Common Component Architecture and the CCA Working Group

http://www.cca-forum.org

Rob Armstrong
rob@sandia.gov

CCA

Common Component Architecture

Bringing Component Technology
to High Performance Software…

Sandia National Laboratories

Pacific Northwest National Laboratory
Operated by Battelle for the U.S. Department of Energy

Los Alamos NATIONAL LABORATORY

ARGONNE NATIONAL LABORATORY • UNIVERSITY OF CHICAGO •

DEPARTMENT OF ENERGY • UNITED STATES OF AMERICA •

Center for
Component **T**echnology in
Terascale
Simulation **S**oftware
(CCTTSS)

THE UNIVERSITY OF UTAH

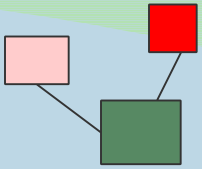
INDIANENSIS UNIVERSITATIS SIGILLUM LUX ET VERITAS MDCCCXX

OAK RIDGE NATIONAL LABORATORY

Lawrence Livermore National Laboratory

http://www.cca-forum.org

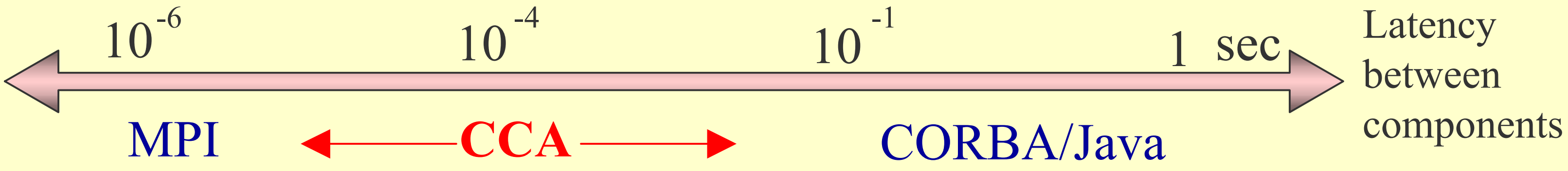
CCA Motivation



Desire to build Science Applications by hooking together drag-N-drop components.

DOE Common Component Architecture provides a mechanism for interoperability of high performance components developed by many different groups in different languages or frameworks.

Existing Component Architecture Standards such as CORBA, Java Beans, and COM do not provide support for parallel components.

