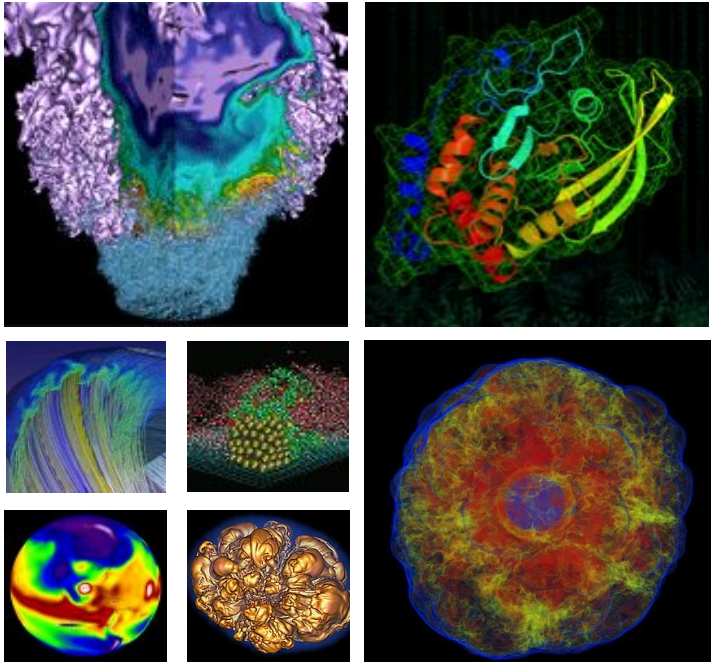
NUG Monthly Meeting



NERSC

17 Jun, 2021

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Today's plan

- Interactive - please participate!
  - Raise hand or just speak up
  - **NERSC User Slack** (link in chat), **#webinars** channel

- Agenda:
  - Win-of-the-month
  - Today-I-learned
  - Announcements/CFPs
  - Topic of the day: **GPU Programming Models and Interoperability**
  - Coming meetings: topic suggestions/requests?
  - Last month's numbers

Win of the month

Show off an achievement, or shout out someone else's achievement, e.g.:

- Had a paper accepted
- Solved a bug
- A scientific achievement (maybe candidate for Science highlight, or **High Impact Scientific Achievement award**)
- An **Innovative Use of High Performance Computing** (also a candidate for an award) (https://www.nersc.gov/science/nersc-hpc-achievement-awards/)

Please let us know of award-worthy work from you or your colleagues - tell us what you did, and what was the key insight?

Today I learned

What surprised you that might benefit other users to hear about?

(and might help NERSC identify documentation improvements!)

Eg:

- Something you got stuck on, hit a dead end, or turned out to be wrong about
  - Give others the benefit of your experience!
  - Opportunity to improve NERSC documentation
- A tip for using NERSC
- Something you learned that might benefit other NERSC users

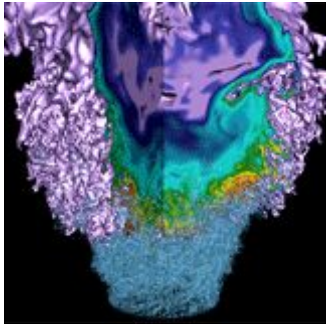
"If we knew what it was we were doing, it would not be called research, would it?" - Einstein

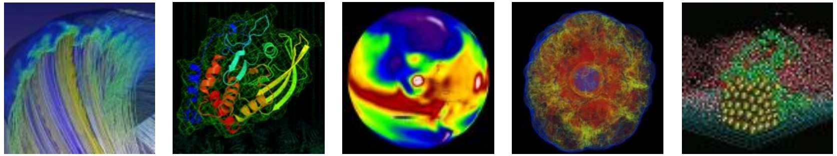
Announcements and CFPs

See weekly email for these and more:

- NERSC Power Outage July 9-12
- Upcoming Cori OS update - August
  - Statically-linked executables likely to need re-link
- CFPs:
  - Parallel Applications Workshop, Alternatives to MPI+X (at SC21)
  - SuperCheck-SC21
- Training:
  - Lmod on Perlmutter, June 22
  - CI at NERSC, July 7
  - ECP-IDEAS webinar on Mining Development Data to Understand and Improve Software Engineering Processes, July 7
  - CUDA Multithreading with Streams, July 16

