



Using GPUs with HTCondor 9.8+/10.x

European HTCondor Workshop 2022

Todd Tannenbaum / John (TJ) Knoeller



How things have worked for a long time...

GPU Support in HTCSS EP

This one line in your config file

```
use feature:gpus
```

tells the EP to:

- Discover GPU devices in the server
- Advertise their capabilities in slot ads
- Assign individual GPU devices to slots
 - Static slots: assign evenly across all slots
 - Partitionable slots: assign GPUs to slots as per job request
- Tell CUDA or OpenCL libs about the GPUs assigned in the slot
- Monitor utilization of GPU cores and GPU memory

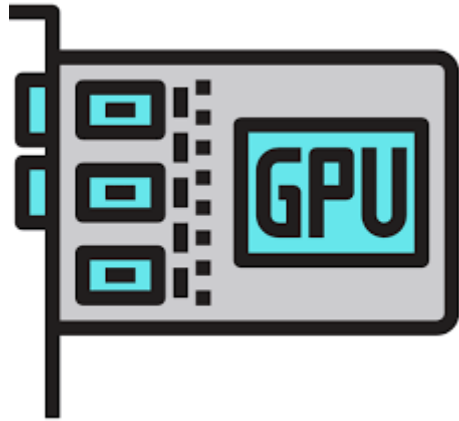
Job Submit File

```
executable = my_cuda_job  
request_cpus = 1  
request_memory = 1GB  
request_gpus = 1  
queue
```

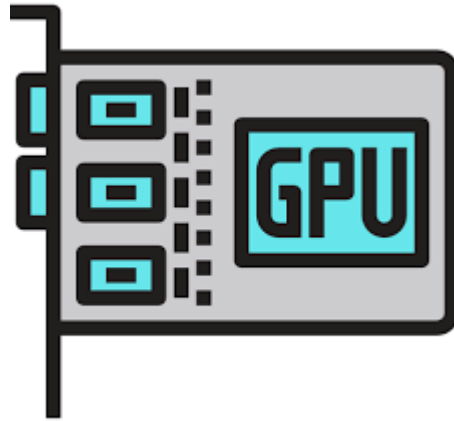
What is new with GPUs in HTCSS 10 ?