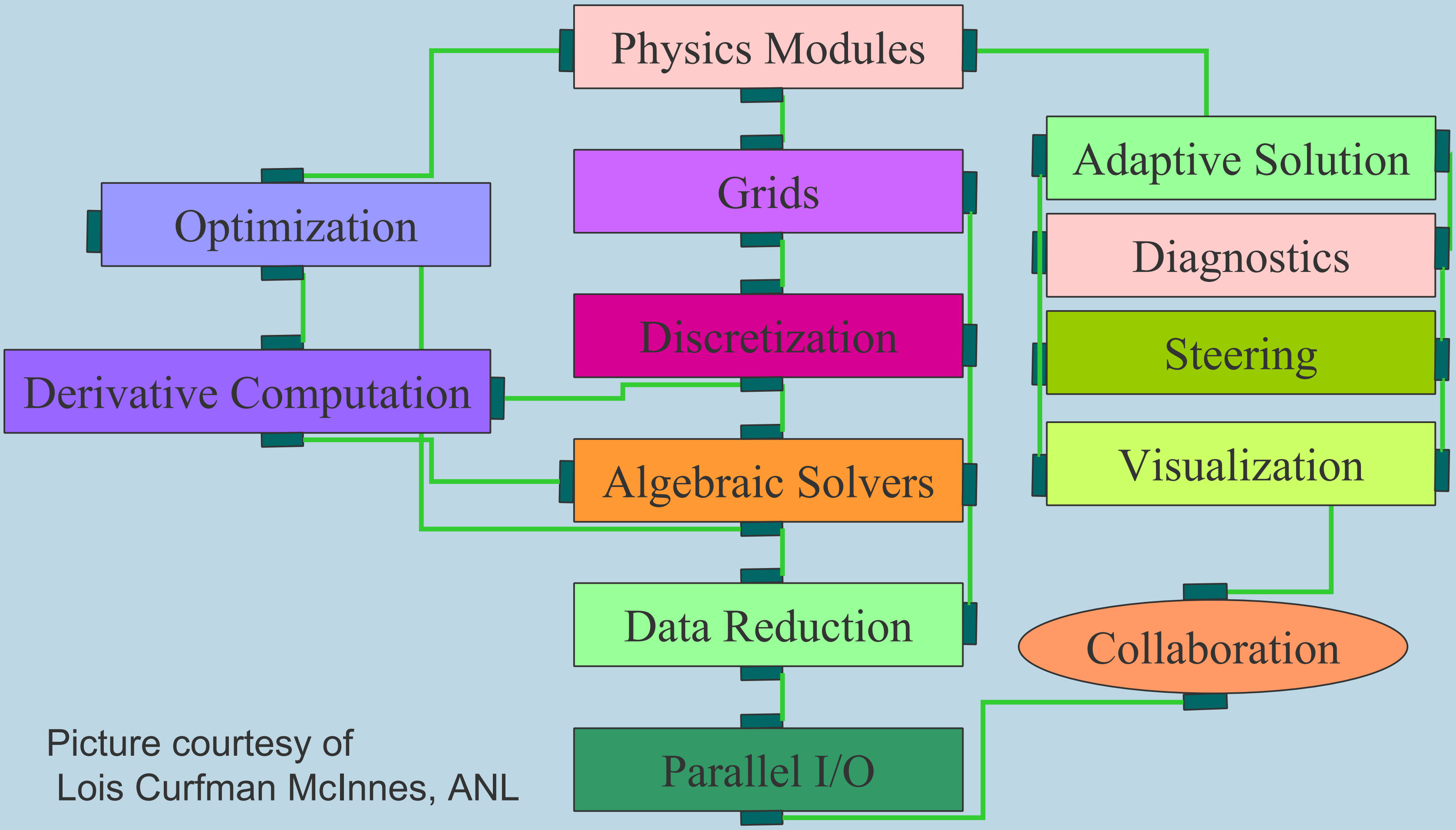


10^{-6} 10^{-4} 10^{-1} 1 sec

Latency between components

MPI ← CCA → CORBA/Java

21st Century Application



Picture courtesy of
Lois Curfman McInnes, ANL

Who are we?

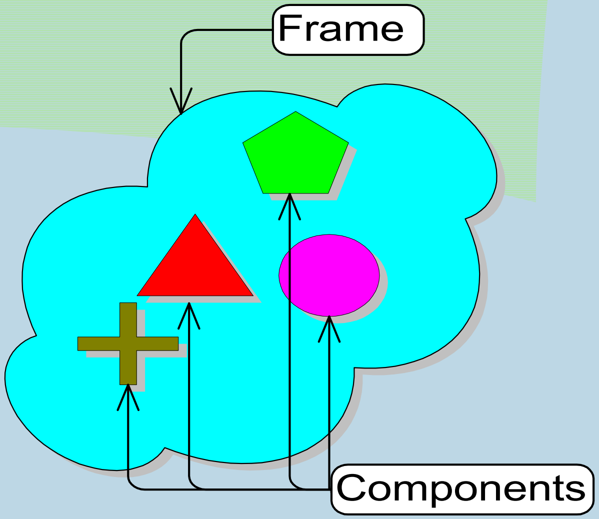
- Researchers in the HP components field that are dedicated to forming an open standard for HP components.
  - addressing the concerns of HPC.
- Originated from DOE's DOE2K program
  - has its place in ASCI: Software Integration Curve.
  - participation from universities.

CCA Active Participants *

- Dennis Gannon (Indiana U)
- Randall Bramley (Cal Tech)
- Gary Kumfert, Scott Kohn, Tom Epperly (Livermore)
- Craig Rasmussen, Kate Keahey (Los Alamos)
- Rob Armstrong, Ben Allan, Jaideep Ray (Sandia)
- David Bernholdt, Jim Kohl (Oak Ridge)
- Lois McInnes, Paul Hovland, Lori Freitag (Argonne)
- Steve Parker (U Utah)
- Jarek Nieplocha (Pacific Northwest Labs)
- Robert Clay, Lee Taylor (Terascale)

*see http://www.cca-forum.org/mailman/listinfo/cca-forum to subscribe.

What we are doing



- The problem
  - Many incompatible frameworks.
  - Much replication of s/w effort.
  - Brave new world of network grids presents many opportunities & challenges to interoperability.
- The solution
  - A component arch. enabling interoperability between modules for scientific computing.
  - Not defining a new framework, but a standard that will work within many frameworks.

