The DOE Advance CompuTational Software (ACTS) Collection

Discovery 2015: HPC and Cloud Computing Workshop
Berkeley, California, June 17, 2011

Tony Drummond
Computational Research Division
Lawrence Berkeley National Laboratory

ACTS COLLECTION

Office of Science
U.S. DEPARTMENT OF ENERGY

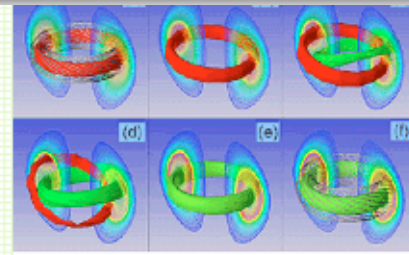
BERKELEY LAB

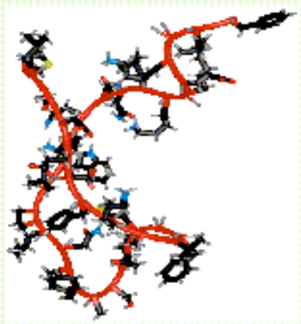
OUTLINE

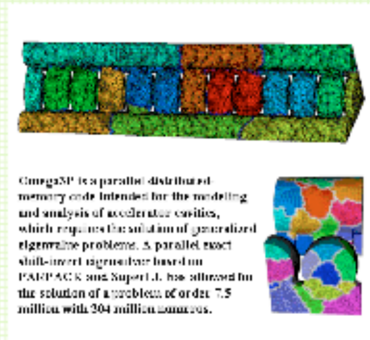
- Motivation
- Introduction to DOE ACTS Collection
- A view of selected functionality in the Collection
- Use of the Tools in the ACTS Collection
- Sustainability through emerging hardware
- Summary

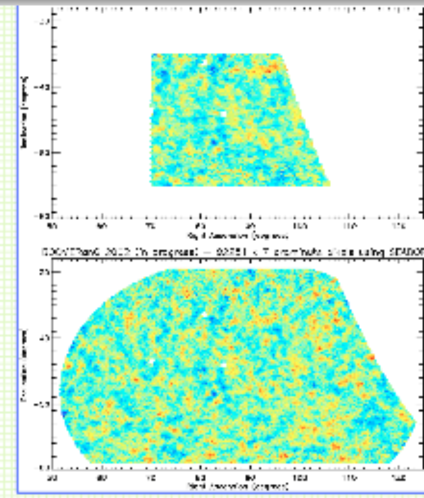
Motivation - HPC Applications

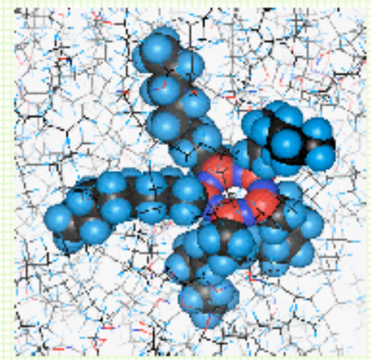
- Accelerator Science
- Astrophysics
- Biology
- Chemistry
- Earth Sciences
- Materials Science
- Nanoscience
- Plasma Science
-
-







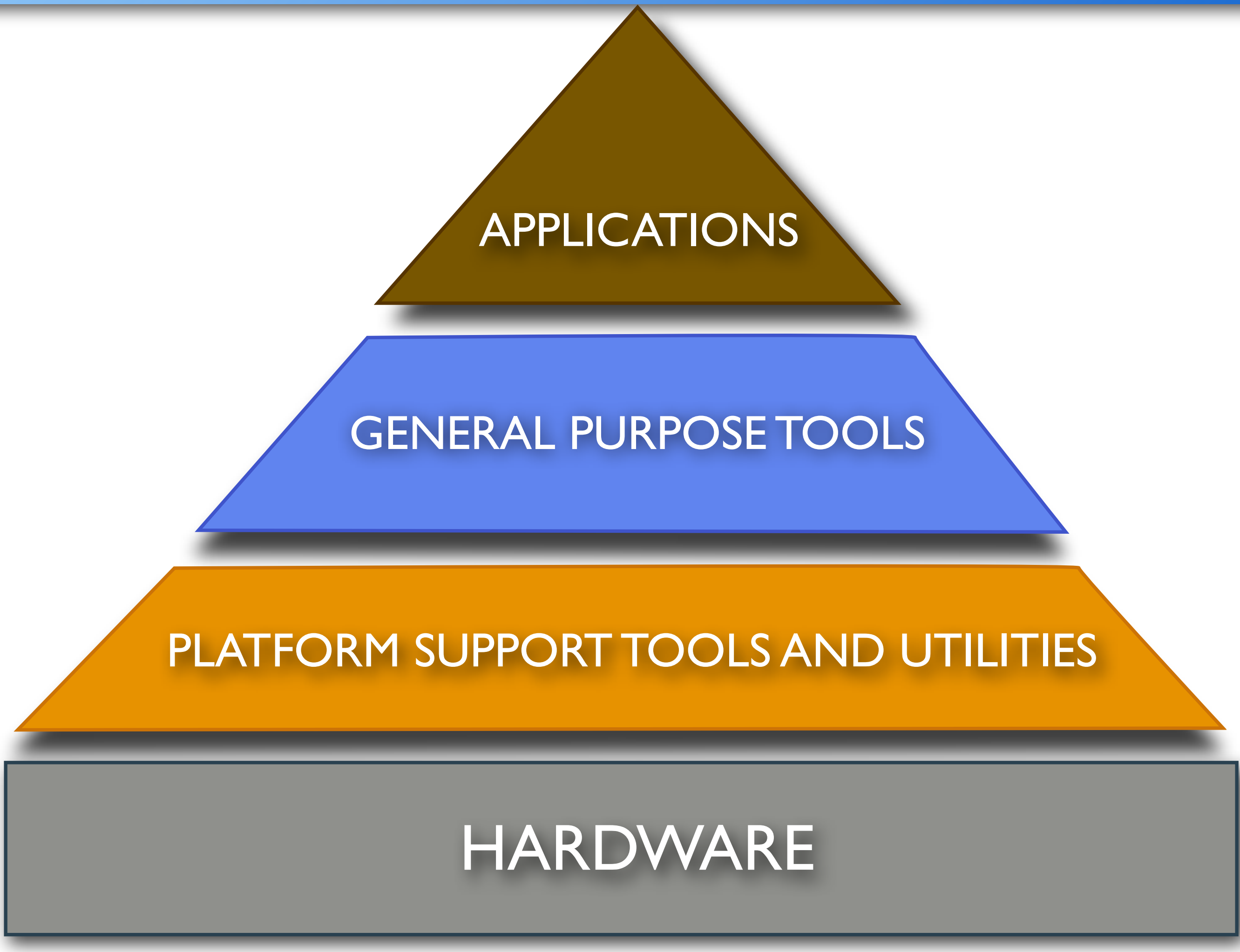




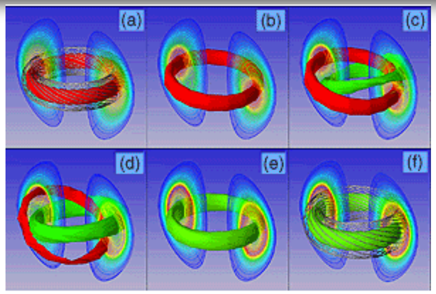
Commonalities:

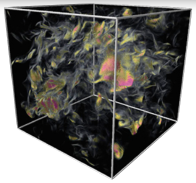
- Major advancements in Science
- Increasing demands for computational power
- Rely on available computational systems, languages, and software tools

