



# 曦云®系列通用计算 GPU

## mx-exporter 使用手册

CSOG-24045-020-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. 工具部署.....              | 2  |
| 2.1 Wheel 包 .....         | 2  |
| 2.1.1 安装 .....            | 2  |
| 2.1.2 使用方法 .....          | 2  |
| 2.1.3 卸载 .....            | 3  |
| 2.2 mx-exporter 镜像.....   | 4  |
| 2.2.1 加载镜像 .....          | 4  |
| 2.2.2 使用方法 .....          | 4  |
| 2.2.3 删除容器 .....          | 6  |
| 3. 配置文件使用说明 .....         | 7  |
| 3.1 默认配置文件.....           | 7  |
| 3.2 修改配置文件.....           | 7  |
| 3.2.1 自定义指标集 .....        | 7  |
| 3.2.2 修改指标名称、指标描述及标签..... | 8  |
| 4. 指标及标签说明 .....          | 9  |
| 4.1 指标说明.....             | 9  |
| 4.2 标签说明.....             | 10 |
| 4.3 指标输出示例.....           | 10 |
| 5. 附录.....                | 15 |
| 5.1 术语/缩略语 .....          | 15 |

## 表目录

|                                           |    |
|-------------------------------------------|----|
| 表 2-1 mx-exporter 参数说明（wheel 包安装） .....   | 2  |
| 表 2-2 mx-exporter 参数说明（镜像部署） .....        | 4  |
| 表 2-3 Docker 参数说明.....                    | 5  |
| 表 3-1 默认配置文件表头说明.....                     | 7  |
| 表 3-2 默认配置文件中 chip_hotspot_temp 指标示例..... | 7  |
| 表 4-1 指标名称及说明.....                        | 9  |
| 表 4-2 标签说明 .....                          | 10 |

## 1. 概述

---

mx-exporter 是用于在集群环境中收集曦云®GPU 设备指标数据的工具。集群监控系统，如 Prometheus，可以通过 HTTP 从运行于每个节点的 mx-exporter 拉取设备指标数据。

## 2. 工具部署

mx-exporter 有多种安装方式，本文介绍两种安装方式：wheel 包安装和镜像安装。基于 Kubernetes 集群部署的安装方式，请参见《曦云®系列通用计算 GPU mx-exporter Kubernetes 集群监控部署手册》。

### 2.1 Wheel 包

#### 2.1.1 安装

##### 操作步骤

1. 确保系统已安装 Python 3。
2. 确保系统已安装 Python 库 prometheus\_client、grpcio、protobuf（版本需大于 3.12.0）。
3. 终端输入 `mx-smi -L`，查看板卡信息是否正常。
4. 安装 MXMACA SDK 之后，在 `/opt/maca/wheel` 目录下找到 Python 3 的 wheel 包：  
`mx_exporter_*.whl`。
5. 安装 wheel 包。

```
sudo pip3 install mx_exporter_*.whl
```

#### 2.1.2 使用方法

表 2-1 mx-exporter 参数说明（wheel 包安装）

| 参数                                           | 描述                                                                                                               |
|----------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| -p PORT, --port PORT                         | 主机需要开放的 HTTP 端口号。若不指定，默认为 8000                                                                                   |
| -i INTERVAL,<br>--interval INTERVAL          | mx-exporter 持续收集指标的间隔时间，单位 ms。若不指定，默认为 10000ms                                                                   |
| -c CONFIG_FILE,<br>--config-file CONFIG_FILE | 若不指定，默认配置文件为 <code>/opt/maca/etc/default-counters.csv</code><br>若自定义配置文件，需基于默认配置文件，参考 <a href="#">3.2 修改配置文件</a> |
| -m {0 1}, --mode {0 1}                       | 收集模式，默认为 0<br>0：根据设置的间隔时间收集数据，需与 <code>-i INTERVAL</code> 配合使用<br>1：当有请求时收集数据                                    |
| -lm {0 1}, --log-monitor {0 1}               | 监控 kernel 日志功能，默认为 0<br>0：收集 kernel 日志功能关闭                                                                       |

| 参数         | 描述                                                 |
|------------|----------------------------------------------------|
|            | 1: 收集 kernel 日志功能打开（配置该项需在 sudo 权限下启动 mx-exporter） |
| -h, --help | 显示帮助信息                                             |

