Software Productivity for eXtreme-scale Science (SWP4XS)

Hans Johansen (LBNL), David E. Bernholdt (ORNL), Bill Collins (LBNL),
Michael Heroux (SNL), Robert Jacob (ANL), Phil Jones (LANL),
Lois Curfman McInnes (ANL), J. David Moulton (LANL),
Thomas Ndousse-Fetter (DOE), Douglass Post (DOD), William Tang (PPPL)

SWP4XS Workshop 1/13/2014

Agenda

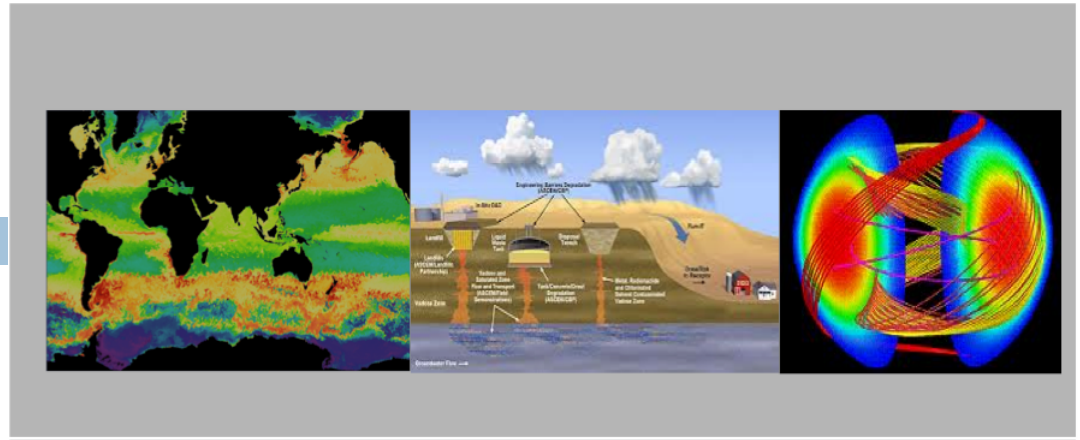
- **Welcome and introduction**
  - Background and goals for this SWP4XS workshop
- **Scientific application software challenges at extreme scale**
  - Extreme-scale architecture trends *(Hans Johansen)*
  - Towards software productivity for computational science *(Mike Heroux)*
- **Software productivity for extreme science applications**
  - Example: DOE Climate community *(Phil Jones)*
  - Cross-cutting application issues *(Lois Curfman McInnes)*
- **Workshop participants and agenda**
  - Review of the agenda
  - Results of Pre-Workshop Survey *(Jeffrey Carver)*

Timeline / background

- Pre-history:
  - DARPA-HPCS, DOE community meetings, SciDAC, SC, ICSE-CSE, etc.
  - Climate and environment NRC reports, FSP planning
- Feb 28, 2013: Extreme-scale application software productivity summit
- May 2013: Working group initiates writing a whitepaper
- Aug 16: Organizing committee kickoff for workshop on "SWP4XS"
- Sept 16: DOE HQ briefing on whitepaper / workshop
- Jan 13-14: Workshop in Rockville
- mid-February: Draft workshop report submission

Summary

- HPC computational science is rapidly approaching a ***productivity crisis*** in scientific application development
- Extreme-scale HPC changes require math/CS innovations AND tools, processes and methodology to effectively use
- Need a strategic vision for software productivity supporting extreme-scale science
- **Goal:** improve DOE's scientific and computing productivity to support long-lived, mission-focused scientific applications



Extreme-Scale Scientific Application Software Productivity:

Harnessing the Full Capability of Extreme-Scale Computing

September 9, 2013

Hans Johansen (LBNL), David E. Bernholdt (ORNL), Bill Collins (LBNL),
Michael Heroux (SNL), Robert Jacob (ANL), Phil Jones (LANL),
Lois Curfman McInnes (ANL), J. David Moulton (LANL),
Thomas Ndousse-Fetter (DOE/ASCR), Douglass Post (DOD), William Tang (PPPL)

Goals for this workshop

Create a SWP4XS research agenda

- Vision for the ***workshop report***
  - Unique research challenges
  - Relevance to extreme science apps
  - External factors/trends to monitor
- What would a ***coherent research program*** contain?
  - What short-term and long-term objectives?
  - What are the biggest priorities?
  - What risks should be addressed?
  - How does this complement other programs, scientific or exascale?

Establish a SWP4XS community

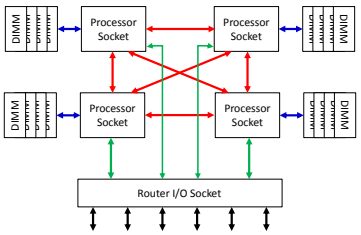
- What efforts to be coordinated across our community?
  - SW req's prioritized for extreme scale (not exclusive to it if applicable)
  - Vs. HW productivity
  - External factors/trends to track?

Agenda

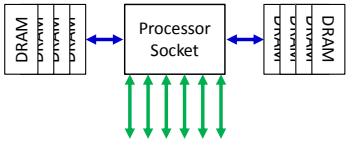
- **Welcome and introduction**
  - Background and goals for this SWP4XS workshop
- **Scientific application software challenges at extreme scale**
  - **Extreme-scale architecture trends *(Hans Johansen)***
  - Towards software productivity for computational science *(Mike Heroux)*
- **Software productivity for extreme science applications**
  - Example: DOE Climate community *(Phil Jones)*
  - Cross-cutting application issues *(Lois Curfman McInnes)*
- **Workshop participants and agenda**
  - Review of the agenda
  - Results of Pre-Workshop Survey *(Jeffrey Carver)*

Performance trends for HPC architectures

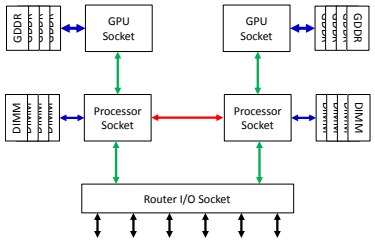
[from P. Kogge (Notre Dame), John Shalf (LBNL), preprint]



