Overview: Software Productivity Challenges for Extreme-Scale Science

Lois Curfman McInnes, Mike Heroux, Hans Johansen
In collaboration with IDEAS project members
Outline

- **Confluence of trends: complexity!**
  - Multiphysics and multiscale simulations
  - Emerging extreme-scale architectures

- **Software productivity crisis**
  - Recent DOE activities: SWP for extreme-scale science
  - Related community efforts

- **Introduction to IDEAS project**
  - Interoperable Design of Extreme-scale Application Software
New architectures provide unprecedented opportunities for new science

**Multiphysics:** greater than 1 component governed by its own principle(s) for evolution or equilibrium
- Also: broad class of coarsely partitioned problems possess similarities to multiphysics problems

“The great frontier of computational physics and engineering is in the challenge posed by high-fidelity simulations of real-world systems … typically characterized by multiple, interacting physical processes (multiphysics), interactions that occur on a wide range of both temporal and spatial scales.”

— *The Opportunities and Challenges of Exascale Computing*, ASCAC

A. Hakim, PPPL
E. Kaxiras, Harvard
E. Myra, Univ. of Michigan
K. Evans, ORNL
Software productivity challenges permeate HPC multiphysics, multiscale applications

IJHPCA, Feb 2013: Special issue

- Fluid-structure interaction
- Fission reactor fuel performance
- Reactor core modeling
- Crack propagation
- Fusion
- Subsurface science, hydrology
- Climate
- Radiation hydrodynamics
- Geodynamics
- Accelerator design

doi:10.1177/1094342012468181
Flexible multiphysics/multiscale software is essential

We must fundamentally rethink approaches to multiphysics models, algorithms, and solvers with attention to data motion, data structure conversion, and overall application design.

Challenges:

- Enabling the introduction of new models, algorithms, and data structures
- Addressing CS issues for coupled codes, e.g.,
  - mapping codes to machine topologies
  - load balancing
  - resilience strategies
- Competing goals of software interface stability and software reuse with the ability to innovate algorithmically and develop new physical models
- Composability, sharing methods and code, common infrastructure

“*The way you get programmer productivity is by eliminating lines of code you have to write.*”

Increasing complexity of extreme-scale architectures

Machine peak flops grow steadily … … but it has not come from clock speed (10+ years ago)

Peak has come from core counts (MPI parallelism) and accelerators (threads) … … and more programming difficulties with distributed memory and communication.

Reference: P. Kogge (Notre Dame), John Shalf (LBNL), preprint
# Software engineering and HPC: Efficiency vs. other quality metrics

The table below shows how focusing on the factor below affects the factor to the right:

<table>
<thead>
<tr>
<th>How focusing on the factor below affects the factor to the right</th>
<th>Correctness</th>
<th>Usability</th>
<th>Efficiency</th>
<th>Reliability</th>
<th>Integrity</th>
<th>Adaptability</th>
<th>Accuracy</th>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctness</td>
<td>↑</td>
<td></td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td></td>
<td></td>
<td>↓</td>
</tr>
<tr>
<td>Usability</td>
<td></td>
<td>↑</td>
<td></td>
<td>↑</td>
<td>↑</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td>↓</td>
<td></td>
<td></td>
<td>↓</td>
<td>↓</td>
<td>←</td>
<td></td>
<td>↓</td>
</tr>
<tr>
<td>Reliability</td>
<td>↑</td>
<td></td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td></td>
<td></td>
<td>↓</td>
</tr>
<tr>
<td>Integrity</td>
<td></td>
<td></td>
<td>↓</td>
<td>↑</td>
<td>↑</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptability</td>
<td></td>
<td></td>
<td></td>
<td>↓</td>
<td>↑</td>
<td></td>
<td></td>
<td>↓</td>
</tr>
<tr>
<td>Accuracy</td>
<td>↑</td>
<td></td>
<td>↓</td>
<td>↑</td>
<td>↓</td>
<td>↑</td>
<td></td>
<td>↓</td>
</tr>
<tr>
<td>Robustness</td>
<td>↓</td>
<td></td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
<td>↑</td>
<td></td>
<td>↓</td>
</tr>
</tbody>
</table>

Source: **Code Complete**  
Steve McConnell
Confluence of trends

- **Fundamental trends:**
  - Disruptive HW changes: Requires thorough algorithm/code refactoring
  - Demands for coupling: Multiphysics, multiscale

