CCA Status, Code Walkthroughs, and Demonstrations

CCTTSS Tutorial Working Group
http://www.cca-forum.org/tutorial/

This work has been sponsored by the Mathematics, Information and Computational Sciences (MICS) Program of the U.S. Dept. of Energy, Office of Science.

Contributors

- Ben Allan, SNL
- Rob Armstrong, SNL
- David Bernholdt, ORNL
- Lori Freitag, ANL
- Jim Kohl, ORNL
- Lois Curfman McInnes, ANL
- *Boyana Norris, ANL (Presenter)*
- Craig Rasmussen, LANL
- Jaideep Ray, SNL

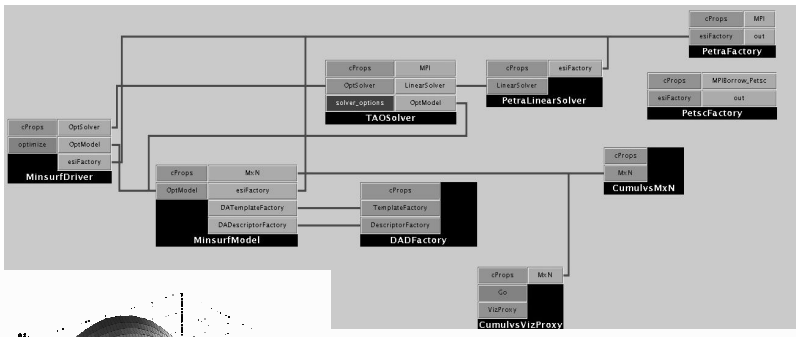
Prototype CCA Frameworks

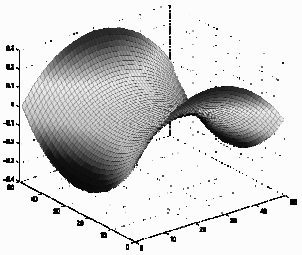
- XCAT, Indiana University, Dennis Gannon
  - Distributed
  - Network connection
- CCAFFEINE, Sandia National Laboratories, Rob Armstrong
  - SPMD/SCMD parallel
  - Direct connection
- SCIRun/Uintah, University of Utah, Steve Parker
  - Parallel, multithreaded
  - Direct connection

Current Status of CCA

- Specification version 0.5
- Working prototype frameworks
- Working multi-component parallel and distributed demonstration applications
- Draft specifications for
  - Basic scientific data objects
  - MxN parallel data redistribution
- SC01 demonstrations *available for download*
  - four different “direct connect” applications, add’l distributed
  - DC demos: 31 distinct components, up to 17 in any single application, 6 used in more than one application
  - Components leverage and extend parallel software tools including CUMULVS, GrACE, LSODE, MPICH, PAWS, PETSc, PVM, SUMAA3d, TAO, and Trilinos.

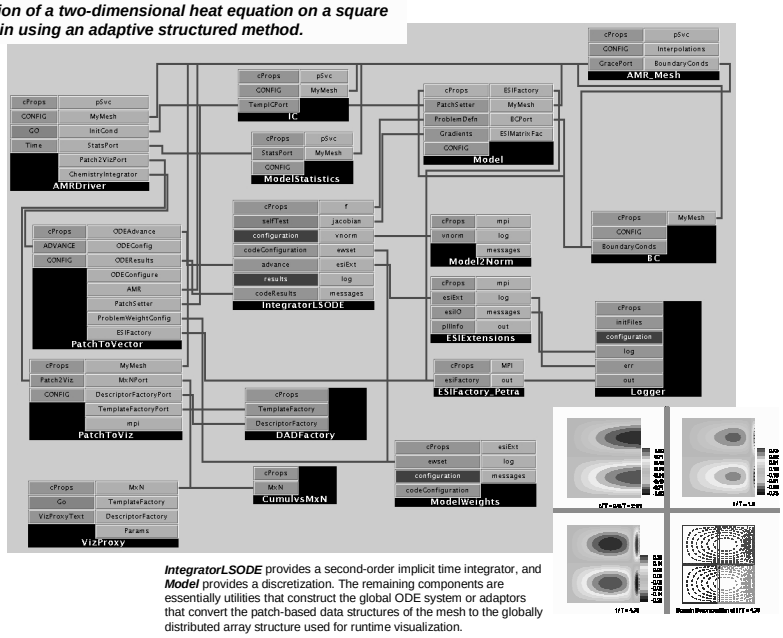
Solution of an unconstrained minimization problem (determining minimal surface area given boundary constraints) using the TAOSolver optimization component



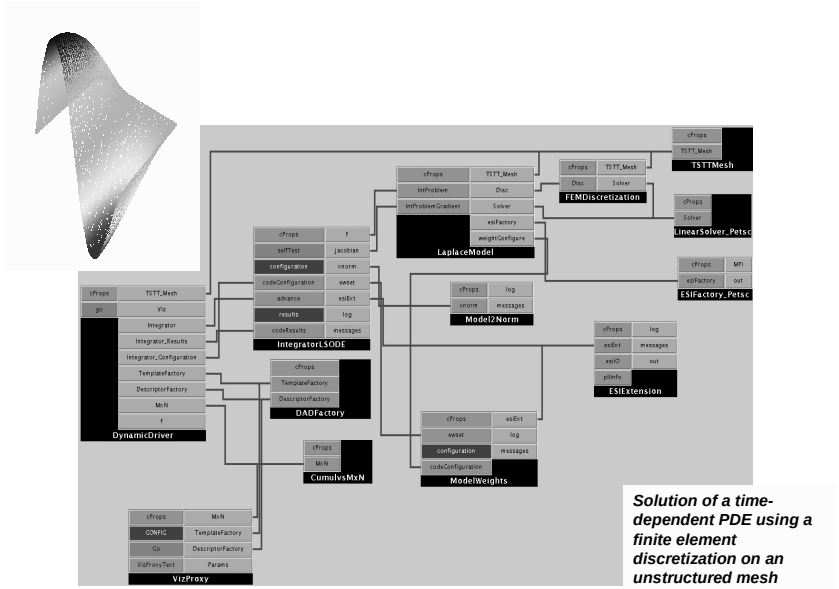


TAOSolver uses linear solver components that incorporate abstract interfaces under development by the Equation Solver Interface (ESI) working group; underlying implementations are provided via the new ESI interfaces to parallel linear solvers within the PETSc and Trilinos libraries. These linear solver components are employed in the other two applications as well.

Solution of a two-dimensional heat equation on a square domain using an adaptive structured method.



IntegratorLSODE provides a second-order implicit time integrator, and ***Model*** provides a discretization. The remaining components are essentially utilities that construct the global ODE system or adaptors that convert the patch-based data structures of the mesh to the globally distributed array structure used for runtime visualization.



Solution of a time-dependent PDE using a finite element discretization on an unstructured mesh

IntegratorLSODE provides a second-order implicit time integrator, and ***FEMDiscretization*** provides a discretization.

This application (and the other two applications as well) use the ***DADFactory*** component to describe the parallel data layout so that the ***CumulsMxN*** data redistribution component can then collate the data from a multi-processor run to a single processor for runtime visualization.