Enabling Scientific Applications with the Common Component Architecture

David E. Bernholdt

Oak Ridge National Laboratory

for the CCTTSS

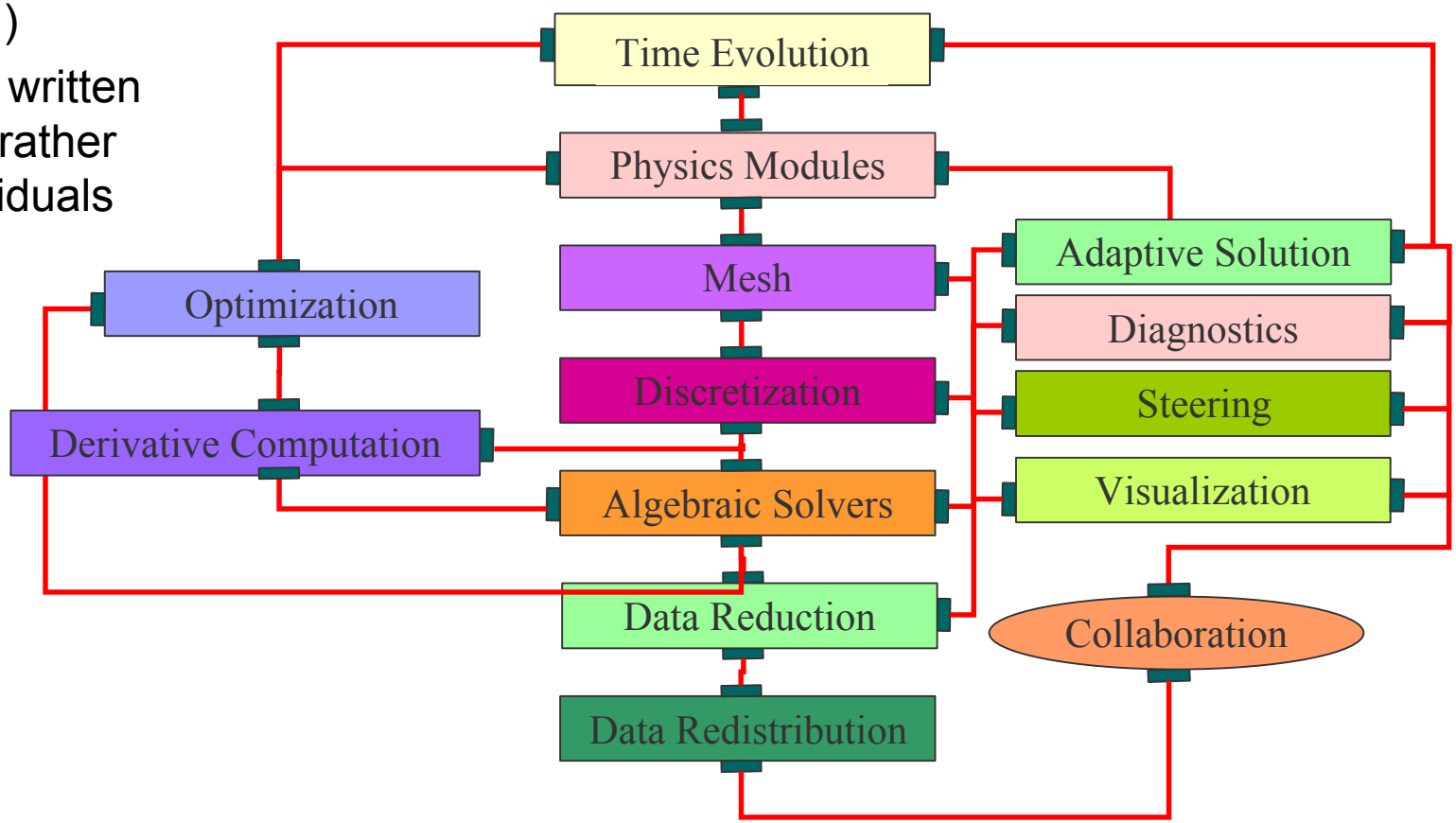
ANL, Indiana, LANL, LLNL, ORNL, PNNL, SNL, Utah