**Let's say your server has
multiple GPU devices...**

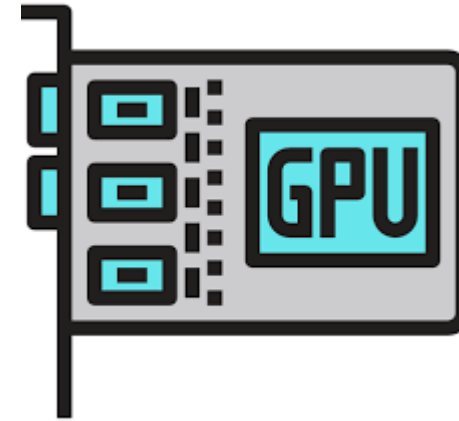
Failures with Multi-GPU servers



GPU0

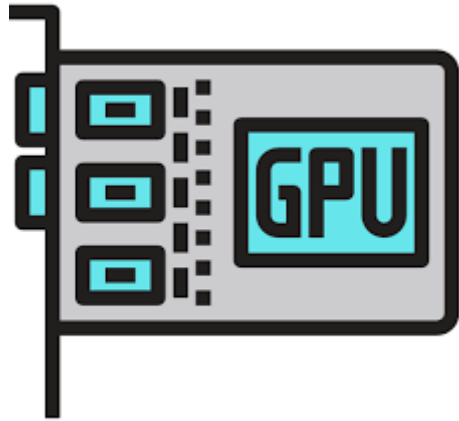


GPU1

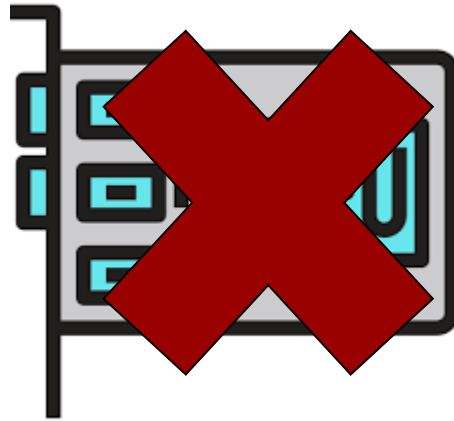


GPU2

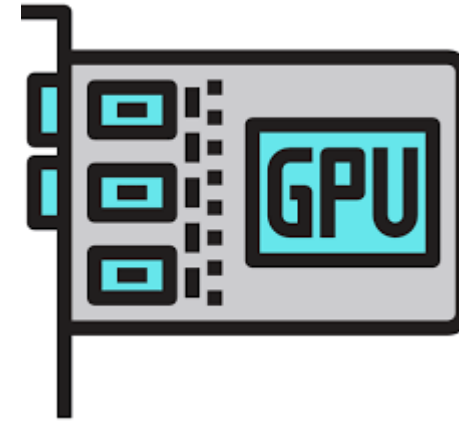
Failures with Multi-GPU servers



GPU0

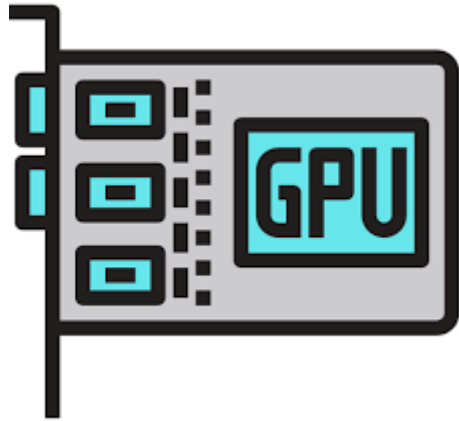


GPU1

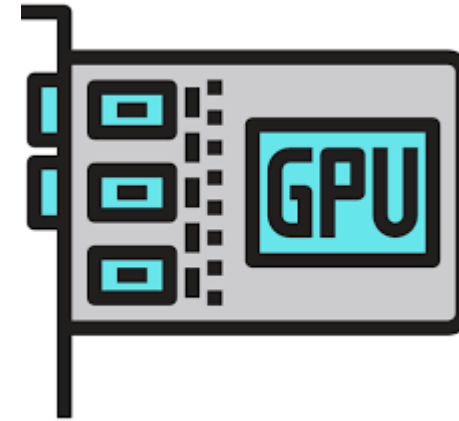
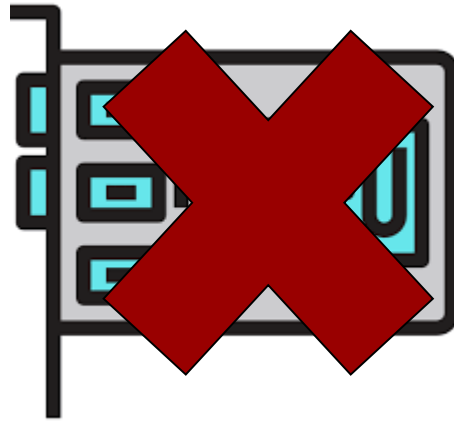


