The Forthcoming Petascale Systems Era
"Got Tools?"

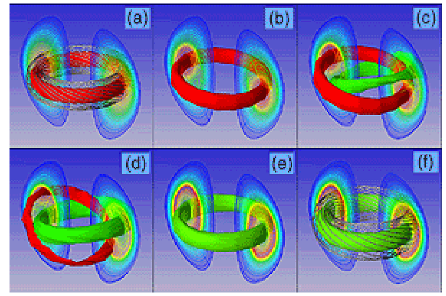
Tony Drummond
Computational Research Division
Lawrence Berkeley National Laboratory
Salishan April 21, 2005

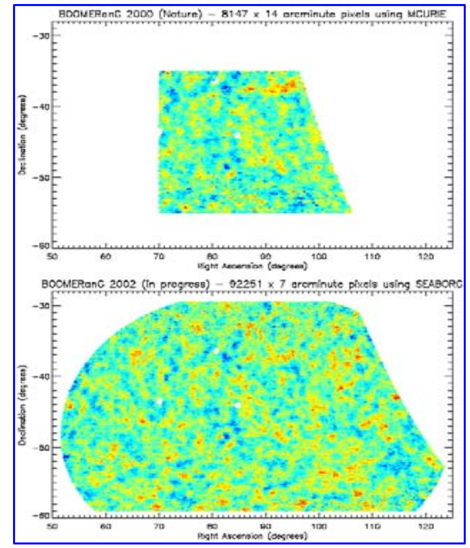
The Forthcoming Petascale Systems Era “GOT TOOLS?”

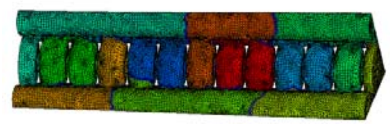
Where are the applications?

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

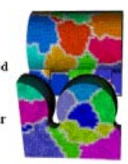
:

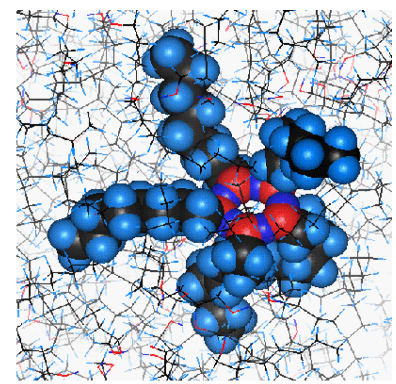






Omega3P is a parallel distributed-memory code intended for the modeling and analysis of accelerator cavities, which requires the solution of generalized eigenvalue problems. A parallel exact shift-invert eigensolver based on PARPACK and SuperLU has allowed for the solution of a problem of order 7.5 million with 304 million nonzeros.





Commonalities:

