Software and High Performance Computing: Challenges for Research

The Implications of PITAC for High-End Computing

Ken Kennedy
Center for High Performance Software
Rice University

http://www.cs.rice.edu/~ken/Presentations/HPCSoftwareChallenges.pdf
PITAC Charter

- The Committee shall provide an independent assessment of:
  - Progress made in implementing the High-Performance Computing and Communications (HPCC) Program;
  - Progress in designing and implementing the Next Generation Internet initiative;
  - The need to revise the HPCC Program;
  - Balance among components of the HPCC Program;
  - Whether the research and development undertaken pursuant to the HPCC Program is helping to maintain United States leadership in advanced computing and communications technologies and their applications;
  - Other issues as specified by the Director of the Office of Science and Technology.
    - Review of the entire IT investment strategy — is it meeting the nation’s needs
PITAC Membership 97–99

• Co-Chairs:
  — Bill Joy, Sun Microsystems  — Ken Kennedy, Rice

• Members:
  — Eric Benhamou, 3Com  — Vinton Cerf, MCI
  — Ching-chih Chen, Simmons  — David Cooper, LLNL
  — Steve Dorfman, Hughes  — David Dorman, PointCast
  — Bob Ewald, SGI  — David Farber, Penn
  — Sherri Fuller, U of Washington  — Hector Garcia-Molina, Stanford
  — Susan Graham, UC Berkeley  — Jim Gray, Microsoft
  — Danny Hillis, Disney, Inc  — Robert Kahn, CNRI
  — John Miller, Montana State  — David Nagel, AT&T
  — Larry Smarr, UIUC  — Joe Thompson, Miss. State
  — Les Vadasz, Intel  — Andy Viterbi, Qualcomm
  — Steve Wallach, Centerpoint  — Irving Wladawsky-Berger, IBM
Methodology

• Evaluation of Federal Research Investment Portfolio
  — Plans reviewed for each of the major areas:
    - High End Computing and Computation
    - Large Scale Networking
    - Human Centered Computer Systems
    - High Confidence Systems
    - Education, Training, and Human Resources

• Review of Balance in Federal Research Portfolio
  — Fundamental versus Applied
    - Based on our own definition of these terms
  — High-Risk versus Low-Risk
  — Long-Term versus Short-Term
Principal Finding

- Drift Away from Long-Term Fundamental Research
Principal Finding

• Drift Away from Long-Term Fundamental Research
  — Agencies pressed by the growth of IT needs
    - IT R&D budgets have grown steadily but not dramatically
    - IT industry has accounted for over 30 percent of the real GDP growth over the past five years, but gets only 1 out of 75 Federal R&D dollars
    - Problems solved by IT are critical to the nation—engineering design, health and medicine, defense
Principal Finding

- Drift Away from Long-Term Fundamental Research
  - Agencies pressed by the growth of IT needs
    - IT R&D budgets have grown steadily but not dramatically
    - IT industry has accounted for over 30 percent of the real GDP growth over the past five years, but gets only 1 out of 75 Federal R&D dollars
    - Problems solved by IT are critical to the nation—engineering design, health and medicine, defense
  - Most IT R&D agencies are mission-oriented
    - Natural and correct to favor the short-term needs of the mission
Principal Finding

- Drift Away from Long-Term Fundamental Research
  - Agencies pressed by the growth of IT needs
    - IT R&D budgets have grown steadily but not dramatically
    - IT industry has accounted for over 30 percent of the real GDP growth over the past five years, but gets only 1 out of 75 Federal R&D dollars
    - Problems solved by IT are critical to the nation—engineering design, health and medicine, defense
  - Most IT R&D agencies are mission-oriented
    - Natural and correct to favor the short-term needs of the mission

- This Trend Must Be Reversed
  - Continue the flow of ideas to fuel the information economy and society
Remedy

• Increase the Federal IT R&D Investment by 1.4 billion dollars per year
  — Ramp up over five years
  — Focus on increasing fundamental research

• Invest in Key Areas Needing Attention
  — Software
  — Scalable Information Infrastructure
  — High-End Computing
  — Social, Economic, and Workforce Issues

• Develop a Coherent Management Strategy
  — Establish clear organizational responsibilities
  — Diversify modes of support
Software

- **Recommendations**
  - Make fundamental software research an absolute priority
  - Invest in key areas needing attention:
    - Improving programmer productivity
      Ameliorate the shortage of IT professionals
    - Improving reliability and robustness of software
      Critical infrastructure
    - Improving usability through human interface innovations
    - Improving capabilities for information management
  - Make software research a substantive component of every major information technology research initiative.
High-End Computing

- Findings:
  - High-end computing is essential for science and engineering research
  - High-end computing is an enabling element of the United States national security program
  - New applications of high-end computing are ripe for exploration
  - Suppliers of high-end systems suffer from difficult market pressures
    - High-end market not large
  - Innovations are required in high-end systems and application-development software, algorithms, programming methods, component technologies, and computer architecture
    - Scalable parallel architectures not ideal for every application
  - High-end computing capability for the civilian science and engineering community is falling dangerously behind the state of the art
High-End Recommendations

• Research:
  — Fund research into innovative computing technologies and architectures
  — Fund R&D on software for improving the performance of high-end computing
  — Drive high-end computing research by trying to attain a sustained petaops/petaflops on real applications by 2010 through a balance of hardware and software strategies

