Sustained, Battery life sensitive

 Periodic, Throughput sensitive

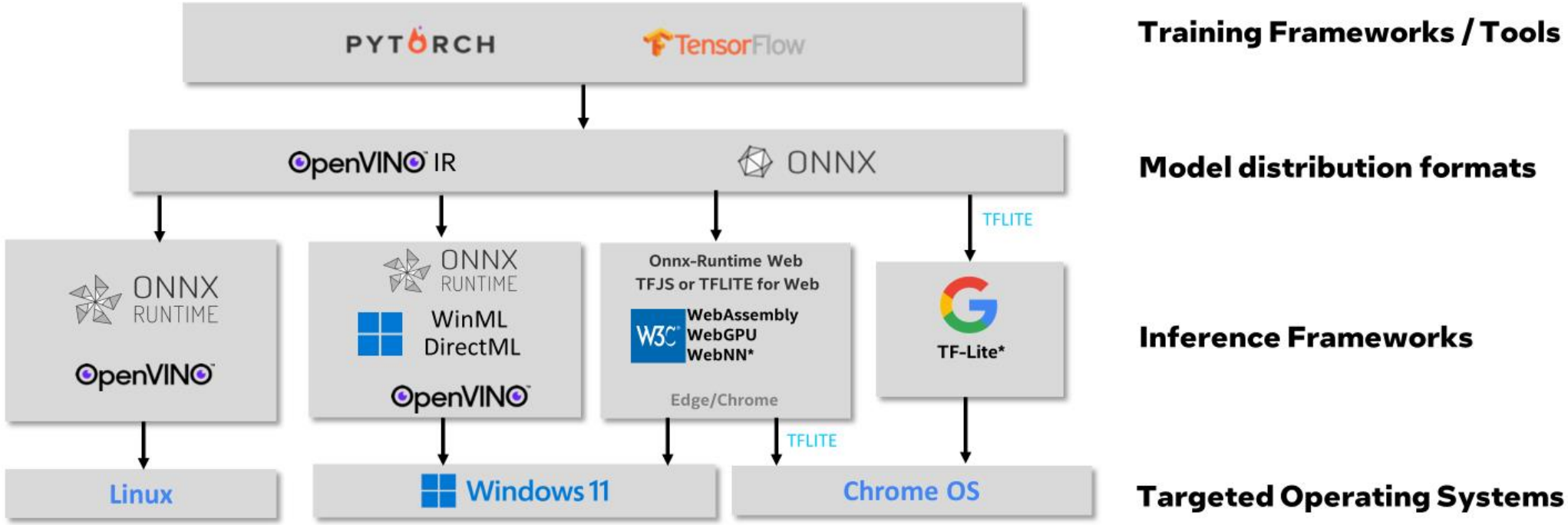
HW	Value	RPL	MTL	ARL	LNL MX	PTL
CPU	SW Programmability; low latency, single inference tasks	**AVX-256 VNNI** H: 4-5 TOPS	**AVX-256 VNNI** H: ~3-6; U: ~2-3 TOPS	**AVX-256 VNNI** H: ~7-9.5; S: 14 TOPS	**AVX-256 VNNI** ~2-5 TOPS	AVX2+ TOPS - H: Up to 11; U: 5
iGPU	AI integrated with 3D/render/ media pipelines; high batch size	**DP4A** H/U: up to 9 TOPS S/HX: 3 TOPS	**DP4a** (U, H) H: up to 19 TOPS U: up to 8 TOPS	**DP4a (U, S, HX)** ~9 TOPS **ARL H w/Xᵉ Matrix Extensions (XMX)** Up to 72 TOPS	**DP4a + Xᵉ Matix Extensions (XMX)** Up to 59 TOPS	**DP4a + XMX** H: Up to ~123 TOPS U: up to 41 TOPS
iNPU	Dedicated AI Offload, Power efficiency for Battery Life	NA	**NPU 2.7** TOPS - H: 11 TOPS; U: 9.5-11; ARL S, HX: 13		**NPU 4.0** Up to 45 TOPS	**NPU 5.0** Up to 48 TOPS

TOPS will vary slightly based on power & frequency of each sku

The Right Frameworks for Innovation and Scale:

DirectML OpenVINO ONNX W3C WebAssembly WebGPU WebNN

Client AI Inference SW For Deployment 2023+



OpenVINO
Underlying technology ingredient and vertical option for Windows & Linux.

OpenVINO powers PyTorch, ONNX-Runtime and ChromeOS interfaces for deployment.

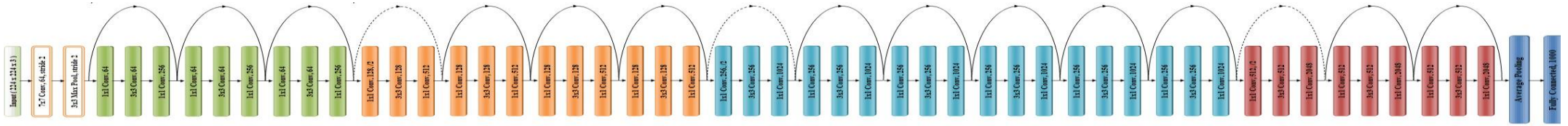
Windows 11
DirectML, OpenVINO

ONNX-Runtime (+ choice DirectML, MLAS, OpenVINO EP's)

TF-Lite

PYTORCH
Native deployment support via torch.compile in development for NPU, available for CPU/GPU

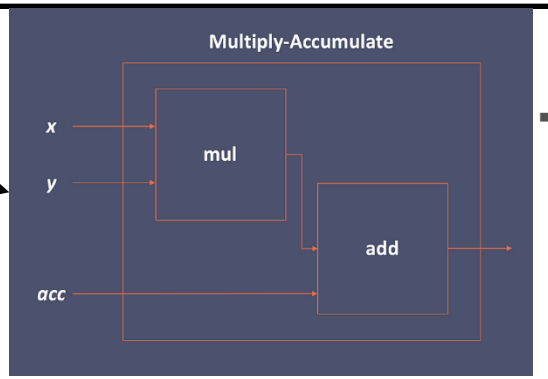
Resnet50 Example Based on MTL



Input 224x224

Workload
(MACs operations on NPU)

Output (frame)



One multiply-accumulate is two operations

- Peak TOPS (pTOPS) = Peak Theoretical Max Performance

  pTOPS = max frequency * (MAC/Clock) * 2

  NPU is 11 pTOPS

- Effective TOPS (eTOPS) = Real Performance on a given AI Workload (the efficiency of pTOPS)

  eTOPS = (fps * each frame GOPs)/1000

  *NPU 8.2 eTOPS = (1000 * 8.216)/1000*

- Operations per frame: constant value per network, for Resnet50 it is 8.216 GOPs

- We use ResNet50: a common network + a good mix of a memory and compute bound network. Is it Perfect? -> No, but it is better than pTOPS as eTOPS shows real workload measured across many HW configs

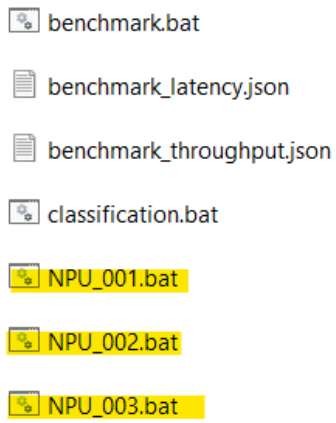
- AI Benchmark for Client: Not 1 standard Today (UL Procyon Redowa (POR)/MLPerf/GeekBenchML)

	pTOPS	Resnet50 FPS	eTOPS	Efficiency
MTL NPU	11	1000	8.2	75%

Validation

Workshop validation

- Download [Intel NPU OEM Workshop Validation Software Kits]
  - \validation_mtl_arl_lnl\validation\test_scripts



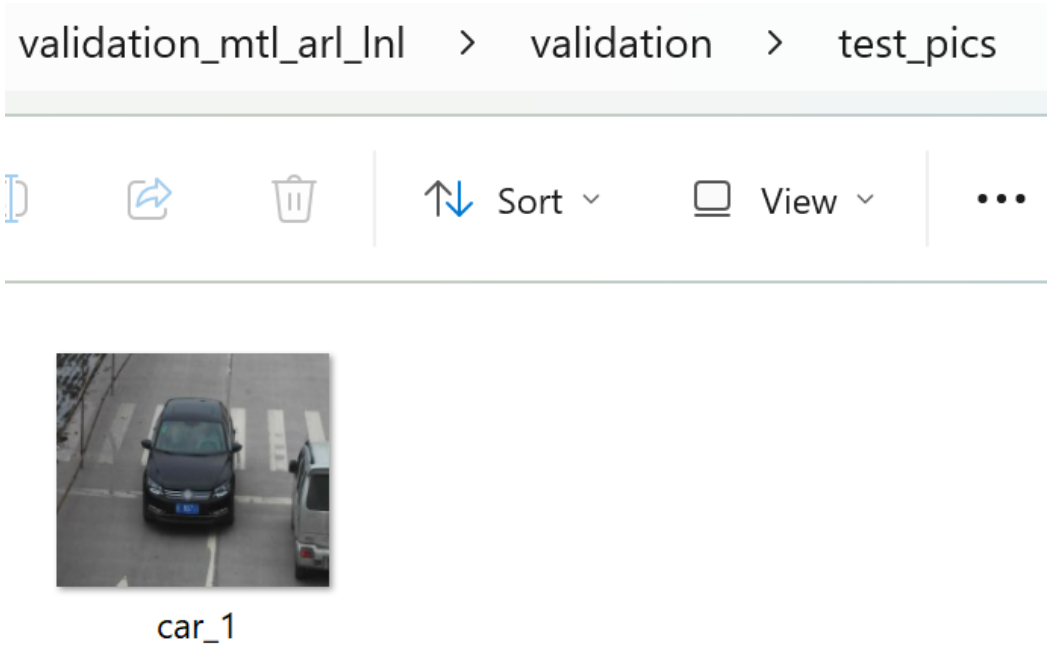
- More MSFT MEP test case
  - [Meteor Lake VPU MEP Enablement and Validation Testcases in OEM Platform]
  - Hot-plug AC adaptor/type C monitor, MS/S4/Reboot stress test, screen on idle, teams call, launch/close camera app, on/off effects...etc.

Workshop validation

NPU_001	NPU	Verify NPU with multiple inference models support using OpenVINO framework (ResNet/MobileNet)	Verify NPU with multiple inference models support using OpenVINO framework. The models will be run sequentially using the application. A single instance of Image Classification will be executed using ResNet followed by MobileNet.	a) Command runs successfully. No errors, BSODs, hangs, TDRs and crashes are observed. b) Open window showing tiles of classified image and the expectation is that the prediction of the input is correct.
NPU_002	NPU	Verify NPU driver for D0 and D3 states	Verify NPU driver for D0 and D3 states. The OpenVINO model used is ResNet 50 and the application used to run the model on VPU IP is Image Classification. The transition from D3 to D0 and D0 to D3 is checked using SocWatch tool	a) Open SoCWatch without error, SoC Watch command : "socwatch.exe -f sys -o filename" b) User should be able to navigate to the mentioned path in device manager and note the current power state from it. If the NPU device is not working Power state should be D3. If the NPU device is working, the power state should be D0. c) Run inference and Stop log collection & verify VPU device should go to D3->D0->D3
NPU_003	NPU	Verify NPU functionality in DC mode real battery	Verify offloading workload to NPU with SUT connected to real battery	a) NPU utilization is observed to increase. NPU Compute utilization > 0% when inferencing is ongoing. b) Command runs successfuly. No errors, BSODs, hangs, TDRs and crashes are observed. c)The first inference time, throughput and latency results are non zero value
NPU_004	NPU	Verify the MEP camera effect on NPU	Verify the MEP installation and optin on NPU	a) Modify the ivd64extn.inf in NPU driver, default SUBSYS HWID LNL (8086_643E), MTL (8086_7D1D) b) Install the ivd64extn.inf and check the status in Device Manager c) Install MEP base package and camera component package, then check the status in Device Manager Software components, Windows Studio Effects should be available d) Optin the camera with CameraOptinUtils from MSFT, or add "FSMEnableMsEffects: REG_DWORD: 0x1" under Device Interface Node registry e) If the MEP and Camera Opt-in are successful, control the MEP camera effects in Windows Setting and test it with Windows Camera application