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

Office of Science U.S. DEPARTMENT OF ENERGY

ACTS COLLECTION

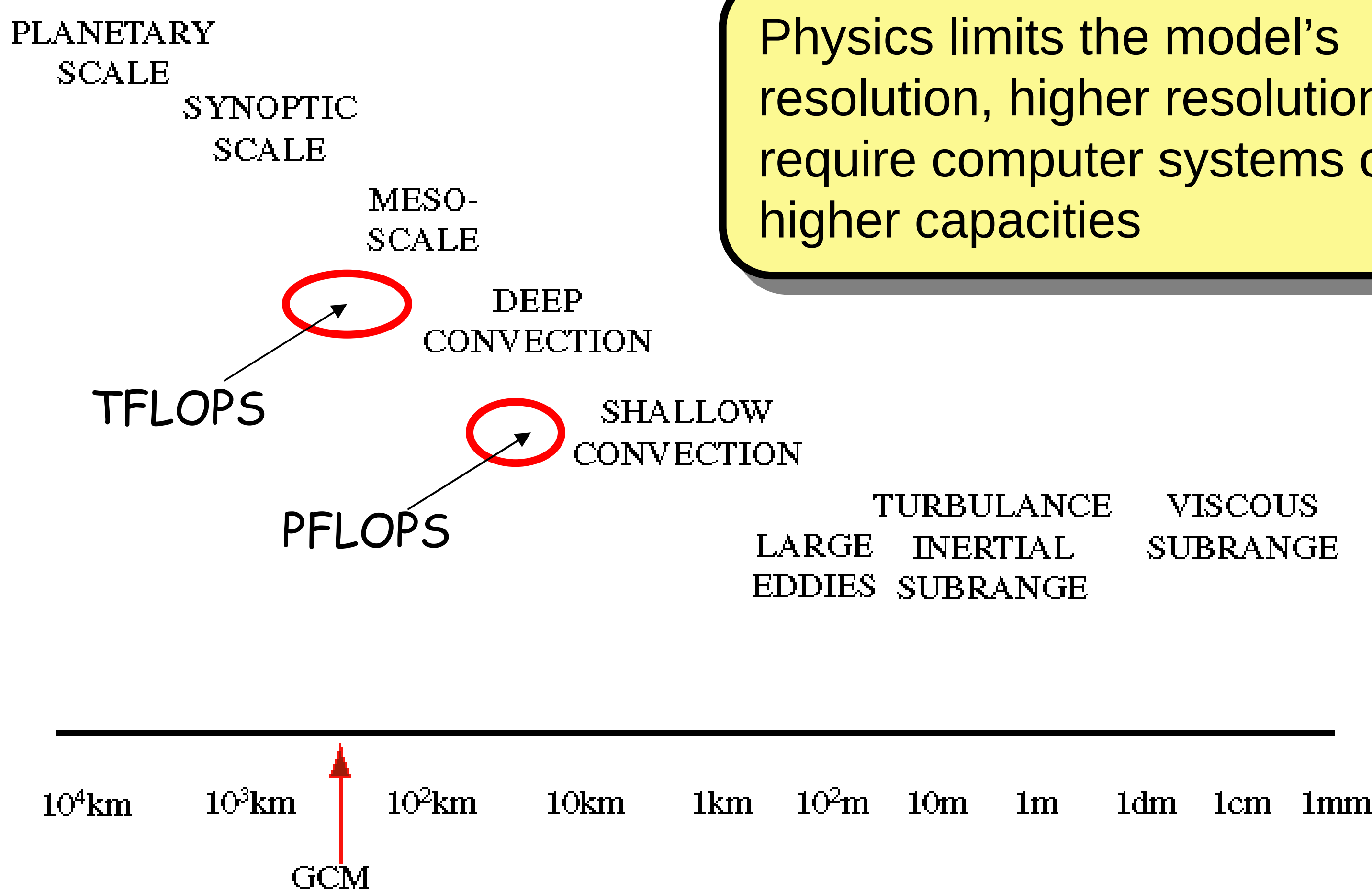
BERKELEY LAB

The Forthcoming Petascale Systems Era "GOT TOOLS?"

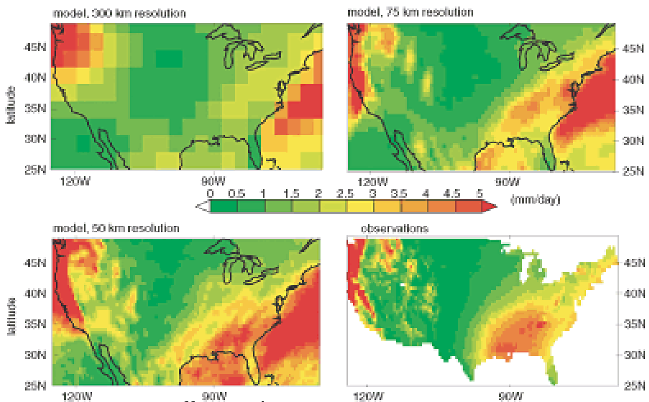
Increasing Computational Demand
(Area: Atmospheric Research)

SPECTRUM OF ATMOSPHERIC PHENOMENA

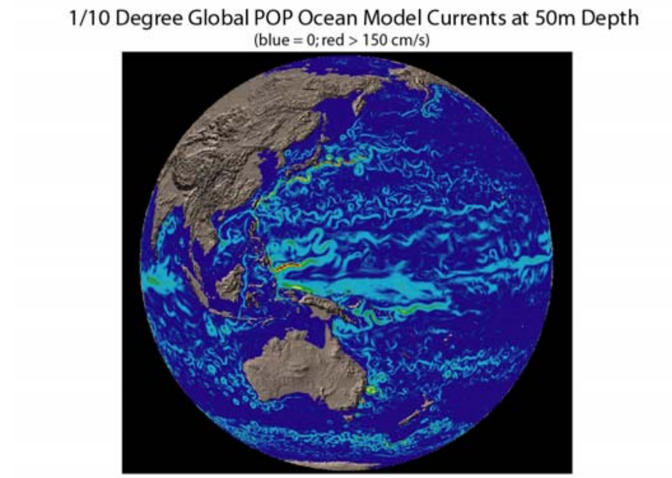
Physics limits the model's resolution, higher resolution will require computer systems of higher capacities



Multidisciplinary Research
(Area: Climate Research)



Duffy et. al.,
Lawrence Livermore National Laboratory



Mathew E. Maltruda and
Julie L. McClean

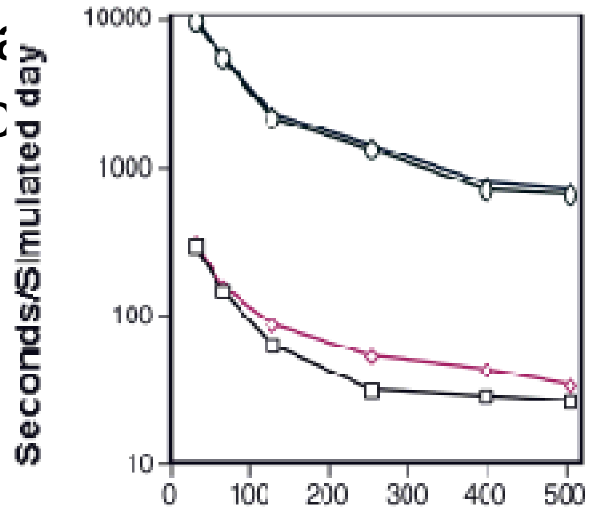
- Atmospheric general circulation model
  - Dynamics
  - Sub-grid scale parameterized physics processes
    - Turbulence, solar/infrared radiation transport, clouds.
- Oceanic general circulation model
  - Dynamics (mostly)
- Sea ice model
  - Viscous elastic plastic dynamics
  - Thermodynamics

- Land Model
  - Energy and moisture budgets
  - Biology
- Chemistry
  - Tracer advection, possibly stiff rate equations.
- Ocean Biology