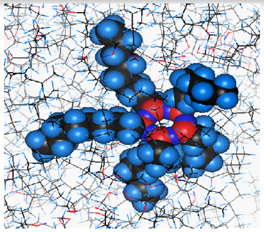
HPC Software Stack

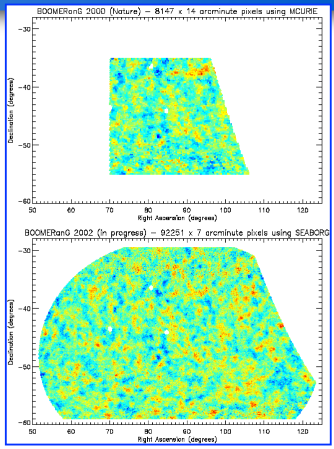


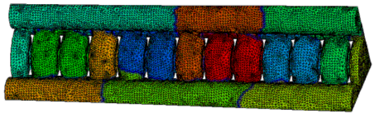
HPC Software Stack











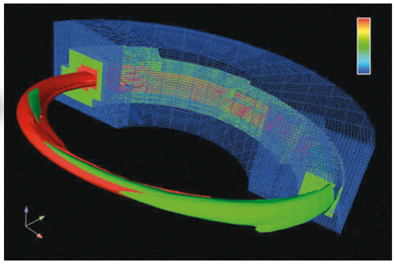
Omega3P is a parallel distributed-memory code intended for the modeling and analysis of accelerator cavities, which requires the solution of generalized eigenvalue problems. A parallel exact shift-invert eigensolver based on PARPACK and SuperLU has allowed for the solution of a problem of order 7.5 million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

APPLICATIONS

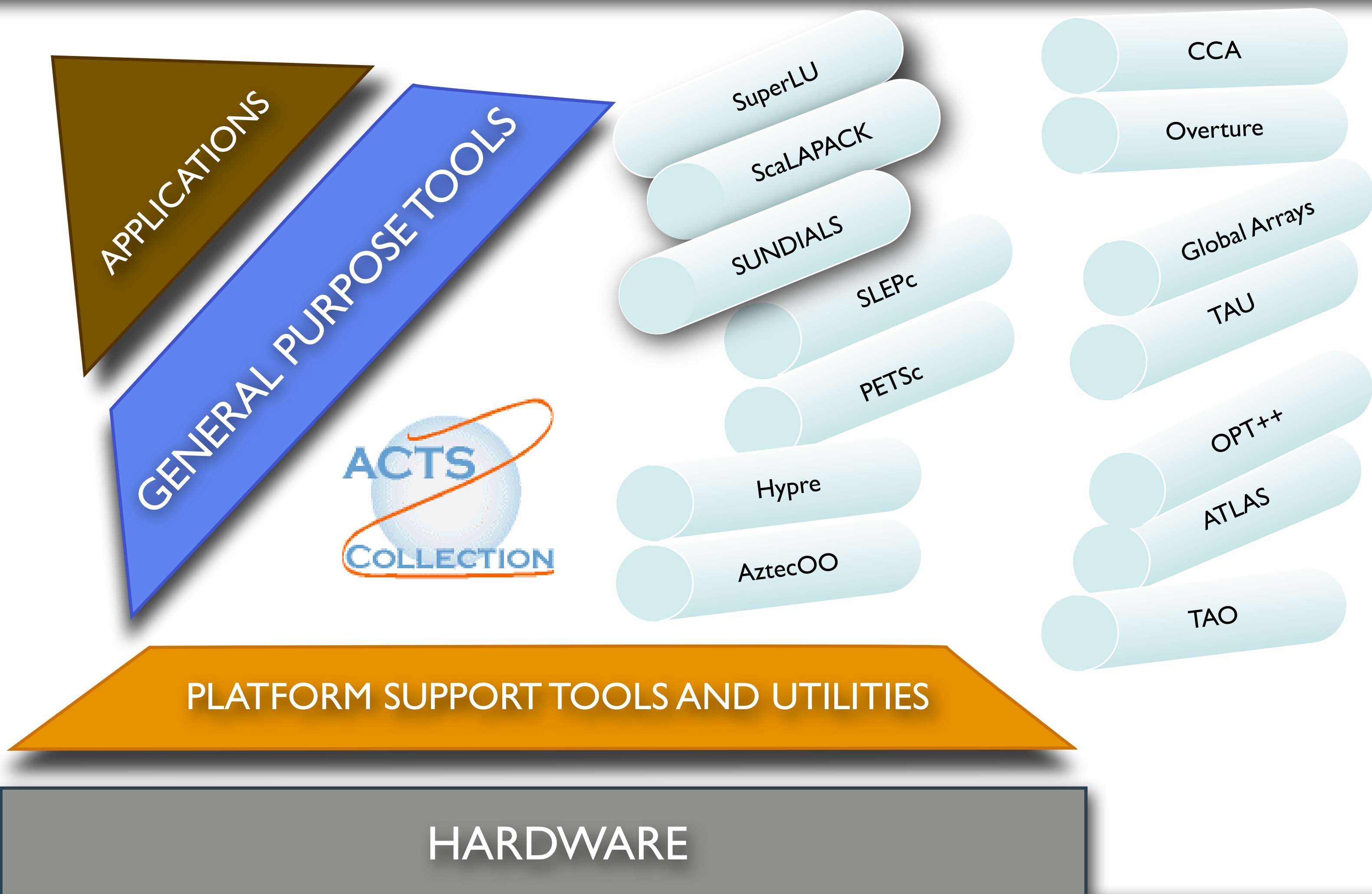
GENERAL PURPOSE TOOLS

PLATFORM SUPPORT TOOLS AND UTILITIES

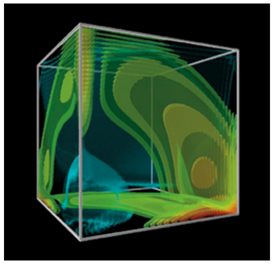
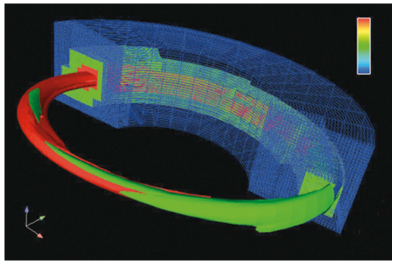
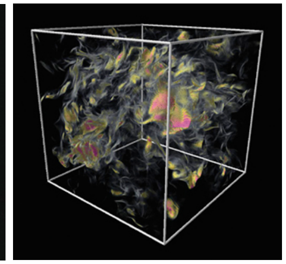
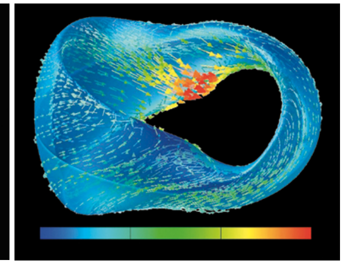
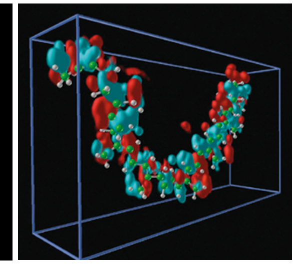
HARDWARE