and the CCA Forum

http://www.cca-forum.org

Modern Scientific Software Development

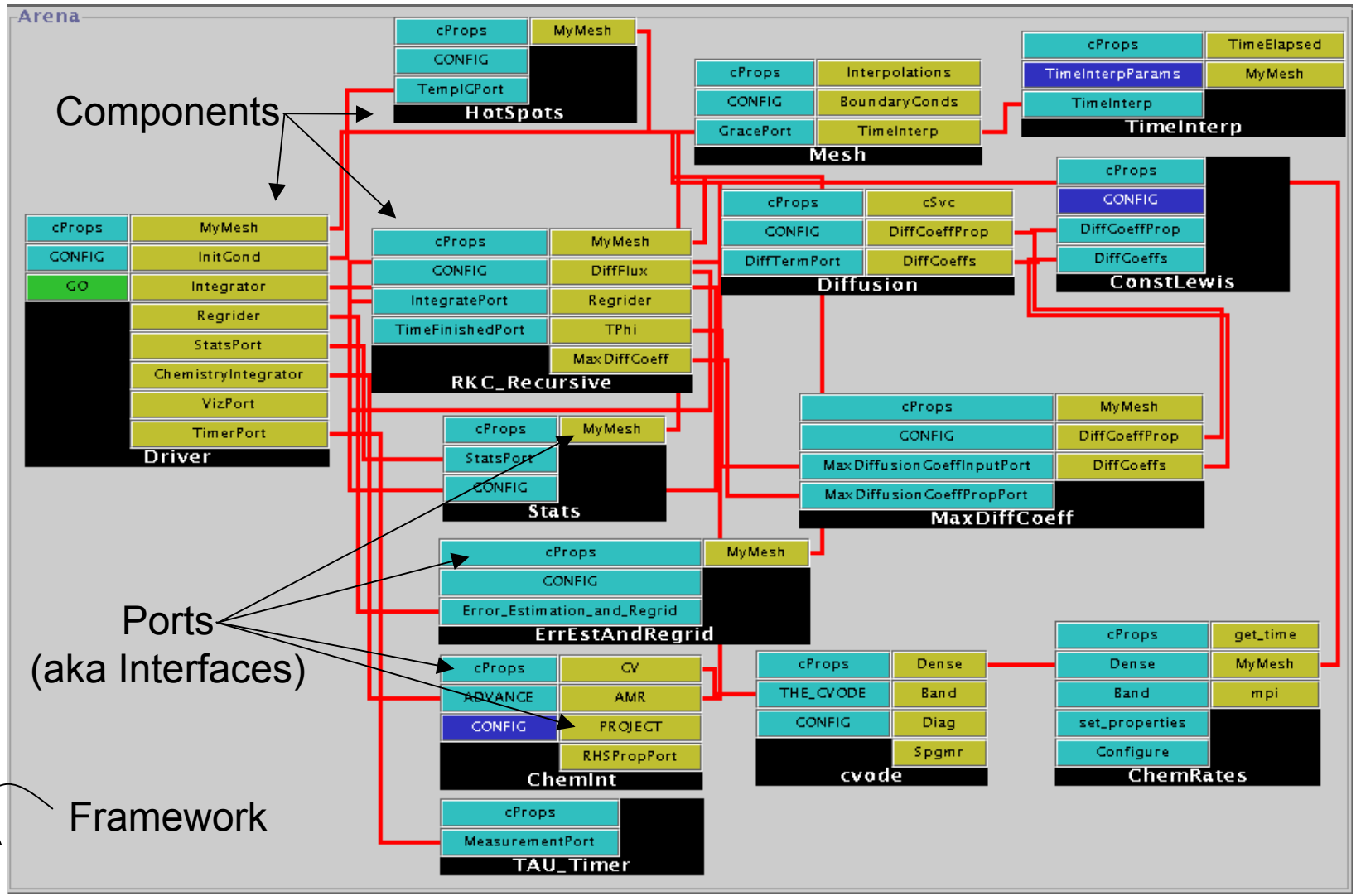
- Complex codes, often coupling multiple types of physics, time or length scales, involving a broad range of computational and numerical techniques
- Different parts of the code require significantly different expertise to write (well)
- Generally written by teams rather than individuals



Component-Based Software Engineering

- **Software productivity**
  - Provides a "plug and play" application development environment
  - Many components available "off the shelf"
  - Abstract interfaces facilitate reuse and interoperability of software
    - *"The best software is code you don't have to write" [Jobs]*
  - Facilitates collaboration around software development
- **Software complexity**
  - Components encapsulate much complexity into "black boxes"
  - Facilitates separation of concerns/interests
  - Plug and play approach simplifies applications & adaptation
  - Model coupling is natural in component-based approach
- **Software performance** (indirect)
  - Plug and play approach and rich "off the shelf" component library simplify changes to accommodate different platforms
- ***CCA is a component environment designed specifically for the needs of HPC scientific computing***

"Wiring Diagram" for Typical CFRFS Application



CCA Delivers Performance

Local

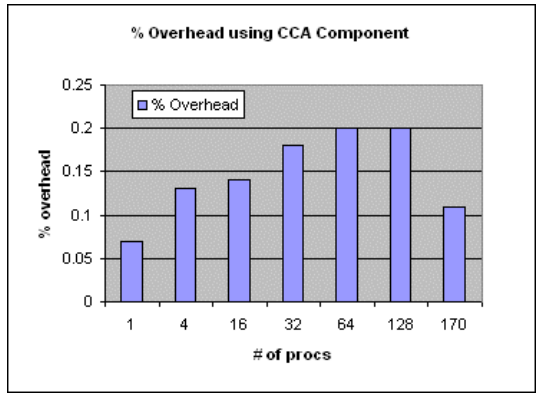
- No CCA overhead within components
- Small overhead between components
- Small overhead for language interoperability
- Be aware of costs & design with them in mind
  - Small costs, easily amortized

Parallel

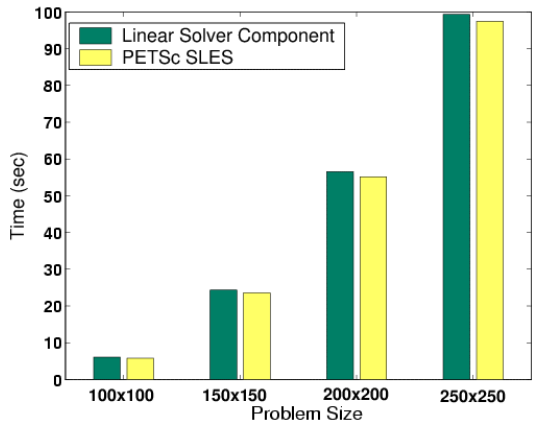
- No CCA overhead on parallel computing
- Use your favorite parallel programming model
- Supports SPMD and MPMD approaches

