Software Infrastructure and Make Systems

2022 GPU Computing Workshop Series

Brian Vanderwende - *vanderwb@ucar.edu*
CISL - HPCD - ***Consulting Services Group***

NCAR
UCAR

March 17, 2022

Workshop Etiquette

- Please mute yourself and turn off video during the session.
- Questions may be submitted in the chat and will be answered when appropriate. You may also raise your hand, unmute, and ask questions during Q&A at the end of the presentation.
- By joining today, you are agreeing to UCAR's Code of Conduct
- Recordings & other material will be archived & shared publicly.
- Feel free to follow up with the GPU workshop team via Slack or submit support requests to support.ucar.edu
  - **Office Hours**: Asynchronous support via Slack or schedule a time

Workshop Series and Logistics

- Scheduled biweekly through August 2022 (short break in May)

- Sequence of sessions detailed on main webpage
  - Full workshop course description document/syllabus
  - Useful resources for independent self-directed learning included

- Registrants may use workshop's Project ID & Casper core hours
  - Please only **submit non-production, test/debug scale jobs**
  - For non-workshop jobs, request an allocation. Easy access startup allocations may be available for new faculty and graduate students.
  - New NCAR HPC users should review our HPC Tutorials page

GPU Community Engagement

Below are recommended community resources

- Join NCAR GPU Users Slack and #gpu_workshop_participants

- Consider joining other Slack communities or online spaces
  - OpenACC and GPU Hackathon Slack workspace (NVIDIA managed)
  - If you're excited about Julia, they have a Slack and #GPU channel
  - NCAR GPU Tiger Team for latest updates and future directions at NCAR
  - Watch Stackoverflow tags for OpenACC, OpenMP, CUDA, or others

- Prepare an application for an upcoming GPU Hackathon

Find your GPU community! Key to modern science is collaboration!

Slack Status Check

Go to the #gpu_workshop_participants channel on the NCAR GPU Users Slack workspace

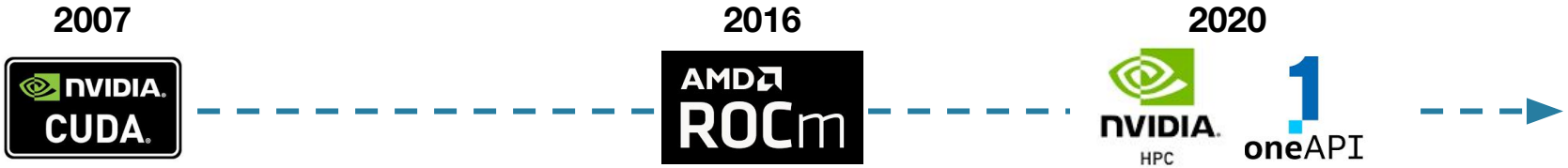
1. Answer our poll about JupyterHub access
2. Share one piece of feedback you have from our workshop sessions thus far. For example:
   a. Something you found useful and/or interesting
   b. Something you found confusing or a topic for which you would simply would like more detail
   c. Something new you would like to see covered in future sessions

If you have yet to join the NCAR GPU Users Slack workspace, use this invite link: https://bit.ly/3ibXPYT

Topics we will cover today

- Tools and libraries for compiled-language GPU codes
- Accessing GPU software on Casper
- Compiling GPU code with the NVIDIA HPC compilers
- Simple compile- and run-time diagnostics
- Writing makefiles to build GPU codes

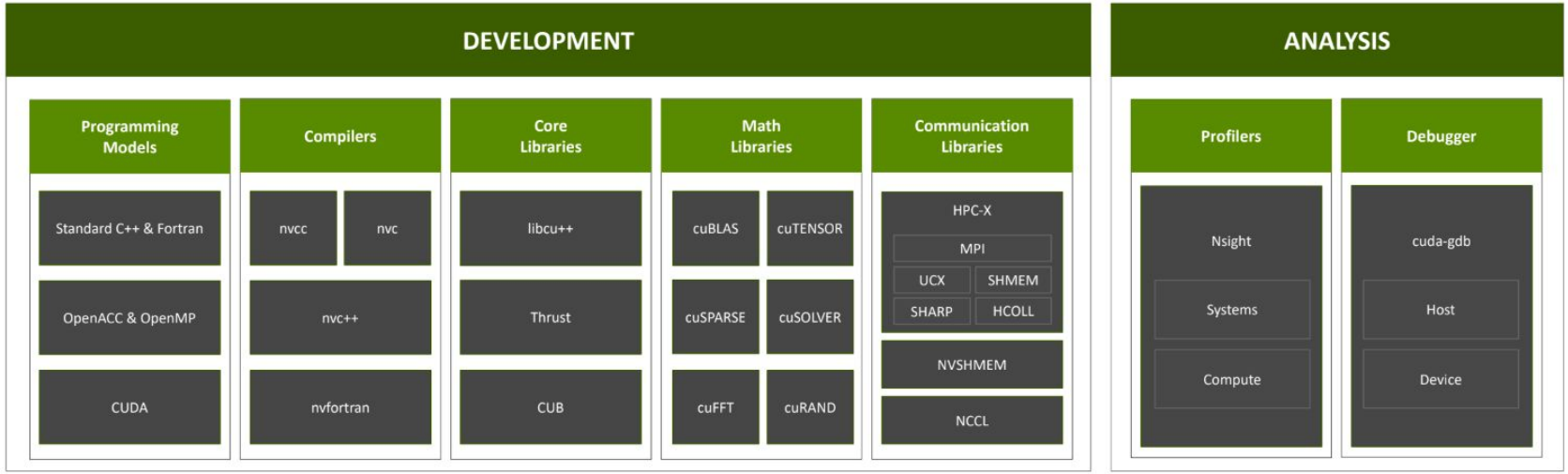
The GPGPU software development ecosystem is growing and diversifying