## 启动 mx-exporter

### 示例 1

执行以下命令启动 mx-exporter，收集方式为有请求时收集，指标配置文件为/home/user/counters.csv：

```
mx-exporter -m 1 -c /home/user/counters.csv
```

### 示例 2

执行以下命令启动 mx-exporter，监听端口为 8002，收集指标间隔为 5s，打开监控 kernel 日志功能（需 sudo 执行）：

```
sudo mx-exporter -p 8002 -i 5000 -lm 1
```

## 查看 mx-exporter 抓取数据

在浏览器输入或 curl [http://<host\\_ip>:<host\\_port>](http://<host_ip>:<host_port>)，其中<host\_port>是主机需要开放的 HTTP 端口号，可用 -p 指定。

抓取到的 GPU 指标信息格式如下：

```
#HELP <指标名称> <指标描述>
#TYPE <指标名称> gauge
<指标名称>{Hostname="xx",bios_version="xx",deviceId="xx",driver_version="xx",exported_container="xx",exported_namespace="xx",exported_pod="xx",modelName="xx",uuid="xx"} XX
```

示例：

```
#HELP chip_hotspot_temp Chip hotspot temperature
#TYPE chip_hotspot_temp gauge
chip_hotspot_temp{Hostname="xx",bios_version="1.16.0.0",deviceId="0",driver_version="2.6.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="xx"} 35.75
```

## 2.1.3 卸载

### 操作步骤

1. 执行以下命令，卸载 mx-exporter wheel 包。

```
sudo pip3 uninstall mx-exporter
```

## 2.2 mx-exporter 镜像

### 2.2.1 加载镜像

#### 操作步骤

1. 解压 mx-exporter 镜像包：

```
tar -zxvf mx-exporter.xxx.tgz
```

2. 根据主机的架构加载对应的镜像。对于 x86 架构主机，使用 amd64 后缀的镜像；对于 Arm 架构主机，使用 arm64 后缀的镜像。

```
cd mx-exporter; docker load -i mx-exporter-xx-arm64.xz
```

### 2.2.2 使用方法

表 2-2 mx-exporter 参数说明（镜像部署）

| 参数                                            | 描述                                                                                                                                                                                             |
|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -p PORT, --port PORT                          | 容器端口号，默认为 8000                                                                                                                                                                                 |
| -i INTERVAL,<br>--interval INTERVAL           | mx-exporter 持续收集指标的间隔时间，单位 ms，默认为 10000ms<br>该时间间隔需与 prometheus pull 请求周期一致<br>若要修改间隔时间并生效，收集模式需为 0；收集模式为 1 时不生效                                                                               |
| -c CONFIG_FILE,<br>--config-file CONFIG_FILE  | 用户自定义的指标配置文件，默认在容器中/opt/mxexporter/mx_exporter/default-counters.csv；也可修改 mx-exporter.xxx.tgz 压缩包解压后的 mx-exporter/config/default-counters.csv，并在启动容器时传入使用<br>若自定义配置文件，需基于默认配置文件更改，参考 3.2 修改配置文件 |
| -m {0 1}, --mode {0 1}                        | 收集模式，默认为 0<br>0：根据设置的间隔时间收集数据，需与 -i INTERVAL 配合使用<br>1：当有请求时收集数据                                                                                                                               |
| -mp MOUNT_POINT,<br>--mount-point MOUNT_POINT | 容器中的挂载路径                                                                                                                                                                                       |
| -lm {0 1},<br>--log-monitor {0 1}             | 监控 kernel 日志功能，默认为 0<br>0：收集 kernel 日志功能关闭<br>1：收集 kernel 日志功能打开<br>需挂载主机 <b>var/log</b> 目录，启动 mx-exporter 时需增加 -v /var/log:/host/var/log 及 -mp host                                           |
| -h, --help                                    | 显示帮助信息                                                                                                                                                                                         |