Distributed (remote)

- No CCA overhead – performance depends on networks, protocols
- CCA frameworks support OGSA/Grid Services/Web Services and other approaches



Maximum 0.2% overhead for CCA vs native C++ code for parallel molecular dynamics up to 170 CPUs



Aggregate time for linear solver component in unconstrained minimization problem w/ PETSc

Easy, Flexible Componentization of Existing Software

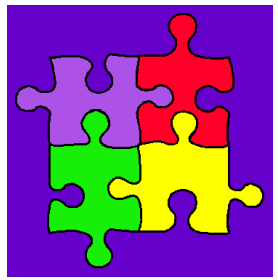
Suitably structured code (programs, libraries) should be relatively easy to adapt to the CCA. Here's how:

1. Decide level of componentization
   - Can evolve with time (start with coarse components, later refine into smaller ones)
2. Define interfaces and write wrappers between them and existing code
3. Add framework interaction code for each component
   - **setServices**
4. Modify component internals to use other components as appropriate
   - **getPort**, **releasePort** and method invocations

CCA Research Thrusts

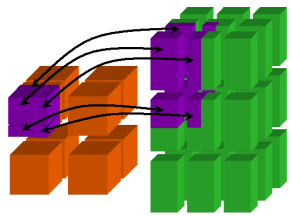
- Frameworks
  - Frameworks (parallel, distributed)
  - Language Interoperability / Babel / SIDL

  Gary Kumfert, LLNL (kumfert@llnl.gov)



- "MxN" Parallel Data Redistribution

  Jim Kohl, ORNL (kohlja@ornl.gov)

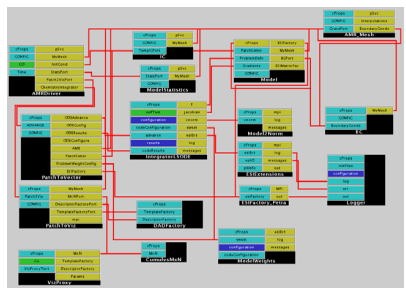


- Scientific Components
  - Scientific Data Objects

  Lois Curfman McInnes, ANL (curfman@mcs.anl.gov)

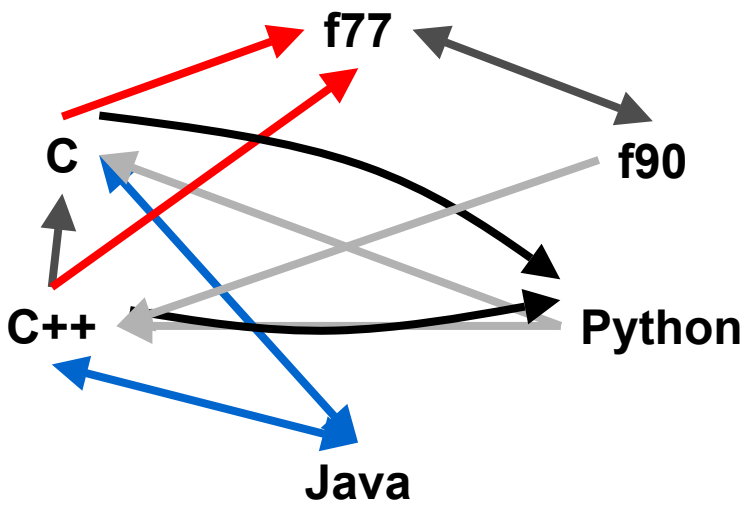
- User Outreach and Applications
  - Tutorials, Coding Camps
  - Interactions with users

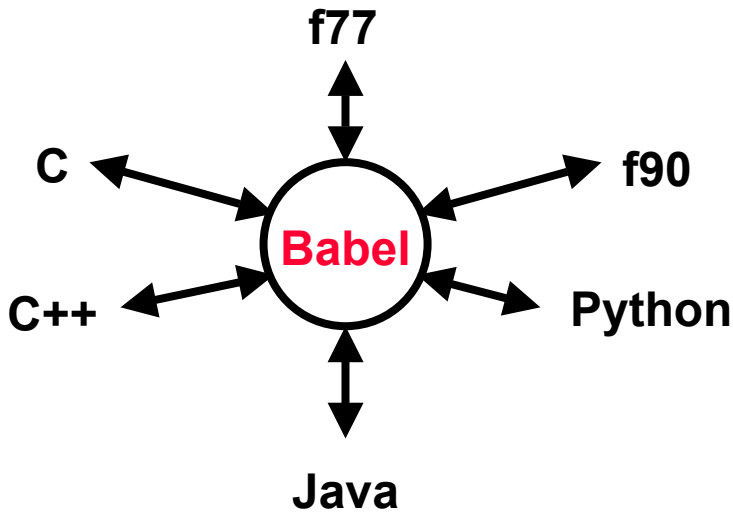
  David Bernholdt, ORNL (bernholdtde@ornl.gov)



