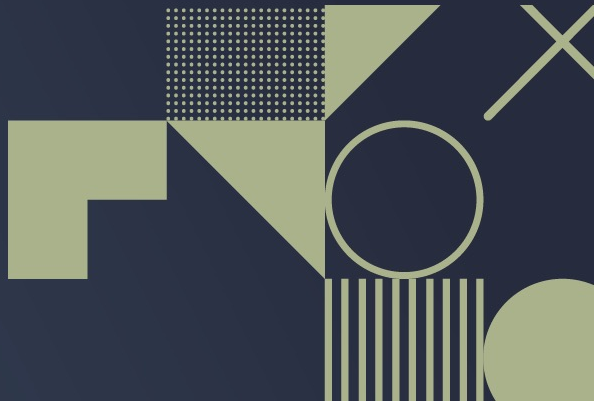
Non-Recurring Engineering (NRE) Best Practices: A Case Study with the NERSC/NVIDIA OpenMP contract

Christopher S. Daley[1], Annemarie Southwell[2], Rahulkumar Gayatri[1], Scott Biersdorff[2], Craig Toepfer[2], Güray Özen[2], Nicholas J. Wright[1]

[1] NERSC, Lawrence Berkeley National Laboratory

[2] NVIDIA Corporation



Overview

- Motivation for an OpenMP NRE contract
  - Perlmutter Supercomputer
  - OpenMP
- Developing a contract to meet NERSC users' requirements
- NVIDIA OpenMP compiler performance results
- Assessment of this NRE contract and our best practices

NERSC and the Perlmutter Supercomputer

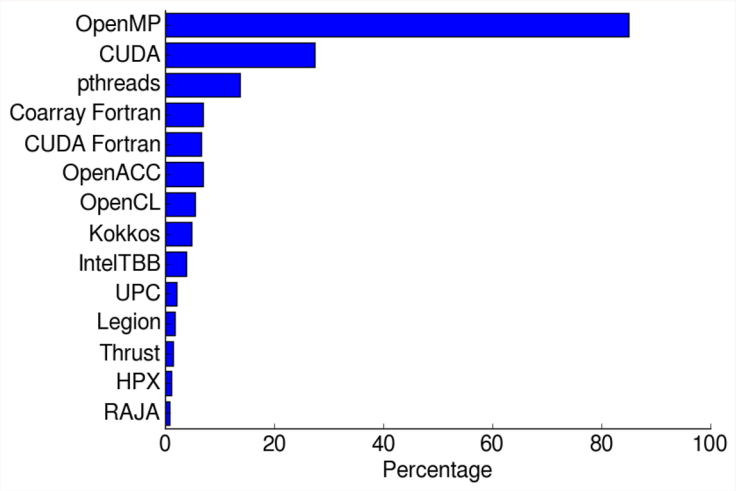
- NERSC provides compute and storage resources for the DOE Office of Science
  - NERSC supports 8000+ users and 700+ codes
- Perlmutter is the first GPU-accelerated supercomputer at NERSC
  - HPE/Cray “Shasta” system with “Slingshot” interconnect and an all Flash Lustre file system
- Perlmutter GPU-accelerated partition:
  - 1500+ nodes
  - Each node has 4 NVIDIA A100 GPUs and 1 AMD Milan CPU
- Perlmutter CPU-only partition (coming soon):
  - 3000+ nodes
  - Each node has 2 AMD Milan CPUs

https://www.nersc.gov/systems/perlmutter/

OpenMP is Very Important at NERSC

Many NERSC users are successfully using OpenMP on the NERSC Cori supercomputer

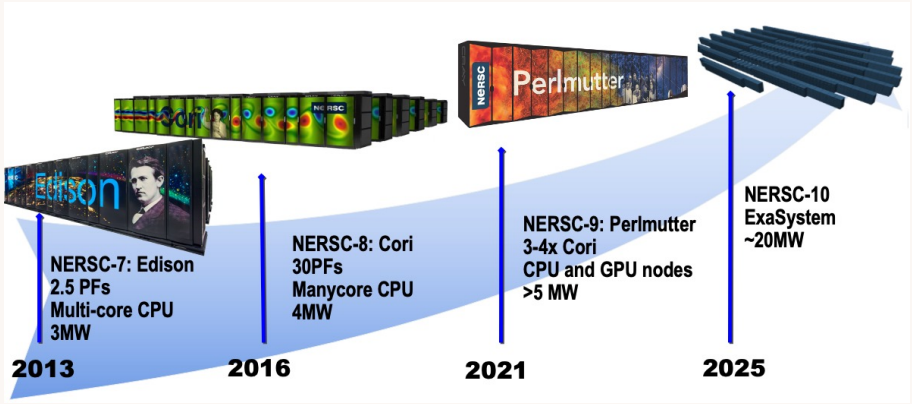
2017 NERSC ERCAP survey:
Do your codes use any of the following?