表 2-3 Docker 参数说明

| 参数                                                               | 描述                                               |
|------------------------------------------------------------------|--------------------------------------------------|
| -d                                                               | 后台运行容器，并返回容器 ID                                  |
| --device=/dev/dri                                                | 将主机上的沐曦设备挂载到容器                                   |
| -v /var/log:/host/var/log                                        | 将主机/var/log 目录挂载到容器中/host/var/log                |
| --name=mx-exporter                                               | 指定容器名称为 <b>mx-exporter</b>                       |
| -p 0.0.0.0:<host_port>:<container_port>                          | host_port 为主机开放的 HTTP 端口<br>container_port 为容器端口 |
| -v /var/lib/kubelet/pod-resources:/var/lib/kubelet/pod-resources | 用于在容器中获取 Kubernetes 资源信息                         |

## 运行 mx-exporter

### 示例 1

执行以下命令运行 mx-exporter，host 端口为 8005，容器中端口为 8002，收集指标间隔为 5s，并修改指标显示 hostname 为 **hz-01**：

```
docker run -d --device=/dev/dri --name=mx-exporter -p 0.0.0.0:8005:8002 --hostname hz-01 <image ID> -p 8002 -i 5000
```

### 示例 2

执行以下命令运行 mx-exporter，用户修改主机目录下压缩包中的配置文件，并重命名为 **new\_counters.csv**，例如/home/<username>/mx-exporter/config/new\_counters.csv：

```
docker run -d --device=/dev/dri --name=mx-exporter -p 0.0.0.0:8000:8000 -v /home/<username>/mx-exporter/config/new_counters.csv:/opt/mxexporter/mx_exporter/new_counters.csv <image ID> -c /opt/mxexporter/mx_exporter/new_counters.csv
```

### 示例 3

执行以下命令运行 mx-exporter，打开监控 kernel 日志功能：

```
docker run -d --device=/dev/dri -v /var/log:/host/var/log --name=mx-exporter -p 0.0.0.0:8000:8000 <image ID> -lm 1 -mp=/host
```

### 示例 4

执行以下命令运行 mx-exporter，可显示 Kubernetes 正在使用的资源信息（指标标签：exported\_container、exported\_namespace、exported\_pod）：

```
docker run -d --device=/dev/dri --name=mx-exporter -p 0.0.0.0:8000:8000 -v /var/lib/kubelet/pod-resources:/var/lib/kubelet/pod-resources <image ID>
```

## 查看 mx-exporter 抓取数据

在浏览器输入或 `curl http://<host_ip>:<host_port>`，其中<host\_port>是主机需要开放的 HTTP 端口号。抓取到的 GPU 指标信息格式如下：

```
#HELP <指标名称> <指标描述>
#TYPE <指标名称> gauge
<指标名称>{Hostname="xx",bios_version="xx",deviceId="xx",driver_version="xx",exported_container="xx",exported_namespace="xx",exported_pod="xx",modelName="xx",uuid="xx"} XX
```

示例：

```
#HELP chip_hotspot_temp Chip hotspot temperature
#TYPE chip_hotspot_temp gauge
chip_hotspot_temp{Hostname="xx",bios_version="1.16.0.0",deviceId="0",driver_version="2.6.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="xx"} 35.75
```

### 2.2.3 删除容器

#### 操作步骤

1. 执行以下命令，删除 mx-exporter 容器。

```
docker stop <mx-exporter_container-id>
docker rm <mx-exporter_container-id>
```

## 3. 配置文件使用说明

### 3.1 默认配置文件

MXMACA SDK 安装完成后，默认配置文件为/opt/maca/etc/default-counters.csv。

表 3-1 默认配置文件表头说明

| 名称                 | 可修改 | 含义                                       |
|--------------------|-----|------------------------------------------|
| metric_id          | 否   | 指标 ID，用于识别指标的唯一标识符                       |
| metric_type        | 否   | 指标类别，如 Gauge                             |
| metric_name        | 是   | 指标名称，可根据实际需要修改                           |
| metric_description | 是   | 指标描述，可根据实际需要修改                           |
| label              | 是   | 指标标签名称，用于筛选指标所需展示的信息<br>只能更改名称，不能更改顺序及增删 |

