Preparing NERSC Applications for Perlmutter as an Exascale Waypoint

Charlene Yang
Application Performance Specialist
cjyang@lbl.gov
Perlmutter Overview
NERSC is the mission High Performance Computing facility for the DOE SC

- 7,000 Users
- 800 Projects
- 700 Codes
- 2000 NERSC citations per year

Simulations at scale

Data analysis support for DOE’s experimental and observational facilities

Photo Credit: CAMERA
NERSC Systems Roadmap

NERSC-7: Edison
2.5 PFs
Multi-core CPU
3MW

NERSC-8: Cori
30PFs
Manycore CPU
4MW

NERSC-9: Perlmutter
3-4x Cori
CPU and GPU nodes
>5 MW

NERSC-10
ExaSystem
~20MW

2013
2016
2020
2024
NERSC-9 will be named after Saul Perlmutter

- Winner of 2011 Nobel Prize in Physics for discovery of the accelerating expansion of the universe.

- Supernova Cosmology Project, lead by Perlmutter, was a pioneer in using NERSC supercomputers, combining large scale simulations with experimental data analysis

- Login “saul.nersc.gov”
Perlmutter: A System Optimized for Science

- GPU-accelerated and CPU-only nodes meet the needs of large scale simulation and data analysis from experimental facilities
- Cray “Slingshot” - High-performance, scalable, low-latency Ethernet-compatible network
- Single-tier All-Flash Lustre based HPC file system, 6x Cori’s bandwidth
- Dedicated login and high memory nodes to support complex workflows
**Compute Node Details**

- **CPU only nodes**
  - AMD CPUs - Next Generation EPYC
  - CPU only cabinets will provide approximately same capability as full Cori system
  - Efforts to optimize codes for KNL will translate to NERSC-9 CPU only nodes

- **CPU + GPU nodes**
  - NVIDIA GPUs, Next Generation Volta with Tensor cores, high bandwidth memory and NVLINK-3
  - GPU Direct, Unified Virtual Memory for improved programmability
  - 4 to 1 GPU to CPU ratio
From the start NERSC-9 had requirements of simulation and data users in mind

- All Flash file system for workflow acceleration
- Optimized network for data ingest from experimental facilities
- Dedicated workflow management and interactive nodes
- Real-time scheduling capabilities
- Supported analytics stack including latest ML/DL software
- System software supporting rolling upgrades for improved resilience
NESAP Overview
Application Readiness Strategy for Perlmutter

NERSC’s Challenge

How to enable NERSC’s diverse community of 7,000 users, 750 projects, and 700 codes to run on advanced architectures like Perlmutter and beyond?
GPU Readiness Among NERSC Codes (Aug’17 - Jul’18)

<table>
<thead>
<tr>
<th>GPU Status &amp; Description</th>
<th>Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled: Most features are ported and performant</td>
<td>32%</td>
</tr>
<tr>
<td>Kernels: Ports of some kernels have been documented.</td>
<td>10%</td>
</tr>
<tr>
<td>Proxy: Kernels in related codes have been ported</td>
<td>19%</td>
</tr>
<tr>
<td>Unlikely: A GPU port would require major effort.</td>
<td>14%</td>
</tr>
<tr>
<td>Unknown: GPU readiness cannot be assessed at this time.</td>
<td>25%</td>
</tr>
</tbody>
</table>

A number of applications in NERSC workload are GPU enabled already.

We will leverage existing GPU codes from CAAR + Community
Application Readiness Strategy for Perlmutter

How to transition a workload with 700 Apps?

- NERSC Exascale Science Application Program (NESAP)
- Engage ~25 Applications
- up to 17 postdoctoral fellows
- Deep partnerships with every SC Office area
- Leverage vendor expertise and hack-a-thons
- Knowledge transfer through documentation and training for all users
- Optimize codes with improvements relevant to multiple architectures

https://nersc.gov/users/application-performance/nesap/perlmutter/
GPU Transition Path for Apps

**NESAP for Perlmutter** will extend activities from **NESAP for Cori**

1. Identifying and exploiting on-node parallelism
2. Understanding and improving data-locality within the memory hierarchy

**Knowledge and skills of multi/many-core optimization on HSW/KNL transferrable to AMD CPUs**

What’s New for NERSC Users?

1. Heterogeneous compute elements - NVIDIA GPUs
2. Identification and exploitation of even more parallelism
3. Data locality again, host/device

**Emphasis on performance-portable programming approach:** Continuity from Cori through future NERSC systems
NESAP for Perlmutter

Simulation
~12 Apps

Data Analysis
~8 Apps

Learning
~5 Apps

- 6 NESAP for Data apps will be continued. Additional apps focused on experimental facilities.
- 5 ECP Apps Jointly Selected (Participation Funded by ECP)
- Open call for proposals. Reviewed by a committee of NERSC staff, external reviewers and input from DOE PMs.
  - App selection will contain multiple applications from each SC Office and algorithm area
  - Additional applications (beyond 25) will be selected for second tier NESAP with access to vendor/training resources and early access
NESAP for Perlmutter Timeline

2018

Dungeon Sessions Begin

2019
Finalize Edison Reference Numbers

2020

System Delivery

2021

NESAP for Data (6 Existing Apps)

NESAP-1

Code Team Selection (Dec. 2018)

