GPUs at TeideHPC

GPU is an acronym for *Graphics Processing Unit* and represents precisely the heart of a graphics card just like the CPU does in a PC. Apart from the heart, it is also your brain, since it is in charge of carrying out all the complex calculations that allow some programs to run much faster than on a CPU.

Among the main uses of GPUs are the following:

- Video edition
- 3D graphics rendering
- Automatic learning
- Scientific applications
- etc...

The TeideHPC cluster offers 2 different GPU models to use with your jobs. We recommend taking a look at the cluster description to get an idea of what it looks like.

GPUs models available.

These are the GPUs currently available at TeideHPC:

GPU model	**# of nodes**	**# of GPUs/ node**	**Slurm type specifier**	**CPU cores/ node**	**CPU memory/ node**	**Compute Capability (*)**	**GPU mem (GiB)**
Nvidia A100	16	4	a100	64	256GB	80	40 GB
Nvidia A100	1	8	a100	64	512GB	80	40 GB
Nvidia Tesla T4	4	1	t4	32	256GB	75	16 GB

(*) *Compute Capability* is a technical term created by NVIDIA as a compact way to describe what hardware functions are available on some models of GPU and not

on others. It is not a measure of performance and is relevant only if you are compiling your own GPU programs. See the page on CUDA programming for more.

What is MIG? *(NVIDIA Multi-Instance GPU)*

Multi-Instance GPU (MIG) is a technology from NVIDIA that allow **divide the GPU** into up to seven fully isolated instances with their own high-bandwidth memory, cache, and processing cores.

Which is the motivation to use MIG?.

Without MIG, different jobs running on the same GPU, like different AI inference requests, compete for the same resources. A job consuming larger memory bandwidth deprives others of it, causing multiple jobs to miss their latency targets.

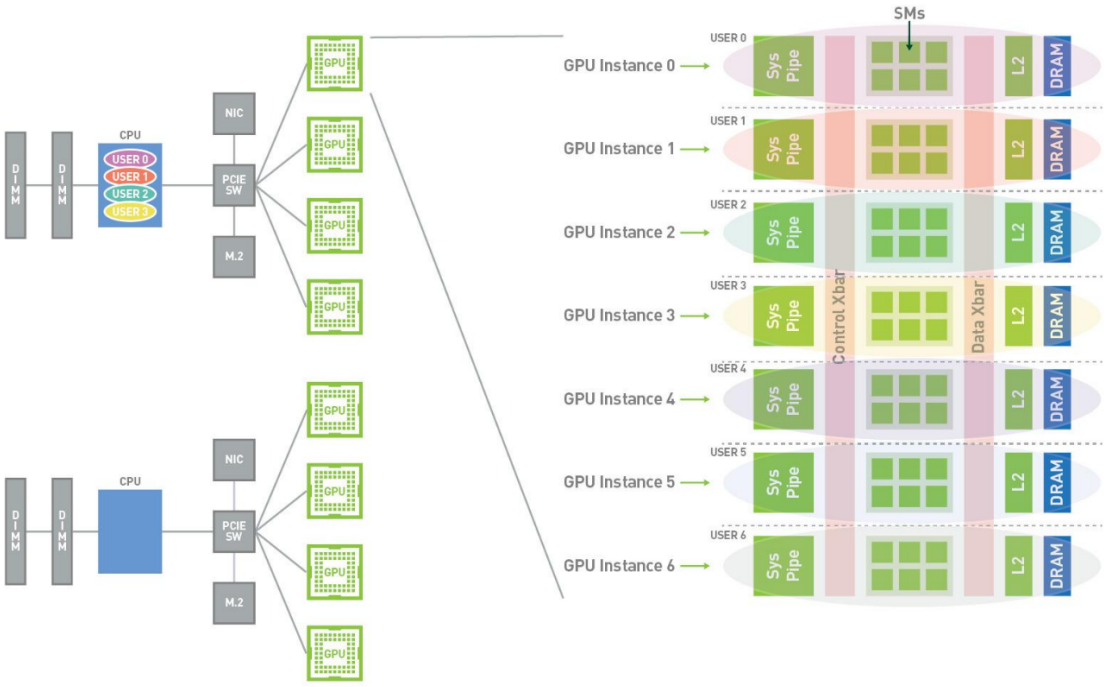
With MIG, jobs run concurrently on different instances, each with dedicated resources for compute, memory, and memory bandwidth usage, resulting in predictable performance with QoS and maximum GPU utilization.