Workshop validation

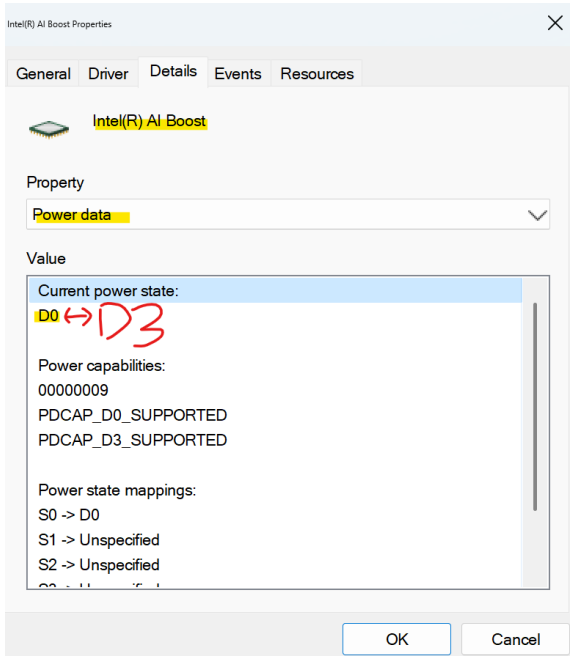
```
Image C:\openvino\test_scripts\\..\test_pics\car_1.bmp

classid probability
------- -----------
656     12.6250000
734     10.9765625
705     10.8203125
468     10.4453125
654     10.1640625
436     9.7421875
874     9.3281250
407     9.1171875
829     9.1562500
675     8.9140625

[ INFO ] Execution successful
```



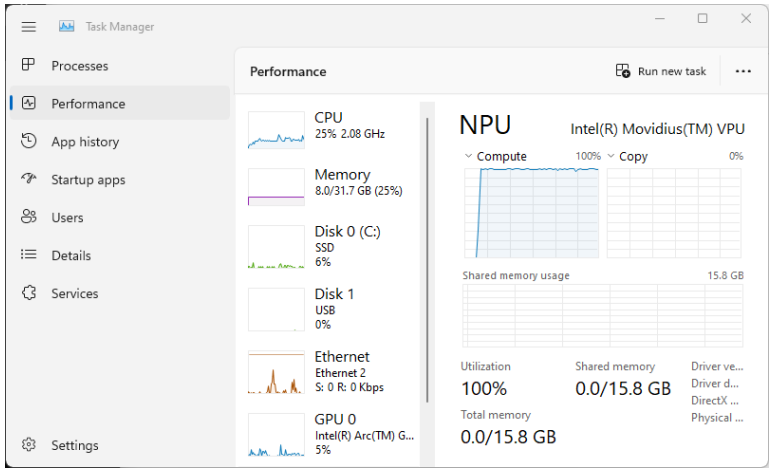
656: 'minivan',
734: 'police van, police wagon, paddy wagon, patrol wagon, wagon, black Maria'
705: 'passenger car, coach, carriage'

1000 class ids to human readable labels
https://gist.github.com/yrevar/942d3a0ac09ec9e5eb3a

Workshop validation




```
[Step 1/11] Parsing and validating input arguments
[ INFO ] Parsing input parameters
[ INFO ] Network is compiled
[ INFO ] Files were added: 1
[ INFO ]     C:\Users\Local_Admin\Desktop\validation_package\test_scripts\\..\test_pics\car_1.bmp
[Step 2/11] Loading Inference Engine
[ INFO ] InferenceEngine:
        IE version ......... 2022.1.1
        Build ........... 2022.1.1-000-a6d0a3c71b1-HEAD
[ INFO ] Device info:
        VPUX
        openvino_intel_vpux_plugin version ......... 2022.1.1
        Build ........... 2022.1.1-000-a6d0a3c71b1-HEAD

[Step 3/11] Setting device configuration
[Step 4/11] Reading network files
[ INFO ] Skipping the step for compiled network
[Step 5/11] Resizing network to match image sizes and given batch
[ INFO ] Skipping the step for compiled network
[Step 6/11] Configuring input of the model
[ INFO ] Skipping the step for compiled network
[Step 7/11] Loading the model to the device
[ INFO ] Import network took 455.97 ms
[Step 8/11] Setting optimal runtime parameters
[Step 9/11] Creating infer requests and filling input blobs with images
[ INFO ] Network input 'result.1' precision U8, dimensions (NCHW): 1 3 224 224
[ WARNING ] Some image input files will be duplicated: 2 files are required but only 1 are provided
[ INFO ] Infer Request 0 filling
[ INFO ] Prepare image C:\Users\Local_Admin\Desktop\validation_package\test_scripts\\..\test_pics\car_1.bmp
[ WARNING ] Image is resized from (749, 637) to (224, 224)
[ INFO ] Infer Request 1 filling
[ INFO ] Prepare image C:\Users\Local_Admin\Desktop\validation_package\test_scripts\\..\test_pics\car_1.bmp
[ WARNING ] Image is resized from (749, 637) to (224, 224)
[Step 10/11] Measuring performance (Start inference asynchronously, 2 inference requests, limits: 10000 ms duration)
[ INFO ] First inference took 5.12 ms

[Step 11/11] Dumping statistics report
Count:      5454 iterations
Duration:   10005.76 ms
Latency:    3.47 ms
Throughput: 545.09 FPS
Press any key to continue . . .
```



NPU benchmarking

- We use ResNet50 to measure eTOPS
  - a common network + a good mix of a memory & compute bound network

- Effective TOPS (eTOPS) = real performance on a given AI workload (the efficiency of pTOPS)
  - eTOPS = (fps * each frame GOPs)/1000
  - Operations per frame: constant value per network, for Resnet50 it's 8.216 GOPs

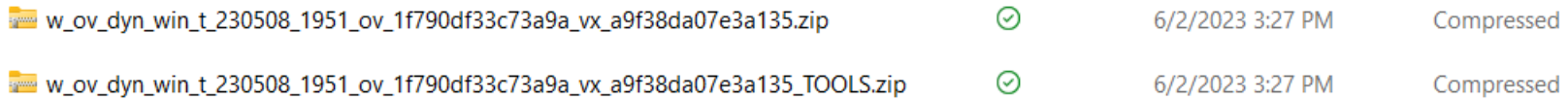
- See MTL/LNL Platform NPU Benchmarking for details

Run Resnet50 on MTL NPU

1. Download NPU driver from RDC and then install it and its extension driver w/ 4-part ID
2. Install one of Python 3.8 - 3.11 and following below steps to create virtual environment, ov_24.0 under C:\Users\Public\

```
$ cd C:\Users\Public\
$ python -m venv ov_24.0
$ ov_24.0\Scripts\activate
$ python -m pip install --upgrade pip
$ pip install openvino==2024.0.0
```

3. Download OpenVINO package, [w_openvino_toolkit_mtl_23ww19.zip](), form WW22'23 BKC release to get Resnet50 model, and then unzip to C:\Users\Public, and you can see below 2 zip files



4. Unzip "w_ov_dyn_win_t_230508_1951_ov_1f790df33c73a9a_vx_a9f38da07e3a135_TOOLS.zip" to C:\Users\Public
5. Create npu_config.json and add below line to it, and then save it at C:\Users\Public\

```
{"NPU": {"NPU_COMPILER_TYPE": "DRIVER", "NPU_COMPILATION_MODE_PARAMS": "enable-activation-sparsity=true" }}
```

6. Pre-check:

Required Files	File Path
ResNet50 int8 with 50% sparsity (IR model)	• C:\Users\Public\tools\ir_models\IRv11\20230309_vpu-models-mtl-por-ir_v11_ov_2022.3.0-9752fafe8eb\resnet-50-v1_5-sparse50\onnx\FP16-INT8\resnet-50-v1_5-sparse50.xml • C:\Users\Public\tools\ir_models\IRv11\20230309_vpu-models-mtl-por-ir_v11_ov_2022.3.0-9752fafe8eb\resnet-50-v1_5-sparse50\onnx\FP16-INT8\resnet-50-v1_5-sparse50.bin
NPU configuration	C:\Users\Public\npu_config.json

Run Resnet50 on MTL NPU (Cont.)

7. Open cmd terminal and then run below commands to setup environment and prepare model before run benchmark_app

```
$ C:\Users\Public\ov_24.0\Scripts\activate
$ cd C:\Users\Public\
$ copy C:\Users\Public\tools\ir_models\IRv11\20230309_vpu-models-mtl-por-ir_v11_ov_2022.3.0-9752fafe8eb\resnet-50-v1_5-sparse50\onnx\FP16-INT8\resnet-50-v1_5-sparse50.* .
```

8. Run NPU in Throughput mode

```
$ benchmark_app -d NPU -m resnet-50-v1_5-sparse50.xml -t 180 -layout [NCHW] -ip f16 -op f16 -hint throughput -load_config npu_config.json
```

9. Run NPU in Latency mode

```
$ benchmark_app -d NPU -m resnet-50-v1_5-sparse50.xml -t 180 -layout [NCHW] -ip f16 -op f16 -api sync -load_config npu_config.json
```

Result for reference:

- Configuration:
  - MTL-H 28W
  - NPU 2196 driver
  - OpenVINO 2024.0
  - MEMORY size: 16GB
- Performance on MTL NPU:

Execution Mode	NPU
Latency	6.43
Throughput	8.47

Unit: eTOPs

One-click Script for Test

- Pre-check:

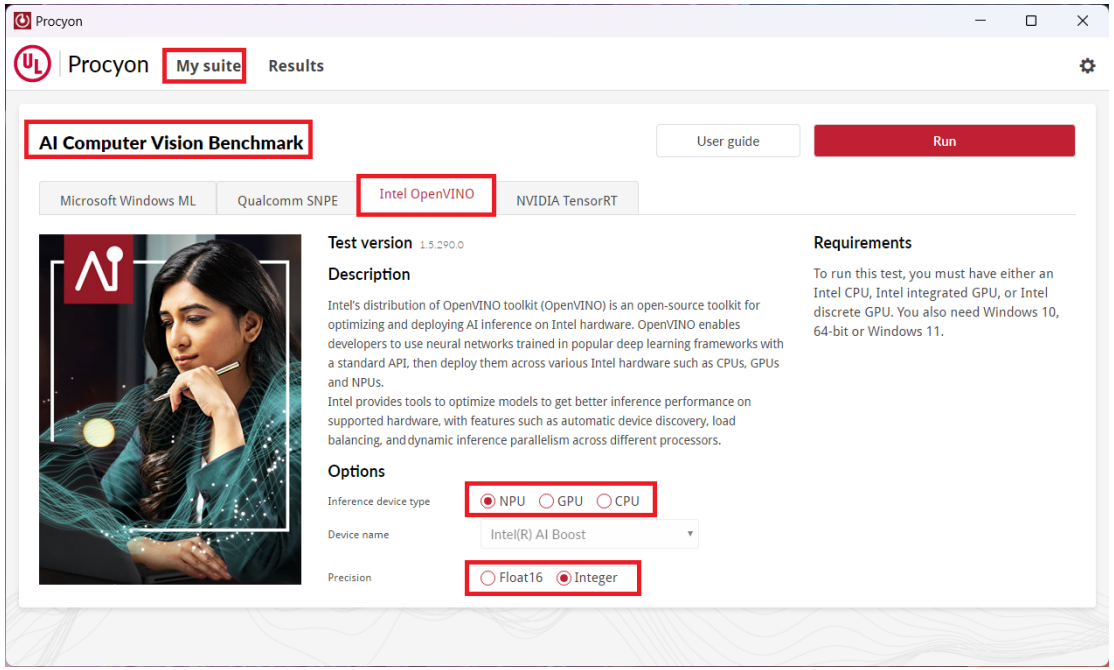
Required Files	File Path
Python environment	C:\Users\Public\ov_24.0\
ResNet50 int8 with 50% sparsity (IR model)	• C:\Users\Public\resnet-50-v1_5-sparse50.xml • C:\Users\Public\resnet-50-v1_5-sparse50.bin
NPU configuration	C:\Users\Public\npu_config.json

