The ACTS Collection:
Functionality and Applications
MS48
SIAM CSE-05 - Orlando, Florida
February 14, 2005

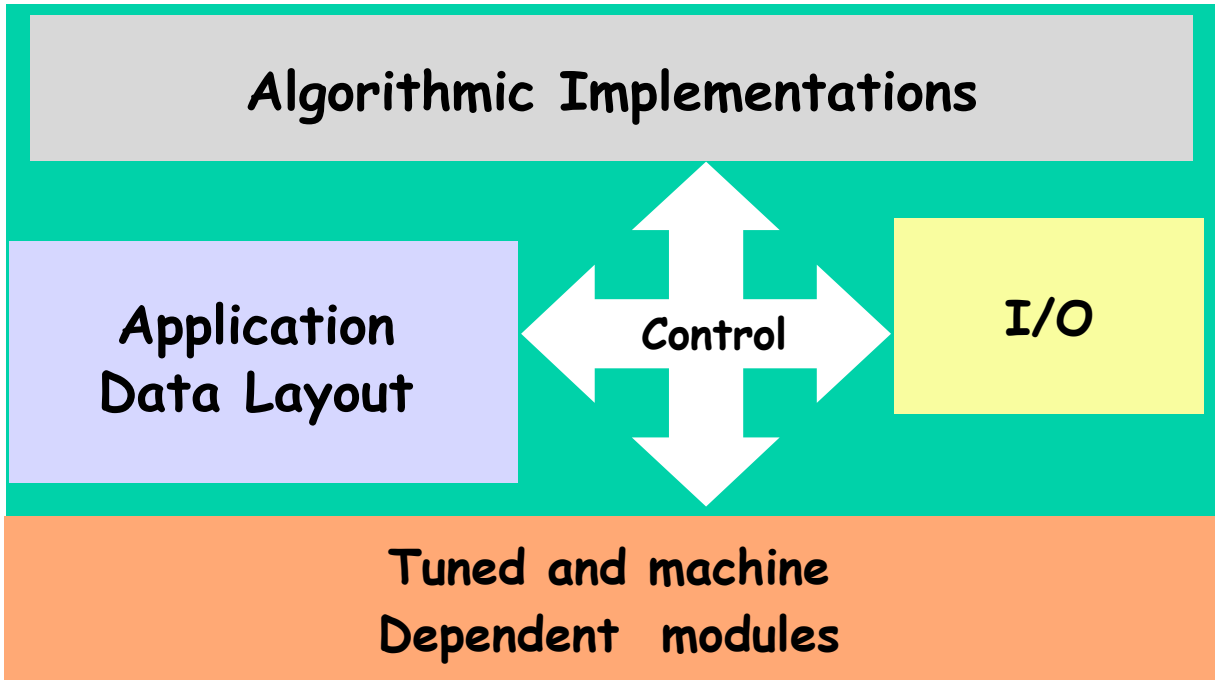
Tony Drummond
Lawrence Berkeley National Laboratory
acts-support@nersc.gov
ladrummond@lbl.gov

Berkeley Lab

Office of Science
U.S. Department of Energy

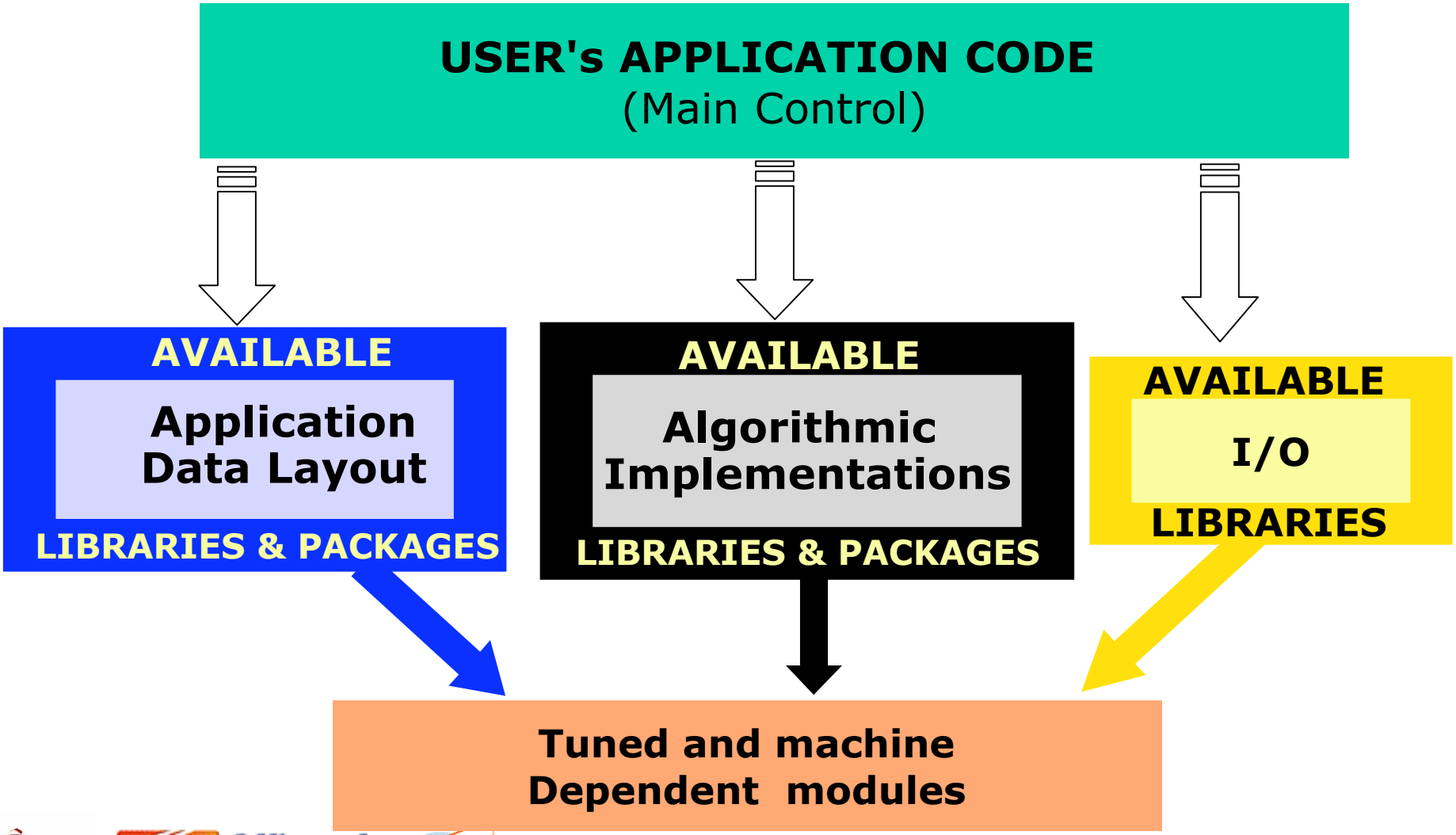
Motivation- Why do we need software libraries?