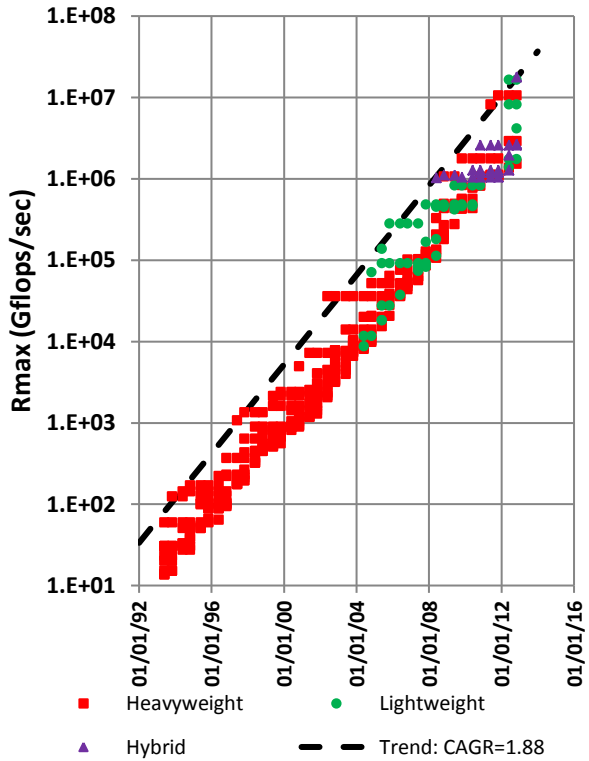
Heavyweight

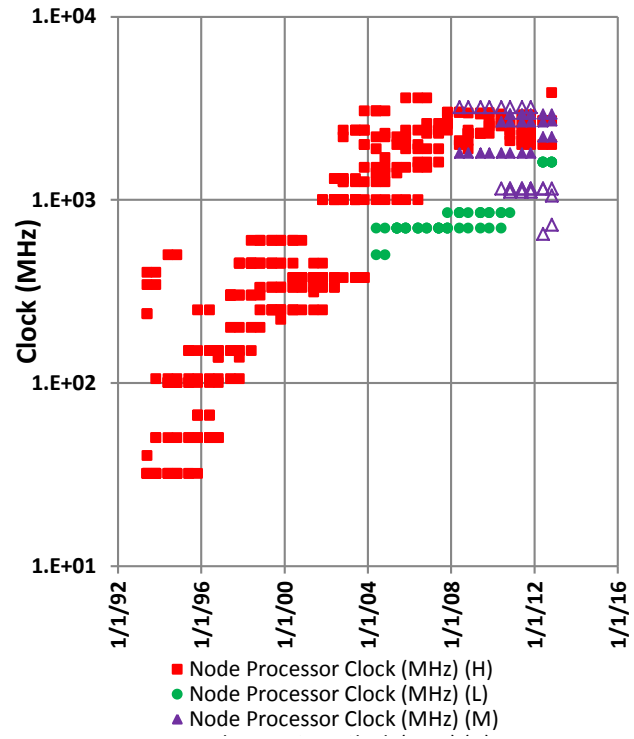


Lightweight



Hybrid



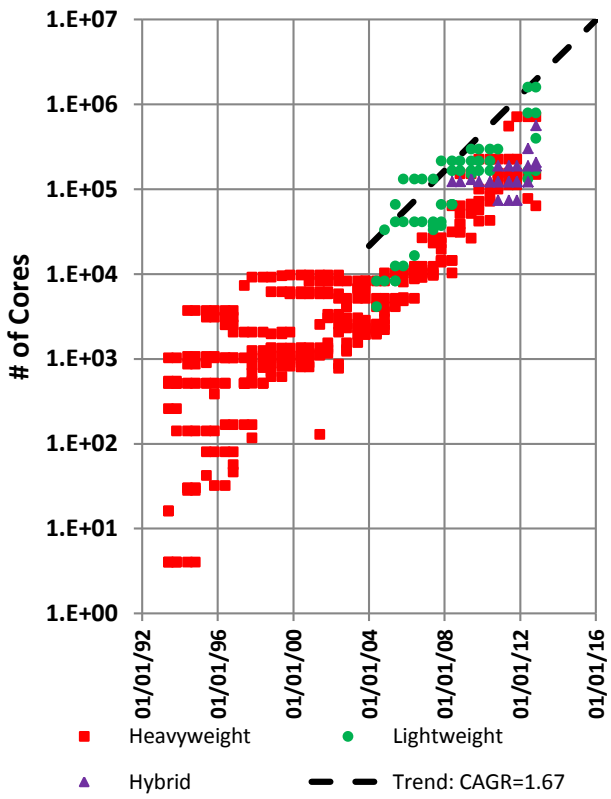


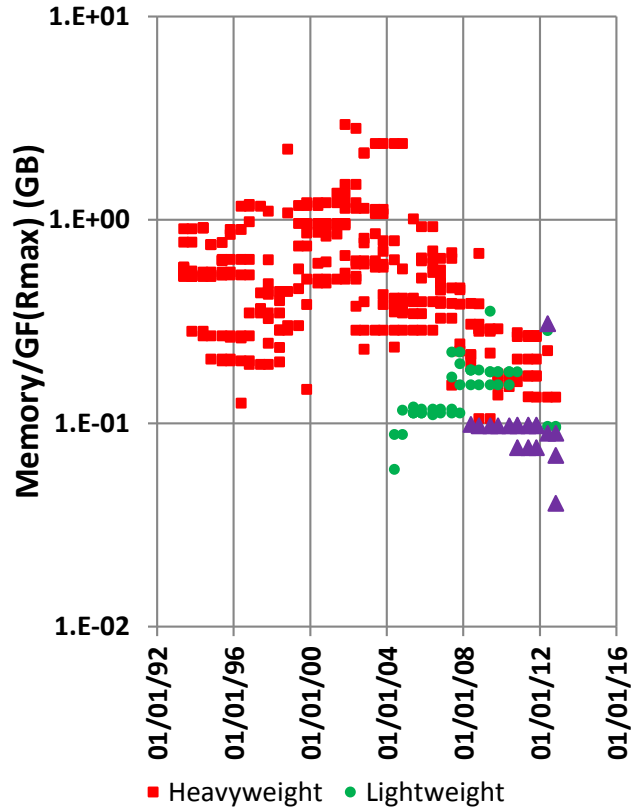
Machine peak flops grow steadily …

… but it has not come from clock speed (10+ years ago)

