Azure A100 验证 MIG 测试

本手册基于 Azure A100 验证 MIG 测试。

MIG 作为一种虚拟化 GPU 的方法，在隔离、安全、QoS 上都有好的表现。相比以前的 Time-sliced 模式，其在带宽均衡、计算损失、故障处理方面都能做到更优。MIG 是一种云服务器 GPU 发展的方向，支持了更多的用户需求。新的 MIG 功能可以将每个 A100 划分为多达 7 个 GPU 实例，以实现最佳利用，有效地扩大每个用户和应用程序的访问量。

本验证中涵盖的相关技术工具：

DCGM（Data Center GPU Manager）是英伟达（NVIDIA）提供的一套数据中心 GPU 管理工具。它用于监控、管理和诊断数据中心环境中的 GPU 资源，提供 GPU 的运行状态、性能指标、健康状况监测、故障诊断和资源调度等功能。DCGM 支持通过命令行工具、API 接口或与其他监控系统集成，帮助管理员更高效地管理 GPU 集群，优化 GPU 资源利用率。

CUDA 测试生成器 （dcgmproftester）

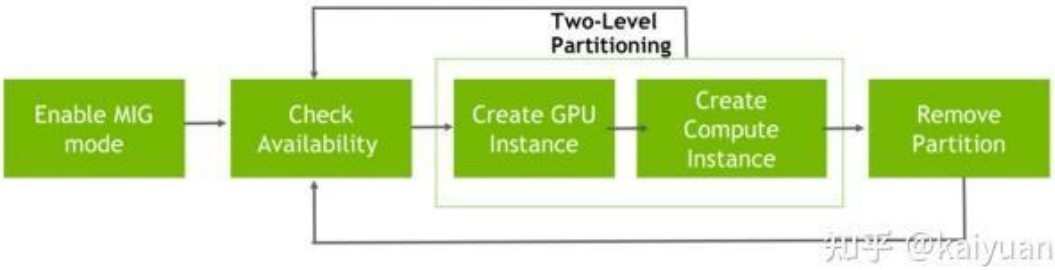
dcgmproftester 是一个 CUDA 负载生成器。它可用于生成确定性的 CUDA 工作负载以进行读取和 验证 GPU 指标。该工具作为简单的 x86_64 Linux 二进制文件以及编译为 PTX 的 CUDA 内核提供。 客户可以结合使用该工具在 GPU 上快速生成负载，并通过 stdout 查看 DCGM 报告的指标。dcgmidcgmi dmon dcgmproftester 将两个重要参数作为输入：用于为特定指标生成负载 （例如，

使用 1004 为 Tensor Core 生成半精度矩阵乘法累加）并指定测试持续时间。

前提准备

MIG 的操作顺序概况为：

Enable MIG -> 创建 GI 实例 -> 创建 CI 实例 -> 删除 CI 实例 -> 删除 GI 实例 -> 关闭 MIG。



1. **产看是否有进程在这个 GPU 上运行**

   如下图可以看到有三个进程正在 GPU 上运行

   (base) olivia@a100-48:~$ nvidia-smi

```
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.230.02             Driver Version: 535.230.02   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA A100 80GB PCIe          On  | 00000001:00:00.0 Off |                    0 |
| N/A   34C    P0              53W / 300W |     16MiB / 81920MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   1  NVIDIA A100 80GB PCIe          On  | 00000002:00:00.0 Off |                    0 |
| N/A   34C    P0              52W / 300W |    160MiB / 81920MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|    0   N/A  N/A      1895      G   /usr/lib/xorg/Xorg                            4MiB |
|    1   N/A  N/A      1895      G   /usr/lib/xorg/Xorg                          130MiB |
|    1   N/A  N/A      2736      G   /usr/bin/gnome-shell                         12MiB |
+---------------------------------------------------------------------------------------+
```

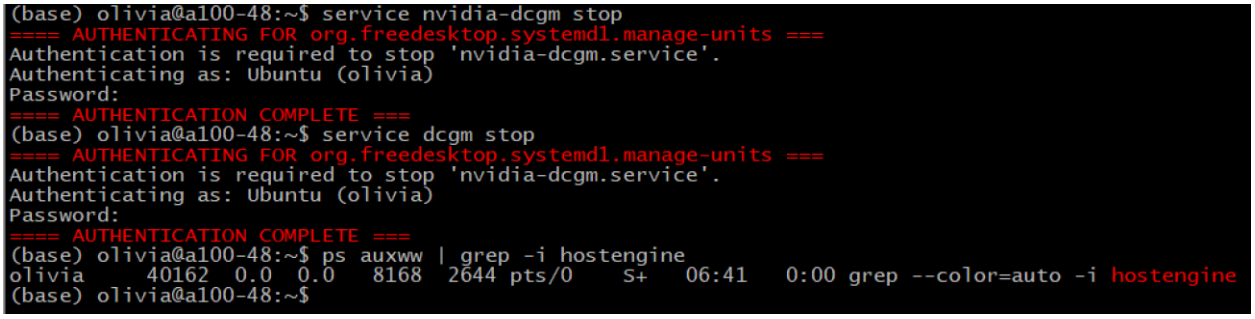
2. 关闭 dcgm：

   运行以下命令，然后再次通过 nvidia-smi 查看是否有进程运行在 GPU 上

(base) olivia@a100-48:~$ service nvidia-dcgm stop

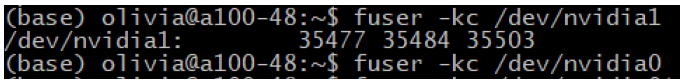
(base) olivia@a100-48:~$ service dcgm stop

