An Overview of the ACTS Collection

Osni Marques
(*oamarques@lbl.gov*)

with special thanks to

The Japan Society for the Progress of Science

The DOE ACTS Collection

http://acts.nersc.gov

- **A**dvanced **C**ompu**T**ational **S**oftware Collection
- Tools for developing parallel applications
- ACTS started as an "umbrella" project

Goals

- *Extended support for experimental software*
- *Make ACTS tools available on DOE computers*
- *Provide technical support* (acts-support@nersc.gov)
- *Maintain ACTS information center* (http://acts.nersc.gov)
- *Coordinate efforts with other supercomputing centers*
- *Enable large scale scientific applications*
- *Educate and train*

Current ACTS Tools and their Functionalities

Category	Tool	Functionalities	
Numerical $Ax = b$ $Az = \lambda z$ $A = U\Sigma V^T$ PDEs ODEs $\vdots$	**Trilinos**	Algorithms for the iterative solution of large sparse linear systems (includes AZTEC00)	SNL
	Hypre	Algorithms for the iterative solution of large sparse linear systems, intuitive grid-centric interfaces, and dynamic configuration of parameters.	LLNL
	PETSc	Tools for the solution of PDEs that require solving large-scale, sparse linear and nonlinear systems of equations.	ANL
	OPT++	Object-oriented nonlinear optimization package.	SNL
	SUNDIALS	Solvers for the solution of systems of ordinary differential equations, nonlinear algebraic equations, and differential-algebraic equations.	LLNL
	ScaLAPACK	Library of high performance dense linear algebra routines for distributed-memory message-passing.	UCB, UT
	SLEPc	Eigensolver package built on top of PETSc	U P Valencia
	SuperLU	General-purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations.	LBNL
	TAO	Large-scale optimization software, including nonlinear least squares, unconstrained minimization, bound constrained optimization, and general nonlinear optimization.	ANL
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.	PPNL
	Overture	Object-Oriented tools for solving computational fluid dynamics and combustion problems in complex geometries.	LLNL
Code Execution	**TAU**	Set of tools for analyzing the performance of C, C++, Fortran and Java programs.	U Oregon
Library Development	**ATLAS**	Tools for the automatic generation of optimized numerical software for modern computer architectures and compilers.	UT

ACTS Tools: *numerical functionalities*

Computational Problem	Methodology	Algorithms	Library
Linear Equations	**Direct Methods**	*LU* factorization	ScaLAPACK (dense), SuperLU (sparse)
		Cholesky factorization	**ScaLAPACK**
		***LDL*^{T} factorization (tridiagonal matrices)**	
		QR factorization	
		QR factorization with column pivoting	
		LQ factorization	
		Full orthogonal factorization	
		Generalized QR factorization	
	Iterative Methods	**Conjugate gradient (CG)**	**AztecOO (Trilinos), PETSc**
		GMRES	**AztecOO, Hypre, PETSc**
		CG Squared	**AztecOO, PETSc**
		Bi-CG-Stab	
		QMR	**AztecOO**
		Transpose free QMR	**AztecOO, PETSc**
		SYMMLQ	**PETSc**
		Richardson	
		Block Jacobi preconditioner	**AztecOO, Hypre, PETSc**
		Point Jacobi preconditioner	**AztecOO**
		Least-squares polynomials	
		SOR preconditioner	**PETSc**
		Overlapping additive Schwarz	
		Approximate inverse	**Hypre**
		Sparse LU preconditioner	**AztecOO, Hypre, PETSc**
		Incomplete LU (ILU) preconditioner	
	Multigrid	**MG preconditioner**	**Hypre, PETSc**
		Algebraic multigrid	**ML (Trilinos), Hypre**
		Semicoarsening	**Hypre**