328 users responded

OpenMP Can Provide Programming Continuity

NERSC wants user code changes to be beneficial over several generations of NERSC supercomputers and portable to other DOE supercomputers



https://www.alcf.anl.gov/aurora

https://www.olcf.ornl.gov/frontier

OpenMP on Perlmutter: What Were the Challenges in 2018?

- OpenMP had not yet demonstrated broad success on GPUs
  - NERSC staff who had used OpenMP on GPUs almost always encountered compiler issues
- No vendor compiler was planned to support OpenMP GPU-offload on Perlmutter
- There was only a 3 year time-frame before Perlmutter would be made available to users

Plan: Engage in an NRE contract with NVIDIA to deliver a GPU-enabled OpenMP compiler

NERSC's Definition of OpenMP Compiler Success on Perlmutter

https://www.openmp.org

- Robust support for large production applications written in C, C++ and Fortran
- Ability to obtain high performance on GPUs without extensive low-level tuning
- Sufficient evidence for NERSC staff to confidently recommend OpenMP to NERSC users

Building a Successful Contract

OpenMP-5.0 specification
https://www.openmp.org

```
#pragma omp target teams distribute
for (j=1; j<n-1; j++)
```

Pre-existing and emerging
DOE / community applications

A Production-ready
GPU-enabled
OpenMP compiler

Years of experience implementing
OpenACC for NVIDIA GPUs and
knowledge of how to best
optimize applications

NERSC application readiness
program for Perlmutter: NESAP-2

Key Elements and Execution of the Contract

- Aggressively prioritize OpenMP features based on application requirements
  - The OpenMP specification is large and not all features are well suited to GPUs
- Spend sufficient time to find the right balance between the cost of implementing and maintaining a feature versus potential value to applications
  - Drive discussions using application use cases and code fragments
  - Embrace newer OpenMP features, such as the “loop” directive, which can deliver productivity and performance advantages
- Have weekly meetings to discuss experiences using the compiler, find solutions to problems, and prioritize features and bug fixes

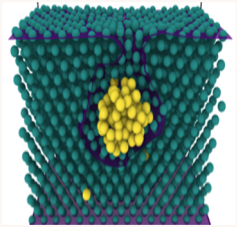
Features Prioritized Based on Representative Apps

- We added an early milestone to create a test suite of representative applications
- The test suite was expanded in a later milestone and now consists of 312 tests:
  - 206 SOLLVE V&V tests
  - 15 SPEC ACCEL tests
  - 42 RAJA Performance Suite tests
  - 49 benchmark and proxy application tests
- The test suite exercises a wide variety of OpenMP features and interoperability use cases anticipated in production applications
  - The codes remained the same, enabling us to track compiler progress over time

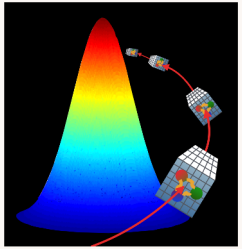
Features Also Prioritized Based on Co-Design of NESAP-2 Apps

- We added an early milestone to develop an OpenMP target offload porting plan for five NESAP-2 applications
  - The codes changed over time, requiring continual re-assessment of OpenMP compiler capabilities

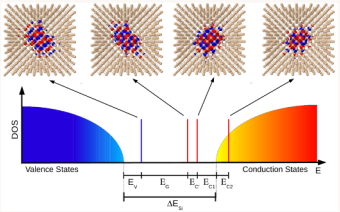
LAMMPS
Molecular Dynamics
C++



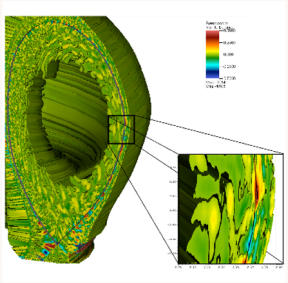
Dslash
Lattice QCD
C++



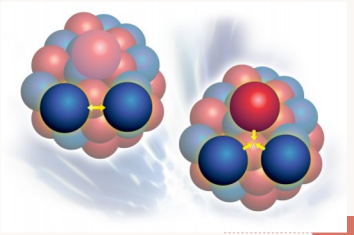
BerkeleyGW
Material Science
Fortran



XGC and GENE
Nuclear Fusion
C++ and Fortran



MFDn
Nuclear Physics
Fortran



The Test Suite Tracked Compiler Correctness

Some highlights of the initial test suite:

Test Name	Languages
SOLLVE V&V Suite	C, C++, Fortran
SPEC ACCEL 1.2	C, Fortran
Stream	C, C++, Fortran
BerkeleyGW-GPP	C++
Toypush	Fortran & C
ElectromagneticPIC	Fortran & C++
miniQMC	C++