Language Interoperability

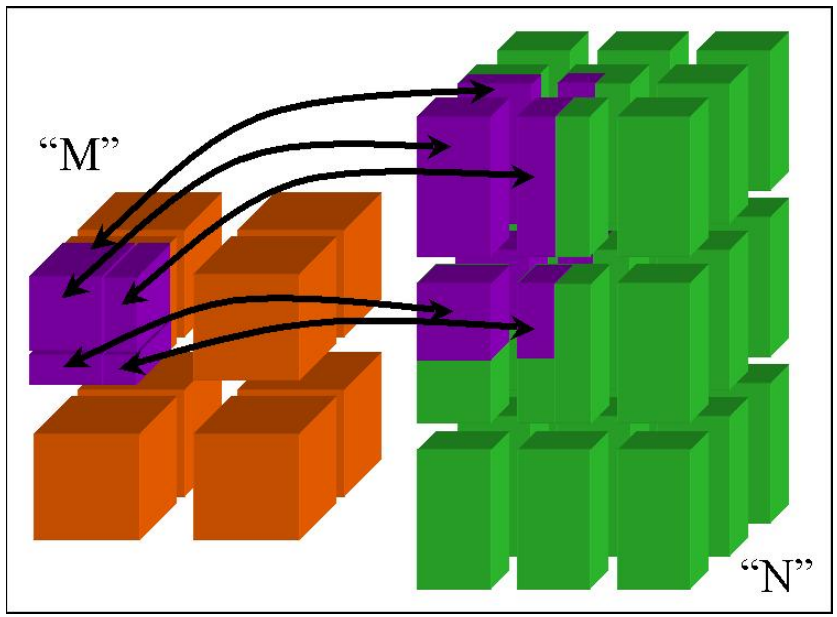
- Existing language interoperability approaches are "point-to-point" solutions
- Babel provides a unified approach in which all languages are considered peers
- Babel used primarily at interfaces





MxN Parallel Data Redistribution

- Share Data Among Coupled Parallel Models
  - Disparate Parallel Topologies (M processes vs. N)
  - e.g. Ocean & Atmosphere, Solver & Optimizer…
  - e.g. Visualization (Mx1, increasingly, MxN)



Research area -- tools under development

Many Scientific Components

Data Management, Meshing and Discretization

- Global Array Component, TSTTMesh, FEMDiscretization, GrACEComponent

Integration, Optimization, and Linear Algebra

- CvodesComponent, TaoSolver, LinearSolver

Parallel Data Description, Redistribution, and Visualization

- DistArrayDescriptorFactory, CumulvsMxN, VizProxy

Services, Graphical Builders, and Performance

- Ccaffeine Services, Graphical Builders, Performance Observation, Port Monitor

Component	Derived From
Distributed Array Descriptor	(New)
Distributed Arrays	Global Arrays
Parallel IO/Data Mgmt	HDF5/PnetCDF
Linear Algebra	Global Arrays
Linear Algebra	ScaLAPACK
Linear Algebra	TOPS SciDAC ISIC
Sparse Linear Algebra	Epetra
Simple Mesh	(New)
Unstructured Mesh	TSTT SciDAC ISIC
Structured AMR	GrACE
Parallel Integrator	CVODE
Parallel Visualization	CUMULVS
Performance Measurement	TAU
Optimization	TAO

Current CCA Application Areas

SciDAC:

- Combustion (CFRFS)
- Climate Modeling (CCSM)
- Meshing Tools (TSTT)
- (PDE) Solvers (TOPS)
- IO, Poisson Solvers (APDEC)
- Fusion (CMRS)
- Supernova simulation (TSI)
- Accelerator simulation (ACCAST)
- Quantum Chemistry

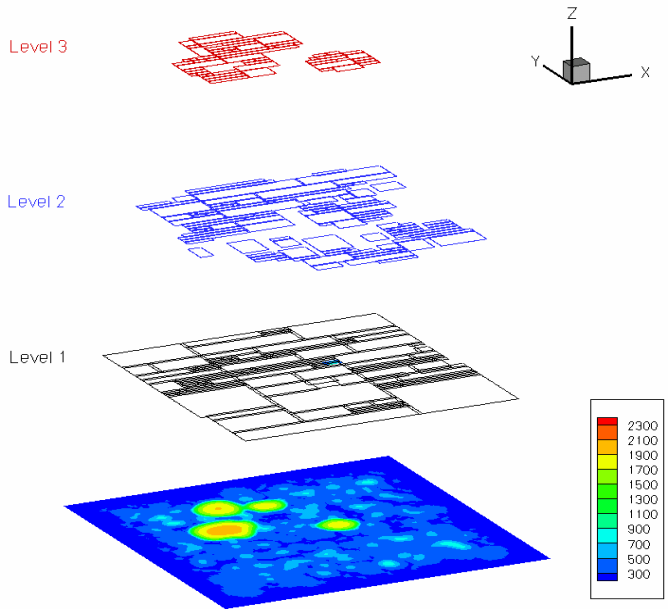
DOE Outside of SciDAC:

- ASCI: C-SAFE, Views, Data Svc's
- Quantum Chemistry
- Materials Science (ORNL LDRD, ANL Nano)
- Fusion (ORNL LDRD)
- Underground radionuclide transport
- Multiphase Flows