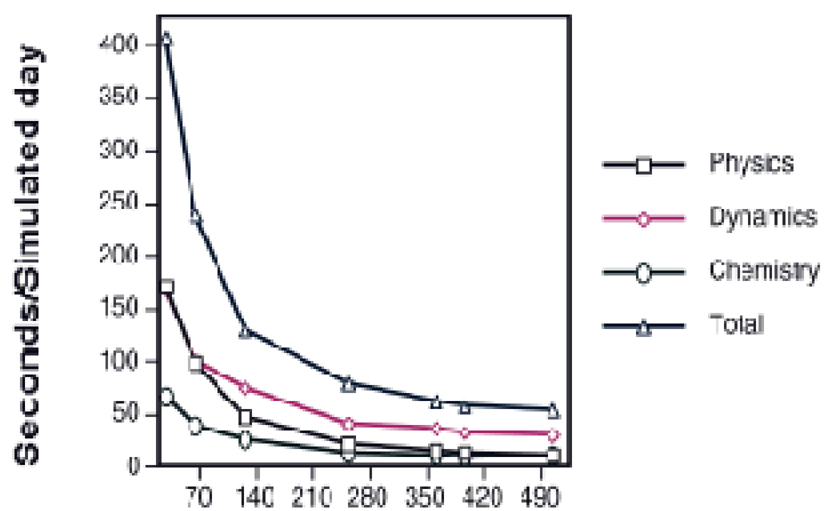
Some Computational Challenges *In Climate Research*

Climate Models:

- Higher resolutions are computational demanding
- No-trivial load-balancing
- Coupling different physics, times and spatial doma
perfc



AGCM/ACM
2.5 long x 2 lat, 30 layers
25-chemical species



AGCM/ACM
2.5 long x 2 lat, 30 layers
2-chemical species

The Forthcoming Petascale Systems Era "GOT TOOLS?"

Key Lesson Learned on the Road to ***PetaFlop Computing . .*** *(Software Development)*

"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."

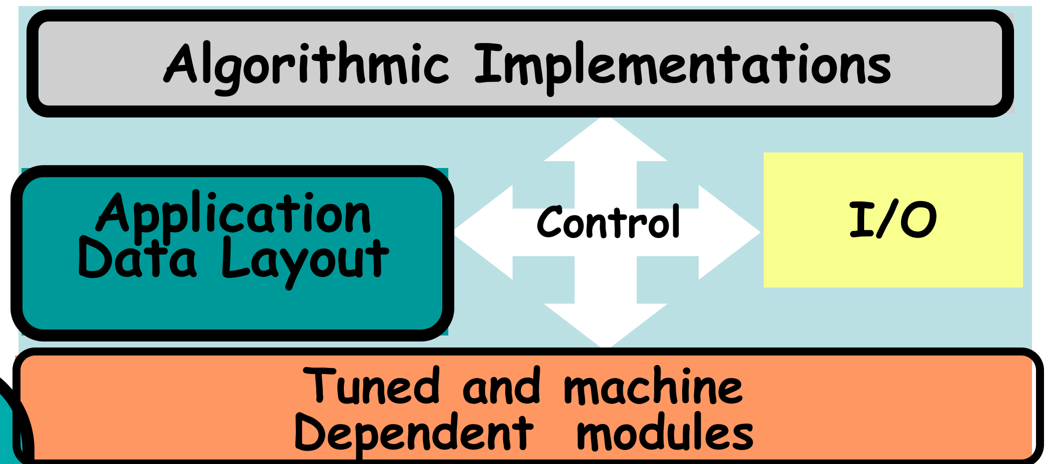
W. H. Press and S. A. Teukolsky, 1997

Numerical Recipes: Does This Paradigm Have a future?

The Forthcoming Petascale Systems Era "GOT TOOLS?"

Key Lesson Learned on the Road to ***PetaFlop Computing . .*** *(Software Development)*

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.**

Key Lesson Learned on the Road to ***PetaFlop Computing . .*** *(Software Development)*

USER's APPLICATION CODE
(Main Control)

Compilers + Expert Drivers + Support

AVAILABLE
Application Data Layout
LIBRARIES & PACKAGES

AVAILABLE
Algorithmic Implementations
LIBRARIES & PACKAGES

AVAILABLE
I / O
LIBRARIES

Tuned and machine Dependent modules

Lesson = High Quality Software Reusability

- Scientific or engineering context
- Domain expertise

- Simulation codes
- Data Analysis codes

General Purpose Libraries

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

Hardware - Middleware - Firmware

The Forthcoming Petascale Systems Era "GOT TOOLS?"

Lesson = High Quality Software Reusability

Funded by DOE/ASCR

LIBRARY DEVELOPMENT

NUMERICAL TOOLS

CODE DEVELOPMENT

RUN TIME SUPPORT

http://acts.nersc.gov

General Purpose Libraries

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

Hardware - Middleware - Firmware

Category	Tool	Functionalities
Numerical $Ax = b$ $Az = \lambda z$ $A = U\Sigma V^T$ PDEs ODEs	**AztecOO**	Algorithms for the iterative solution of large sparse linear systems.
	Hypre	Algorithms for the iterative solution of large sparse linear systems, intuitive grid-centric interfaces, and dynamic configuration of parameters.
	PETSc	Tools for the solution of PDEs that require solving large-scale, sparse linear and nonlinear systems of equations.
	OPT++	Object-oriented nonlinear optimization package.
	SUNDIALS	Solvers for the solution of systems of ordinary differential equations, nonlinear algebraic equations, and differential-algebraic equations.
	ScaLAPACK	Library of high performance dense linear algebra routines for distributed-memory message-passing.
	SuperLU	General-purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations.
	TAO	Large-scale optimization software, including nonlinear least squares, unconstrained minimization, bound constrained optimization, and general nonlinear optimization.
Code Development	**Global Arrays**	Library for writing parallel programs that use large arrays distributed across processing nodes and that offers a shared-memory view of distributed arrays.
	Overture	Object-Oriented tools for solving computational fluid dynamics and combustion problems in complex geometries.
Code Execution	**CUMULVS**	Framework that enables programmers to incorporate fault-tolerance, interactive visualization and computational steering into existing parallel programs
	Globus	Services for the creation of computational Grids and tools with which applications can be developed to access the Grid.
	TAU	Set of tools for analyzing the performance of C, C++, Fortran and Java programs.
Library Development	**ATLAS**	Tools for the automatic generation of optimized numerical software for modern computer architectures and compilers.

