One GPU, Many Models

What works and what Segfaults

Yash Panchal

SDET III @ Percona

Overview

1. Why Partition a GPU ?
2. GPU sharing methods
3. Video Generation use case (Wan2.2 based models)
4. Potential Crashes and Failures
5. Workload strategies
6. Wan2.2 workload H100 vs B200
7. Conclusion

Why Partition a GPU ?

1. Not all models support batching.
2. Application requirement.
3. Want to sell Compute.
4. Simply cannot afford another GPU

GPU Sharing

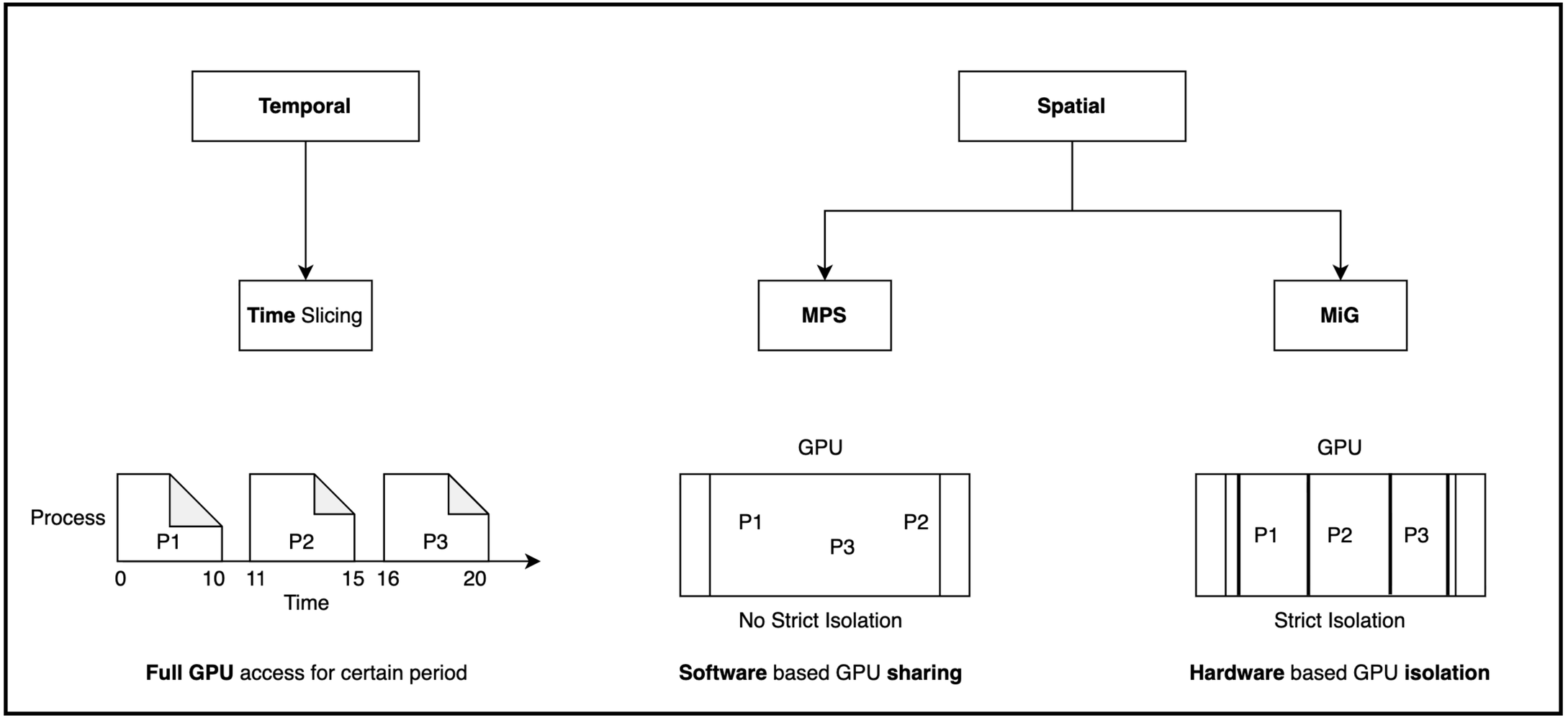
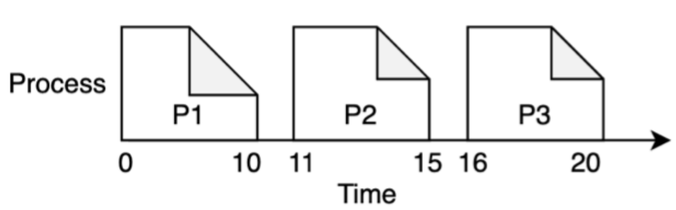
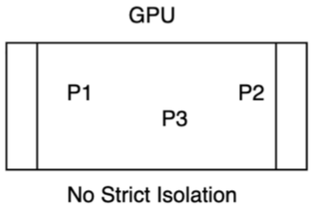
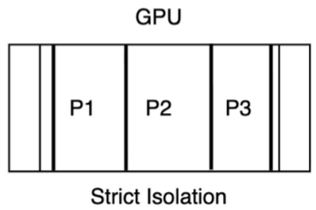


Figure A: GPU Sharing diagram

GPU sharing comparison

Time Slicing	MPS	MIG
		
Full GPU availability	GPU shared between processes	GPU split into isolated portions
High context switch overhead	Low context switch overhead	No context switch overhead
Time sensitive	Resource Sensitive	Fixed resource
Single workload at a time	48 clients/processes are supported	Max 7 fixed sized partitions
Good for workloads that can wait	Trusted workloads that can run together	Workloads that strictly require isolation
Full GPU allocation (QoS maintained)	No QoS Guaranteed	QoS Guaranteed