表 3-2 默认配置文件中 chip\_hotspot\_temp 指标示例

| # Format: metric id | metric type | metric name       | metric description       | label                                                                                                                    |
|---------------------|-------------|-------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------|
| # temperature       |             |                   |                          |                                                                                                                          |
| chip_hotspot_temp   | Gauge       | chip_hotspot_temp | Chip hotspot temperature | deviceId, uuid, exported_pod, exported_namespace, exported_container, Hostname, driver_version, bios_version, modelName. |

### 3.2 修改配置文件

#### 3.2.1 自定义指标集

##### 操作步骤

1. 打开 mx-exporter.xxx.tgz 压缩包解压后的 mx-exporter/config/default-counters.csv。
2. 在不需要监控的指标行前加“#”注释，保存文件。
3. 运行 mx-exporter -c 指定文件。

4. 看到抓取的指标中，不包含已经注释掉的指标信息。

## 3.2.2 修改指标名称、指标描述及标签

### 操作步骤

1. 打开 `mx-exporter.xxx.tgz` 压缩包解压后的 `mx-exporter/config/default-counters.csv`。
2. 修改 metric name、metric description 及标签描述列信息，保存文件。
3. 运行 `mx-exporter -c` 指定文件。
4. 看到抓取的指标中，指标名称、指标描述及标签描述已更改为期望的内容。

### 说明

修改标签时只能更改名称，不能更改顺序及增删。

修改指标名称及标签的规范，请参考 [Prometheus 相关文档](#)。

### 示例

以 `chip_hotspot_temp` 指标为例，修改如下：

指标名称 `chip_hotspot_temp` 修改为 `mx_chip_hotspot_temp`  
 指标描述 `Chip hotspot temperature` 修改为 `Metax Chip hotspot temperature`  
 指标标签 `deviceId` 修改为 `mx_deviceId`  
 指标标签 `modelName` 修改为 `mx_modelName`

运行 `mx-exporter` 后，得到的数据显示如下：

```
#HELP mx_chip_hotspot_temp Metax Chip hotspot temperature
#TYPE mx_chip_hotspot_temp gauge
mx_chip_hotspot_temp{Hostname="xx",bios_version="1.16.0.0",
mx_modelName="MXC500",
mx_deviceId="0",driver_version="2.6.0",exported_container="",exported_namesp
ace="",exported_pod="",uuid="xx"} 35.75
```

## 4. 指标及标签说明

### 4.1 指标说明

表 4-1 指标名称及说明

| 指标名称                | 所属类别        | 指标说明                                                               |
|---------------------|-------------|--------------------------------------------------------------------|
| chip_hotspot_temp   | Temperature | 芯片内部最高温度，单位摄氏度                                                     |
| board_core_temp     | Temperature | 板卡温度，单位摄氏度                                                         |
| optical_module_temp | Temperature | 光模块温度，仅部分设备有光模块，如 C500X                                            |
| vpue_usage          | Usage       | 编码器利用率                                                             |
| vpud_usage          | Usage       | 解码器利用率                                                             |
| memory_usage        | Usage       | HBM 及系统内存利用率，指标标签 type 用于区分 HBM 和系统内存，vram 代表 HBM，xtt 代表系统内存       |
| memory_total        | Usage       | HBM 及系统内存总量，单位 KB，指标标签 type 用于区分 HBM 和系统内存，vram 代表 HBM，xtt 代表系统内存  |
| memory_used         | Usage       | HBM 及系统内存使用量，单位 KB，指标标签 type 用于区分 HBM 和系统内存，vram 代表 HBM，xtt 代表系统内存 |
| gpu_usage           | Usage       | GPU 利用率                                                            |
| board_power         | Power       | 功耗，单位毫瓦                                                            |
| vpue_clock          | Clock       | 编码器时钟频率，单位 MHz                                                     |
| vpud_clock          | Clock       | 解码时钟频率，单位 MHz                                                      |
| mem_clock           | Clock       | HBM 时钟频率，单位 MHz                                                    |
| gpu_clock           | Clock       | GPU 时钟频率，单位 MHz                                                    |
| pcie_bw             | PCIE        | PCIe Tx 和 Rx 吞吐量，单位 MB/s                                           |
| pcie_speed          | PCIE        | 设备每条 lane 上的 PCIe 传输速率，单位 GT/s                                     |
| pcie_width          | PCIE        | 设备带宽，表示 PCIe lanes                                                 |
| pcie_bridge_speed   | PCIE        | PCIe bridge 传输速率，单位 GT/s                                           |
| pcie_bridge_width   | PCIE        | PCIe bridge lanes                                                  |
| mxlk_bw             | MetaXLink   | MetaXLink Tx 和 Rx 吞吐量，单位 MB/s                                      |
| mxlk_speed          | MetaXLink   | MetaXLink 实时速率，单位 GT/s                                             |
| mxlk_width          | MetaXLink   | MetaXLink lanes                                                    |
| xcore_dpm_level     | DPM         | 当前 DPM 等级                                                          |