Future HPC scaling
= heterogeneous parallelism

[from P. Kogge (Notre Dame), John Shalf (LBNL), preprint]

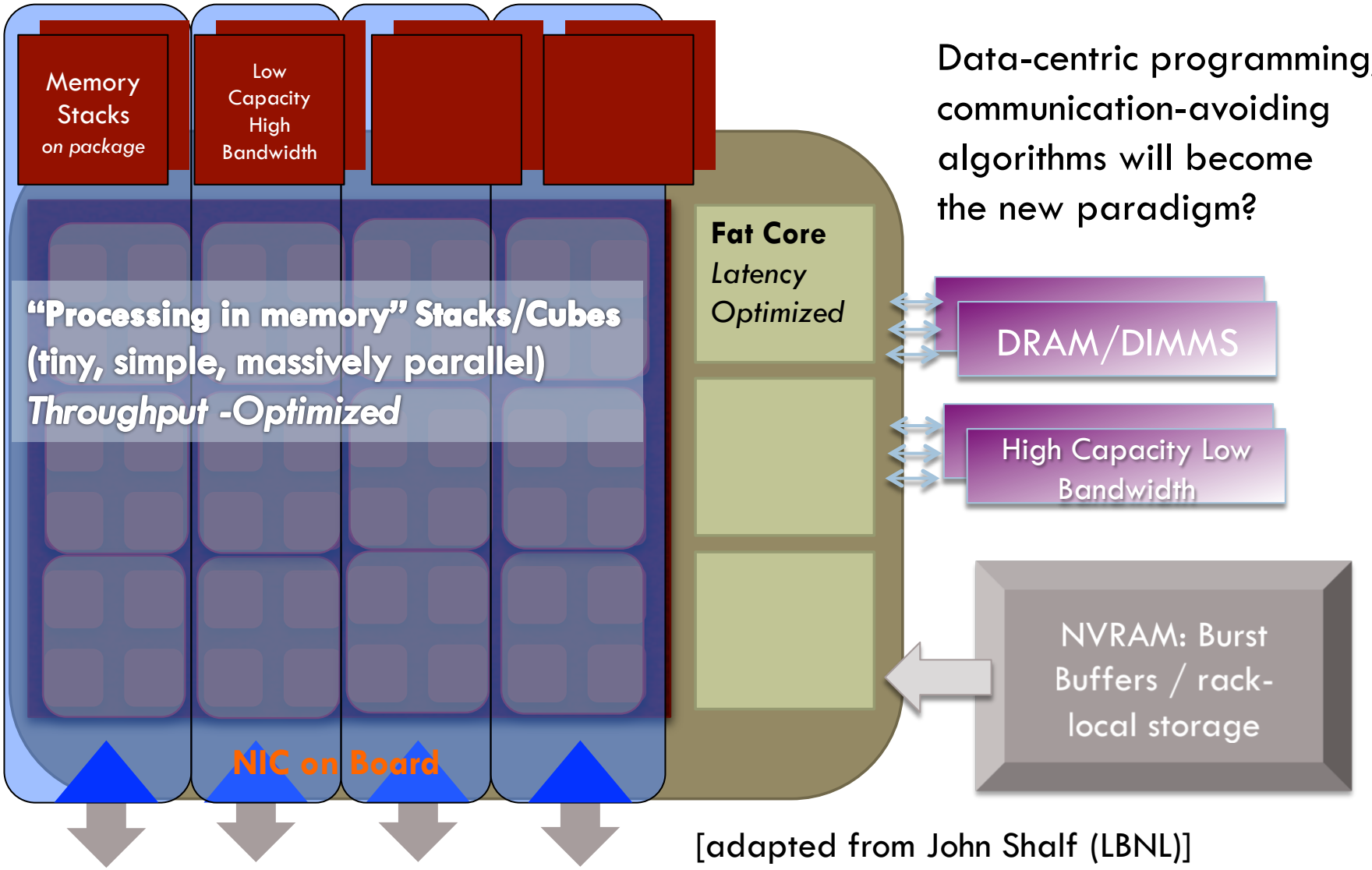




Peak has come from core counts (MPI parallelism) and accelerators (threads) …

… and more programming difficulties with distributed memory and communication.

More heterogeneity, more complexity … more challenges to attain peak performance



Data-centric programming, communication-avoiding algorithms will become the new paradigm?

[adapted from John Shalf (LBNL)]