Software Reusability

What have we gained? What are the goals?

min[*time_to_first_solution*] (prototype)

min[*time_to_solution*] (production)

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

(Software Evolution)

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

(Long-term deliverables)

min[*software-development-cost*]

max[*software_life*] and **max**[*resource_utilization*]

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**

User Interfaces

```
CALL BLACS_GET( -1, 0, ICTXT )
 CALL BLACS_GRIDINIT( ICTXT, 'Row-major', NPROW, NPCOL )
:
 CALL BLACS_GRIDINFO(ICTXT, NPROW, NPCOL, MYROW, MYCOL)
:
:
 CALL PDGESV( N, NRHS, A, IA, JA, DESCA, IPIV, B, IB, JB, DESCB,
 $               INFO )
```

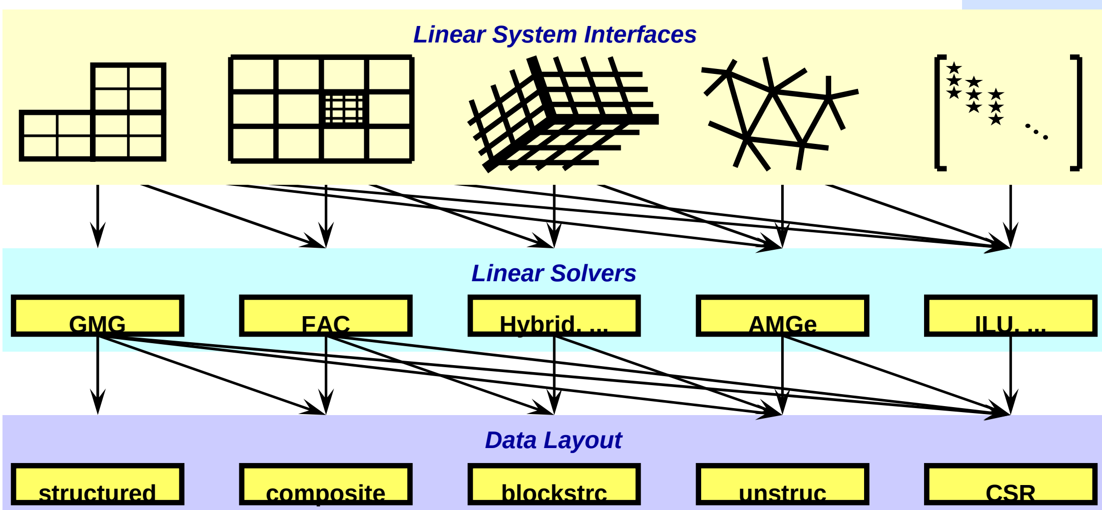
Library Calls

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

More advanced:

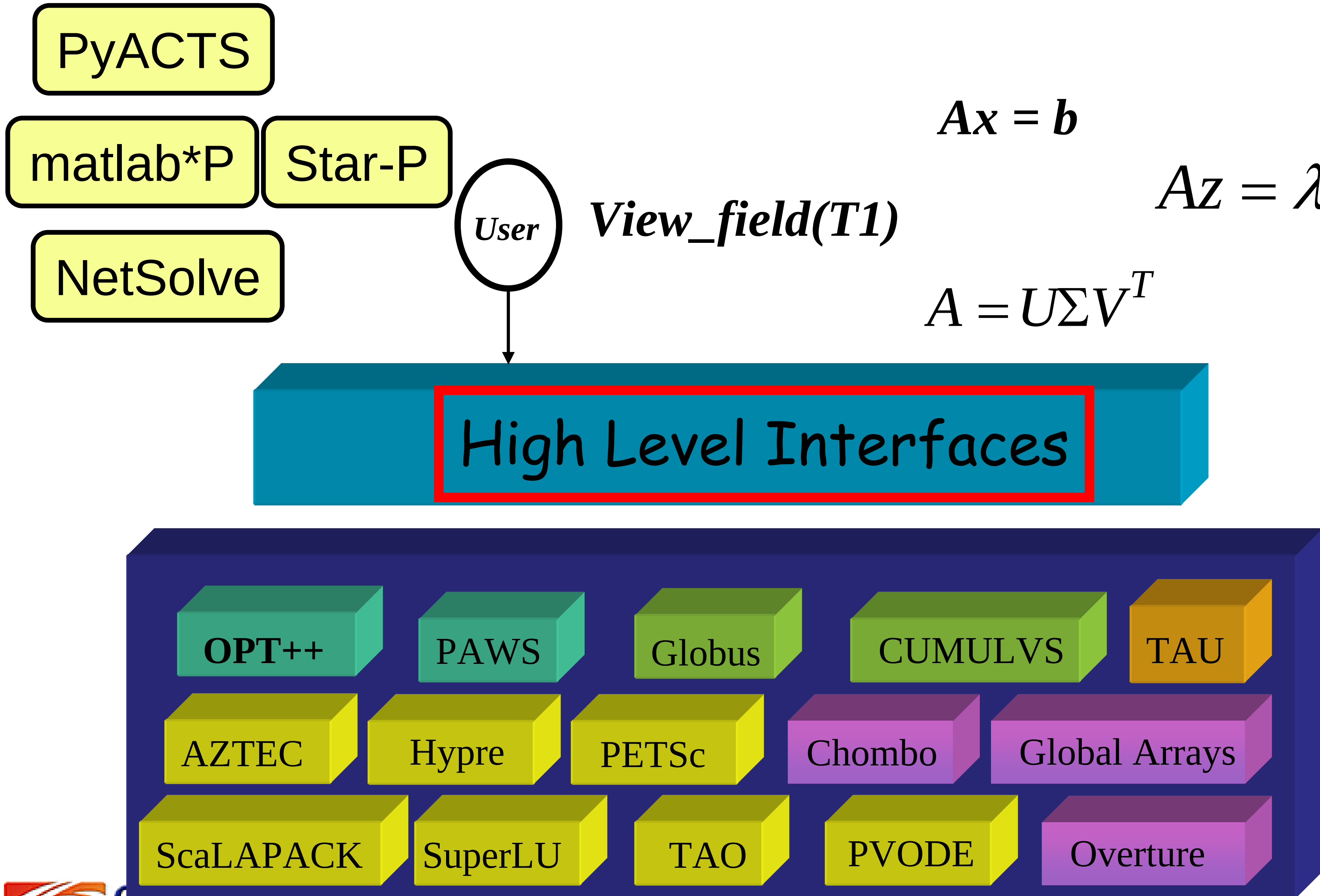
- -ksp_max_it <max_iters>
- -ksp_gmres_restart <restart>
- -pc_asm_overlap <overlap>