Outside of DOE:

- **NASA:** ESMF, SWMF
- Etc.…

Computational Facility for Reacting Flow Science (CFRFS)



- **SciDAC BES project, H. Najm PI**
- **Investigators:** Sofia Lefantzi (SNL), Jaideep Ray (SNL), Sameer Shende (Oregon)
- **Goal:** A "plug-and-play" toolkit environment for flame simulations

- H_2-Air ignition on a structured adaptive mesh, with an operator-split formulation
- RKC for non-stiff terms, BDF for stiff
- 9-species, 19-reactions, stiff mechanism
- 1cm x 1cm domain; max resolution = 12.5 microns
- Kernel for a 3D, adaptive mesh low Mach number flame simulation capability in SNL, Livermore

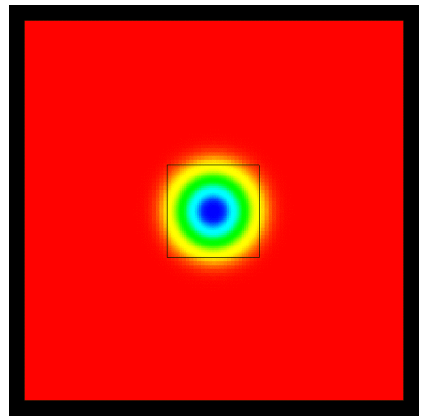
CFRFS Incorporates APDEC Technology using CCA

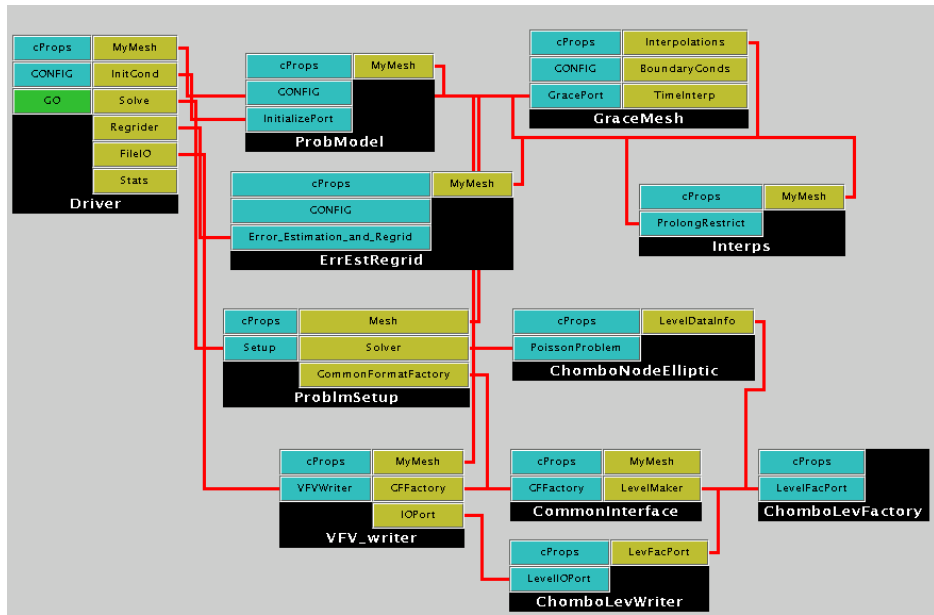
Investigators: Jaideep Ray (SNL), Brian van Straalen (LBL), and Phil Colella (LBL)

- CFRFS needs solvers for elliptic PDEs discretized on a structured adaptive mesh
  - Esp. pressure Poisson eq.
- APDEC solvers (Chombo) also address time and resource constraint issues
- Software reuse via common interfaces provides long-term benefits
- Also use APDEC's
  - HDF5 writer component to save the files
  - ChomboVis to visualize