HPC Software Stack



The DOE ACTS Collection

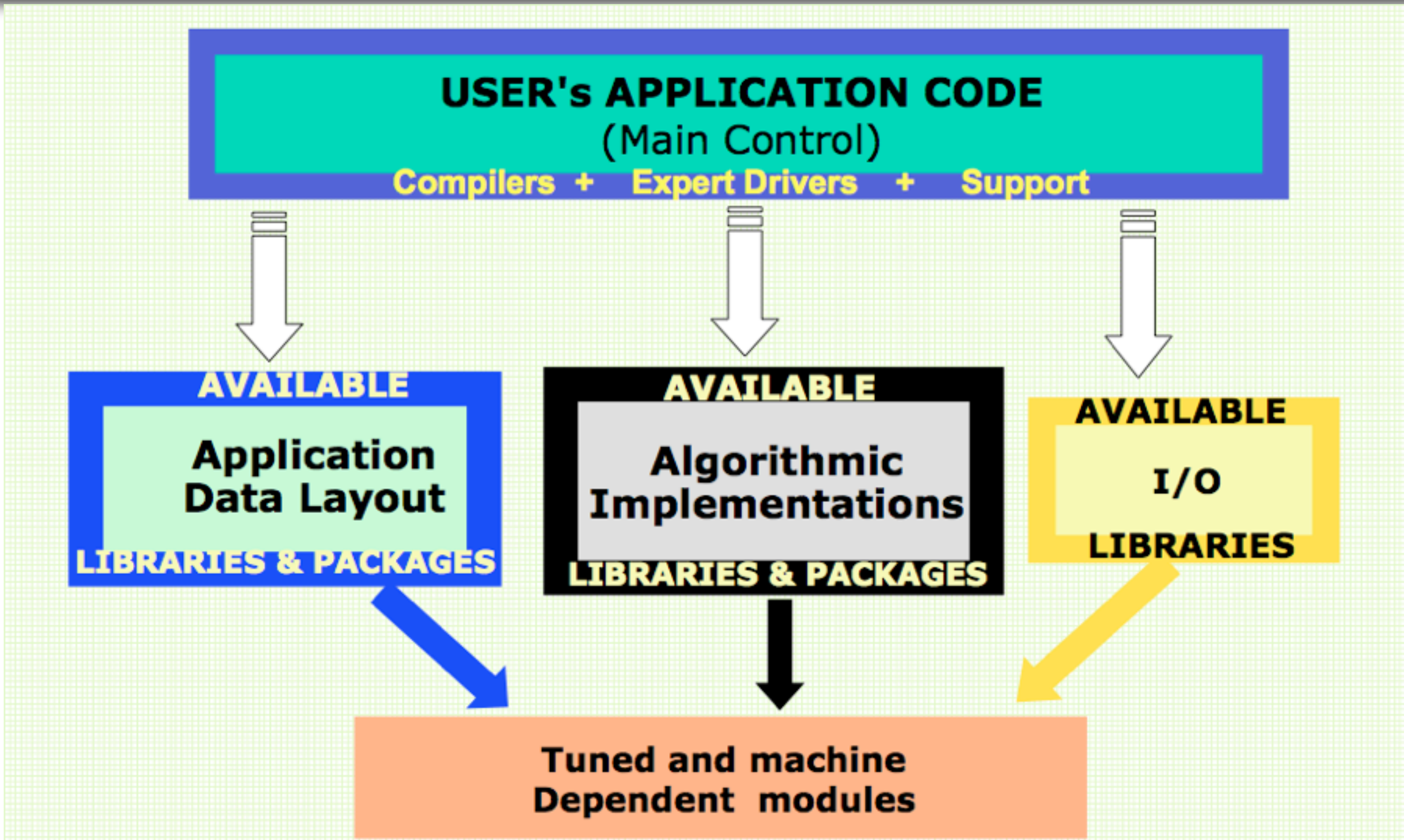
    

Goal: The Advanced CompuTational Software Collection (ACTS) makes reliable and efficient software tools more widely used, and more effective in solving the nation's engineering and scientific problems.

References:

- L.A. Drummond, O. Marques: An Overview of the Advanced CompuTational Software (ACTS) Collection. ACM Transactions on Mathematical Software Vol. 31 pp. 282-301, 2005
- http://acts.nersc.gov

Speeding-Up Software Development



The DOE ACTS Collection

Category	Tool	Functionalities
Numerical Libraries	AztecOO	Scalable linear and non-linear solvers using iterative schemes.
	Hypre	A family of scalable preconditioners.
	PETSc	Scalable linear and non-linear solvers and additional support for PDE related work.
	SUNDIALS	Solvers for the solution of systems of ordinary differential equations, nonlinear algebraic equations, and differential-algebraic equations.
	ScaLAPACK	High performance parallel dense linear algebra.
	SLEPc	Scalable algorithms for the solution of large sparse eigenvalue problems.
	SuperLU	Scalable direct solution of large, sparse, nonsymmetric linear systems of equations.
	TAO	Large-scale optimization software.
Code Development	Global Arrays	Supports the development of parallel programs.
	Overture	Supports the development of computational fluid dynamics codes in complex geometries.
Run Time Support	TAU	Portable and scalable performance analyzes and tracing tools for C, C++, Fortran and Java programs.
Library Development	ATLAS	Automatic generation of optimized numerical dense algebra for scalar processors.

Latest Set of Tutorials and Exercises

http://acts.nersc.gov/events/Workshop2010

Upcoming 12th ACTS Collection Workshop

http://acts.nersc.gov/events/Workshop2011

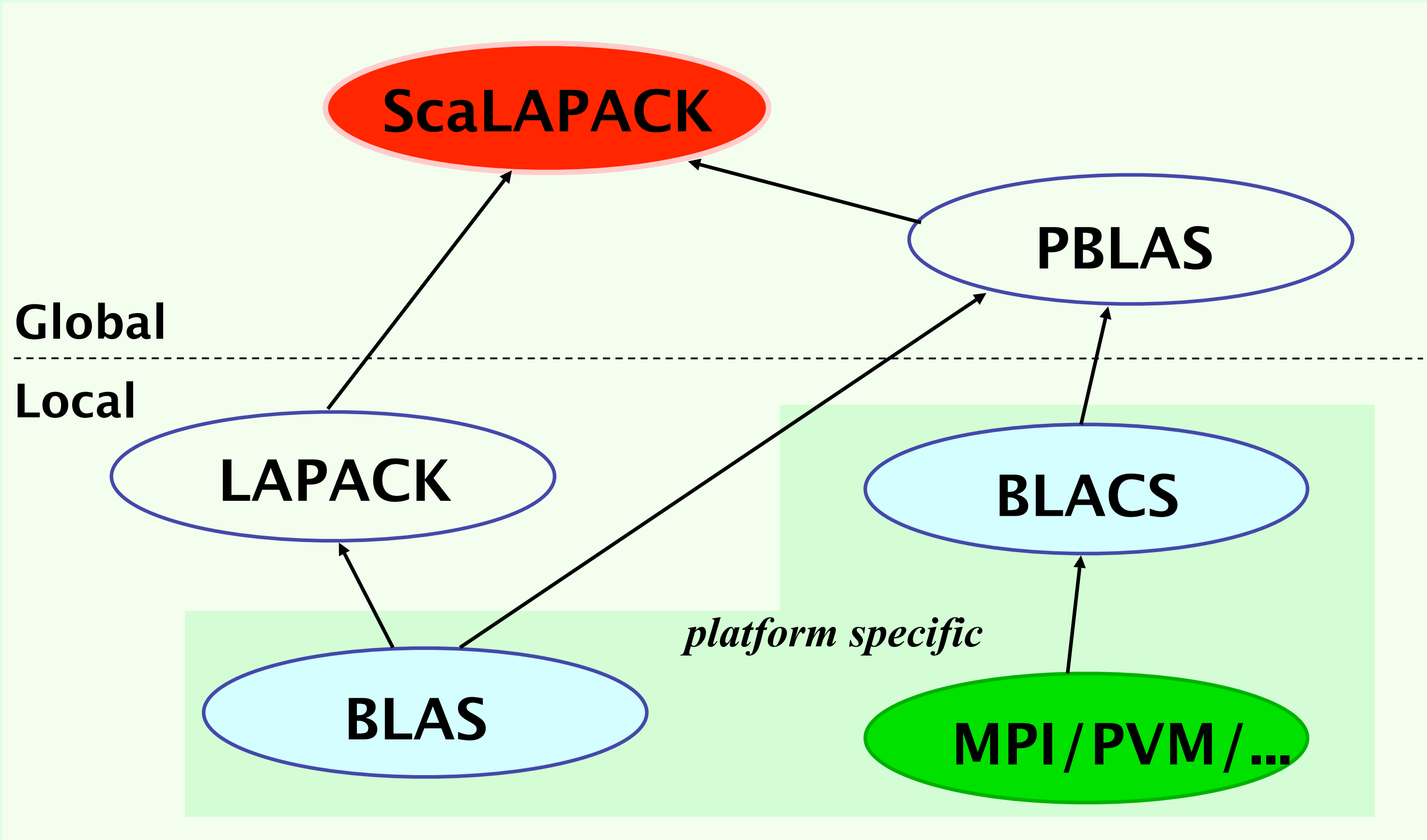
Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations	Direct Methods	LU Factorization	ScaLAPACK(dense) SuperLU (sparse)
		Cholesky Factorization	ScaLAPACK
		LDL^T (Tridiagonal matrices)	ScaLAPACK
		QR Factorization	ScaLAPACK
		QR with column pivoting	ScaLAPACK
		LQ factorization	ScaLAPACK