GPU Support

	MPS	MIG
Enterprise GPUs A100 and +	Yes	Yes
Professional GPUs	Yes	Few Ampere and Blackwell based
Consumer GPUs	Yes	No

MPS Overview

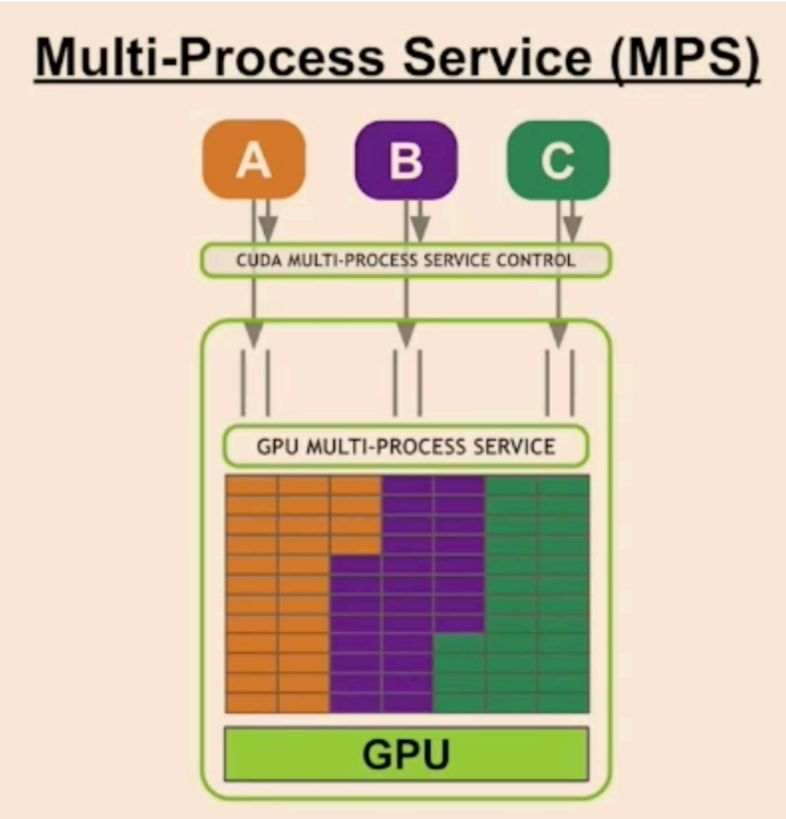


Figure B: MPS diagram

- Software level isolation (client/server based) CUDA API implementation
- Compute and Memory are shared concurrently.

MPS Components:

1. Client: Any CUDA process is a client
2. Control Daemon: Manages MPS server
3. MPS Server: Shares GPU connections with clients

MPS Setup steps

1. export CUDA_VISIBLE_DEVICES=0 **Select the GPU device**

2. nvidia-cuda-mps-control -d **Start the MPS daemon**

MPS client runtime is built into the CUDA Driver library

All the subsequent CUDA work uses the MPS server automatically :)

MiG Overview

Figure C: GPU instances

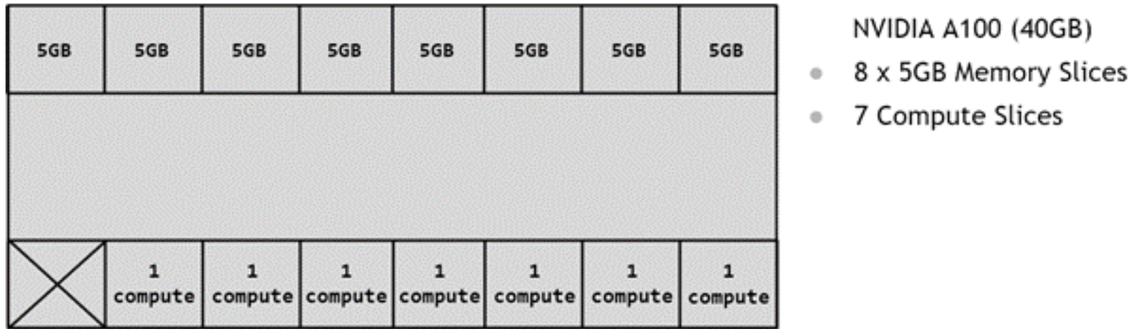


Figure D: GPU slices

- GPU Instance = GPU slices + GPU engines (CE, DEC, JPEG, ENC, OFA)
- GPU slice = GPU Memory slice + GPU Compute slice
- 1 GPU memory slice ~ 1/8 Total GPU Memory
- 1 GPU compute slice ~ 1/7 Total number of compute (SMs)