ACTS Tools: *numerical functionalities*

Computational Problem	Methodology	Algorithms	Library
Linear least squares	least squares	$min_x \| b - Ax \|_2$	ScaLAPACK
	minimum norm	$min_x \| x \|_2$	
	minimum norm least squares	$min_x \| x \|_2$ and $min_x \| b - Ax \|_2$	
Standard eigenvalue problems	iterative, direct	$Az=\lambda z$ for $A=A^T$ or $A=A^H$	ScaLAPACK (dense), SLEPc (sparse)
Generalized eigenvalue problems		$Az=\lambda Bz$, $ABz=\lambda z$, $BAz=\lambda z$	
Singular value decomposition		$A=U\Sigma V^T$, $A=U\Sigma V^H$	
Non-linear equations problems	Newton-based	Line search	PETSc, KINSOL (SUNDIALS)
		Trust regions	PETSc
		Pseudotransient continuation	PETSc
		Matrix-free	PETSc
Nonlinear optimization	Newton-based	Newton	OPT++, TAO
		Finite differences	OPT++
		Quasi-Newton	OPT++, TAO (LMVM)
		Nonlinear interior point	OPT++, TAO
	CG	Standard nonlinear CG	OPT++, TAO
		Limited memory BFGS	OPT++
		Gradient projection	TAO
	Direct Search	Without derivative information	OPT++
	Semismooth	Infeasible semismooth	TAO
		Feasible semismooth	

ACTS Tools: *numerical functionalities*

Computational Problem	Methodology	Algorithms	Library
ODEs	Integration	Variable coefficient Adams-Moulton	CVODE (SUNDIALS)
	Backward differential	Direct and iterative solvers	
ODEs with sensitivity analysis	Integration	Variable coefficient Adams-Moulton	
	Backward differential	Direct and iterative solvers	
Differential-algebraic equations	Backward differential formula	Direct and iterative solvers	IDA (SUNDIALS)
Nonlinear equations with sensitivity analysis	Inexact Newton	line search	SensKINSOL (SUNDIALS)
Tuning and optimization	Automatic code generator	BLAS and some LAPACK routines	ATLAS

