



Moore Threads GPU Management Interface (mthreads-gmi)

User manual

Version: v1.8.1

2023-09-18

Contents

1 Supported platforms	1
2 Command reference	1
2.1 GPU commands	1
2.1.1 mthreads-gmi	1
2.1.2 mthreads-gmi [--list-gpus -L]	1
2.1.3 mthreads-gmi [--query -q]	2
2.1.4 mthreads-gmi [--help -h]	2
2.1.5 mthreads-gmi [--reset -r] [--id -i]	3
2.1.6 mthreads-gmi [--process-monitor -pm]	3
2.2 MPC commands	3
2.2.1 mthreads-gmi mpc [--enable] [--id -i]	3
2.2.2 mthreads-gmi mpc [--list-mpc-config -lmc] [--id -i]	3
2.2.3 mthreads-gmi mpc [--config-mpc -cm] [--id -i]	3
2.2.4 mthreads-gmi mpc [--list-mpc-instances -lmi]	4
2.2.5 mthreads-gmi mpc [--disable] [--id -i]	4
2.3 Topology commands	4
2.3.1 mthreads-gmi topo [--matrix -m]	4
2.3.2 mthreads-gmi topo [--p2pstatus -p2p]	4
2.3.3 mthreads-gmi topo [--nearest-gpus -n]	5
2.3.4 mthreads-gmi topo [--gpu-path -p]	5
2.4 MTLINK commands	6
2.4.1 mthreads-gmi mtlink [--status -s]	6
2.4.2 mthreads-gmi mtlink [--capabilities -c]	6
2.4.3 mthreads-gmi mtlink [--remotelinkinfo -r]	6
2.4.4 mthreads-gmi mtlink [--layout -la] [--id -i]	7
2.5 vGPU commands	7
2.5.1 mthreads-gmi vgpu [--support -s]	7
2.5.2 mthreads-gmi vgpu [--created -c]	7
2.5.3 mthreads-gmi vgpu [--encodersessions -es]	7
2.5.4 mthreads-gmi vgpu [--decodersessions -ds]	8
2.6 VPU commands	8
2.6.1 mthreads-gmi [--encodersessions -es]	8
2.6.2 mthreads-gmi [--decodersessions -ds]	9
3 Changelog	9
3.1 Changes of v1.8.1	9
3.2 Changes of v1.8.0	9
3.3 Changes of v1.7.0	10
3.4 Changes of v1.6.0	10
3.5 Changes of v1.5.0	10
3.6 Changes between v1.4.0 ~ v 1.4.2	10
3.7 Changes of v1.3.0	11
3.8 Changes of v1.2.0	11
3.9 Changes of v1.1.0	11
3.10 Changes between v0.1.0 ~ v1.0.0	11

The Moore Threads GPU Management Interface (mthreads-gmi) is a command line tool built on the GPU Management Center SDK. mthreads-gmi is designed to monitor and manage Moore Threads GPUs. mthreads-gmi is automatically installed when the GPU driver is installed.

1 Supported platforms

mthreads-gmi v1.8.1 supports the following platforms.

CPU architecture	Operating system
X86_64	Ubuntu 20.04, UOS, and CentOS
AArch64	Kylin 1041 and UOS 2107 SP1

2 Command reference

mthreads-gmi v1.8.1 supports the following commands.

2.1 GPU commands

2.1.1 mthreads-gmi

Display an overview of all physical GPUs and process-level information running on GPU. The latter is not supported temporarily in a virtual GPU environment.

[plus optional]

`-i, --id=ID`

Display data for a single specified GPU by selected id, identified by the count of GPUs.

`--debug=FILE`

Produces an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query information for the GPU with index 0.

```
mthreads-gmi -i 0
```

2.1.2 mthreads-gmi [--list-gpus | -L]

List each of the Moore Threads GPUs in the system, along with their UUIDs.

[plus optional]

`--debug=FILE`

Produces an encrypted log to a specified file, which is convenient for the developer to debug.

2.1.3 mthreads-gmi [--query | -q]

Display detailed information for all physical GPUs, including name, brand, mtbios version, pci, memory, utilization, temperature, fan, power, clock, encode stats, and decoder stats. Some devices and environments do not support all possible information. Any unsupported data is indicated by a "N/A" in the output.

[plus optional]

`-i, --id=ID`

Display data for a single specified GPU by selected id, identified by the count of GPUs.