Command lines



Problem Domain

User Interfaces



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)

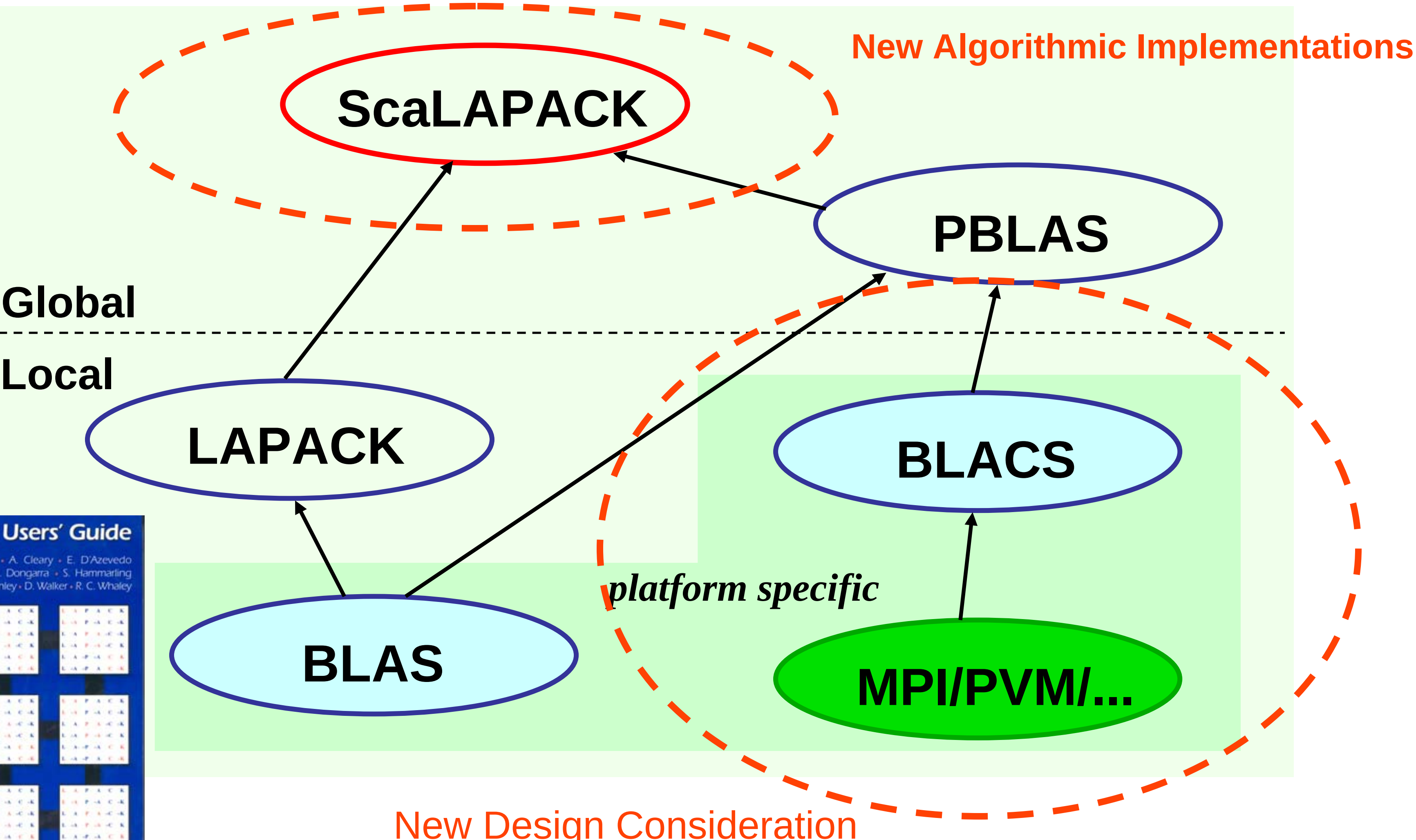
- Sanity-check (robustness)
- Interoperability (maintained)
- Consistent Documentation