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations	Direct Methods	LU Factorization	ScaLAPACK(dense) SuperLU (sparse)
		Cholesky Factorization	ScaLAPACK
		LDL^T (Tridiagonal matrices)	ScaLAPACK
		QR Factorization	ScaLAPACK
		QR with column pivoting	ScaLAPACK
		LQ factorization	ScaLAPACK

Software Structure of ScaLAPACK



Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations *(cont..)*	Iterative Methods	Conjugate Gradient	AztecOO (Trilinos) PETSc
		GMRES	AztecOO PETSc Hypre
		CG Squared	AztecOO PETSc
		Bi-CG Stab	AztecOO PETSc
		Quasi-Minimal Residual (QMR)	AztecOO
		Transpose Free QMR	AztecOO PETSc

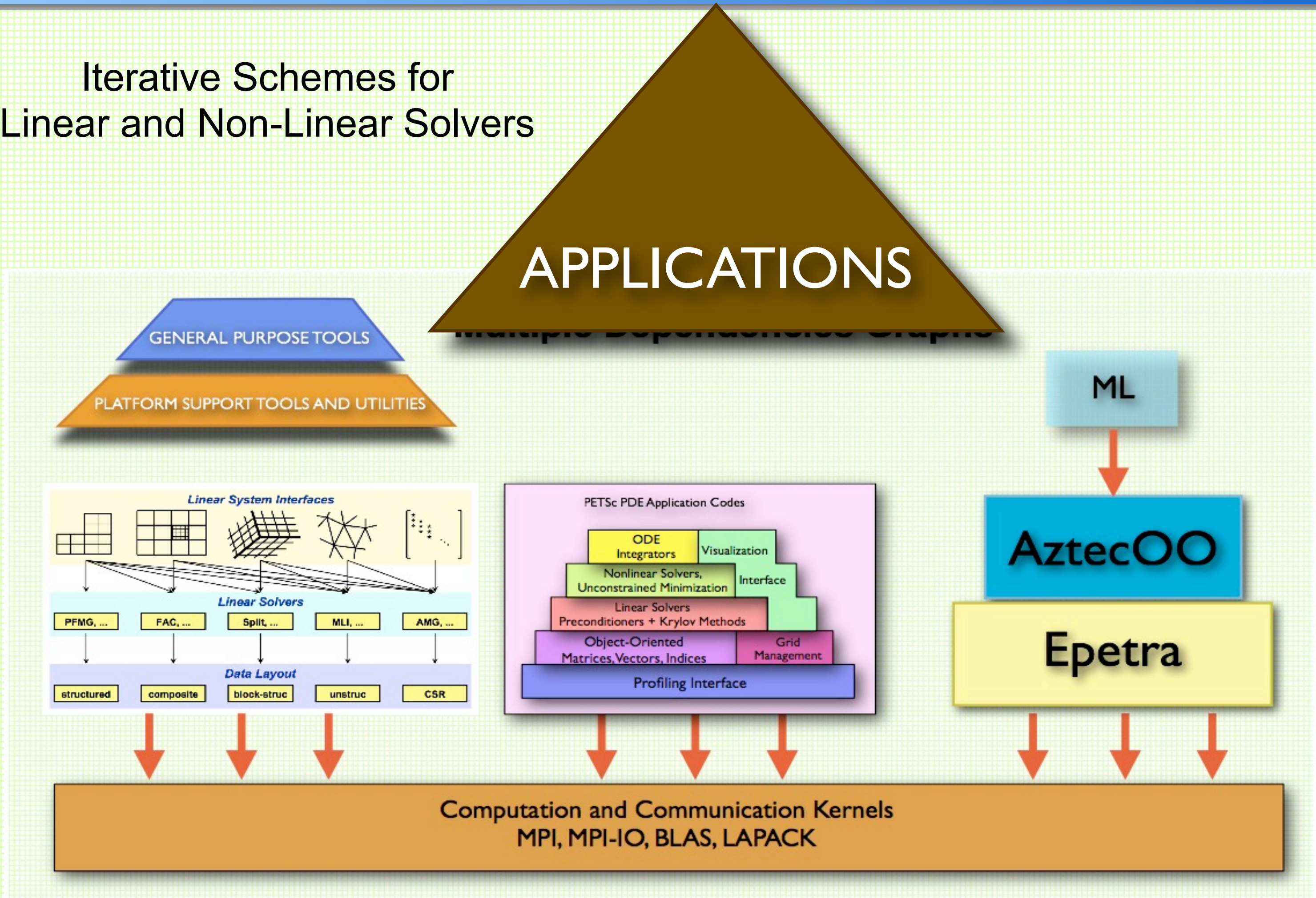
Trilinos

Full Vertical Solver Coverage

Problem class	Formulation		Package
Optimization **Unconstrained:** **Constrained:**	Find $u \in \Re^n$ that minimizes $g(u)$ Find $x \in \Re^m$ and $u \in \Re^n$ that minimizes $g(x,u)$ s.t. $f(x,u) = 0$	**Sensitivities** (Automatic Differentiation: Sacado)	MOOCHO
Bifurcation Analysis	Given nonlinear operator $F(x,u) \in \Re^{n+m}$ For $F(x,u) = 0$ find space $u \in U \ni \frac{\partial F}{\partial x}$		LOCA
Transient Problems **DAEs/ODEs:**	Solve $f(\dot{x}(t), x(t), t) = 0$ $t \in [0,T], x(0) = x_0, \dot{x}(0) = x_0'$ for $x(t) \in \Re^n, t \in [0,T]$		Rythmos
Nonlinear Problems	Given nonlinear operator $F(x) \in \Re^m \rightarrow \Re$ Solve $F(x) = 0 \quad x \in \Re^n$		NOX
Linear Problems **Linear Equations:** **Eigen Problems:**	Given Linear Ops (Matrices) $A, B \in \Re^{m \times n}$ Solve $Ax = b$ for $x \in \Re^n$ Solve $A\nu = \lambda B\nu$ for (all) $\nu \in \Re^n, \quad \lambda \in$		AztecOO Belos Ifpack, ML, etc... Anasazi
Distributed Linear Algebra **Matrix/Graph Equations:** **Vector Problems:**	Compute $y = Ax; A = A(G); A \in \Re^{m \times n}, G \in \Im^{m \times n}$ Compute $y = \alpha x + \beta w; \alpha = \langle x, y \rangle; x, y \in \Re^n$		Epetra Tpetra

Overlapping Functionality

Iterative Schemes for
Linear and Non-Linear Solvers



Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations *(cont..)*	Iterative Methods	Conjugate Gradient	AztecOO (Trilinos) PETSc
		GMRES	AztecOO PETSc Hypre
		CG Squared	AztecOO PETSc
		Bi-CG Stab	AztecOO PETSc
		Quasi-Minimal Residual (QMR)	AztecOO
		Transpose Free QMR	AztecOO PETSc

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations *(cont..)*	Iterative Methods *(cont..)*	SYMMLQ	PETSc
		Precondition CG	AztecOO PETSc Hypre
		Richardson	PETSc
		Block Jacobi Preconditioner	AztecOO PETSc Hypre
		Point Jocobi Preconditioner	AztecOO
		Least Squares Polynomials	PETSc

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations *(cont..)*	Iterative Methods *(cont..)*	SOR Preconditioning	PETSc
		Overlapping Additive Schwartz	PETSc
		Approximate Inverse	Hypre
		Sparse LU preconditioner	AztecOO PETSc Hypre
		Incomplete LU (ILU) preconditioner	AztecOO
		Least Squares Polynomials	PETSc
	MultiGrid (MG) Methods	MG Preconditioner	PETSc Hypre
		Algebraic MG	Hypre
		Semi-coarsening	Hypre