GPU Programming Models and Interoperability



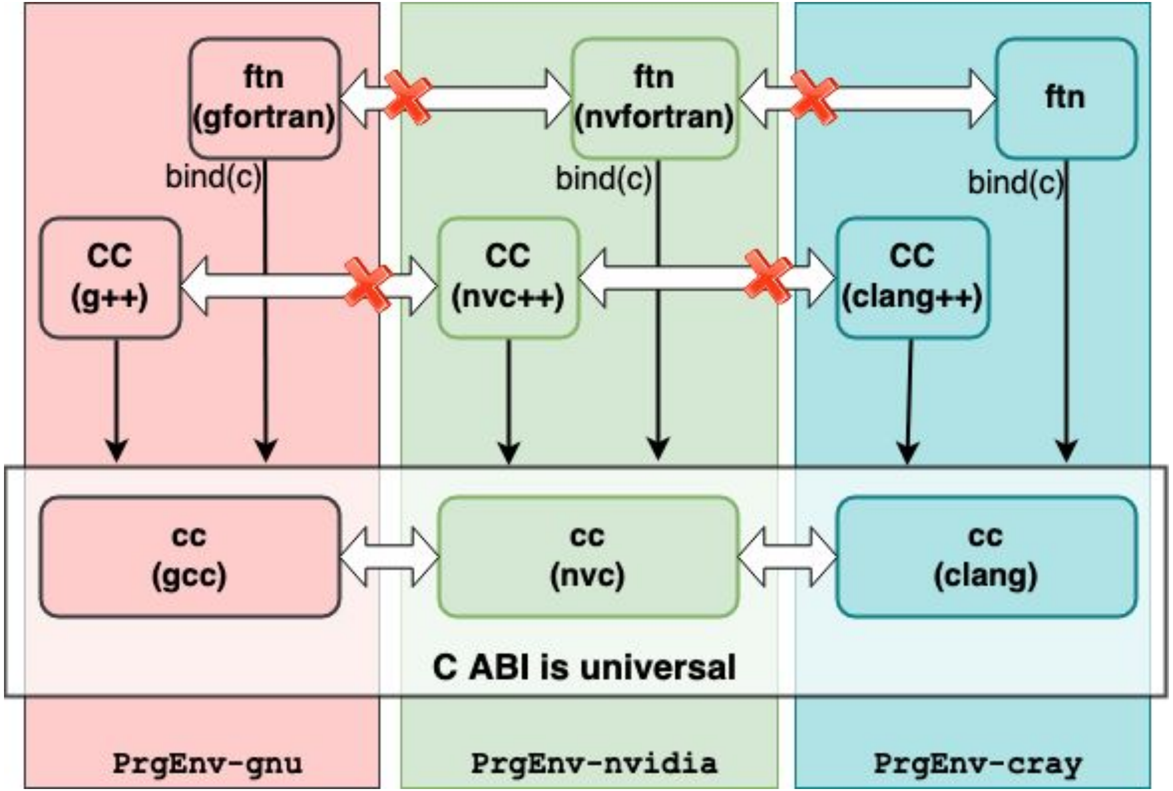


NERSC

Between PrgEnvs: no interoperability

- In general: Fortran and C++ can call C libs - even across PrgEnvs

- Fortran and C++ between PrgEnvs are generally not interoperable



Which PrgEnv?