- Create resnet50.bat and add below lines into it, and then save it at C:\Users\Public\

```
call C:\Users\Public\ov_24.0\Scripts\activate
cd C:\Users\Public\
benchmark_app -d NPU -m resnet-50-v1_5-sparse50.xml -t 180 -ip f16 -op f16 -layout [NCHW] -api sync -load_config npu_config.json
benchmark_app -d NPU -m resnet-50-v1_5-sparse50.xml -t 180 -ip f16 -op f16 -layout [NCHW] -hint throughput -load_config npu_config.json

pause
```

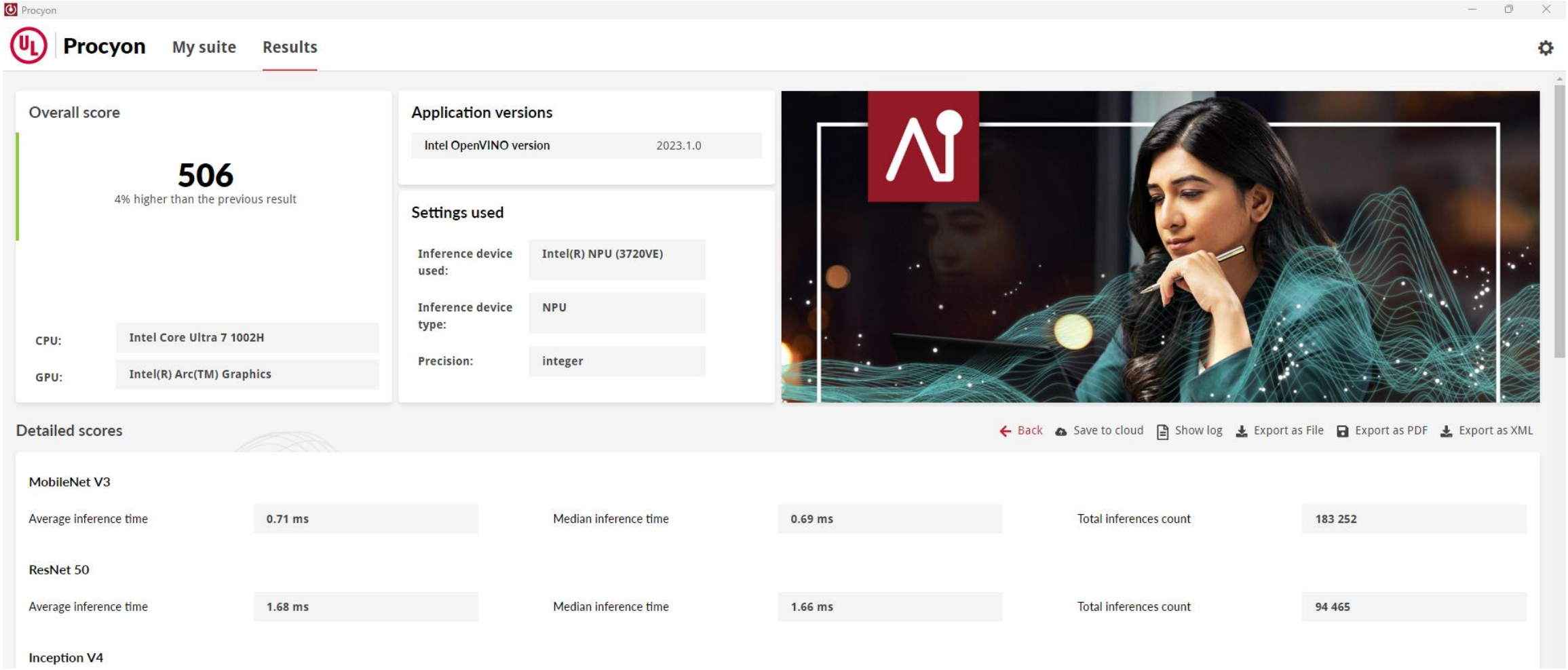
Procyon

- Apply access
  https://wiki.ith.intel.com/display/ITSVPUCSE/How+to+get+Redowa+packages
- Procyon package
  \\pwb-release.amr.corp.intel.com\release\Futuremark\Procyon (Release & Pre-release)
- Procyon system configuration
  - Procyon version: >=2.6.896
  - NPU version: 31.0.100.1688
  - Score: >= 480 (+- 5%)
  - AC, Best Performance
  - NPU, select OpenVino tab



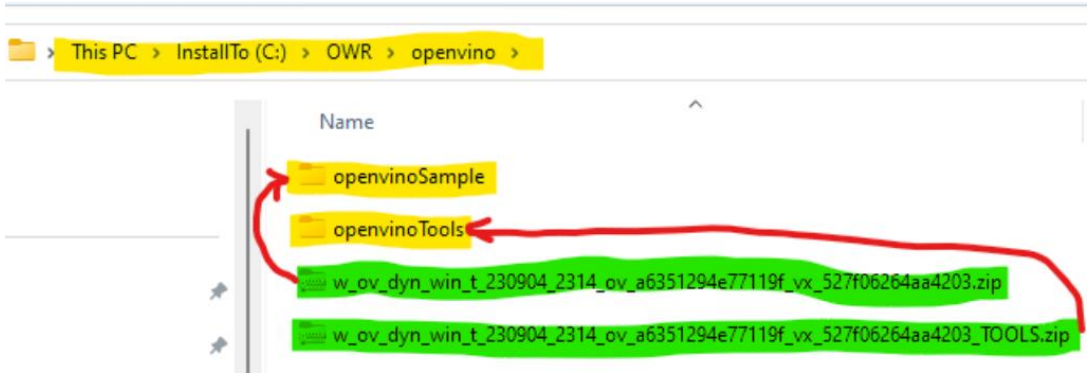
Procyon

Example on MTL RVP



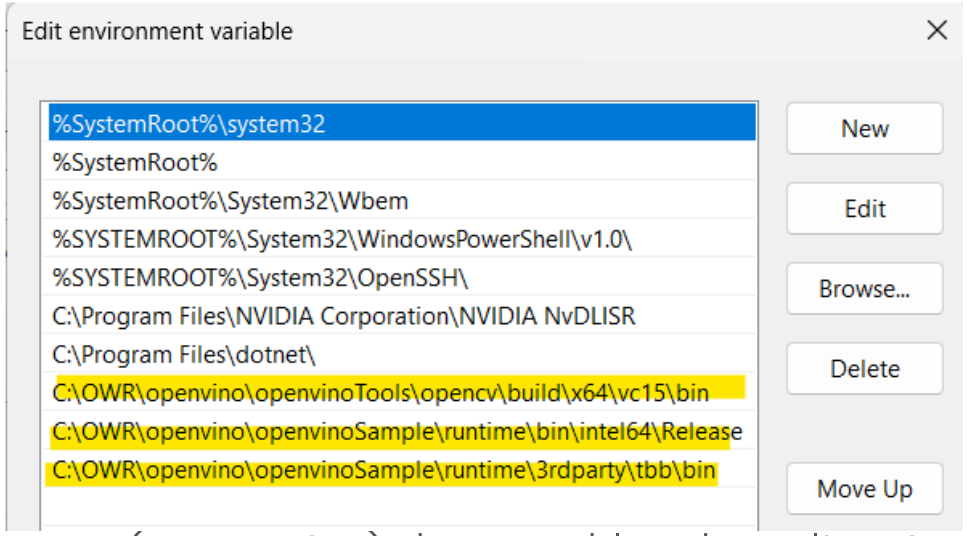
PTAT

- Download Intel® Power And Thermal Analysis Tool
- Download the latest toolkit "w_openvino_toolkit_YYWWNN".zip (internal)
- Follow below steps to make the OpenVINO library files available for running the NPU workload
  - There are two w_vo_*.zip files of which one has the sample files, and another has tool files.
  - Unzip the sample zip file and tool zip file to a directory and rename it to "openvinoSample" and "openvinoTools



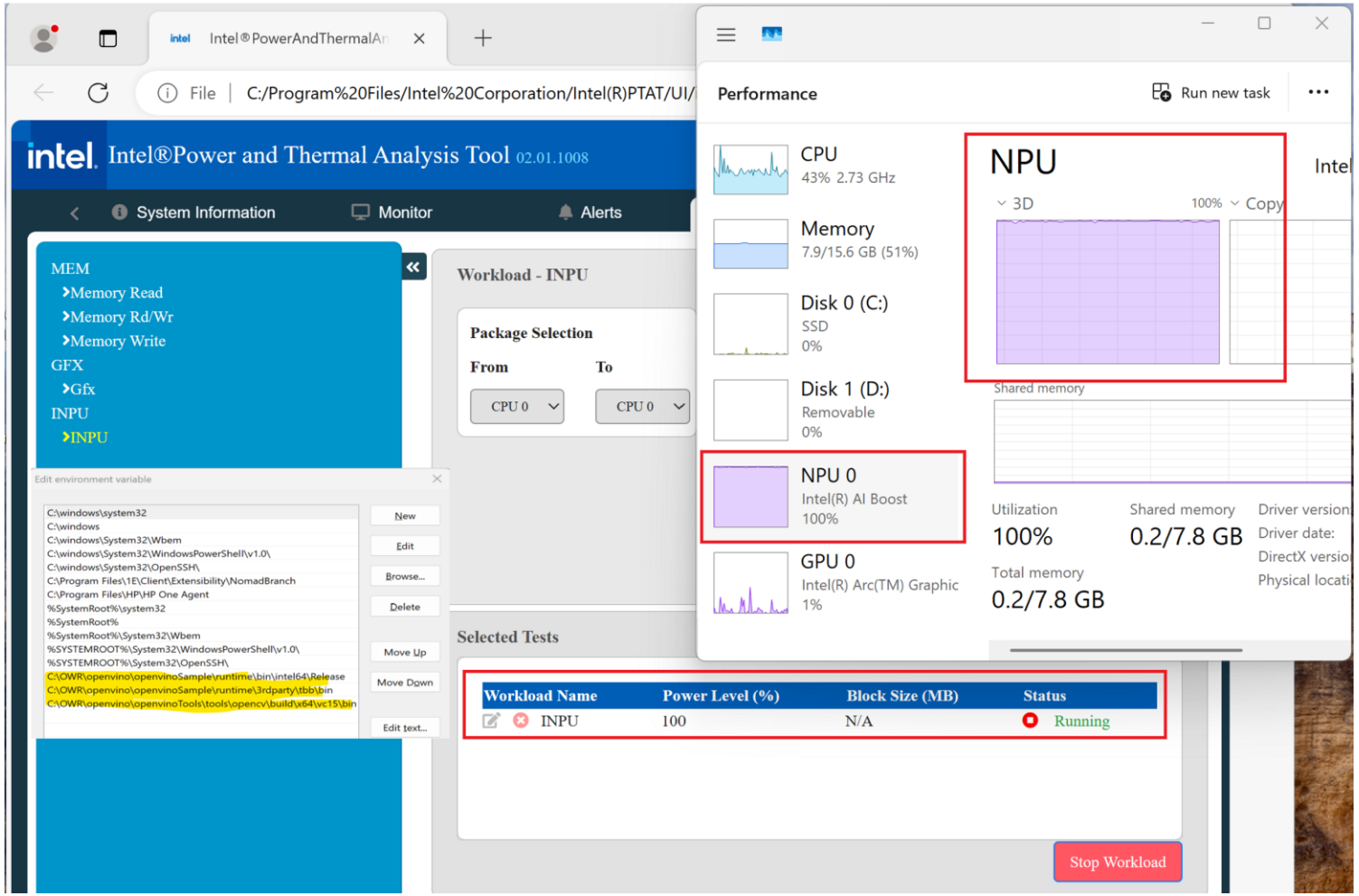
PTAT

- add the below three paths to the environment variables as follows (add to both systems and user path variables)



- copy(overwrite) that workload application “benchmark_app.exe” to the PTAT installed directory
C:\Program Files\Intel Corporation\Intel(R)PTAT\Vpu”