GPU2

Failures with Multi-GPU servers

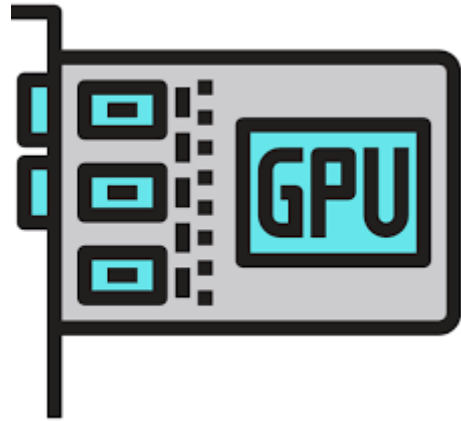


GPU0

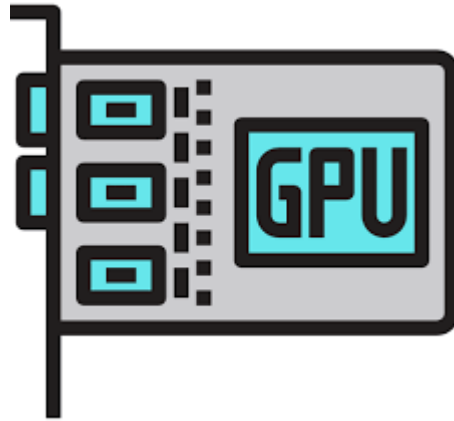


GPU1

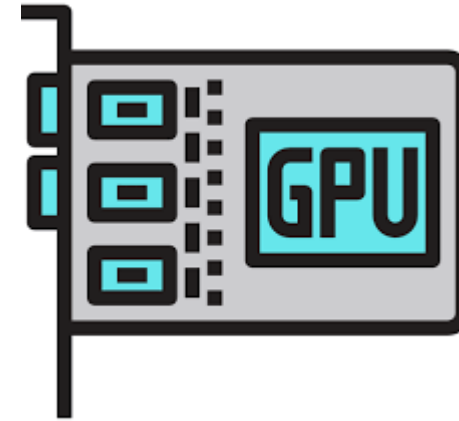
Failures with Multi-GPU servers



GPU0



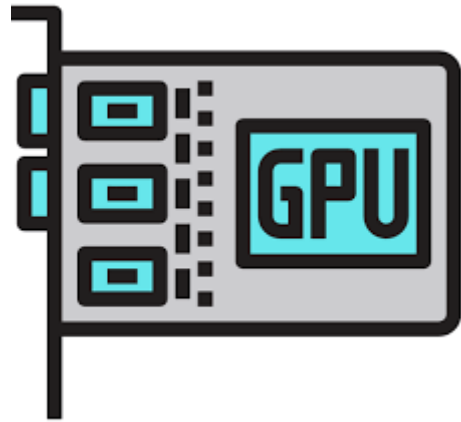
GPU1



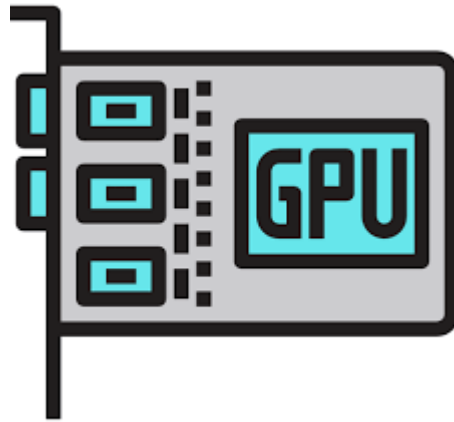
GPU2

***"Index"* identifiers are not stable!**

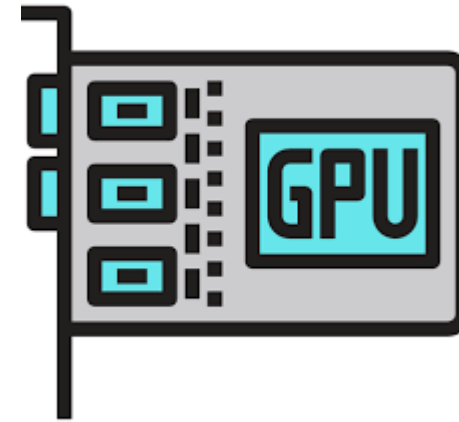
Failures with Multi-GPU servers



GPU-c4a646d7



GPU-8AC64982



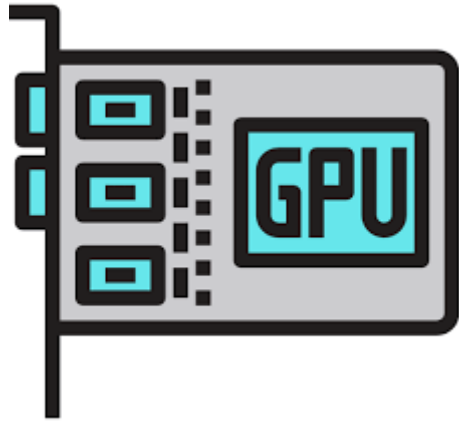
GPU-BB7246CC

...but "*uuid*" identifiers are stable,
and thus used by default in HTCSS 10

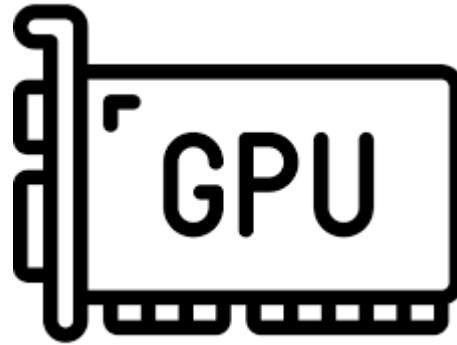