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Linear Least Squares Problems	Least Squares	$min_x \parallel b - Ax \parallel_2$	ScaLAPACK
	Minimum Norm Solution	$min_x \parallel x \parallel_2$	ScaLAPACK
	Minimum Norm Least Squares	$min_x \parallel b - Ax \parallel_2$ $min_x \parallel x \parallel_2$	ScaLAPACK
Standard Eigenvalue Problem	Symmetric Eigenvalue Problem	$Az = \lambda z$ For $A=A^H$ or $A=A^T$	ScaLAPACK (dense) SLEPc (sparse)
Singular Value Problem	Singular Value Decomposition	$A = U\Sigma V^T$ $A = U\Sigma V^H$	ScaLAPACK (dense) SLEPc (sparse)
Generalized Symmetric Definite Eigenproblem	Eigenproblem	$Az = \lambda Bz$ $ABz = \lambda z$ $BAz = \lambda z$	ScaLAPACK (dense) SLEPc (sparse)

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Non-Linear Equations	Newton Based	Line Search	PETSc
		Trust Regions	PETSc
		Pseudo-Transient Continuation	PETSc
		Matrix Free	PETSc

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization	Newton Based	Newton	OPT++ TAO
		Finite-Difference Newton	OPT++ TAO
		Quasi-Newton	OPT++ TAO
		Non-linear Interior Point	OPT++ TAO
	CG	Standard Non-linear CG	OPT++ TAO
		Limited Memory BFGS	OPT++
		Gradient Projections	TAO
	Direct Search	No derivate information	OPT++

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization	Newton Based	Newton	OPT++
			TAO
		Finite-Difference Newton	OPT++
			TAO
		Quasi-Newton	OPT++
			TAO
		Non-linear Interior Point	OPT++
			TAO
	CG	Standard Non-linear CG	OPT++
			TAO
		Limited Memory BFGS	OPT++
		Gradient Projections	TAO
	Direct Search	No derivate information	OPT++

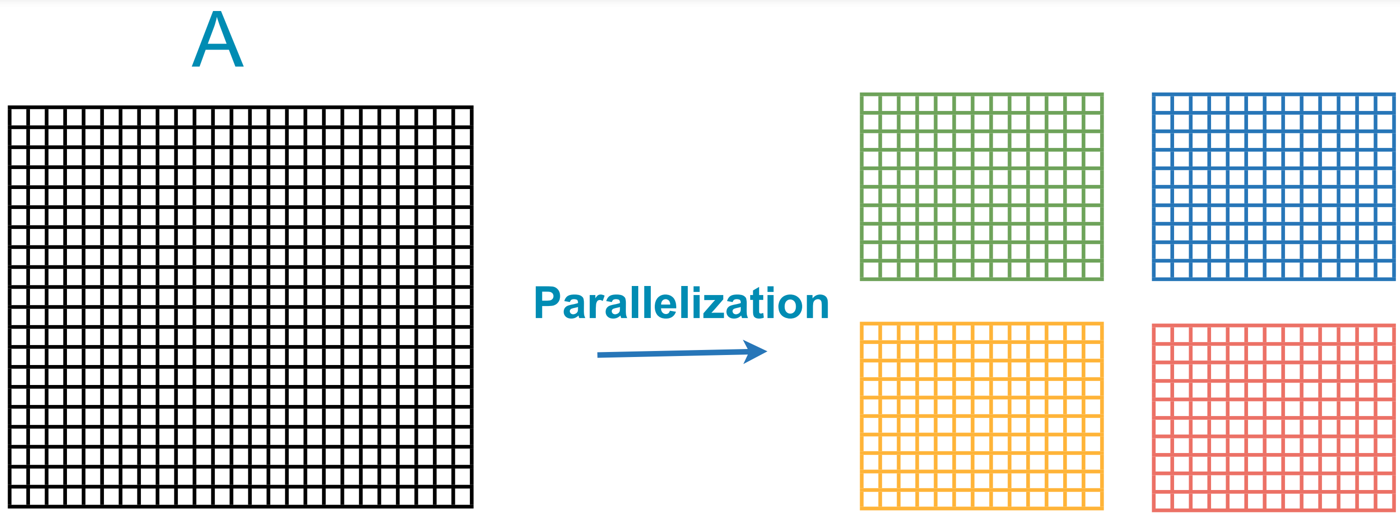
Functionality in The DOE ACTS Collection

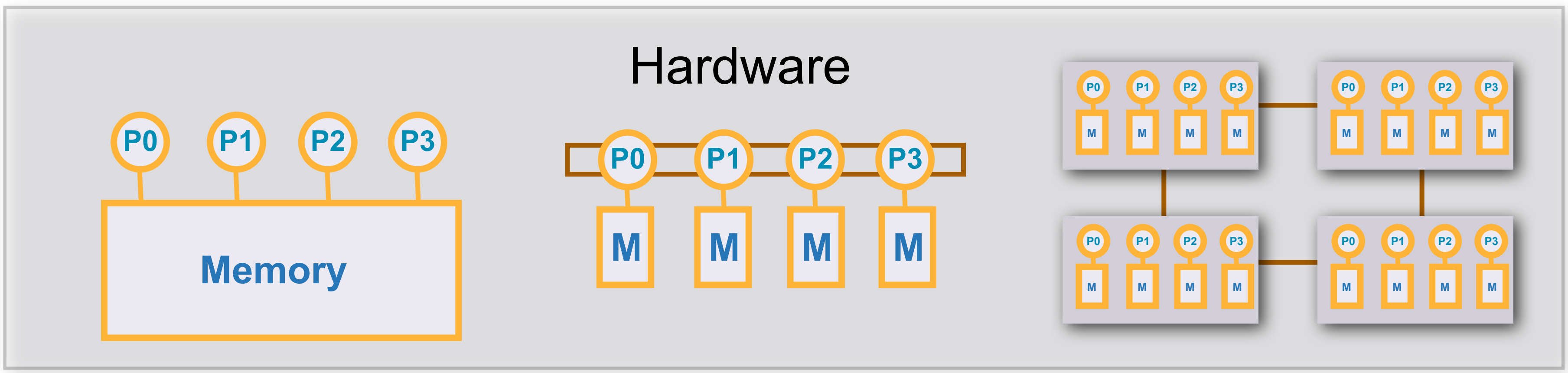
Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization (cont..)	Semismoothing	Feasible Semismooth	TAO
		Unfeasible semismooth	TAO
Ordinary Differential Equations	Integration	Adam-Moulton (Variable coefficient forms)	CVODE (SUNDIALS) CVODES
	Backward Differential Formula	Direct and Iterative Solvers	CVODE CVODES
Nonlinear Algebraic Equations	Inexact Newton	Line Search	KINSOL (SUNDIALS)
Differential Algebraic Equations	Backward Differential Formula	Direct and Iterative Solvers	IDA (SUNDIALS)

Functionality in The DOE ACTS Collection

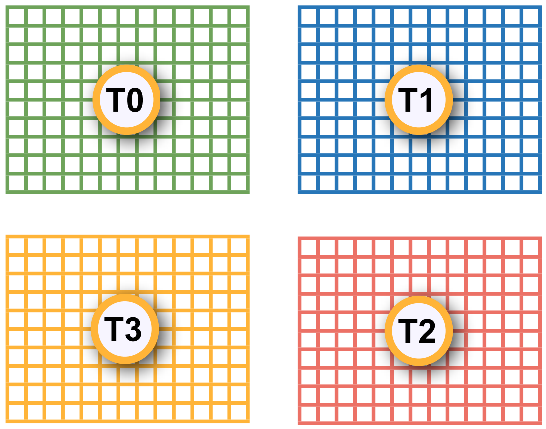
Computational Problem	Support	Techniques	Library
Writing Parallel Programs	Distributed Arrays	Shared-Memory	Global Arrays
		Grid Generation	OVERTURE
		Structured Meshes	Hypre OVERTURE PETSc
		Semi-Structured Meshes	Hypre OVERTURE