Commercial innovation in software productivity

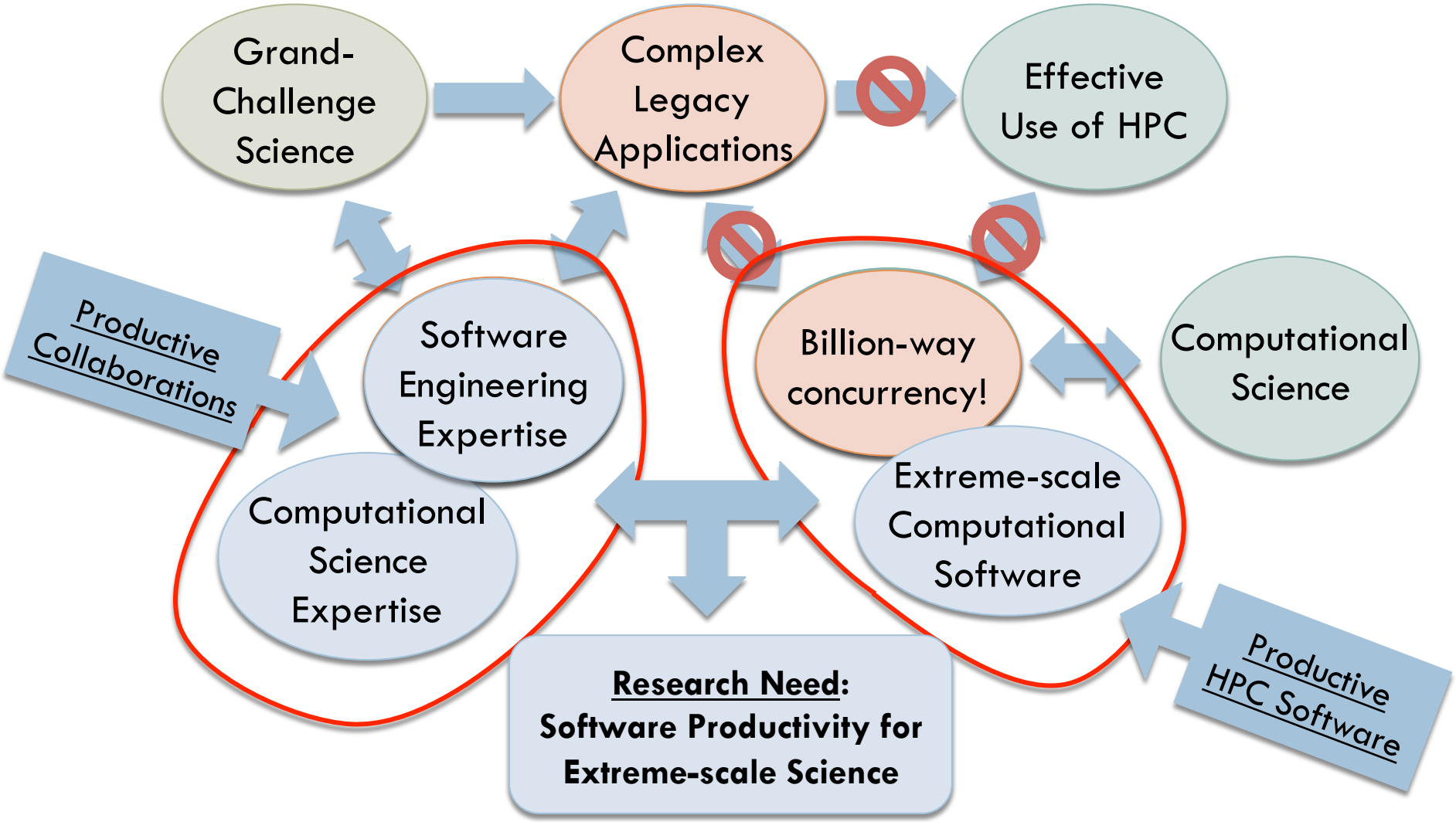
In the past 20 years, extensive commercial software productivity gains came from:

- A transition from "we code everything ourselves" to a **software "ecosystem"**:
  - 80's: Monolithic "waterfall" development evolves to **distributed computing**, TCP/IP, CMM
  - 90's: **Complexity**: layers of API's, frameworks (CORBA, application/db servers, Rational)
  - 00's: **Development innovations** (IDE's, agile, OSS), **run-time** (web services, virtualization)
  - **Today**: Open source ecosystem, cloud computing for everything, agile methodologies

- Recognizing that software productivity relies on development professionals!
  - The profession is not just "CS" – architecture, collaboration, domain expertise, training
  - Management techniques ("methodology"), mechanisms for evaluating goals ("-ilities")
  - Dynamically organizing development teams and tools, constantly evolving and replacing

Commercial software productivity has been transformed, mostly due to its key role in business efficiency, agility and "value"!

Productivity in science fundamentally depends on productivity in software

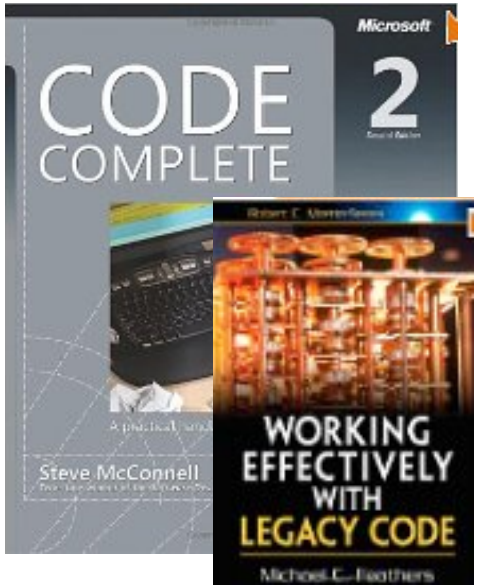


Agenda

- **Welcome and introduction**
  - Background and goals for this SWP4XS workshop
- **Scientific application software challenges at extreme scale**
  - Extreme-scale architecture trends *(Hans Johansen)*
  - **Towards software productivity for computational science *(Mike Heroux)***
- **Software productivity for extreme science applications**
  - Example: DOE Climate community *(Phil Jones)*
  - Cross-cutting application issues *(Lois Curfman McInnes)*
- **Workshop participants and agenda**
  - Review of the agenda
  - Results of Pre-Workshop Survey *(Jeffrey Carver)*

SE for CSE: Recent years, present

- **Agile/Lean principles can work**
  - With discipline, accommodations
    - Sprints great for feature development
    - Must be balanced w/ R&D (longer time cycle)
    - Distributed teams: Extend team-room concept
  - Rigorous V&V required, esp. stand-alone tests
    - Long-lived products
    - Confidence to refactor
- **Community Education**
  - Widely-read material: Common Sensibility
  - Materials exist, not widely know, more needed