- **Challenges:**
  - Need refactoring: Really, continuous change
  - Modest app development funding: No monolithic apps
  - Requirements are unfolding, evolving, not fully known *a priori*

- **Opportunities:**
  - Better design and SW practices & tools are available
  - Better SW architectures: Toolkits, libraries, frameworks

- **Basic strategy: Focus on productivity**
Recent DOE activities: Exploring crisis in SW productivity for extreme-scale science

- **Pre-history:**
  - DARPA-HPCS, DOE community meetings, SciDAC, SC, ICSE-CSE, etc.
  - Climate and environment NRC reports, FSP planning

- **Summit:**
  - *Extreme-Scale Application Software Productivity*, Feb 2013

- **Whitepaper:**
  - *Extreme-Scale Scientific Application Software Productivity: Harnessing the Full Capability of Extreme-Scale Computing*, Sept 2013

- **Workshop and report:**
  - *Software Productivity for Extreme-Scale Science*, Jan 2014
    - Whitepapers
    - Relevant reading

- **Minisymposium at SIAM PP14:**
  - *Software Productivity for the Next Generation of Scientific Applications* (8 presentations)

Related community efforts

- **WSSSPE: Working towards Sustainable Software for Science**
  - [wssspe.researchcomputing.org.uk](http://wssspe.researchcomputing.org.uk)

- **Software Carpentry**
  - [software-carpentry.org](http://software-carpentry.org)

- **CSE15 Minitutorial: Lab Skills for Scientific Computing,** Greg Wilson
  - MT3, MT4: Tues, March 17, 2:15-6:05 pm

- **Software Engineering for Science**
  - [www.se4science.org](http://www.se4science.org)

- **Software Engineering for HPC in CSE (SEHPCCSE)**
- **Software Engineering for CSE (SECSE)**

- **Computational Science Stack Exchange**
Outline

- Confluence of trends: complexity!
  - Multiphysics and multiscale simulations
  - Emerging extreme-scale architectures

- Software productivity crisis
  - Recent DOE activities: SWP for extreme-scale science
  - Related community efforts

- Introduction to IDEAS Project
  - Interoperable Design of Extreme-scale Application Software
Interoperable Design of Extreme-scale Application Software (IDEAS)

**Motivation**
Enable *increased scientific productivity*, realizing the potential of extreme-scale computing, through a new interdisciplinary and agile approach to the scientific software ecosystem.

**Objectives**
Address confluence of trends in hardware and increasing demands for predictive multiscale, multiphysics simulations.
Respond to trend of continuous refactoring with efficient agile software engineering methodologies and improved software design.

**Impact on Applications & Programs**
Terrestrial ecosystem *use cases tie IDEAS to modeling and simulation goals* in two Science Focus Area (SFA) programs and both Next Generation Ecosystem Experiment (NGEE) programs in DOE Biologic and Environmental Research (BER).

**Approach**
ASCR/Ber partnership ensures delivery of both crosscutting methodologies and metrics with impact on real application and programs.

**Interdisciplinary multi-lab team** (ANL, LANL, LBNL, LLNL, ORNL, PNNL, SNL)
- ASCR Co-Leads: Mike Heroux (SNL) and Lois Curfman McInnes (ANL)
- BER Lead: David Moulton (LANL)
- Topic Leads: David Bernholdt (ORNL) and Hans Johansen (LBNL)

*Integration and synergistic advances in three communities* deliver scientific productivity; outreach establishes a new holistic perspective for the broader scientific community.

www.ideas-productivity.org
IDEAS: Interoperable Design of Extreme-scale Application Software

ASCR Co-Leads: Mike Heroux (SNL) and Lois Curfman McInnes (ANL)
BER Lead: J. David Moulton (LANL)

Use Cases for Terrestrial Modeling
Lead: J. David Moulton (LANL)
Tim Scheibe (PNNL)
Carl Steefel (LBL)
Glenn Hammond (SNL)
Reed Maxwell (CSM)
Scott Painter (ORNL)
Ethan Coon (LANL)
Xiaofan Yang (PNNL)

Methodologies for Software Productivity
Lead: Hans Johansen (LBNL)
Roscoe Bartlett (ORNL)
Todd Gamblin* (LLNL)
Christos Kartsaklis (ORNL)
Pat McCormick (LANL)
Sri Hari Krishna Narayanan (ANL)
Andrew Salinger* (SNL)
Jason Sarich (ANL)
Dali Wang (ORNL)
Jim Willenbring (SNL)