Tool 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 )
```

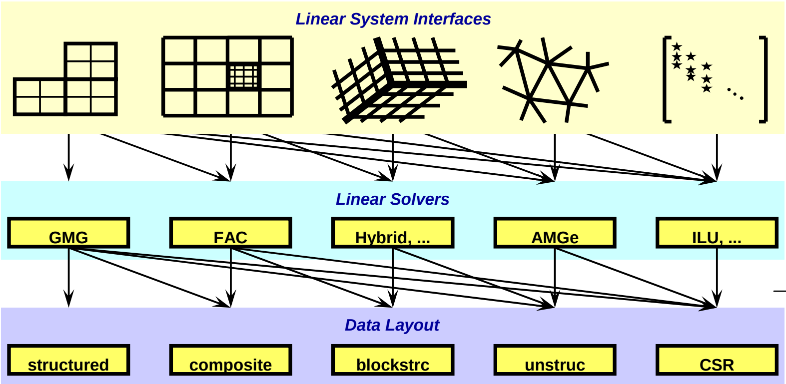
function call
(ScaLAPACK)

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

More advanced:

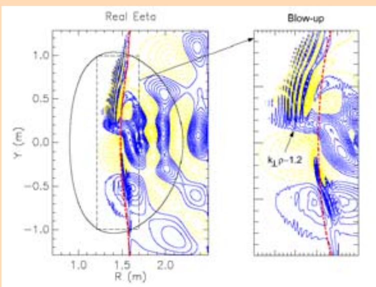
- **-ksp_max_it <max_iters>**
- **-ksp_gmres_restart <restart>**
- **-pc_asm_overlap <overlap>**
- **-pc_asm_type [basic,restrict,interpolate,none]**

command line
(PETSc)

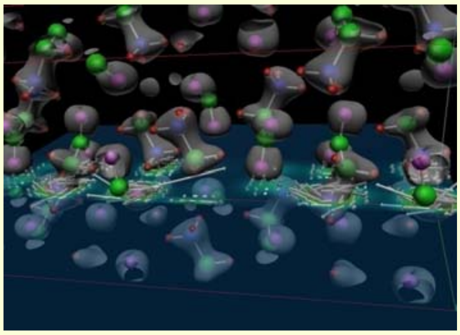


problem domain
(Hypre)

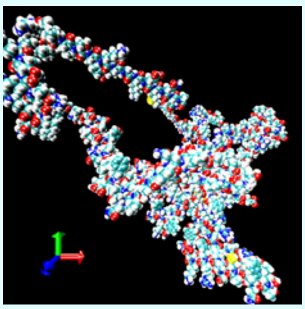
ACTS Tools: *examples of use*



*Two **ScaLAPACK** routines, PZGETRF and PZGETRS, are used for solution of linear systems in the spectral algorithms based AORSA code (Batchelor et al.), which is intended for the study of electromagnetic wave-plasma interactions. The code reaches 68% of peak performance on 1936 processors of an IBM SP.*



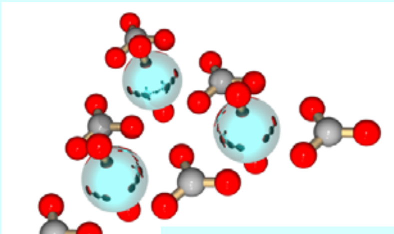
*Induced current (white arrows) and charge density (colored plane and gray surface) in crystallized glycine due to an external field (Louie, Yoon, Pfrommer and Canning), eigenvalue problems solved with **ScaLAPACK**.*



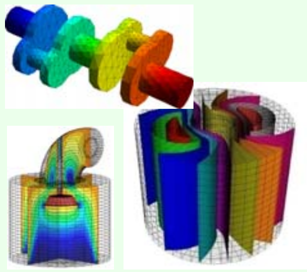
***OPT++** is used in protein energy minimization problems (shown here is protein T162 from CASP5, courtesy of Meza , Oliva et al.)*

*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. Finding 10 eigenvalues requires about 2.5 hours on 24 processors of an IBM SP.*

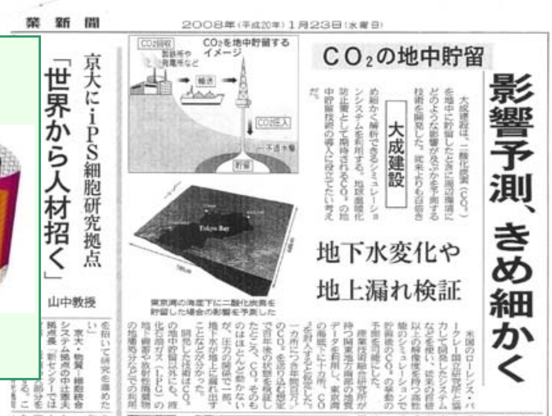
ACTS Tools: *examples of use*



*Electronic structure optimization performed with **TAO**, $(UO_2)_3(CO_3)_6$ (courtesy of deJong).*



*Problems (different grid types) solved with **Hypre**.*