SDKs are typically designed around specific GPU architectures

NVIDIA's HPC SDK is the primary set of tools for building and analyzing GPGPU code at NCAR…

Recall from last session …



The **NVIDIA HPC SDK** includes *compilers* and *libraries* for building GPU applications, *profilers* for optimizing, and a *debugger* for troubleshooting runtime bugs and crashes

Using the NVIDIA HPC SDK

SDK releases (7-8 per year):
https://developer.nvidia.com/hpc-sdk

Installed on Casper as *nvhpc* module

- Adds compilers and NSight utilities to your PATH

NVIDIA also provides pre-built containers with SDK installed

```
[23:05] ~$ module avail nvhpc

/glade/u/apps/dav/modulefiles/defau
  nvhpc/20.9      nvhpc/21.9  (L,D)
  nvhpc/20.11     nvhpc/21.11
  nvhpc/21.3      nvhpc/22.1
  nvhpc/21.7      nvhpc/22.2

 Where:
  L:  Module is loaded
  D:  Default Module
```

Compilers included in the HPC SDK

nvc - the C compiler (formerly pgcc)
nvc++ - the C++ compiler (formerly pgc++)
nvfortran - the Fortran compiler (formerly pgf90/pgfortran)
nvcc - the C++ CUDA driver (also included in CUDA toolkit)

Don't confuse **nvc/nvc++** with **nvcc**. The latter will process CUDA C++ code and rely on an underlying C++ compiler.

OpenACC, OpenMP, and CUDA Fortran code is handled directly by the three compilers shown above.

The NVIDIA HPC SDK installation also comes with a plethora of example codes

```
$NVHPC/Linux_x86_64/22.2/examples/
|-- AutoPar
|-- CUDA-Fortran
|-- CUDA-Libraries
|-- F2003
|-- MPI
|-- NVLAmath
|-- OpenACC
|-- OpenMP (CPU-only at present)
|-- README
`-- stdpar
```

Most examples include a Makefile for generating binaries

Some examples (e.g., OpenACC samples) depend on included libraries, so it's best to copy the entire examples directory to your work/scratch

CUDA modules on Casper allow for greater flexibility

Casper also offers **cuda** modules:

- Allows you to customize CUDA version when using **nvhpc**
- Allows you to compile C++ CUDA code with other compilers using **nvcc**
- Only provides the CUDA toolkit
  - *No compilers or NSight tools*

```
[17:04] ~$ module load nvhpc/22.2
[17:04] ~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2021 NVIDIA Corporation
Built on Fri_Dec_17_18:16:03_PST_2021
Cuda compilation tools, release 11.6, V11.6.55
Build cuda_11.6.r11.6/compiler.30794723_0
[17:04] ~$ module load cuda/11.4.0
[17:05] ~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2021 NVIDIA Corporation
Built on Wed_Jun__2_19:15:15_PDT_2021
Cuda compilation tools, release 11.4, V11.4.48
Build cuda_11.4.r11.4/compiler.30033411_0
[17:05] ~$ nvfortran -show |& grep USER_SET_CUDA
USER_SET_CUDA       =/glade/u/apps/dav/opt/cuda/11.4.0/
```

Where are the C/C++ CUDA examples?