`-f FILE, --filename=FILE`

Redirect query output to a specified file instead of the default stdout.

`-d TYPE, --display=TYPE`

Allow the caller to pass an explicit list of properties to query: MEMORY, UTILIZATION, TEMPERATURE, CLOCK, FAN, VIDEO, and POWER.

`-l SEC, --loop=SEC`

Continuously report query data at the specified interval, rather than the default of just once.

`--debug=FILE`

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query the MEMORY information of the GPU with index 0.

```
mthreads-gmi --query -i 0 -d MEMORY
```

Query attributes for all GPUs every 10 seconds and save the output to the log file.

```
mthreads-gmi --query -l 10 -f log
```

2.1.4 mthreads-gmi [--help | -h]

Print usage information and exit.

[plus optional]

`--debug=FILE`

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

2.1.5 mthreads-gmi [--reset | -r][--id | -i]

Facilitate the reset of GPU hardware and software state, eliminating the need for a machine reboot in such scenarios. GPU reset is not guaranteed to work in all cases. It is not recommended for production environments at this time.

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

2.1.6 mthreads-gmi [--process-monitor| -pm]

Monitor the processes active on GPUs installed within the system.

[plus optional]

`-I SEC, --loop=SEC`

Continuously report query data at the specified interval, rather than the default of just once.

2.2 MPC commands

2.2.1 mthreads-gmi mpc [-enable][--id | -i]

Enable MPC mode for the target device.

Example

Enable MPC mode for device 0.

```
mthreads-gmi mpc -enable -i 0
```

2.2.2 mthreads-gmi mpc [--list-mpc-config | -lmc] [--id | -i]

Query supported configurations and profiles.

Example

Query supported configurations and profiles for device 0.

```
mthreads-gmi mpc -lmc -i 0
```

2.2.3 mthreads-gmi mpc [--config-mpc | -cm] [--id | -i]

Configure MPC mode for the target device using the configuration of the specified ConfigID.

Example

Configure MPC mode for device 0 using configuration 15.

```
mthreads-gmi mpc -cm 15 -i 0
```

2.2.4 mthreads-gmi mpc [--list-mpc-instances | -lmi]

List all MPC instances that have been created.

2.2.5 mthreads-gmi mpc [-disable][--id|-i]

Disable MPC mode for the target device.

2.3 Topology commands

List topology information about the system GPUs, how they connect as well as qualified NICs capable of RDMA.

2.3.1 mthreads-gmi topo [--matrix | -m]

Display a matrix of available GPUs with the following legend:

X = Self

SYS = Topology path in the system

NODE = Topology path connected to the same NUMA node but possibly multiple host bridges

HPB = Topology path connected to the same host bridge

MPB = Topology path that does not need to traverse a host bridge

SPS = Topology path that only needs to traverse a single PCIe switch

INT = Topology path within the same card, e.g. devices on the same S2000 card

[plus optional]

```
--debug=FILE
```

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

2.3.2 mthreads-gmi topo [--p2pstatus | -p2p]

Display the p2p status between the GPUs of a given p2p capability:

r - p2p read capability

w - p2p write capability

[plus optional]

```
--debug=FILE
```

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query p2p read capability between GPUs.

```
mthreads-gmi topo -p2p r
```

2.3.3 mthreads-gmi topo [--nearest-gpus | -n]

Display the nearest GPUs for a given traversal path, used in conjunction with -i which must be a pair of device IDs.

0 = internal connection (e.g., s2000)

1 = a single PCIe switch

2 = multiple PCIe switches

3 = a PCIe host bridge

4 = an on-CPU interconnect link between PCIe host bridges

5 = an SMP interconnect link between NUMA nodes

[plus optional]

`--debug=FILE`

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query nearest GPUs connected to GPU 0 by multiple PCIe switches.

```
mthreads-gmi topo -n 2 -i 0
```

2.3.4 mthreads-gmi topo [--gpu-path | -p]

Display the most direct path traversal for a pair of GPUs, used in conjunction with -i which must be a pair of device IDs.

[plus optional]

`--debug=FILE`

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query the most direct path traversal between GPU 0 with GPU 1.

```
mthreads-gmi topo -p -i 0,1
```

2.4 MTLINK commands

2.4.1 mthreads-gmi mtlink [--status| -s]

Query mtlink state. Valid states: LINK UP and LINK DOWN.

[plus optional]

`-i, --id=ID`

Display data for a single specified GPU.