Parallel Programming Paradigm



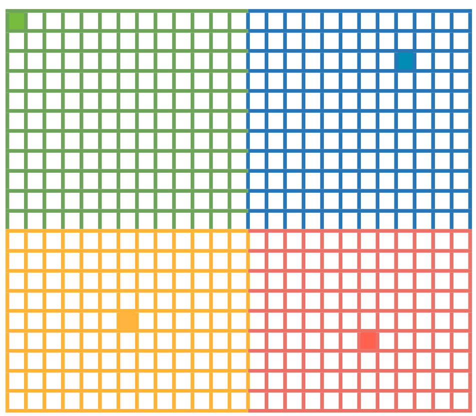


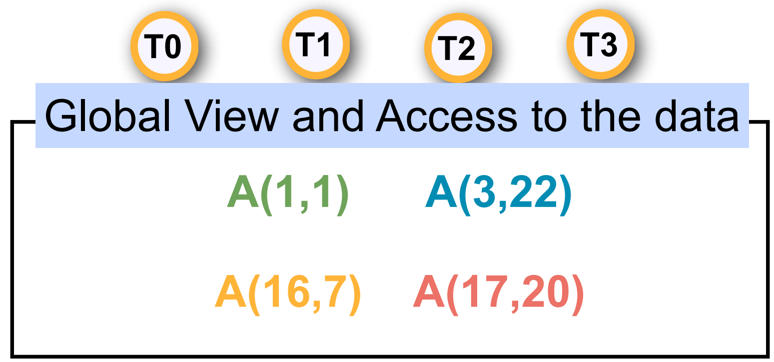
Shared Memory Programming Paradigm



Program Design Considerations

- Data Distribution (problem partition)
- Interprocess work coordination
  - load sharing
  - communication
  - synchronization





Comparing Programming Paradigms

Message Passing Paradigm (MPI or PVM)

```
if (my_id != 0) {
         1) pack data in message
         2) send message to task 0
else for i=1,3 {
     receive message from task ( i )
     unpack message
     place the data in message received in local data array }
end if
```

Global Arrays

```
if (my_id != 0) {
         NGA_GET(A,lo,hi,local_array,stride)
else for i=1,3 {
         NGA_PUT(A,lo,hi,local_array,stride)
end if
```

Functionality in The DOE ACTS Collection

Computational Problem	Support	Technique	Library
Profiling	Algorithmic Performance	Automatic instrumentation	PETSc
		User Instrumentation	PETSc
	Execution Performance	Automatic Instrumentation	TAU
		User Instrumentation	TAU
Code Optimization	Library Installation	Linear Algebra Tuning	ATLAS