MiG Slice hierarchy

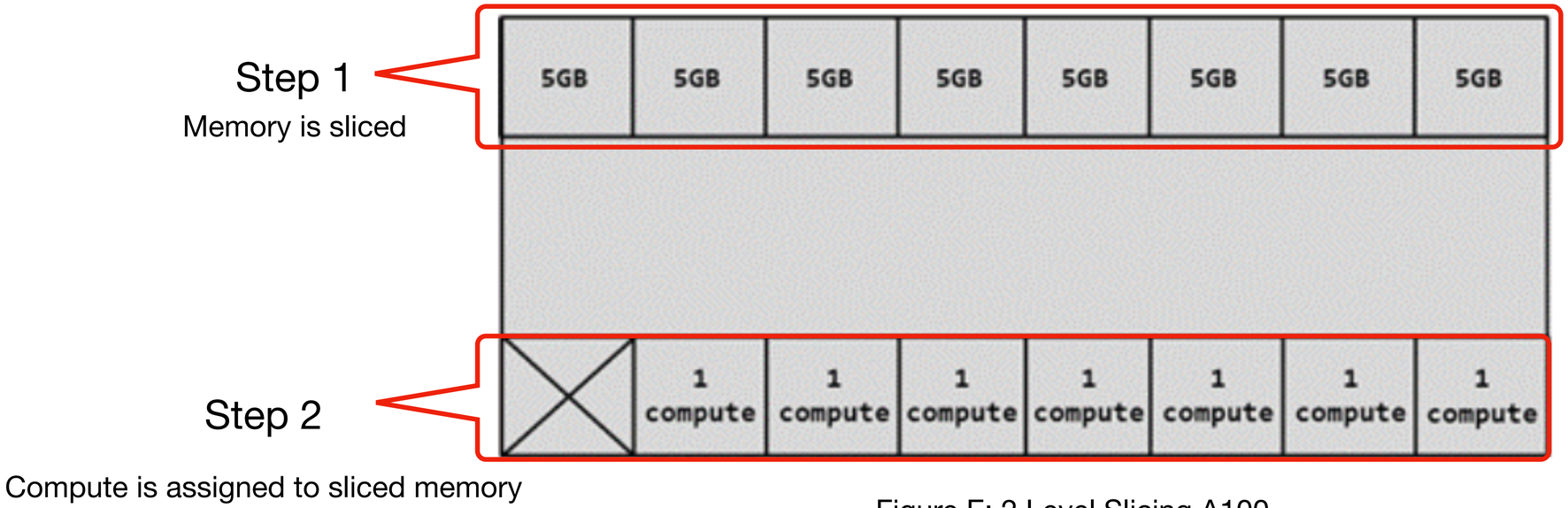


Figure E: 2 Level Slicing A100

- MiG instance creation has has two level hierarchy
- You cannot create compute slice first and then assign memory slice to it.

MiG Partitioning Combinations

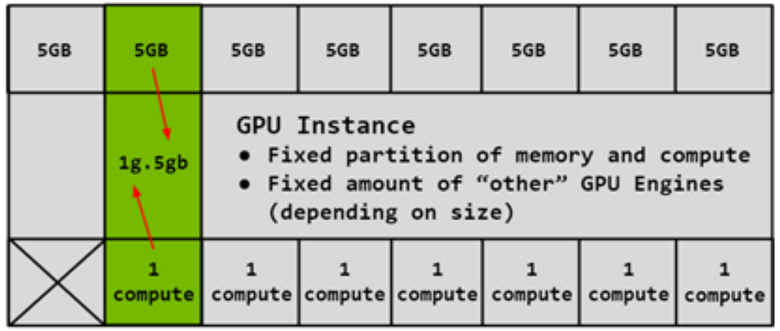


Figure G: Smallest Instance

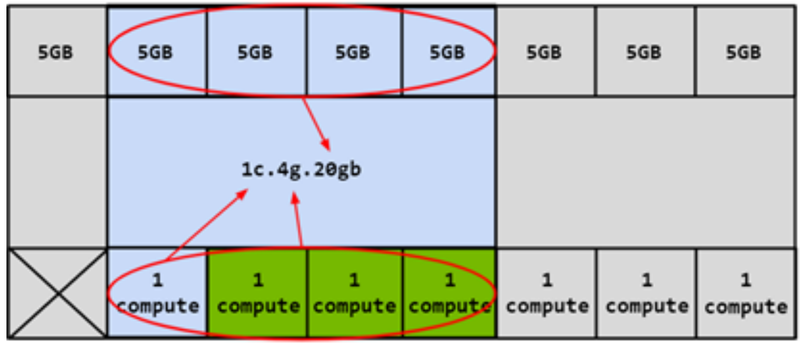


Figure H: isolated compute with shared Memory

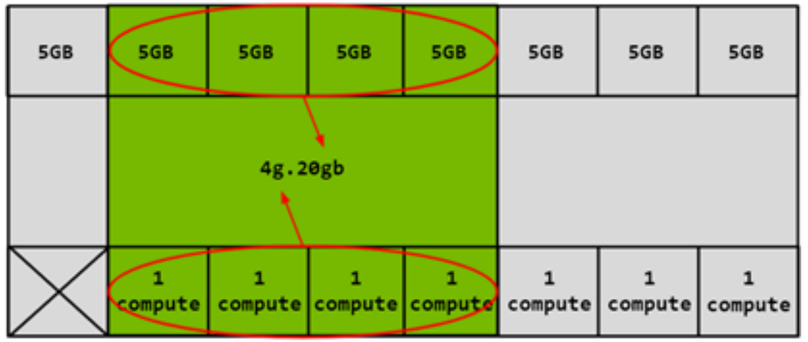


Figure I: Multiple compute with Large Memory

Single isolated compute instance

- 1 isolated 1g.5gb instance
- Full isolation of compute and memory
- Size might be an issue

Multiple isolated compute instances

- Single 4g.20gb split into 4 isolated 1c.4g.20gb instances
- The 20gb memory is shared by 4 instances
- Memory issues (potential OOM)