The Requirements for HP Component Architecture

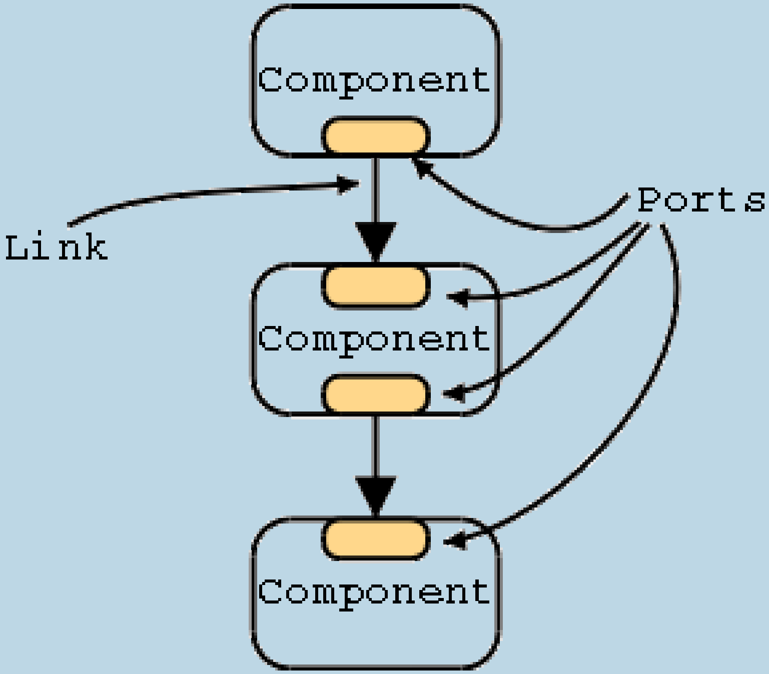
- Simple/Flexible
  - to adopt
  - to understand
  - to use
- Support a composition mechanism that does not impede high-performance component interactions.
- Permits the SPMD paradigm in component form.
- Meant to live with and rely on **other** commodity component frameworks to provide services ...
  - (e.g. JavaBeans, CCM, ...).

CCA is a component architecture: doesn't dictate frameworks, runtime.

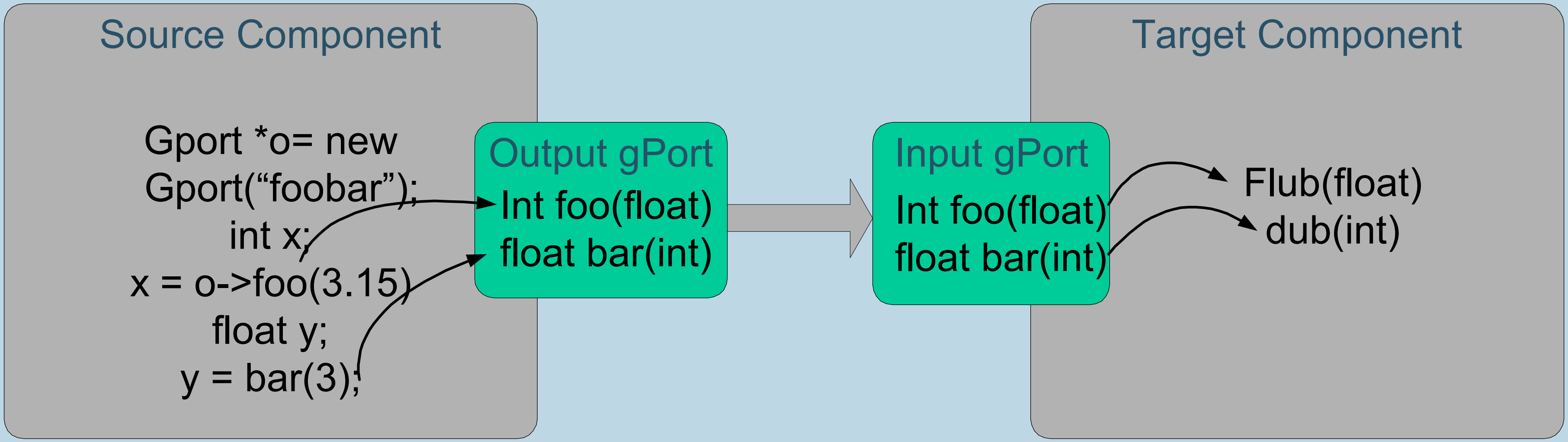
- Create components that are usable under a variety of frameworks.
- We are ***not*** making a new framework implementation or specification.
  - provide a means for discovering interfaces, similar to COM *Iunknown.*
  - specifically exclude *how* the components are linked, that is the job of a framework.
  - provide language-independent means for creating components from existing and future languages.