In short:

- A100 NVIDIA GPUs are currently the most powerful GPUs the money can buy.
- Their memory ranges from 40 GB to 80GB per card.
- **Not many applications can take advantage of the full power of these cards.**
- Unfortunately once **SLURM allocates one GPU no other job can make use of the GPU**.

> **Jobs are GPU exclusive**
>
> Even though there are other approaches such as NVIDIA MPS, but unfortunately SLURM can only useone GPU with MPS per node.

- Basically It allows us to physically partition the GPUs so more than one job can make use of a GPU.



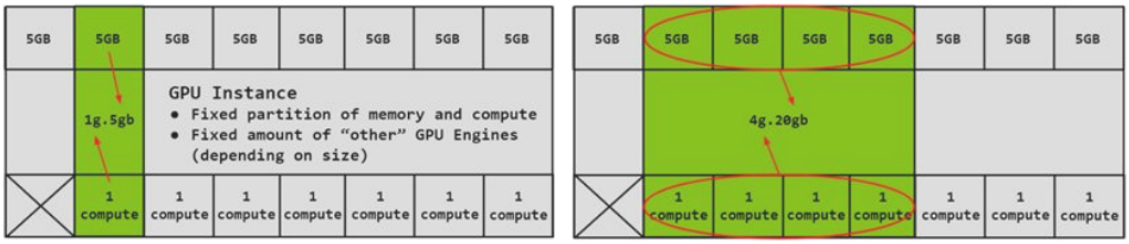
GPU partitions for a Nvidia A100 (40GB).

A GPU can be partitioned into MIG instance partitions of different sizes. For example, on a 40 GB NVIDIA A100, an administrator could create two instances with 20 gigabytes (GB) of memory each, three instances with 10 GB, or seven instances with 5 GB. Or a combination.

Not every partition is possible. There are restrictions. These are the posibilities for each card.

Config	GPC Slice #0	GPC Slice #1	GPC Slice #2	GPC Slice #3	GPC Slice #4	GPC Slice #5	GPC Slice #6
1	7						
2	4				2		1
3	4				1	1	1
4	3			3			
5	3			2		1	
6	3			1	1	1	
7	2		2		3		
8	2		1	1	3		
9	1	1	2		3		
10	1	1	1	1	3		
11	2		2		2		1
12	2		1	1	2		1
13	1	1	2		2		1
14	2		1	1	1	1	1
15	1	1	2		1	1	1
16	1	1	1	1	2		1
17	1	1	1	1	1	2	
18	1	1	1	1	1	1	1

Here you can see 2 examples of partitioning for a 40GB A100 GPU.



To get the list of features and resources of each node and mig partition you can use this command.

```
sinfo -o "%40N  %10c  %10m  %50f  %10G "
```

```
NODELIST                                CPUS       MEMORY     AVAIL_FEATURES
GRES
node18109-1                             64         257214     ilk,gpu,a100                    gpu:a100:8
node2204-[3-4]                          20         31906      ivy                             (null)
node17109-1,node17110-1,node18110-1,node  64         257214     ilk,viz,t4
gpu:t4:1
node0303-2,node0304-[1-4],node1301-[1-4]  16         30000+     sandy
(null)
node17101-1,node17103-1,node17104-1,node  64         257214
ilk,gpu,a100                    gpu:a100:4(S:0-1)
node17102-1                             64         257214     ilk,gpu,a100,3g.20gb,2g.10gb,1g.5gb
gpu:3g.20gb:1(S:0),gpu:2g.10gb
```

