



曦云®系列通用计算 GPU

mx-smi 使用手册

CSOG-24049-020-F3_V01

2024-09-06

沐曦专有和三级保密信息

本文档受 NDA 管控

更新记录

版本	日期	更新说明
V01	2024-09-06	首次发布

目录

1. 概述	1
2. 安装及使用方法	2
2.1 安装包安装与卸载	2
2.1.1 安装	2
2.1.2 卸载	2
2.2 在容器中使用	2
2.3 使用方法	2
3. 命令介绍	3
3.1 通用选项	3
3.2 归纳选项	3
3.3 查询选项	3
3.4 控制选项	5
3.5 RAS 选项	6
3.6 查询拓扑关系	6
3.7 VM 虚拟化管理	6
3.8 光模块信息（当前仅适用于 C500X）	7
3.9 轮询设备状态信息	7
4. MXSML 使用说明	8
4.1 简介	8
4.2 安装	8
4.3 编程指南	8
5. Pymxsmi 使用说明	10
5.1 简介	10
5.2 安装	10
5.3 编程指南	10
6. 附录	11
6.1 术语/缩略语	11

表目录

表 4-1 MXSML 安装文件列表 8

飞腾信息 MetaX Confidential
2024-11-18 15:00:00

1. 概述

mx-smi (MetaX System Management Interface) 是用于管理曦云®系列 GPU 的命令行工具，可以实时查询 GPU 的信息，包括版本，温度，使用率，功耗，时钟频率等，并具有启用和禁用 GPU 配置选项，执行固件升降级等功能。

2. 安装及使用方法

2.1 安装包安装与卸载

2.1.1 安装

曦云系列 GPU 驱动包安装过程中会默认安装 mx-smi 工具。驱动包安装完成后，mx-smi 工具放置在 /opt/mxdriver/bin/目录下，并会在 /usr/bin 目录下创建软链接。

2.1.2 卸载

曦云系列 GPU 的驱动卸载后，mx-smi 工具会自动卸载。驱动卸载参见《曦云®系列通用计算 GPU 用户指南》。

2.2 在容器中使用

若需在容器中使用 mx-smi，可在运行容器时传入以下参数：

```
docker run -v --device=/dev/dri -v /opt/mxdriver/bin/mx-smi:/opt/mxdriver/bin/mx-smi <image ID>
```

其中 --device=/dev/dri 将主机上的沐曦设备挂载到容器，-v /opt/mxdriver/bin/mx-smi:/opt/mxdriver/bin/mx-smi 将 mx-smi 工具挂载到容器。

以上方式仅可在容器中使用 /opt/mxdriver/bin/mx-smi 执行一些查询类的命令，如需执行设置类的命令，需在启动容器时添加 --privileged=true。

2.3 使用方法

mx-smi 是一个 Linux 命令行工具，其调用遵守如下格式。

```
mx-smi [选项 1] [参数 1]] [参数 2] ...
```

3. 命令介绍

mx-smi 命令说明，可以通过在工作环境中执行 `man mx-smi` 获取。

3.1 通用选项

```
-h, --help
```

打印使用帮助信息。

```
-v, --version
```

打印 mx-smi 的版本信息。

3.2 归纳选项

```
./mx-smi （直接回车）
```

mx-smi 命令不添加任何选项和参数，默认归纳并显示板卡的主要信息，显示信息包括功耗，内存，温度，版本，使用率，进程等。

3.3 查询选项

```
--show-temperature
```

显示板卡上温度传感器读取的温度，单位是摄氏度。

```
--show-version
```

显示 BIOS，驱动和 Firmware 的版本信息。

```
--show-hbm-bandwidth
```

显示显存的动态带宽信息。

```
--show-pcie-bandwidth
```

显示 PCIe 动态带宽信息。

```
--show-usage
```

显示 GPU、VPU 的使用率。使用率反应过去一段采样周期内硬件资源的使用情况。

```
--show-memory
```

显示存储资源的使用情况，包括板卡显存和板卡可以访问的系统内存。

```
--show-board-power
```

显示板卡当前的电压，电流和功耗信息。

```
--show-pmbus-power
```

显示芯片内的电压，电流和功耗信息。

```
--show-eeeprom
```

显示 EEPROM 记录的信息，包括板卡版本，板卡序列号等。

```
--show-clocks all
```

显示板卡的时钟信息。

```
--show-dpm cur
```

显示当前的 performance level 设置。

```
--show-dpm all
```

显示支持的 performance level 及其对应的时钟和电压信息，为设置 performance level 做参考。

```
--show-dpm-max
```

显示支持的 performance level 的最高级别。

```
--show-pcie
```

显示 PCIe 当前及最大带宽和速率。

```
--show-sn
```

显示板卡序列号。

```
--show-process
```

显示当前运行的进程信息。

```
--show-power-mode
```

显示板卡的电源模式，未用 `-i` 指定板卡时，默认显示所有板卡的电源模式信息。

```
-i ID, --index ID
```

显示指定板卡的信息。如果没有指定，默认显示全部板卡的信息。ID 是从 0 开始的自然数，可以通过 `-L`，`--list` 获取板卡的 ID 信息。可指定一个或多个 ID，多个 ID 用 “,” 分隔，如 “0,1,5”，“0-2”，“0-4,6”，“all” 等。

```
-o, fileName, --output fileName
```

