A Facility's View of User Software Engineering Practices

Richard Gerber
NERSC Senior Science Advisor
User Services Group Lead

September 21, 2015
Facility Software & User Applications

- **Facility-supported software**
  - NERSC supports about 250 different SW packages (not versions!)
  - Not generally developing code, but users expect it to work correctly

- **User-supported software**
  - NERSC has 6,000 users, 700 projects, ~650 “named” application codes
  - Many codes developed and supported by application scientists

- **Common challenges**
  - Vast number of combinations to support: different architectures, compilers, libraries, versions
  - I’ll concentrate on user needs at NERSC as representative of programmatic user needs
Computing Usage at NERSC 2014

NERSC 2014 MPP Usage by Scientific Discipline

- **Combustion**: 16 Codes (2%)
- **Biosciences**: 77 Codes (4%)
- **Geoscience**: 39 Codes (5%)
- **Astrophysics**: 63 Codes (6%)
- **Climate Research**: 42 Codes (10%)
- **Chemistry**: 103 Codes (12%)
- **Lattice QCD**: 12 Codes (13%)
- **Materials Science**: 150 Codes (19%)
- **Fusion Energy**: 94 Codes (20%)
- **Other**: 7% (remaining codes)

Total Codes: 150
NERSC Usage by Algorithm

- Other codes
- Fusion PIC
- Lattice QCD
- Density Functional Theory
- Molecular Dynamics
- Climate
- Fusion Continuum
- Quantum Chemistry
- QMC
- Fast Math
- CMB
- Bioinformatics
- Accelerator PIC
- Other codes
How Do We Know What Users are Doing and Need?

• NERSC HPC Requirements Reviews
• Annual User Survey
• ERCAP Allocations Process
• Targeted Emails
• Workshops
• NESAP Application Readiness Program
• Day-to-day Consulting

Spoiler: From a User’s Perspective the Crisis is porting and developing code for next-generation architectures!
Requirements Reviews

• 1½-day reviews with each Program Office
• Computing and storage requirements for next 5 years
• Participants
  – DOE ADs & Program Managers
  – Top NERSC users, CS
  – NERSC staff
• We tried to listen, so the requirements come from the application scientists themselves and reflect their top concerns

http://www.nersc.gov/science/hpc-requirements-reviews/reports/
Priority Needs Summary Across all Offices from Requirements Reviews: Target 2017

• More hours and storage
  – Continued exponential increase
  – Science already limited by lack of resources

• Application readiness
  – Help porting & developing code for manycore
  – Need for supported software applications, libraries, tools

• Jobs at all scales
  – Need to run at largest scales
  – High throughput workflows (e.g., for data analysis)
  – Ensemble runs for V&V, statistics, & exploration
Priority Needs Across all Offices

• High performance, available, stable systems
  – “Leading but not bleeding”
  – Availability and stability crucial for throughput
  – Expensive to deal with job failures & workflow interruptions

• Science in Data
  – Data storage, I/O bandwidth, data management tools
  – Support for data analytics, workflows, analysis of experimental data, integration with simulations
• ASCR researchers will require software applications, libraries, and tools that will run efficiently on many-core architectures.
  – Many existing software packages need to be ported, optimized and supported on next-generation platforms like Cori.
  – In order to provide applications, libraries, and tools to users that run efficiently on many-core architectures, software developers need sustained support from DOE.
  – Development and measurement tools are needed to optimize codes on new systems.

• Code teams need help porting applications, libraries, frameworks, and tools to run well on next-generation architectures.
Enabling and maintaining scientific productivity, while still advancing the state of the art, is required.

Quantifying uncertainty in simulations and validating those simulations to much higher precision is becoming required. This will increase the demand for additional compute cycles and better workflow management systems.

Larger simulations and data sets may require new approaches (in-situ) to data management, data analytics and scientific visualization. The BER community will need assistance to implement these types of analysis methods.
• Users will need assistance from NERSC to prepare for Cori follow-on manycore systems
  – Scientists need guidance, advice, and training from NERSC to transition codes for efficient computation on manycore systems like Cori.
  – BES researchers depend substantially on third-party ISV and community software and there is an expectation that this software will be available and run well on future systems.
  – Connections with computer science experts are needed to develop new algorithms.

• Research teams need to run complex jobs of many different types and scales.
  – Workflows are becoming more complicated and tools are needed to accommodate this need. Some teams’ workflows involve multiple resources across different sites.
FES Requirements Executive Summary Excerpts

• FES codes will require **updated mathematical and I/O libraries that will run efficiently on next-generation architectures.**
  – Threaded linear solvers are needed for many-core systems. The PETSc library is heavily used on today’s machines and must either be updated or replaced.
  – The HDF5, ADIOS, and MPI I/O libraries must be supported on next-generation platforms.

