The SciDAC Center for Technology for Advanced Scientific Component Software(TASCS)

Lead PI: David Bernholdt
Oak Ridge National Lab, *bernholdtde@ornl.gov*

Institution	Institutional Lead PI	Institution	Institutional Lead PI
ANL	Lois McInnes	PNNL	Jarek Nieplocha
Binghamton U	Madhu Govnidaraju	SNL	Rob Armstrong
Indiana U	Randy Bramley	Tech-X	Sveta Shasharina
LANL	Craig Rasmussen	U Maryland	Alan Sussman
LLNL	Gary Kumfert	U Utah	Steve Parker
ORNL	Jim Kohl		

General contact point (PIs and technical leads):
tascs-leads@cca-forum.org

TASCS Overview

- CCTTSS enabled us to bring the CCA from the idea stage to the prototype stage
  - Core ideas well-developed
  - Useful implementations, but not polished
  - Numerous users, but not “standard”
- TASCS will bring the CCA from prototype to production for general computational science users
  - Strong emphasis on robustness of tools
  - Strong emphasis on usability of components and component environment
  - Strong emphasis on development of an ecosystem (component toolkit) around CCA
  - Research activities focus on how to exploit componentness to provide more value for software developers
  - Continue outreach and support activities

Current Status

- Proposal was submitted on 3 March 2006
  - Thanks to everyone who contributed to the development and writing of the proposal!
- Reviewed well
- Fred Johnson requested a revised budget and scope of work in early June
  - Funding is likely
- Nothing is official until DOE (Ray Orbach) makes the announcements
  - Had been expected “any day now”
  - Latest news: late September

TASCS Project Focus Areas

Organized around four main focus areas:

- Component Technology Initiatives
  - Lois McInnes, ANL
- CCA Environment
  - Gary Kumfert, LLNL
- CCA Toolkit
  - Rob Armstrong, SNL
- User Outreach and Application Support
  - David Bernholdt, ORNL

TASCS Project Focus Areas

Organized around four main focus areas:

- Component Technology Initiatives
  - Lois McInnes, ANL
- CCA Environment
  - Gary Kumfert, LLNL
- CCA Toolkit
  - Rob Armstrong, SNL
- User Outreach and Application Support
  - David Bernholdt, ORNL

Component Technology Initiatives

How can we exploit the component environment to provide computational scientists with better ways to develop their software?

- Computational Quality of Service
  - Lois McInnes, ANL
- Software Quality and Verification
  - Tammy Dahlgren, LLNL
- Support for Emerging HPC Hardware and Software Paradigms
  - Jarek Nieplocha, PNNL

CTI: Computational Quality of Service

- Adaptation of running component applications in response to changing conditions
  - Performance, accuracy, mathematical consistency, reliability, etc.
  - Adapt by changing components on the fly
- Planned activities
  - Develop common software infrastructure for adaptation
    - Analysis (when to change), control (what to change)
  - Test with various applications
    - Many possibilities, depending on funding (still largely unknown)
  - Strong connection to Software Quality initiative

CTI: Software Quality and Verification

- Extend component interface definitions with semantic annotations which can be verified/enforced at composition or run time (contracts)
  - Developers provide “executable documentation” about correct use
  - Users protected against improper usage due to code error, unforeseen circumstances, etc.
  - Can facilitate verification that the code is actually doing what it was intended to do
- Planned activities
  - Emphasize annotations needed to support CQoS work
  - Method invocation sequencing constraints
  - Implement and demonstrate annotations in selected Toolkit components

CTI: Support for Emerging HPC Hardware and Software Paradigms

- Investigate how component environments can be used to enhance the user's experience programming for new and emerging high-end systems
  - Leadership Class, DOE petaflop systems at ORNL and ANL(?), NSF petaflop systems, HPCS systems, etc.
- Two key trends apparent
  - Massive levels of parallelism (10^5-10^6+ concurrent threads)
  - Heterogeneous processor environments
    - FPGAs, scalar+vector+threaded, co-processors, etc.
- Planned activities
  - Develop and implement management for progress groups and (multi-level) MCMD parallelism
  - Develop techniques to support asynchronous interactions of components representing co-processors/accelerators with those running on the traditional processors
  - Applications in quantum chemistry and comp. biology

TASCS Project Focus Areas

Organized around four main focus areas:

- Component Technology Initiatives
  - Lois McInnes, ANL
- CCA Environment
  - Gary Kumfert, LLNL
- CCA Toolkit
  - Rob Armstrong, SNL
- User Outreach and Application Support
  - David Bernholdt, ORNL

CCA Environment

Supporting and improving the foundation of the CCA environment

- Core Tool Support and Maintenance
  - Ben Allan, SNL
- Enhancements
  - Tom Epperly, LLNL
- Usability
  - Craig Rasmussen, LANL

Env: Core Tool Support and Maintenance

- Keeping the core software tools running in the face of change
  - Computer hardware, operating systems, compilers and other environmental concerns
- Planned activities
  - Porting to hew hardware (especially high-end)
  - Updating for changes in surrounding software environment
  - Provide “help desk” support for core tools
  - Develop and deploy conformance test for CCA specification
  - Reference and developer documentation for core tools