What have we come up with?

- ❑ Scientific IDL spec.
  - provides language interoperability only - not part of the component arch. per se.
  - introspection.
- ❑ Metaphor from visual programming (e.g. AVS)
  - instead of data-flow, ports are linked with interfaces
  - Provides/Uses design pattern
  - 0.5 CCA specification: http://www.cca-forum.org



Component Connection Mechanism



Port model hooks up an interface from one component to another.

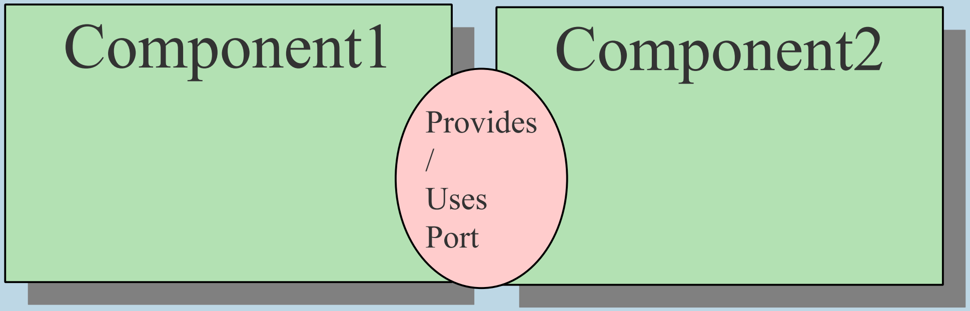
Data Flow Analogue:

Input Port == *provides* port (is provided for use)

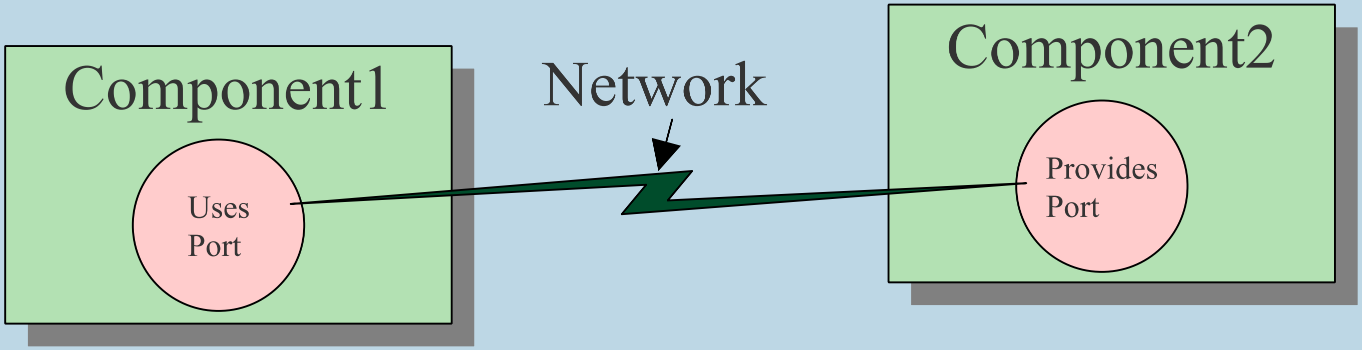
Output Port == *uses* port (uses what is provided)

Ports preserve the high-perf. of direct connections plus versatility of distributed object systems