(base) olivia@a100-48:~$ ps auxww | grep -i hostengine



3.如果以上操作仍旧不生效，请 kill 进程

(base) olivia@a100-48:~$ fuser -kc /dev/nvidia1

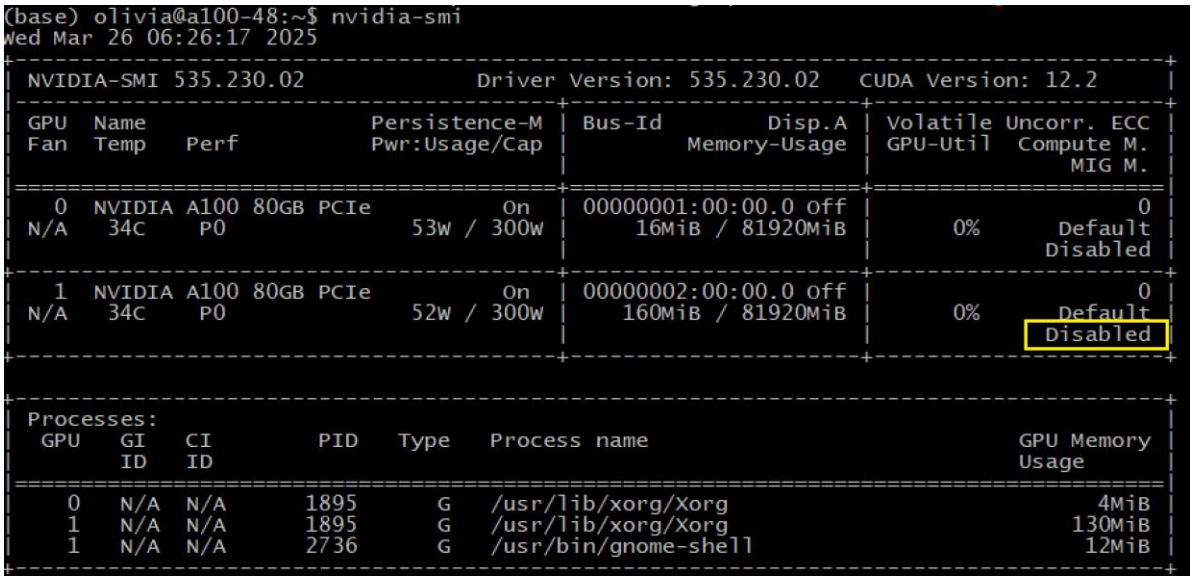


一、 启动 MIG 模式

1. 登录 GPU 后，查看是否启动 MIG 模式.

通过运行命令，可以看到当前 A100 的 MIG 模式是 Disabled

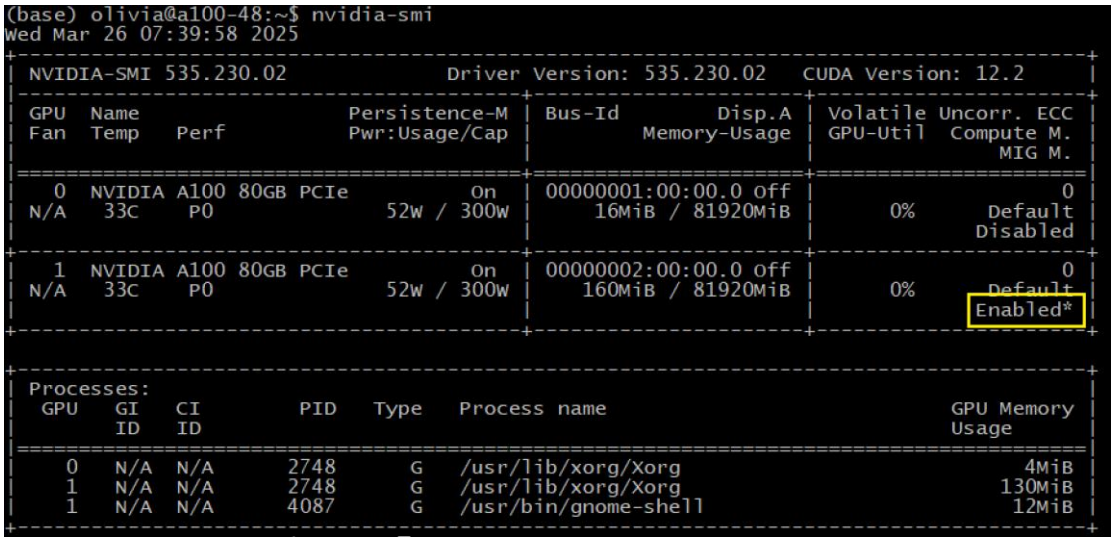
(base) olivia@a100-48:~$ nvidia-smi



2. 启动 MIG 模式

启动第 1 个 GPU 的 MIG 模式, 然后查看 MIG 是否启动:

(base) olivia@a100-48:~$ sudo nvidia-smi -i 1 -mig 1



可以发现 MIG 的状态是“Enbled*”，此时重启机器. 重启后，再次查看设备状态，发现已经是“Enabled”状态。

```
(base) olivia@a100-48:~$ nvidia-smi
Wed Mar 26 07:52:04 2025
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.230.02              Driver Version: 535.230.02   CUDA Version: 12.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A100 80GB PCIe          On | 00000001:00:00.0 Off |                    0 |
| N/A   30C    P0              42W / 300W |      0MiB / 81920MiB |      0%      Default |
|                                         |                      |             Disabled |
+-------------------------------+----------------------+----------------------+
|   1  NVIDIA A100 80GB PCIe          On | 00000002:00:00.0 Off |                   On |
| N/A   31C    P0              42W / 300W |      0MiB / 81920MiB |     N/A      Default |
|                                         |                      |             Enabled |
+-------------------------------+----------------------+----------------------+
```

二、 创建 MIG 实例

1. 查看 GPU 上支持的 MIG profile，如果用第 9 个 profile，可以创建两个 MIG ：

   (base) olivia@a100-48:~$ sudo nvidia-smi mig -lgip

```
(base) olivia@a100-48:~$ sudo nvidia-smi mig -lgip
+--------------------------------------------------------------------------------+
| GPU instance profiles:                                                         |
| GPU   Name             ID    Instances   Memory     P2P    SM    DEC   ENC   |
|                              Free/Total   GiB              CE    JPEG  OFA   |
|================================================================================|
|   1  MIG 1g.10gb       19     7/7         9.50       No     14     0     0   |
|                                                              1     0     0   |
+--------------------------------------------------------------------------------+
|   1  MIG 1g.10gb+me    20     1/1         9.50       No     14     1     0   |
|                                                              1     1     1   |
+--------------------------------------------------------------------------------+
|   1  MIG 1g.20gb       15     4/4         19.50      No     14     1     0   |
|                                                              1     0     0   |
+--------------------------------------------------------------------------------+
|   1  MIG 2g.20gb       14     3/3         19.50      No     28     1     0   |
|                                                              2     0     0   |
+--------------------------------------------------------------------------------+
|   1  MIG 3g.40gb        9     2/2         39.25      No     42     2     0   |
|                                                              3     0     0   |
+--------------------------------------------------------------------------------+
|   1  MIG 4g.40gb        5     1/1         39.25      No     56     2     0   |
|                                                              4     0     0   |
+--------------------------------------------------------------------------------+
|   1  MIG 7g.80gb        0     1/1         78.75      No     98     5     0   |
|                                                              7     1     1   |
+--------------------------------------------------------------------------------+
```

   列出不同 profile 可能的 GPU 实例位置：

   (base) olivia@a100-48:~$ smcuser@smc:~$ sudo nvidia-smi mig -lgipp

```
(base) olivia@a100-48:~$ sudo   nvidia-smi mig -lgipp
GPU  1 Profile ID 19 Placements: {0,1,2,3,4,5,6}:1
GPU  1 Profile ID 20 Placements: {0,1,2,3,4,5,6}:1
GPU  1 Profile ID 15 Placements: {0,2,4,6}:2
GPU  1 Profile ID 14 Placements: {0,2,4}:2
GPU  1 Profile ID  9 Placements: {0,4}:4
GPU  1 Profile ID  5 Placement : {0}:4
GPU  1 Profile ID  0 Placement : {0}:8
```

2. 使用 profile 9 创建 2 个 GPU instance：

(base) olivia@a100-48:~/llm$ sudo nvidia-smi mig -cgi 9

```
(base) olivia@a100-48:~/llm/deepseek$ sudo nvidia-smi mig -cgi 9
Successfully created GPU instance ID  1 on GPU  1 using profile MIG 3g.40gb (ID  9)
```

查看结果：

sudo nvidia-smi mig -lgi

```
(base) olivia@a100-48:~/llm/deepseek$ sudo nvidia-smi mig -lgi
+-------------------------------------------------------+
| GPU instances:                                        |
| GPU   Name             Profile  Instance   Placement  |
|                          ID       ID       Start:Size |
|=======================================================|
|   1  MIG 3g.40gb          9        1          4:4     |
+-------------------------------------------------------+
```

再次运行新建命令并产看结果：

base) olivia@a100-48:~/llm$ sudo nvidia-smi mig -cgi 9

```
(base) olivia@a100-48:~/llm/deepseek$ sudo nvidia-smi mig -lgi
+-------------------------------------------------------+
| GPU instances:                                        |
| GPU   Name             Profile  Instance   Placement  |
|                          ID       ID       Start:Size |
|=======================================================|
|   1  MIG 3g.40gb          9        1          4:4     |
+-------------------------------------------------------+
|   1  MIG 3g.40gb          9        2          0:4     |
+-------------------------------------------------------+
```

3. 根据 GPU instance 实例创建计算实例 compute instance：

(base) olivia@a100-48:~/llm$ sudo nvidia-smi mig -cci -gi 1,2

```
(base) olivia@a100-48:~/llm/deepseek$ sudo nvidia-smi mig -cci -gi 1, 2
Successfully created compute instance ID  0 on GPU  1 GPU instance ID  1 using profile MIG 3g.40gb (ID  2)
```