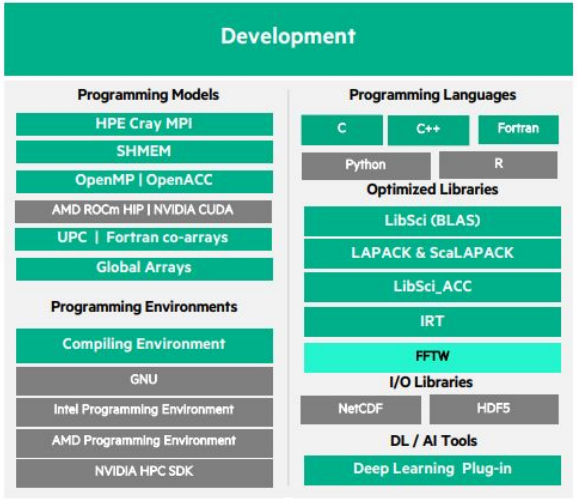
Sample codes used to come with the CUDA toolkit, but recently NVIDIA distributes in a separate Git repo

```
0_Introduction
1_Utilities
2_Concepts_and_Techniques
3_CUDA_Features
4_CUDA_Libraries
5_Domain_Specific
6_Performance
```

```
git clone -b v11.6 --depth 1
https://github.com/NVIDIA/cuda-samples
```

GPU capabilities of Derecho and the Cray Programming Environment

DERECHO
NCAR Supercomputer



- **Derecho** will have 82 GPU compute nodes with 4 NVIDIA A100s each
- Cray Compiling Environment is projected to support:
  - A100 GPUs (*June 2022*)
  - Partial OpenMP 5.1 (*June 2022*)
  - OpenACC 3.1 (*Q4 2022*)
  - CUDA support **for Fortran only**

CISL will fully support both NVIDIA HPC SDK and Cray Programming Environment on Derecho

Requesting GPUs on Casper

Casper current has two types of GPUs:

- NVIDIA V100 Volta (GPGPU)
- NVIDIA Quadro GP100 (vis and analysis)

4 nodes with 4xV100 GPUs
6 nodes with 8xV100 GPUs

These resources are in high demand; please be mindful!

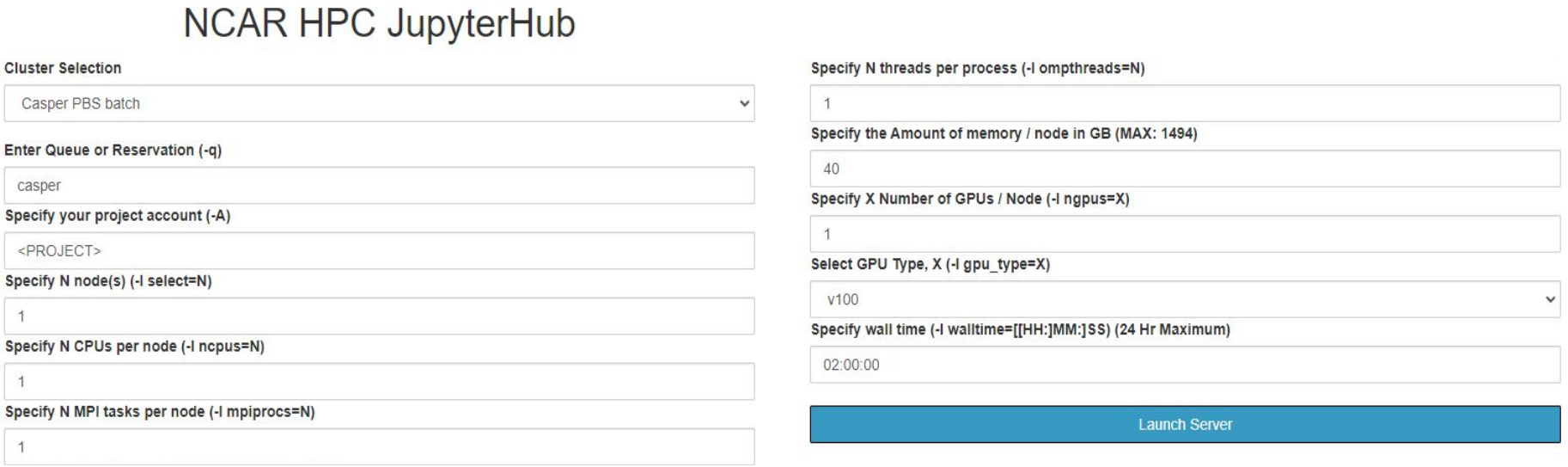
In PBS job scripts, request GPUs using batch directives:

```
#PBS -l select=1:ncpus=1:ngpus=1:mem=40GB
#PBS -l gpu_type=v100
```

or use execcasper for interactive command-line sessions:

```
execcasper --ngpus 1 --gpu v100 --mem=40GB
```

Requesting Casper GPUs in JupyterHub



Proceeding with the interactive Jupyter Notebook

Next, we will run an interactive Jupyter Notebook

1. In a browser, sign in to https://jupyterhub.hpc.ucar.edu/stable
2. Choose a server name and click “Add…”
3. Select “Casper PBS Batch”
4. Modify the following settings:
   a. Account = **UCIS0004**
   b. Queue = **gpuworkshop** (if participating live)
   c. ngpus = **1**; gpu_type=**gp100;** walltime=**00:45:00**
5. In JupyterLab, navigate to your clone of the workshop Git repo and pull the latest changes