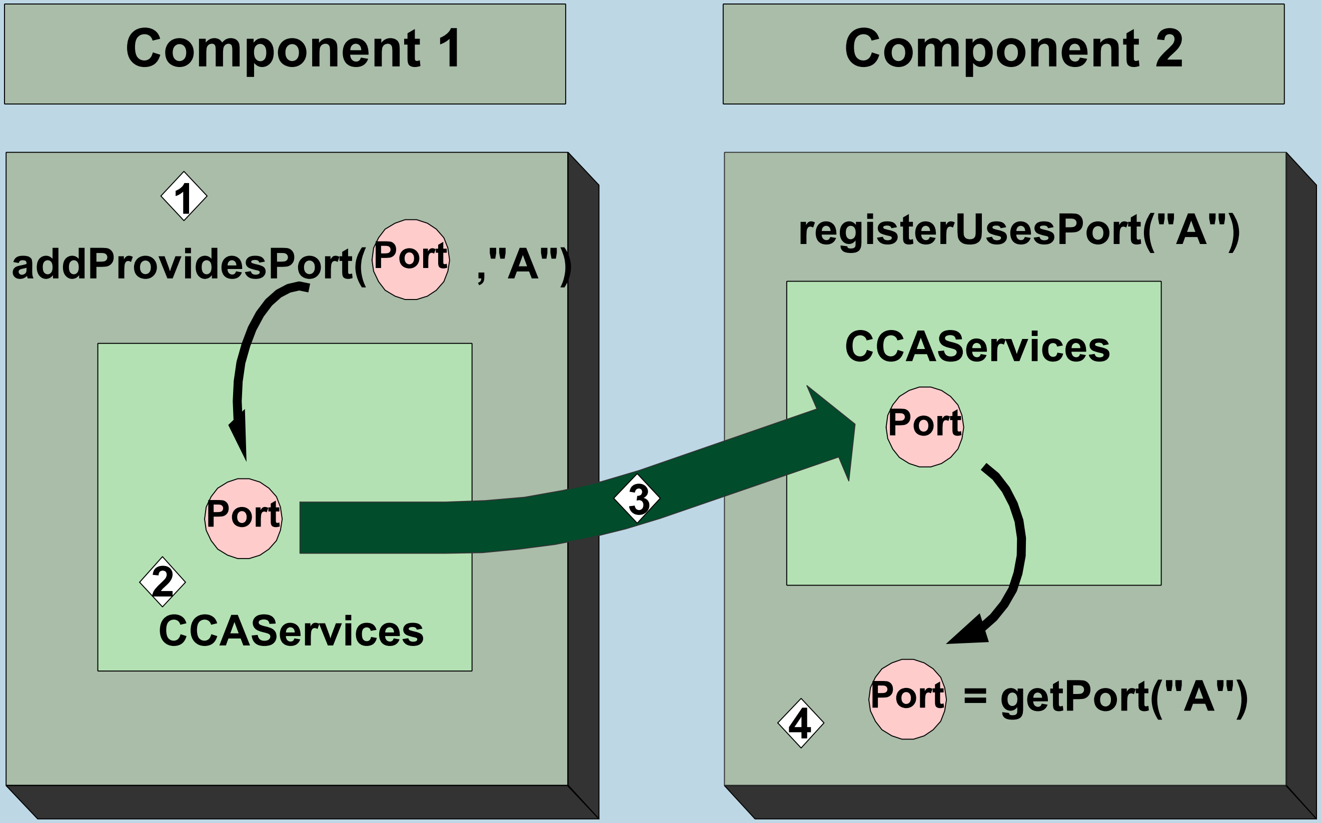
- Allows for directly connected interfaces: the next component is a few function calls away (w/ SIDL).



- Adapters will create network-distributed objects out of the same components without altering them.