CO₂の地中貯留

影響予測、きめ細かく

大成建設

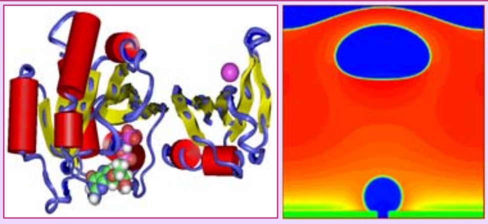
地下水変化や地上漏れ検証

京大にiPS細胞研究拠点

「世界から人材招く」

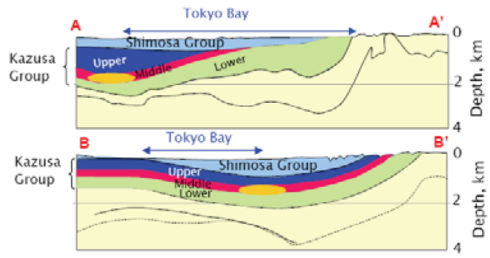
無重量下の液体再現

東大、模擬実験モデル開発

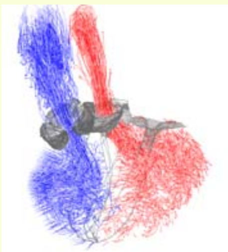


*Molecular dynamics and thermal flow simulation using codes based on **Global Arrays**. GA* have been employed in large *simulation codes such as NWChem, GAMESS-UK, Columbus, Molpro, Molcas, MWPhys/Grid, etc.*

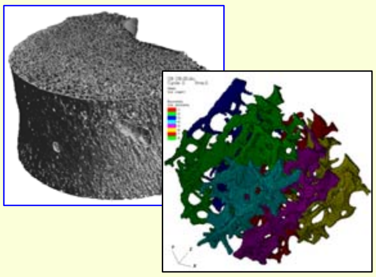
*TOUGH2, a code for multi-phase fluid and heat flow in porous and fractured media uses **Aztec** for solving systems of linear-equation (http://www.tough2.com)*



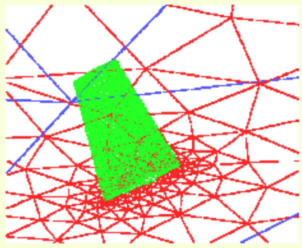
ACTS Tools: *examples of use*



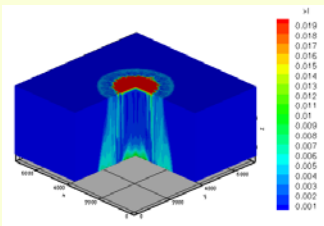
*Model of the heart mechanics (blood-muscle-valve) by an adaptive and parallel version of the immersed boundary method, using **PETSc**, **Hypre** and SAMRAI (courtesy of Boyce Griffith, New York University).*



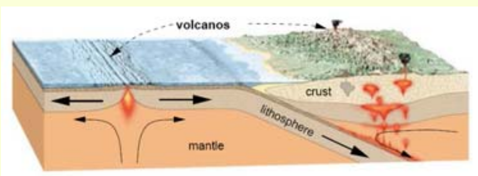
*Micro-FE bone modeling using ParMetis, Prometheus and **PETSc**; models up to 537 million DOF (Adams, Bayraktar, Keaveny, and Papadopoulos).*



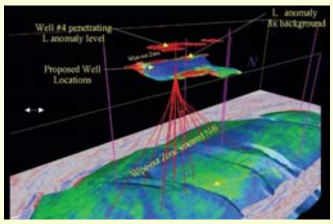
*3D incompressible Euler,tetrahedral grid, up to 11 million unknowns, based on a legacy NASA code, FUN3d (W. K. Anderson), fully implicit steady-state, parallelized with **PETSc** (courtesy of Kaushik and Keyes).*



*PFLOTRAN, a 3-D reservoir simulator based on **PETSc** (https://software.lanl.gov/pflotran)*



*Modeling magma dynamics using **PETSc** and Semi-Lagrangian Advection Schemes (R. Katz, Columbia University).*