Single isolated large instance

- 1 isolated 4g.20gb instance
- The 20gb memory is shared by single instance
- Potential of unused compute (Idle compute)

MIG Overhead

```
root@h100-server1:~# nvidia-smi mig -i 0 -lgip
+-----------------------------------------------------------------------------+
| GPU instance profiles:                                                      |
| GPU   Name               ID    Instances   Memory     P2P    SM    DEC   ENC  |
|                               Free/Total   GiB              CE    JPEG  OFA  |
|=============================================================================|
|   0  MIG 1g.10gb         19     7/7         9.75       No     16     1     0   |
|                                                                  1     1     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 1g.10gb+me      20     1/1         9.75       No     16     1     0   |
|                                                                  1     1     1   |
+-----------------------------------------------------------------------------+
|   0  MIG 1g.20gb         15     4/4         19.62      No     26     1     0   |
|                                                                  1     1     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 2g.20gb         14     3/3         19.62      No     32     2     0   |
|                                                                  2     2     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 3g.40gb          9     2/2         39.50      No     60     3     0   |
|                                                                  3     3     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 4g.40gb          5     1/1         39.50      No     64     4     0   |
|                                                                  4     4     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 7g.80gb          0     1/1         79.25      No     132    7     0   |
|                                                                  8     7     1   |
+-----------------------------------------------------------------------------+
```

Figure J: H100 MiG available profiles

- No Free Lunch: There will always be some overhead.
- You will compromise for utilizing MiG feature
- SM CE and Memory GiB are not exactly partitioned
- 1g = 16 SMs (base)
- 3g = 60 SMs (3 x base + few additional SMs)
- 7g = 132 SMs (all SMs)

MiG Partition Creation

```
root@h100-server1:~# nvidia-smi -L
GPU 0: NVIDIA H100 80GB HBM3 (UUID: GPU-f0adccc6-9d17-0af6-ba0b-82402cce84b5)
```

Figure K : List available GPU devices (Only one device)

```
root@h100-server1:~# nvidia-smi mig -i 0 -lgip
+-----------------------------------------------------------------------------+
| GPU instance profiles:                                                      |
| GPU   Name               ID    Instances   Memory     P2P    SM    DEC   ENC  |
|                               Free/Total   GiB              CE    JPEG  OFA  |
|=============================================================================|
|   0  MIG 1g.10gb         19     0/7        9.75       No     16     1     0   |
|                                                               1     1     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 1g.10gb+me      20     0/1        9.75       No     16     1     0   |
|                                                               1     1     1   |
+-----------------------------------------------------------------------------+
|   0  MIG 1g.20gb         15     0/4        19.62      No     26     1     0   |
|                                                               1     1     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 2g.20gb         14     0/3        19.62      No     32     2     0   |
|                                                               2     2     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 3g.40gb          9     0/2        39.50      No     60     3     0   |
|                                                               3     3     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 4g.40gb          5     0/1        39.50      No     64     4     0   |
|                                                               4     4     0   |
+-----------------------------------------------------------------------------+
|   0  MIG 7g.80gb          0     0/1        79.25      No     132    7     0   |
|                                                               8     7     1   |
+-----------------------------------------------------------------------------+
```

Figure L : List available MIG gpu instance profiles on a H100 GPU

Steps to create MIG Profiles

1. nvidia-smi -i 0 -mig 1 (Enable MiG Mode for GPU 0)
2. nvidia-smi mig -i 0 -cgi 9,9 (GPU instance creation)
3. nvidia-smi mig -i 0 -cci (Compute instance assignment)

```
root@h100-server1:~# nvidia-smi mig -i 0 -lgi
+-------------------------------------------------------+
| GPU instances:                                        |
| GPU   Name               Profile  Instance   Placement  |
|                            ID       ID       Start:Size |
|=======================================================|
|   0  MIG 3g.40gb           9        1          0:4     |
+-------------------------------------------------------+
|   0  MIG 3g.40gb           9        2          4:4     |
+-------------------------------------------------------+
```

Figure M: List created GPU Instances

```
root@h100-server1:~# nvidia-smi mig -i 0 -lci
+-----------------------------------------------------------------------+
| Compute instances:                                                    |
| GPU     GPU       Name             Profile   Instance   Placement  |
|       Instance                       ID        ID       Start:Size |
|         ID                                                          |
|=======================================================================|
|   0      1       MIG 3g.40gb          2         0          0:4     |
+-----------------------------------------------------------------------+
|   0      2       MIG 3g.40gb          2         0          0:4     |
+-----------------------------------------------------------------------+
```