可以将命令输出写入 csv 格式的文件中，支持显示的命令有：

- `--show-temperature`

- --show-board-power
- --show-version
- --show-usage
- --show-memory
- --show-pcie
- --show-dpm cur
- --show-pmbus-power
- --show-vpu

```
-l ms, --loop ms
```

以特定的间隔周期持续显示板卡信息，单位是毫秒。执行 Ctrl+C 可退出持续查询。

示例：

以 500ms 周期持续显示所有卡的温度，功耗，使用率信息：

```
mx-smi --show-temperature --show-board-power --show-usage -l 500
```

以 1s 周期将 GPU0 - 3 的温度，功耗，使用率信息写文件：

```
mx-smi --show-temperature --show-board-power --show-usage -i 0-3 -l 1000 -o sample.csv
```

3.4 控制选项

```
-u, --vbios-upgrade <vbios bin file>
```

默认对所有板卡进行 BIOS 固件升级。需要 Root 权限。在执行完固件升级命令后，需要对系统进行 reboot 或者 warm reset 以使新固件生效。

```
-r, --reset
```

对指定的卡执行 warm reset。必须使用 -i ID, --index ID 选项以指定一个特定的板卡。需要 Root 权限。如果 warm reset 失败（设备无法恢复），需要重启服务器，对同一组使用 MetaXLink 连接的卡，只需做一次 warm reset，详情参见《曦云®系列通用计算 GPU Warm Reset 使用指南》。

说明

warm reset 功能目前仅在 H3C UniServer R5300 G6 服务器型号中经过验证，在未经验证的机型中使用 warm reset 功能可能会导致功能失效等不可预期的结果。

```
--set-power-mode <mode>
```

用 -i 指定板卡进行电源模式设置，0 为 Normal，1 为 High。

```
--set-dpm-max <IP, DpmLevel>
```

对指定的 IP 设置 max performance level。必须使用 -i ID, --index ID 选项以指定一个特定的板卡。

需要 Root 权限。

例如，执行以下命令，将板卡 1 的 xcore 模块的 max performance level 有效配置设置成 level7:

```
mx-smi -set-dpm-max xcore,7 -i 1
```

3.5 RAS 选项

mx-smi ras 子命令用于显示与芯片可靠性，可用性，安全性相关的信息。

可显示 IP 的错误统计。例如，执行以下命令，显示板卡 0 相关的 corrected error 和 uncorrected error 统计数目：

```
mx-smi ras --show-count -i 0
```

可显示 IP 的错误状态。例如，执行以下命令，显示板卡 0 相关的错误状态：

```
mx-smi ras --show-status -i 0
```

3.6 查询拓扑关系

mx-smi topo 子命令用于查询主机上曦云 GPU 设备之间的拓扑关系。

```
-m, --matrix
```

显示曦云 GPU 设备之间的连接方式，以及 CPU 亲和性、NUMA node 亲和性等信息。

```
-d, --distance
```

显示曦云 GPU 设备之间连接的相对距离信息。

3.7 VM 虚拟化管理

mx-smi vm 子命令用于 GPU 卡虚拟化管理。注意：此命令仅支持在物理机中操作。

```
--enable-vf <1|2|4|8>
```

开启虚拟化，一张 GPU 卡可支持虚拟化的个数为 1，2，4，8。

```
--disable-vf
```

关闭虚拟化功能。

```
--show-vf
```

显示虚拟化后的 GPU 信息。

3.8 光模块信息（当前仅适用于 C500X）

mx-smi om 子命令用于显示光模块信息。

```
--show-status
```

显示光模块电压，温度，模块状态，通道状态及接收状态信息。

```
--show-info
```

显示光模块固件信息。

```
--show-rx-status
```

显示光模块接收的详细信息。

3.9 轮询设备状态信息

mx-smi dmon 子命令用于轮询显示设备状态信息，包括温度，功耗，VPUE，VPUD 和 GPU 使用率，板卡显存及内存；时间间隔默认为 1000ms。执行 Ctrl+C 可退出轮询。

例如，执行以下命令，可轮询 100 次板卡 0 的温度及功耗信息：

```
mx-smi dmon --show-temperature --show-board-power -i 0 -c 100
```

4. MXSML 使用说明

4.1 简介

MXSML (MetaX System Management Library) 是一套 C API 程序开发库。应用程序使用这套开发库，可以方便的对沐曦的设备进行状态查询和管理。MXSML 提供的 API 可以被使用 C/C++/PYTHON/GO 等语言开发的程序使用。

4.2 安装

MXSML 动态链接库是随 Driver 软件包一起发布，同时在 MXMACA SDK 包中会包含头文件，stub 库，示例代码，Python bindings 等文件。

Driver 软件包的安装参见《曦云®系列通用计算 GPU 用户指南》中“安装驱动”章节。驱动包安装后，MXSML 动态链接库位于 `/opt/mxdriver/lib/libmxsml.so`。