*Improving depth imaging using **PETSc** for the inversion of sensitivities of traveltimes calculated with a gridded ray tracing (J. Buur and T. Kuehnel).*

Important Questions for Application Developers

- *How does performance vary with different compilers?*
- *Is poor performance correlated with certain OS features?*
- *Has a recent change caused unanticipated performance?*
- *How does performance vary with MPI variants?*
- *Why is one application version faster than another?*
- *What is the reason for the observed scaling behavior?*
- *Did two runs exhibit similar performance?*
- *How are performance data related to application events?*
- *Which machines will run my code the fastest and why?*
- *Which benchmarks predict my code performance best?*

From http://acts.nersc.gov/events/Workshop2005/slides/Shende.pdf

TAU

(**T**uning and **A**nalysis **U**tilities)

- Multi-level performance instrumentation
- Multi-language automatic source instrumentation
- Flexible and configurable performance measurement
- Widely-ported parallel performance profiling system
  - Computer system architectures and operating systems
  - Different programming languages and compilers
- Support for multiple parallel programming paradigms

- *Profiling*: recording of summary information during execution
- *Tracing*: recording of information about significant points (events) during

TAU: *Example* (1/2)

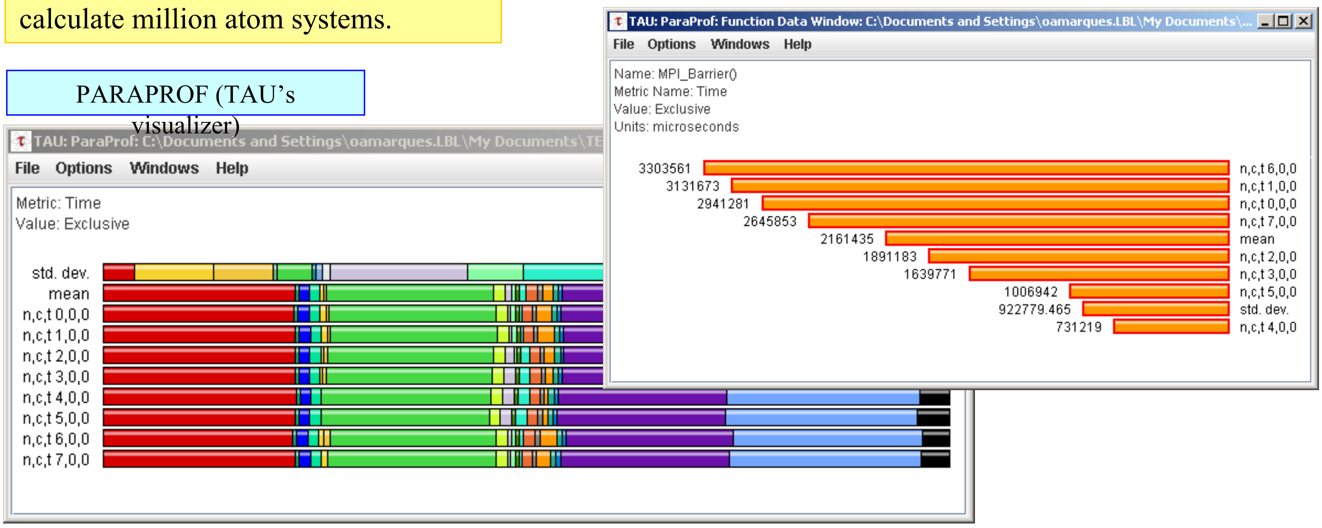
PESCAN is a code that uses the folded spectrum method for nonselfconsistent nanoscale calculations. It uses a planewave basis, and conventional Kleinman-Bylander nonlocal pseudopotetials in real space. It is parallelized using MPI and can calculate million atom systems.

```
# Makefile for PESCAN

include $(TAULIBDIR)/Makefile.tau-multiplecounters-mpi-papi-pdt

FC = $(TAU_COMPILER) mpxlf90_r
CC = $(TAU_COMPILER) mpcc_r

      :
```

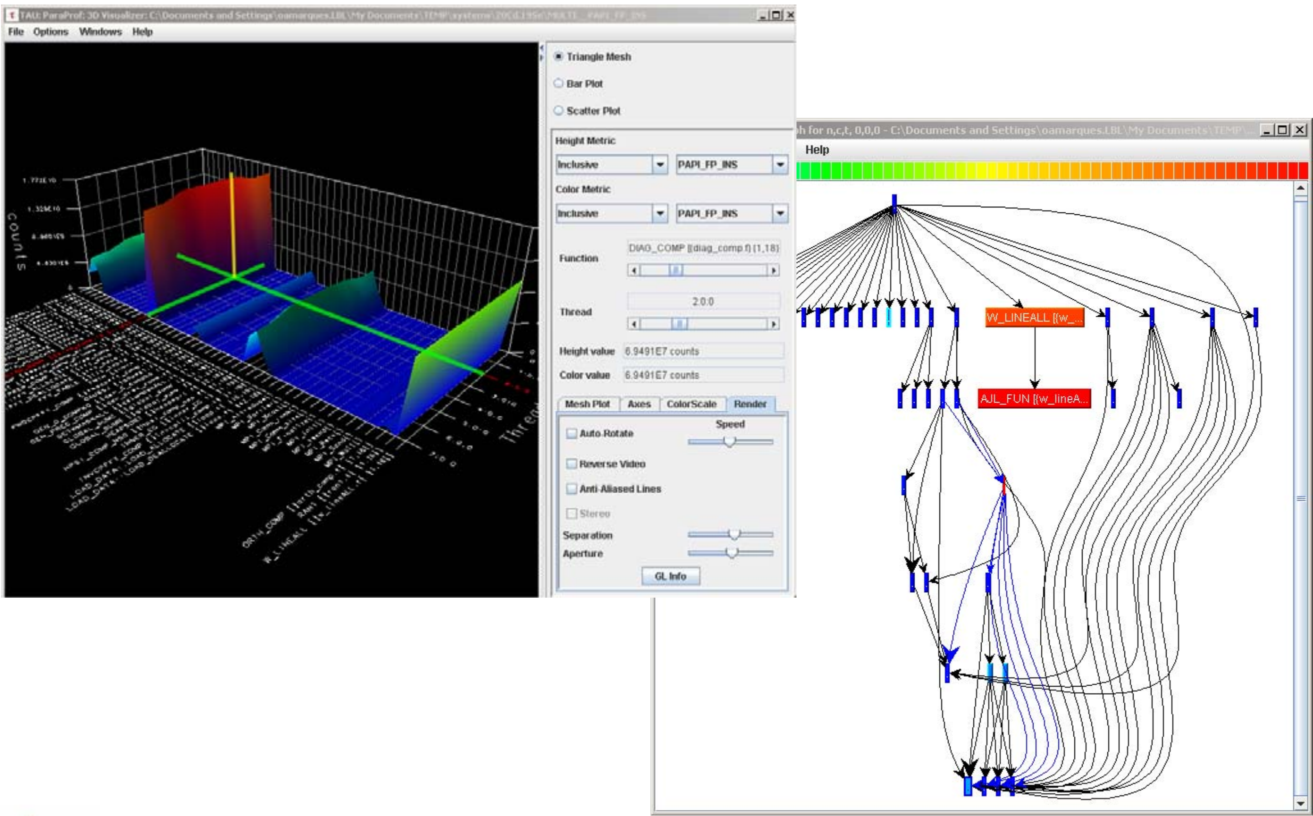
sets features to be instrumented

performs automatic instrumentation

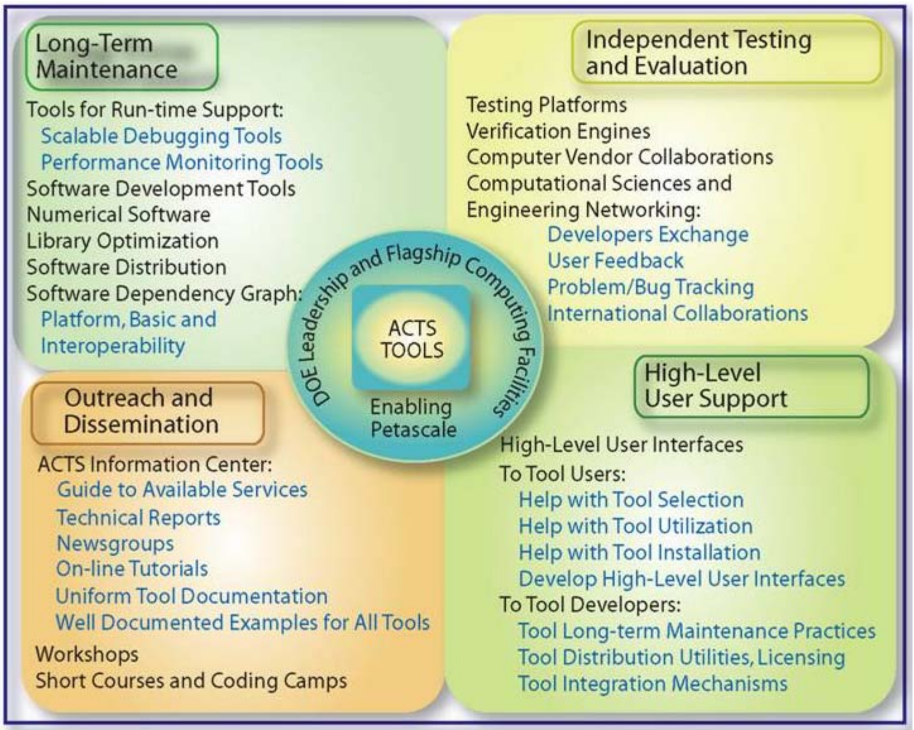
PARAPROF (TAU's visualizer)



TAU: *Example* (2/2)



ACTS Center: *set of services*

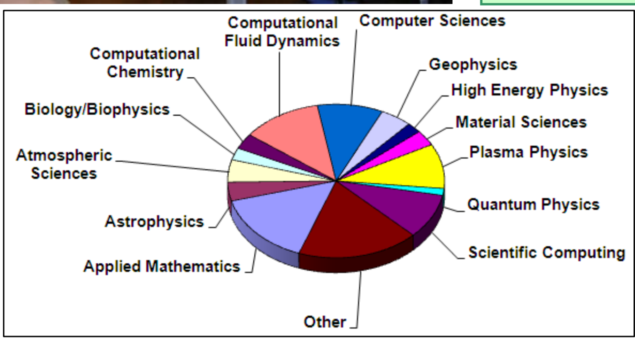


acts-support@nersc.gov - http://acts.nersc.gov

The ACTS Project: *outreach efforts*

Yearly Workshop at LBNL

- ***Ninth Workshop*, August 19-22, 2008**
- Previous (eight) workshops (and other events) attended by more than 350 participants from DOE labs, academia and industry
- Tutorials and hands-on (on NERSC computers) delivered by tool developers
- Help with start up accounts (on NERSC computers)



Participants in the ACTS Collection Workshops: 53% of students, 15% postdoctoral fellows, 22% researchers and faculty, and 9% from industry, supercomputer centers, computer vendors, ISVs, and the military. *Other* includes electromagnetism, numerical relativity, computational social sciences, computational neuroscience, medical imaging, hydrology, etc.

Recent Short Courses and Minisimposia

- ***Workshop and Advanced School on Eigenvalue Problems, Software and Applications***, Porto, Portugal, 2007.
- ***ACTS Collection Workshop at SDSC***, San Diego, CA, 2006.
- ***Tutorial on Robust and High Performance Software Libraries for Computational Sciences***, VECPAR'06, Rio de Janeiro, Brazil.
- ***Short Course on the ACTS Collection***, SIAM CSE05 Conference, Orlando, FL.

Productivity: *software development timeline*

min[*time_to_first_solution*] (prototype)

min[*time_to_solution*] (production)

- Outlive complexity
  - *Increasingly sophisticated models*
  - *Model coupling*
  - *Interdisciplinary*
- Sustained performance
  - *Increasingly complex algorithms*
  - *Increasingly diverse architectures*
  - *Increasingly demanding applications*

min[*software-development-cost*] and
max[*software_lifespan*] and
max[*resource_utilization*]

$$\psi = \frac{U(T)}{C_s + C_o + C_m}$$

- Ψ = overall system productivity
- *U(T)* = utility (a function of time); the longer the time to solution, the lower the utility of that solution
- C_S = cost of software
- C_O = cost of operator
- C_M = cost of machine

Funk, Basili, Hochstein and Kepner, CTWatch, November 2006 A.

Additional Information

- *An Overview of the Advanced CompuTational Software* (*ACTS*) *Collection*, L.A. Drummond and O. Marques, ACM Transactions on Mathematical Software, 31:282-301, 2005.
- *The ACTS Collection, Robust and High-Performance Tools for Scientific Computing: Guidelines for Tool Inclusion and Retirement*, L.A. Drummond and O. Marques,Technical Report LBNL/PUB-3175, 2002.

September 2005 Issue

Summer 2006 Issue