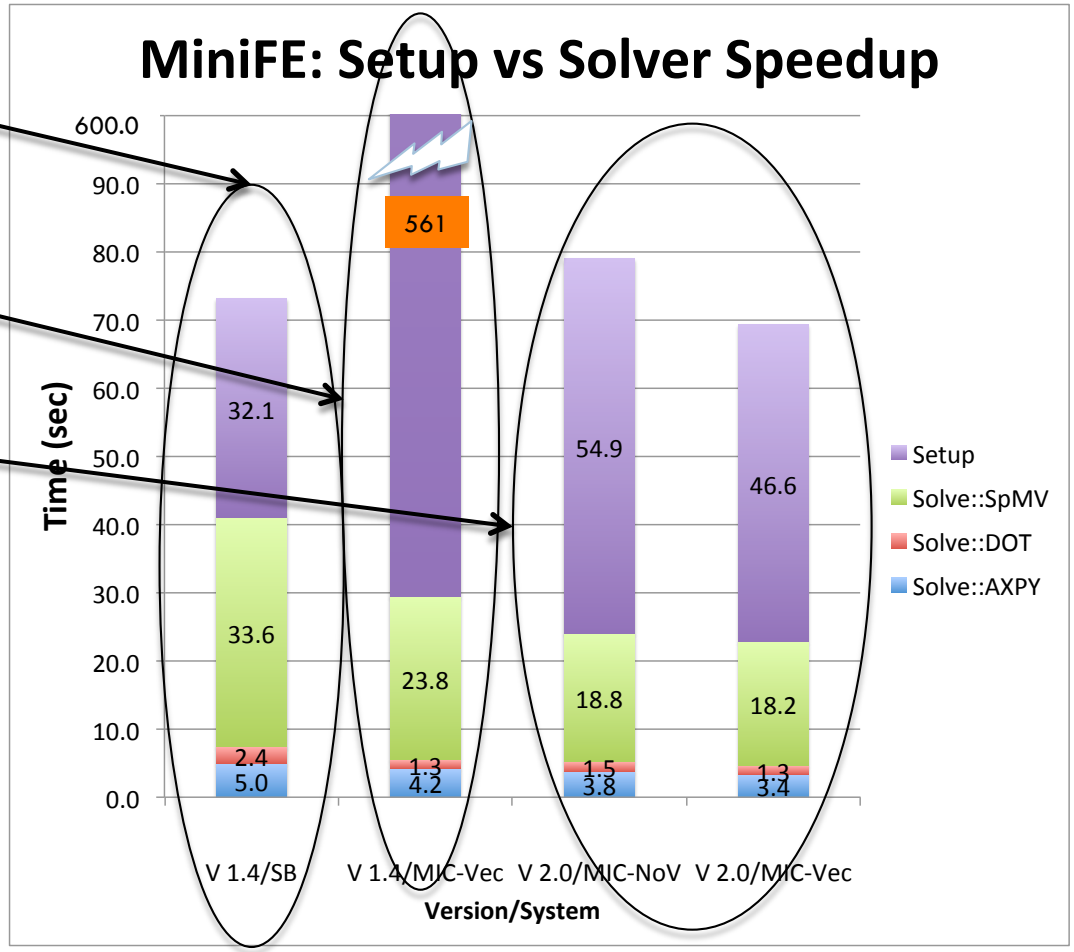
Toward software productivity for CSE

- Some hindering factors for most computational scientists:
  - **Training**: most aren't using modern software practices (awareness, avoidance)
  - **Profession**: career/incentive system isn't available, like it is for science careers
  - **ROI**: SW rewarded for short-term results, not sustained science capabilities
- Key concepts that might benefit computational science:
  - **Software development infrastructure** – especially for extreme-scale computing
  - **Software management workflows** – improved effectiveness for large projects
  - **Verification and validation** – to improve software testing and confidence
  - **Agile research software processes** – coevolution of software and science goals
  - **Group dynamics and management** – human aspects of large, distributed teams
  - **Legacy code refactoring** - transitioning legacy code to new architectures
  - **Multiphysics/multiscale components** – encapsulation, coupling, with performance
- **Overall** focus on:
  - Productivity and quality of large-scale computational science software projects
  - Creating applications that are long-lived investments for science research

The work ahead of us: Threads and vectors

MiniFE 1.4 vs 2.0 as Harbingers

- Typical MPI-only run:
  - Balanced setup vs solve
- First MIC run:
  - Thread/vector solver
  - No-thread setup
- V 2.0: Thread/vector
  - Lots of work:
    - Data placement, const /restrict declarations, avoid shared writes, find race conditions, …
  - Unique to each app
- <u>Opportunity</u>: Look for new crosscutting patterns, libraries (e.g., libs of data containers)



Software Engineering and HPC: Efficiency vs Other Quality Metrics

How focusing on the factor below affects the factor to the right	Correctness	Usability	Efficiency	Reliability	Integrity	Adaptability	Accuracy	Robustness
Correctness	↑		↑	↑			↑	↓
Usability		↑				↑	↑	
Efficiency	↓		↑	↓	↓	↓	↓	
Reliability	↑			↑	↑		↑	↓
Integrity			↓	↑	↑			
Adaptability					↓	↑		↑
Accuracy	↑		↓	↑		↓	↑	↓
Robustness	↓	↑	↓	↓	↓	↑	↓	↑