Env: Enhancements

- Extending the CCA environment (specification, core tools) with additional features/capabilities required by customers and other activities in TASCS
- Planned activities
  - CCA spec additions
    - i.e. EventService, MPIService, CommandLineService
  - Interoperability between Kepler workflow environment and CCA (collaboration with SDM)
  - SIDL/Babel enhancements
    - F2003 support (collaboration w/ Tech-X SBIR)
    - Struct support
    - Typemaps (in the SWIG sense, not the CCA sense)

Env: Usability

- Making HPC component technology more accessible to users
- Planned activities
  - CCA Lite
    - Tiered approach to CCA
    - Limited language interop in exchange for simpler C-based environment
    - Tools to facilitate upgrade of Lite cmpts to full CCA
  - Component Debugging and Testing
    - Document tools & techniques for multilanguage debugging
    - Testing harnesses (i.e. unit testing, port monitors)

TASCS Project Focus Areas

Organized around four main focus areas:

- Component Technology Initiatives
  - Lois McInnes, ANL
- CCA Environment
  - Gary Kumfert, LLNL
- CCA Toolkit
  - Rob Armstrong, SNL
- User Outreach and Application Support
  - David Bernholdt, ORNL

CCA Toolkit

- Making it easier to create components, and making available a suite of real, useful components
- Planned activities
  - Development Tools and CCA Base Installation
    - "Standard" structure and scaffolding for components and collections of components (derived from tutorial)
    - Commandline and (Eclipse) IDE tools to facilitate creating components
  - CCA Component Collection
    - Linear and nonlinear solver components; Structured and unstructured mesh data models; General purpose utility components; Parallel coupling components
  - Community interface development
  - CCA Component Repository

TASCS Project Focus Areas

Organized around four main focus areas:

- Component Technology Initiatives
  - Lois McInnes, ANL
- CCA Environment
  - Gary Kumfert, LLNL
- CCA Toolkit
  - Rob Armstrong, SNL
- User Outreach and Application Support
  - David Bernholdt, ORNL

User Outreach and Application Support

- Broaden awareness and adoption of component technology and the CCA
- Planned activities
  - Application support
  - User outreach and support
    - Tutorials, coding camps, etc
  - Community outreach

Changes from Proposal to Final Budget

- Expected award is ~69% of request
- Generally scale back deliverables to match
  - Drop selected deliverables (mostly later or more speculative)
  - Stretch out schedule of remaining deliverables
  - Some specific changes, highlighted in following slides

- Important reminder:
  - Deliverables are things we've promised we'll do for the money we're getting
  - If we can't do them, we need to have a good explanation, and present an equally "valuable" alternative
  - But dropping something as a deliverable doesn't prohibit us from working on it!
    - As long as we can satisfy the deliverables we did promise

Changes in Component Technology Initiatives

- Computational Quality of Service
  - Fewer external collaborators, reduced breadth of topics
- Parallel Coupling Infrastructure
  - Strongly dependent on external proposals
  - Concern about what could be delivered, given TASCS reductions and external uncertainties
  - **Eliminate PCI as an initiative**, shift PCI **effort** into Toolkit
    - Emphasize development of coupling-related **components**
- Software Quality and Verification
  - Drop more speculative aspects
  - Increase focus on connections with CQoS
- Support for Emerging HPC…
  - Eliminate fault-tolerance related activities

Changes in Other Focus Areas

- Environment
  - Core Tools and Usability (including CCA Lite) suffered less than other elements
  - Enhancements will focus more narrowly on the needs of the DOE user base
    - Drop Babel Matlab binding
    - Retain CCA-Kepler interoperability work (in collaboration with SDM), but drop others (ESMF, VTK, Cactus)
- Toolkit
  - Suffered less than other elements
  - Some PCI-related deliverables (and effort) added
    - This is ***not*** the PCI initiative reincarnated!
- Outreach
  - Less “walkup” support

TASCS Requires a Different Approach

- In CCTTSS, we were strongly curiosity-driven
- Our work in CCTTSS has allowed us to map out a path forward to make CCA part of the standard toolbox of computational science
- TASCS is a building phase for CCA – the roadmap (proposal) shows us what need to build, we just have to make it happen!
  - More emphasis on delivering specifications, code (components), documentation, etc. in smaller chunks and at shorter intervals than in the past
- Much important research, but more focused – we already have a pretty good idea where we need to go in many areas

Thinking Ahead: After TASCS

- Just as with CCTTSS, TASCS serves two purposes
  - Continued R&D on HPC components
  - Putting CCA in the best position to carry on *after* the end of the project
- My working assumption is that after 10 years of continuous funding (2011+), another large-scale 3-5 year CCA proposal is unlikely to succeed
  - More likely that smaller spin-off proposals will succeed
- Therefore, TASCS needs to…
  - Insure that we have a robust, well-tested, easily maintained, and "self-sustaining" set of core tools to allow spin-offs to focus on research topics
  - Insure that we have a strong impact on users so that they *demand* continued support for CCA

Time to Get Started!

- More detailed kick-off discussions next
  - These discussions meant to be organizational, *not technical*
  - Should spur follow-on technical discussions afterwards
- Can't afford to wait until next quarter to get started
  - We have a lot of important work ahead of us!
- Will begin presenting brief updates on TASCS at the quarterly CCA Forum meetings
  - More awareness of ongoing activities, not just things that are ready for separate talks, papers, etc.
  - Encourage involvement in/leverage of activities across project and outside