Figure N: List created Compute Instances

Combining MiG and MPS

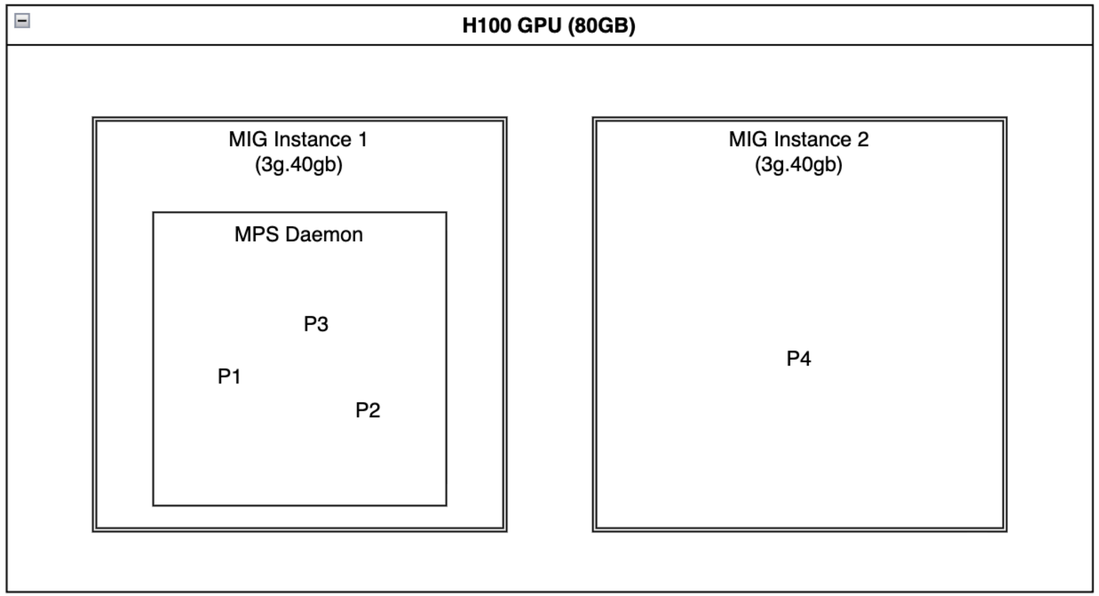


Figure O: H100 with 2 MIG and 1 MIG with MPS

MIG Instance can have MPS within it

MIG instances cannot be created if MPS is already enabled earlier.

(stop the MPS daemon and then MIG instances can be created)

Video Generation

Wan2.2 S2V 14B

- Does not support batching like LLMs
- Large Model 12B Parameter
- prompt + image + audio to generate Video
- 480P
  - ~ 58 GB VRAM (Peak)
  - ~ 55 GB VRAM (Constant)

Wan2.2 TI2V 5B

- Does not support batching like LLMs
- Medium Model 5B Parameter
- prompt to generate Video
- 720P
  - ~ 33 GB VRAM (Peak)
  - ~ 21 GB VRAM (Constant)

OOM: 2x Wan2.2 s2v 14B Model

B

```
Every 1.0s: nvidia-smi

Fri Jan 30 15:23:50 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 580.95.05              Driver Version: 580.95.05      CUDA Version: 13.0     |
+-----------------------------------------+------------------------+----------------------+
| 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 H100 80GB HBM3          On  |   00000000:05:00.0 Off |                  Off |
| N/A   53C    P0            534W /  700W |   55443MiB /  81559MiB |    100%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A           30354      C   nvidia-cuda-mps-server                   66MiB |
|    0   N/A  N/A           34081    M+C   python3                                 55368MiB |
+-----------------------------------------------------------------------------------------+
```

A

```
Every 1.0s: nvidia-smi

Fri Jan 30 15:23:43 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 580.95.05              Driver Version: 580.95.05      CUDA Version: 13.0     |
+-----------------------------------------+------------------------+----------------------+
| 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 H100 80GB HBM3          On  |   00000000:05:00.0 Off |                  Off |
| N/A   48C    P0            616W /  700W |   79612MiB /  81559MiB |    100%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A           30354      C   nvidia-cuda-mps-server                   66MiB |
|    0   N/A  N/A           34048    M+C   python3                                 39768MiB |
|    0   N/A  N/A           34081    M+C   python3                                 39768MiB |
+-----------------------------------------------------------------------------------------+
```

C

```
  File "/Wan2.2/myenv/lib/python3.10/site-packages/torch/nn/modules/module.py", line 928, in _apply
    module._apply(fn)
  [Previous line repeated 2 more times]
  File "/Wan2.2/myenv/lib/python3.10/site-packages/torch/nn/modules/module.py", line 955, in _apply
    param_applied = fn(param)
  File "/Wan2.2/myenv/lib/python3.10/site-packages/torch/nn/modules/module.py", line 1355, in convert
    return t.to(
torch.OutOfMemoryError: CUDA out of memory. Tried to allocate 80.00 MiB. GPU 0 has a total capacity o
f 79.18 GiB of which 41.38 MiB is free. Process 30354 has 66.00 MiB memory in use. Including non-PyTo
rch memory, this process has 40.23 GiB memory in use. Process 34081 has 38.84 GiB memory in use. Of t
he allocated memory 39.60 GiB is allocated by PyTorch, and 18.18 MiB is reserved by PyTorch but unall
ocated. If reserved but unallocated memory is large try setting PYTORCH_CUDA_ALLOC_CONF=expandable_se
gments:True to avoid fragmentation.  See documentation for Memory Management  (https://pytorch.org/do
cs/stable/notes/cuda.html#environment-variables)
(myenv) root@h100-server1:/Wan2.2#
```

D

```
ero_init ue, 'zero_timestep': True, 'enable_motioner': False, 'add_last_motion': True, 'trainable
_token': False, 'enable_tsm': False, 'enable_framepack': True, 'framepack_drop_mode': 'padd', 'audio_
dim': 1024, 'motion_frames': 73, 'cond_dim': 16}, 'drop_first_motion': True, 'sample_shift': 3, 'samp
le_steps': 40, 'sample_guide_scale': 4.5}
[2026-01-30 15:22:35,428] INFO: Input prompt: The Person is speaking professionally while looking at
the viewer. Make sure the face is not distorted or blurred and has realistic and relevant emotions/ex
pressions
[2026-01-30 15:22:35,435] INFO: Input image: f1.jpg
[2026-01-30 15:22:35,435] INFO: Creating WanS2V pipeline.
[2026-01-30 15:23:19,951] INFO: loading ./Wan2.2-S2V-14B/models_t5_umt5-xxl-enc-bf16.pth
[2026-01-30 15:23:28,037] INFO: loading ./Wan2.2-S2V-14B/Wan2.1_VAE.pth
[2026-01-30 15:23:28,360] INFO: Creating WanModel from ./Wan2.2-S2V-14B/
Loading checkpoint shards: 100%|████████████████████████████████| 4/4 [00:04<00:00,  1.16s/it]
[2026-01-30 15:23:33,460] INFO: Generating video ...
  0%|                                                                | 0/10 [00:00<?, ?it/s]
```