PTAT


Intel®PowerAndThermalAr
File | C:/Program%20Files/Intel%20Corporation/Intel(R)PTAT/UI/
intel. Intel®Power and Thermal Analysis Tool 02.01.1008
System Information
Monitor
Alerts
MEM
Memory Read
Memory Rd/Wr
Memory Write
GFX
Gfx
INPU
INPU
Workload - INPU
Package Selection
From
To
CPU 0
CPU 0
Edit environment variable
C:\windows\system32
C:\windows
C:\windows\System32\Wbem
C:\windows\System32\WindowsPowerShell\v1.0\
C:\windows\System32\OpenSSH\
C:\Program Files\1E\Client\Extensibility\NomadBranch
C:\Program Files\HP\HP One Agent
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\OWR\openvino\openvinoSample\runtime\bin\intel64\Release
C:\OWR\openvino\openvinoSample\runtime\3rdparty\tbb\bin
C:\OWR\openvino\openvinoTools\tools\opencv\build\x64\vc15\bin
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
Selected Tests
Workload Name
Power Level (%)
Block Size (MB)
Status
INPU
100
N/A
Running
Stop Workload
Performance
Run new task
CPU
43% 2.73 GHz
Memory
7.9/15.6 GB (51%)
Disk 0 (C:)
SSD
0%
Disk 1 (D:)
Removable
0%
NPU 0
Intel(R) AI Boost
100%
GPU 0
Intel(R) Arc(TM) Graphic
1%
NPU
3D
100%
Copy
Shared memory
Utilization
100%
Shared memory
0.2/7.8 GB
Total memory
0.2/7.8 GB
Driver version
Driver date:
DirectX versio
Physical locati

MSFT WSE Tool

- Download WSE Tool (WseEnablingStatus.exe and WsePerformanceAssessmentTool.exe)
- WseEnablingStatus.exe
  - Checks the status of MEP installed on the system

```
- System Name: device name
- System OS Info: operating system info
- Opt-In Camera Status: True/False
- Opt-In Camera FriendlyName: friendly name of camera
- Opt-In Camera Hardware ID: hardware id of camera
- Opt-In Camera Driver: driver version of camera
- Windows Studio Effects Camera: version number
- Windows Studio Effects AudioBlur APO: version number (by platforms)
- PerceptionCore.dll: version number [path]
```

  - You will see MEP opt-in camera and Windows Studio Effect version when you run this tool if you opt-in MEP successfully.

```
C:\WseEnablingStatus>WseEnablingStatus.exe
System Name: LAPTOP-CD63QRRN
System OS Info: Windows 10 Pro (26085.1)
Opt-In Camera Status: True
Opt-In Camera FriendlyName: 9MP Camera
Opt-In Camera Hardware ID: USB\VID_0408&PID_546E&REV_0006&MI_00
Opt-In Camera Driver: 10.0.26085.1
Windows Studio Effects Camera: 1.0.38.0
```

MSFT WSE Tool

- The tool will point out the problem if you don’t opt-in MEP correctly.

```
F:\WseEnablingStatus>WseEnablingStatus.exe
can not find 'Windows Studio Effects Camera' in device manager, extension .inf for MEP camera was not correctly deployed
```

```
F:\WseEnablingStatus>WseEnablingStatus.exe
can not find Opt-in camera instance in registry, there is no 'FSMEnableMsEffects' key in registry
```

MSFT WSE Tool

- WsePerformanceAssessmentTool.exe
  - Allows profiling the following metrics

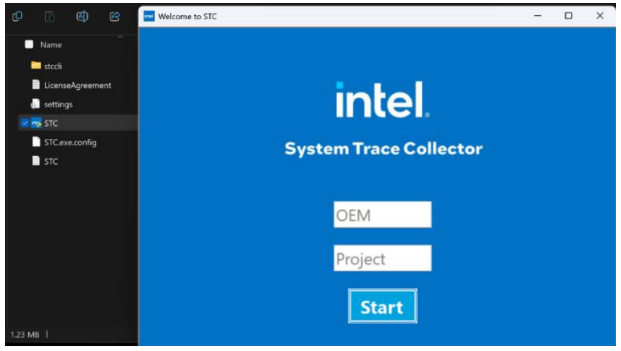
```
- Camera Effects Performance:
- FPS (FrameRate): the average frame per seconds of recorded clip
- timeToFirstFrame: the time cost of successfully loading AI models and applying effects onto 1st camera frame
- avgProcessingTime: the average processing time of dealing with one frame, in milliseconds
- (numberOfFramesAbove33ms/numberOfProcessFrame): the ratio of processing time over 33 milliseconds
- initTimeCameraApp: the time cost between launch camera app and applying effects onto 1st camera frame
```

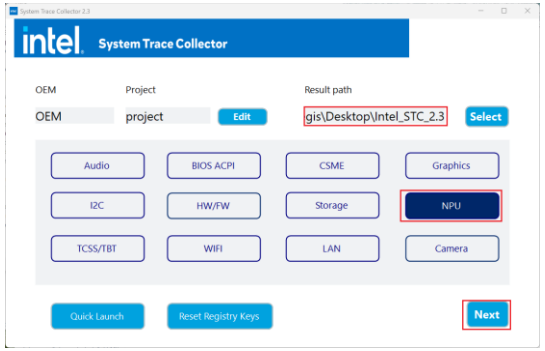
```
PerformanceAssessment [2 / 3] rounds
[Camera Effects Performance]
GenericError - PerceptionCore-v2.3.7
FPS (FrameRate): 24.36 frames/sec, (1920x1080, 00:00:59)
timeToFirstFrame: 6242 millisecs
avgProcessingTime: 21.69 millisecs
(numberOfFramesAbove33ms/numberOfProcessFrame): (0/1568)
InitTimeCameraApp: 7.048 secs

PerformanceAssessment [3 / 3] rounds
[Camera Effects Performance]
GenericError - PerceptionCore-v2.3.7
FPS (FrameRate): 25.34 frames/sec, (1920x1080, 00:00:59)
timeToFirstFrame: 6307 millisecs
avgProcessingTime: 21.29 millisecs
(numberOfFramesAbove33ms/numberOfProcessFrame): (0/1622)
InitTimeCameraApp: 7.116 secs
```

Logs/Debug

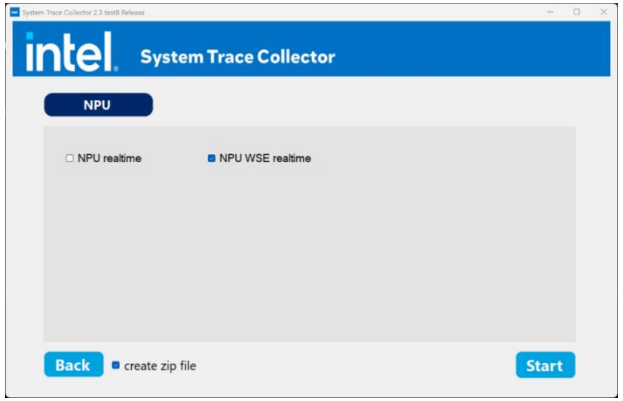
Intel STC v2.3

- To collect Camera and NPU UMD/KMD trace, Xperf log
- Download Intel System Trace Collector (STC) Tool, launch it by STC.exe



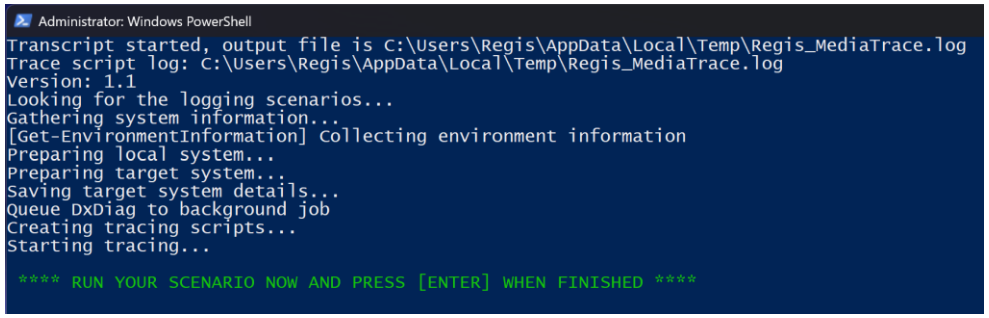


- Select “NPU WSE realtime” (Camera and NPU UMD/KMD)

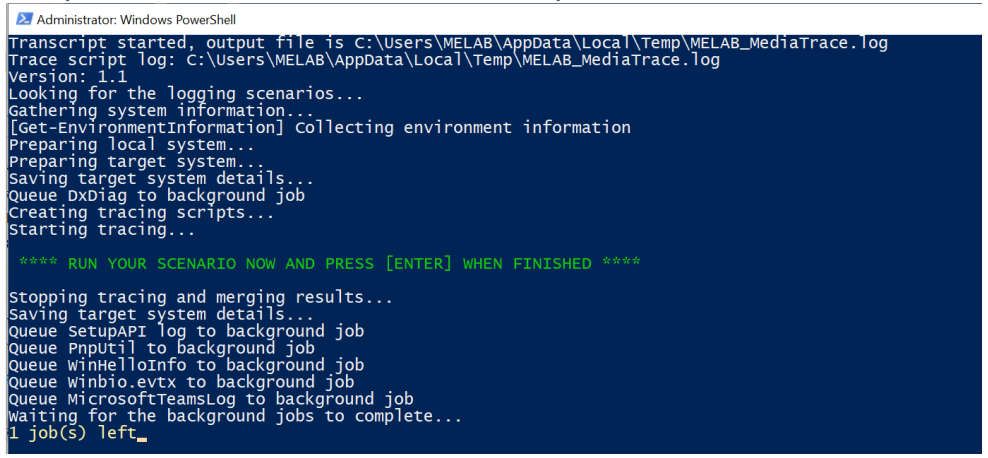


Intel STC v2.3

- You will see another console pop up.

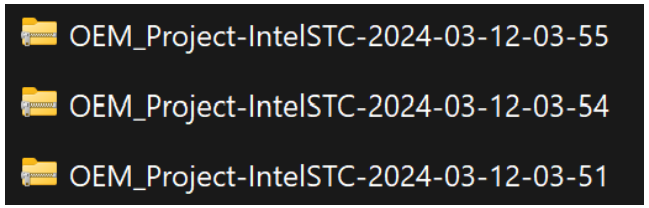


- Reproduce the issue and press ENTER when finished.

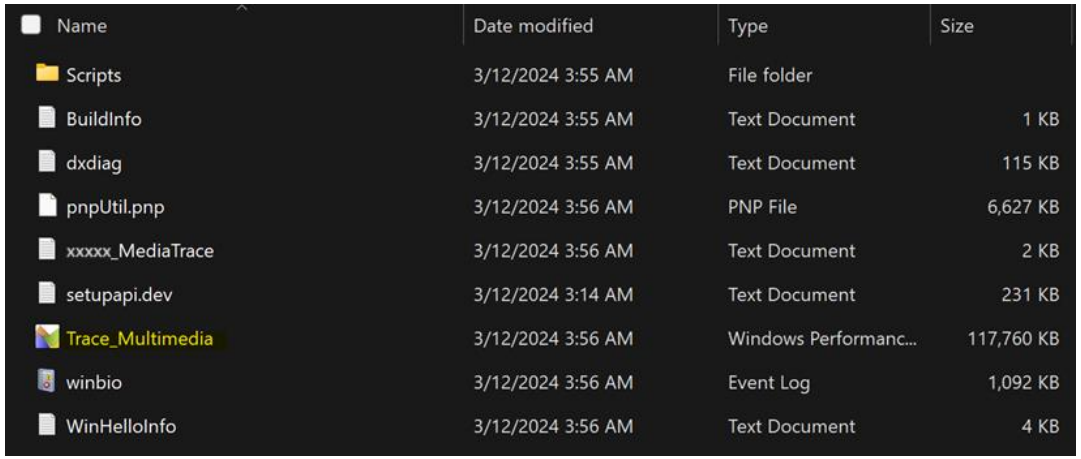


Intel STC v2.3

- It will auto zip the log



- Use Windows Performance Analyzer (WPA) tool in ADK to open Trace_MultImedia.etl