- The test suite consists of diverse benchmarks in C, C++ and Fortran
- The final milestone requires that all benchmarks execute correctly on Perlmutter

The Test Suite Tracked Compiler Performance

Some highlights of the initial test suite:

Test Name	Assess performance
SOLLVE V&V Suite	
SPEC ACCEL 1.2	✓
Stream	✓
BerkeleyGW-GPP	✓
Toypush	✓
ElectromagneticPIC	✓
miniQMC	

- Many applications in the test suite have implementations in OpenMP and OpenACC enabling us to assess performance
- The final milestone requires that the OpenMP version achieves **>= 90%** of the performance of the OpenACC version

The Test Suite was Expanded Part Way Through the Contract

Additions were driven by NERSC/DOE challenges using OpenMP target offload in representative applications:

Test Name	Languages	Assess performance	Challenge
GAMESS-rimp2	Fortran	✓	CUDA library interoperability
HPGMG	C	✓	Data management
RAJA Perf. Suite	C++		Data management
SU3_Bench	C++	✓	std::complex variables
TestSNAP	C++		Data management

NVIDIA HPC SDK Performance Results

- The next slides show microbenchmark and NESAP-2 proxy app results
- Perlmutter was not accepted when we collected results for this publication
- Results are obtained on an NVIDIA DGX development system which has the same GPUs as Perlmutter
  - Each node consists of 2 AMD Rome CPUs and 8 NVIDIA A100 GPUs
  - All results used a single process on 1 CPU and 1 GPU
- All results used NVIDIA HPC SDK 21.7 unless otherwise stated

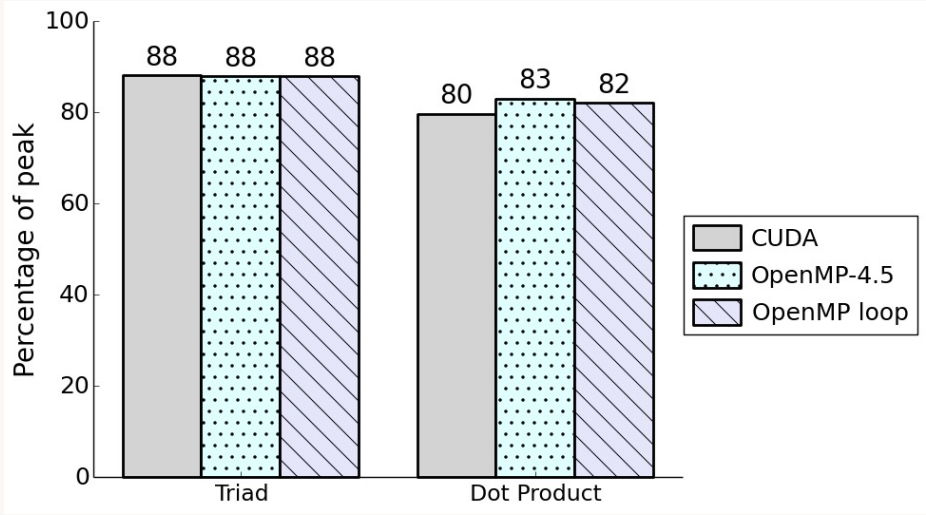
Stream Microbenchmark (Babelstream)

We considered 2 Stream kernels using CUDA, OpenMP-4.5 and OpenMP loop

```
// Triad
for (i = 0; i < array_size; i++)
  a[i] = b[i] + scalar * c[i];

// Dot Product
for (i = 0; i < array_size; i++)
  sum += a[i] * b[i];
```

Higher is better

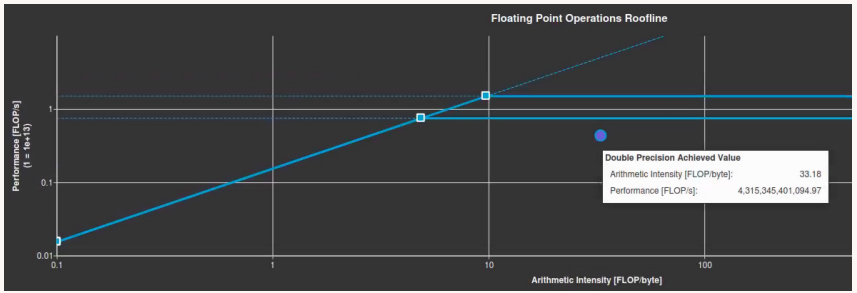


- Similar performance between both OpenMP implementations and CUDA
- The Dot Product kernel is slightly faster with OpenMP because of the high performance data reduction in the NVIDIA compiler

BerkeleyGW GPP Mini App

- **Team plan**: migrate BerkeleyGW from Fortran + C interface + CUDA to Fortran OpenMP target offload
- The Fortran GPP mini-app captures one of the hot spots in BerkeleyGW

