

The Common Component Architecture: Status and Plans

David E. Bernholdt
***CCA Coordinator for Application
Integration***

bernholdtde@ornl.gov

<http://www.csm.ornl.gov/~bernhold/>

Why CCA?

- Commodity component models are very useful/popular – “revolutionizing software”
 - CORBA, COM/DCOM, EJB, etc.
- These models have no concept of parallel computing, and are rather heavyweight (performance penalties even on same machine)
- The Common Component Architecture is intended to provide a component model suitable for high-performance simulation applications
 - Preserve HPC performance
 - Bring benefits of component-based software development to HPC

What is the CCA Forum?

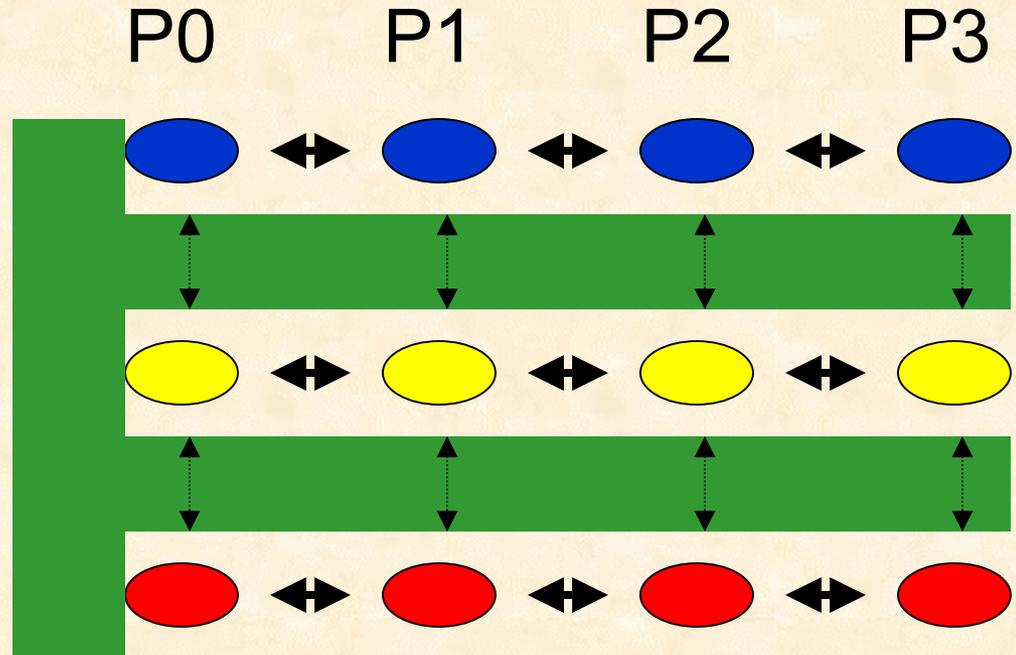
- The primary goal of CCA is *specifications*. Implementations secondary.
- The CCA Forum is the “standards body” for CCA
- Began January 1998 as an informal/unofficial outgrowth of DOE’s ACTS program
 - (Active) membership primarily DOE and two universities, but all welcome
- Holds quarterly face-to-face meetings, operates a mailing list, web site (soon), etc.
- Online voting (Quorum server, developed by LLNL) used to manage proposals
- Must attend m of the last n meetings to be eligible to vote

CCA and SciDAC

- To date CCA has not been funded as a separate project
- DOE SciDAC program is a tailor-made opportunity.
- Proposal: “Center for Component Technology for Terascale Simulation Software”
- Indiana U, ANL, LANL, LLNL, ORNL, PNNL, SNL, U Utah
- More than 20 projects/proposals plan to use CCA
- CCA proposal will include applications integration work in computational chemistry and climate (technology “push”)
- Work with other projects wishing to adopt (technology “pull”)

What is CCA: Technical

- Supports parallel and distributed computing
- Minimalist standard
- Meant to work with commodity component models
- Currently have draft specifications (incomplete) and early prototypes
- Running parallel reaction-diffusion+visualization demonstration (SC00)



Component Interfaces

- *Ports* are interface definitions. Component writers write to port specifications
- Components *provide* or *use* ports. If you *provide* a port, that means you implement it
- In current parallel framework (CCAFFEINE) “getting” a port means getting a pointer to the function lookup table for the component providing the port
- Calls between components are equivalent to a C++ virtual function call

Future Work

- Unify parallel and distributed frameworks
- Flesh out framework specification (esp. framework services)
- Language interoperability (incl F77, F90)
- Develop a “suite” of components to bootstrap adopters (numerical, data objects, parallel programming models, etc.)
- MxN data redistribution (MCT taken to extremes)