执行完上述命令后，确认 MIG 实例是否可见：

(base) olivia@a100-48:~/llm$ nvidia-smi

```
(base) olivia@a100-48:~/llm/deepseek$ !nvi
nvidia-smi
Thu Mar 27 03:40:20 2025
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 535.230.02              Driver Version: 535.230.02   CUDA Version: 12.2     |
|-------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                     |                      |               MIG M. |
|=====================================+======================+======================|
|   0  NVIDIA A100 80GB PCIe          On  | 00000001:00:00.0 Off |                    0 |
| N/A   29C    P0              43W / 300W |      0MiB / 81920MiB |      0%      Default |
|                                     |                      |             Disabled |
+-------------------------------------+----------------------+----------------------+
|   1  NVIDIA A100 80GB PCIe          On  | 00000002:00:00.0 Off |                   On |
| N/A   29C    P0              44W / 300W |     74MiB / 81920MiB |     N/A      Default |
|                                     |                      |             Enabled |
+-------------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| MIG devices:                                                                |
+------------------+----------------------------------+-----------+-----------------------+
| GPU  GI  CI  MIG |                  Memory-Usage    |       Vol|      Shared           |
|      ID  ID  Dev |                    BAR1-Usage    | SM    Unc| CE ENC DEC OFA JPG    |
|                  |                                  |       ECC|                       |
|==================+==================================+===========+=======================|
|  1    1   0   0  |              37MiB / 40192MiB    | 42      0 |  3   0    2    0    0 |
|                  |               0MiB / 65535MiB    |           |                       |
+------------------+----------------------------------+-----------+-----------------------+
|  1    2   0   1  |              37MiB / 40192MiB    | 42      0 |  3   0    2    0    0 |
|                  |               0MiB / 65535MiB    |           |                       |
+------------------+----------------------------------+-----------+-----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

三、通过 CUDA dcgmproftester 执行压测

通过 DCGM 服务监控 MIG

1. 启动 DCGM 服务：

(base) olivia@a100-48:~$ sudo service dcgm start

(base) olivia@a100-48:~$ sudo service nvidia-dcgm start

(base) olivia@a100-48:~$ ps -ef |grep -i dcgm

```
(base) olivia@a100-48:~$ sudo service nvidia-dcgm start
(base) olivia@a100-48:~$ ps -ef |grep -i dcgm
root        1376       1  0 03:15 ?        00:00:01 /snap/dcgm/45/usr/bin/nv-hostengine -n -p 5555
olivia     40099    4137  0 03:57 pts/0    00:00:00 grep --color=auto -i dcgm
(base) olivia@a100-48:~$ sudo service dcgm start
```

有的时候 DCGM 服务会无法正常启动，可执行如下操作修复：

你可以尝试以下步骤排查问题：

1. 检查 DCGM 日志，查看具体错误信息：

```
sudo journalctl -u nvidia-dcgm.service --no-pager
```

2. 确认 NVIDIA 驱动和 DCGM 版本兼容性：

```
nvidia-smi
dcgmi --version
```

3. 尝试手动运行 DCGM，查看具体报错：

```
sudo /usr/bin/nv-hostengine -n --service-account nvidia-dcgm
```

4. 如果以上步骤未解决，尝试重新安装或更新 DCGM：

```
sudo apt update
sudo apt install --reinstall datacenter-gpu-manager
```

完成上述步骤后，再次尝试启动服务：

```
sudo systemctl restart nvidia-dcgm
sudo systemctl status nvidia-dcgm
```

如果仍旧尚未启动，请 reboot 系统

2. 确认服务已经启动:

(base) olivia@a100-48:~/llm$ sudo systemctl status nvidia-dcgm

```
● (base) olivia@a100-48:~/llm/deepseek$ sudo systemctl status  nvidia-dcgm
● nvidia-dcgm.service - NVIDIA DCGM service
     Loaded: loaded (/lib/systemd/system/nvidia-dcgm.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2025-03-27 04:11:14 UTC; 3min 25s ago
   Main PID: 4483 (nv-hostengine)
      Tasks: 8 (limit: 532061)
     Memory: 13.1M
     CGroup: /system.slice/nvidia-dcgm.service
             └─4483 /usr/bin/nv-hostengine -n --service-account nvidia-dcgm

Mar 27 04:11:36 a100-48 systemd[1]: nvidia-dcgm.service: Dependency Conflicts=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:11:36 a100-48 systemd[1]: nvidia-dcgm.service: Dependency ConflictedBy=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:13:07 a100-48 systemd[1]: nvidia-dcgm.service: Dependency Conflicts=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:13:07 a100-48 systemd[1]: nvidia-dcgm.service: Dependency ConflictedBy=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:13:07 a100-48 systemd[1]: nvidia-dcgm.service: Dependency Conflicts=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:13:07 a100-48 systemd[1]: nvidia-dcgm.service: Dependency ConflictedBy=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:13:08 a100-48 systemd[1]: nvidia-dcgm.service: Dependency Conflicts=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:13:08 a100-48 systemd[1]: nvidia-dcgm.service: Dependency ConflictedBy=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:13:08 a100-48 systemd[1]: nvidia-dcgm.service: Dependency Conflicts=dcgm.service dropped, merged into nvidia-dcgm.service
Mar 27 04:13:08 a100-48 systemd[1]: nvidia-dcgm.service: Dependency ConflictedBy=dcgm.service dropped, merged into nvidia-dcgm.service
```

保证 dcgm 和 nv-hostengine 的版本保持一致:

(base) olivia@a100-48:~/llm$ nv-hostengine –version
(base) olivia@a100-48:~/llm$ dcgmi -v

```
(base) olivia@a100-48:~/llm/deepseek$ nv-hostengine --version
Version : 3.3.9
Build ID : 45
Build Date : 2024-11-13
Build Type : Release
Commit ID : 9e2b5d2b8914d2571537f9f633e5a91986d4eecd
Branch Name : rel_dcgm_3_3
CPU Arch : x86_64
Build Platform : Linux 4.15.0-180-generic #189-Ubuntu SMP Wed May 18 14:13:57 UTC 2022 x86_64
CRC : 813bd4bc82cddbb63b59936dc0740c84

