Common Component Architecture

http://www.tascs-scidac.org
http://www.cca-forum.org/dev

Kosta Damevski

Department of Mathematics and Computer Science
Virginia State University

July 27th, 2009

Overview of the Common Component Architecture (CCA)

- **Scalable Software Engineering for HPC Applications**
- Problem:
  - Program size and complexity in many projects tends to grow - making the code difficult to: design, grow, update, debug.
  - Reusing existing code should be easy, even when using different different languages.
- CCA Solution:
  - Applications are built from a set of reusable software components.
  - Components can be written (or created from existing code bases) in a variety of languages and platforms.
  - Built for the HPC community: usually small performance overhead, support for parallel components.

A Brief Overview of Software Components

- Each component has a set of ports (interfaces).
- Ports are either *uses* or *provides*, depending whether they use or implement some functionality.
- Components are assembled into an application by connecting matching uses and provides ports.

CCA is for the HPC Community

- Software components are available across many computing domains, and in a varety of flavors (e.g. CORBA, COM, Enterprise Java Beans, Spring etc.).
- *Most of these don't fit the needs of the HPC community*
- The CCA provides:
  - low overhead
  - multidimensional array and complex number support
  - parallel components
  - support for several languages (C++, C, Fortran77, Fortran90, Java).

Example of CCA Use SciDAC ITAPS project (unstructured meshes)

- The ITAPS team developed common interfaces for several aspects of their meshes.
- Rather than making invasive changes to use different meshing libraries, developers of scientific applications can code to the common interface and seamlessly substitute any of the eight different libraries that implement the interface without changes to their application.
- Enables experimentation with different meshes.

Example of CCA Use SciDAC ITAPS project (unstructured meshes)

- The ITAPS team developed common interfaces for several aspects of their meshes.
- Rather than making invasive changes to use different meshing libraries, developers of scientific applications can code to the common interface and seamlessly substitute any of the eight different libraries that implement the interface without changes to their application.
- Enables experimentation with different meshes.

Example of CCA Use SciDAC ITAPS project (unstructured meshes)

- The ITAPS team developed common interfaces for several aspects of their meshes.
- Rather than making invasive changes to use different meshing libraries, developers of scientific applications can code to the common interface and seamlessly substitute any of the eight different libraries that implement the interface without changes to their application.
- Enables experimentation with different meshes.

How This Workshop May Help the CCA

- Scaling to very large applications containing large numbers of components.
- Improving support for parallel components and parallel coupling (MPMD - in our case, it's MCMD)

How This Workshop May Help the CCA

- Scaling to very large applications containing large numbers of components.
- Improving support for parallel components and parallel coupling (MPMD - in our case, it's MCMD)

Shameless Self-Promotion...

- The CCA project has been around for over 10 years, so many of the tools are relatively mature.
- A number of applications use the CCA (20?). Give it a try...
- Webpages again: http://tascs-scidac.org and http://www.cca-forum.org/dev