Figure P: Running two Large Wan2.2 S2V 14B Model processes for 480P Video Generation on H100 having 80GB VRAM using MPS

Deliberate Failure example: Wan2.2 S2V 14B requires ~ 56GB VRAM for single 480P video generation

OOM: 3x Wan2.2 TI2v 5B Model

ssh

```
Every 1.0s: nvidia-smi                                  h100-server1: Fri Jan 30 18:08:47 2026

Fri Jan 30 18:08:47 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 580.95.05              Driver Version: 580.95.05      CUDA Version: 13.0     |
+-----------------------------------------+------------------------+----------------------+
| 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 H100 80GB HBM3          Off |   00000000:05:00.0 Off |                  Off |
| N/A   66C    P0            668W /  700W |   66770MiB /  81559MiB |     100%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A           72221      C   python3                                22250MiB |
|    0   N/A  N/A           72257      C   python3                                22250MiB |
|    0   N/A  N/A           72374      C   python3                                22250MiB |
+-----------------------------------------------------------------------------------------+
```

Figure Q : 3 x Wan2.2 TI2V 5B processes started at same time

ssh

```
Every 1.0s: nvidia-smi                                  h100-server1: Fri Jan 30 18:09:24 2026

Fri Jan 30 18:09:24 2026
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 580.95.05              Driver Version: 580.95.05      CUDA Version: 13.0     |
+-----------------------------------------+------------------------+----------------------+
| 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 H100 80GB HBM3          Off |   00000000:05:00.0 Off |                  Off |
| N/A   62C    P0            692W /  700W |   78504MiB /  81559MiB |     100%      Default |
|                                         |                        |             Disabled |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|    0   N/A  N/A           72221      C   python3                                33984MiB |
|    0   N/A  N/A           72257      C   python3                                22250MiB |
|    0   N/A  N/A           72374      C   python3                                22250MiB |
+-----------------------------------------------------------------------------------------+
```

Figure R : One peaks at 33GB

3 x Wan2.2 TI2V 5B	MPS 3 process	OOM	1 OOM 2 Survived	5min 45s

ssh

```
    return forward_call(*args, **kwargs)
  File "/Wan2.2/wan/modules/vae2_2.py", line 230, in forward
    x = layer(x, feat_cache[idx])
  File "/Wan2.2/myenv/lib/python3.10/site-packages/torch/nn/modules/module.py", line 1773,
in _wrapped_call_impl
    return self._call_impl(*args, **kwargs)
  File "/Wan2.2/myenv/lib/python3.10/site-packages/torch/nn/modules/module.py", line 1784,
in _call_impl
    return forward_call(*args, **kwargs)
  File "/Wan2.2/wan/modules/vae2_2.py", line 40, in forward
    x = F.pad(x, padding)
  File "/Wan2.2/myenv/lib/python3.10/site-packages/torch/nn/functional.py", line 5290, in p
ad
    return torch._C._nn.pad(input, pad, mode, value)
torch.OutOfMemoryError: CUDA out of memory. Tried to allocate 2.60 GiB. GPU 0 has a total c
apacity of 79.18 GiB of which 2.59 GiB is free. Process 72221 has 33.19 GiB memory in use.
Including non-PyTorch memory, this process has 21.65 GiB memory in use. Process 72374 has 2
1.73 GiB memory in use. Of the allocated memory 16.23 GiB is allocated by PyTorch, and 4.47
 GiB is reserved by PyTorch but unallocated. If reserved but unallocated memory is large tr
y setting PYTORCH_CUDA_ALLOC_CONF=expandable_segments:True to avoid fragmentation.  See doc
umentation for Memory Management  (https://pytorch.org/docs/stable/notes/cuda.html#environm
ent-variables)

real    11m10.203s
user    20m15.296s
sys     5m14.712s
(myenv) root@h100-server1:/Wan2.2#
```

Figure S : PID 72374 peaks before crashing

1 Crashed 2 Survived

Video Generation: Performance Tests

Test	Method	Memory Usage	Result	Time/Video
Baseline Wan2.2 S2V 14B	Full GPU	~ 55 GB Constan ~ 58 GB Peak	Works	4m 5s
Baseline Wan2.2 TI2V 5B	Full GPU	~ 21 GB Constant ~ 33 GB Peak	Works	4m 2s
2 x Wan2.2 s2v 14B	MPS 2 process	OOM	1 OOM 1 Survived	8m 50s
Wan2.2 S2V 14B + Wan2.2 TI2V 5B	MPS 2 process	~ 80 GB	Worked The 14B task finished early	6m 50s for 14B 7m 30s for 5B
3 x Wan2.2 TI2V 5B	MPS 3 process	OOM	1 OOM 2 Survived	5min 45s
2 x Wan2.2 TI2V 5B	MIG 2 Instances	~ 80 GB	Works	6m 50s
3 x Wan2.2 TI2V 5B 1m start delay	MPS 2 -> 3 -> 2 -> 1 Processes	~ 79 GB	Works	3m 20s
2 x Wan2.2 s2v 14B	MIG 2 Large instance	NA	Not possible Model too large for H100 MiG	Use B200 :) 180GB VRAM 3 Videos using MPS 12 mins

