University of Manitoba

Digital Research Alliance of Canada

Using GP GPU compute on Grex

UofM-Spring-Workshop 2022

May 2022

Grigory Shamov

University of Manitoba

What is GPU for

- GP GPU (general-purpose Graphical Processor Unit) can be used for HPC
  - They have thousands of specialized computing units
  - Accelerating math (integer, floating point in single and double precision)
  - Accelerating Machine Learning w tensor cores
  - NVidia is pioneer, has largest scale of the market
- Software needs to be rewritten for GPUs
  - Need dev tools (CUDA, Nvidia HPC pack, libraries like CUDNN)
  - ML packages (TensorFlow, etc.)
- On HPC or cloud, need to be able to find and specify the GPU resources

NVIDIA TESLA V100 FOR PCIe

	Tesla V100 for NVLink	Tesla V100 for PCIe	Tesla V100S for PCIe
PERFORMANCE with NVIDIA GPU Boost™	DOUBLE-PRECISION 7.8 teraFLOPS SINGLE-PRECISION 15.7 teraFLOPS DEEP LEARNING 125 teraFLOPS	DOUBLE-PRECISION 7 teraFLOPS SINGLE-PRECISION 14 teraFLOPS DEEP LEARNING 112 teraFLOPS	DOUBLE-PRECISION 8.2 teraFLOPS SINGLE-PRECISION 16.4 teraFLOPS DEEP LEARNING 130 teraFLOPS
INTERCONNECT BANDWIDTH Bi-Directional	NVLINK 300 GB/s	PCIE 32 GB/s	PCIE 32 GB/s
MEMORY CoWoS Stacked HBM2	CAPACITY 32/16 GB HBM2 BANDWIDTH 900 GB/s		CAPACITY 32 GB HBM2 BANDWIDTH 1134 GB/s
POWER Max Consumption	300 WATTS	250 WATTS	

University of Manitoba

What GPU capacity is available

- National systems: every HPC and Arbutus cloud has a GPU partition
  - Cedar, Graham, Beluga, Narval have a mix of P100s , V100s, A100s
  - Niagara has a sister GPU cluster, Mist
  - Arbutus OpenStack cloud has virtual GPUs (V100s)

- Local HPC resource (Grex, GP nodes)
  - Two nodes (**--partition=gpu**) of 4xV100 32GB VRAM each, NVLink, 192GB RAM, 32 CPU cores (Intel 5128)

- Local HPC resource (Grex, user-contributed nodes)
  - Three nodes (**--partition=stamps-b**) of 4xV100 16GB VRAM each, NVLink, 192GB RAM, 32 CPU cores (Intel 5128)
  - One HGX-2 node (**--partition=livi-b**) 16x V100 32GB VRAM each, NVSwitch 1.5TB RAM, 48 CPU cores (Intel 6248R)
  - Two nodes (**--partition=aggro-b**) 2x A30 24GB VRAM each, 512 GB RAM, 24 CPU AMD 7402

- Contributors would use **livi** , **stamps** and **aggro** partitions to have preemptive access with 1h delay

University of Manitoba

How to get a GPU on Grex

- SLURM syntax for GPUs
  - You will need to select a Partition that has GPUs! (**--partition=stamps-b**)
  - You will need to specify number of GPUs and other resources (CPU, mem, time)
  - Something called cons_TRES; **–gpus=N**; **–gpus-per-node=N**; **–mem-per-gpu=M**
  - Not all combinations of –nodes, –ntasks and –X-per-Y are sensible!

- How much CPUs and memory per GPU is to ask?
  - Start with average (i.e., on the 4x V100 node of 32 GPU, **–cpus-per-gpu=8 –mem-per-cpu=4000M**)

- Interactive job example w salloc
  - ***salloc --partition=stamps-b --gpus=1 --cpus-per-gpu=8 --mem-per-cpu=4000***
  - Try **nvidia-smi** ; try a sample from /global/software/cuda/11.4.3-gcc48/samples

- Batch job example with sbatch

University of Manitoba

Prerequisites for a GPU calculation

- A physical GPU ! (lets talk about NVidia)
- GPU kernel drivers and libraries installed and working (check with **nvidia-smi**)
- Some Ready-made GPU software or
- CUDA for code development (must match the supported GPU driver version and GPU capabilities)
- Sometimes, NVidia HPC suite for OpenACC etc., or other GPU-based high level coding language
- Or some Libraries (cuBLAS, ML, Magma/Plasma, PETSc) that use GPUs

University of Manitoba

Getting GPU software

- Canned GPU codes, commercial : Gaussian, etc.; Guppy (bioinformatics); Matlab
- ML Packages (Tensorflow etc.) – can installed via a Python packaging like Conda.
- Compiling your own software;
  - Need “**module load gcc/$ver**” or “**module load intel/$ver**” first, then “**module load cuda/$ver**”
  - CUDA versions 10.2 and 11.3 are available on Grex (**module spider cuda**) . Gives **nvcc**
  - After loading the modules, proceed with cmake or configure, make etc. as per package’s instruction
  - Some GPU codes need NVidia HPC toolkit (Portland Group compilers for OpenACC)
- Containers: Singularity (now Apptainer) and NVidia NGC repository
  - https://catalog.ngc.nvidia.com/
  - Get package from NGC Cloud using **singularity pull**
  - Run **singularity exec** as described (requires bind-mounting container directories to local directories , usually).