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 )
```

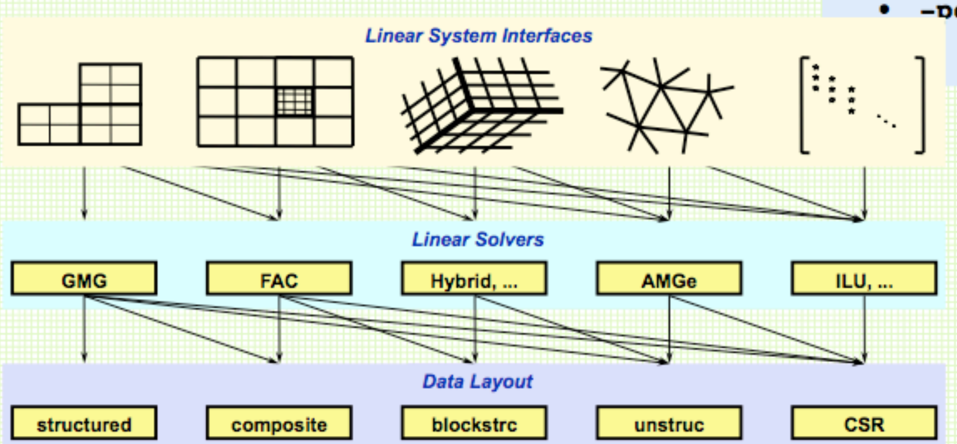
Library Calls

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

More advanced:

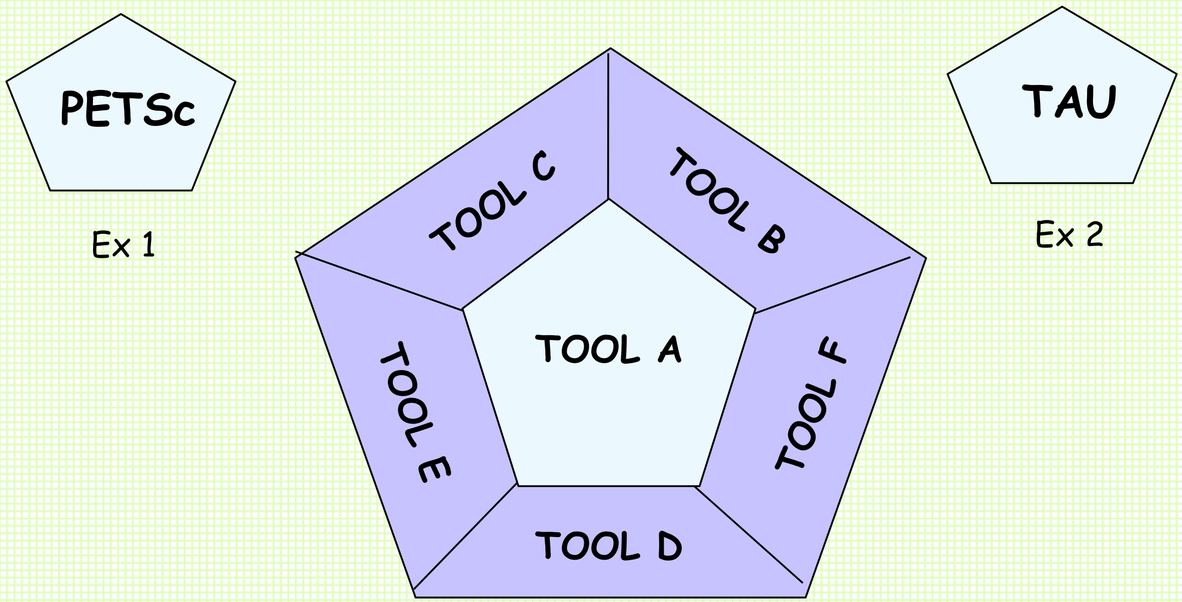
- **-ksp_max_it <max_iters>**
- **-ksp_gmres_restart <restart>**
- **-pc_asm_overlap <overlap>**

Command lines

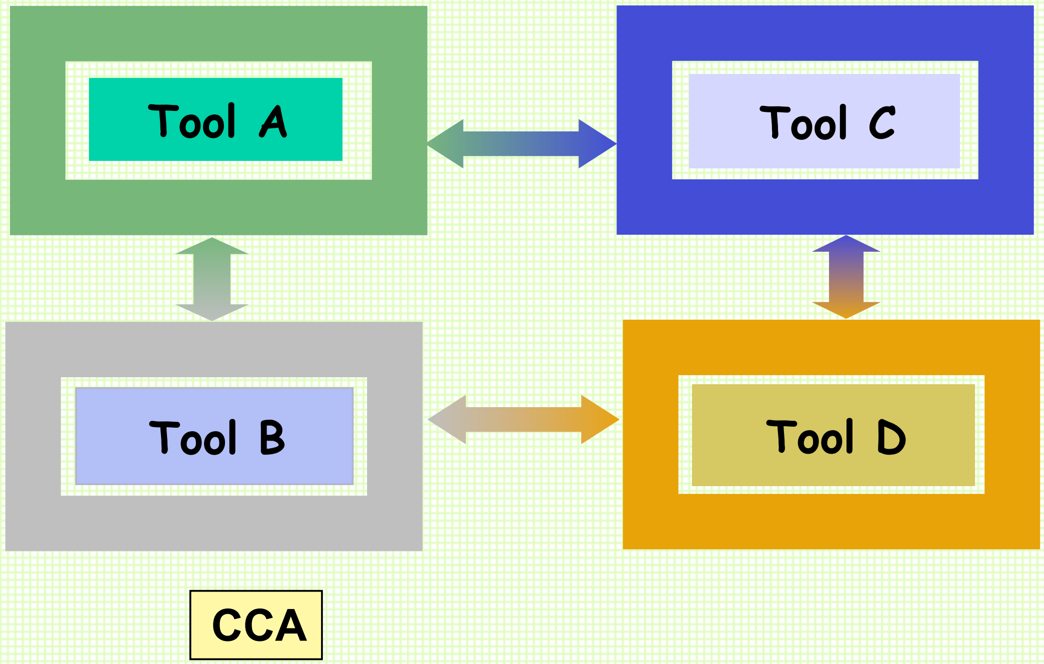


Problem Domain

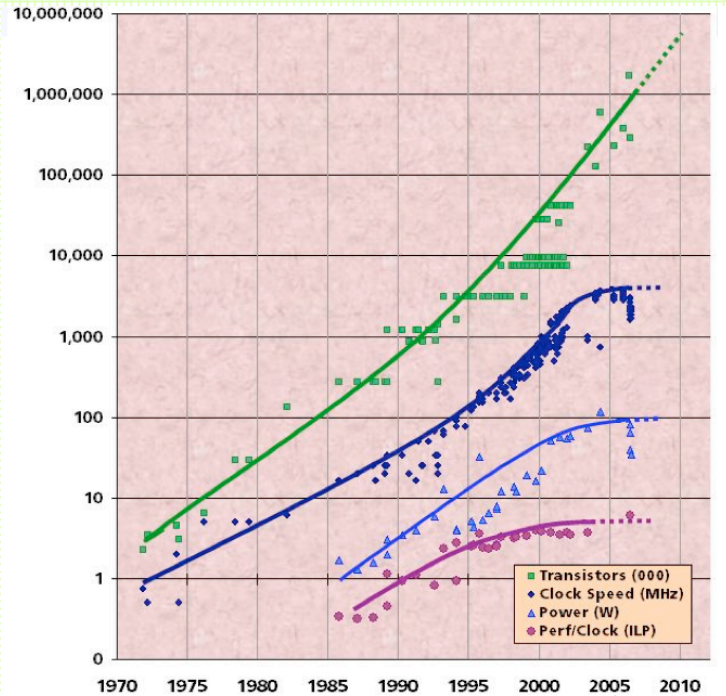
Interoperability



Interoperability



Scalability and Sustainability



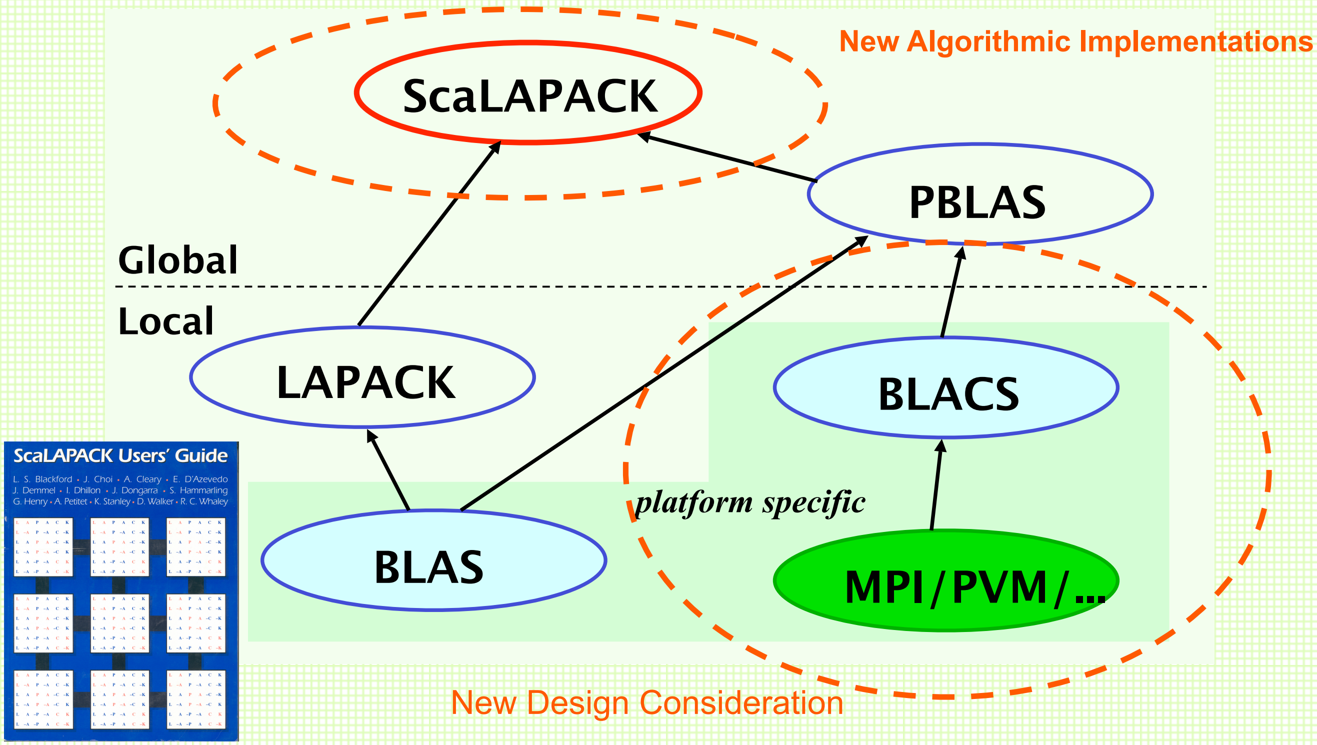
K. Olukotun et. al.,

- Number of cores per node will continue to increase.
- Memory per node is not increasing at the same rate

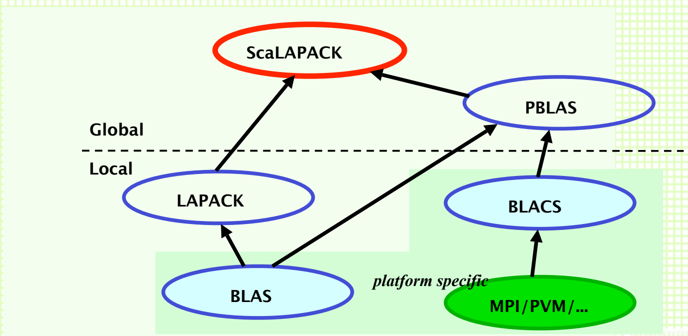
Looking back in history. ..

- Processor speed increased at faster rate than memory speed.
- Memory bandwidth relevant to overall performance

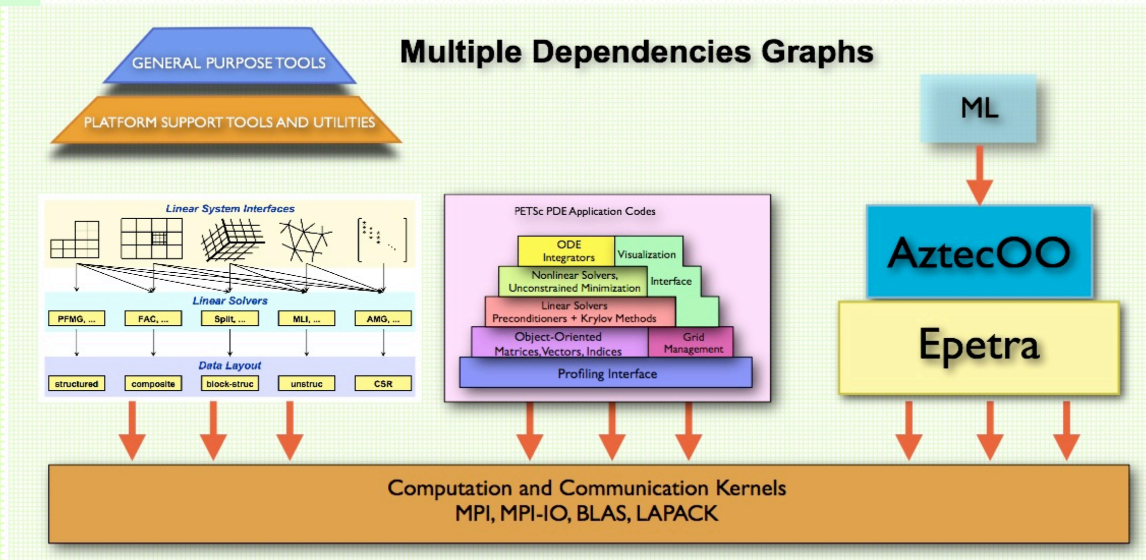
Scalability and Sustainability



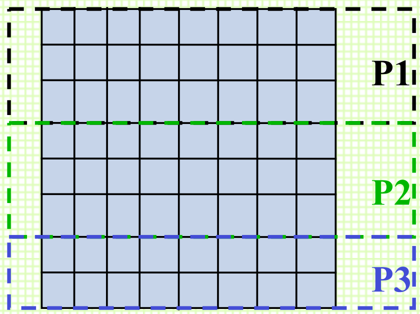
Software Dependency Graph



- Identify key common kernels
- Identify parameters that drive performance
- Profile and test (bottom-up)



Kernel Optimization

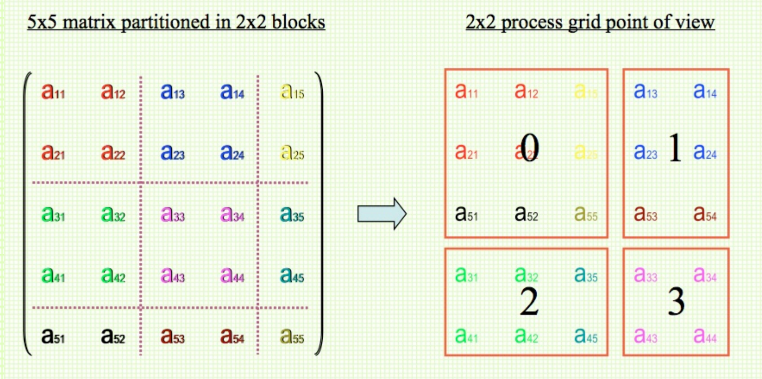


PETSc

Hypre

AztecOO

Row-Projection Methods



Dense Linear Algebra

SuperLU

ScaLAPACK

Kernel Optimization

$Ax = b$ or $AX = B$

$Hx = b'$

$min_x \; || \; b - Ax \; ||_2$