| 指标名称            | 所属类别   | 指标说明                   |
|-----------------|--------|------------------------|
| mx_kernel_error | Log    | Kernel 日志级别为 Error 的信息 |
| gpu_state       | Status | 设备状态信息，0 为不可用，1 为可用    |

## 4.2 标签说明

表 4-2 标签说明

| 名称                 | 含义                                                                                                                        |
|--------------------|---------------------------------------------------------------------------------------------------------------------------|
| Hostname           | 主机名称<br>若 mx-exporter 在容器中运行，则为容器名称，若需修改，可在容器运行传入--hostname <hostname>                                                    |
| bios_version       | BIOS 版本信息                                                                                                                 |
| deviceId           | 设备号，即板卡号（通过 mx-smi -L 获取板卡 ID 信息）                                                                                         |
| driver_version     | 设备驱动版本信息                                                                                                                  |
| exported_container | 正在使用设备的 k8s pod 的 container 名称，仅当启动 mx-exporter 容器传入-v /var/lib/kubelet/pod-resources:/var/lib/kubelet/pod-resources 时可显示 |
| exported_namespace | 正在使用设备的 pod 所在 k8s 命名空间，仅当启动 mx-exporter 容器传入-v /var/lib/kubelet/pod-resources:/var/lib/kubelet/pod-resources 时可显示        |
| exported_pod       | 正在使用设备的 k8s pod 名称，仅当启动 mx-exporter 容器传入-v /var/lib/kubelet/pod-resources:/var/lib/kubelet/pod-resources 时可显示             |
| modelName          | 芯片型号                                                                                                                      |
| uuid               | 设备 UUID                                                                                                                   |

## 4.3 指标输出示例

使用以下命令启动 mx-exporter：

```
docker run -d --device=/dev/dri -v /var/log:/host/var/log -v
/var/lib/kubelet/pod-resources:/var/lib/kubelet/pod-resources --name=mx-
exporter -p 0.0.0.0:8000:8000 <image ID> -lm 1 -mp=/host
```