Recommendation: use PrgEnv-nvidia for GPU applications

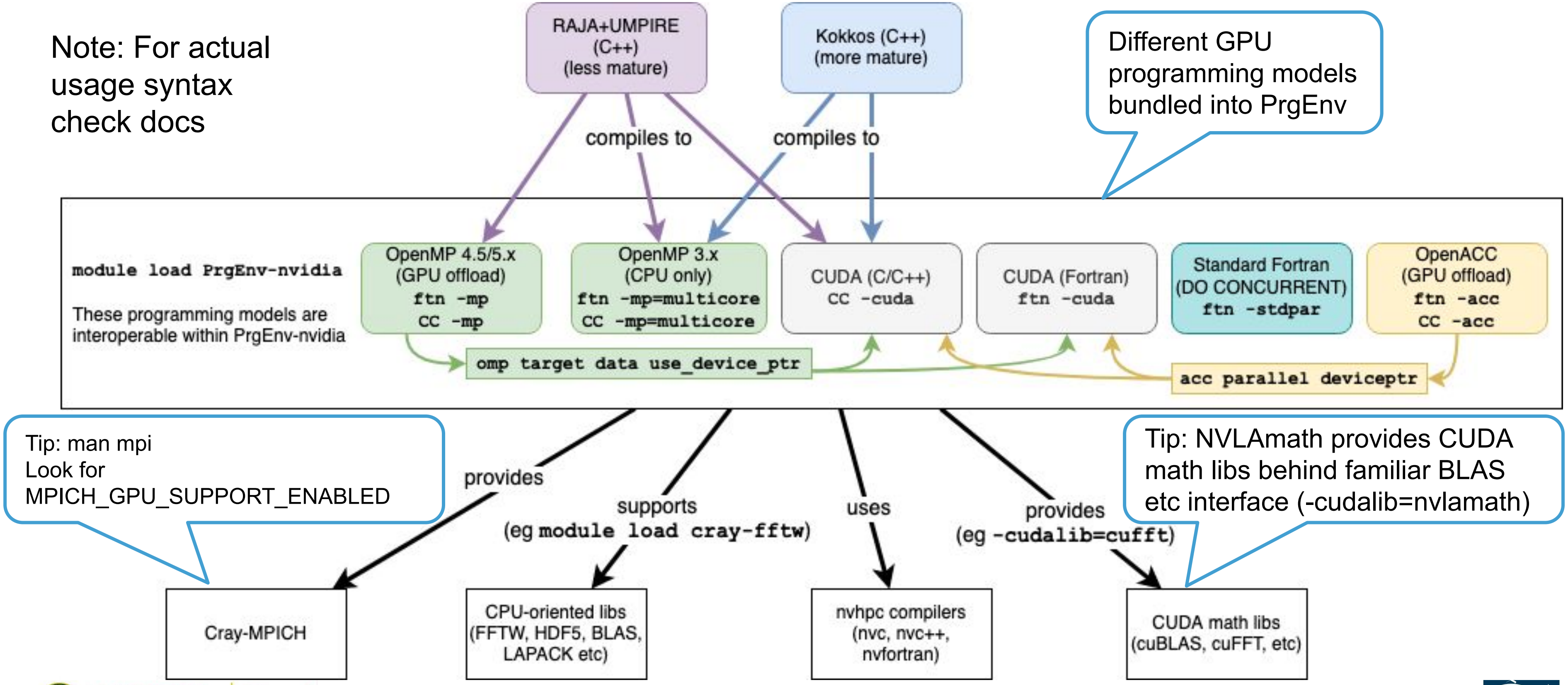
- The most GPU-oriented
- Significant work has gone into making PrgEnv-nvidia a GPU-friendly environment
  - Including support for interoperability
- **Caveats:**
  - Uses nvc / nvc++ / nvfortran for CPU code too
  - Another PrgEnv may offer better performance for CPU code

Recommendation: Avoid mixing GPU programming models

- Higher-level models are more likely to interoperate, eg Kokkos calling CUDA or OpenMP-based libs, OpenMP calling CUDA-based libs

PrgEnv-nvidia

Note: For actual usage syntax check docs



RAJA+UMPIRE (C++) (less mature)

Kokkos (C++) (more mature)

compiles to

compiles to

Different GPU programming models bundled into PrgEnv

`module load PrgEnv-nvidia`

These programming models are interoperable within PrgEnv-nvidia

OpenMP 4.5/5.x (GPU offload)
`ftn -mp`
`CC -mp`

OpenMP 3.x (CPU only)
`ftn -mp=multicore`
`CC -mp=multicore`

CUDA (C/C++)
`CC -cuda`

CUDA (Fortran)
`ftn -cuda`

Standard Fortran (DO CONCURRENT)
`ftn -stdpar`

OpenACC (GPU offload)
`ftn -acc`
`CC -acc`

`omp target data use_device_ptr`

`acc parallel deviceptr`

Tip: man mpi
Look for
MPICH_GPU_SUPPORT_ENABLED

provides

supports (eg `module load cray-fftw`)

uses

provides (eg `-cudalib=cufft`)

Tip: NVLAmath provides CUDA math libs behind familiar BLAS etc interface (-cudalib=nvlamath)