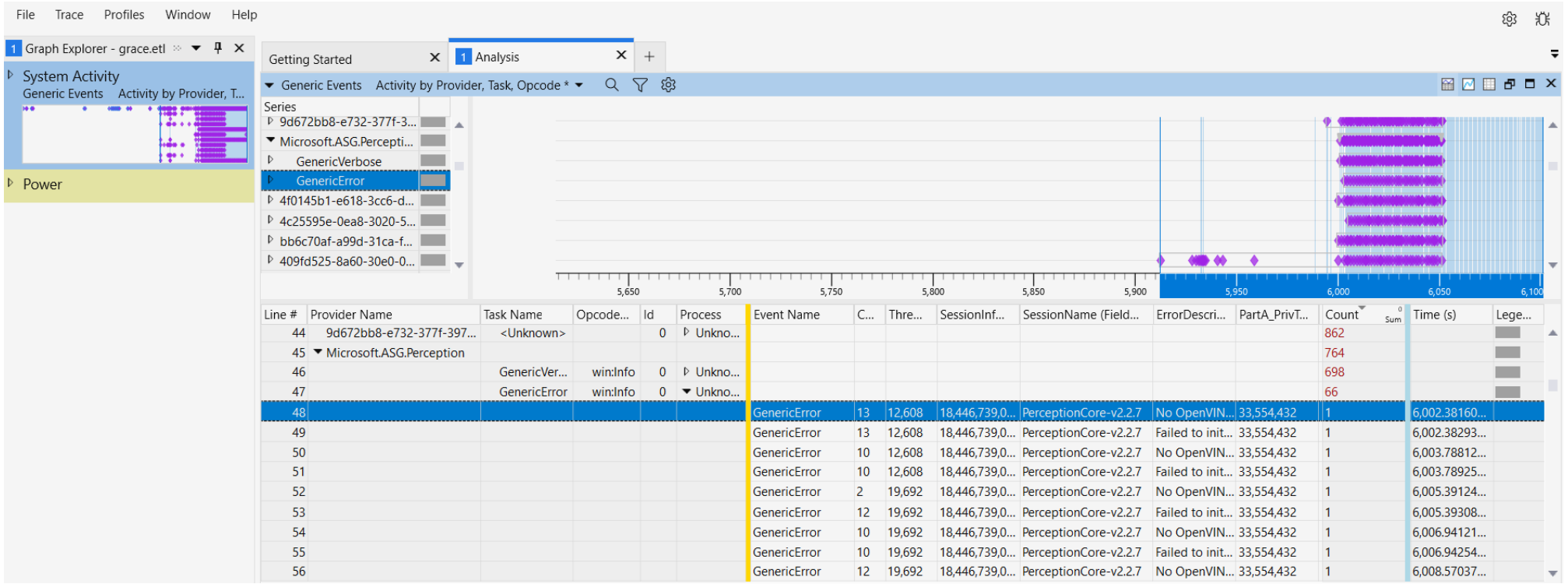


Name	Date modified	Type	Size
Scripts	3/12/2024 3:55 AM	File folder	
BuildInfo	3/12/2024 3:55 AM	Text Document	1 KB
dxdiag	3/12/2024 3:55 AM	Text Document	115 KB
pnpUtil.pnp	3/12/2024 3:56 AM	PNP File	6,627 KB
xxxxx_MediaTrace	3/12/2024 3:56 AM	Text Document	2 KB
setupapi.dev	3/12/2024 3:14 AM	Text Document	231 KB
Trace_Multimedia	3/12/2024 3:56 AM	Windows Performanc...	117,760 KB
winbio	3/12/2024 3:56 AM	Event Log	1,092 KB
WinHelloInfo	3/12/2024 3:56 AM	Text Document	4 KB

Intel STC v2.3

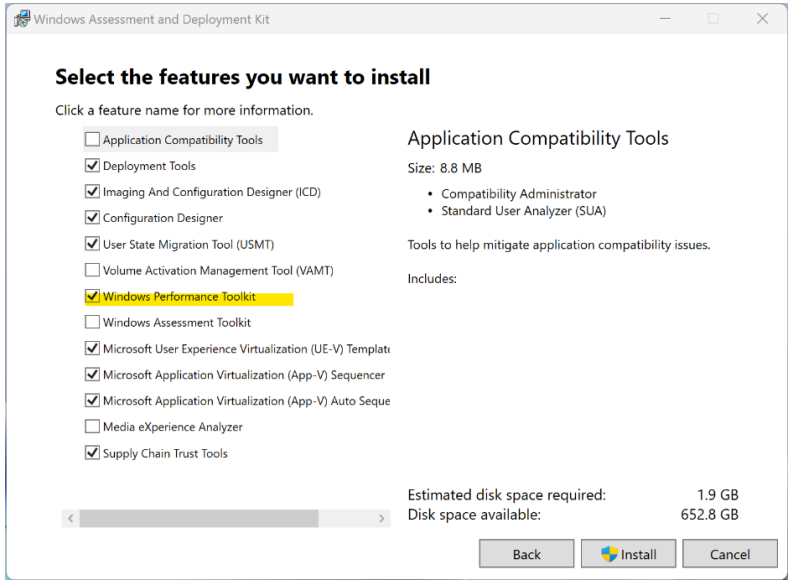
The following strings can be used to collect the MEP related logs.

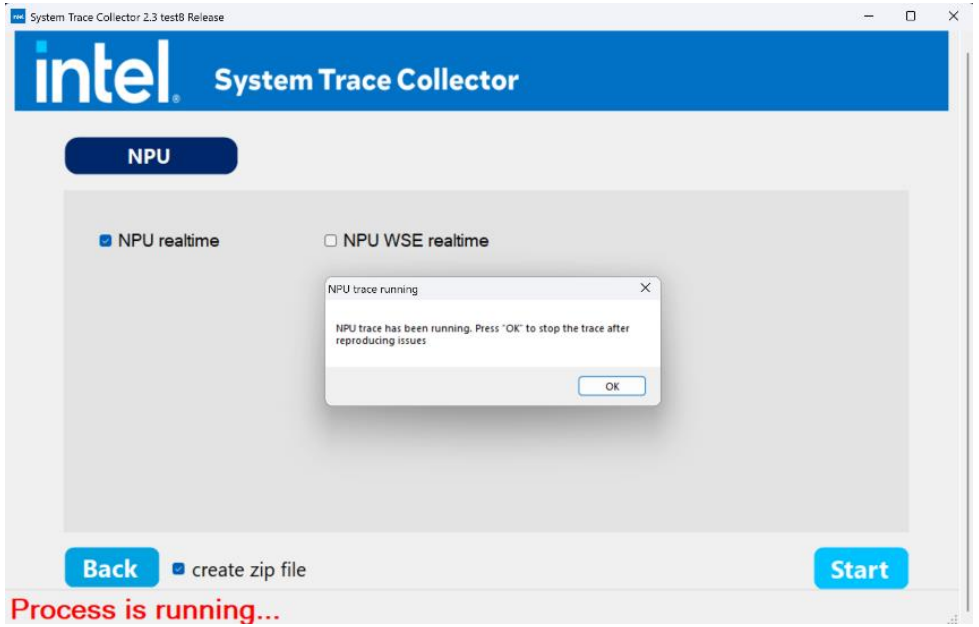
- Microsoft.Windows.Capture.WindowsEffectCameraMediaSource
- Microsoft.ASG.Perception



Intel STC v2.3

- Collect Xperf log by STC tool
- Pre-request: install Windows ADK

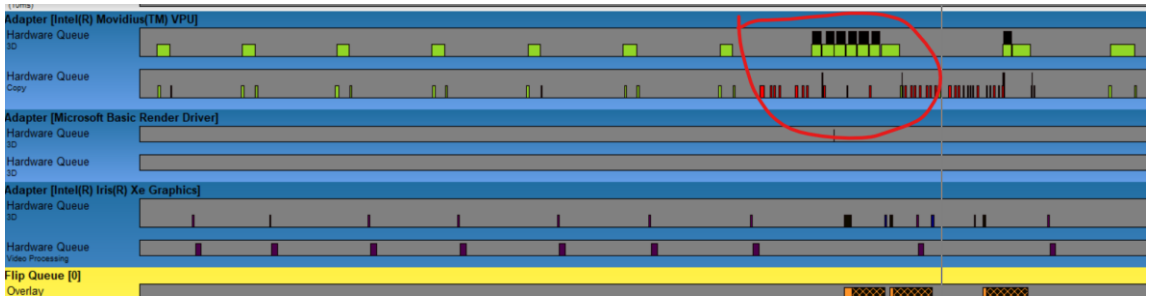




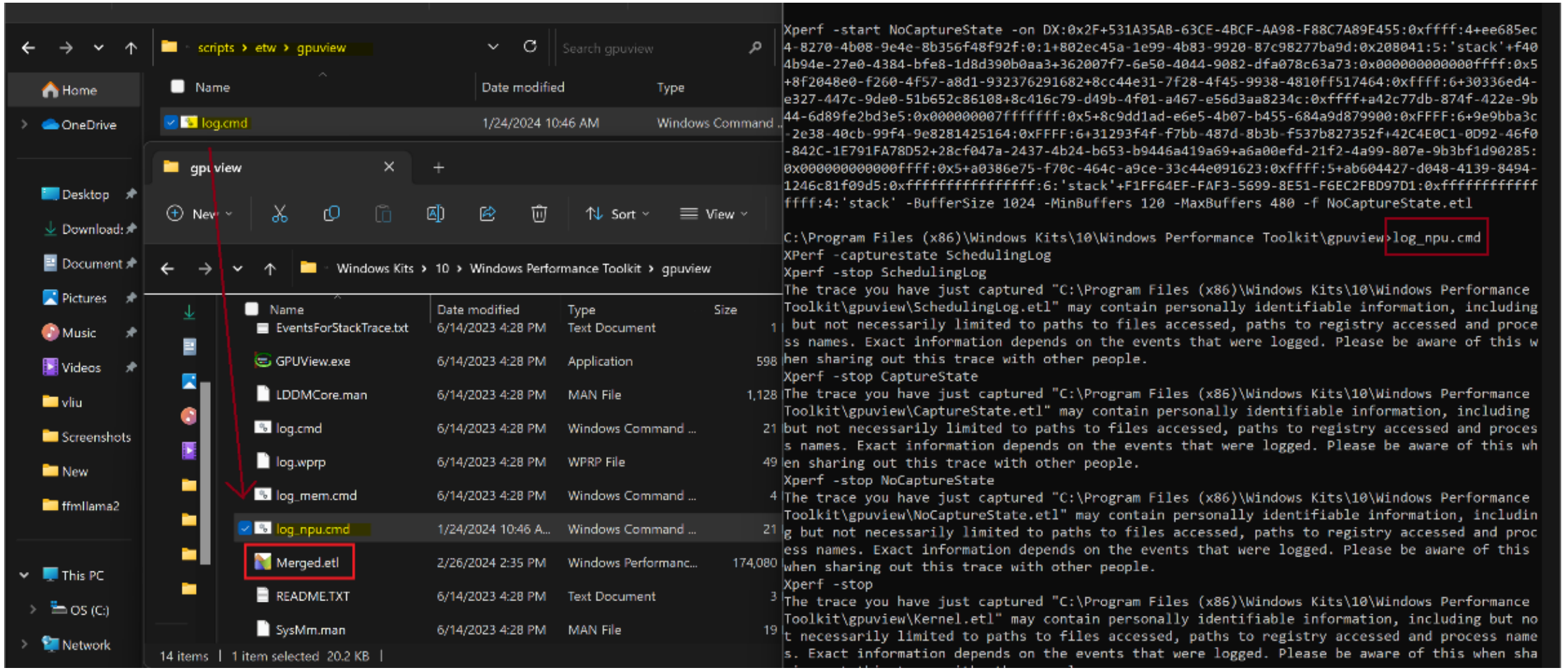
GPUView

GPUView is useful to debug performance issues. The adaptor NPU will record the inference event, click the block can get this inference details, including execution time, submission and complete time.