Figure P: Running two Large Wan2.2 S2V 14B Model processes for 480P Video Generation on H100 having 80GB VRAM using MPS

MPS staggered start method

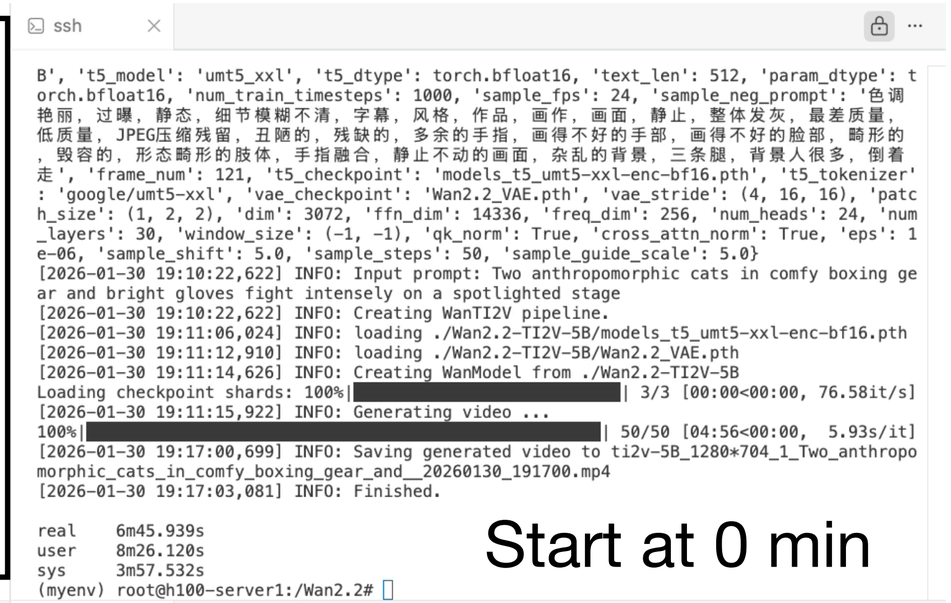
Wan2.2 TI2V 5B predictive workload

(last 30 seconds is where 33GB peak memory utilization occurs)

Add 1 min delay start between each workloads

This took 10m 30s to generate 3 Videos

(This approach is not possible using MiG, max 2 parallel 40GB MiG partitions)



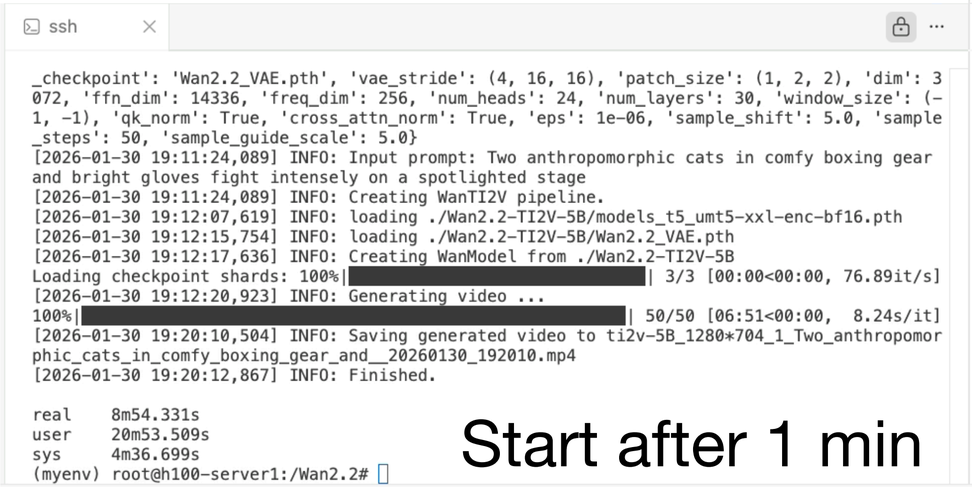


Figure T: Staggered start 3 x Wan2.2 TI2V on H100 GPU

B200 GPU MiG Profiles

```
root@b200-server:~# nvidia-smi mig -i 0 -lgip
+-----------------------------------------------------------------------------+
| GPU instance profiles:                                                      |
| GPU   Name               ID    Instances   Memory     P2P    SM    DEC   ENC  |
|                               Free/Total   GiB              CE    JPEG  OFA  |
|=============================================================================|
|   0  MIG 1g.23gb         19     7/7         20.50      No     18     1     0  |
|                                                                 2     1     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 1g.23gb+me      20     1/1         20.50      No     18     1     0  |
|                                                                 2     1     1  |
+-----------------------------------------------------------------------------+
|   0  MIG 1g.45gb         15     4/4         44.25      No     30     1     0  |
|                                                                 2     1     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 2g.45gb         14     3/3         44.25      No     36     2     0  |
|                                                                 4     2     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 3g.90gb          9     2/2         89.00      No     70     3     0  |
|                                                                 6     3     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 4g.90gb          5     1/1         89.00      No     72     4     0  |
|                                                                 8     4     0  |
+-----------------------------------------------------------------------------+
|   0  MIG 7g.180gb         0     1/1         178.50     No     148    7     0  |
|                                                                 16    7     1  |
+-----------------------------------------------------------------------------+
root@b200-server:~#
```