Source:
Code Complete
Steve McConnell

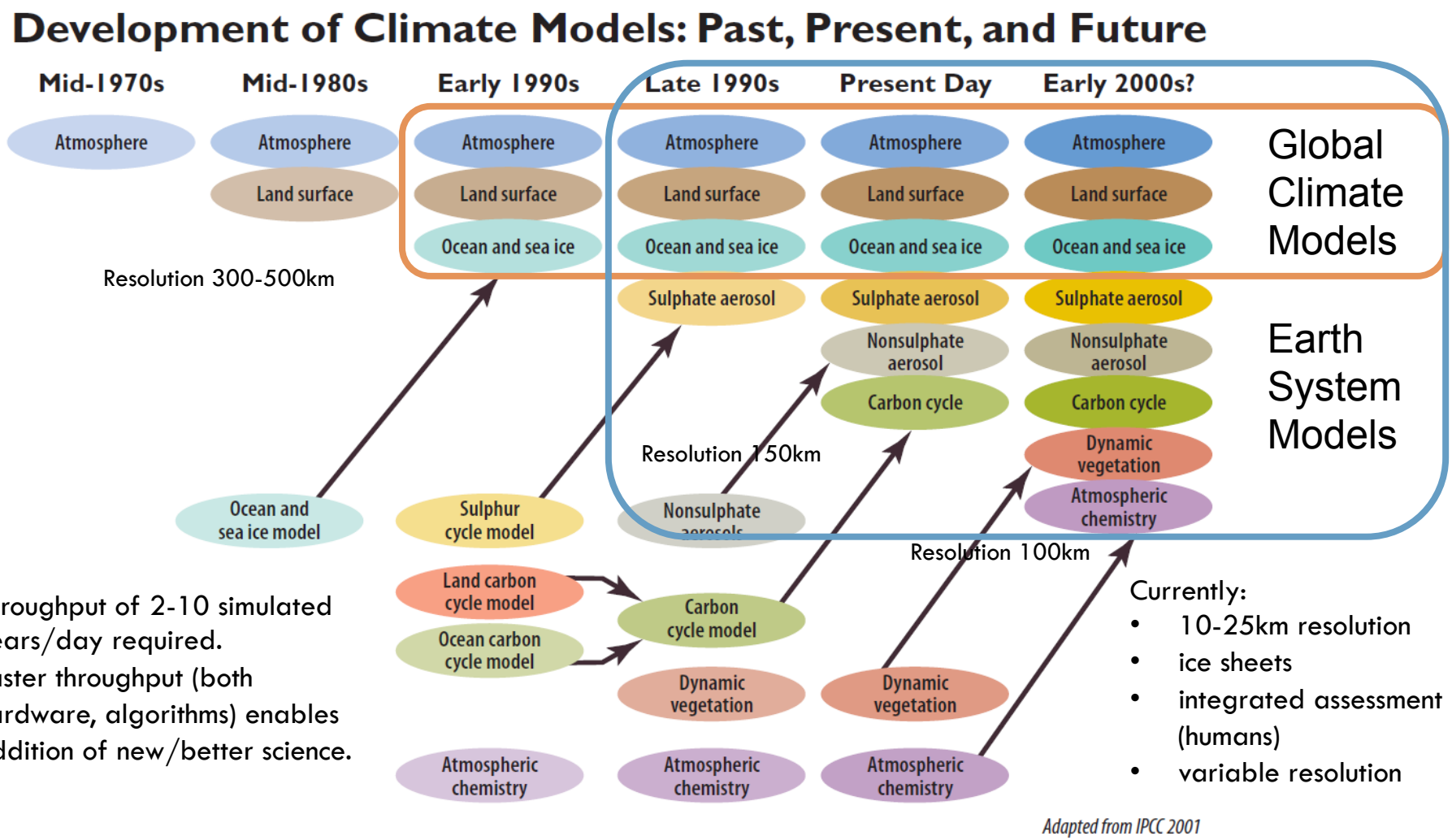
Helps it ↑

Hurts it ↓

Agenda

- Welcome and introduction
  - Background and goals for this SWP4XS workshop
- Scientific application software challenges at extreme scale
  - Extreme-scale architecture trends *(Hans Johansen)*
  - Towards software productivity for computational science *(Mike Heroux)*
- **Software productivity for extreme science applications**
  - **Example: DOE Climate community *(Phil Jones)***
  - Cross-cutting application issues *(Lois Curfman McInnes)*
- Workshop participants and agenda
  - Review of the agenda
  - Results of Pre-Workshop Survey *(Jeffrey Carver)*

New architectures provide unprecedented opportunities for new science



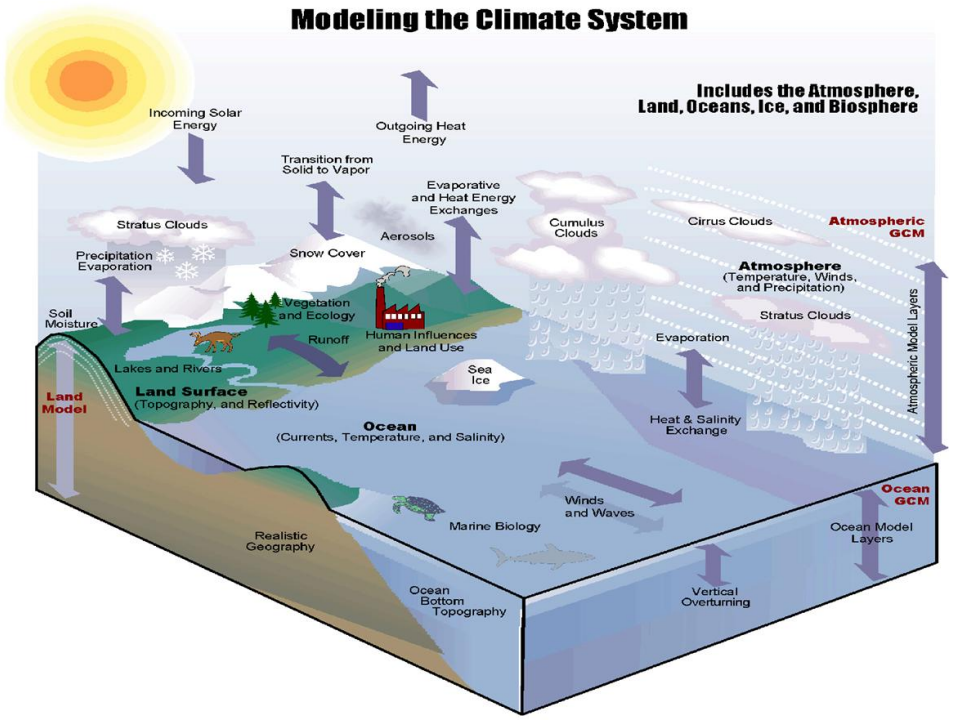
Throughput of 2-10 simulated years/day required.
Faster throughput (both hardware, algorithms) enables addition of new/better science.