• **Teams need effective tools for managing workflows, performing data analysis, and profiling and developing codes.**
  – Many codes use Python and other software that requires dynamic loading of modules and libraries.
  – Visualization and analysis tools are needed, including VisIt, Matlab, and AVS-Express.
  – Optimization tools

• **Code teams need help** porting applications to run well on next-generation architectures.
HEP Requirements Executive Summary

Excerpts

• Scientists need vastly improved data I/O rates and better facilities for performing data-intensive science.
  – Research teams need to efficiently process large data sets, perform visualization and analysis on them, and share them among collaborators and the public. Researchers need databases and web portals.
  – The ability to archive and manage data is needed. Data sharing, curation, and provenance must be accommodated.

• NERSC must help enable and ease the transition to next-generation architectures.
  – Since most codes will need to be rewritten or extensively modified to run efficiently on next-generation architectures, HEP research teams will need assistance transitioning their codes.
NP requirements executive summary excerpts

- NP researchers and their teams require assistance getting codes ready for Cori and subsequent advanced architecture systems.
  - While some teams are using GPUs for portions of their work, most of the codes in use have not been ported to run well—or at all—on manycore systems like the Intel Xeon Phi that will comprise the NERSC Cori system, which is scheduled to be deployed in 2016.
  - Scientists want NERSC to provide expertise and manpower to assist the porting effort. This includes documentation, training, and consulting, in addition to dedicated coding help and/or liaisons.
  - Researchers in NP want DOE to provide expert direct assistance to help port to new architectures.

- Application teams need a supported software stack that executes well on next-generation architectures.
  - Scientists want supported software—libraries, tools, solvers—that will run efficiently on Cori and request that DOE support multidisciplinary teams along the lines of what is available through SciDAC.
  - NP codes rely heavily on basic math libraries—LAPACK, SCALAPACK, BLAS, ARPACK, PARPACK, MKL—and researchers need them to work well on Cori and future systems. Some teams are planning to begin using HDF5 for I/O.
Requirements Reviews: Users Want Continued and Expanded SciDAC-like Programs

• We collaborate with a CS researcher funded by the ExaCT Co-design Center and various X-stack projects have interacted with us on these activities. – J. Bell
• Chombo and PETSc support comes by way of the SciDAC FASTMath project.
• We are currently working with two SciDAC3 teams (Roberto Car and John Pask) to integrate PPEXSI with CP2K and DGDFIT (see below). –
• Building on recent successes of SciDAC and the Ice Sheet Initiative for Climate Extremes (ISICLES), we are developing two dycores ... 
• The SciDAC SUPER institute is helping us with all these activities.
• We also have an ongoing effort with ExaCT codesign to explore the Legion dynamic runtime and Domain Specific Languages (DSLs) for taking advantage of the increasingly parallel and heterogeneous architectures available for high performance computing.
Top Team Questionnaire: Describe Your SW Engineering Practices and Needs

- Of those that responded: somewhat surprisingly high level of sophistication (or at least thought)!
  - Some are VERY sophisticated, e.g. CESM climate code
  - Many doing a few things, but want to do more (more on this later)

- Few common tools beyond the old standards (e.g. make, C/C++/Fortran, LAPACK)
  - One exception: Git / GitHub is commonly used for version control and distributing software

- My belief (anecdotal evidence) is that many NERSC projects have little software engineering in place.
Questionnaire: Top Concerns & Needs

- Continued support for third-part SW libraries and applications
- Sustainable software across different and frequently changing architectures
  - Are many branches needed?
- Effort needed to port (manpower)
- Lack of unit and regression testing
- Lack of code review procedures; reactive after problem found
- Need documentation systems
- Collaboration software
- Code (source and binaries) sharing and installation/maintenance tools
- Workflow tools
All These Libraries are Needed!

Library Usage on Edison
(Mar 01, 2014 - Aug. 31, 2014)

- mpich
- darshan
- libsci
- hdf5
- fftw
- shmem
- petsc
- mkl
- openmpi
- metis
- dmapp
- netcdf
- upc
- superlu
- hdf5-para
- parmetis
- superlu_dist
- gsl
- HYPRE
- mumps
- scotch
- tpsi
- python
- rca
- valgrind
- xtpsi
Languages & Programming Models

- C, C++, Fortran heavily and approximately equally used
  - Python becoming popular for both coding and workflows
- MPI, MPI/OpenMP, LLVM/Clang JIT, CUDA, OpenACC, something better and portable

What are you using or have tried as MPI+X or other?
Where are Applications Today?

Is your application ready for:

- Cori in general
- On-package memory
- Vectorization
- OpenMP

NERSC User Survey
How Satisfied are You With Current Software?

