The Forthcoming Petascale Systems Era

“GOT TOOLS?”

Tony Drummond
Computational Research Division
Lawrence Berkeley National Laboratory
Salishan April 21, 2005
The Forthcoming Petascale Systems Era “GOT TOOLS?”

- Accelerator Science
- Astrophysics
- Biology
- Chemistry
- Earth Sciences
- Materials Science
- Nanoscience
- Plasma Science

Commonalities:
- Major advancements in Science
- Increasing demands for computational power
- Rely on available computational systems, languages, and software tools

Where are the applications?
Increasing Computational Demand
(Area: Atmospheric Research)

SPECTRUM OF ATMOSPHERIC PHENOMENA

Physics limits the model’s resolution, higher resolution will require computer systems of higher capacities.

- PLANETARY SCALE
- SYNOPTIC SCALE
- MESO-SCALE
- DEEP CONVECTION
- SHALLOW CONVECTION
- TURBULANCE
- VISCOUS
- LARGE INERTIAL EDDIES
- SUBRANGE

10^4 km 10^3 km 10^2 km 10 km 1 km 10^2 m 10 m 1 m 1 dm 1 cm 1 mm

Office of Science
U.S. Department of Energy

The Forthcoming Petascale Systems Era “GOT TOOLS?”
The Forthcoming Petascale Systems Era “GOT TOOLS?”

Multidisciplinary Research (Area: Climate Research)

Atmospheric general circulation model
- Dynamics
- Sub-grid scale parameterized physics processes
  - Turbulence, solar/infrared radiation transport, clouds.

Oceanic general circulation model
- Dynamics (mostly)

Sea ice model
- Viscous elastic plastic dynamics
- Thermodynamics

Land Model
- Energy and moisture budgets
- Biology

Chemistry
- Tracer advection, possibly stiff rate equations.

Ocean Biology

Duffy et. al., Lawrence Livermore National Laboratory

Mathew E. Maltruda and Julie L. McClean

U.S. DEPARTMENT OF ENERGY

COLLECTION
Climate Models:

- Higher resolutions are computational demanding
- No-trivial load-balancing
- Coupling different physics, times and spatial domains needs very flexible, high performance and robust tools.

**AGCM/ACM**
- 2.5 long x 2 lat, 30 layers
- 25-chemical species
- 2.5 long x 2 lat, 30 layers
- 2-chemical species
"We need to move away from a coding style suited for serial machines, where every macrostep of an algorithm needs to be thought about and explicitly coded, to a higher-level style, where the compiler and library tools take care of the details. And the remarkable thing is, if we adopt this higher-level approach right now, even on today's machines, we will see immediate benefits in our productivity."


*Numerical Recipes: Does This Paradigm Have a future?*
Changes in algorithms sometimes lead to several years advancement in computations. Needs Flexibility!

Its performance is influenced by system parameters and in steps in the algorithm. Critical points: portability and scalability.

New Architecture requires extensive tuning, may even require new programming paradigms. This is Difficult to maintain and not “very” portable.
The Forthcoming Petascale Systems Era “GOT TOOLS?”

Key Lesson Learned on the Road to _PetaFlop Computing_ . . .
(Software Development)

**USER's APPLICATION CODE**
(Main Control)

- Compilers +
- Expert Drivers +
- Support

**AVAILABLE LIBRARIES & PACKAGES**

- Application Data Layout
- Algorithmic Implementations
- I/O

Tuned and machine Dependent modules
The Forthcoming Petascale Systems Era “GOT TOOLS?”

Lesson = High Quality Software Reusability

- Scientific or engineering context
- Domain expertise

- Simulation codes
- Data Analysis codes

General Purpose Libraries

- Data Structures
- Algorithms
- Programming Languages
- Code Optimization
- O/S - Compilers
- O/S - Compilers

Hardware - Middleware - Firmware
Lesson = High Quality Software Reusability

Library Development

Numerical Tools

Code Development

General Purpose Libraries

• Data Structures
• Algorithms
• Code Optimization
• Programming Languages
• O/S - Compilers

Hardware - Middleware - Firmware

The Forthcoming Petascale Systems Era “GOT TOOLS?”