• Facilities
  — Fund the acquisition of the most powerful high-end computing systems to support science and engineering research

• Management
  — Expand the NSTC CIC High End Computing and Computation (HECC) Working Group’s coordination process to include all major elements of the government’s investment in high-end computing
Software Challenges for HPC
Software Challenges for HPC

• Overcome fundamental limitations of high performance computers
  — Scalability
  — Memory hierarchy performance
Software Challenges for HPC

• Overcome fundamental limitations of high performance computers
  — Scalability
  — Memory hierarchy performance

• Improve productivity of application developers
  — Achieve truly portable performance
  — Develop more powerful programming interfaces
  — Foster effective software component reuse
  — Provide support for real applications
    - Dynamic, adaptive
Software Challenges for HPC

• Overcome fundamental limitations of high performance computers
  – Scalability
  – Memory hierarchy performance

• Improve productivity of application developers
  – Achieve truly portable performance
  – Develop more powerful programming interfaces
  – Foster effective software component reuse
  – Provide support for real applications
    – Dynamic, adaptive

• Respond to the challenges of new HPC platforms
  – Ultrascale computing systems
  – Computational grids
Our Record To Date

• Many good ideas
  — Innovative research activity over the past decade

• Not many useful products
  — MPI, PVM, OpenMP are examples
  — Missing: higher-level languages, machine-independent debuggers
Our Record To Date

• Many good ideas
  — Innovative research activity over the past decade

• Not many useful products
  — MPI, PVM, OpenMP are examples
  — Missing: higher-level languages, machine-independent debuggers

• Why?
  — Pace of architectural change
    - Continuous concentration on the next architectural trick
  — Errors in HPCC investment strategy
    - Not enough software money, Grand Challenges unproductive as generators of software innovation
  — Tech transfer mechanisms flawed
    - High end business is small ➔ limited resources for software
What Should Be Done

• Focus the HPC software research community on long-term, high-risk approaches
  —HPC software grand challenges
What Should Be Done

• Focus the HPC software research community on long-term, high risk approaches
  — HPC software grand challenges

• Foster collaborations that produce software as well as science
  — It should be OK for a collaboration on applications to drive work on software technologies that might not directly affect the application
    - New language and compiler strategies that require years to develop
What Should Be Done

• Focus the HPC software research community on long-term, high risk approaches
  — HPC software grand challenges

• Foster collaborations that produce software as well as science
  — It should be OK for a collaboration on applications to drive work on software technologies that might not directly affect the application
  — New language and compiler strategies that require years to develop

• Invest in technology transfer
  — But do not make this an immediate short-term goal
What Should Be Done

- Focus the HPC software research community on long-term, high risk approaches
  - HPC software grand challenges
- Foster collaborations that produce software as well as science
  - It should be OK for a collaboration on applications to drive work on software technologies that might not directly affect the application
    - New language and compiler strategies that require years to develop
- Invest in technology transfer
  - But do not make this an immediate short-term goal
- Demand good software on procurements and be willing to pay for it!
HPC Software Grand Challenges I

- Automatic Application Tuning for Ultrascale Computation
  - Problem: Application development for:
    - Thousands of processors
    - Deep memory hierarchies (10 levels or more)

- Strategy
  - Language, compiler, and library support
  - Automatic tuning for new architectures
    - Example: Atlas
  - Run-time optimization

- Challenges
  - General automatic tuning strategies and algorithms
  - Producing a taxonomy of tuning parameters for all architectures
  - Minimizing support required from programmer/library developer
HPC Software Grand Challenges II

• Efficient Script-Based Scientific Programming
  – Problem:
    - Productivity of scientific application developers
    - Too much emphasis on low-level programming

• Strategy:
  – Problem-solving environments for HPC
  – Component libraries developed by professionals
  – Scripts used to integrate applications

• Challenges:
  – High performance on a variety of target architectures
    - Global compilation strategies that do not overtax script compilation time
Telescoping Languages

Domain Library
Telescoping Languages

Domain Library

Exhaustive Global Pre-Compiler

Fast Compiler
Telescoping Languages

- Domain Library
- Exhaustive Global Pre-Compiler
- Script
- Script Translator
- Fast Compiler
- Optimized Application
HPC Software Grand Challenges III

• Application Development and Performance Management for Grids
  — Problem:
    - Reliable performance on heterogeneous platforms
    - Varying load
    On computation nodes and on communications links

• Strategy
  — Programs prepared for adaptability
  — Continuous monitoring and reconfiguration during execution

• Challenge:
  — Mapping applications to dynamically changing architectures
  — Determining when to interrupt execution and remap
    - Application monitors
    - Performance estimators
Grid Compilation Architecture

- **Goal:** reliable performance under varying load

GrADS Project (NSF NGS): Berman, Chien, Cooper, Dongarra, Foster, Gannon, Johnsson, Kennedy, Kesselman, Reed, Torczon, Wolski
Summary

• HPC Will Continue to Provide Enormous Challenges for Software
  — Scalability, memory hierarchy, adaptability, portability

• Capitalization of New Ideas Will Take Time
  — Not enough resources for high-end computing software

• Focus Must Remain on the Long Term
  — But intermediate byproducts are useful

• Grand Challenges
  — Automatic performance tuning of applications for HPC platforms
  — Efficient, script-based problem-solving environments
  — Application development for computational grids