- Install Windows ADK
- In NPU driver release package, rename script/etw/gpuview/log.cmd to log_npu.cmd and copy it to Windows Performance Toolkit\gpuview folder.
- Execute cmd.exe as administrator and run log_npu.cmd.
- Reproduce the issue.
- When test complete, go back to the Admin cmd window and run log_npu.cmd again to generate the Merged.etl file



GPUView


scripts › etw › gpuview
Search gpuview
Name
Date modified
Type
log.cmd
1/24/2024 10:46 AM
Windows Command ...
Home
OneDrive
Desktop
Download:
Document
Pictures
Music
Videos
vliu
Screenshots
New
ffmllama2
This PC
OS (C:)
Network
Sort
View
Windows Kits › 10 › Windows Performance Toolkit › gpuview
Name
Date modified
Type
Size
EventsForStackTrace.txt
6/14/2023 4:28 PM
Text Document
1
GPUView.exe
6/14/2023 4:28 PM
Application
598
LDDMCore.man
6/14/2023 4:28 PM
MAN File
1,128
log.cmd
6/14/2023 4:28 PM
Windows Command ...
21
log.wprp
6/14/2023 4:28 PM
WPRP File
49
log_mem.cmd
6/14/2023 4:28 PM
Windows Command ...
4
log_npu.cmd
1/24/2024 10:46 A...
Windows Command ...
21
Merged.etl
2/26/2024 2:35 PM
Windows Performanc...
174,080
README.TXT
6/14/2023 4:28 PM
Text Document
3
SysMm.man
6/14/2023 4:28 PM
MAN File
19
14 items | 1 item selected 20.2 KB
Xperf -start NoCaptureState -on DX:0x2F+531A35AB-63CE-4BCF-AA98-F88C7A89E455:0xffff:4+ee685ec4-8270-4b08-9e4e-8b356f48f92f:0:1+802ec45a-1e99-4b83-9920-87c98277ba9d:0x208041:5:'stack'+f404b94e-27e0-4384-bfe8-1d8d390b0aa3+362007f7-6e50-4044-9082-dfa078c63a73:0x000000000000ffff:0x5+8f2048e0-f260-4f57-a8d1-932376291682+8cc44e31-7f28-4f45-9938-4810ff517464:0xffff:6+30336ed4-e327-447c-9de0-51b652c86108+8c416c79-d49b-4f01-a467-e56d3aa8234c:0xffff+a42c77db-874f-422e-9b44-6d89fe2bd3e5:0x000000007fffffff:0x5+8c9dd1ad-e6e5-4b07-b455-684a9d879900:0xFFFF:6+9e9bba3c-2e38-40cb-99f4-9e8281425164:0xFFFF:6+31293f4f-f7bb-487d-8b3b-f537b827352f+42C4E0C1-0D92-46f0-842C-1E791FA78D52+28cf047a-2437-4b24-b653-b9446a419a69+a6a00efd-21f2-4a99-807e-9b3bf1d90285:0x000000000000ffff:0x5+a0386e75-f70c-464c-a9ce-33c44e091623:0xffff:5+ab604427-d048-4139-8494-1246c81f09d5:0xffffffffffffffff:6:'stack'+F1FF64EF-FAF3-5699-8E51-F6EC2FBD97D1:0xffffffffffffffff:4:'stack' -BufferSize 1024 -MinBuffers 120 -MaxBuffers 480 -f NoCaptureState.etl
C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit\gpuview>log_npu.cmd
XPerf -capturestate SchedulingLog
Xperf -stop SchedulingLog
The trace you have just captured "C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit\gpuview\SchedulingLog.etl" may contain personally identifiable information, including but not necessarily limited to paths to files accessed, paths to registry accessed and process names. Exact information depends on the events that were logged. Please be aware of this when sharing out this trace with other people.
Xperf -stop CaptureState
The trace you have just captured "C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit\gpuview\CaptureState.etl" may contain personally identifiable information, including but not necessarily limited to paths to files accessed, paths to registry accessed and process names. Exact information depends on the events that were logged. Please be aware of this when sharing out this trace with other people.
Xperf -stop NoCaptureState
The trace you have just captured "C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit\gpuview\NoCaptureState.etl" may contain personally identifiable information, including but not necessarily limited to paths to files accessed, paths to registry accessed and process names. Exact information depends on the events that were logged. Please be aware of this when sharing out this trace with other people.
Xperf -stop
The trace you have just captured "C:\Program Files (x86)\Windows Kits\10\Windows Performance Toolkit\gpuview\Kernel.etl" may contain personally identifiable information, including but not necessarily limited to paths to files accessed, paths to registry accessed and process names. Exact information depends on the events that were logged. Please be aware of this when sha

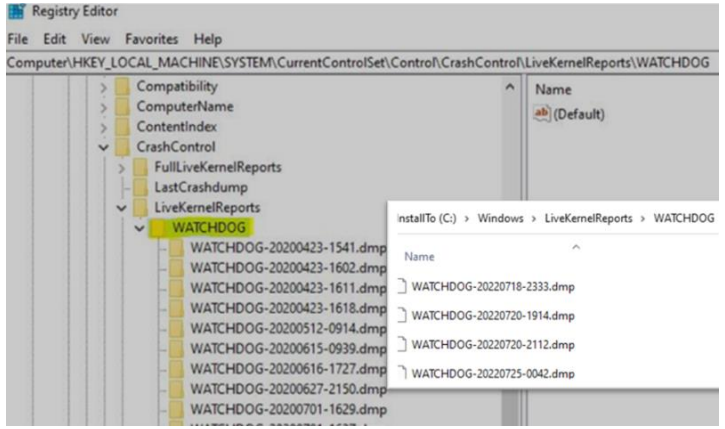
Windbg-Live kernel dump/BSOD memory dump

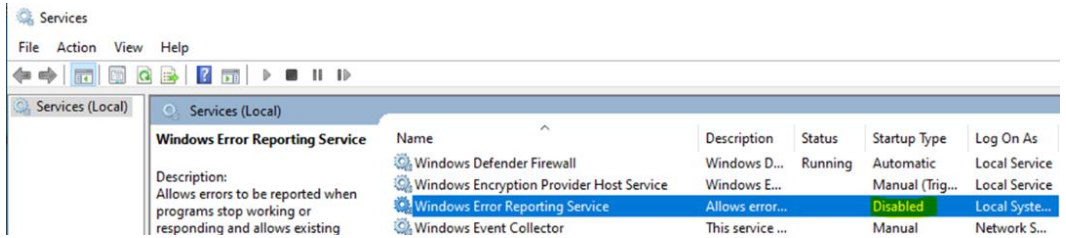
NPU driver also supports kernel live dump, we can use Windbg tool to check who causes the TDR.

The dump file is in C:\Windows\LiveKernelReports\WATCHDOG

If dump was not observed

- Make sure sub reg keys under Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\CrashControl\LiveKernelReports\WATCHDOG are cleaned up, if it reaches 10, you won't see OS report files dump
- Disable the Windows Error Reporting Service





Windbg-Live kernel dump/BSOD memory dump

```
Loading User Symbols
Mini Kernel Dump does not contain unloaded driver list
For analysis of this file, run !analyze -v
watchdog!WdpDbgCaptureTriageDump+0xb7:
fffff804`3356396d 488b4628        mov     rax,qword ptr [rsi+28h] ds:002b:ffffbd0f`9499d038=??????
9: kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

VIDEO_ENGINE_TIMEOUT_DETECTED (141)
One of the display engines failed to respond in timely fashion.
(This code can never be used for a real BugCheck.)
Arguments:
Arg1: ffffbd0f94d6c010, Optional pointer to internal TDR recovery context (TDR_RECOVERY_CONTEXT).
Arg2: fffff80457c03330, The pointer into responsible device driver module (e.g. owner tag).
Arg3: 0000000000000000, The secondary driver specific bucketing key.
Arg4: 000000000000309c, Optional internal context dependent data.
```

```
FILE_IN_CAB:  WATCHDOG-20230315-0332.dmp

DUMP_FILE_ATTRIBUTES: 0x18
  Kernel Generated Triage Dump
  Live Generated Dump

BUGCHECK_CODE:  141

BUGCHECK_P1: ffffbd0f94d6c010

BUGCHECK_P2: fffff80457c03330

BUGCHECK_P3: 0

BUGCHECK_P4: 309c

TAG_NOT_DEFINED_202b:  *** Unknown TAG in analysis list 202b


VIDEO_TDR_CONTEXT: dt dxgkrnl!_TDR_RECOVERY_CONTEXT ffffbd0f94d6c010
Symbol dxgkrnl!_TDR_RECOVERY_CONTEXT not found.

PROCESS_OBJECT: 000000000000309c

PROCESS_NAME:  System

STACK_TEXT:
ffffdc8d`6f9a7550 fffff804`33578f24     : ffffbd0f`8768b030 ffffbd0f`8768b030 ffffbd0f`9499d010 ffffbd0f`94d6c010 : watchdog!WdpDbgCaptureTriageDump+0xb7
ffffdc8d`6f9a75c0 fffff804`333ddb0d     : ffffbd0f`94d6c010 ffffdc8d`6f9a7780 00000000`00000000 fffff804`1baf54e9 : watchdog!WdDbgReportRecreate+0xd4
ffffdc8d`6f9a7620 fffff804`1ba2bf5b     : ffffbd0f`00000000 ffffbd0f`90b36bd0 ffffbd0f`94d6c010 ffffbd0f`90b3c000 : dxgkrnl!TdrUpdateDbgReport+0x11d
ffffdc8d`6f9a7680 fffff804`1baf5859     : ffffbd0f`90b3c000 00000000`00000000 ffffbd0f`90b36000 ffffbd0f`90b3c001 : dxgmms2!VidSchiResetEngine+0x70f
ffffdc8d`6f9a7830 fffff804`1bac512b     : ffffbd0f`90b36000 00000000`00000000 00000000`00000000 00000000`00000000 : dxgmms2!VidSchiResetEngines+0xb1
ffffdc8d`6f9a7880 fffff804`1ba97c9f     : 00000000`00000000 00000000`00000000 00000000`000029e6 00000000`00989680 : dxgmms2!VidSchiCheckHwProgress+0x2d44b
ffffdc8d`6f9a7900 fffff804`1b9fb689     : 00000000`00000000 ffffbd0f`90b36000 ffffdc8d`6f9a7a39 00000000`00000000 : dxgmms2!VidSchiWaitForSchedulerEvents+0x37f
ffffdc8d`6f9a79d0 fffff804`1baa3a95     : ffffbd0f`8acbba00 ffffbd0f`90b36000 ffffbd0f`8acbbaa0 ffffbd0f`961c6010 : dxgmms2!VidSchiScheduleCommandToRun+0x309
ffffdc8d`6f9a7aa0 fffff804`1baa3a0a     : 00000000`00000000 fffff804`1baa3940 ffffbd0f`90b36000 00000000`00000080 : dxgmms2!VidSchiRun_PriorityTable+0x35
ffffdc8d`6f9a7af0 fffff804`17c0f5b7     : ffffbd0f`90b39100 fffff804`00000001 ffffbd0f`90b36000 006fe47f`b19bbdff : dxgmms2!VidSchiWorkerThread+0xca
ffffdc8d`6f9a7b30 fffff804`17e2e364     : ffffe081`1d9a1180 ffffbd0f`90b39100 fffff804`17c0f560 97f29678`aa1603e0 : nt!PspSystemThreadStartup+0x57
ffffdc8d`6f9a7b80 00000000`00000000     : ffffdc8d`6f9a8000 ffffdc8d`6f9a1000 00000000`00000000 00000000`00000000 : nt!KiStartSystemThread+0x34


SYMBOL_NAME:  ivdkmd+3330

MODULE_NAME: ivdkmd

IMAGE_NAME:  ivdkmd.sys
```