Example

Query mtlink state for all GPUs.

```
mthreads-gmi mtlink -s
```

2.4.2 mthreads-gmi mtlink [--capabilities| -c]

Query the capability status for all mtlinks of a specified GPU.

[plus optional]

`-i, --id=ID`

Display data for a single specified GPU.

Example

Query the capability status for all GPU mtlinks.

```
mthreads-gmi mtlink -c
```

2.4.3 mthreads-gmi mtlink [--remotelinkinfo| -r]

Query the remote device directly connected via a specified mtlink of the specified GPU.

[plus optional]

`-i, --id=ID`

Display data for a single specified GPU.

`-l, --link=ID`

Display data for a single specified mtlink.

Example

Query the remote device connected via mtlink 1 of GPU 0.

```
mthreads-gmi mtlink -r -i 0 -l 1
```

2.4.4 mthreads-gmi mtlink [--layout| -la] [--id | -i]

Query the shortest mtlink paths that connect two GPUs.

Example

Query the shortest mtlink paths that connect GPU 0 and GPU 1.

```
mthreads-gmi mtlink -la -i 0,1
```

2.5 vGPU commands

2.5.1 mthreads-gmi vgpu [--support | -s]

Display vGPU types supported on each device.

[plus optional]

```
--debug=FILE
```

Produces an encrypted log to a specified file, which is convenient for the developer to debug.

2.5.2 mthreads-gmi vgpu [--created | -c]

Display vGPUs created on each device.

[plus optional]

```
--debug=FILE
```

Produce an encrypted debug log.

2.5.3 mthreads-gmi vgpu [--encodersessions | -es]

Display active encoder session metrics at the default interval (5 seconds).

[plus optional]

```
-i, --id=ID
```

Display data for a single specified GPU.

```
-l SEC, --loop=SEC
```

Probe until Ctrl+C at a specified interval (seconds).

```
--debug=FILE
```

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query active encoder session metrics in GPU 0 every second.

```
threads-gmi vgpu -es -l 1 -i 0
```

2.5.4 threads-gmi vgpu [--decodersessions | -ds]

Display active decoder session metrics at the default interval (5 seconds).

[plus optional]

`-i, --id=ID`

Display data for a single specified GPU.

`-l SEC, --loop=SEC`

Probe until Ctrl+C at a specified interval (seconds).

`--debug=FILE`

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query active decoder session metrics in GPU 0 every second.

```
threads-gmi vgpu -ds 0 -l 1
```

2.6 VPU commands

2.6.1 threads-gmi [--encodersessions | -es]

Display active encoder session metrics at the default interval (5 seconds).

[plus optional]

`-i, --id=ID`

Display data for a single specified GPU.

`-l SEC, --loop=SEC`

Probe until Ctrl+C at a specified interval (seconds).

`--debug=FILE`

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query active encoder session metrics for all GPUs every 2 seconds.

```
threads-gmi -es -l 2
```

2.6.2 threads-gmi [--decodersessions | -ds]

Display active decoder session metrics at the default interval.

[plus optional]

`-i, --id=ID`

Display data for a single specified GPU.

`-l SEC, --loop=SEC`

Probe until Ctrl+C at a specified interval (seconds).

`--debug=FILE`

Produce an encrypted log to a specified file, which is convenient for the developer to debug.

Example

Query active decoder session metrics for GPU 2 every 2 seconds.

```
threads-gmi -ds -l 2 -i 2
```

3 Changelog

3.1 Changes of v1.8.1

- Resolved an issue where the codec resolution value obtained may occasionally be incorrect.

3.2 Changes of v1.8.0

- Support threads-gmi mlink [-s|--status] to query mlink state.
- Support threads-gmi mlink [-c|--capabilities] to query capability status of mlinks.
- Support threads-gmi mlink [-r|--remotelinkinfo][-l|--link][-i|--id] to query the remote device connected via a specified mlink of a specified GPU.
- Support threads-gmi mlink [-la|--layout][-i|--id] to query the shortest mlink paths that connect two GPUs.
- Support threads-gmi [-pm|--process-monitor] to monitor the processes active on GPUs installed within the system.
- Change threads-gmi topo [--nearest_gpus|-n] to threads-gmi topo [--nearest-gpus|-n].
- Change threads-gmi topo [--gpu_path|-p] to threads-gmi topo [--gpu-path|-p]

3.3 Changes of v1.7.0

- Support `mtthreads-gmi [-r|--reset][-i|--id]` to facilitate the reset of GPU hardware and software state, eliminating the need for a machine reboot in such scenarios.
- Updated the debug option from `[-debug FILE]` to `[--debug=FILE]`.