Large Scientific Codes:
A Common Programming Practice



Motivation- Why do we need software libraries?

An Alternative Approach



Good Reasons to Use Libraries

- **Productivity**
  - Time to the first solution (prototype)
  - Time to solution (production)
  - Long term and upgrades
- **Complexity**
  - Increasingly sophisticated models
  - Model coupling
  - Interdisciplinary Nature
- **Performance**
  - Increasingly complex algorithms
  - Increasingly complex architectures
  - Increasingly demanding applications

The 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

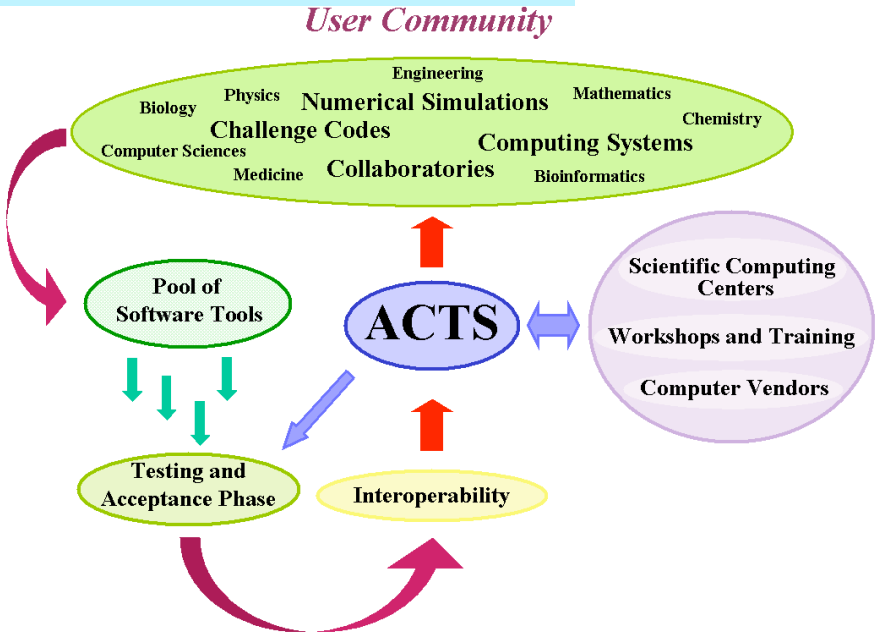
- *Extended support for experimental software*
- *Make ACTS tools available on HPC centers*
- *Provide technical support (acts-support@nersc.gov)*
- *Maintain ACTS information center (http://acts.nersc.gov)*
- *Coordinate efforts with other supercomputing centers*
- *Enable large scale scientific applications*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

- ***High***
  - Intermediate level
  - Tool expertise
  - Conduct tutorials
- ***Intermediate***
  - Basic level
  - Higher level of support to users of the tool
- ***Basic***
  - Help with installation
  - Basic knowledge of the tools
  - Compilation of user's reports

Why is ACTS unique?

- Provides pointers and documentation about software tools.
- Accumulates the expertise and user feedback on the use of the software tools and scientific applications that used them:
  - independent software evaluations
  - participation in the developer user groups e-mail list
  - leverage between tool developers and tool users
  - presentation of a gallery of applications
  - workshops and tutorials
  - tool classification
  - support

6th ACTS Collection Workshop, Berkeley
Third Week August!



ACTS Tools http://acts.nersc.gov/tools

Category	Tool	Functionalities
Numerical $Ax = b$ $Az = \lambda z$ $A = U\Sigma V^T$ PDEs ODEs M	**Aztec00 - Trilinos**	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.
	SLEPc	Sparse general eigenvalue solvers
	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.
	PAWS	Framework for coupling parallel applications within a component-like model.
	SILOON	Tools and run-time support for building easy-to-use external interfaces to existing numerical codes.
	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 Selection

```
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 )
```