$$\nabla^2 \Phi = R$$

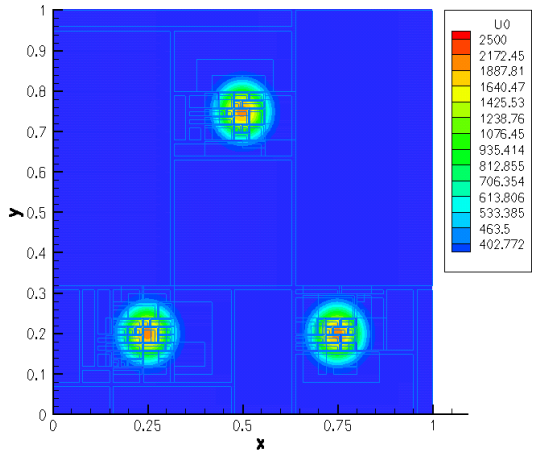
Solution

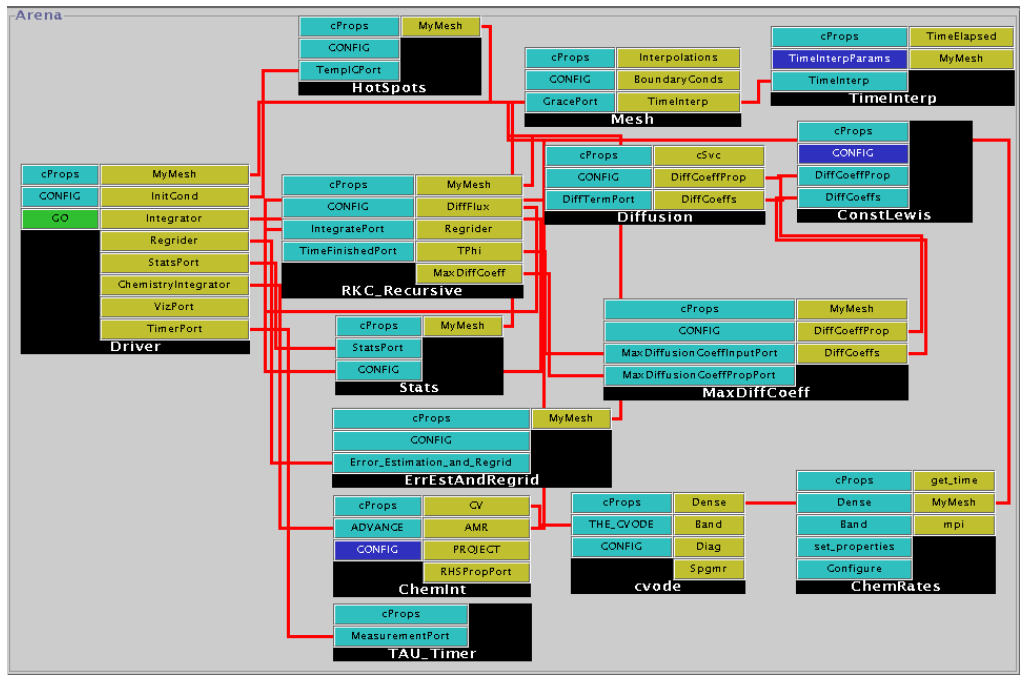




Component-Based CFRFS Applications

- Components mostly C++ or wrappers around old F77 code
- Developed numerous components
  - Integrator, spatial discretizations, chemical rates evalutator, etc.
  - Structured adaptive mesh, load-balancers, error-estimators (for refining/coarsening)
  - In-core, off-machine, data transfers for post-processing
- Integrating solver and viz capabilities from Chombo (LBL, APDEC)
- TAU for timing (Oregon, PERC)
- CVODES integrator (LLNL, TOPS)





Component-Based Integration of Chemistry and Optimization Packages: Molecular Geometry Optimization

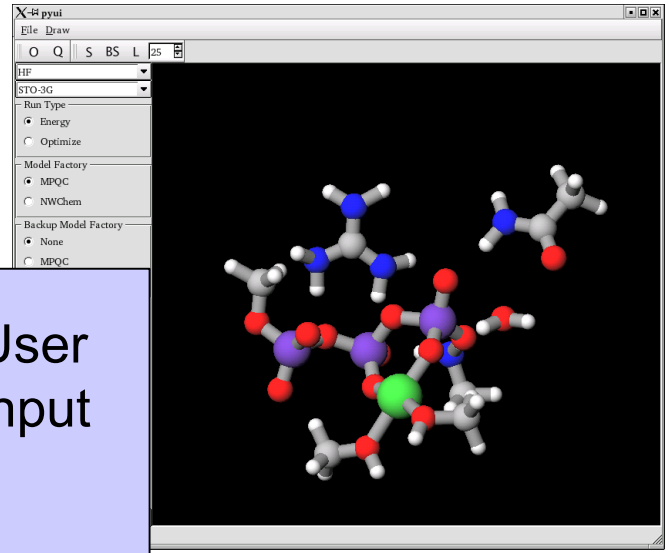
Investigators: Yuri Alexeev, Manoj Krishnan, Jarek Nieplocha, Theresa Windus (PNNL), Curtis Janssen, Joseph Kenny (SNL), Steve Benson, Lois McInnes, Jason Sarich (ANL), David Bernholdt (ORNL)

Underlying software packages:

- Quantum Chemistry
  - NWChem (PNNL), MPQC (SNL)
- Optimization
  - Toolkit for Advanced Optimization (TAO, ANL)
- Linear Algebra
  - Global Arrays (PNNL), PETSc (ANL)

- Performance evaluation of optimization components
  - Examine efficiency of algorithms in TAO for quantum chemistry
- Further development of optimization capabilities
  - Provide internal coordinate generation, constrained optimization, configurable convergence control
- Future plans: Exploring chemistry package integration through hybrid calculation schemes and sharing of lower-level intermediates such as integrals and wavefunctions

Software Architecture



Builder
Construct application using framework builder services

build options

GUI

User Input

u_{i+1} (visualization)

Optimization
$u_{i+1} = u_i + \alpha s$ …

f,g,H

u_{i+1}

Coordinate Model
(perform transformations)

g,H

s

u_{i+1}

f,g,H

Linear Algebra

PETSc

GA

Model Factory (instantiate model)