Funded by DOE/ASCR

http://acts.nersc.gov
<table>
<thead>
<tr>
<th>Category</th>
<th>Tool</th>
<th>Functionalities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Numerical</strong></td>
<td>AztecOO</td>
<td>Algorithms for the iterative solution of large sparse linear systems.</td>
</tr>
<tr>
<td></td>
<td>Hypre</td>
<td>Algorithms for the iterative solution of large sparse linear systems, intuitive grid-centric interfaces, and dynamic configuration of parameters.</td>
</tr>
<tr>
<td></td>
<td>PETSc</td>
<td>Tools for the solution of PDEs that require solving large-scale, sparse linear and nonlinear systems of equations.</td>
</tr>
<tr>
<td></td>
<td>OPT++</td>
<td>Object-oriented nonlinear optimization package.</td>
</tr>
<tr>
<td></td>
<td>SUNDIALS</td>
<td>Solvers for the solution of systems of ordinary differential equations, nonlinear algebraic equations, and differential-algebraic equations.</td>
</tr>
<tr>
<td></td>
<td>ScaLAPACK</td>
<td>Library of high performance dense linear algebra routines for distributed-memory message-passing.</td>
</tr>
<tr>
<td></td>
<td>SuperLU</td>
<td>General-purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations.</td>
</tr>
<tr>
<td></td>
<td>TAO</td>
<td>Large-scale optimization software, including nonlinear least squares, unconstrained minimization, bound constrained optimization, and general nonlinear optimization.</td>
</tr>
<tr>
<td><strong>Code</strong></td>
<td>Global Arrays</td>
<td>Library for writing parallel programs that use large arrays distributed across processing nodes and that offers a shared-memory view of distributed arrays.</td>
</tr>
<tr>
<td><strong>Development</strong></td>
<td>Overture</td>
<td>Object-Oriented tools for solving computational fluid dynamics and combustion problems in complex geometries.</td>
</tr>
<tr>
<td><strong>Code</strong></td>
<td>CUMULVS</td>
<td>Framework that enables programmers to incorporate fault-tolerance, interactive visualization and computational steering into existing parallel programs</td>
</tr>
<tr>
<td><strong>Execution</strong></td>
<td>Globus</td>
<td>Services for the creation of computational Grids and tools with which applications can be developed to access the Grid.</td>
</tr>
<tr>
<td></td>
<td>TAU</td>
<td>Set of tools for analyzing the performance of C, C++, Fortran and Java programs.</td>
</tr>
<tr>
<td><strong>Library</strong></td>
<td>ATLAS</td>
<td>Tools for the automatic generation of optimized numerical software for modern computer architectures and compilers.</td>
</tr>
<tr>
<td><strong>Development</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Forthcoming Petascale Systems Era “GOT TOOLS?”

Software Reusability
What have we gained? What are the goals?

\[
\begin{align*}
\min [\text{time}_\text{to}_\text{first}_\text{solution}] & \quad \text{min (prototype)} \\
& \\
\rightarrow \min [\text{time}_\text{to}_\text{solution}] & \quad \min \text{ (production)} \\
& \\
& \\
\min [\text{software}_\text{development}_\text{cost}] & \\
\max [\text{software}_\text{life}] \quad \text{and} \quad \max [\text{resource}_\text{utilization}] & \\
& \\
\end{align*}
\]

- Outlive Complexity
  - Increasingly sophisticated models
  - Model coupling
  - Interdisciplinary

- Sustained Performance
  - Increasingly complex algorithms
  - Increasingly diverse architectures
  - Increasingly demanding applications

\{(Software Evolution)\}
\{(Long-term deliverables)\}
Minimum Requirements for Reusable High Quality Software Tools

- Robustness
  - Maintained across platforms
  - Compiler independent
  - Precision Independent
  - Error Handling
  - Check Pointing
Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable (across large Petascale systems)
Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible (New Algorithms, New Techniques)
Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable

- Frameworks/PSE
- Tool-to-Tool
- Component Technology
  - More Flexible
  - Retains better Robustness, Scalability, and Extensibility
  - Long term pay-offs

http://www.cca-forum.org
Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable
- User Friendly Interfaces
- Well documented
The Forthcoming Petascale Systems Era “GOT TOOLS?”

**User Interfaces**

**Library Calls**

- `-ksp_type [cg, gmres, bcgs, tfqmr, ...]
- `-pc_type [lu, ilu, jacobi, sor, asm, ...]

**More advanced:**

- `-ksp_max_it <max_iters>
- `-ksp_gmres_restart <restart>
- `-pc_asm_overlap <overlap>

**Command lines**

**Problem Domain**
The Forthcoming Petascale Systems Era “GOT TOOLS?”

PyACTS
matlab*P  Star-P
NetSolve

User Interfaces

Ax = b

View_field(T1)

Az = λz

A = UΣVT

High Level Interfaces

OPT++  PAWS  Globus  CUMULVS  TAU
AZTEC  Hypre  PETSc  Chombo  Global Arrays
ScaLAPACK  SuperLU  TAO  PVODE  Overture
Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable
- User Friendly Interfaces
- Well documented
- Periodic Tests and Evaluations

Versions (tools, systems, O/S, compilers)

- Sanity-check (robustness)
- Interoperability (maintained)
- Consistent Documentation
Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable
- User Friendly Interfaces
- Well documented
- Periodic Tests and Evaluations
- **Portability and Fast Adaptability (The Evolution)**

The Forthcoming Petascale Systems Era “GOT TOOLS?”