Extreme-Scale Scientific Software Development Kit
Lead: Mike Heroux (SNL)
Jed Brown (ANL)
Irina Demeshko (SNL)
Kerstin Kleese-Van Dam (PNNL)
Sherry Li (LBNL)
Vijay Mahadevan (ANL)
Daniel Osei-Kuffour (LLNL)
Barry Smith (ANL)
Ulrike Yang (LLNL)

Outreach and Community
Lead: David Bernholdt (ORNL)
Katie Antypas* (NERSC)
Lisa Childers* (ALCF)
Judy Hill* (OLCF)

Crosscutting Lead: Lois Curfman McInnes (ANL)

SDAs  NGEE  Exascale Co-Design  ASCR Math & CS  ALCF
CLM  ACME  Exascale Roadmap  SciDAC  NERSC  OLCF

BER Terrestrial Programs  DOE Extreme-scale Programs  DOE Computing Facilities

IDEAS project structure and interactions
Use cases: Multiscale, multiphysics representation of watershed dynamics

- **Use Case 1**: Hydrological and biogeochemical cycling in the Colorado River System
- **Use Case 2**: Thermal hydrology and carbon cycling in tundra at the Barrow Environmental Observatory

- Leverage and complement existing SBR and TES programs:
  - LBNL and PNNL SFAs
  - NGEE Arctic and Tropics

- **General approach**:
  - Leverage existing open source application codes
  - Improve software development practices
  - Targeted refactoring of interfaces, data structures, and key components to facilitate interoperability
  - Modernize management of multiphysics integration and multiscale coupling
IDEAS interconnections

- **Use cases:** Drive efforts. Traceability from all efforts
  - But generalized for future efforts

- **Methodologies ("HowTo") for SWP:**
  - Metrics: Define for all levels of project. Track progress
  - Workflows, lifecycles: Document and formalize. Identify best practices

- **xSDK:** frameworks + components + libraries
  - Build apps by aggregation and composition

- **Outreach:** Foster communication, adoption, interaction

- **First of a kind:** Focus on **software productivity**
xSDK focus

- Common configure and link capabilities
  - Initial emphasis: Chombo, hypre, PETSc, SuperLU, Trilinos
  - Approach:
    - Determine common definition of configure arguments, eliminate namespace collisions
    - Develop approach that can be adapted by any library development team for standardized configure/link process
    - Develop testing capabilities to assure configure/link processes continue to work indefinitely

- Library interoperability
- Designing for performance portability
Agile, iterative, and incremental cycles for extreme-scale science

- Issues for extreme-scale software:
  - software engineering
  - science/library testing
  - refactoring
  - performance portability
**Goal:** Put steps in place to encourage adoption and reuse of research libraries, and improve longevity of ASCR software investments through refactoring, componentization.
Outreach and community

- Begin changing the way computational and domain science communities think about software development.
- Training: Bringing *practical* information about techniques and tools to software developers, and advice on tailoring.
  - Targeting BER and broader DOE (via SciDAC, INCITE, ALCC, NNSA, ACTT, etc.) with 2 trainings per year.
- Community development: Online tools to facilitate conversation about productivity issues and solutions.
  - Incrementally build and refine community via outreach to different programs, offices.
- Leverage computing facility liaisons, and team’s existing relationships with other programs for Outreach.
Better software productivity is essential for extreme-scale CSE

- **Better SW productivity can give us better, faster and cheaper**
  - **Better**: Science, portability, robustness, composability
  - **Faster**: Execution, development, dissemination
  - **Cheaper**: Fewer staff hours and lines of code

- **IDEAS project**
  - Enabling production of high-quality science results, rapidly and efficiently
    - Multiscale terrestrial ecosystem science
    - Broadly: DOE extreme-scale scientific apps
  - Delivering first-of-a-kind extreme-scale scientific software ecosystem
    - xSDK
    - SWP methodologies (“HowTo”)
    - Outreach and community

Essential mechanism for progress
- In time of disruptive change
- In presence of multiple design tradeoffs
Coming up …

  - David Bernholdt

- **2:20-2:40 Software Productivity Challenges in Environmental Applications**
  - David Moulton

- **2:45-3:05 Software Productivity Community Input: Concerns and Priorities**
  - Jeff Carver and Mike Heroux