Windbg-Live kernel dump/BSOD memory dump

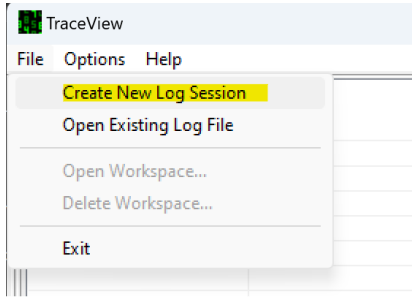
- If the system shows blackscreen or blue screen (BSOD), see if any memory dump created under C:\Windows\MEMORY.DMP.
- Using Windbg tool to check if NPU driver npu_kmd.sys or ivdkmd.sys in Windbg **!analyze –v**

```
2: kd> !analyze -v
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

VIDEO_TDR_FAILURE (116)
Attempt to reset the display driver and recover from timeout failed.
Arguments:
Arg1: ffff810fa50c6400, Optional pointer to internal TDR recovery context (TDR_RECOVERY_CONTEXT).
Arg2: fffff80583073000, The pointer into responsible device driver module (e.g. owner tag).
Arg3: 0000000000000000, Optional error code (NTSTATUS) of the last failed operation.
Arg4: 000000000000000d, Optional internal context dependent data.
```

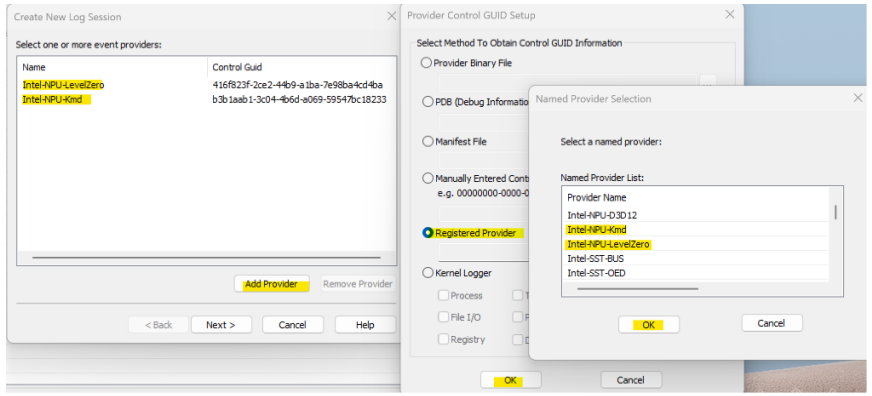
```
STACK_TEXT:
ffffaf81`00e776f8 fffff805`6e2a8e8e     : 00000000`00000116 ffff810f`a50c6400 fffff805`83073000 00000000`00000000 : nt!KeBugCheckEx
ffffaf81`00e77700 fffff805`6e2a851d     : fffff805`83073000 ffff810f`a50c6400 ffffaf81`00e77819 00000000`0000160d : dxgkrnl!TdrBugcheckOnTimeout+0xfe
ffffaf81`00e77740 fffff805`64fb5189     : 00000000`0000160d ffff810f`a3e6a000 01d81c05`d0dff0db fffff805`64f8546d : dxgkrnl!TdrIsRecoveryRequired+0x19d
ffffaf81`00e77770 fffff805`6503826c     : ffff810f`a3e6a000 00000000`00000001 ffff810f`a3e6a000 00000000`00000000 : dxgmms2!VidSchiReportHwHang+0x5e9
ffffaf81`00e77880 fffff805`65006bdb     : 00000000`00000000 00000000`00000000 00000000`00034580 00000000`00989680 : dxgmms2!VidSchiCheckHwProgress+0x3165c
ffffaf81`00e77900 fffff805`64f7b949     : 00000000`00000000 ffff810f`a3e6a000 ffffaf81`00e77a39 00000000`00000000 : dxgmms2!VidSchiWaitForSchedulerEvents+0x37b
ffffaf81`00e779d0 fffff805`6501e335     : ffff810f`a1c0b000 ffff810f`a3e6a000 ffff810f`a1c0b010 ffff810f`a1c11010 : dxgmms2!VidSchiScheduleCommandToRun+0x2f9
ffffaf81`00e77aa0 fffff805`6501e2ea     : ffff810f`a3e6a400 fffff805`6501e220 ffff810f`a3e6a000 ffff810f`a3e02000 : dxgmms2!VidSchiRun_PriorityTable+0x35
ffffaf81`00e77af0 fffff805`680478f5     : ffff810f`a3e020c0 fffff805`00000001 ffff810f`a3e6a000 006fe47f`b19bbdff : dxgmms2!VidSchiWorkerThread+0xca
ffffaf81`00e77b30 fffff805`68219bb4     : ffffdf81`14b40180 ffff810f`a3e020c0 fffff805`680478a0 48c03302`ebc00000 : nt!PspSystemThreadStartup+0x55
ffffaf81`00e77b80 00000000`00000000     : ffffaf81`00e78000 ffffaf81`00e71000 00000000`00000000 00000000`00000000 : nt!KiStartSystemThread+0x34


SYMBOL_NAME:  ivdkmd+3000

MODULE_NAME: ivdkmd

IMAGE_NAME:  ivdkmd.sys

IMAGE_VERSION:  29.21.10.257

STACK_COMMAND:  .cxr; .ecxr ; kb

FAILURE_BUCKET_ID:  0x116_IMAGE_ivdkmd.sys
```

Trace View

- Install Windows Driver Kit (WDK)
  - The traceview.exe can be found in C:\Program Files (x86)\Windows Kits\10\Tools\10.0.22621.0\x64\traceview.exe
- Right click traceview.exe, select “Run as administrator”. “File” -> “Create New Log Session”.

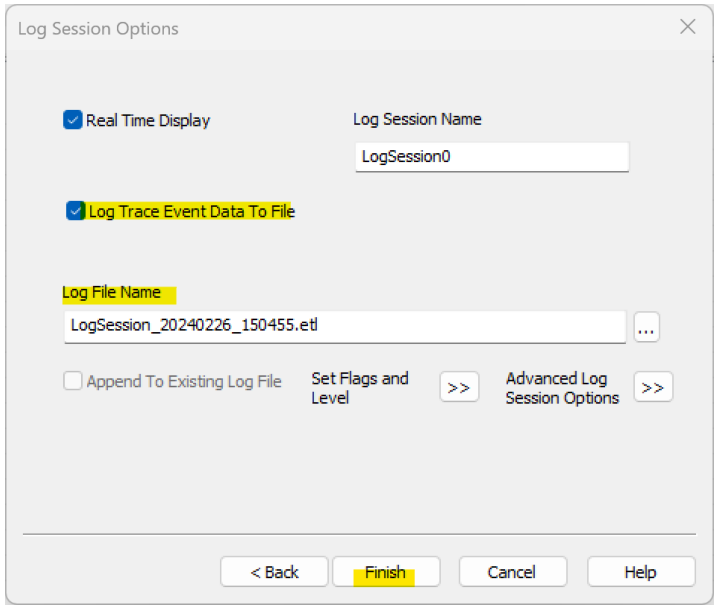


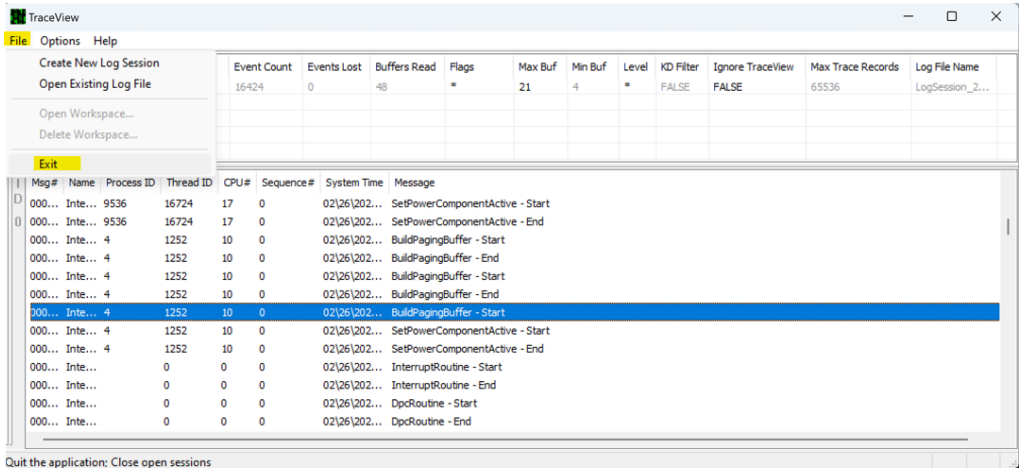
- In Registered Provider, add Intel-NPU-Kmd and Intel-NPU-LevelZero

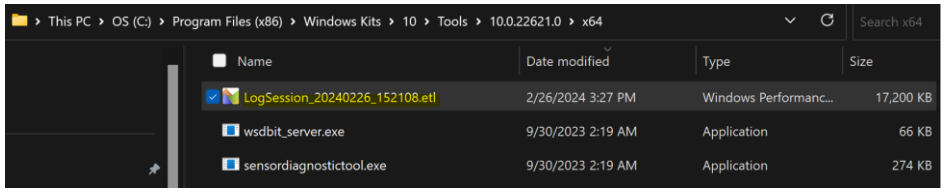


Trace View

- Click Next and check “Log Trace Event Data To File” and then “Finish” to start the log session.
- Reproduce the issue
- Selecting “File”> “Exit” to stop the log session







WPR (Windows Performance Recorder)

Next Gen AI Experiences - NPU ETL recording and analysis

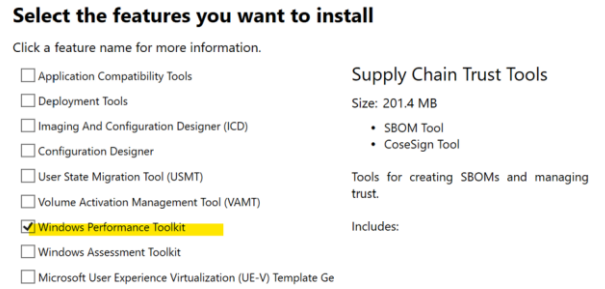
- wpr.exe -start NeuralProcessing -filemode
- repro issue
- wpr -stop trace.etl
- can be combined with other built-in profiles or your own tracing profile(.WPRP file)
  - ex: wpr -start NeuralProcessing -start CPU -start file.wprp!myprofile

```
C:\Windows\System32>wpr.exe -start NeuralProcessing -start CPU -start D:\Tool\WPR\aitrace.wprp

C:\Windows\System32>wpr.exe -stop trace_ai.etl
Press Ctrl+C to cancel the stop operation.
100%  [>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>]
The trace was successfully saved.
```

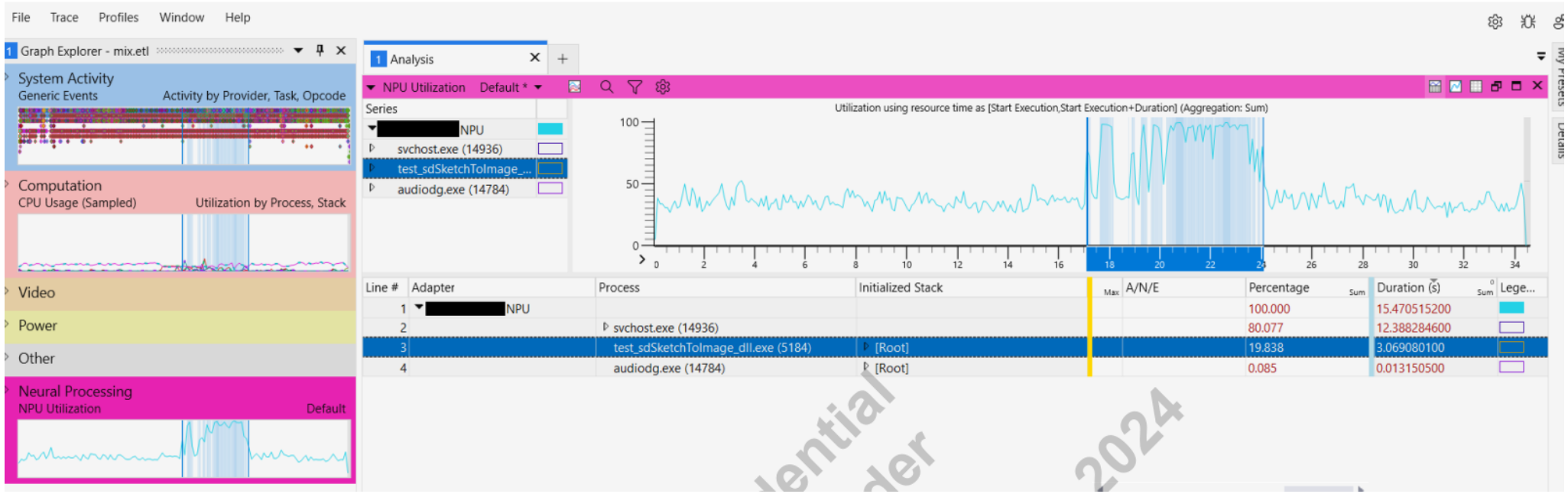
AITrace.wprp

[4/15/2024 Update] Version 1.1 of this package comes with a AITrace.wprp custom tracing profile that contains some ONNX/DirectML and other core AI framework ETW instrumentation. A few providers from inbox Windows AI experiences are also included (Paint, Live Caption, Recall, etc.). It can be used on top of the NeuralProcessing flag to obtain additional performance markers.