Figure U: MiG Partitions options on B200

- B200 has 180GB VRAM
- Wan2.2 parallel video generation
  - TI2V 5B:
    - Max 3 relevant MiG Partitions
  - S2V 14B:
    - Max 2 relevant MiG Partitions

MPS on B200 for 3 Wan2.2 S2V 14B

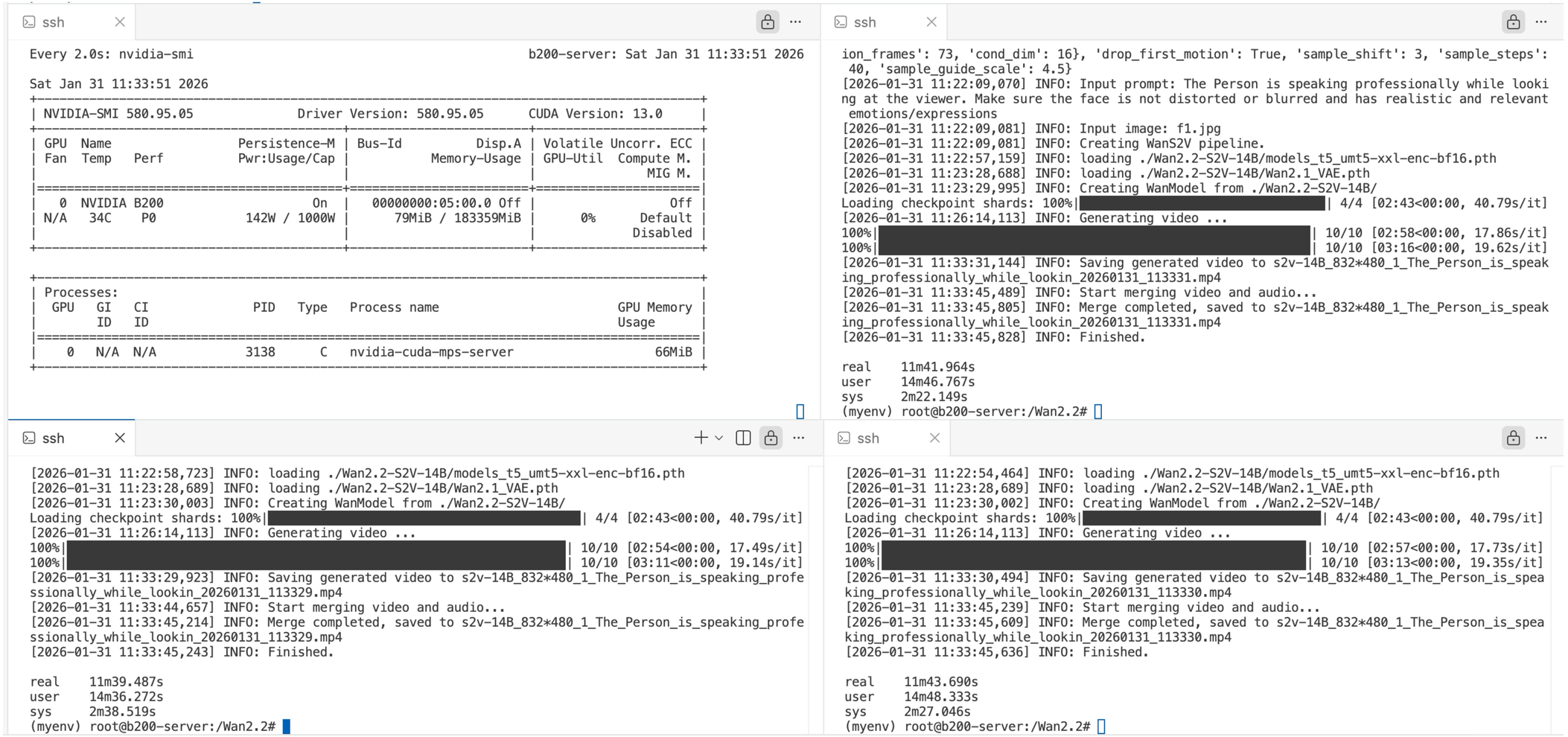


Figure V: MPS start 3 x Wan2.2 S2V 14B on B200 GPU with 180GB

H100 vs B200 for Wan2.2

Max Videos that can be generated in parallel	**TI2V 5B**	**S2V 14B**
H100 using MPS	2 to 3(staggered)	1
B200 using MPS	5	3
H100 using MiG	2	1
B 200 using MiG	3 (faster) to 4	2

Figure W: Max no of video generation using MPS and MiG for two Wan2.2 models

Conclusion

- No single partitioning method is perfect
- MIG is not possible at places but MPS works
- For large video generation models like wan 2.2 S2V 14B MiG use B200
- Identify the behavior of the workload first
- Optimize the model as much as possible

Thank You

MiG specific Talk @ 6:00 PM Today (30 mins): Virtualization and Cloud Infrastructure

GPU monitoring methods Talk @ 9:50 AM Tomorrow (40 mins): Software Performance

Reference

- Images B,C,D,E,F,G,H,I are sourced from NVIDIA Documentations,
- Rest of the images are from CLI and created using draw.io
- Models used for demo are Wan2.2 S2V 14B and Wan2.2 TI2V 5B