$min_x \; || \; x \; ||_2$

$min_x \; || \; b - Ax \; ||_2$

$min_x \; || \; x \; ||_2$

$Az = \lambda z$

$A = U \Sigma V^T$

$A = U \Sigma V^H$

$Az = \lambda Bz$

$ABz = \lambda z$

$BAz = \lambda z$

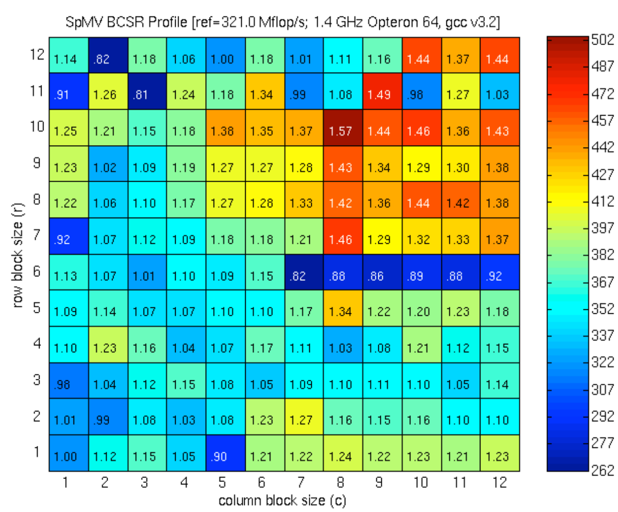
Exploit concurrency :
(in and out a node)
- Hybrid programming (MPI+threads)
- NUMA Aware operations

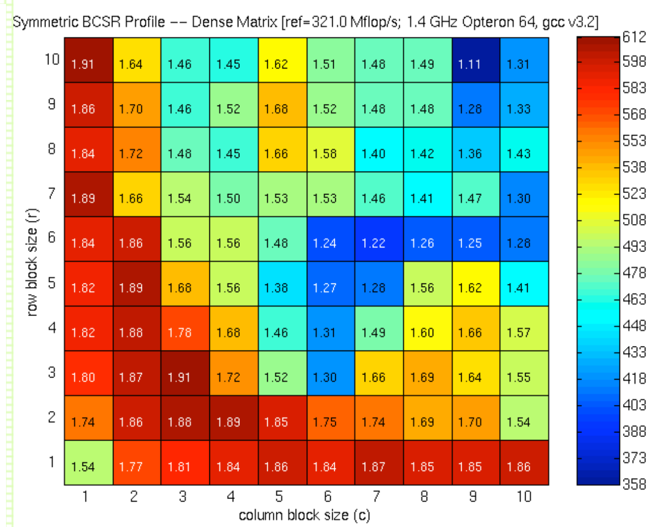
Kernel reusability:
- Bottom-Up automatic optimization
- Identify key parameters in the algorithm
- Run-time parameter control

Functionality in The DOE ACTS Collection

OSKI

Symmetric peak = 612 MFlops

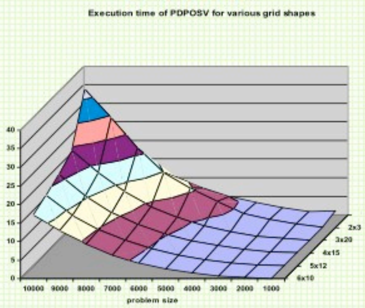




University of California, Berkeley

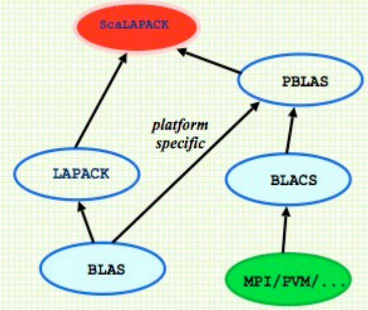
ACTS Software Sustainability Cycle

Profiling and Tracing Tools: TAU

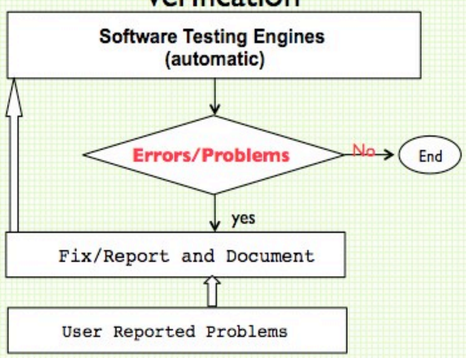


Performance and Scalability

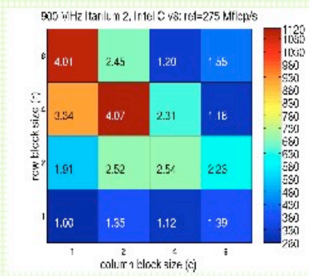
Software Dependency Graph



Automatic Testing and Verification



Auto-Tuning (OSKI, ATLAS,)



Summary

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*]

References

- L.A. Drummond, O. Marques: An Overview of the Advanced CompuTational Software (ACTS) Collection. ACM Transactions on Mathematical Software Vol. 31 pp. 282-301, 2005

- http://acts.nersc.gov

- http://acts.nersc.gov/events/Workshop2010

- LiveDVD: http://www.paratools.com/livedvd.php