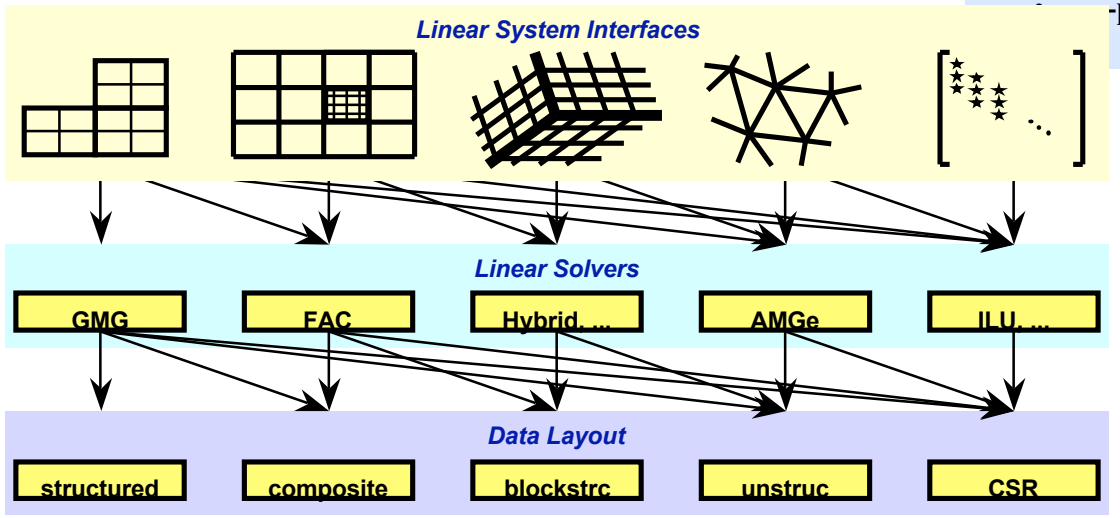
Language Calls

Command lines

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

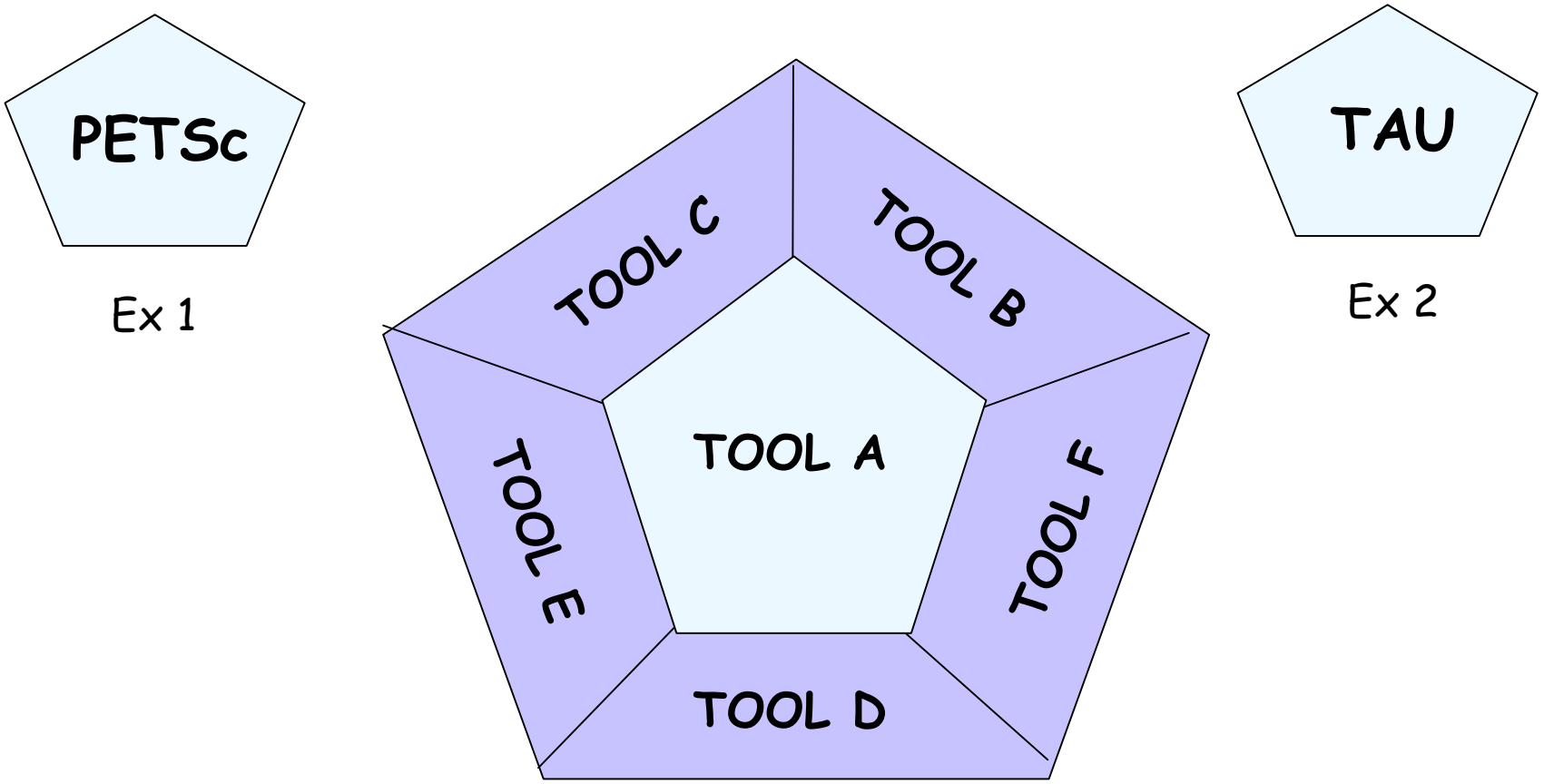
More advanced:

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

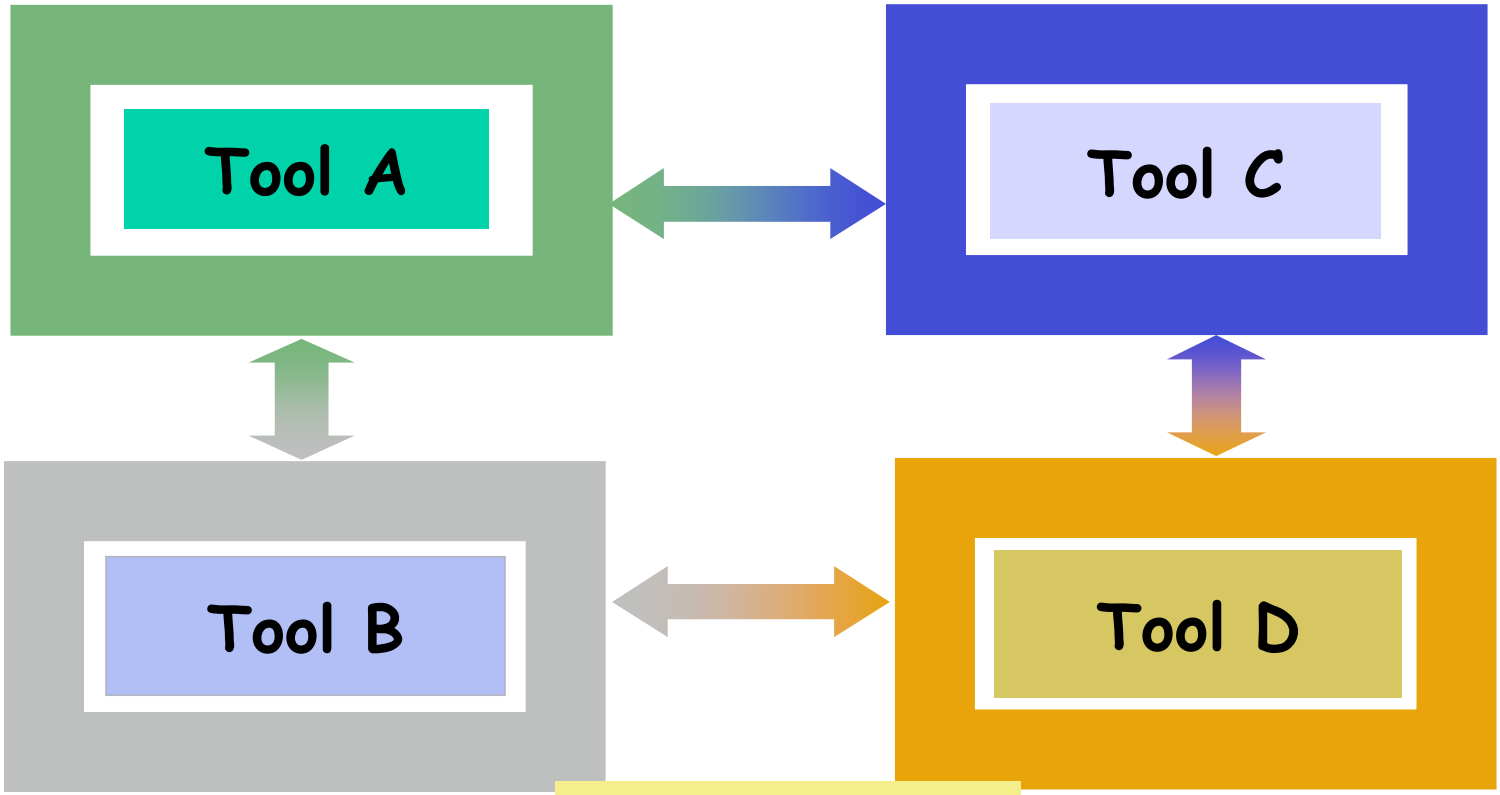


Problem Domain

Tool Interoperability
Tool-to-Tool



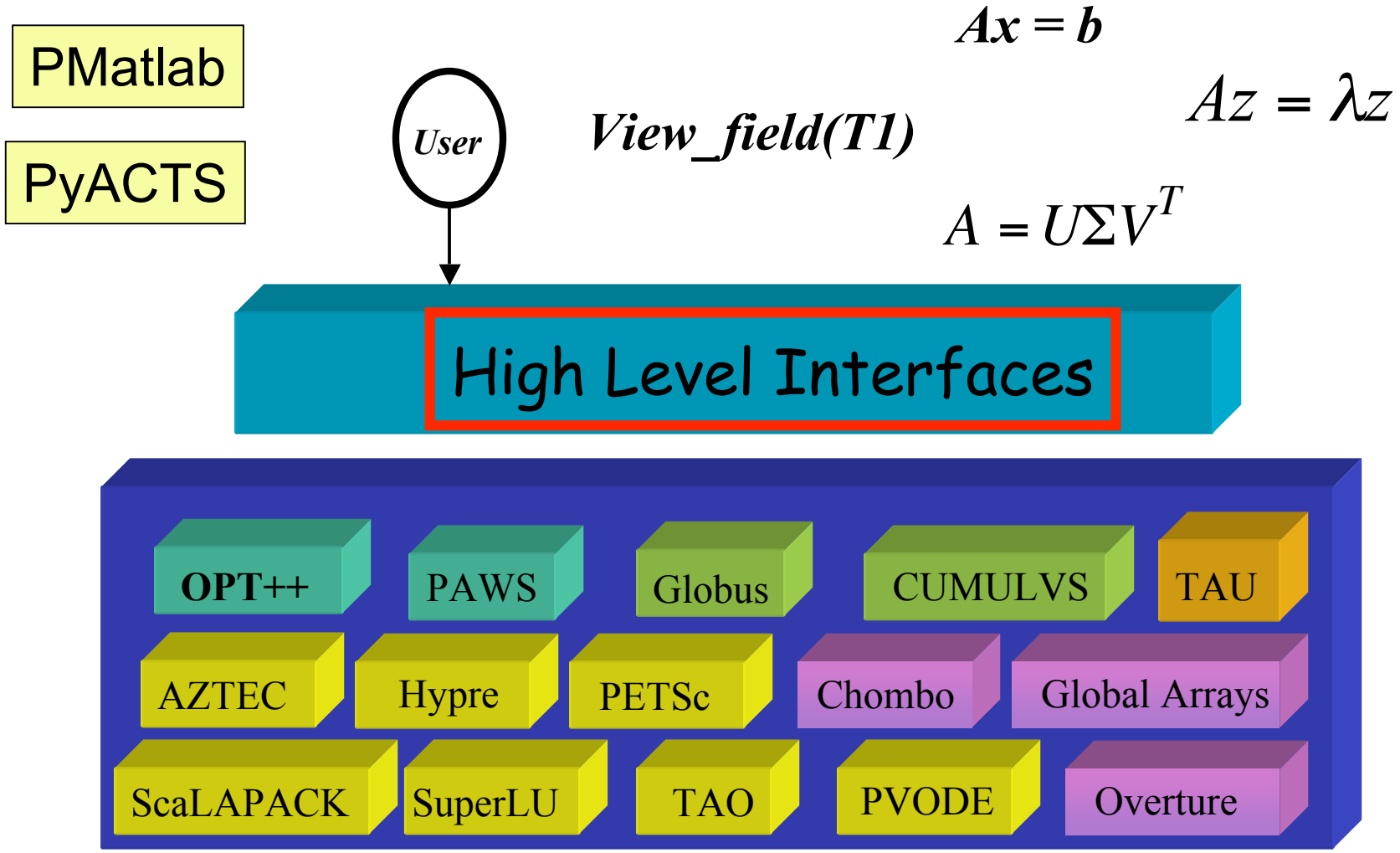
Component Technology!