MPQC

NWChem

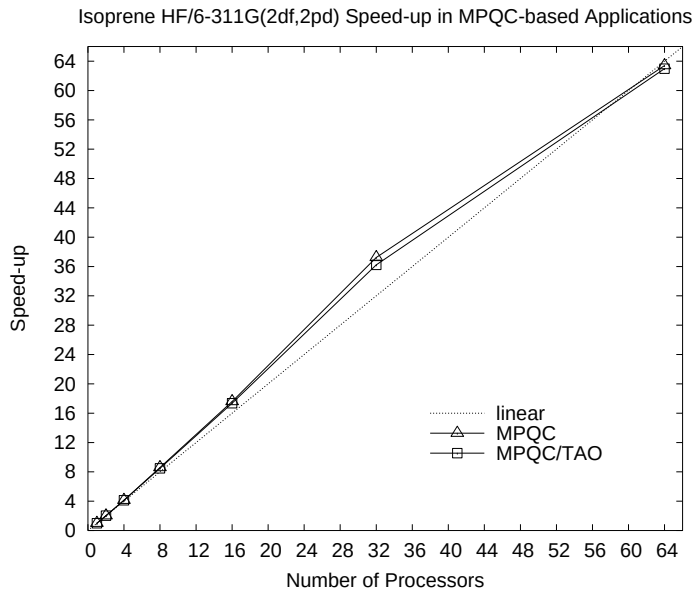
f(**u**) energy
u Cartesian coordinates
u internal coordinates
g gradient (in Cartesians)
g gradient (in internals)
H Hessian (in Cartesians)
H Hessian (in internals)
s update (in internals)

Preliminary Performance Evaluation

Comparison of NWChem's internal optimizer vs. TAO for HF/6-31G level of theory

Electronic Structure Package	NWChem		NWChem	
Optimizer	NWChem (BFGS, no line searches)		TAO (LMVM)	
Number of Evaluations Required	Energy	Gradient	Energy	Gradient
Glycine (10 atoms)	33	33	19	19
Isoprene (15)	58	58	54	54
Phosphoserine (19)	101	101	69	69
Acetylsalicylic acid (21)	116	116	94	94
Cholesterol (74)	33	33	22	22

Parallel Scaling of MPQC w/ native and TAO optimizers



Global Climate Modeling

Community Climate System Model (CCSM)

- SciDAC BER project, John Drake and Robert Malone PIs
- **Goals:** Investigate model coupling and parameterization-level componentization within models

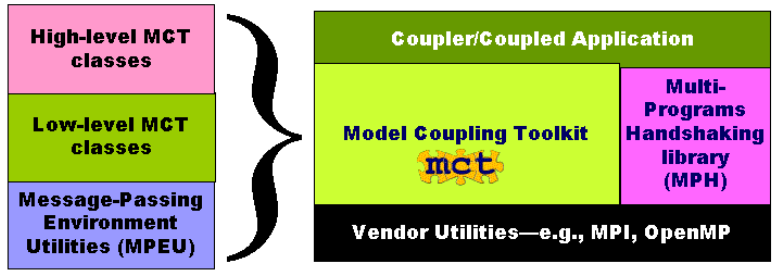
Earth System Modeling Framework

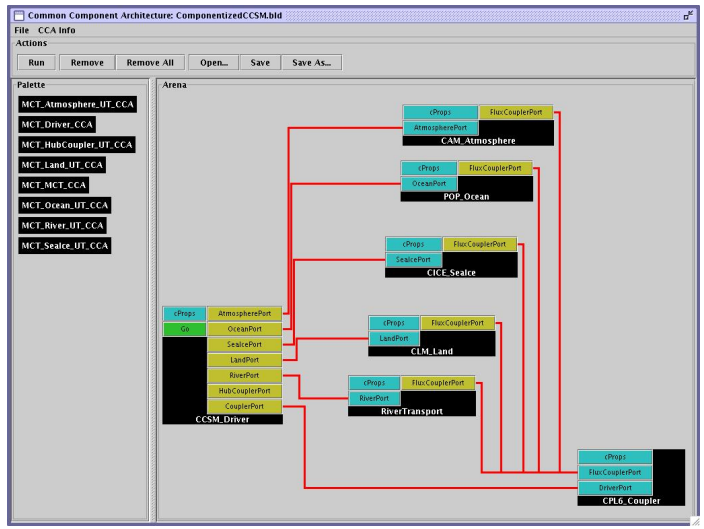
- NASA project, Tim Killeen, John Marshall, and Arlindo da Silva PIs
- **Goal:** Build domain-specific framework for the development of climate models

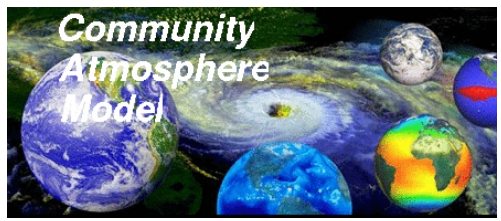
- **Investigators:** John Drake (ORNL), Wael Elwasif (ORNL), Michael Ham (ORNL), Jay Larson (ANL), Everest Ong (ANL), Nancy Collins (NCAR), Craig Rasmussen (LANL)