GPP obtains 4.3 TF/s on one A100 GPU
(44% of the DP peak without tensor cores)



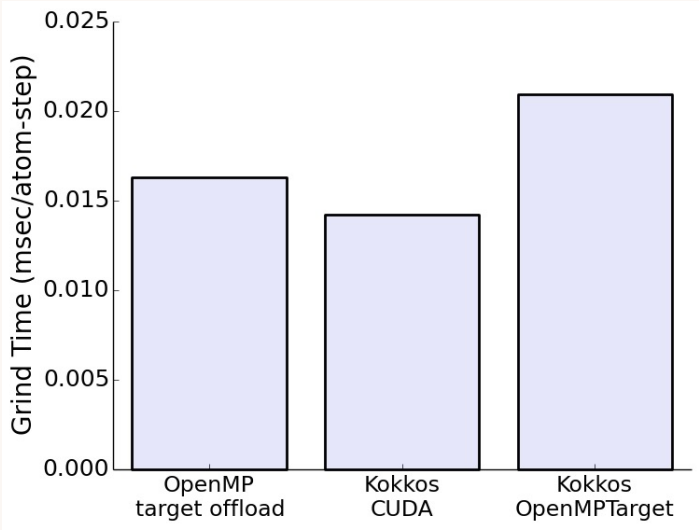
Success story: The NVIDIA compiler:

- Provides significant support for Fortran applications
- Generates high quality code even for highly compute-bound kernels

LAMMPS TestSNAP Mini App

- **Team plan**: use Kokkos, but have a reliable Kokkos OpenMPTarget backend as fallback

Lower is better



- Native use of OpenMP target offload is competitive with Kokkos CUDA backend
- Performance advantage of Kokkos CUDA backend versus OpenMPTarget backend is under investigation
- **Success story**: The NVIDIA compiler successfully compiled and executed representative apps that use the Kokkos OpenMPTarget backend

(Kokkos OpenMPTarget result obtained with 21.3 compiler)

Other Successful Outcomes

Test Suite: The first production compiler in the contract (NVIDIA HPC SDK 21.5) obtained a 98% pass rate and achieved >= 90% performance requirement on NERSC's Cori-GPU system

Educating NERSC users: Training deliverable had over 200 users in attendance

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

Demonstrating production-readiness: Mutually-beneficial participation in an ECP hackathon:

https://sites.google.com/view/ecpomphackjan2021

Assessment of NRE Activity

- We had success by prioritizing OpenMP features based on application requirements
- Our initial milestones helped bound the OpenMP feature work and gave us confidence that the final product would be successful
  - Making hard decisions about what features to exclude, e.g. OpenMP tasks, locks and ordered regions on the GPU, kept us on track
- There was no need to change project scope, only slight tweaks:
  - We prioritized compiler stability and performance over features in one milestone
  - We refined OpenMP code specialization features (metadirective and declare variant) according to updates in the OpenMP-5.1 specification
- The latest NVIDIA compilers are available on NERSC systems and are working well
- The contract is nearly complete but we expect collaboration to continue, albeit in a slightly different form

Our Best Practices for NRE Contracts

- Invest in early milestones to address any uncertainty, e.g.
  - How to measure success in later milestones? …create a test suite
  - How to ensure the appropriate work is prioritized? …spend time justifying the subset of features based on application requirements
- Meet often to understand each other's perspective and establish a strong relationship
  - Keeps the focus on the highest priority areas
  - Encourages participation in mutually-beneficial activities outside of the contract, e.g. intermediate compiler drops and hackathon activities improved the compiler deliverables
- For the HPC Center: Make sure an NRE activity is beneficial
  - Perform a cost-benefit analysis of short-term benefits as well as long-term benefits beyond the lifetime of a single supercomputer
- For the HPC Center: Tailor the milestones to your unique needs
  - 1:1 user training and support is attractive if supporting fewer users and codes

Acknowledgements

- The NRE activity was funded by Subcontract 7437777. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.
- Many people contributed to the effort to add OpenMP target offload support to the NVIDIA HPC Compilers. This paper's authors from NVIDIA give heartfelt thanks to the following colleagues: Simone Atzeni, Evgeny Baskakov, Mathew Colgrove, Sebastien Deldon, Morris Hafner, Zahra Khatami, Mark LeAir, Brent Leback, Jie Li, Razvan Lupusoru, Pallavi Mathew, Scott McMillan, Doug Miles, Dave Parks, Adam Smith, Mahdi Soltan Mohammadi, Zhen Wang, Michael Wolfe, and Yu You.
- The authors from NERSC would like to thank Charlene Yang, Mauro Del Ben, Dhruva Kulkarni, Brandon Cook and Douglas Doerfler for contributing NESAP-2 application test cases and Brian Friesen for installing NVIDIA compilers on NERSC systems.