Software Satisfaction Questions

NERSC User Survey
NERSC Software Support

• NERSC supports 250 SW packages
• Challenges and need
  – Many versions
  – Many compilers
  – Heterogeneous systems (Cori Haswell/KNL)
• SW management
  – Many codes teams ahead of us
  – Drastic need to implement SW engineering practices for installation, support, sustainability, management, security
  – DOE HPC facilities should collaborate on tools and practices
    • spack? smithy? easybuild?
    • V&V, unit and regression testing
Summary

• There are hundreds of codes (probably more) being used by Office of Science researchers.
• Codes are not ready for next-generation HPC systems.
• Office of Science researchers want & need help preparing their codes for these systems.
• They rely heavily on third-party libraries, tools, applications.
• There are large and diverse software needs and dependencies.
• They are very worried about application portability and sustainability and the manpower needed to support their codes.
Extra Slides: Comments and Case Studies from NERSC Requirements Reviews
Case study: John Bell (LBL), Reacting Flows

• **Software we need:**
  - MPI / OpenMP / C++ and F90 / Visit / htar / hypre. We also need better performance analysis tools and an improved programming model to support multicore.

• **We believe NERSC needs to:**
  - Provide high quality tools needed to make this transition
  - Support development of new programming models needed to effectively implement algorithms on these types of architectures

• **We believe DOE and ASCR need to:**
  - Continue to fund applied math research groups working to develop algorithms for these architectures
  - Provide support for software developed by these groups to facilitate availability of libraries / frameworks on new architectures
Case Study: C. Yang, Electronic and Nuclear Structure

• We have no strategy for exploiting the above technologies because we do not know if any of them is sustainable. We have limited human resources and cannot afford to rewrite codes that will become obsolete in a few years.

• We believe the role that NERSC should play is in porting existing codes, performing benchmark tests, and sharing experiences by giving presentations and providing tutorials.

• We believe the role that DOE and ASCR should play is by providing sufficient and sustained funding for algorithm development.
• We would like to see NERSC provide, at the very least, training and support for users making the transition to these architectures. - Ng

• We strongly believe that DOE/ASCR needs to better support development of programming libraries, tools, and algorithms. - Ng

• The role of DOE and ASCR in this work would involve funding of algorithms R&D, access to emerging systems, researcher training on new systems. – Heroux

• NERSC could probably help with n-node performance models, performance tools for threads, GPUs. DOE and ASCR needs to provide sustained funding for code development and maintenance. - Li
Additional Comments from Requirements

Reviews

• We would like to see an improvement in the way that changes to NERSC system software are communicated to users so that we can better associate differences we observe with NERSC changes.

• A key challenge: the logistics of coupling multiple codes, already each of which is highly complex, in a multiscale simulation workflow. We are loosely coupling the SPH and STOMP simulators using the SWIFT workflow management tool and a series of custom python scripts. - Scheibe

• Workflow and data management will be critical to our hybrid simulations, as they will involve running many individual simulations in coordinated manner, with exchange of information between different simulations either through loose coupling (file I/ O, current approach) or tight coupling (direct communication, potential future approach).
Additional Comments from Requirements
Reviews

• Given the growing computational demands of bioinformatics, it is critical that the underlying tools keep pace with the technology. While JGI has a number of expert staff that could contribute to this effort, the problem is much larger than one center. A coordinated effort, such as SciDAC, is needed to make progress on this front. – Rubin

• While many of the current applications are threaded, the current implementations are typically inefficient. Significant investment is needed to improve these implementations and prepare them to fully exploit emerging many-core architectures. – Kbase
Additional Comments from Requirements Reviews

• ... due to a shortage of human resources and the requirements to publish, we focus only on what we absolutely must change to obtain good performance on installed and upcoming computer systems... - P. Kent

• We plan to standardize on open community codes as much as possible, to avoid the difficulties of contributing to proprietary codes and to benefit from international contributions. We hope NERSC focuses effort on open codes and actively steers users towards them. – P. Kent

• We developed our own general workflow software (“FireWorks”, http://pythonhosted.org/FireWorks) targeted largely at running high-throughput scientific workflows at HPC centers. – A. Jain

• Fault tolerance is a concern

• In sum, we will engage in significant reprogramming of our software to take advantage of specialized and heterogeneous computing environments when there is a clear benefit to performance in our computations. – G. Newman
Additional Comments from Requirements Reviews

• In addition to these hardware-driven changes to programs, we also starting to use more flexible event driven programming models, such as run-time systems like CometCloud. NERSC should be working with their users and computer scientists who are using event driven programming models to make sure that these new classes of simulations will be able to run on NERSC computers. – E. Bylaska

• For AMR codes, it is probably fair to say that efficient implementations on next-generation architectures are still a research problem. A key issue here is code portability. For a complex code, it will be unlikely that it will be worth the effort to restructure it so that it runs well on a single architecture. – S. Habib
Additional Comments from Requirements Reviews

• Hopefully, then, many of the programming difficulties inherent in highly optimizing our codes for attached (i.e. through a PCI bus) accelerators can be bypassed into a single framework for spawning tasks/threads amongst the various components. Ideally this framework will be transparent to the programmer. – J. Borrill
Thank you.