CCA

CompFrame05
Atlanta, GA
June, 2005

PSE's and Frameworks



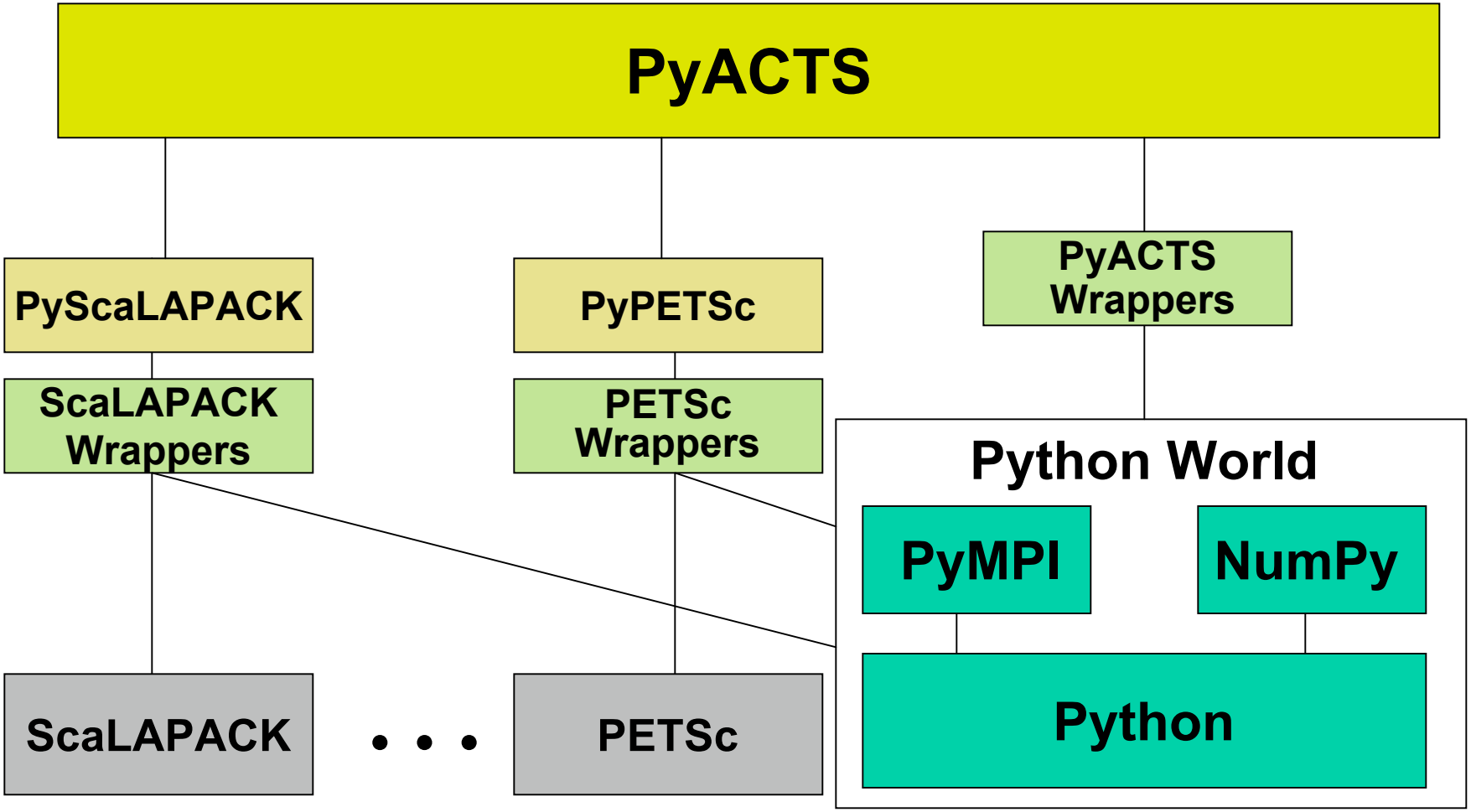
Motivation and Design Consideration

- High-level user friendly interface
- Hides details of parallelism from users
- Teaches users how to use the tools
- Flexible parameter reconfiguration
- Interoperability
- Choice of language: Python

Motivation and Design Consideration

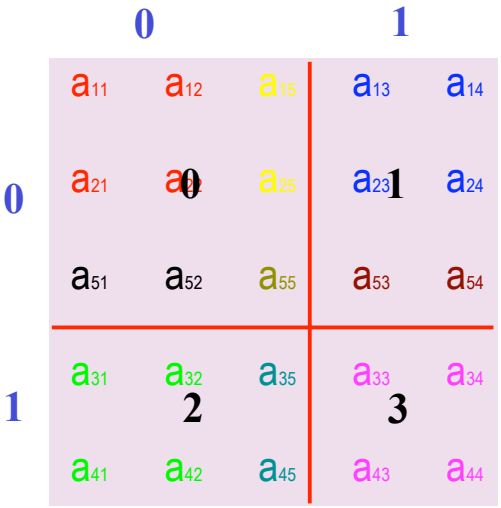
- Choice of scripting language: Python
- Uses PyMPI and Numeric
- Intended for testing and not high-performing production runs.