Currently:

- 10-25km resolution
- ice sheets
- integrated assessment (humans)
- variable resolution

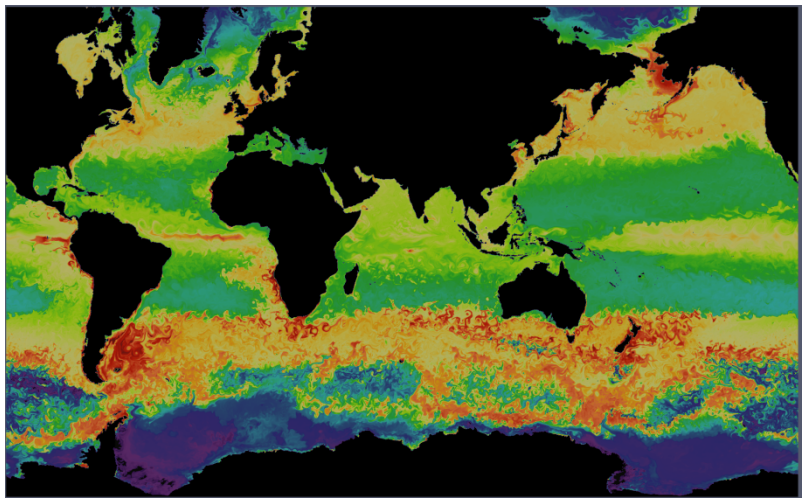
Climate / Earth System modeling

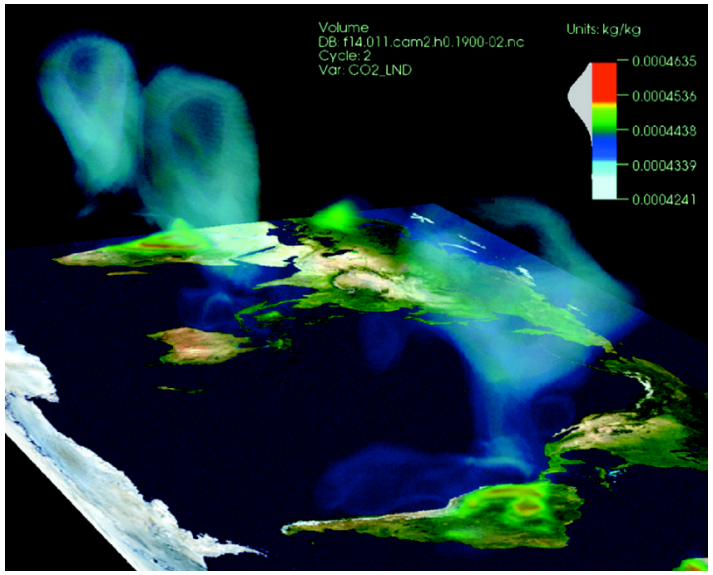
- **Multiphysics**
  - Atmosphere, ocean, land surface, sea-ice, ice sheets
  - Each component also complex multi-physics model
- **Multiscale**
  - Global to regional scales to inform impacts at decision-relevant scales
  - High resolution or variable resolution to resolve important processes (clouds, ocean eddies)
- **Community development**
  - Requires wide range of expertise
- **Informs decisions**
  - Requires a trusted, robust model
- **High-performance computing**



Software productivity challenges for climate

- **Refactoring for advanced architectures**
  - Rapid evolution and widespread changes
  - Robust software development and testing process to ensure software quality during transition
  - Software engineering techniques for performance portability
- **Community multiphysics development**
  - Improved software process for distributed development and software integration
  - Training all developers to adopt processes
  - Pre-commit testing to avoid disruptions
  - Community-wide HPC infrastructure
  - Improved coupling techniques as model complexity increases
- **Comprehensive testing and quality**
  - Full hierarchy of tests
  - Improved UQ and V&V techniques, esp for processes where known solutions do not exist





Climate extreme-science software opportunities

DOE and others already starting to address challenges:

- SciDAC Multi-Scale project
- Climate HPC community (like Sep 2013 multi-core workshop)
- Opportunities for Co-Design (like ExaCT for combustion)

Targeting Titan, Mira, Edison, with initial MIC prototypes, NERSC-8 (late 2015), LCF upgrades (2017)

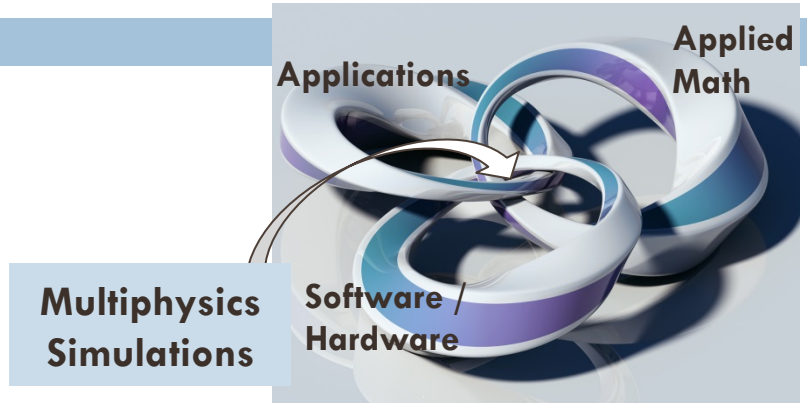
Common software productivity cultural issues include:

- Testing, quality assurance, verification and validation
  - → Accelerates performance porting, builds confidence in science results
- Recruiting and training
  - → Working with legacy codes vs. newest architectures
  - → Trial-and error refactoring, performance improvements

Agenda

- Welcome and introduction
  - Background and goals for this SWP4XS workshop
- Scientific application software challenges at extreme scale
  - Extreme-scale architecture trends *(Hans Johansen)*
  - Towards software productivity for computational science *(Mike Heroux)*
- **Software productivity for extreme science applications**
  - Example: DOE Climate community *(Phil Jones)*
  - **Cross-cutting application issues *(Lois Curfman McInnes)***
- Workshop participants and agenda
  - Review of the agenda
  - Results of Pre-Workshop Survey *(Jeffrey Carver)*

These software productivity challenges echo throughout many HPC multiphysics applications



ICiS multiphysics workshop

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

ICiS Institute for Computing in Science

THE INTERNATIONAL JOURNAL of HIGH PERFORMANCE COMPUTING APPLICATIONS

Multiphysics simulations: Challenges and opportunities

The International Journal of High Performance Computing Applications
27(1) 4–83
© The Author(s) 2012
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1094342012468181
hpc.sagepub.com
SAGE

David E Keyes[1,2], Lois C McInnes[3], Carol Woodward[4], William Gropp[5], Eric Myra[6], Michael Pernice[7], John Bell[8], Jed Brown[3], Alain Clo[1], Jeffrey Connors[4], Emil Constantinescu[3], Don Estep[9], Kate Evans[10], Charbel Farhat[11], Ammar Hakim[12], Glenn Hammond[13], Glen Hansen[14], Judith Hill[10], Tobin Isaac[15], Xiangmin Jiao[16], Kirk Jordan[17], Dinesh Kaushik[3], Efthimios Kaxiras[18], Alice Koniges[8], Kihwan Lee[19], Aaron Lott[4], Qiming Lu[20], John Magerlein[17], Reed Maxwell[21], Michael McCourt[22], Miriam Mehl[23], Roger Pawlowski[14], Amanda P Randles[18], Daniel Reynolds[24], Beatrice Rivière[25], Ulrich Rüde[26], Tim Scheibe[13], John Shadid[14], Brendan Sheehan[9], Mark Shephard[27], Andrew Siegel[3], Barry Smith[3], Xianzhu Tang[28], Cian Wilson[2] and Barbara Wohlmuth[23]

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

  What works well for each component (separately) may not be optimal for coupled codes!
- 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."