Community Climate System Model Componentization Activities

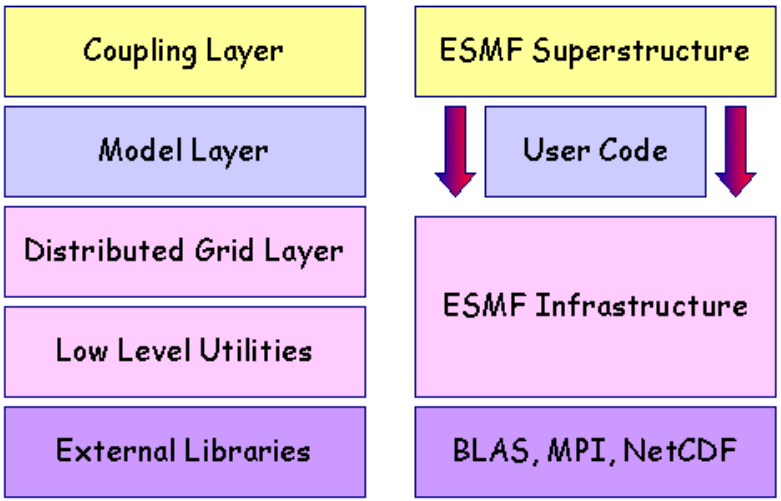
- Model Coupling Toolkit (MCT)
  - Coupler for CCSM
  - Componentization at the level of system integration (model coupling)
  - Contributions to MxN
- Community Atmosphere Model (CAM)
  - Componentization at physics/dynamics interface
- River Transport Model
  - Componentization at algorithmic level
  - Using components derived from MCT



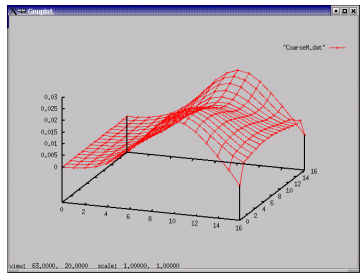




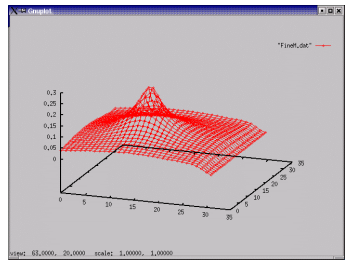
Earth System Modeling Framework



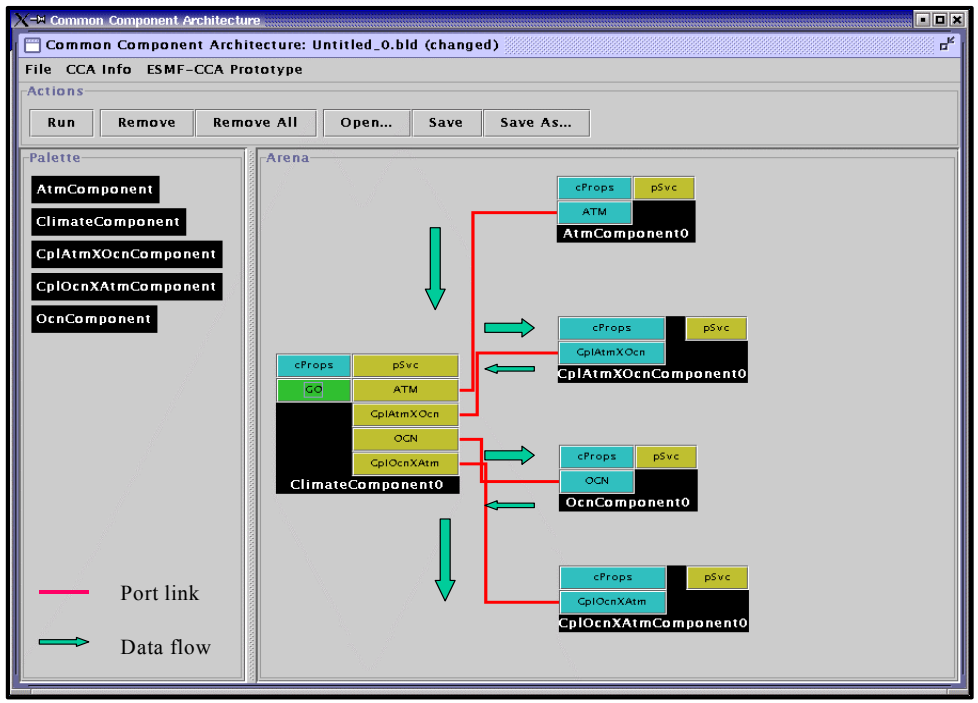
ATM



OCN



- Prototype superstructure
- Investigating grid layer interfaces



Courtesy Shujia Zhou, NASA Goddard

Summary

- CCA is a tool to help manage software complexity, increase productivity
- Under active development, with many exciting capabilities in store
- Stable and robust enough for use in applications
- CCA is allowing users to focus on doing their science
  - e.g. CFRFS publishing science results obtained with CCA-based applications

Information Pointers & Acknowledgements

- On the web: http://www.cca-forum.org
  - Mailing lists, meeting information, tutorial presentations, software, etc.
- Email us: cca-pi@cca-forum.org
- Talk to us here
  - Rob Armstrong (Lead PI)
  - David Bernholdt, Dennis Gannon, Gary Kumfert, Steven Parker (co-PIs)
  - Jaideep Ray (lead developer of CFRFS component appl.)
- Get your feet wet!
- Come to "Coding Camps"
- Thanks to the *many* people who have contributed to the development and use of the CCA, who's work this talk represents!

Oak Ridge National Laboratory is managed by UT-Battelle, LLC for the US Dept. of Energy under contract DE-AC-05-00OR22725