MXMACA SDK 包的安装参见《曦云®系列通用计算 GPU 快速上手指南》。下表是安装的文件列表：

表 4-1 MXSML 安装文件列表

文件	说明
<code>/opt/maca/include/mxsml/MxSml.h</code>	MXSML 头文件
<code>/opt/maca/include/mxsml/MxSmlExtension.h</code>	MXSML 扩展头文件
<code>/opt/maca/include/mxsml/mxsmlBindings.py</code>	MXSML Python bindings
<code>/opt/maca/lib/libmxsml.so</code>	MXSML stub 库
<code>/opt/maca/share/mxsml/pymxsml-xxx.whl</code>	Pymxsml 安装包
<code>/opt/maca/share/mxsml/mxsmlDemo.c</code>	C 示例程序
<code>/opt/maca/share/mxsml/mxsml_demo.py</code>	Python 示例程序
<code>/opt/maca/share/mxsml/mxsmlDemo.go</code>	Go 示例程序

4.3 编程指南

用 C 语言调用 MXSML API 读取所有卡的信息，示例如下：

```
#include <stdio.h>
#include "MxSml.h"
```

```
int main()
{
    // 调用任何 MXSML API 前都需要先调用 mxSmlInit 完成初始化
    mxSmlReturn_t ret = mxSmlInit();
    if(ret != MXSML_Success)
    {
        // 可通过 mxSmlGetErrorString 将错误号转换为错误信息
        printf("mxSmlInit failed: %u, %s\n", ret, mxSmlGetErrorString(ret));
        return 1;
    }
    // 获取可用设备总数, 设备 ID 从 0 开始依次编号: 0,1,2...deviceNum-1
    int deviceNum = mxSmlGetDeviceCount();
    for(int i = 0; i < deviceNum; i++)
    {
        mxSmlDeviceInfo_t deviceInfo;
        if(mxSmlGetDeviceInfo(i, &deviceInfo) == MXSML_Success)
        {
            printf("device id %d, bdf id: %s, type: %s\n", i, deviceInfo.bdfId,
deviceInfo.deviceName);
        }
    }

    return 0;
}
```

使用以下命令完成编译:

```
gcc example.c -I/opt/maca/include/mxsml -L/opt/maca/lib -lmxsml -o example
```

运行 example:

```
export LD_LIBRARY_PATH=/opt/mxdriver/lib:$ LD_LIBRARY_PATH
./example
```

5. Pymxsmi 使用说明

5.1 简介

Pymxsmi 是基于 MXSML 封装的一套 Python API 程序开发库。使用这套开发库，可以方便地对沐曦 GPU 设备进行状态查询和管理。

5.2 安装

操作步骤

1. Pymxsmi 依赖 MXSML 动态链接库，确保驱动包已安装，相关信息参见 4.2 安装。
2. Pymxsmi 以 wheel 包形式随 MXMACA SDK 包一起发布。MXMACA SDK 包安装后，Pymxsmi wheel 包位于 `/opt/maca/share/mxsml/pymxsmi-*.whl`。
3. 确保系统已安装 Python 3，使用以下命令安装 Pymxsmi wheel 包。

```
pip3 install /opt/maca/share/mxsml/pymxsmi-*.whl
```

5.3 编程指南

用 Python 语言导入 Pymxsmi 模块读取所有卡的信息，示例如下：

```
from pymxsmi import *

# 调用任何 Pymxsmi API 前都需要先调用 mxSmlInit 完成初始化
# 接口调用失败，会产生对应的 python 异常
mxSmlInit()

# 获取可用设备总数，设备 ID 从 0 开始依次编号：0,1,2...deviceNum-1
device_num = mxSmlGetDeviceCount()
for i in range(device_num):
    device_info = mxSmlGetDeviceInfo(i)
    print("device id", i, ", bdf id:", device_info.bdfId, ", type:",
          device_info.deviceName)
```

运行 example：

```
python3 example.py
```

6. 附录

6.1 术语/缩略语

术语/缩略语	全称	描述
RAS	Reliability, Availability, Serviceability	
VM	Virtual Management	虚拟化管理

声明

版权所有 ©2024 沐曦集成电路（上海）有限公司。保留所有权利。

本文档中呈现的信息属于沐曦集成电路（上海）有限公司和/或其附属公司（以下统称为“沐曦”），非经沐曦事先书面许可，任何实体或个人均不得获得本文档的副本，且无权以任何方式处理本文档，包括但不限于使用、复制、修改、合并、出版、发行、销售或传播本文档的部分或全部。

本文档内容仅供参考，不提供任何形式的、明示或暗示的保证，包括但不限于对适销性、适用于任何目的和/或不侵权的保证。在任何情况下，沐曦均不对因本文档引起的、由本文档造成的、或与之相关的任何索赔、损害或其他责任负责。

沐曦保留自行决定随时更改、修改、添加或删除本文档的部分或全部的权利。沐曦保留最终解释权。

沐曦、MetaX 和其他沐曦图标是沐曦的商标。本文档中提及的所有其他商标和商品名称均为其各自所有者的财产。