Take a GPU offline

- › Take a GPU offline with a reconfig of the Startd
- › Offline GPU ids are not assigned to new dynamic slots
- › Offline GPUs are not in AvailableGPUs list
- › Requires stable GPU ids to work properly
 - Stable GPU ids are the default (-short-uuid)
 - Works with GPU indexes only if the GPU is not hung

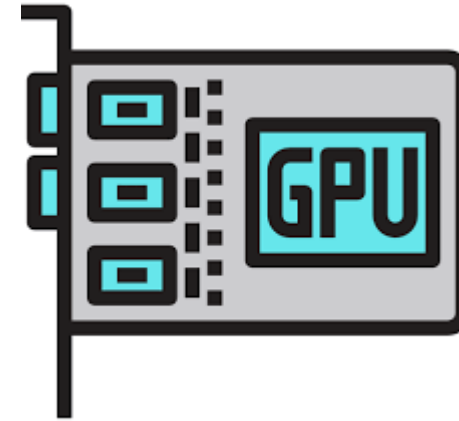
Challenges with *heterogeneous* Multi-GPU servers



Titan RTX

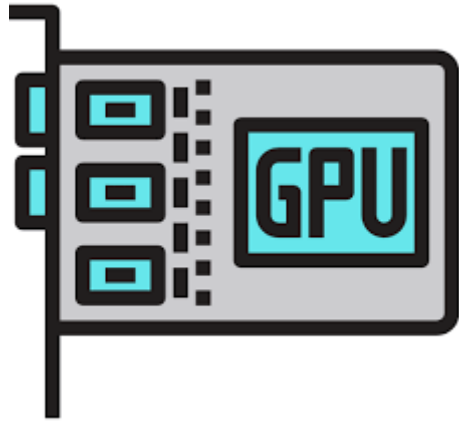


Tesla V100

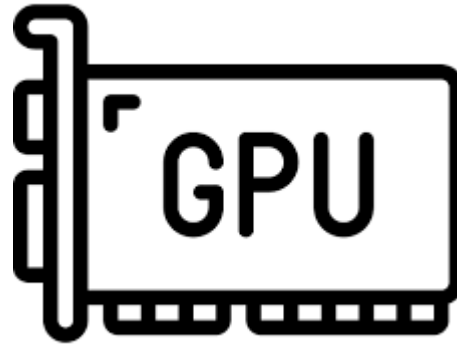


Titan RTX

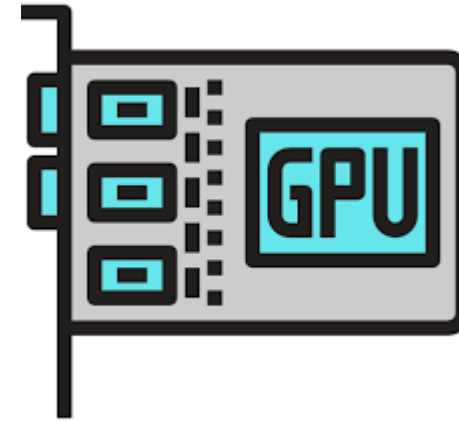
Challenges with *heterogeneous* Multi-GPU servers