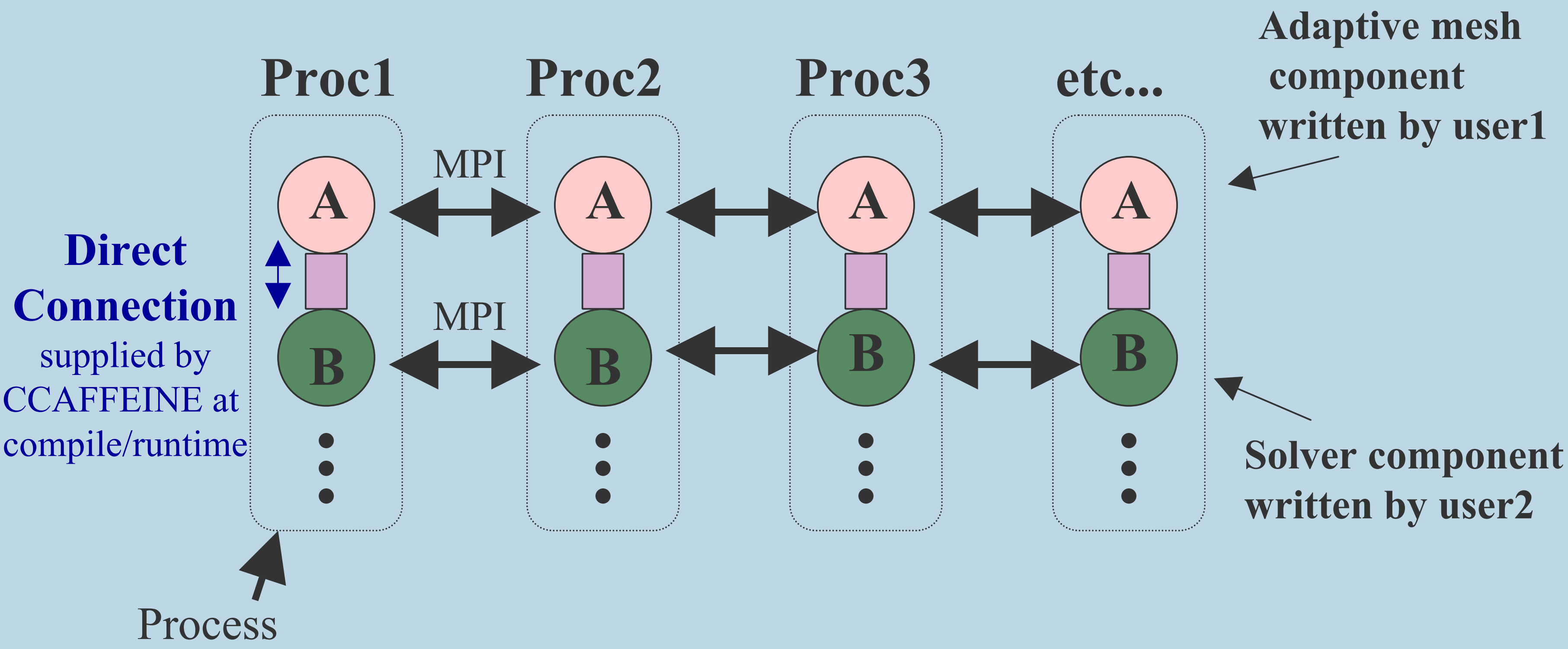


Generalized ports and the provides/uses design pattern for coupling components

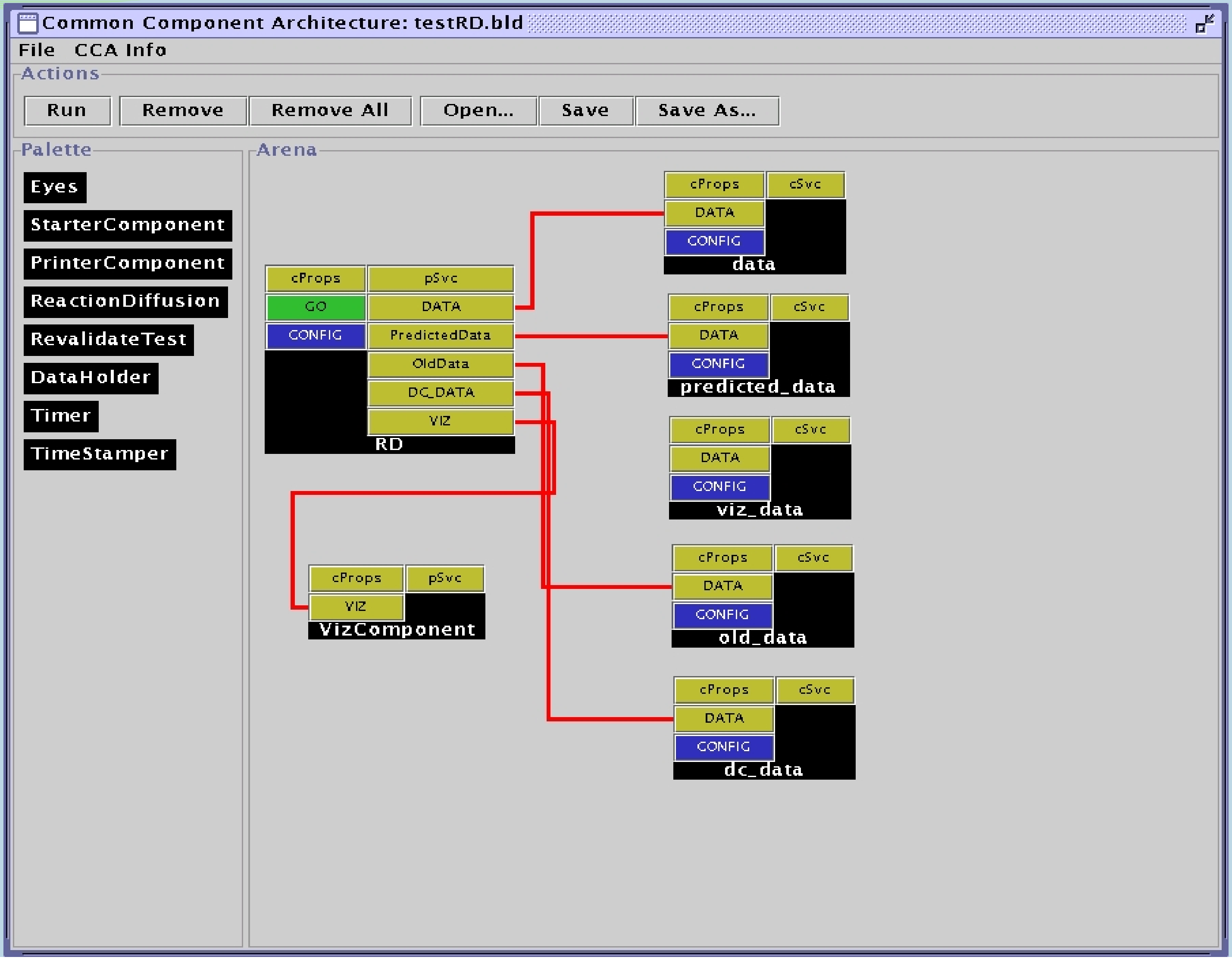


