OSG Features to Support Machine Learning

Mats Rynge
OSG User Support

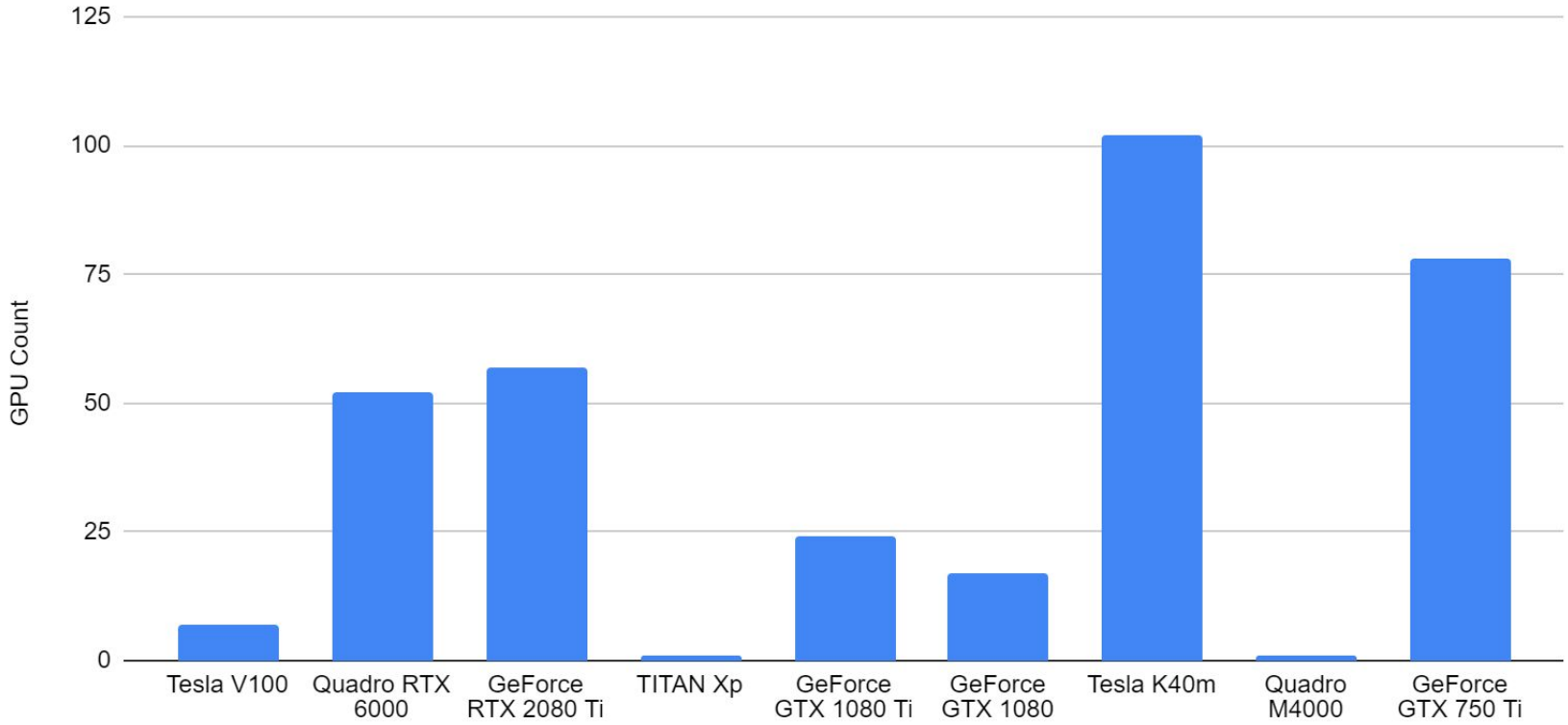
Open Science Grid

OSG All Hands Meeting 2020

Overview

- Many OSG sites now provide access to GPUs
- GPU software stacks are generally more complex, both at system level and user level
- Singularity integration
- Singularity images

GPU Availability (varies over time!)



Count	Resource	GPU	CUDAVersion	CUDACapability
40	CMSHTPC_T3_US_NotreDame_gpu	Quadro RTX 6000	10.2	7.5
6	CMSHTPC_T3_US_NotreDame_gpu	Tesla V100-PCIE-32GB	10.2	7.0
95	FNAL_WILSON	Tesla K40m	10.2	3.5
1	Omaha	GeForce GTX 1060 6GB	10.2	6.1
5	Omaha	Quadro RTX 5000	10.2	7.5
1	Omaha	Quadro RTX 8000	10.2	7.5
3	Omaha	Tesla K20m	10.2	3.5
1	Omaha	Tesla K40m	10.2	3.5
3	Omaha	Tesla P100-PCIE-16GB	10.2	6.0
2	Omaha	Tesla V100-PCIE-16GB	10.2	7.0
14	Omaha	Tesla V100-PCIE-32GB	10.2	7.0
12	OSG_US_NEWJERSEY_ELSA	GeForce GTX 1080 Ti	10.1	6.1
12	SDSC-PRP	GeForce GTX 1080	11.0	6.1
4	SDSC-PRP	GeForce GTX 1080 Ti	11.0	6.1
46	SDSC-PRP	GeForce RTX 2080 Ti	11.0	7.5
77	SU-ITS	GeForce GTX 750 Ti	11.0	5.0