TITAN RTX



TESLA V100



TITAN RTX

Should pslot have DeviceName="TITAN RTX" or
DeviceName="TESLA V100" ?

Nested ClassAd attributes to the rescue in HTCSS 9.8+

```
GPUs = 1
TotalGPUs = 3
AssignedGPUs= { GPU_c4a646d7 GPU_6a96bd13 }
AvailableGPUs = { GPU_c4a646d7 }
GPU_6a96bd13 = [
    Capability=7.5;
    Id = "GPU_6a96bd13";
    DeviceName="TITAN RTX";
    DeviceUuid="6a96bd13-70bc-6494-6d62-1b77a9a7f29f";
    GlobalMemoryMb=24220; ]
GPU_c4a646d7 = [
    Capability=7.0;
    DeviceName="TESLA V100";
    ...
```

Job Submit File

```
executable = my_cuda_job  
request_cpus = 1  
request_memory = 1GB  
request_gpus = 1  
require_gpus = DeviceName == "TITAN RTX"  
queue
```


Job Submit File

```
executable = my_cuda_job  
request_cpus = 1  
request_memory = 1GB  
request_gpus = 1  
require_gpus = Capability > 7.0  
queue
```

Targeted binding of GPUs to slots

- › New multi-line STARTD slot resource configuration

```
SLOT_TYPE_1 @=slot1
  CPUs = 50%
  Memory = 50%
  GPUs = 2 : Capability > 7.0
@slot1
```

- › Evaluated against each GPU property ad
 - Affects binding of GPUs into static slots and p-slots

NVIDIA Multi-Instance GPU (MIG) Support

- › Good News: Some NVIDIA GPUs have MIG capability
 - Split a GPU device into up to 7 MIGs
 - Each MIG behaves like a smaller GPU device
 - The GPU can no longer be used directly
 - (MIGs are usually heterogenous, GlobalMemoryMb will vary)
- › Bad News: NVIDIA changed APIs drastically
- › Good News: HTCSS now discovers the MIGs
 - (And hides the MIG parent GPU)
- › Bad News: GPU Utilization monitoring not yet supported for MIG devices

NVIDIA Multi-Instance GPU (MIG) Support, cont

Admin needs to split-up the MIG GPU however they wish before starting the EP

Future work: HTCSS dynamically splits up MIGs ?

MIGs

- › Lots of caveats from NVIDIA
 - Big differences between driver 460 and 470
 - Only one MIG can be used per-process
 - Use long uuid name, (Short uuid name not supported)
 - CUDA device enumeration does not see them
 - GPU forgets MIGs on reboot
- › MIGs "just work" in HTCondor 9.0
 - Must be set up before the Startd runs discovery
 - Restart required to re-run discovery

Speaking of future work...

Multiple Jobs Sharing a GPU

What you can do TODAY: Multiple jobs sharing a GPU (the unsafe way)

- › New arguments to condor_gpu_discovery in 9.0
 - -repeat : duplicate GPU ids
 - -divide : duplicate GPU ids and reduce advertised GPU memory
 - Add to GPU_DISCOVERY_EXTRA config knob

GPU_DISCOVERY_EXTRA = -divide 2

Upcoming - safer GPU sharing

› New slot splitting protocol

- Schedd will request that partitionable slot be split into N slots that share the same GPU
- Schedd will do this only for jobsets that request GPU sharing
 - Share GPUs only with your own jobs

Thank You!



This work is supported by NSF under Cooperative Agreement OAC-2030508 as part of the PATH Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.