SCMD (SPMD HPC) components

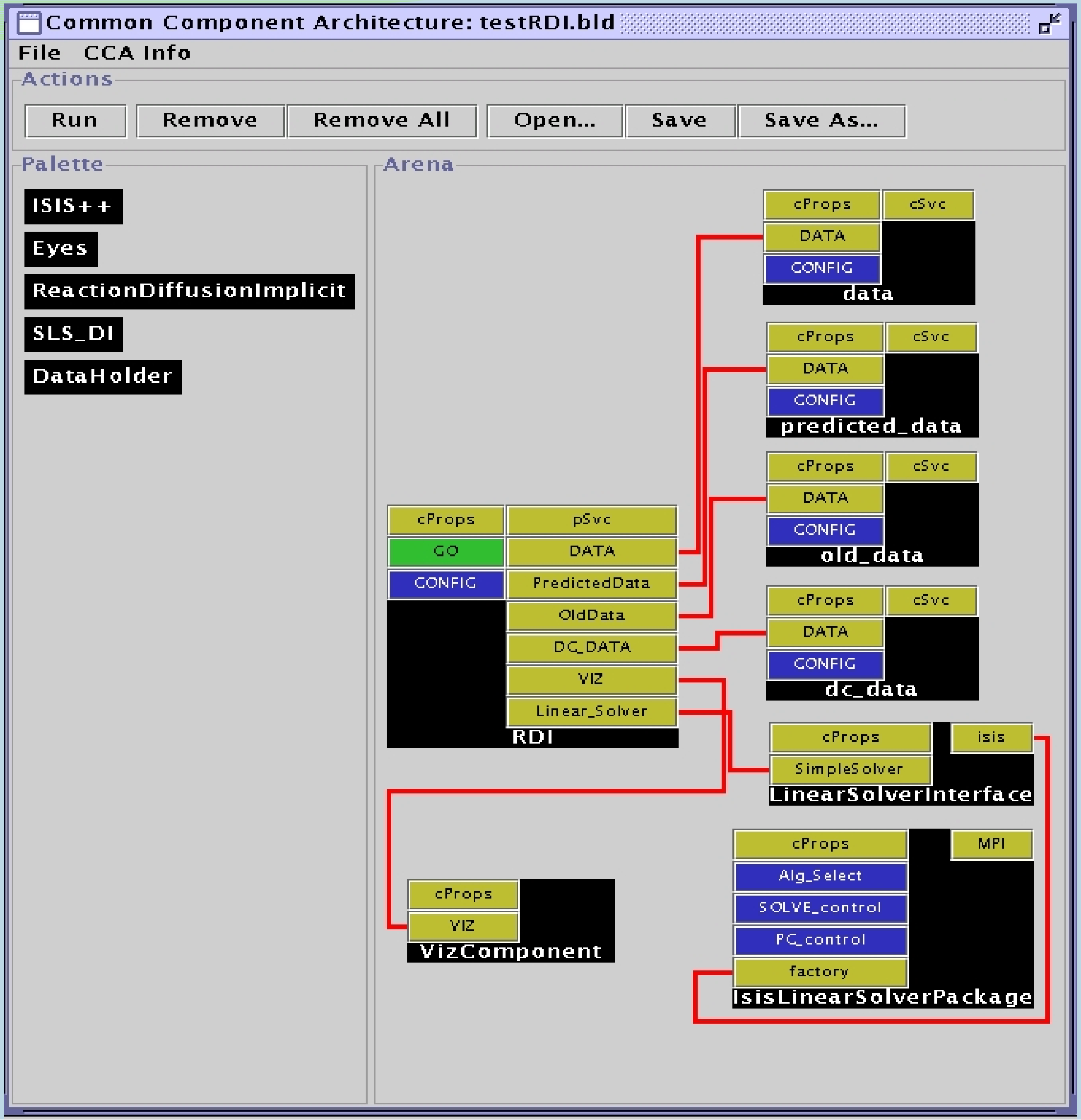
MPI application using CCA for interaction between components A and B within the same address space