3.4 Changes of v1.6.0

- Support `mtthreads-gmi mpc [-enable] [-i|--id]` to enable MPC mode for the target device.
- Support `mtthreads-gmi mpc [-lmc|--list-mpc-config][-i|--id]` to query supported MPC configurations and MPC instance profiles.
- Support `mtthreads-gmi mpc [-cm|--config-mpc][-i|--id]` to configure MPC mode for the target device using the specified configuration.
- Support `mtthreads-gmi mpc [-lmi|--list-mpc-instances]` to list all created MPC instances.
- Support `mtthreads-gmi mpc [-disable][-i|--id]` to disable MPC mode for the target device.
- Support querying device types and MPC capabilities.
- Display device information in the following style:
 - Sort physical devices (MPC parents) by index in an ascending order.
 - Sort MPC instances derived from the same MPC parent by index in an ascending order.
 - Indent MPC instances under their MPC parent.

3.5 Changes of v1.5.0

- Support `mtthreads-gmi [-es|--encodersessions][-i|--id][-l|--loop]` to query encoder session metric.
- Support `mtthreads-gmi [-ds|--decodersessions][-i|--id][-l|--loop]` to query decoder session metric.
- Support `mtthreads-gmi vgpu [-es|--encodersessions][-i|--id][-l|--loop]` to query encoder session metric in vdi environment.
- Support `mtthreads-gmi vgpu [-ds|--decodersessions][-i|--id][-l|--loop]` to query decoder session metric in vdi environment.
- Display vpu encoder and decoder utilization separately.
- Change `mtthreads-gmi -vgpu [-s|--support]` to `mtthreads-gmi vgpu [-s|--support]`.
- Change `mtthreads-gmi -vgpu [-c|--created]` to `mtthreads-gmi vgpu [-c|--created]`.
- Change `mtthreads-gmi -topo` to `mtthreads-gmi topo`.

3.6 Changes between v1.4.0 ~ v 1.4.2

- Added new command `mtthreads-gmi [-d file | -debug FILE]` : Log encrypted debug information to a specified file.
- Added new command `mtthreads-gmi -topo` : Display help information about topological command.
- Added new command `mtthreads-gmi -topo [-m | --matrix]` : Display the devices communication matrix for the system.
- Added new command `mtthreads-gmi -topo [-p2p | --p2pstatus]` : Displays the p2p status between the GPUs of read or write p2p capability.
- Added new command `mtthreads-gmi -topo [-n | --nearest_gpus] [-i | --id]` : Display the nearest GPUs for a given traversal path.
- Added new command `mtthreads-gmi -topo [-p | --gpu_path] [-i | --id]` : Display the most direct path traversal for a pair of GPUs.
- `mtthreads-gmi -q` can query power usage, which is supported for QY card.
- `mtthreads-gmi -vgpu -s` can query max vgpu instance.

3.7 Changes of v1.3.0

- mthreads-gmi supports all commands in mthreads-gmi v1.3 user manual on QY1.

3.8 Changes of v1.2.0

- Support to query fan nums and fan speed.
- Support to query properties of video codec in virtual gpu, which including utilization, clock, max clock, active sessions and average FPS of video codec.

3.9 Changes of v1.1.0

- Support to query pid, process name and gpu memory usage of process working on mt-gpu.
- Support to query properties of video codec, which including utilization, clock, max clock, active sessions and average FPS of video codec.
- Support to query vGPU types supported on each device.
- Support to query vGPUs created on each device.

3.10 Changes between v0.1.0 ~ v1.0.0

- Added mthreads-gmi [-i,--id] command-line to display an overview of all GPUs.
- Added mthreads-gmi [-L,--list-gpus] command-line to list all GPUs and their corresponding UUIDs.
- Added mthreads-gmi [-q,--query][-i,--id][-d,--display][-l,--loop][-f,--filename] command-line to query GPU information (clock, utilization, pci, memory, fan, power, mtbios, and so on).
- Added mthreads-gmi [-h,--help] command-line to display usage information.

Please visit the official website of Moore Threads at

<https://www.mthreads.com> for more information.

Scan the QR code to follow the WeChat official account.



Our Website



Our WeChat