curl http://<host\_ip>:8000 可得到以下内容：

```
# # HELP mx_kernel_error Kernel errors
# TYPE mx_kernel_error gauge
mx_kernel_error{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driv
er_version="2.3.0",err_level="ERROR",exported_container="",exported_namespac
e="",exported_pod="",modelName="MXC500",module="SMP",uuid="GPU-1d34ca5a-
b3d9-65df-0ffc-8ebe0d6347d9"} 1.0
```

```
mx_kernel_error{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",err_level="ERROR",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",module="SMI",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 1.0
# HELP device_type Device type
# TYPE device_type gauge
device_type{deviceId="0",deviceType="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 1.0
# HELP bios_ver Bios version
# TYPE bios_ver gauge
bios_ver{bios="1.7.4.0",deviceId="0"} 1.0
# HELP driver_ver Driver version
# TYPE driver_ver gauge
driver_ver{deviceId="0",driver="2.3.0"} 1.0

# HELP chip_hotspot_temp Chip hotspot temperature
# TYPE chip_hotspot_temp gauge
chip_hotspot_temp{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 36.25
# HELP board_core_temp Board DrMOS Core temperature
# TYPE board_core_temp gauge
board_core_temp{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 35.0
# HELP vpu_usage Vpu usage in percent
# TYPE vpu_usage gauge
vpu_usage{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
# HELP vpu_usage Vpu usage in percent
# TYPE vpu_usage gauge
vpu_usage{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
# HELP memory_usage HBM and system memory usage in percent
# TYPE memory_usage gauge
memory_usage{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",type="vram",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 1.2849688529968262
memory_usage{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",type="xtt",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.001035634935770525
# HELP memory_total Total HBM and system memory in KB
# TYPE memory_total gauge
memory_total{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",
```



```

modelName="MXC500",type="vram",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 6.7108864
memory_total{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",type="xtt",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 1.585114545
# HELP memory_used HBM and system memory used in KB
# TYPE memory_used gauge
memory_used{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",type="vram",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 862328.0
memory_used{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",type="xtt",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 16416.0
# HELP gpu_usage GPU usage in percent
# TYPE gpu_usage gauge
gpu_usage{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
# HELP board_power Board power in milliwatt
# TYPE board_power gauge
board_power{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 55983.0
# HELP vpue_clock Encoder clock frequency in MHz
# TYPE vpue_clock gauge
vpue_clock{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 535.0
# HELP vpud_clock Decoder clock frequency in MHz
# TYPE vpud_clock gauge
vpud_clock{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 525.0
# HELP mem_clock Memory clock frequency in MHz
# TYPE mem_clock gauge
mem_clock{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 1800.0
# HELP gpu_clock GPU clock frequency in MHz
# TYPE gpu_clock gauge
gpu_clock{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 428.0
# HELP pcie_speed Pcie current speed in GT/s
# TYPE pcie_speed gauge
pcie_speed{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 32.0

```



```
# HELP pcie_width Pcie current lanes
# TYPE pcie_width gauge
pcie_width{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 16.0
# HELP mxlk_speed MetaXLink current link speed in GT/s
# TYPE mxlk_speed gauge
mxlk_speed{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="0",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
mxlk_speed{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="1",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
mxlk_speed{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="2",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
mxlk_speed{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="3",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 32.0
mxlk_speed{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="4",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 32.0
mxlk_speed{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="5",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 32.0
mxlk_speed{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="6",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
# HELP mxlk_width MetaXLink current link width
# TYPE mxlk_width gauge
mxlk_width{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="0",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
mxlk_width{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="1",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
mxlk_width{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="2",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
mxlk_width{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="3",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
```

```
delName="MXC500",mxlkId="3",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"}
16.0
mxlk_width{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="4",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"}
16.0
mxlk_width{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="5",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"}
16.0
mxlk_width{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",mxlkId="6",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"}
0.0
# HELP xcore_dpm_level Dpm xcore performance level
# TYPE xcore_dpm_level gauge
xcore_dpm_level{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 0.0
# HELP gpu_state GPU state: 0(not available) 1(available)
# TYPE gpu_state gauge
gpu_state{Hostname="mxsrv003",bios_version="1.7.4.0",deviceId="0",driver_version="2.3.0",exported_container="",exported_namespace="",exported_pod="",modelName="MXC500",uuid="GPU-1d34ca5a-b3d9-65df-0ffc-8ebe0d6347d9"} 1.0
```

## 5. 附录

### 5.1 术语/缩略语

| 术语/缩略语    | 全称                                        | 说明              |
|-----------|-------------------------------------------|-----------------|
| PCIe      | Peripheral Component Interconnect Express | 一种高速串行计算机扩展总线标准 |
| DPM       | Dynamic Power Management                  | 动态功率管理功能        |
| VPUE      | Video Processing Unit Encoder             | 视频处理单元编码        |
| VPUD      | Video Processing Unit Decoder             | 视频处理单元解码        |
| MetaXLink |                                           | 沐曦 GPU D2D 接口总线 |

## 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。