GPU specific machine attributes

```
CUDACapability = 7.5
CUDAClockMhz = 1620.0
CUDAComputeUnits = 72
CUDADeviceName = "Quadro RTX 6000"
CUDADriverVersion = 10.2
CUDAECCEnabled = true
CUDAGlobalMemoryMb = 22699
CUDAOpenCLVersion = 1.2
```

The compute capability of a GPU determines its general specifications and available features: https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#compute-capabilities

```
request_gpus = 1
request_cpus = 1
request_memory = 4 GB

requirements = HAS_SINGULARITY == True && \
               CUDACapability >= 3
```

Table 14. Feature Support per Compute Capability

Feature Support	Compute Capability				
(Unlisted features are supported for all compute capabilities)	**3.5, 3.7, 5.0, 5.2**	**5.3**	**6.x**	**7.x**	**8.0**
Atomic functions operating on 32-bit integer values in global memory (Atomic Functions)	Yes				
Atomic functions operating on 32-bit integer values in shared memory (Atomic Functions)	Yes				
Atomic functions operating on 64-bit integer values in global memory (Atomic Functions)	Yes				
Atomic functions operating on 64-bit integer values in shared memory (Atomic Functions)	Yes				
Atomic addition operating on 32-bit floating point values in global and shared memory (atomicAdd())	Yes				
Atomic addition operating on 64-bit floating point values in global memory and shared memory (atomicAdd())	No		Yes		
Warp vote functions (Warp Vote Functions)	Yes				
Memory fence functions (Memory Fence Functions)					
Synchronization functions (Synchronization Functions)					
Surface functions (Surface Functions)					
Unified Memory Programming (Unified Memory Programming)					
Dynamic Parallelism (CUDA Dynamic Parallelism)					
Half-precision floating-point operations: addition, subtraction, multiplication, comparison, warp shuffle functions, conversion	No	Yes			
Tensor Cores	No		Yes		
Mixed Precision Warp-Matrix Functions (Warp matrix functions)	No		Yes		
Hardware-accelerated async-copy (Asynchronously Copy Data from Global to Shared Memory)	No				Yes
Hardware-accelerated Split Arrive/Wait Barrier (Split Arrive/Wait Barrier)	No				Yes
L2 Cache Residency Management (Device Memory L2 Access Management)	No				Yes

--nv / CUDA / OpenCL

Singularity documentation: Commands that run, or otherwise execute containers (shell, exec) can take an --nv option, which will setup the container's environment to use an NVIDIA GPU and the basic CUDA libraries to run a CUDA enabled application. The --nv flag will:

- Ensure that the /dev/nvidiaX device entries are available inside the container, so that the GPU cards in the host are accessible.
- Locate and bind the basic CUDA libraries from the host into the container, so that they are available to the container, and match the kernel GPU driver on the host.
- Set the LD_LIBRARY_PATH inside the container so that the bound-in version of the CUDA libraries are used by applications run inside the container.

What this means for the OSG user: when the job starts up inside the Singularity container, the environment is fully set up, with a configured LD_LIBRARY_PATH containing the host libraries

Base Images

OSG open pool maintains a set of Singularity base images, which you may either use directly or derive your own image from:

/cvmfs/singularity.opensciencegrid.org/opensciencegrid/

- osgvo-el7-cuda10:10.1
- osgvo-el7-cuda10:10.2
- ~~osgvo-el7-cuda10:latest~~
- tensorflow-gpu:2.2-cuda-10.1
- tensorflow-gpu:2.3-cuda-10.1
- ~~tensorflow-gpu:latest~~

See documentation for a list and links to container definitions: https://support.opensciencegrid.org/support/solutions/articles/12000073449-available-containers-list

Summary

1. GPUs are now widely available in the OSG open pool
2. Match jobs against attributes/capabilities, not specific models
3. Use provided Singularity images to get started

Documentation:

https://support.opensciencegrid.org/support/solutions/articles/5000653025-gpu-jobs

Available Containers:

https://support.opensciencegrid.org/support/solutions/articles/12000073449-available-containers-list

Questions?

support@osgconnect.net