Cray-MPICH

CPU-oriented libs (FFTW, HDF5, BLAS, LAPACK etc)

nvhpc compilers (nvc, nvc++, nvfortran)

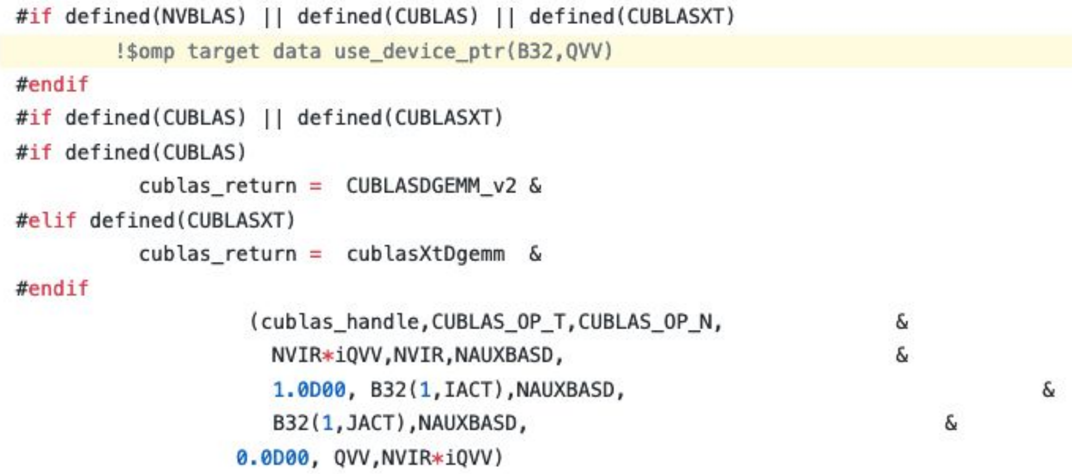
CUDA math libs (cuBLAS, cuFFT, etc)

Eg: OpenMP app calling CUDA lib

From: https://github.com/jkwack/GAMESS_RI-MP2_MiniApp/blob/ECP-proxy/source/rimp2_energy_whole_KERN.f90#L274

Tip: **NVLAmath** provides CUDA math libs behind familiar BLAS etc interface (-cudalib=nvlamath)

(See https://drive.google.com/file/d/1yaRtAWJ13sHrskwP6PCz3NYmjSogPhS6/view?usp=sharing)



```
#if defined(NVBLAS) || defined(CUBLAS) || defined(CUBLASXT)
        !$omp target data use_device_ptr(B32,QVV)
#endif
#if defined(CUBLAS) || defined(CUBLASXT)
#if defined(CUBLAS)
          cublas_return =  CUBLASDGEMM_v2 &
#elif defined(CUBLASXT)
          cublas_return =  cublasXtDgemm   &
#endif
                   (cublas_handle,CUBLAS_OP_T,CUBLAS_OP_N,           &
                    NVIR*iQVV,NVIR,NAUXBASD,                          &
                    1.0D00, B32(1,IACT),NAUXBASD,                               &
                    B32(1,JACT),NAUXBASD,                                  &
                  0.0D00, QVV,NVIR*iQVV)
```