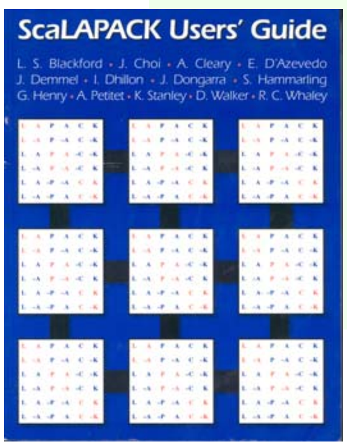
Minimum Requirements for Reusable High Quality Software Tools

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

Tool Evolution

Example: ScaLAPACK



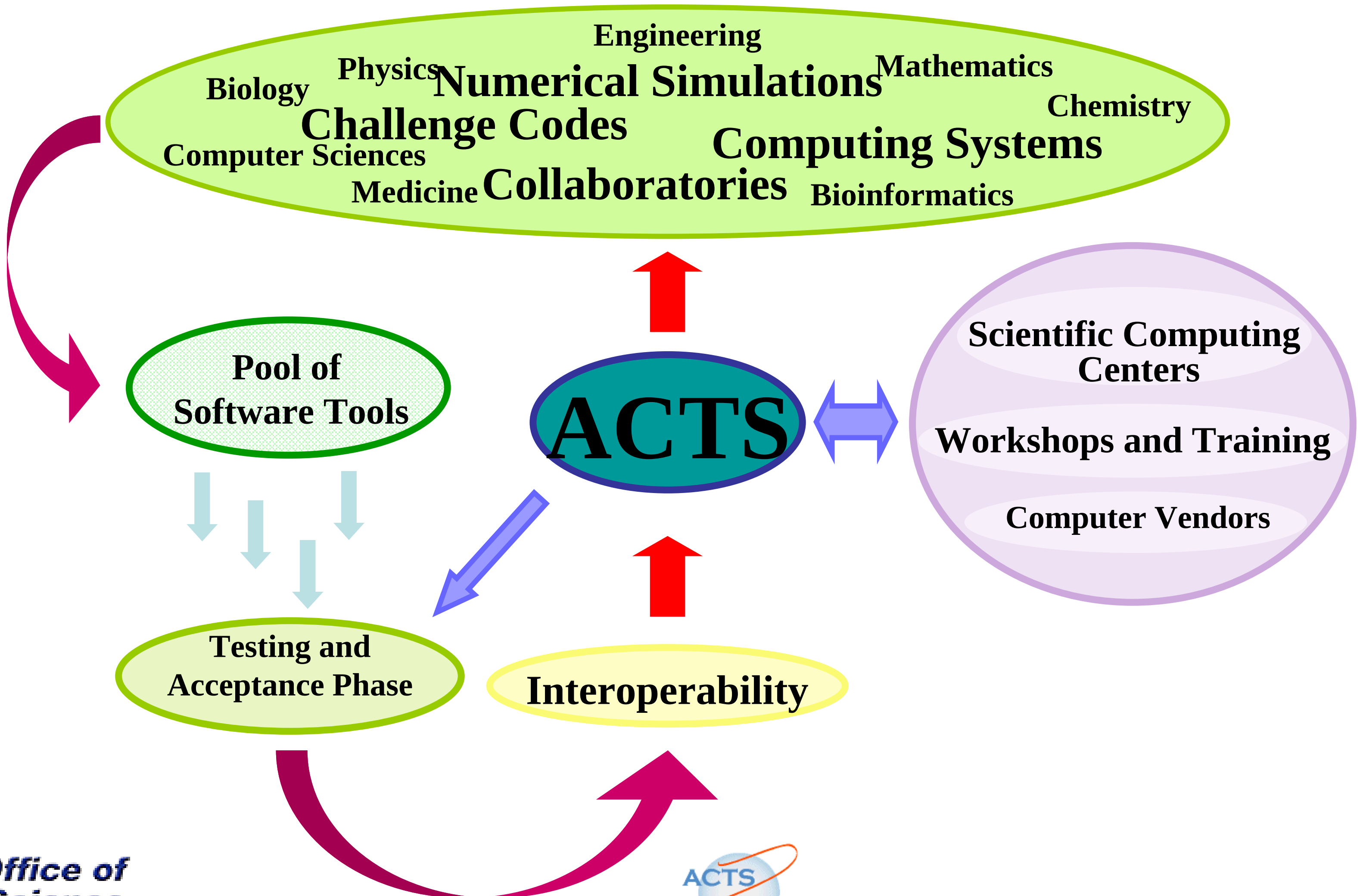


Minimum Requirements for Reusable High Quality Software Tools

- 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**

Advanced CompuTational Software Collection (ACTS) Project

User Community



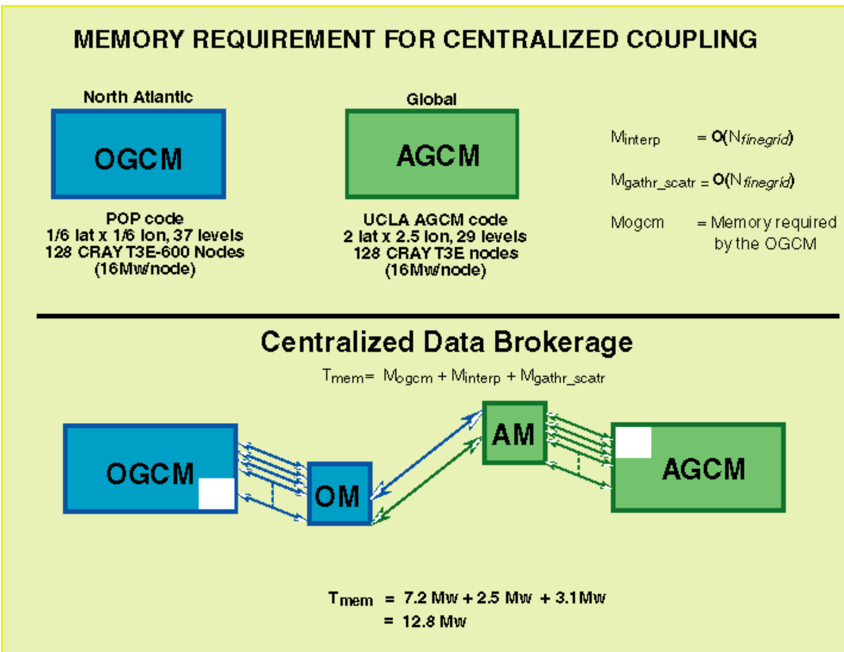
Minimum Requirements for Reusable High Quality Software Tools

