Application Readiness for NERSC Cori
Zhengji Zhao¹, Scott French¹, Jack Deslippe¹, Mathias Jacquelin², Brian Friesen¹, and Helen He¹
¹) National Energy Scientific Computing Center at Lawrence Berkeley National Laboratory
²) Lawrence Berkeley National Laboratory

NERSC Exascale Science Application Program (NESAP)

Preparing NERSC science applications for Cori
NERSC’s next supercomputer, Cori, will begin to transition from workload to more energy efficient architectures
- Gray HD system with over 9300 Intel Knights Landing compute nodes
- Teraflour model: 5200 nodes (2/3 of core); Larger vector units (512 bit)
- On-package high-bandwidth memory

NERSC Exascale Science Application Program
- 20 application code teams were selected to work with Cray, Intel and NERSC staff (August, 2014)
- The focus is on high-performance computing, including: Access to resources and other access including “dungeon sessions” with Intel and Cray Center of Excellence, early access to HPC “testbeds” systems, early access and time on Cori
- Funding available for the NESAP Postdoctoral Fellows to work directly with application teams
- Many IPIs discussed optimization strategies have been explored, and here we will highlight our efforts and achievements with a few selected NESAP codes

Additional NESAP readiness and support efforts
- NERSC Node and/or organizes various training exams for users and developers, e.g. OpenMP Vectorization, MPI-3, Intel and Cray profiling and optimization tools, Intel and Cray compiler optimizations
- Hosting code optimizations hackathons for NERSC users and application developers
- Intensive participation in the Intel 5200 FEIN User Group (MPUG)
- Involved in the OpenMP Committee

BoxLib: Block-structured AMR Framework

Developed by Center for Computational Sciences and Engineering at LBNL
NERSC PI: Ann Almgren (LBNL) NERSC Posdoc: Brian Friesen

- Designed for numerical solution of PDEs on distributed, structured grids
- Hybrid MPI/OpenMP parallelism

Traditional threading model
- Domain decomposed into boxes, distributed among MPI tasks, each with in threads
- Each thread operates on a subset of boxes (Fig. 1)
- Leads to load imbalance with large # of boxes (Fig. 2)
- Exp. in AMR where box distribution among MPI tasks is uneven

Tiling threading model
- Iteration space within each box is divided into smaller "tiles" which are distributed among threads
- Tile size specified by user; contiguous in unit-dimension for optimal caching
- Tile-level threading reduces "surface area-to-volume" ratio of memory hails to FP data for distributed worksets

Optimizations lead to significant performance improvements
- Threading enables us to use all 240 hardware threads
- Optimal code performs 2.5X better than baseline
- Up to 60X better compared to 1 MPI Task

Summary
- Efficient implementation of strong scalability is up to ~60% on 32 threads on Skylake (PH) tested at NERSC
- Solves close memory bandwidth at ~28 TFLOPs/s bandwidth and scalar floating point operations
- Even with only 1 thread, performance with tiling is still better due to improved locality within the loop iteration space

BerkeleyGW: Materials Science Applications

Developed at LBNL/JUCB. Used as “prototypes” for App Readiness at NERSC.
NERSC PI: Jack Deslippe (NERSC)
NESAP PI: Jack Deslippe (NERSC)

Significant Bottleneck is large matrix reduction operations.

Optimization steps:
1. Target code on node parallelism (MPI-model already falling away)
2. Ensure key loop parallelism can be vectorized

Additional BerkeleyGW optimizations
- Using more aggressive compiler flags
- Replace division with inversion of multiplication
- Use intrinsics for vectorization
- Use internal GAMMA function
- Use !$OMP SIMD ALIGNED to force memory alignment

System based on the OpenMP thread model version 4.5

EMGeo: Geophysical Imaging Applications

Developed by researchers at LBL Earth Science Division
Project PI: Gregory Newman (ESD); NESAP Liasison: Scott French (NERSC)

EMGeo (EM Imaging and Seismics imaging) dominated by Krylov solver (> 90% of wallclock)
- Different solver methods (288 vs. ESP), but similar motifs

Optimization Steps:
- Adding OpenMP to solvers (starting with ESP)
  - Overall thread scaling very good on Knights Corner
- Adding memory bandwidth saturation
  - Solves dominated by SPG, STRUMPACK tests, etc.
- Focusing initial architectural optimization experiments on complex*16 ELLPACK SpMV
  - Many potential optimizations techniques to KNL - Aligned vectors, Loop unrolling, unrolling, Memory layout optimizations, Fourier "SMP solvers"
- Developed SpMV kernel variants that open the space of variability
- Hardware profiling when using XLU access
- Identify the candidate for HBM - Use vector, tune for HBM in a dual socket system

CESM: Community Earth System Model

Developed by Center for Computational Sciences and Engineering at LBNL
NERSC PI: John Dennis (NCAR). NERSC Liaison: Helen He

MG2: CESM kernel for radiation transfer workload
- Typically takes >10% of CESM time
- Very little vectorization pipeline dependencies, heavy use of math intrinsics
- Optimizing CSEMs performance and scalability

Prototype Periodic-Diagonalization

Preliminary performance results
- Tiling implementation strong scalability efficiently up to ~250 threads on Skylake (PH) tested at NERSC
- Solves close memory bandwidth at ~28 TFLOPs/s bandwidth
- Strong scale more efficiently than non-tiled version up to memory BW saturation point
- Even with only 1 thread, performance with tiling is still better due to improved data locality within the loop iteration space

Acknowledgements
- Intel Inc. NESAP partnership, Cray Inc., Collaboration via Center of Excellence