(base) olivia@a100-48:~/llm/deepseek$ dcgmi -v
Version : 3.3.9
Build ID : 45
Build Date : 2024-11-13
Build Type : Release
Commit ID : 9e2b5d2b8914d2571537f9f633e5a91986d4eecd
Branch Name : rel_dcgm_3_3
CPU Arch : x86_64
Build Platform : Linux 4.15.0-180-generic #189-Ubuntu SMP Wed May 18 14:13:57 UTC 2022 x86_64
CRC : 813bd4bc82cddbb63b59936dc0740c84

Hostengine build info:
Version : 3.3.9
Build ID : 45
Build Date : 2024-11-13
Build Type : Release
Commit ID : 9e2b5d2b8914d2571537f9f633e5a91986d4eecd
Branch Name : rel_dcgm_3_3
CPU Arch : x86_64
Build Platform : Linux 4.15.0-180-generic #189-Ubuntu SMP Wed May 18 14:13:57 UTC 2022 x86_64
CRC : 813bd4bc82cddbb63b59936dc0740c84
```

3. 验证 DCGM 的列举的设备：
   (base) olivia@a100-48:~/llm$ dcgmi discovery -c

```
(base) olivia@a100-48:~/llm/deepseek$ dcgmi discovery -c
+-------------------+--------------------------------------------------------------------+
| Instance Hierarchy                                                                     |
+===================+====================================================================+
| GPU 1             | GPU GPU-9d676d4a-1a5a-e5e0-f4a0-ab9bc8d406e5 (EntityID: 1)         |
| -> I 1/1          | GPU Instance (EntityID: 7)                                         |
|    -> CI 1/1/0    | Compute Instance (EntityID: 7)                                     |
| -> I 1/2          | GPU Instance (EntityID: 8)                                         |
|    -> CI 1/2/0    | Compute Instance (EntityID: 8)                                     |
+-------------------+--------------------------------------------------------------------+
```

4. 查看当前的负载：

   dcgmi dmon -i i:7,i:8 -e 1001,1002,1003,1004,1005,1006,1007,1008,1009,1010 -c 1

```
(base) olivia@a100-48:~/llm/deepseek$ sudo dcgmi dmon  -i i:7,i:8  -e 1001,1002,1003,1004  -c 10
#Entity   GRACT        SMACT        SMOCC        TENSO
ID
GPU-I 7   0.000        0.000        0.000        0.000
GPU-I 8   0.000        0.000        0.000        0.000
GPU-I 7   0.000        0.000        0.000        0.000
GPU-I 8   0.000        0.000        0.000        0.000
GPU-I 7   0.000        0.000        0.000        0.000
GPU-I 8   0.000        0.000        0.000        0.000
GPU-I 7   0.000        0.000        0.000        0.000
GPU-I 8   0.000        0.000        0.000        0.000
GPU-I 7   0.000        0.000        0.000        0.000
GPU-I 8   0.000        0.000        0.000        0.000
GPU-I 7   0.000        0.000        0.000        0.000
GPU-I 8   0.000        0.000        0.000        0.000
GPU-I 7   0.000        0.000        0.000        0.000
GPU-I 8   0.000        0.000        0.000        0.000
GPU-I 7   0.000        0.000        0.000        0.000
GPU-I 8   0.000        0.000        0.000        0.000
```

命令作用：

该命令会以动态监控模式，针对 GPU 实例 7 和 8，每隔 10 秒采集一次以下性能指标,当前输出可见负载为 0:

- 图形引擎使用率 (GRACT)
- 流式多处理器使用率 (SMACT)
- Tensor Core 使用率 (TCACT)
- 显存使用率 (DRAMACT)
- PCIe 传输带宽（发送和接收方向）
- NVLink 传输带宽（发送和接收方向）
- SM 时钟频率
- 显存时钟频率

命令参数解析：

1) **dcgmi dmon**

   启动 DCGM 的动态监控模式（Dynamic Monitoring），用于实时监控 GPU 的性能指标。

2) **-i i:7,i:8**

   指定监控的 GPU 实例（Instance）编号为 7 和 8。
   - i:7 和 i:8 表示 GPU 实例编号（MIG 模式下的实例 ID）。
   - 如果未启用 MIG 模式，则可以直接指定 GPU ID（如 -i 0,1）。

3) **-e 1001,1002,1003,1004,1005,1006,1007,1008,1009,1010**

   指定需要监控的性能指标（Metrics），每个编号对应一个具体的指标:
   - 1001：GRACT (Graphics Activity)，GPU 图形引擎的使用率（百分比）。
   - 1002：SMACT (SM Activity)，流式多处理器（Streaming Multiprocessor）的使用率（百分比）。
   - 1003：TCACT (Tensor Core Activity)，Tensor Core 的使用率（百分比）。
   - 1004：DRAMACT (DRAM Activity)，显存（DRAM）的使用率（百分比）。
   - 1005：PCIE_TX (PCIe TX Throughput)，PCIe 传输带宽（发送方向，单位为 MB/s）。
   - 1006：PCIE_RX (PCIe RX Throughput)，PCIe 传输带宽（接收方向，单位为 MB/s）。
   - 1007：NVLINK_TX (NVLink TX Throughput)，NVLink 传输带宽（发送方向，单位为 MB/s）。
   - 1008：NVLINK_RX (NVLink RX Throughput)，NVLink 传输带宽（接收方向，单位为 MB/s）。
   - 1009：SM Clock，流式多处理器（SM）的时钟频率（单位为 MHz）。
   - 1010：Memory Clock，显存的时钟频率（单位为 MHz）。

4) **-d 5**

   指定采集数据的时间间隔为 5 秒，即每隔 5 秒采集一次监控数据。

5) **-c 1**

   指定采集数据的次数为 1，即命令会采集一次监控数据后停止

输出解读：

- **#Entity ID**：监控的 GPU 实例编号（如 GPU-I 7 和 GPU-I 8）。
- **GRACT**：图形引擎使用率（百分比）。
- **SMACT**：流式多处理器使用率（百分比）。
- **TCACT**：Tensor Core 使用率（百分比）。
- **DRAMACT**：显存使用率（百分比）。
- **PCIE_TX 和 PCIE_RX**：PCIe 传输带宽（单位为 MB/s）。
- **NVLINK_TX 和 NVLINK_RX**：NVLink 传输带宽（单位为 MB/s）。
- **SM_CLK**：流式多处理器的时钟频率（单位为 MHz）。
- **MEM_CLK**：显存的时钟频率（单位为 MHz）。
  如果所有指标均为 0.000 或异常值，说明 GPU 当前处于空闲状态，或任务未正确分配到 GPU。

5. 生成确定性的 CUDA 工作负载

   下面命令中 dcgmproftester12 的版本需要根据当前系统做调整：
   (base) olivia@a100-48:~$ sudo dcgmproftester12 --no-dcgm-validation -d 5 -t 1001,1002,1003,1004,1005,1006,1007,1008,1009,1010