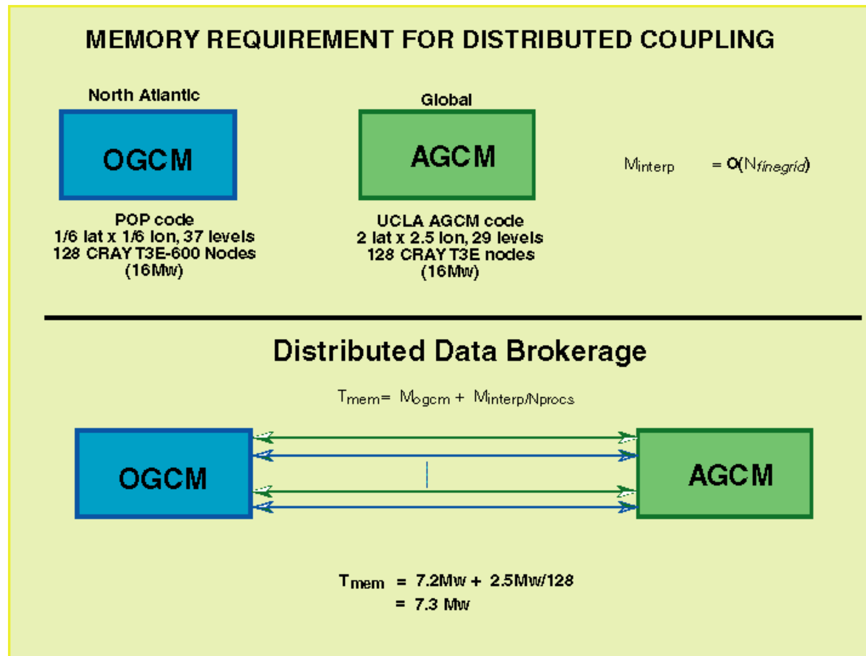
- Robust
- Scalable
- Extensible
- Interoperable
- User Friendly Interfaces
- Well documented
- Periodic Tests and Evaluations
- Portability and Fast Adaptability
- Long-term support
- Training (hands-on code) and High level support
- **Community support (developers, users, computer vendors, mailing lists, commercial software development and user groups)**

Open Challenges
DISTRIBUTED INTEGRATION

- Distributed Coupling

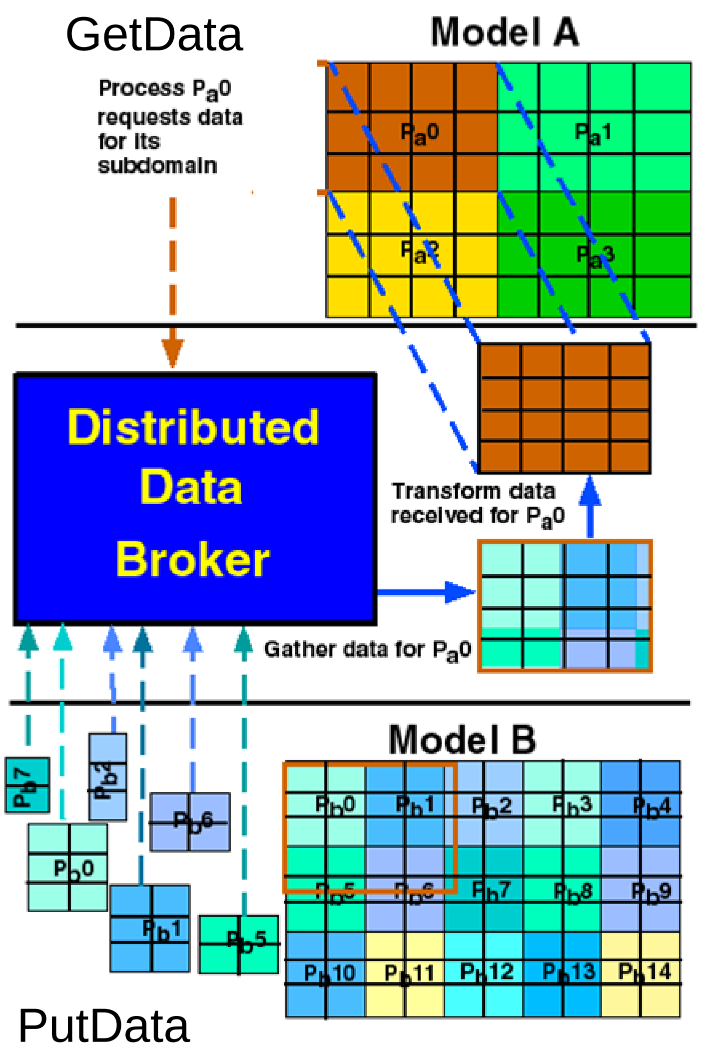
multi-physics, multi-resolutions, multi-domains