• for a specific node with mig partitions:

```
scontrol show node node17102-1
```

```
NodeName=node17102-1 Arch=x86_64 CoresPerSocket=32
   CPUAlloc=8 CPUEfctv=64 CPUTot=64 CPULoad=0.00
   AvailableFeatures=ilk,gpu,a100,3g.20gb,2g.10gb,1g.5gb
   ActiveFeatures=ilk,gpu,a100,3g.20gb,2g.10gb,1g.5gb
   Gres=gpu:3g.20gb:1(S:0),gpu:2g.10gb:1(S:0),gpu:a100:3(S:0-1),gpu:1g.5gb:2(S:0)
   NodeAddr=node17102-1 NodeHostName=node17102-1 Version=22.05.8
   OS=Linux 4.18.0-372.9.1.el8.x86_64 #1 SMP Tue May 10 14:48:47 UTC 2022
   RealMemory=257214 AllocMem=0 FreeMem=142408 Sockets=2 Boards=1
   State=MIXED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=main,batch,express,long
   BootTime=2023-07-24T08:38:22 SlurmdStartTime=2023-07-24T09:20:41
   LastBusyTime=2023-08-03T09:07:43
   CfgTRES=cpu=64,mem=257214M,billing=64,gres/gpu=7,gres/gpu:1g.5gb=2,gres/gpu:
2g.10gb=1,gres/gpu:3g.20gb=1,gres/gpu:a100=3
   AllocTRES=cpu=8,gres/gpu=2,gres/gpu:a100=2
   CapWatts=n/a
```

```
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
```

- One node without mig partitions:

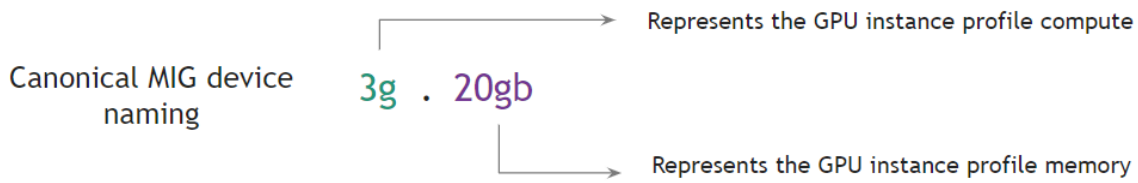
```
scontrol show node node17101-1
```

```
NodeName=node17101-1 Arch=x86_64 CoresPerSocket=32
   CPUAlloc=8 CPUEfctv=64 CPUTot=64 CPULoad=0.00
   AvailableFeatures=ilk,gpu,a100
   ActiveFeatures=ilk,gpu,a100
   Gres=gpu:a100:4(S:0-1)
   NodeAddr=node17101-1 NodeHostName=node17101-1 Version=22.05.8
   OS=Linux 4.18.0-372.9.1.el8.x86_64 #1 SMP Tue May 10 14:48:47 UTC 2022
   RealMemory=257214 AllocMem=0 FreeMem=119559 Sockets=2 Boards=1
   State=MIXED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
   Partitions=main,batch,express,long
   BootTime=2023-07-17T13:29:12 SlurmdStartTime=2023-07-21T12:42:08
   LastBusyTime=2023-08-03T11:06:24
   CfgTRES=cpu=64,mem=257214M,billing=64,gres/gpu=4,gres/gpu:a100=4
   AllocTRES=cpu=8,gres/gpu=4,gres/gpu:a100=4
   CapWatts=n/a
   CurrentWatts=0 AveWatts=0
   ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
```

MIG Device Names

By default, a MIG device consists of a single *“GPU Instance”* and a single *“Compute Instance”*. The table below highlights a naming convention to refer to a MIG device by its GPU Instance's compute slice count and its total memory in GB (rather than just its memory slice count).

When only a single CI is created (that consumes the entire compute capacity of the GI), then the CI sizing is implied in the device name.



The description below shows the profile names on the A100-SXM4-40GB. These are the device name when using single CI.

Memory	**20gb**	**10gb**	**5gb**
GPU Instance	3g	2g	1g
Compute Instance	3c	2c	1c
MIG Device	3g.20gb	2g.10gb	1g.5gb
	GPCGPCGPC	GPCGPC	GPC

In next section you can see how to get a GPU node to execute with slurm. Also, we recommend to visit the request GPU and compute page.