Office of Science
U.S. Department of Energy
The Forthcoming Petascale Systems Era “GOT TOOLS?”

Tool Evolution
Example: ScaLAPACK

- ScaLAPACK
- PBLAS
- BLACS
- MPI/PVM/...
- BLAS
- LAPACK
- Global
- Local

New Algorithmic Implementations

New Design Consideration

Platform specific

ScaLAPACK Users’ Guide

Office of Science
U.S. Department of Energy
Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable
- User Friendly Interfaces
- Well documented
- Periodic Tests and Evaluations
- Portability and Fast Adaptability
- Long-term support
- Training (hands-on code) and High level support

The Forthcoming Petascale Systems Era “GOT TOOLS?”
The Forthcoming Petaflop Systems Era “GOT TOOLS?”

Advanced CompuTational Software Collection (ACTS) Project

User Community

- Numerical Simulations
- Challenge Codes
- Computing Systems
- Collaboratories
- Biology
- Physics
- Computer Sciences
- Medicine
- Engineering
- Mathematics
- Chemistry
- Bioinformatics

Pool of Software Tools

Testing and Acceptance Phase

Interoperability

Scientific Computing Centers

Workshops and Training

Computer Vendors

Office of Science
U.S. DEPARTMENT OF ENERGY
Minimum Requirements for Reusable High Quality Software Tools

- Robust
- Scalable
- Extensible
- Interoperable
- User Friendly Interfaces
- Well documented
- Periodic Tests and Evaluations
- Portability and Fast Adaptability
- Long-term support
- Training (hands-on code) and High level support
- Community support (developers, users, computer vendors, mailing lists, commercial software development and user groups)
The Forthcoming Petascale Systems Era “GOT TOOLS?”

Open Challenges

**DISTRIBUTED INTEGRATION**

- Distributed Coupling

**multi-physics, multi-resolutions, multi-domains**

**MEMORY REQUIREMENT FOR CENTRALIZED COUPLING**

- North Atlantic: OGCM
  - POP code: 1/8 lat x 1/8 lon, 37 levels
  - 128 CRAY T3E-600 Nodes (16MIPS/node)

- Global: AGCM
  - UCLA AGCM code:
    - POP code: 2 lat x 2.5 lon, 29 levels
    - 128 CRAY T3E nodes (16MIPS/node)

**MEMORY REQUIREMENT FOR DISTRIBUTED COUPLING**

- North Atlantic: OGCM
  - POP code: 1/8 lat x 1/8 lon, 37 levels
  - 128 CRAY T3E-600 Nodes (16MIPS)

- Global: AGCM
  - UCLA AGCM code:
    - 2 lat x 2.5 lon, 29 levels
    - 128 CRAY T3E nodes (16MIPS)

**Centralized Data Brokerage**

- $T_{\text{central}} = T_{\text{ogcm}} + T_{\text{interp}} + T_{\text{agcm scale}}$

- $T_{\text{central}} = 7.2 \text{MIPS} + 2.5 \text{MIPS} + 3.1 \text{MIPS}$
- $= 12.8 \text{MIPS}$

**Distributed Data Brokerage**

- $T_{\text{distrib}} = T_{\text{ogcm}} + T_{\text{interp/128}}$

- $T_{\text{distrib}} = 7.2 \text{MIPS} + 2.5 \text{MIPS/128}$
- $= 7.3 \text{MIPS}$
The Forthcoming Petascale Systems Era “GOT TOOLS?”

**DISTRIBUTED INTEGRATION**
A single-controller free approach

- Avoid Intrusive lines of codes.
- Provide interfaces to formulate data (flux, variables, fields, etc.) translations between applications.
- Variable Synchronization.
- Scalability!
The Forthcoming Petascale Systems Era “GOT TOOLS?”

• Distributed Coupling

![Diagram showing normalized seconds per simulated day with problem size and number of nodes on axes.](image-url)
The Forthcoming Petascale Systems Era “GOT TOOLS?”

Open Challenges

- Distributed Coupling
- **Improve interactions between Tool-Compilers-Hardware**

- Automatic Tuning and Profiling (TAU, IPM, etc)
- Automatic Code Generators (ATLAS-like)
- Debugging tools
- Tools and Language Interoperability
Open Challenges

- Distributed Coupling
- Improve interactions between Tool-Compilers-Hardware
- **Software Availability**
  - Installation and Configuration
  - Adaptability
“Got Tools?”

• There are several software development efforts enabling scientific and engineering applications meet the computational challenges at the Petaflops and beyond levels. We need to ensure that they meet the aforementioned tool requirements.

• Good Trend: High-level tool interfaces that hide software complexity from end users but won’t compromise performance.

• SOFTWARE REUSE!
Some References

- ACTS Information Center: http://acts.nersc.gov
- Two Upcoming Journal Issues dedicated to ACTS
- Sixth ACTS Collection Workshop, August 23-26, 2005