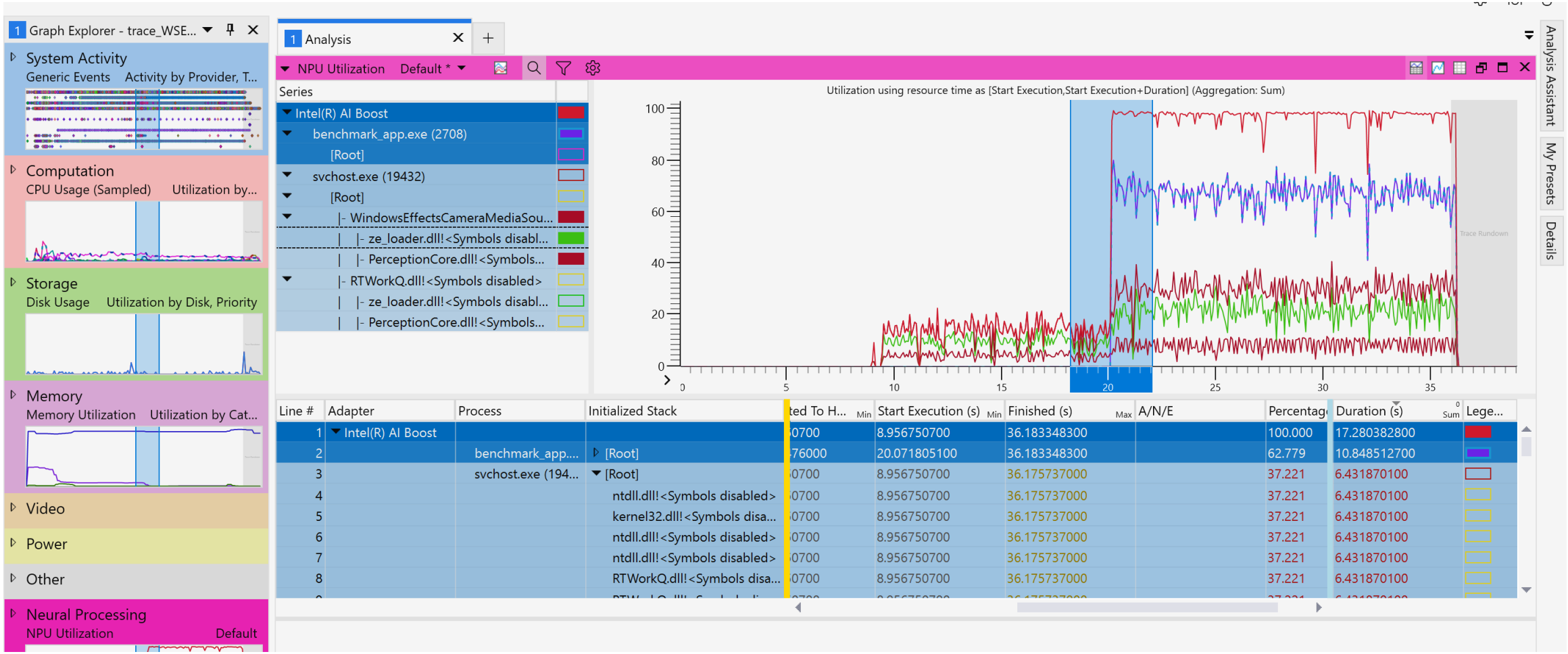
NPU ETL recording and analysis



- ADK 26063+ [Download the Windows ADK 10.1.26100.1 (May 2024)]
- It shows each process consuming the resource, the callstack dispatching the work to the NPU, and the amount of time spent consuming the resource(in absolute time and relative % based on the time interval focused on)

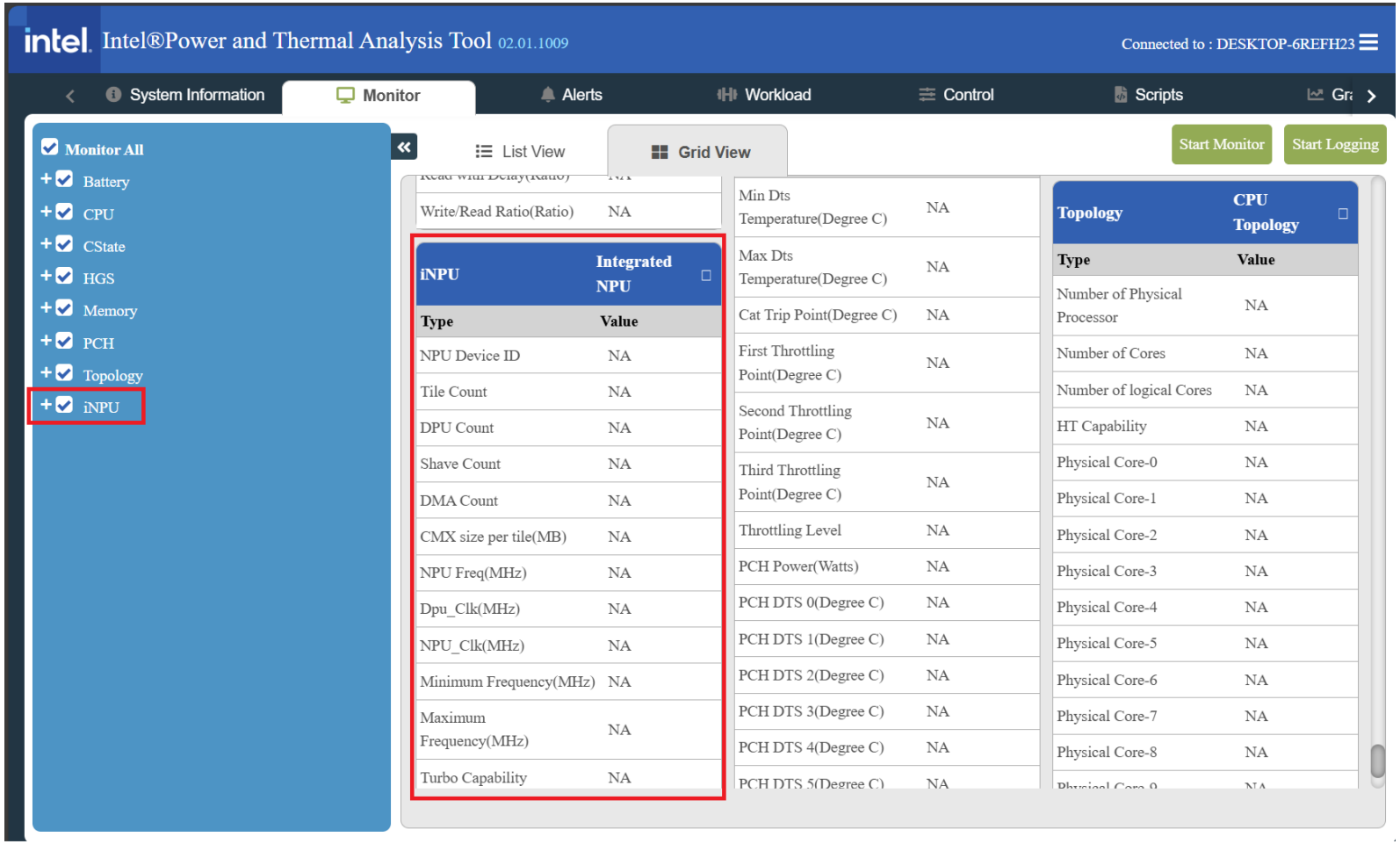


NPU ETL recording and analysis



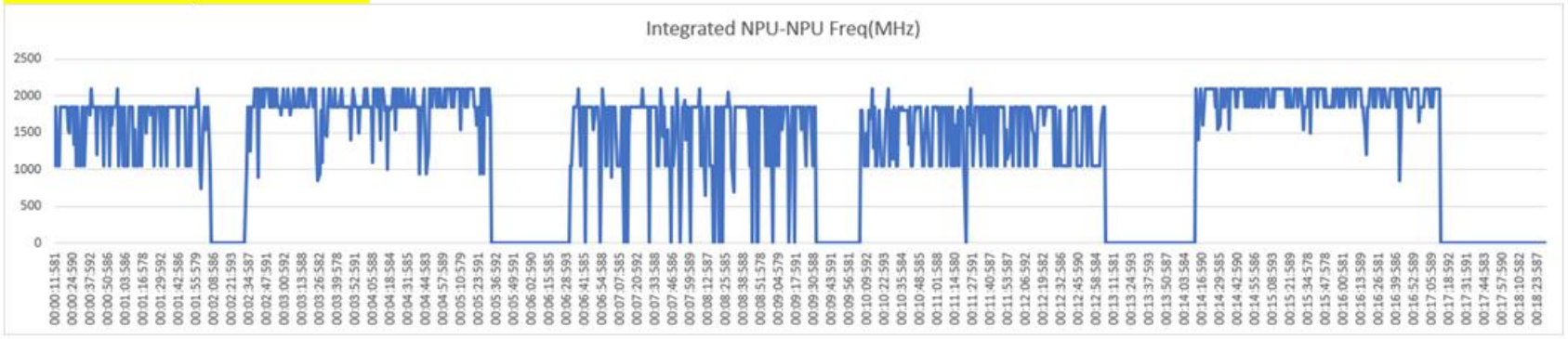
PTAT

For NPU performance issues, we can also get NPU frequency data by PTAT tool

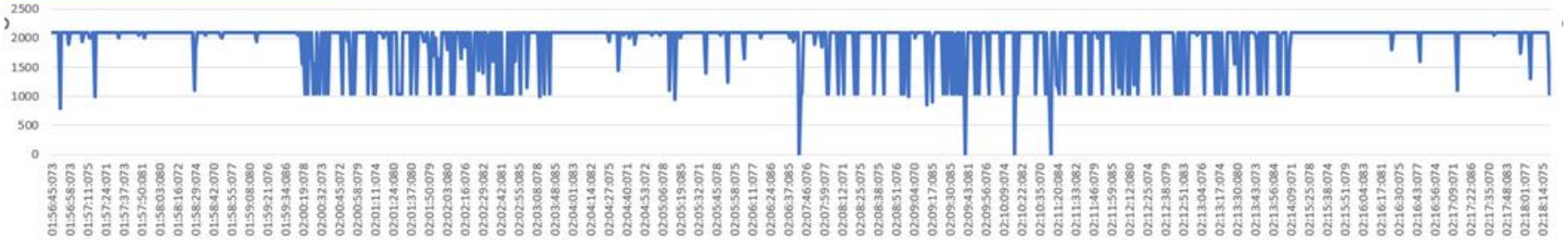


PTAT

HP Ernesto: Procyon Score is 461



HP Willie: Procyon Score is 511



Integrated NPU-NPU Freq(MHz)	Integrated NPU-Dpu_Clk(MHz)	Integrated NPU-NPU_Clk(MHz)
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
1450	966	475
950	633	475
950	633	475
950	633	475
1550	1033	475
950	633	475
1000	666	500
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
950	633	475
1050	700	525
950	633	475
950	633	475

Reference

ARL NPU Introduction and Features (video)

2023 2H TPnP Customer Tour - MTL-TnP-O025 - MTL Resnet50 BKM (video)

MTL-AI-D006MTL iVPU Validation and Debug Training (video)

Meteor Lake VPU MEP Enablement and Validation Testcases in OEM Platform

Debug NPU/MEP software stack debug guide


intel