A Conceptual View Of PyACTS



2D Block-Cyclic Distribution

$$\begin{bmatrix} 1.1 & 1.2 & 1.3 & 1.4 & 1.5 \\ -2.1 & 2.2 & 2.3 & 2.4 & 2.5 \\ -3.1 & -3.2 & 3.3 & 3.4 & 3.5 \\ -4.1 & -4.2 & -4.3 & 4.4 & 4.5 \\ -5.1 & -5.2 & -5.3 & -5.4 & 5.5 \end{bmatrix}$$

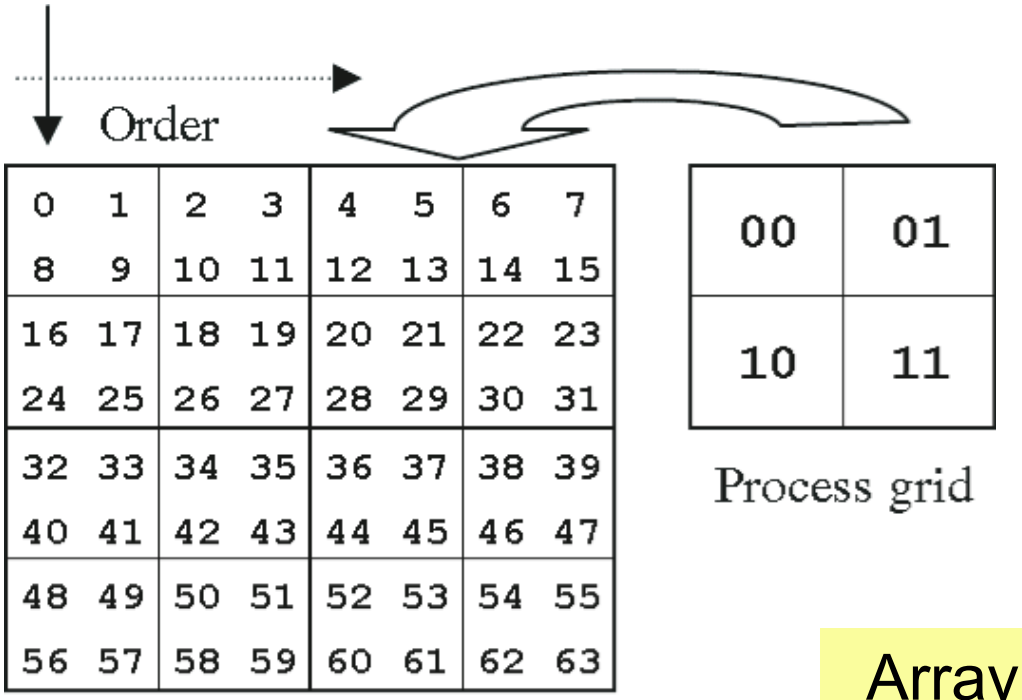



```
CALL BLACS_GRIDINFO( ICTXT, NPROW, NPCOL, MYROW, MYCOL )

IF      ( MYROW.EQ.0 .AND. MYCOL.EQ.0 ) THEN
        A(1) = 1.1; A(2) = -2.1; A(3) = -5.1;
        A(1+LDA) = 1.2; A(2+LDA) = 2.2; A(3+LDA) = -5.2;
        A(1+2*LDA) = 1.5; A(2+3*LDA) = 2.5; A(3+4*LDA) = -5.5;
ELSE IF ( MYROW.EQ.0 .AND. MYCOL.EQ.1 ) THEN
        A(1) = 1.3; A(2) = 2.3; A(3) = -5.3;
        A(1+LDA) = 1.4; A(2+LDA) = 2.4; A(3+LDA) = -5.4;
ELSE IF ( MYROW.EQ.1 .AND. MYCOL.EQ.0 ) THEN
        A(1) = -3.1; A(2) = -4.1;
        A(1+LDA) = -3.2; A(2+LDA) = -4.2;
        A(1+2*LDA) = 3.5; A(2+3*LDA) = 4.5;
ELSE IF ( MYROW.EQ.1 .AND. MYCOL.EQ.1 ) THEN
        A(1) = 3.3; A(2) = -4.3;
        A(1+LDA) = 3.4; A(2+LDA) = 4.4;
END IF

CALL PDGESVD( JOBU, JOBVT, M, N, A, IA, JA, DESCA, S, U, IU,
              JU, DESCU, VT, IVT, JVT, DESCVT, WORK, LWORK,
              INFO )
```

LDA is the leading dimension of the local array (see next slides)

Array descriptor for A (see next slides)

PyBLACS Example Operation



Array Distribution and Processor Layout

PyBLACS Example Operation

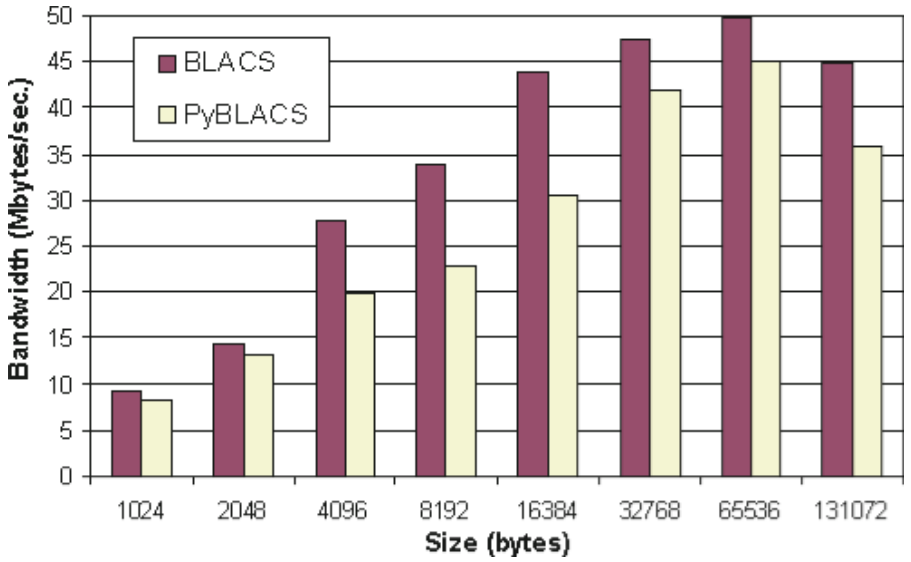
```
> import PyACTS
> import Numeric
> # HERE USER HAS READ DATA ARRAY FROM FILE (DIM N)
> PyACTS.gridinit(nb=2)
> ACTS_lib=1 # ScaLAPACK library
> if PyACTS.iread==1:
> a=Numeric.reshape(range(n*n),[n,n])
> else:
> a=None
> a=PyACTS.Num2PyACTS(a,ACTS_lib) # convert array to PyACTS array
> print "PyACTS Array Properties in
      [",PyACTS.myrow,",",PyACTS.mycol,"]"
> print " lib=",a.lib
> print " desc=",a.desc
> print " data=",a.data
> PyACTS.gridexit().
```

PyBLACS operation

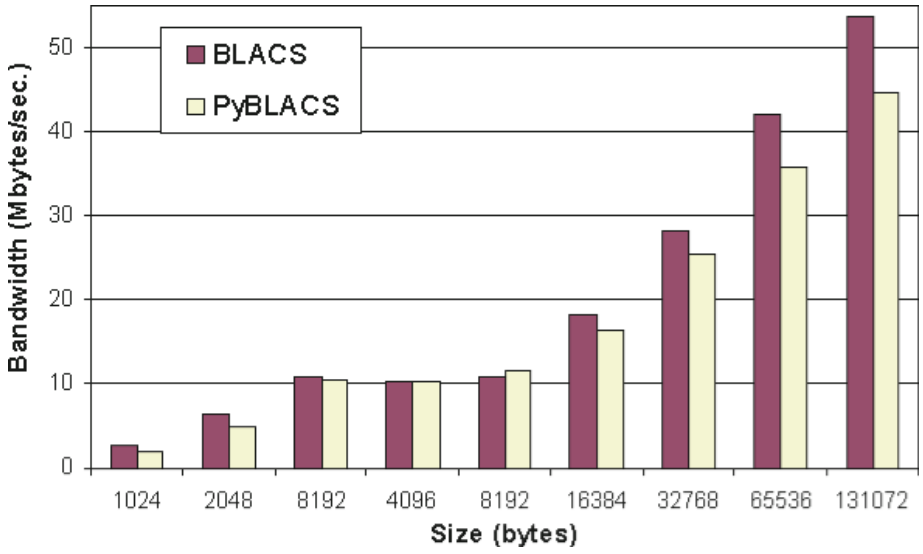
```
PyACTS Array Properties in [ 0 , 0 ]
       lib= 1; desc= [1 0 8 8 2 2 0 0 4]
       data= [ 0  8 32 40  1  9 33 41
               4 12 36 44  5 13 37 45]
PyACTS Array Properties in [ 1 , 0 ]
       lib= 1;desc= [1 0 8 8 2 2 0 0 4]
       data= [16 24 48 56 17 25 49 57
              20 28 52 60 21 29 53 61]
PyACTS Array Properties in [ 1 , 1 ]
       lib= 1; desc= [1 0 8 8 2 2 0 0 4]
       data= [18 26 50 58 19 27 51 59
              22 30 54 62 23 31 55 63]
PyACTS Array Properties in [ 0 , 1 ]
       lib= 1; desc= [1 0 8 8 2 2 0 0 4]
       data= [ 2 10 34 42  3 11 35 43
               6 14 38 46  7 15 39 47]
```

Output from code

BLACS vs PyBLACS



Linux Cluster (2Ghz)



IBM SP - PWR 3

Example of PBLAS: pvgemm

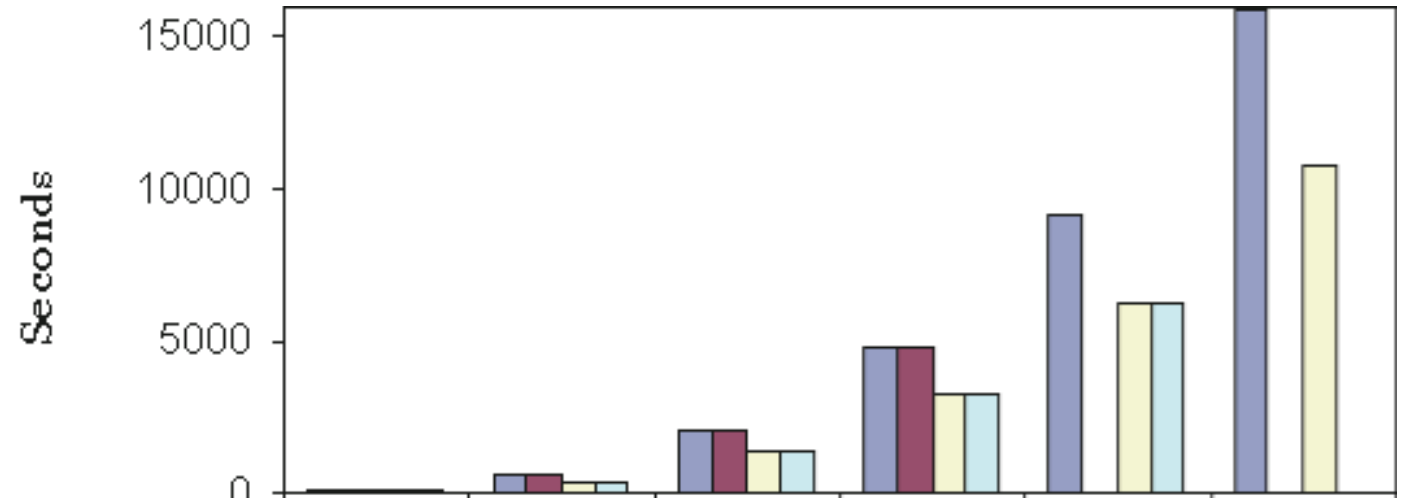
```
PVGEMM( TRANSA, TRANSB, M, N, K, ALPHA,
        A, IA, JA, DESCA,
        B, IB, JB, DESCB,
        BETA, C, IC, JC, DESCC )
```

- User needs to know about the parallel environment (data layout)
- User needs to initialize the process grid (BLACS)
- User needs to distribute data arrays
- Know details about the BLAS 3 call

Example of PyPBLAS: pvgemm

```
> from PyACTS import *
> import PyACTS.PyPBLAS as PyPBLAS
> import time
> n=500
> ACTS_lib=1 # ScaLAPACK library
> PyACTS.gridinit() # grid initialization
> alpha=Scal2PyACTS(2,ACTS_lib) # convert scalar
                                # to PyACTS scalar
> beta=Scal2PyACTS(3,ACTS_lib)
> a=Rand2PyACTS(n,n,ACTS_lib) # generate a random
                              # PyACTS array
> b=Rand2PyACTS(n,n,ACTS_lib)
> c=PyPBLAS.pvgemm(alpha,a,b,beta,c) # call level 3
                                     # PBLAS routine
> PyACTS.gridexit()
.
```

Example of PBLAS: pdgemm (cluster)



	2000	4000	6000	8000	10000	12000
PyPBLAS 2x2	76,0	608,3	2038,7	4812,6	9149,1	15890,7
PBLAS 2x2	76,0	610,2	2043,3	4805,9		
PyPBLAS 3x2	51,3	403,1	1360,0	3236,3	6319,4	10826,5
PBLAS 3x2	51,0	404,0	1367,2	3241,1	6310,4	

Matrix size

Future Work

- Work on release and documentation
- Include other tools and data formats
- Scriber function ⇒ high performance codes in C, C++ and Fortran flavors.
- Visit http://acts.nersc.gov for updates

Who Benefits from these tools?

Application	Computational Problem	Software Tools	Highlights
MADCAP	Matrix factorization and triangular solves	ScaLAPACK	• 50% peak performance on an IBM SP • Nearly perfect scalability on 1024, 2048, 3072 and 4096 processors • Fast implementation of numerical algorithms
3-Charged Particles	Solution of large, complex unsymmetric linear systems	SuperLU	• Solves systems of equations of order 8.4 million on 64 processors in 1 hour of wall clock time • 30 GFLOPs
NWChem	Distribute large data arrays, collective operations	Global Arrays and LAPACK	• Very good scaling for large problems

http://acts.nersc.gov/MatApp

Enabling sciences and discoveries… with high performance and scalability...

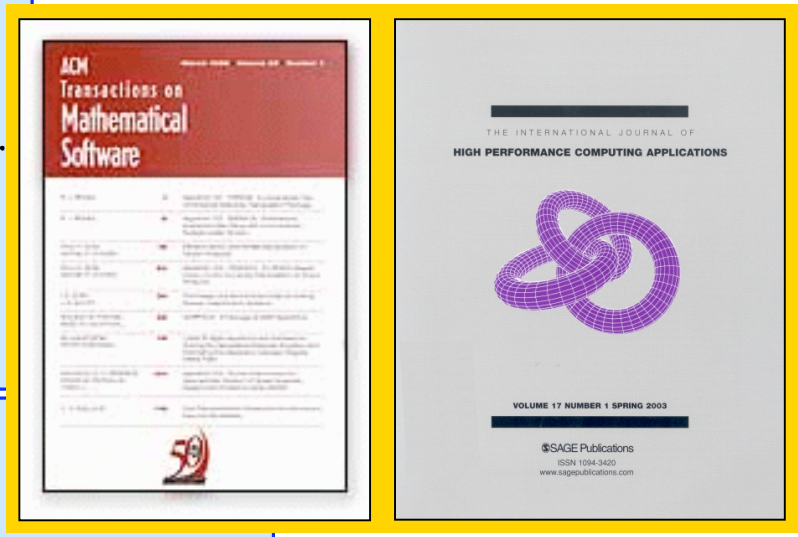
FUN3D &	Unstructured grids, compressible and incompressible Euler and Navier-Stokes equations.	PETSc	• Parallelization of legacy code • Gordon Bell price, 0.23 Tflop/s on 3072 procs of ASCI Red
P-FLAPW	Eigenvalue problems	ScaLAPACK	• Study of systems up to 700 atoms (mat size=35,000) • Runs efficiently • Facilitated the study of new problems in materials science such as impurities and disordered systems.
NIMROD	Quad and triangular high order finite elements, semi-implicit time integration,sparse matrix solvers	SuperLU	• Code improvement of 5 fold, equivalent to 3-5 years progress in computing hardware.

... More Applications …

Study of robust scientific libraries for the advancement of science and engineering
LNCS, *In Proceedings in VECPAR 2004*, Springer Verlag

References

- **An expanded Framework for the Advanced Computational Testing and Simulation Toolkit,** *http://acts.nersc.gov/documents/Proposal.pdf*
- **The Advanced Computational Testing and Simulation (ACTS) Toolkit.** ***Technical Report LBNL-50414.***
- **A First Prototype of PyACTS.** ***Technical Report LBNL-53849.***
- **ACTS - A collection of High Performing Tools for Scientific Computing.** ***Technical Report LBNL-53897.***
- **The ACTS Collection: Robust and high-performance tools for scientific computing. Guidelines for tool inclusion and retirement.** ***Technical Report LBNL/PUB-3175.***
- **An Infrastructure for the creation of High End Scientific and Engineering Software Tools and Applications.** ***Technical Report LBNL/PUB-3176.***



Tutorials and Workshops

- **How Can ACTS Work for you?,** *http://acts.nersc.gov/events/Workshop2000*
- **Solving Problems in Science and Engineering,** *http://acts.nersc.gov/events/Workshop2001*
- **Robust and High Performance Tools for Scientific Computing,** *http://acts.nersc.gov/events/Workshop2002*
- **Robust and High Performance Tools for Scientific Computing,** *http://acts.nersc.gov/events/Workshop2003*
- **The ACTS Collection: Robust and High Performance Libraries for Computational Sciences, SIAM PP04** *http://www.siam.org/meetings/pp04*
- **Enabling Technologies For High End Computer Simulations** *http://acts.nersc.gov/events/Workshop2004*