Other PrgEnv may be more complicated

PrgEnv-gnu: patchy GPU support, generally lower (GPU) performance

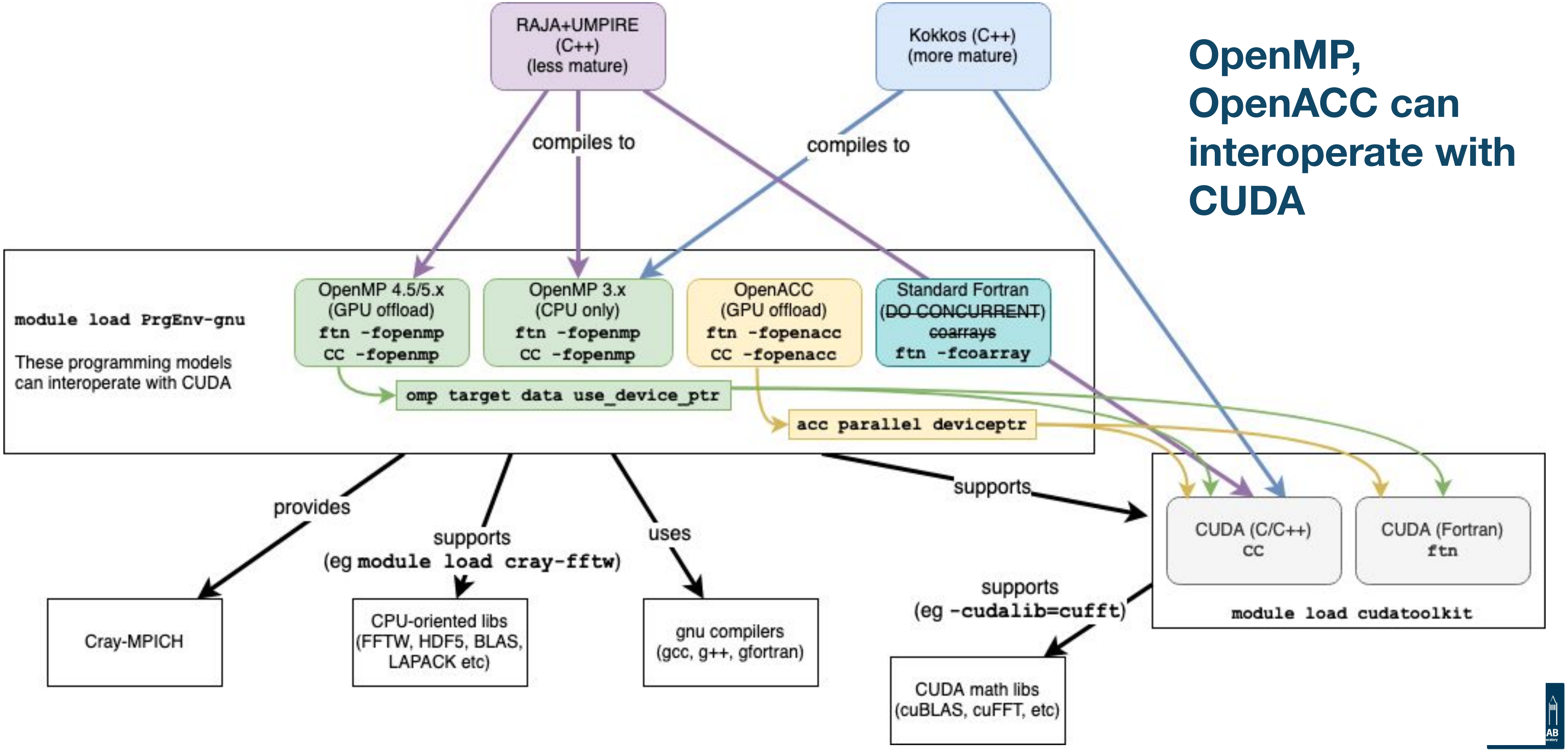
PrgEnv-cray: good CPU performance, C++ and Fortran compilers are based on different suites

- C/C++: based on Clang
- Fortran: Cray-classic (in-house development)

Using CUDA requires separate **`module load cudatoolkit`**, usage may be more complex

PrgEnv-gnu

OpenMP, OpenACC can interoperate with CUDA



RAJA+UMPIRE (C++) (less mature)

Kokkos (C++) (more mature)

compiles to

compiles to

`module load PrgEnv-gnu`

These programming models can interoperate with CUDA

OpenMP 4.5/5.x (GPU offload)
`ftn -fopenmp`
`CC -fopenmp`

OpenMP 3.x (CPU only)
`ftn -fopenmp`
`CC -fopenmp`

OpenACC (GPU offload)
`ftn -fopenacc`
`CC -fopenacc`

Standard Fortran
~~(DO CONCURRENT)~~
~~coarrays~~
`ftn -fcoarray`

`omp target data use_device_ptr`

`acc parallel deviceptr`

provides

supports (eg `module load cray-fftw`)

uses

supports

Cray-MPICH

CPU-oriented libs (FFTW, HDF5, BLAS, LAPACK etc)

gnu compilers (gcc, g++, gfortran)