– Steve Jobs, Apple World Wide Developers Conference, Closing Keynote Q&A, 1997

Critical science applications at scale

What do these critical science applications have in common?

- **Complexity**
  - Coupled multiscale, multiphysics, multiuse, millions of lines of code
- **Mission-focused**
  - Confidence in scientific results, V&V + UQ vs theory and experiment
- **Longevity**
  - Long-lived applications and supporting infrastructure (decades)
- **Extreme-scale**
  - Must run on DOE's latest, fastest machines to achieve science goals

→ Unique requirements require careful planning and investment

Addressing scientific software productivity NOW is essential to DOE's scientific productivity

Our best way to keep development costs in line during the new era of disruptive architectural changes and unprecedented computing power

- **Required changes will affect every key data structure** and function in existing applications
  - Approaches underway in libraries: Separate control logic of algorithms from computational kernels
- **New scalable algorithms will be essential** for key computations with global data dependence to overcome difficulties with
  - Recursions, collective operations, data-driven parallelism, etc.

Opportunity in DOE software productivity to exploit unique features of DOE mathematical software and codes simulating physics systems

SWP4XS Workshop Organizers

- **Organizing Committee**
  - Hans Johansen (LBNL), Co-Chair
    Lois Curfman McInnes (ANL), Co-Chair
    David Bernholdt (ORNL)
    Jeffrey Carver (University of Alabama)
    Mike Heroux (SNL)
    Rich Hornung (LLNL)
    Phil Jones (LANL)
    Bob Lucas (University of Southern California)
    Andrew Siegel (ANL)
- **ORISE**
  - Keri Cagle
  - Deneise Terry

FAQ on SWP4XS workshop scope

Develop a strategic vision for DOE computational science software productivity that encompasses the unique aspects of extreme-scale computing

SWP4XS is …

- Maturing computational science
- Long-term investment in research on computational science infrastructure
- Coordinated with LCF roadmaps (petscale to exascale)
- Transfer of HPC research to science applications
- People and methodology-focused
- All for extreme-scale application software productivity that enables scientific discovery on emerging architectures

… but is not …

- Creating specific algorithms, applications
- How to train grad students on tools
- Short-term development/consulting
- Low-end HPC-focused
- New exascale tools, algorithms (e.g., X-STACK)
- "Non-professional" developer IDE

Goals for this workshop

Create a SWP4XS research agenda

- Vision for the ***workshop report***
  - Unique research challenges
  - Relevance to extreme science apps
  - External factors/trends to monitor
- What would a ***coherent research program*** contain?
  - What short-term and long-term objectives?
  - What are the biggest priorities?
  - What risks should be addressed?
  - How does this complement other programs, scientific or exascale?

Establish a SWP4XS community

- What efforts be coordinated across our community?
  - SW req's prioritized for extreme scale (not exclusive to it if applicable)
  - Vs. HW productivity
  - External factors/trends to track?

DOE needs an extreme-scale SWP focus!

- Manycore migration and emerging extreme-scale architectures represent true crisis/opportunity
- Software engineering community have the experience to help with this transformation
- Better SWP can give us better, faster *and* cheaper
  - **Better:** Science, portability, robustness, composability
  - **Faster:** Execution, development, dissemination
  - **Cheaper:** Fewer staff hours, lines of code
- Modest SWP investment is best leveraged investment
  - Sustainable and symbiotic relationship between ASCR & Science offices, as well as NNSA

Agenda

- **Welcome and introduction**
  - Background and goals for this SWP4XS workshop
- **Scientific application software challenges at extreme scale**
  - Extreme-scale architecture trends *(Hans Johansen)*
  - Towards software productivity for computational science *(Mike Heroux)*
- **Software productivity for extreme science applications**
  - Example: DOE Climate community *(Phil Jones)*
  - Cross-cutting application issues *(Lois Curfman McInnes)*
- **Workshop participants and agenda**
  - **Review of the agenda**
  - Results of Pre-Workshop Survey *(Jeffrey Carver)*

Today, Day 1 Highlights (8:30 – 5:15)

Monday Jan 13th Agenda	**Breakout Topic 1:** ***Transforming computational science software research for extreme-scale computing: Patterns and best practices***	**Breakout Topic 2:** ***Bridging the gap between domain science applications and computational science software: Research and development needs***
Lightning presentations	10:30 – 11:00	2:00 – 2:30
Breakouts	11:00 – 12:15	2:30 – 3:45
Report outs	1:30 – 2:00	4:15 – 4:45

- Working Lunch 12:15-1:30
  - Doug Post (DOD)
    "Addressing Application Software Productivity Challenges for Extreme-scale Computing"
- 4:45 – 5:15 Lightning presentations on "Software engineering and community issues"

Tue 1/14, Day 2 Highlights (8:30 – 1:00)

- Panel discussion 8:45 – 9:45
  - *Have your science teams seen productivity impacts from ever-changing, diverse supercomputing architectures?*
  - *What symptoms can you see in your scientific applications and team productivity?*
  - *How have you changed your team composition and collaborations to address these changes?*
  - *What skills or resources do you wish your teams had to be more productive?*
- Concurrent breakout session (10:00 – 11:30), Report out (12:00 – 12:45)
  - ***Computational science software productivity at extreme scale: Short-term/long-term priorities***
  - ***Discussion will include priorities for the workshop report***
- Box lunch to go 1:00