DISTRIBUTED INTEGRATION

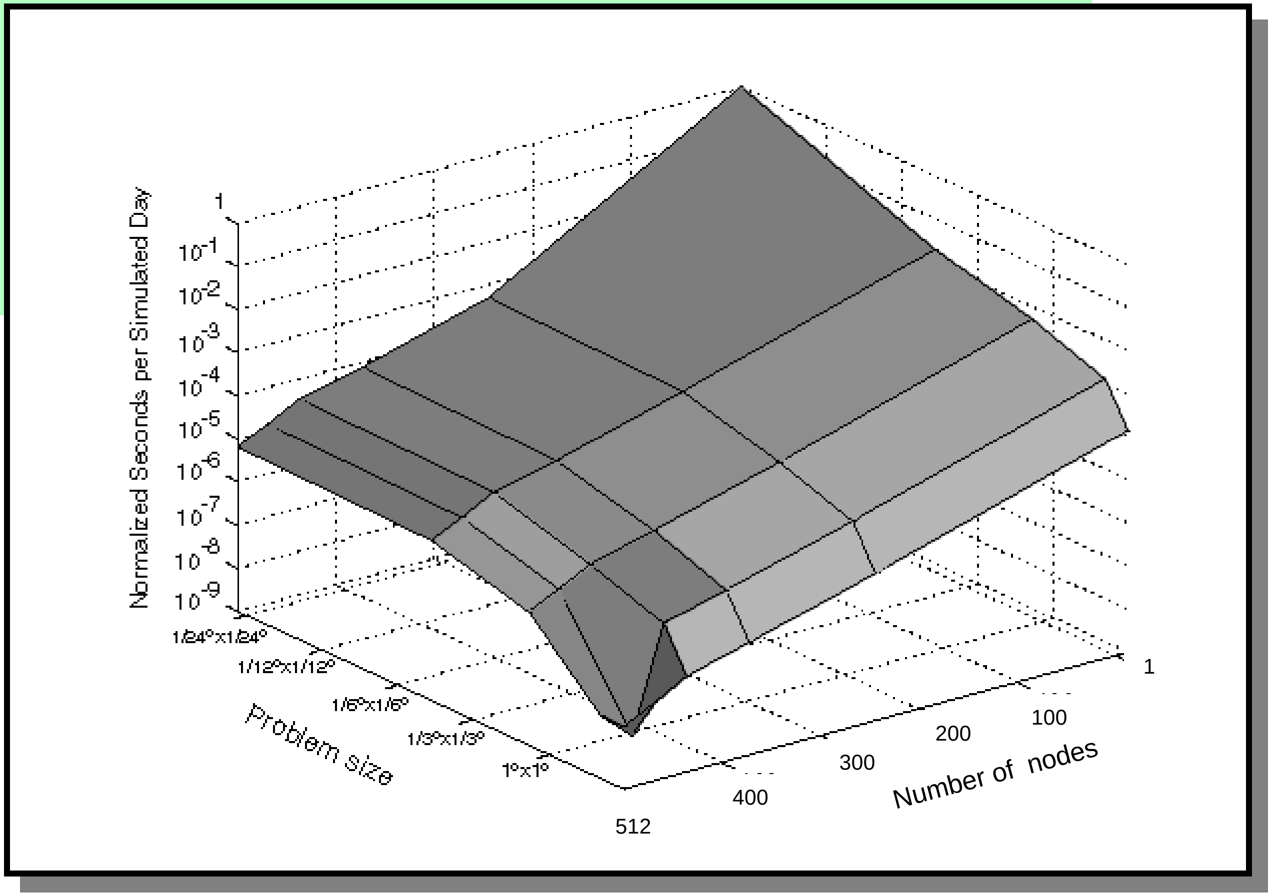
A single-controller free approach



- Avoid Intrusive lines of codes.
- Provide interfaces to formulate data (flux, variables, fields, etc. .) translations between applications.
- Variable Synchronization.
- Scalability!

Distributed Coupling

- Distributed Coupling



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

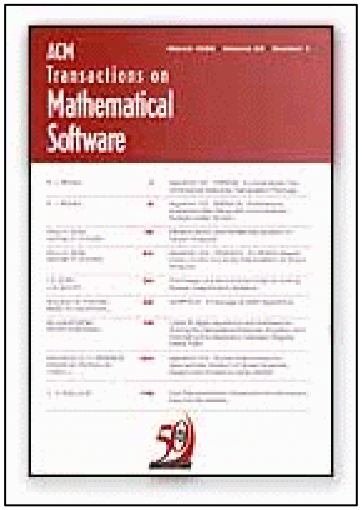
"Got Tools?"

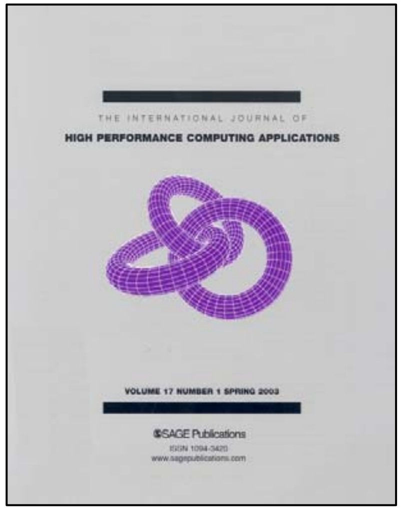
- There are several software development efforts enabling scientific and engineering applications meet the computational challenges at the Petaflops and beyond levels. We need to ensure that they meet the aforementioned tool requirements.
- *Good Trend*: High-level tool interfaces that hide software 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

ACM
TOMS





IJHPCA

- Sixth ACTS Collection Workshop, August 23-26, 2005