CUDA (C/C++)
`CC`

CUDA (Fortran)
`ftn`

`module load cudatoolkit`

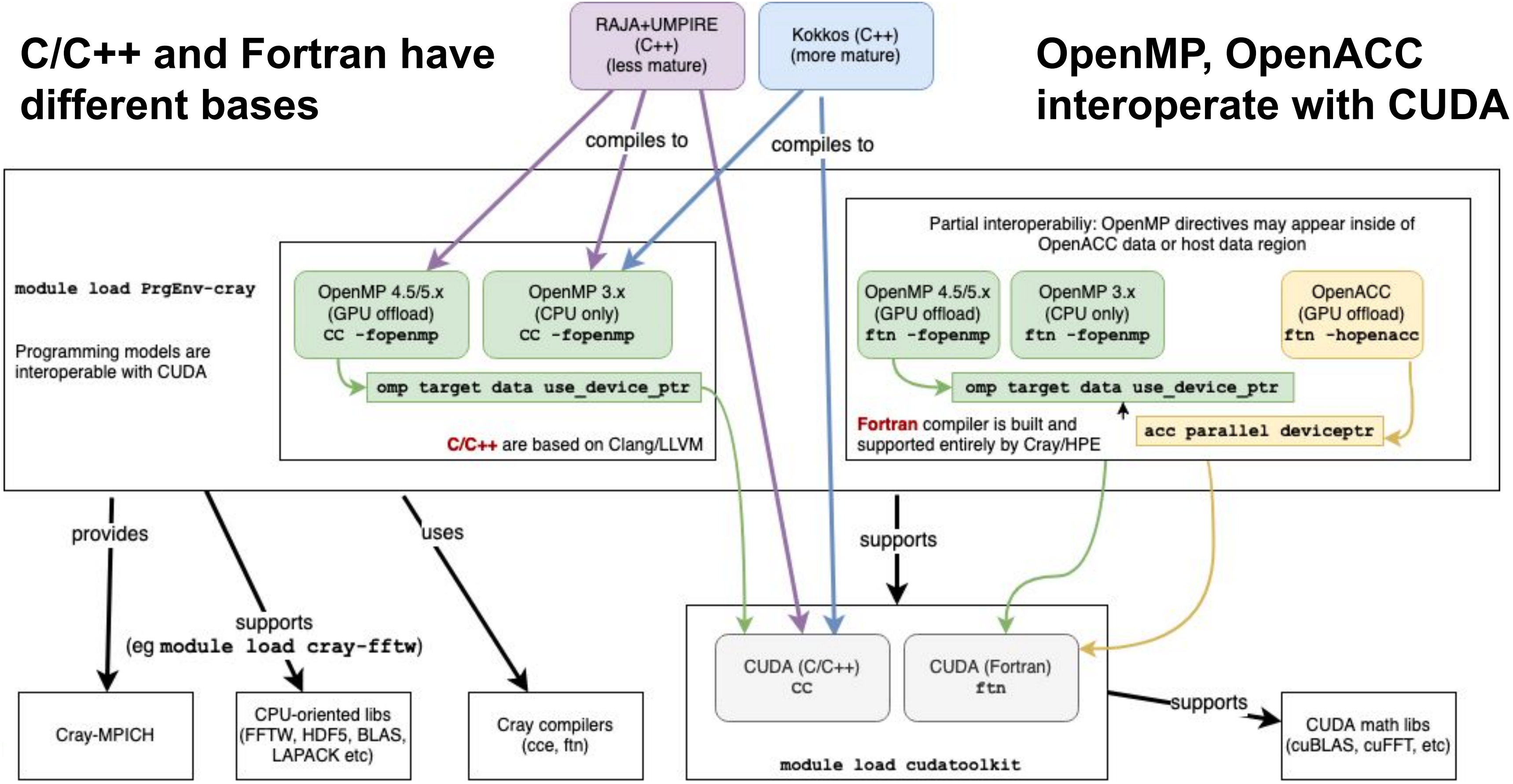
supports (eg `-cudalib=cufft`)

CUDA math libs (cuBLAS, cuFFT, etc)

PrgEnv-cray

C/C++ and Fortran have different bases

OpenMP, OpenACC interoperate with CUDA



RAJA+UMPIRE (C++) (less mature)

Kokkos (C++) (more mature)

compiles to

compiles to

`module load PrgEnv-cray`

Programming models are interoperable with CUDA

OpenMP 4.5/5.x (GPU offload) `CC -fopenmp`

OpenMP 3.x (CPU only) `CC -fopenmp`

`omp target data use_device_ptr`

C/C++ are based on Clang/LLVM

Partial interoperabiliy: OpenMP directives may appear inside of OpenACC data or host data region