NESAP-2
NESAP-2 Call for Proposals: (Oct. 2018)
HPC, Data/Analytics, Learning

Begin engagement with ECP applications

Early Access
Resources Available to NESAP Awardees

– 1 hackathon session per quarter (Center of Excellence)
  • NERSC, Cray, NVIDIA engineer attendance
  • 3-4 code teams per hackathon
– Cray/NVIDIA engineer time before and after sessions
  • 6-week ‘ramp-up’ period with code teams and Cray/NVIDIA to ensure everyone is fully prepared to work on hackathon day 1
  • Tutorials/deep dives into GPU programming models, profiling tools, etc
– NESAP postdocs (NERSC will hire up to 17)
– NERSC application performance specialist attention
– General programming, performance and tools training
– Early access (perlmutter and GPU testbed)
Training, Case Studies and Documentation

● For those teams not in NESAP, there will be a robust training program

● Lessons learned from deep dives from NESAP teams will be shared through case studies and documentation
Programming & Performance Portability

NERSC
NERSC will work with PGI to enable OpenMP GPU acceleration with PGI compilers

- Ensures continuity of OpenMP added to NERSC apps for N8
- Co-design with PGI to prioritize OpenMP features for GPU
- Use lessons learned to influence future versions of OpenMP

NERSC has joined membership in OpenACC

- kokkos training in March 2019
- UPC++ available
- Are you part of an ECP ST project? Interested in contributing a NERSC hosted training?

NERSC collaborating with OLCF and ALCF on development of performanceportability.org
Performance Portability Strategies

- Conditional compilation
- Directives: OpenMP, OpenACC
- Libraries: Use a library when possible
- Abstractions: Kokkos, Raja
- General-purpose high-level programming languages: UPC, Coarray Fortran
- DSLs: NMODL for neuroscience

Good coding practices
- Modularity, some high-level abstractions
- Data structures flexibly allocatable to different memory spaces
- Task level flexibility so work can be allocated to different compute elements (GPU & CPU)
Performance Portability

There is no consensus on the definition or metric for performance portability, but...

**DOE SC Facility Definition (performanceportability.org)**
An application is performance portable if it achieves a consistent ratio of the actual time to solution to either the best-known or the theoretical best time to solution on each platform with minimal platform specific code required.

**Performance portability metric proposed by Pennycook et al. [1]**

\[
\Phi(a, p, H) = \begin{cases} 
\frac{|H|}{\sum_{i \in H} e_i(a, p)} & \text{if } i \text{ is supported, } \forall i \in H \\
0 & \text{otherwise}
\end{cases}
\]

\[
e_i(a, p) = \frac{P_i(a, p)}{\min(F_i, B_i \times I_i(a, p))}
\]


Methodology to Measure Perf. Port.

1. Measure empirical compute and bandwidth ceilings: ERT [3]
2. Measure application performance: SDE and LIKWID on KNL; NVPROF on V100

Performance = \( \frac{\text{SDE or } \text{nvprof} \text{ FLOPs}}{\text{Runtime}} \), Arithmetic Intensity = \( \frac{\text{SDE or } \text{nvprof} \text{ FLOPs}}{\text{LIKWID or } \text{nvprof} \text{ Data Movement}} \)

An example: GPP kernel from BerkeleyGW (Roofline)

Methodology to Measure Perf. Port.

An example: GPP kernel from BerkeleyGW (Perf. Port. Scores)

<table>
<thead>
<tr>
<th></th>
<th>Architectural Efficiency</th>
<th>FMA</th>
<th>V100</th>
<th>Performance Portability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nw=1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>FMA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNL</td>
<td>84.98%</td>
<td>77.50%</td>
<td>66.77%</td>
<td>55.28%</td>
</tr>
<tr>
<td>V100</td>
<td>97.36%</td>
<td>91.50%</td>
<td>76.70%</td>
<td>65.44%</td>
</tr>
<tr>
<td><strong>Performance Portability</strong></td>
<td>90.76%</td>
<td>83.92%</td>
<td>71.39%</td>
<td>59.93%</td>
</tr>
<tr>
<td><strong>No-FMA</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KNL</td>
<td>82.06%</td>
<td>72.95%</td>
<td>73.74%</td>
<td>78.72%</td>
</tr>
<tr>
<td>V100</td>
<td>92.88%</td>
<td>92.88%</td>
<td>97.43%</td>
<td>98.91%</td>
</tr>
<tr>
<td><strong>Performance Portability</strong></td>
<td>87.14%</td>
<td>81.72%</td>
<td>83.95%</td>
<td>87.67%</td>
</tr>
</tbody>
</table>

- Roofline captures application bottlenecks, machine balance, problem size, etc
- Perf. Port. metric captures performance changes across architectures

Summary
NERSC-9: A System Optimized for Science

- Cray Shasta System providing 3-4x capability of Cori system
- First NERSC system designed to meet needs of both large scale simulation and data analysis from experimental facilities
  - Includes both NVIDIA GPU-accelerated and AMD CPU-only nodes
  - Cray Slingshot high-performance network will support Terabit rate connections to system
  - Optimized data software stack enabling analytics and ML at scale
  - All-Flash filesystem for I/O acceleration
- Robust readiness program for simulation, data and learning applications and complex workflows
- Delivery in late 2020
We are hiring!

- Postdoctoral fellows
  - including Grace Hopper fellowship
- Application performance specialists