Proc1
Proc2
Proc3
etc...
MPI
MPI
A
A
A
A
B
B
B
B
Adaptive mesh component written by user1
Direct Connection
supplied by CCAFFEINE at compile/runtime
Solver component written by user2
Process

A Simple example app



A little less simple



Component Technology for Terascale Software Simulation:

Bringing Components to High Performance Computing

http://www.cca-forum.org/ccttss

CCA

Common Component Architecture

Participants:

ANL

Indiana Univ.

LANL

LLNL

ORNL

PNNL

SNL

Univ. of Utah

Partners:

Climate Modeling

Quantum Chemistry

Combustion Modeling

Currently DOE Apps Are:

- One-off, stove-pipe, few participants

But Also:

- Scalable, and low latency
- Large & rich legacy investment

→

Challenges

- Practical parallel HPC component model
- Integrate legacy software investment
- Connection to Grid components
- Multi-language

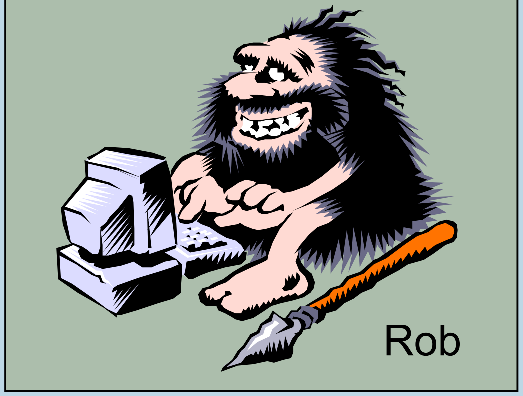
→

Goals

- Enable plug-and-play parallel simulations
- Establish component "marketplace" with applications partners
- Extend commodity component technology for HPC

Today:

Single-purpose, monolithic, tightly-coupled parallel codes



HPC Component Framework

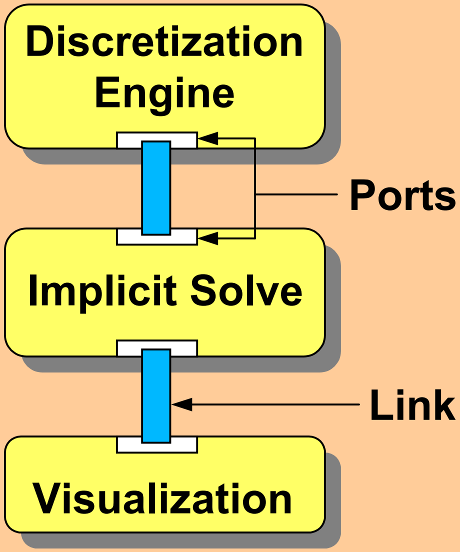
Scientific Components

CCA →

"MxN" Parallel Data Redistribution

Applications Integration

Component-Based Scientific Application


Discretization Engine
Ports
Implicit Solve
Link
Visualization