NUG Monthly Meeting

17 Jun, 2021
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
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
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)

Different GPU programming models bundled into PrgEnv

module load PrgEnv-nvidia
These programming models are interoperable within PrgEnv-nvidia

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

Tip: man mpi
Look for MPICH_GPU_SUPPORT_ENABLED

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)
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
C/C++ and Fortran have different bases

OpenMP, OpenACC interoperate with CUDA

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

C/C++ are based on Clang/LLVM

module load PrgEnv-cray

Programming models are interoperable with CUDA
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/gpus-for-science/gpus-for-science-2020/
Discussion / Q&A
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:

<table>
<thead>
<tr>
<th></th>
<th>Scheduled</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cori</td>
<td>98.5%</td>
<td>92.9%</td>
</tr>
<tr>
<td>HPSS</td>
<td>100%</td>
<td>99.5%</td>
</tr>
<tr>
<td>CFS</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

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
Thank You