```
(base) olivia@a100-48:~/llm/deepseek$ sudo dcgmproftester12 --no-dcgm-validation -t 1004 -d 5
Skipping CreateDcgmGroups() since DCGM validation is disabled
Skipping CreateDcgmGroups() since DCGM validation is disabled
Skipping CreateDcgmGroups() since DCGM validation is disabled
Worker 1:1 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (0.00 gflops).
Worker 1:0 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (0.00 gflops).
Worker 0:0 [1004]: TensorEngineActive:  generated ???, dcgm 0 (2.17e+05 gflops).
Worker 1:1 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.68e+04 gflops).
Worker 1:0 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.72e+04 gflops).
Worker 0:0 [1004]: TensorEngineActive:  generated ???, dcgm 0 (2.22e+05 gflops).
Worker 1:1 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.77e+04 gflops).
Worker 1:0 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.82e+04 gflops).
Worker 0:0 [1004]: TensorEngineActive:  generated ???, dcgm 0 (2.20e+05 gflops).
Worker 1:1 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.78e+04 gflops).
Worker 1:0 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.82e+04 gflops).
Worker 0:0 [1004]: TensorEngineActive:  generated ???, dcgm 0 (2.19e+05 gflops).
Worker 1:1 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.75e+04 gflops).
Worker 1:0 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.80e+04 gflops).
Worker 0:0 [1004]: TensorEngineActive:  generated ???, dcgm 0 (2.18e+05 gflops).
Worker 1:1 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.78e+04 gflops).
Worker 1:0 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.81e+04 gflops).
Worker 0:0 [1004]: TensorEngineActive:  generated ???, dcgm 0 (2.19e+05 gflops).
Worker 1:1 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.75e+04 gflops).
Worker 1:0 [1004]: TensorEngineActive:  generated ???, dcgm M:{ GPU: 0, GI: 0, CI: 0 } (9.82e+04 gflops).
Worker 1:1[1004]: Message: std::setenv successfully set CUDA_VISIBLE_DEVICES to MIG-GPU-9d676d4a-1a5a-e5e0-f4a0-ab9bc8d406e5/2/0
DCGM CudaContext Init completed successfully.

CU_DEVICE_ATTRIBUTE_MAX_THREADS_PER_MULTIPROCESSOR: 2048
CUDA_VISIBLE_DEVICES: MIG-GPU-9d676d4a-1a5a-e5e0-f4a0-ab9bc8d406e5/2/0
CU_DEVICE_ATTRIBUTE_MULTIPROCESSOR_COUNT: 42
CU_DEVICE_ATTRIBUTE_MAX_SHARED_MEMORY_PER_MULTIPROCESSOR: 167936
CU_DEVICE_ATTRIBUTE_COMPUTE_CAPABILITY_MAJOR: 8
CU_DEVICE_ATTRIBUTE_COMPUTE_CAPABILITY_MINOR: 0
CU_DEVICE_ATTRIBUTE_GLOBAL_MEMORY_BUS_WIDTH: 2560
CU_DEVICE_ATTRIBUTE_MEMORY_CLOCK_RATE: 1512
Max Memory bandwidth: 967680000000 bytes (967.7 GiB)
CU_DEVICE_ATTRIBUTE_ECC_SUPPORT: true
Worker 1:0[1004]: Message: std::setenv successfully set CUDA_VISIBLE_DEVICES to MIG-GPU-9d676d4a-1a5a-e5e0-f4a0-ab9bc8d406e5/1/0
DCGM CudaContext Init completed successfully.
```

1. 开启另一个控制台中，使用命令查看由 DCGM 作为 CUDA 工作负载在 GPU 上运行，可见大多数指标已经非 0。 通过输出可以得出如下的一些结论：

   1) GPU 的计算能力为 8.0，属于 Ampere 架构（如 NVIDIA A100）。
   2) GPU 支持 ECC 内存，适合高可靠性计算任务。
   3) MIG 模式已启用，当前测试工具正在针对不同的 MIG 实例运行。
   4) CUDA 上下文初始化成功，说明 GPU 可以正常工作。
   5) 全局 GPU 和 MIG 实例的硬件属性被正确识别，说明测试工具能够正常与

GPU 通信。

6) MIG 实例的资源（如 SM 数量、内存带宽）是全局 GPU 的一部分。
7) 全局 GPU 和 MIG 实例的硬件属性被正确识别。
8) MIG 模式下的 GPU 分区可以独立运行任务，且资源隔离良好。

(base) olivia@a100-48:~/llm$ sudo dcgmi dmon -i i:7,i:8 -e 1001,1002,1003,1004,1005,1006,1007,1008,1009,1010 -c 10

```
(base) olivia@a100-48:~/llm/deepseek$ sudo  dcgmi dmon -i i:7,i:8 -e 1001,1002,1003,1004,1005,1006,1007,1008,1009,1010 -c  10
#Entity   GRACT        SMACT        SMOCC        TENSO        DRAMA        FP64A        FP32A        FP16A        PCITX
     PCIRX
ID

GPU-I 7   1.000        0.992        0.932        0.000        0.763        0.000        0.000        0.000        N/A
     N/A
GPU-I 8   1.000        0.992        0.920        0.000        0.742        0.000        0.000        0.000        N/A
     N/A
GPU-I 7   1.000        0.992        0.932        0.000        0.751        0.000        0.000        0.000        N/A
     N/A
GPU-I 8   1.000        0.991        0.919        0.000        0.747        0.000        0.000        0.000        N/A
     N/A
GPU-I 7   1.000        0.992        0.932        0.000        0.751        0.000        0.000        0.000        N/A
     N/A
GPU-I 8   1.000        0.991        0.919        0.000        0.747        0.000        0.000        0.000        N/A
     N/A
GPU-I 7   0.998        0.991        0.957        0.000        0.750        0.431        0.000        0.000        N/A
     N/A
GPU-I 8   0.998        0.991        0.950        0.000        0.747        0.434        0.000        0.000        N/A
     N/A
GPU-I 7   0.995        0.995        0.993        0.000        0.000        0.994        0.000        0.000        N/A
     N/A
GPU-I 8   0.995        0.994        0.992        0.000        0.000        0.994        0.000        0.000        N/A
     N/A
GPU-I 7   0.995        0.996        0.994        0.000        0.000        0.995        0.000        0.000        N/A
     N/A
GPU-I 8   0.995        0.994        0.993        0.000        0.000        0.994        0.000        0.000        N/A
     N/A
GPU-I 7   0.995        0.995        0.994        0.000        0.000        0.995        0.000        0.000        N/A
     N/A
GPU-I 8   0.995        0.995        0.994        0.000        0.000        0.995        0.000        0.000        N/A
     N/A
```

Metric 详细说明请参考:
https://docs.nvidia.com/datacenter/dcgm/latest/user-guide/feature-overview.html#id5

设备级别 GPU 指标

度量	定义	DCGM 字段名称（DCGM_FI_*）和 ID
图形引擎活动	图形或计算引擎的任何部分处于活动状态的时间分数。 如果绑定了图形/计算上下文并且图形/计算管道繁忙，则图形引擎处于活动状态。 该值表示某个时间间隔内的平均值，而不是瞬时值。	PROF_GR_ENGINE_ACTIVE（ID： 1001）
SM 活动	至少一个 warp 在多处理器上处于活动状态的时间分数，是所有多处理器的平均值。 请注意，“active”并不一定意味着 warp 正在积极计算。例如，等待内存的 warps 请求被视为活动请求。该值表示某个时间间隔内的平均值，而不是瞬时 价值。值 0.8 或更大是有效使用 GPU 所必需的，但还不够。小于 0.5 可能表示 GPU 使用无效。 给定一个简化的 GPU 架构视图，如果一个 GPU 有 N 个 SM，那么一个使用 N 个块的内核运行在整个 时间间隔将对应于 1 （100%） 的活动。使用 N/5 个块的内核，在整个时间内运行 interval 将对应于 0.2 （20%） 的活动。使用 N 个块的内核，运行时间超过 1/5 interval 的 samp;在 SM 处于空闲状态的情况下，activity 也将为 0.2 （20%）。该值对数字不敏感 每个块的线程数（请参阅）。 DCGM_FI_PROF_SM_OCCUPANCY	PROF_SM_ACTIVE （ID： 1002）
SM 入住率	多处理器上常驻 warp 的分数，相对于 上支持的最大并发 warp 数 一个多处理器。该值表示某个时间间隔内的平均值，而不是瞬时值。更高的入住率 并不一定表示 GPU 使用率更高。对于 GPU 内存带宽受限的工作负载（请参阅）， 较高的占用率表示 GPU 的使用效率更高。但是，如果工作负载受计算限制（即不是 GPU 内存带宽或延迟受限），则较高的占用率不一定与更有效的 GPU 使用相关。 DCGM_FI_PROF_DRAM_ACTIVE 计算占用率并不简单，它取决于 GPU 属性、每个块的线程数、 registers per thread 和 shared memory per block.使用 CUDA 占用计算器探索各种占用场景。	PROF_SM_OCCUPANCY （ID： 1003）
Tensor 活动	张量 （HMMA / IMMA） 管道处于活动状态的周期分数。该值表示一段时间内的平均值 并且不是瞬时值。值越高，表示 Tensor Core 的利用率越高。活动 1 （100%） 为 相当于在整个时间间隔内每隔一个周期发出一条 Tensor 指令。0.2 （20%） 的活性可以 表示 20% 的 SM 在整个时间段内的利用率为 100%，100% 的 SM 的利用率为 20% 在整个时间段内，100% 的 SM 在 20% 的时间段内处于 100% 利用率，或介于两者之间的任何组合 （参见 以帮助消除这些可能性的歧义）。 DCGM_FI_PROF_SM_ACTIVE	PROF_PIPE_TENSOR_ACTIVE （ID： 1004）
FP64 引擎活动	FP64 （双精度） 管道处于活动状态的循环次数。该值表示一段时间内的平均值 并且不是瞬时值。值越高，表示 FP64 内核的利用率越高。活动 1 （100%） 为 相当于在整个时间间隔内，Volta 上每四个周期在每个 SM 上发出一条 FP64 指令。活动 0.2 （20%） 可能表示 20% 的 SM 在整个时间内处于 100% 的利用率 期间，100% 的 SM 在整个时间段内的利用率为 20%，100% 的 SM 在 20% 的时间内处于 100% 的利用率 句号，或两者之间的任何组合（参见 DCGM_FI_PROF_SM_ACTIVE 以帮助消除这些可能性的歧义）。	PROF_PIPE_FP64_ACTIVE （ID： 1006）
FP32 引擎活动	FMA（FP32（单精度和整数））管道处于活动状态的循环分数。该值表示一段时间内的平均值 interval 的 SET 而不是 instantaneous 值。值越高，表示 FP32 内核的利用率越高。活动 1 （100%） 为 相当于在整个时间间隔内每隔一个周期执行一次 FP32 指令。0.2 （20%） 的活性可能表示 20% 的 SM 在整个时间段内的利用率为 100%，100% 的 SM 在整个时间段内的利用率为 20%，100% 的 SM 在 20% 的时间段内处于 100% 利用率，或者两者之间的任何组合（请参阅帮助 消除这些可能性的歧义）。 DCGM_FI_PROF_SM_ACTIVE	PROF_PIPE_FP32_ACTIVE （ID： 1007）
FP16 引擎活动	FP16 （半精度） 管道处于活动状态的循环次数。该值表示一段时间内的平均值，并且 而不是瞬时值。值越高，表示 FP16 内核的利用率越高。活动 1 （100%） 是等效的 到 FP16 指令。0.2 （20%） 的活性可能表示 20% 的 SM 在整个时间段内处于 100% 的利用率，100% 的 SM 在整个时间段内处于 20% 的利用率，100% 的 SM 在 20% 的时间段内处于 100% 利用率，或者两者之间的任何组合（请参阅帮助 消除这些可能性的歧义）。 DCGM_FI_PROF_SM_ACTIVE	PROF_PIPE_FP16_ACTIVE （ID： 1008）
内存 BW 利用率	向设备内存发送数据或从设备内存接收数据的周期数。该值表示一段时间内的平均值 并且不是瞬时值。 值越高表示设备内存的利用率越高。活动 1 （100%） 是等效的 发送到 DRAM 指令（实际上，~0.8 （80%） 的峰值是可实现的最大值）。 活动 0.2 （20%） 表示在时间间隔内有 20% 的周期正在读取或写入设备内存。	PROF_DRAM_ACTIVE （ID： 1005）

Reference:
https://docs.nvidia.com/datacenter/dcgm/latest/user-guide/feature-overview.html#cuda-test-generator-dcgmproftester

四、 模型训练和推理的 MIG 测试

通过改变 GI 的切片大小来改变算力，并测试训练作业端到端的时间变化。训练包含 train，test 两个步骤，时间变化主要计算 train 的时间变

化。

规格	训练时间	时间变化(整卡/MIG)
整卡 A100（7c）	train: 95s test: 12s	1

另外，由于 128 的 batch_size 内存消耗比较大，所以切分的时候，显存都是采用了 40gb 的规格。为了避免 OOM， 建议 batch_size 设置 128 以下。

1. **测试案例**

基于如下代码进行 MIG 测试，算法参数如下：

swin-transformer：
DATA: Imagenet-mini （276M）
configs: swin_large_patch4_window7_224_22k.yaml
/swin_base_patch4_window7_224/default
data-size: 128
epoch: 1

#启动方式:
CUDA_VISIBLE_DEVICES=<GPU UUID> python -m torch.distributed.launch\
--nproc_per_node 1 --master_port 25566 main.py \
--cfg configs/swin/swin_base_patch4_window7_224.yaml \
--data-path ./Imagenet \
--batch-size 128

代码如下:
microsoft/Swin-Transformer: This is an official implementation for "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows".

运行结果:

4/7 算力：MIG 4g.40gb	train:157s test: 14s	实际值：0.605 理论值：0.571
3/7 算力：MIG 3g.40gb	train:189s test: 18s	实际值：0.502 理论值：0.428

以上为模型训练的时间。说明：

1. 实际值比理论值要大，可能是训练对 core 的使用没有达到极限，所以往小调节算力的时候时间增加的量比理论值少。
2. 算力降低后，时间变化的比例基本上与算力缩减的比例相近。