OpenMP 4.5/5.x (GPU offload) `ftn -fopenmp`

OpenMP 3.x (CPU only) `ftn -fopenmp`

OpenACC (GPU offload) `ftn -hopenacc`

`omp target data use_device_ptr`

`acc parallel deviceptr`

Fortran compiler is built and supported entirely by Cray/HPE

provides

supports (eg `module load cray-fftw`)

uses

supports

Cray-MPICH

CPU-oriented libs (FFTW, HDF5, BLAS, LAPACK etc)

Cray compilers (cce, ftn)

CUDA (C/C++) `CC`

CUDA (Fortran) `ftn`

`module load cudatoolkit`

supports

CUDA math libs (cuBLAS, cuFFT, etc)

Summary

- Interoperability is possible - but best not to rely on it
- Higher-level frameworks (Kokkos, RAJA) give the most flexibility
  - But C++ only
- OpenMP (especially OpenMP 5.x) and OpenACC have support for interoperability
  - But support for OpenMP 5.x is still immature/in-development

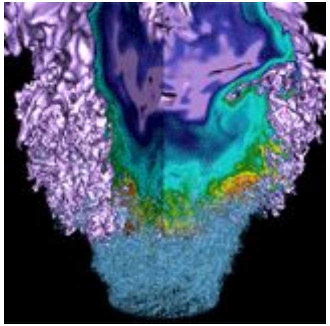
See interoperability section of
https://www.nersc.gov/users/training/events/nvidia-hpcsdk-openmp-target-offload-training-december-2020/

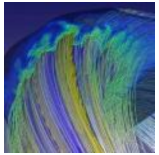
Also see:

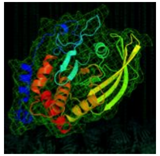
https://www.nersc.gov/users/training/events/openacc-data-management-part-2-of-3-openacc-training-series-may-28-2020/

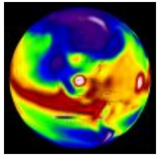
https://www.nersc.gov/users/training/gpus-for-science/gpus-for-science-2020/

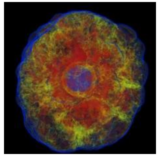
Discussion / Q&A

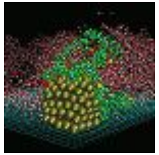












NERSC

Coming up

Topic requests/suggestions?

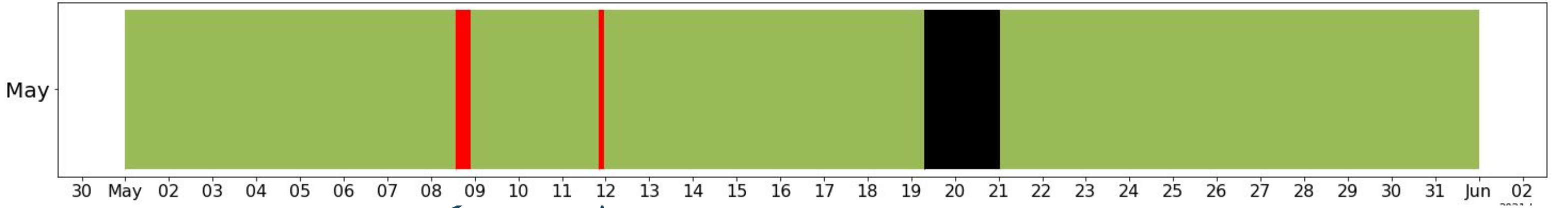
We'd love to hear some lightning talks **from NERSC users** about the research you use NERSC for!

Last month's numbers - May

Scheduled and overall availability:

	Scheduled	Overall
Cori	98.5%	92.9%
HPSS	100%	99.5%
CFS	100%	100%

Cori: 2 outages (2 unscheduled totaling 10 hrs)



Power supply issue in a cabinet caused HSN issues

CSCRATCH hanging

Scheduled maintenance

Last month's numbers - April

Cori Utilization: 94.1%

Large jobs: 30.7%

New Tickets: 528

Closed Tickets: 548

Backlog at 1 May: 466

NERSC

Thank You

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory