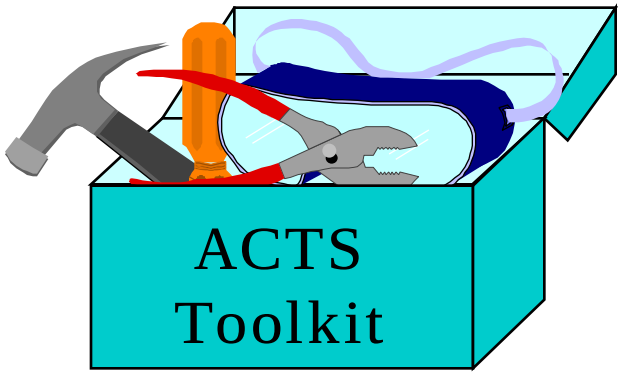
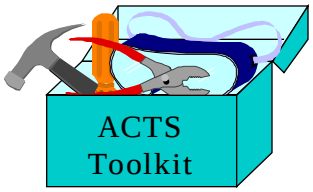
DOE 2000:
The Advanced Computational Testing and Simulation Toolkit



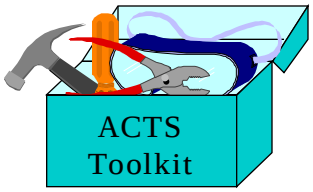
James R. McGraw

Lawrence Livermore National Laboratory



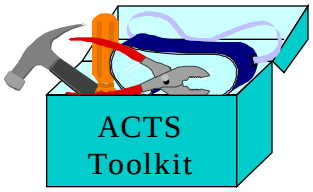
Technical Credit

A talented and flexible group of researchers at seven DOE Labs and five universities.



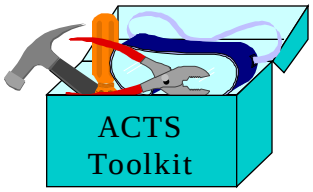
DOE 2000 Vision

Accelerate the ability of DOE to accomplish its mission through advanced computing and collaboration technologies.



DOE 2000 Objectives

- **Improved ability to solve DOE's increasingly complex scientific problems**
  - » **Bring together scientists/engineers from multiple disciplines and organizations**
  - » **Solve problems for which it is impossible to set up experiments**
- **Improved R&D Productivity and Efficiency**
  - » **Share DOE's unique knowledge, facilities, personnel and other resources**
  - » **Provide better problem solving tools:**
- **Increased access to DOE resources by R&D partners**
  - » **Provide access to facilities**
  - » **Ensure access to researchers**



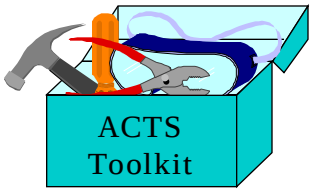
DOE 2000 Components

- **National Collaboratories**

  develop technologies and tools that unite expertise, instruments and computers to enable collaborative work across the DOE system, building ties among disciplines and scientists, speeding research, and coupling discovery and application.

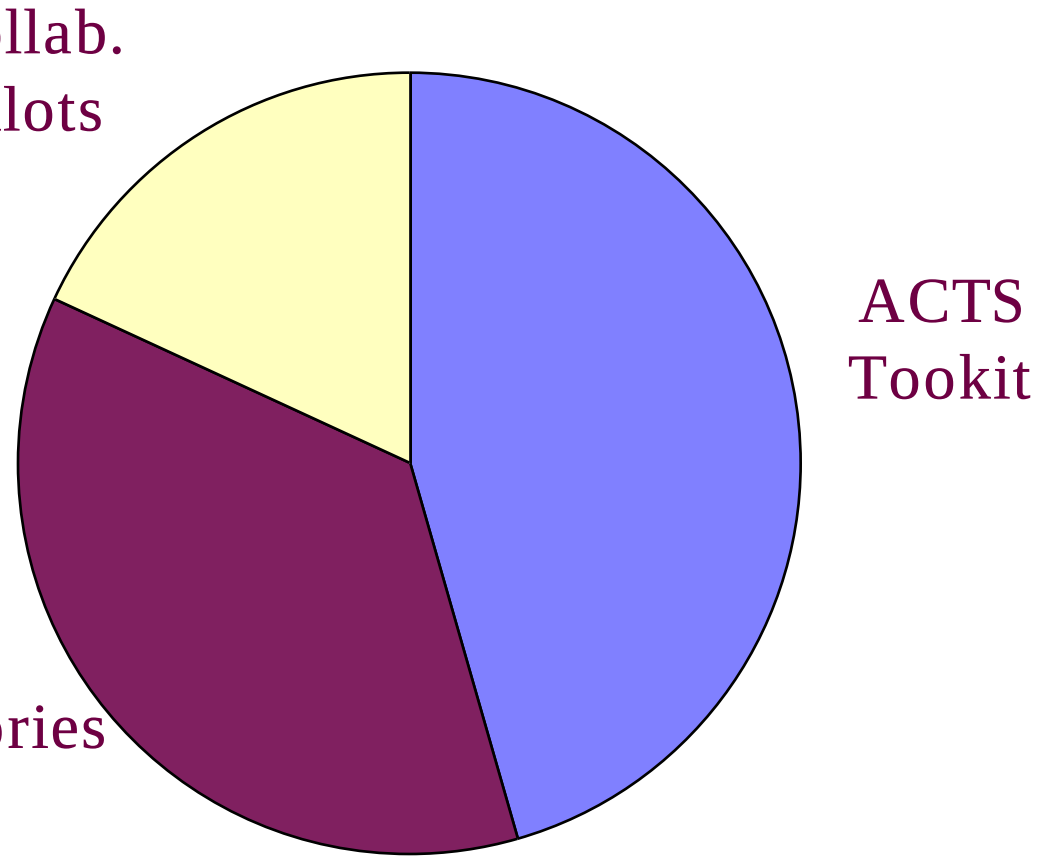
- **Advanced Computational Testing & Simulation (ACTS) Toolkit**

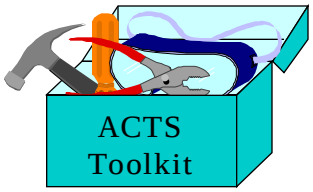
  develop an integrated set of software tools, algorithms and environments that accelerate the adoption and use of advanced computing, enabling DOE programs to thoroughly exploit large scale computing resources to address mission critical problems.



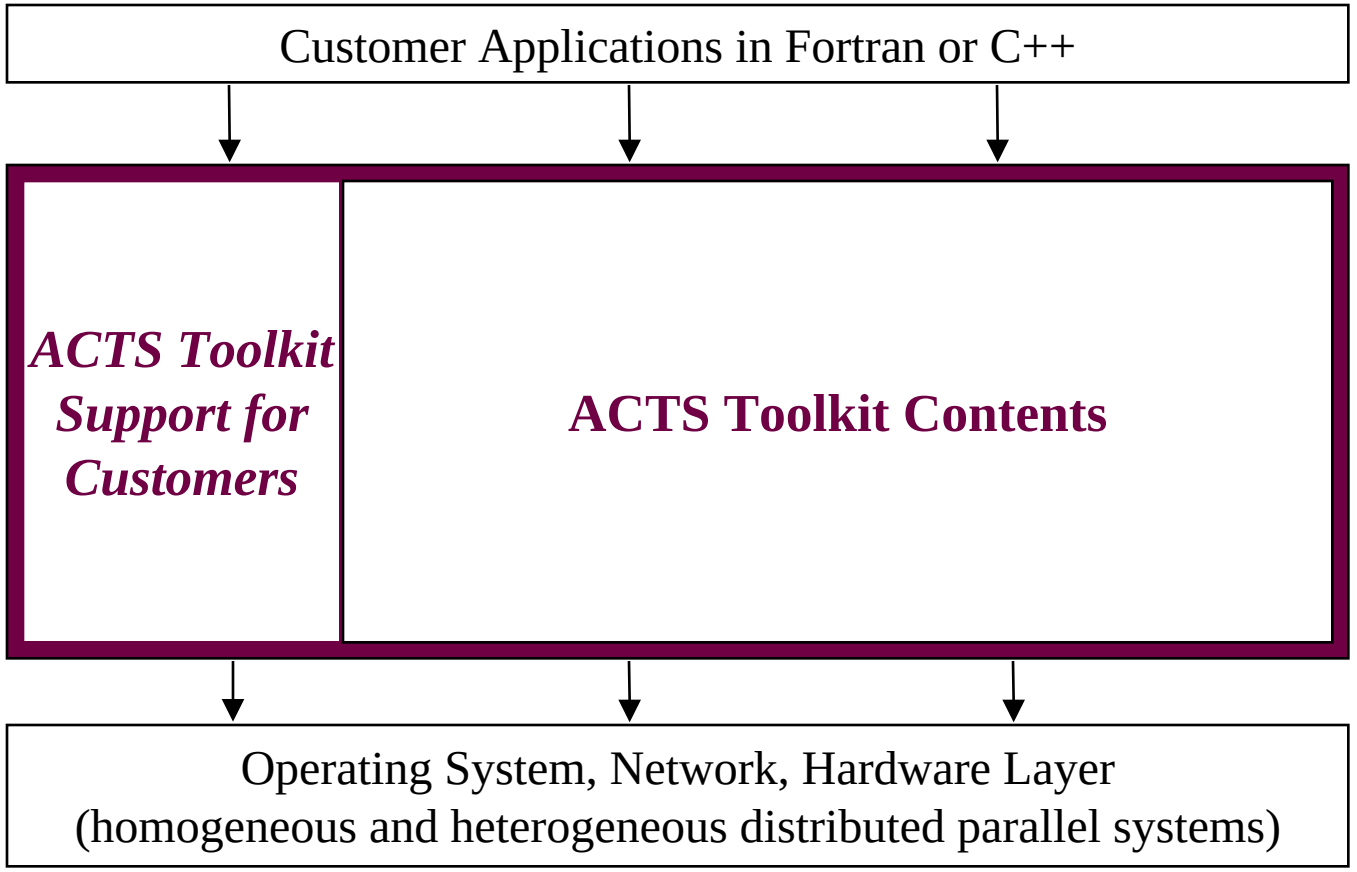
DOE 2000 FY 99 Budget

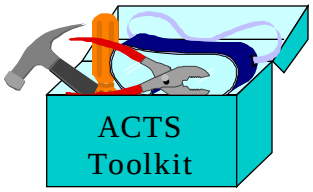
$ 11 M total





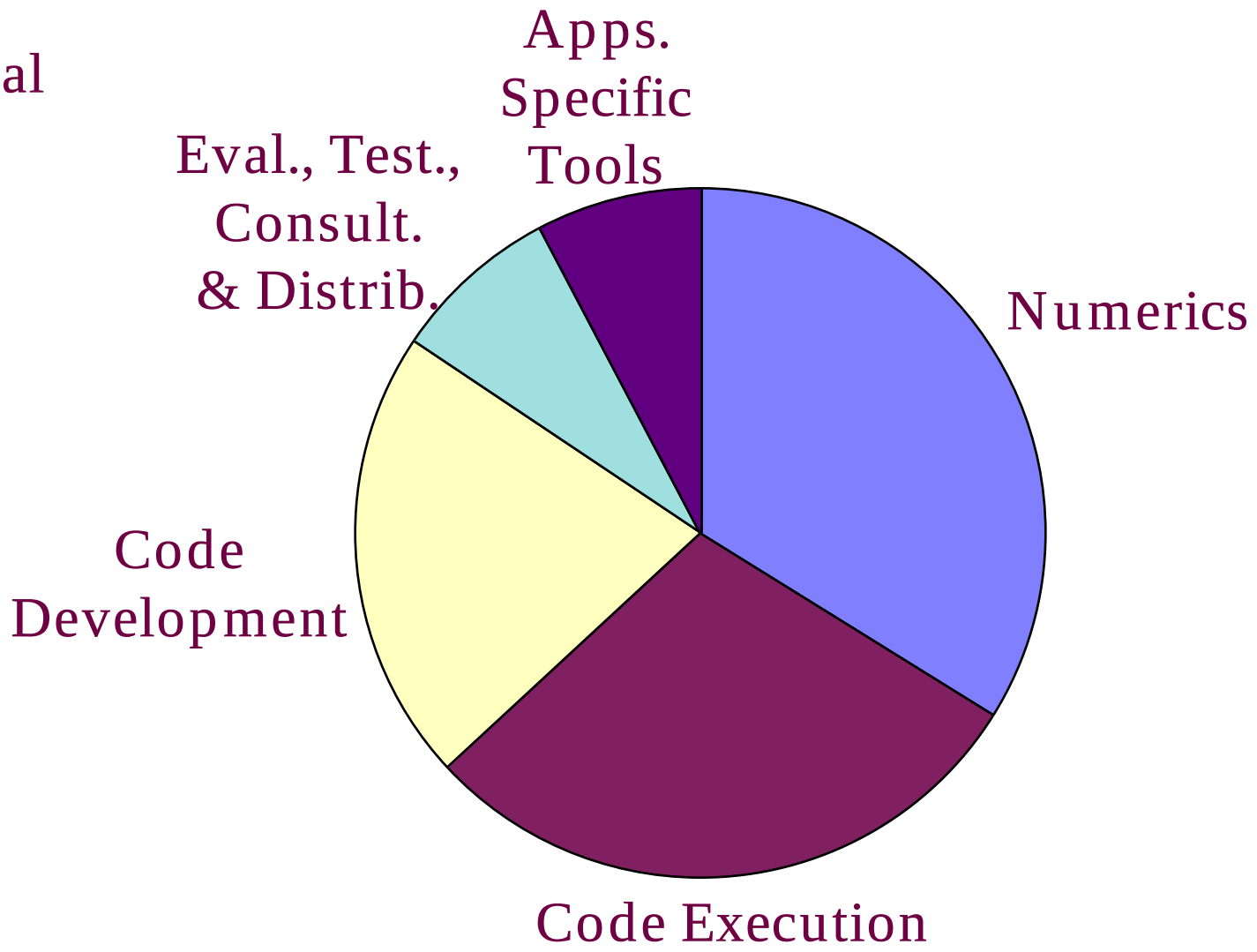
ACTS Toolkit: Context

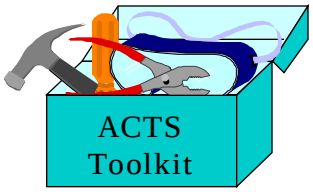




ACTS Toolkit Budget

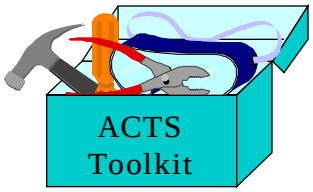
$ 5 M Total





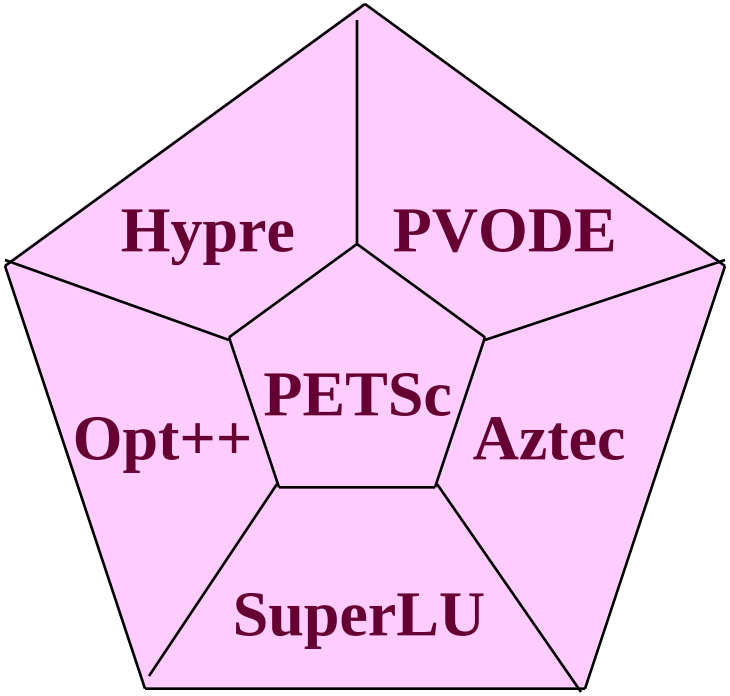
Metrics for Success

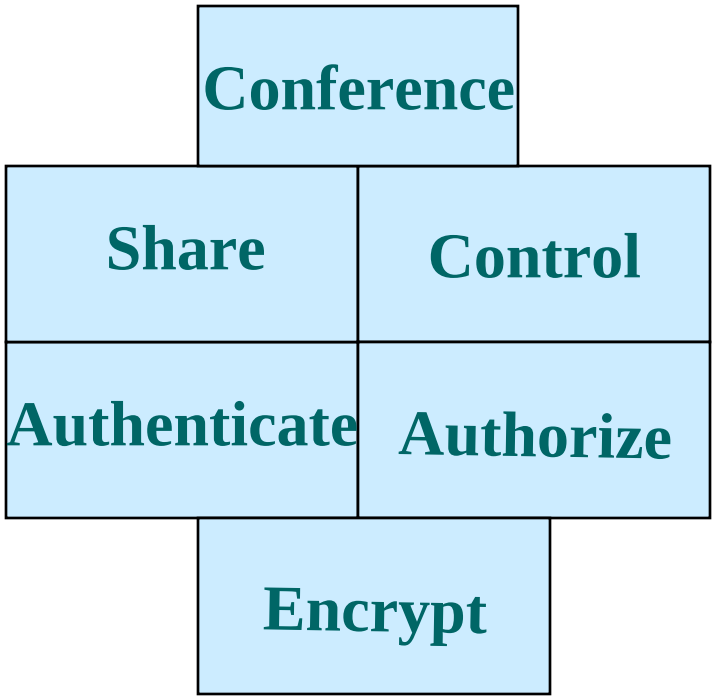
- More clients using the software tools
- Improved performance of tools for clients
- Reduced duplication of effort in tool development
- Providing new capabilities
- Publications

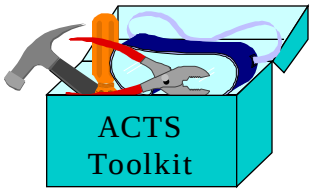


Overarching Theme:

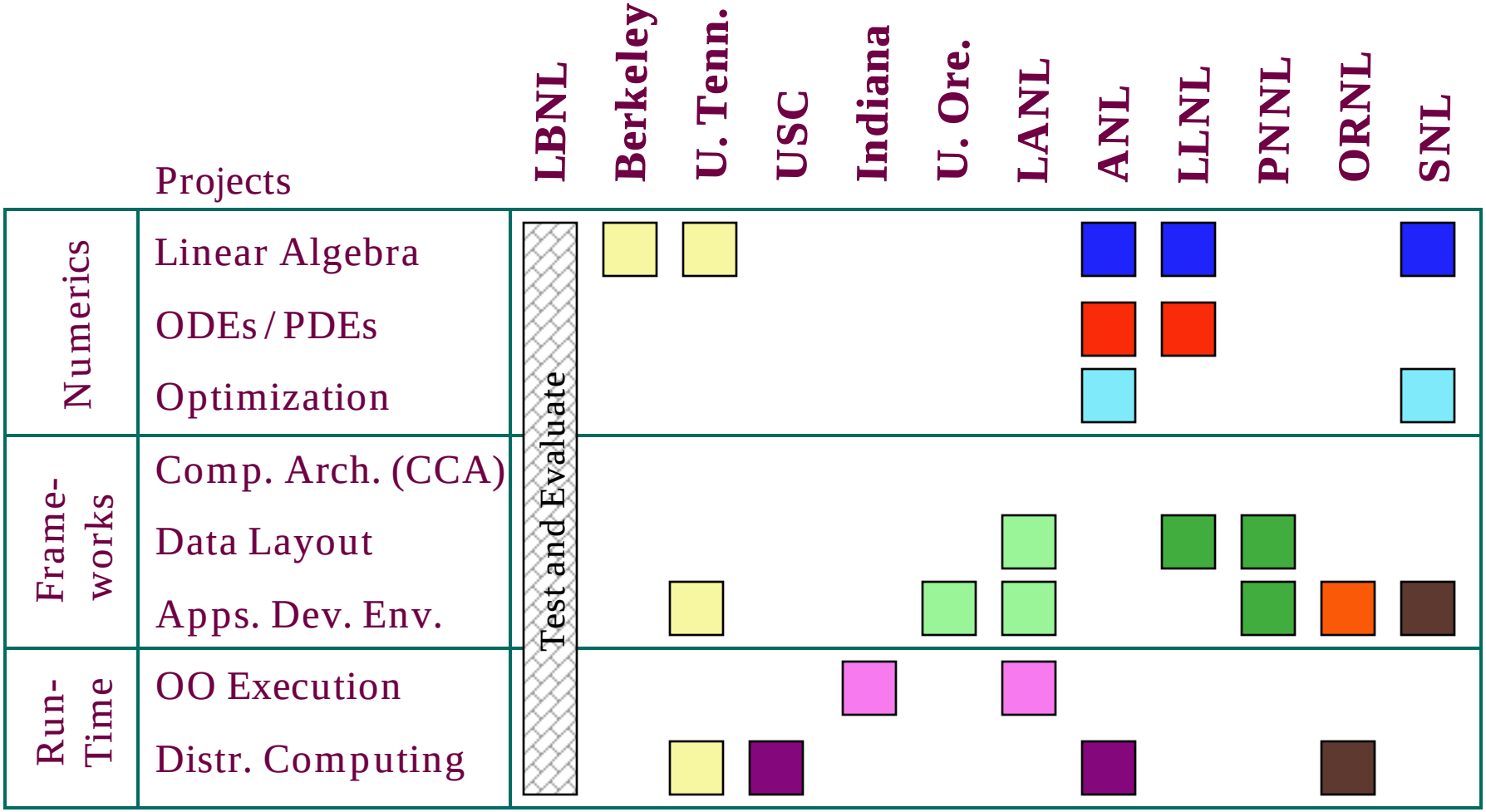
Interoperability

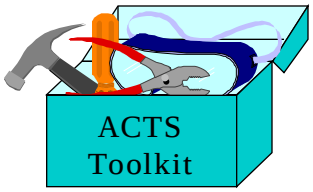




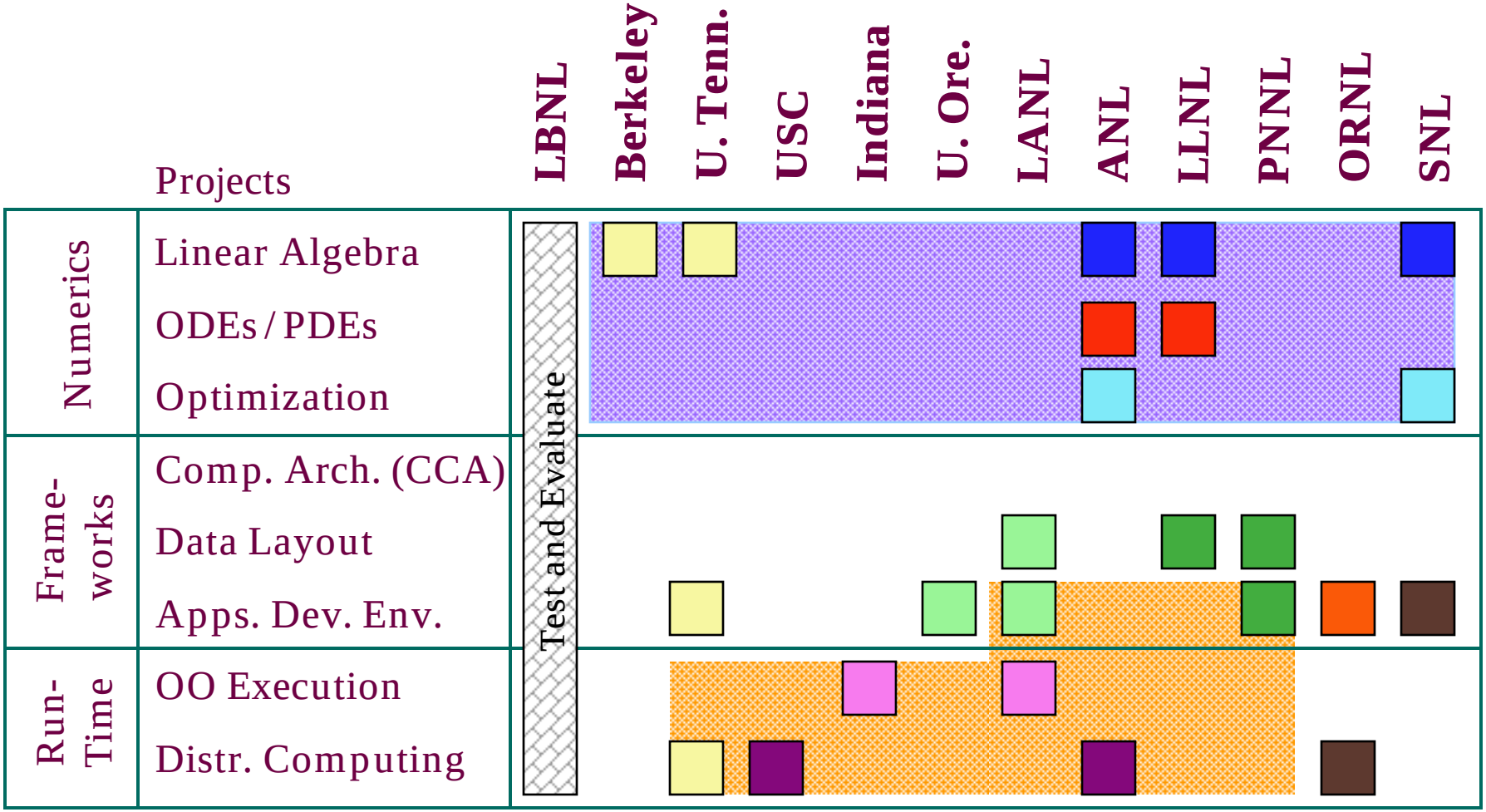


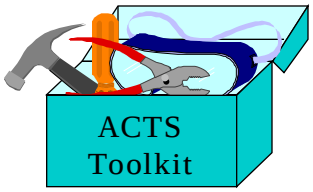
Collaborating Projects



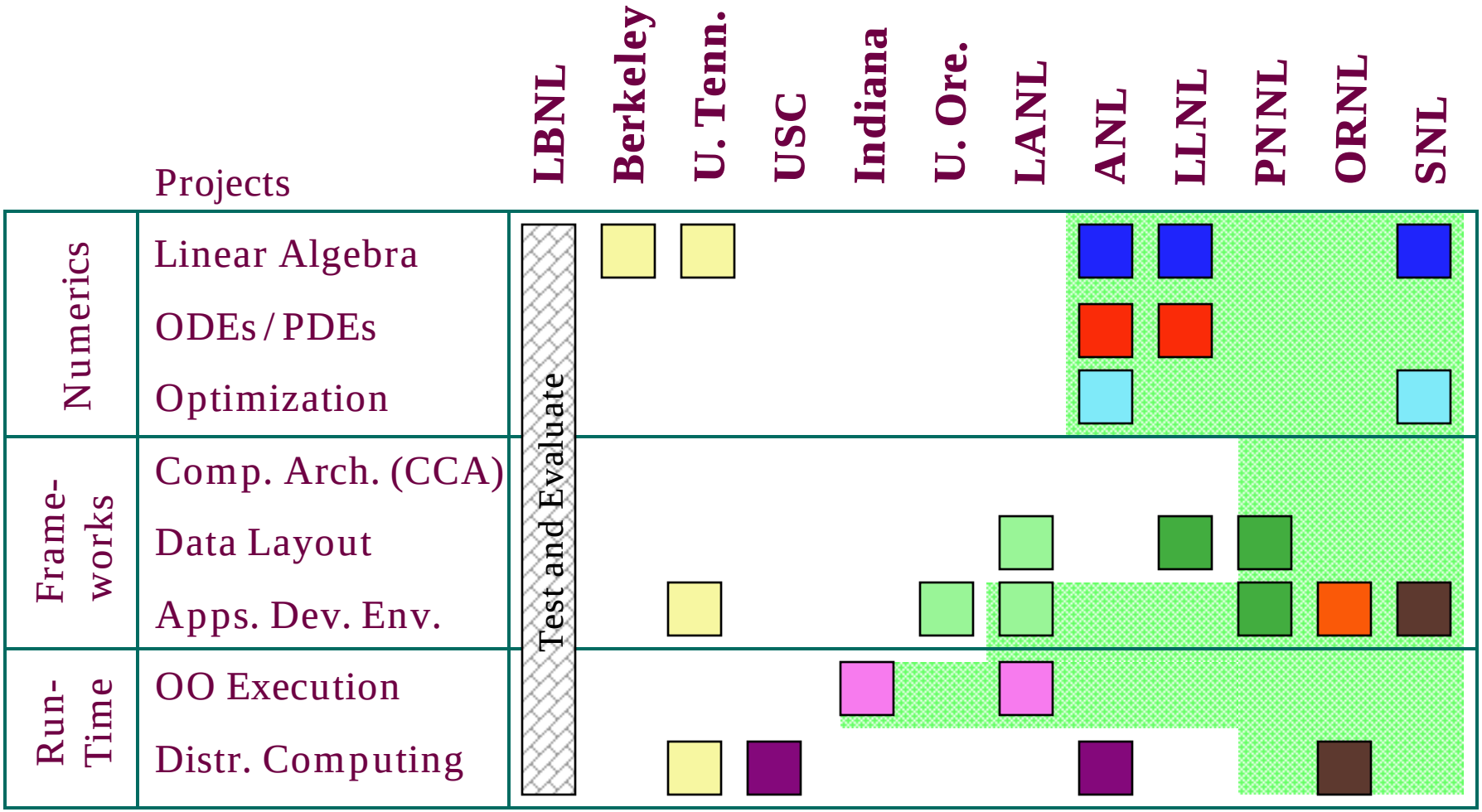


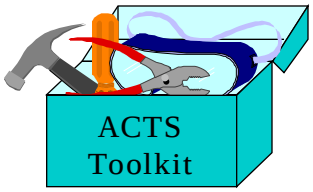
Collaborating Projects





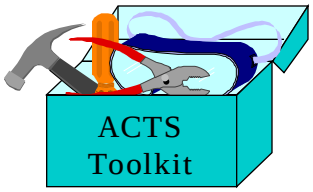
Collaborating Projects





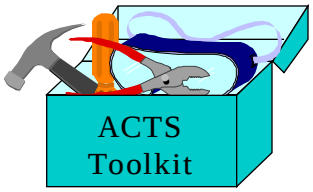
Numerical Toolkit Efforts

- **Large-scale (100,000,000+ DOF) simulations**
  - » **computational fluid dynamics**
  - » **combustion**
  - » **structures**
  - » **materials**
  - » **usually PDE based**
- **Large-scale optimization**
  - » **often involving simulation**
  - » **may be stochastic**



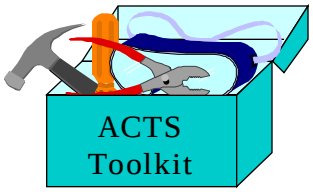
PVODE-PETSc (LLNL-ANL)

- **Complementary functionality**
  - » **Parallel ODE integrators (PVODE)**
    - – **sophisticated time-step control for accuracy**
    - – **special scaled non-linear solver**
    - – **object based**
  - » **Scalable preconditioners and Krylov methods (PETSc)**
    - – **run on 1000+ processors**
    - – **highly efficient block matrix storage formats**
    - – **object oriented**



PVODE-PETSc Use

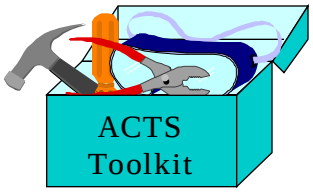
- Sintering - simulating the dynamics of micro-structural interactions via the method of lines, requiring the solution of a large set of coupled ODEs
- Previously used LSODE, limited to 100s of DOF, now can handle 10,000s
- Wen Zhang, Oakland University and Joachim Schneibel, Oak Ridge



Utah ASCI ASAP Center

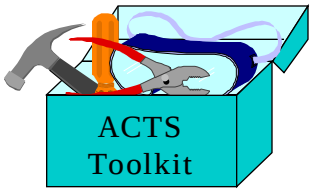
Center for the Simulation of Accidental Fires and Explosions

- **"...problem-solving environment in which fundamental chemistry and engineering physics are fully coupled with non-linear solvers, optimization, computational steering, …"**
- **PETSc + SAMRAI (LLNL)**
- **Using the PETSc nonlinear PDE solvers**
- **Already has fed back into PETSc nonlinear solver enhancements**



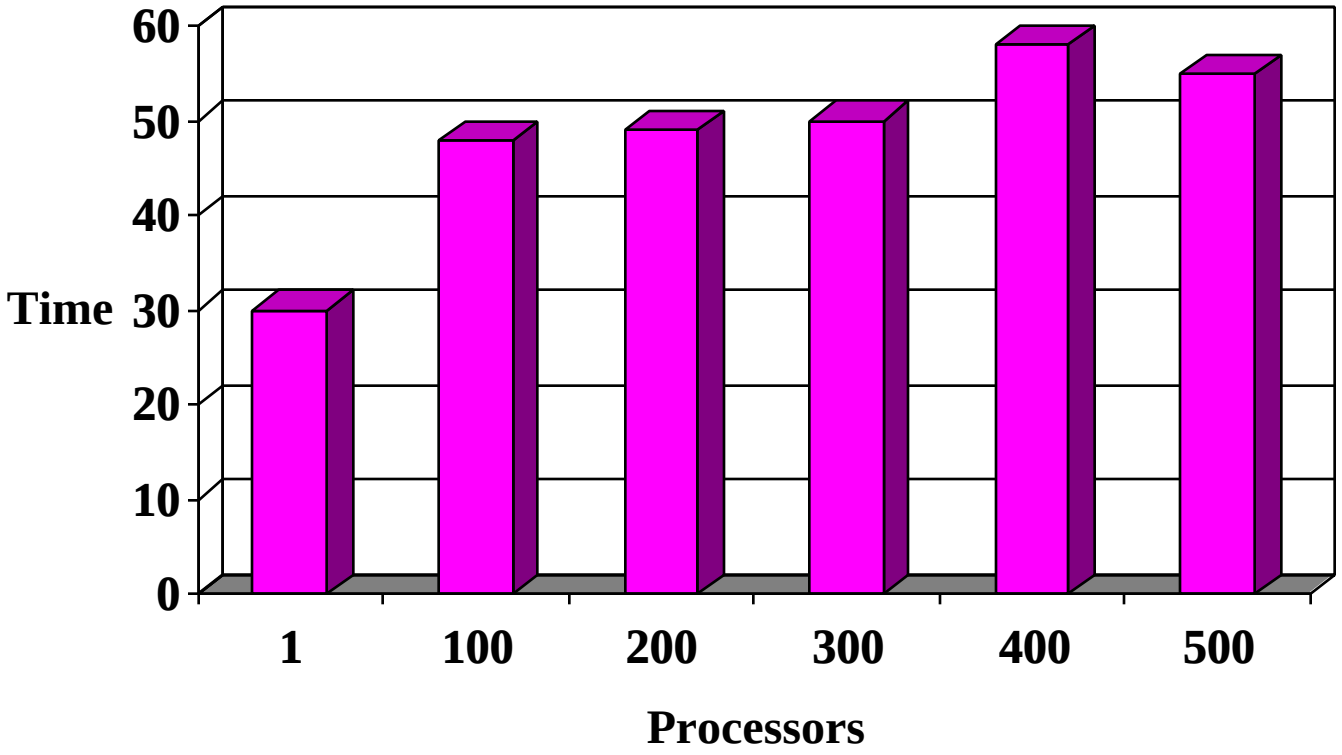
Lab Related Toolkit Usage

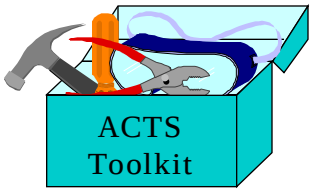
- **ALE3D test problems run with PETSc based parallel multigrid solver**
  - » **Run on NERSC 512 processor T3E and LLNL ASCI Blue Pacific**
  - » **Simulation with 16 million+ degrees of freedom**
  - » **Mark Adams (Berkeley)**
- **Version of ARES and Teton run with PETSc**
- **Ground water flow simulation (gimrt-LLNL)**
- **Multi-phase flow (LANL), sequential legacy code**



ALE3D Test Problem Performance

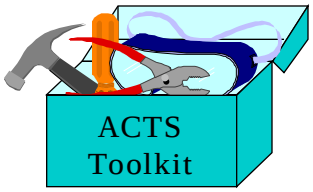
NERSC Cray T3E Scaled Performance
15,000 DOF per processor



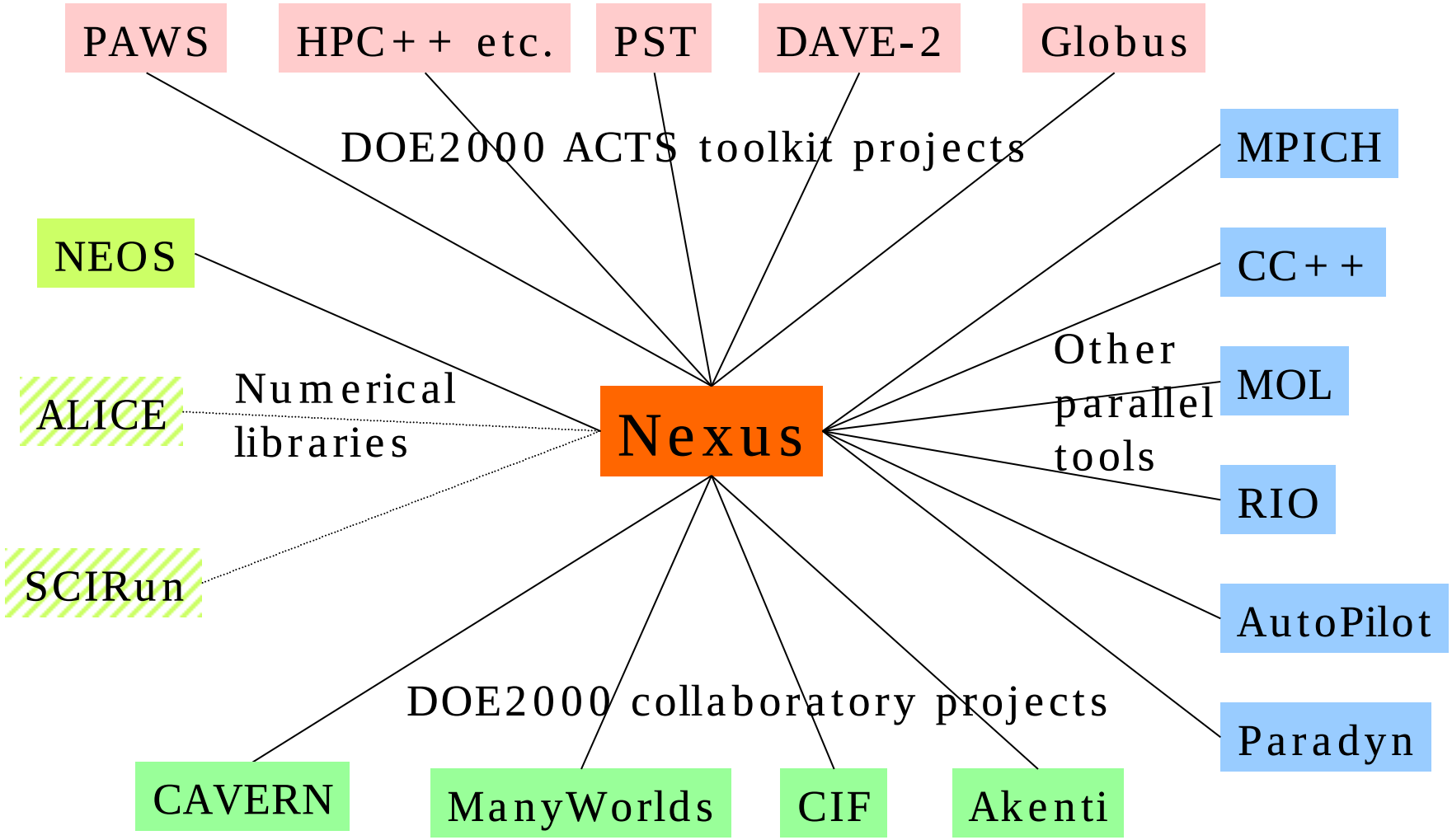


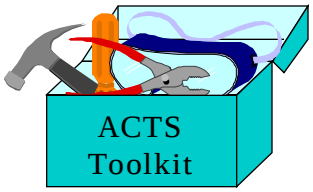
System Software - Distributed Computing and Communications

- **Build on Nexus communication library to provide ACTS toolkit with**
  - » **Integrated support for multithreading and communication**
  - » **High-performance multi-method communication**
  - » **Dynamic process creation**
  - » **Integration with distributed computing**
- **Apply these capabilities to ACTS toolkit components & applications**



Connections: Nexus

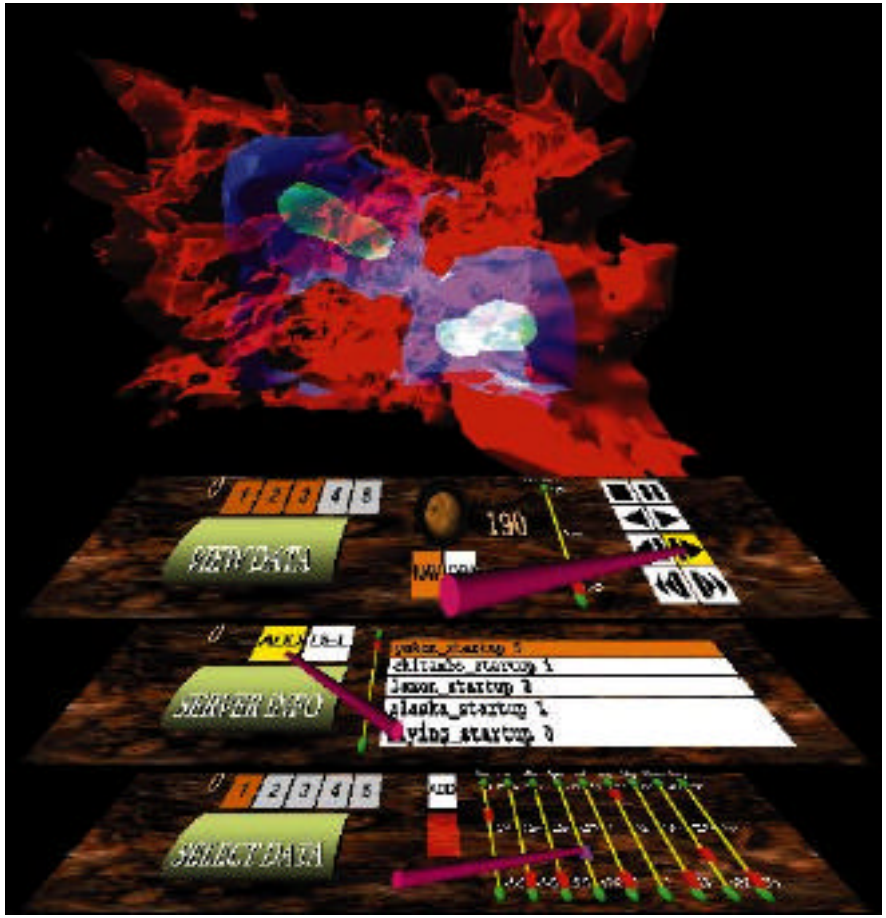


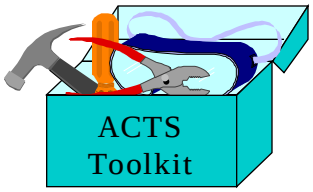


Remote Browsing of Large Datasets

- **Problem: interactively explore very large (TB+) datasets**
- **Interactive client VRUI with view mgmt support**
- **Data reduction at remote client (subsampling)**
- **Use of Globus to authenticate, transfer data, access data**

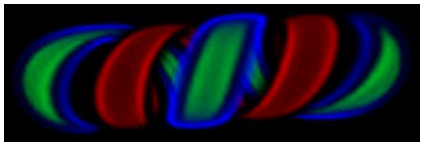
ANL
USC/ISI
UC ASAP
CIT ASAP





Applications in Flash
ASCI ASAP Center

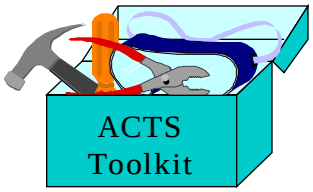
- **Prototype "global shell" that permits:**
  - » **Sign-on once via public key technology**
  - » **Locate available computers**
  - » **Start computation on an appropriate system**
  - » **Monitor progress of computation**
  - » **Get [subsampled] output files**
  - » **Manipulate locally**



the globus project
www.globus.org

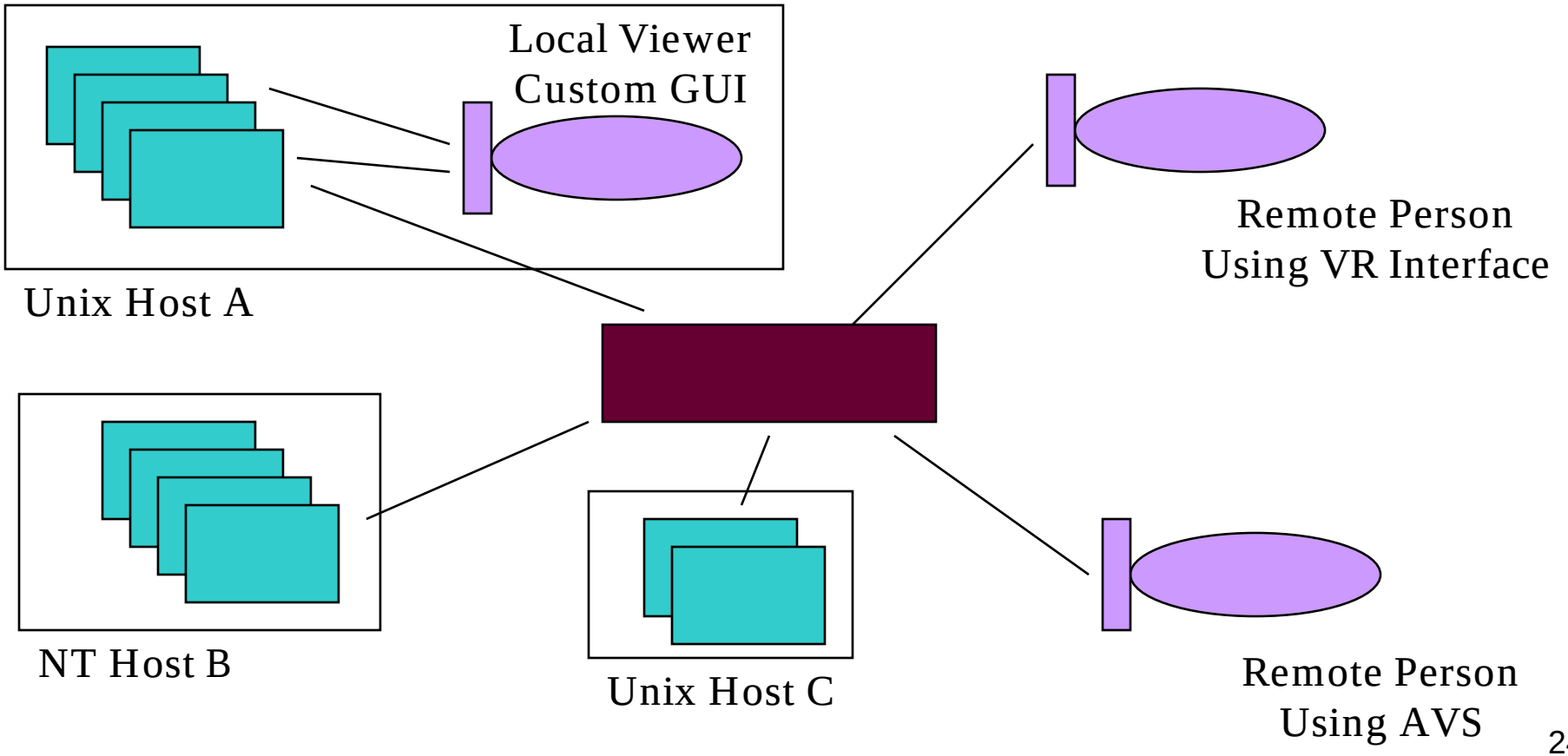
Site - Machine Type	Total	Free
ANL / MCS-DEC	2	2
ANL / MCS-IBM	81	37
ANL / MCS-SGI	114	86
ANL / MCS-SUN	11	10
CalTech / CACR-HP	256	225
The University of Chicago-SGI	8	0
UIUC / NCSA-SGI	672	297
Totals	1144	657

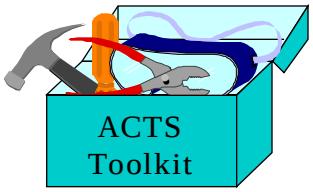
Active Jobs / Pending Jobs



Cumulvs: Collaborative Infrastructure for Interacting with Scientific Simulations

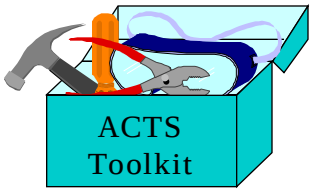
Coordination of collection and dissemination of information to/from parallel tasks to multiple viewers.





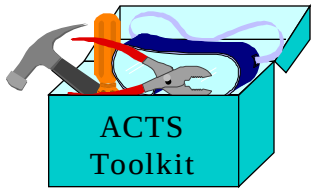
Cumulvs Capabilities

- Multiple distinct data views
- Links to a variety of visualization tools
- Dynamic linking to running application
- Coordinated computational steering
- Hetergeneous checkpointing

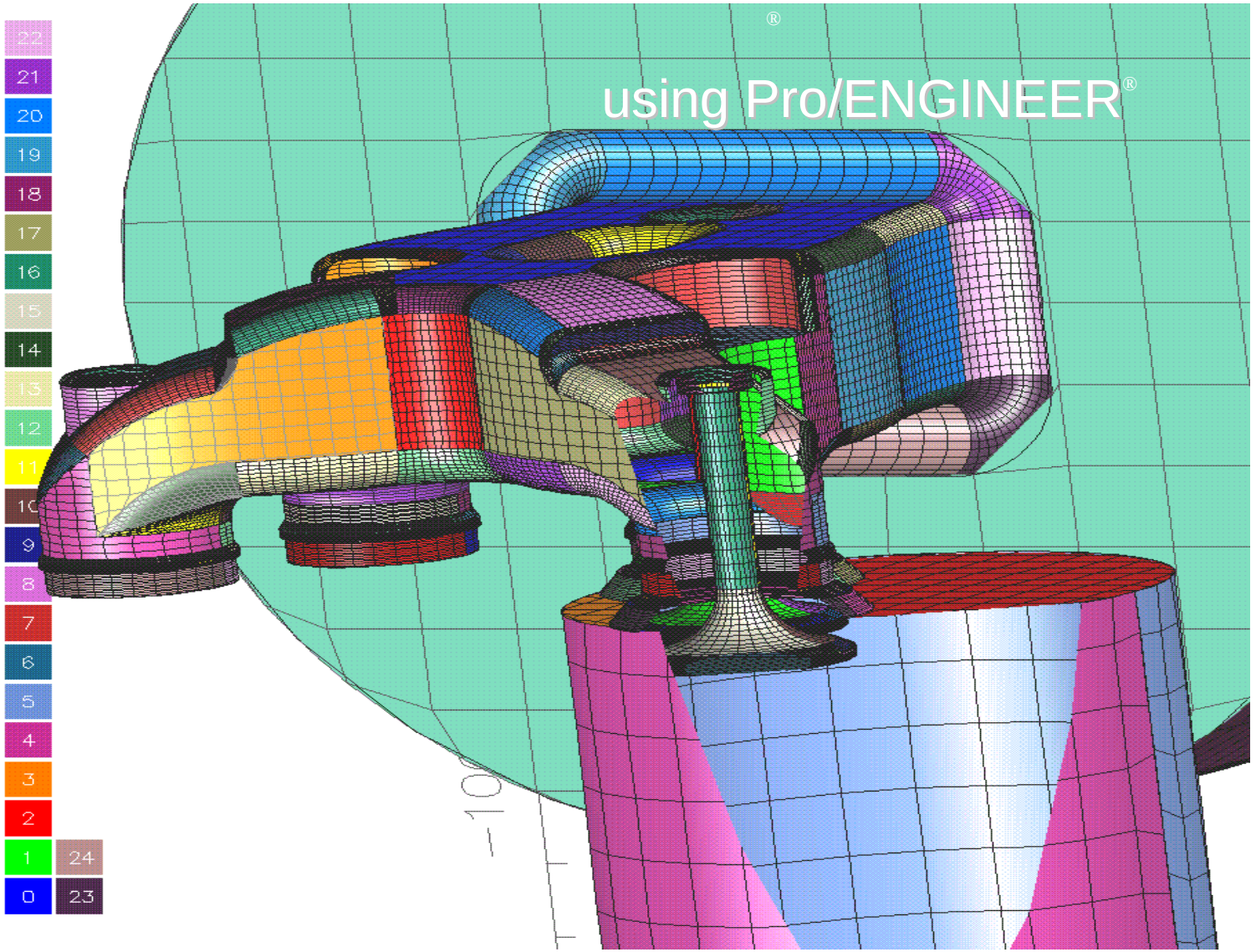


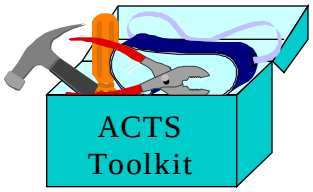
Cumulvs Integration & Interoperability

- **Integration with InDEPS code development environment (ORNL & SNL)**
  - » **Combustion Simulations**
  - » **Material Impact and Deformation**
  - » **Smooth Particle Hydrodynamics**
- **Remotely monitored T3E Applications (ORNL, LBL, and LLNL)**
- **Tcl/Tk language bindings for Cumulvs (NCSA, ORNL)**
  - » **Viewers: VTK & Tango, VR / OpenGL viewer, Immersadesk, and CAVE**
  - » **Apps.: Chesapeake Bay Simulation, Neutron Star Collision, DOD codes**



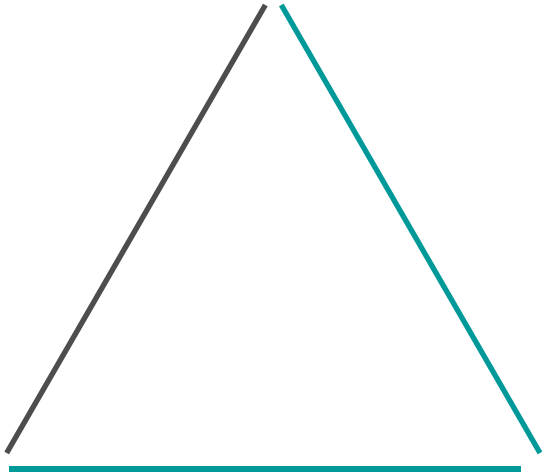
PADRE: Parallel Asynchronous Data Routing Engine





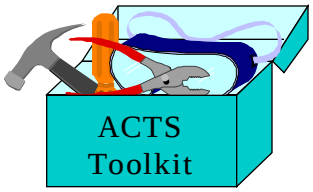
PADRE Purpose : Data Distribution Interoperability

Application Libraries
e.g. Overture, POOMA, AMR++



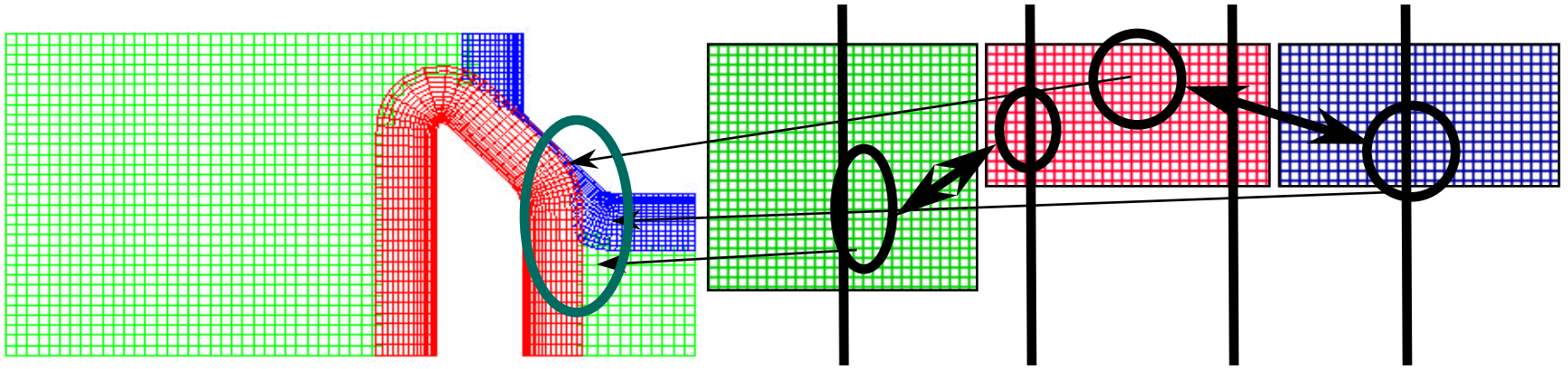
Communication Library
e.g. MPI, PVM

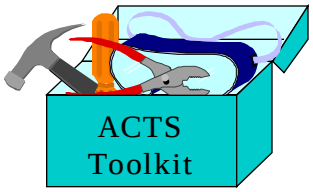
Data Distribution Library
e.g. Multi-block PARTI (UM), KeLP (UCSD), PGSLib (LANL), Global Arrays (PNNL)



PADRE Services

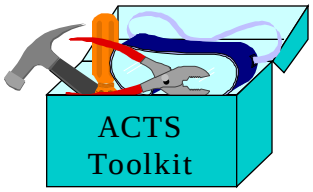
- Dynamic Distribution (redistribution, etc.)
- Generation and Execution of Communication Schedules
  - » General Array Language Support
  - » TULIP Interface Support
  - » Cached schedules for performance
  - » Lazy Evaluation
  - » Message Latency Hiding
- Subarray operation support for different distributions (AMR)
- Gather/Scatter support for indirect addr. (Overlapping Grid Apps)



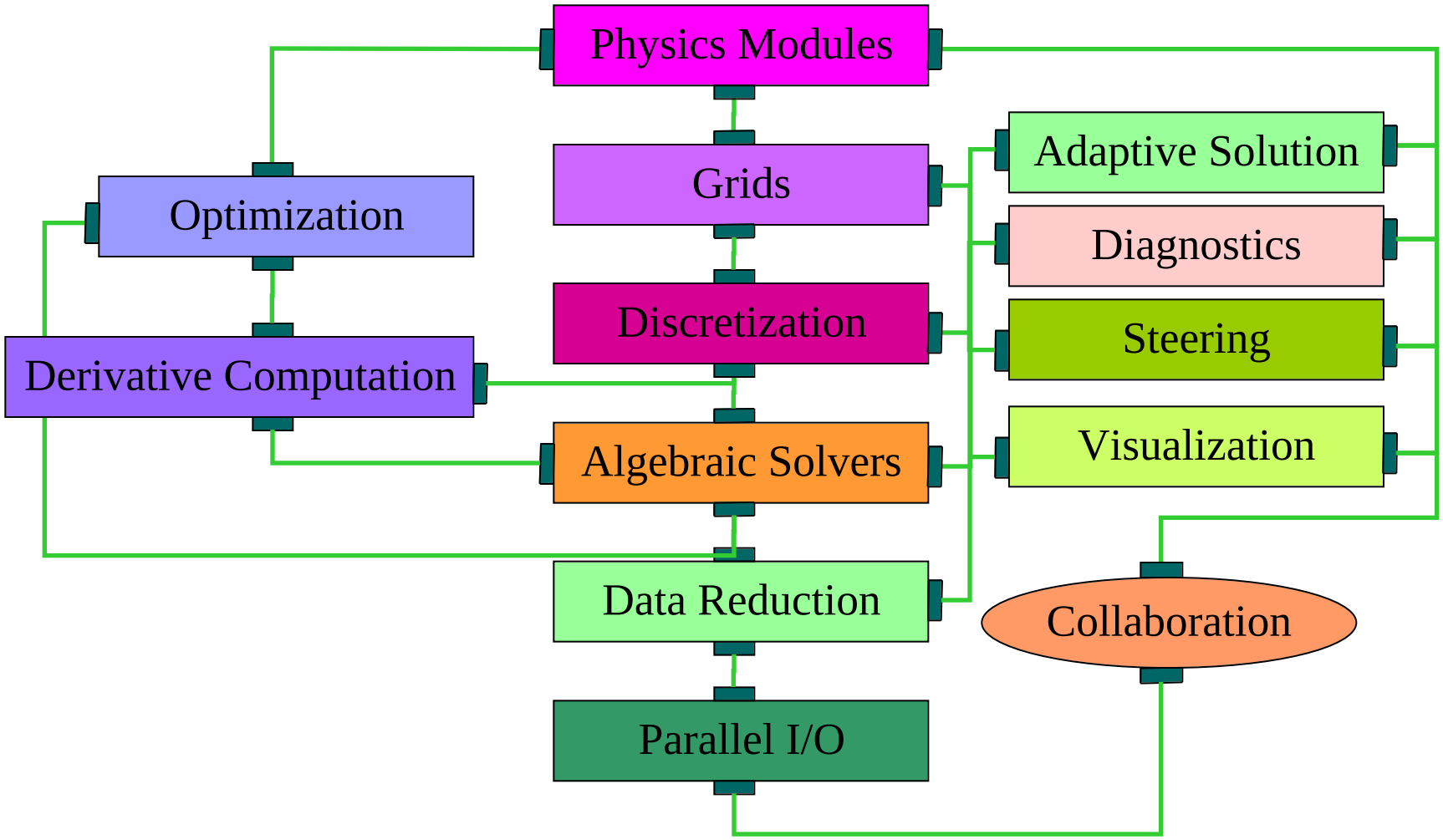


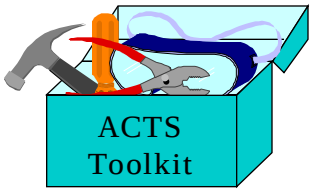
PADRE in Use

- **Default data distribution mechanism in Overture**
  - » **Permits multiple distribution mechanisms within Overture for specialized applications (e.g. AMR)**
  - » **Optimized communication using standard MPI and ROSE Optimizing preprocessor**
- **Global Arrays**
  - » **Will provide alternative to GA interface for C++ users of GA**
  - » **Provides access to Geometry in a form identical to what they already use**
  - » **Gives PADRE access to one-sided message passing (important for parallel grid generation mechanisms within Overture)**
- **Ready for Use in POOMA II**
  - » **KeLP distributions in PADRE close to domain layout in POOMA**
  - » **Can provide geometry and AMR capabilities to POOMA apps.**



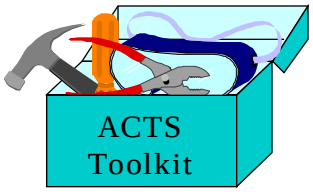
Interoperability Goal: Scientific Apps. via Plug-and-Play





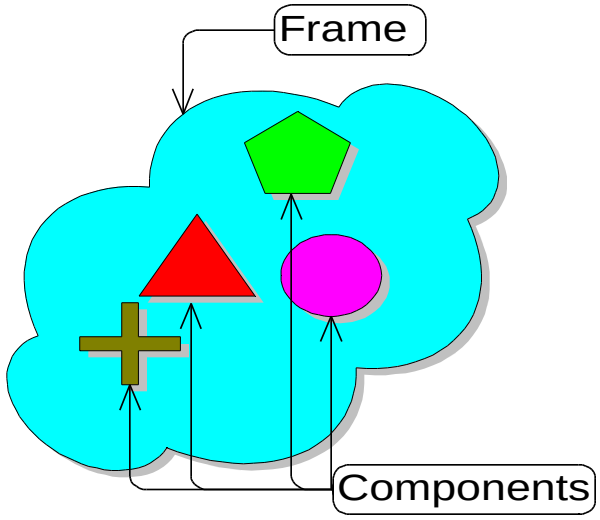
Common Component Architecture: Active Participants

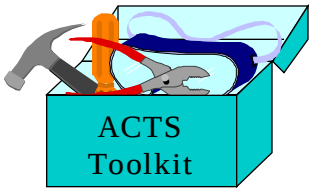
- Cal Tech
- Indiana University
- University of Utah
- Sandia National Laboratory
- Argonne National Laboratory
- Lawrence Berkeley National Laboratory
- Lawrence Livermore National Laboratory
- Los Alamos National Laboratory
- Oak Ridge National Laboratory
- Pacific Northwest Laboratories



Basic Idea

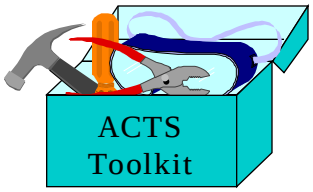
- **Component Concepts**
  - » **Autonomous interchangeable parts**
  - » **Links between components that create an application**
  - » **Builders**
  - » **Framework provides context or environment to components**





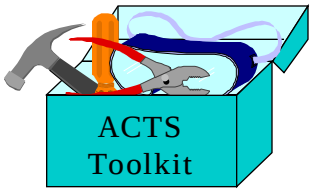
Existing Component Models: The good, bad, and ugly

- What's good about them?
  - » Everyone knows how they should work.
  - » They are accepted standards.
- What's bad about them?
  - » Not geared toward high performance computing.
  - » Often based on a single framework.
  - » Geared toward a single language (e.g. Beans).
  - » Meant for a single environment (e.g. COM).



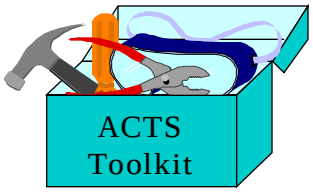
CCA Extensions to Existing Models

- **gPorts (generalized Ports)**
  - » **Similar to the CORBA3.0 User/Provider Port spec.**
  - » **Uses a type of *IUnknown* pattern similar to COM.**
  - » **Stolen from data-flow component models**
  - » **Defined by *linked interfaces*.**
  - » **Draft gPort spec in pre-RFC stage:**
    `http://z.ca.sandia.gov/~cca-forum/gport-spec`
- **Scientific IDL**
  - » **Once component model is defined, language interoperability is *the* issue.**
    - – **Include legacy programs.**
    - – **Include legacy programmers**
  - » **Draft IDL spec in pre-RFC stage:**
    `http://www.llnl.gov/CASC/babel`



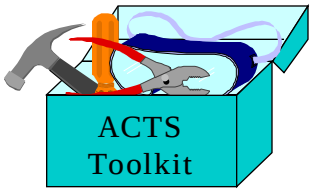
ACTS Toolkit: Remaining Holes

Visualization & Analysis Tools	Integration of tools for helping application users to visualize and analyze the results of complex 3D computations.
Application Support Tools	Expansion of tools for helping application developers share common domain-specific tools, such as adaptive mesh refinement and unstructured grids.
Numerics	Automated differentiation tools and ???
Code Development Tools	Expansion of tools to provide portable performance, multi-language programming environments, and tools for software maintenance of scientific applications.
Code Execution Tools	Expansion of tools to provide parallel I/O, dynamic load-balancing, intelligent storage management.



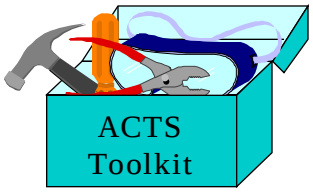
A Management Perspective

- "Resistance is futile -- you will collaborate."
- Tool identities vs. Toolkit vision
- Planning for the future
  - » precise architecture or loose confederation?
  - » evolving tool development
- Buy-in from the contributors
- Enlightened oversight is critical



Final Thoughts

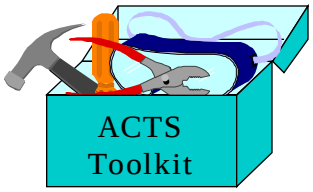
- A different model for R&D cooperation
- Success to date shows promise
- Partnerships must extend beyond DOE
- Much more work to be done



For More Information

- **Web Pages:**
  - » Overview: `www.mcs.anl.gov/DOE2000/`
  - » ACTS Toolkit: `www.nersc.gov/ACTS/`

- **Technical Contacts:**
  - » Collaboratories: Stu Loken - `SCLoken@lbl.gov`
  - » ACTS Toolkit: Jim McGraw - `JMcGraw@llnl.gov`



ACTS Toolkit: Current Contents

	Round 1 Funding	Round 2 Funding
Application Support Tools		Develop object-oriented tools for assisting code developers to write particle simulation-based applications.
Numerics	Adapt previously autonomous numerical libraries to permit interoperable use on selected DP applications.	Expand the interoperable behavior with more numerical algorithms relevant to ER and other DOE applications.
Code Development Tools	Adapt and integrate object-oriented libraries for defining and manipulating parallel distributed arrays.	Expand tools for managing arrays on complex memory hierarchies and enable run-time linking to diverse class libraries.
Code Execution Tools	Develop a portable, parallel run-time class library and integrate it with tools for remote steering and visualization.	Incorporate use of parallel program debugging and analysis tools with the current system and improve the tools for use on computational grids.