CCA
Common Component Architecture
http://www.cca-forum.org

PERI

Computational Quality of Service (CQoS) in Quantum Chemistry

Joseph Kenny[1], Kevin Huck[2], Li Li[3], Lois Curfman McInnes[3], Heather Netzloff[4], Boyana Norris[3], Meng-Shiou Wu[4], Alexander Gaenko[4] , and Hirotoshi Mori[5]

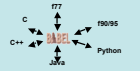
[1]Sandia National Laboratories, [2]University of Oregon, [3]Argonne National Laboratory, [4]Ames Laboratory, [5]Ochanomizu University, Japan

This work is a collaboration among participants in the SciDAC Center for Technology for Advanced Scientific Component Software (TASCS), Performance Engineering Research Institute (PERI), Quantum Chemistry Science Application Partnership (QCSAP), and the Tuning and Analysis Utilities (TAU) group at the University of Oregon.

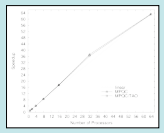
Quantum Chemistry and the Common Component Architecture (CCA)

CCA Overview:

- The CCA Forum provides a specification and software tools for the development of **high-performance components**.
- Components = Composition
  - A component is a unit of software deployment/reuse
  - Components interact through standard interfaces – no restrictions on implementation (language, parallel model, etc.)
  - A component architecture specifies a *framework* for composition of units into applications
- Key CCA benefits:
  - Programming language interoperability via Scientific Interface Definition Language (SIDL) and Babel



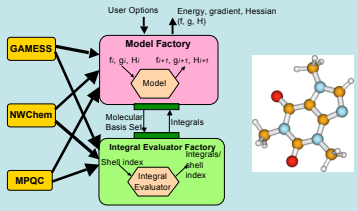
  - Common component interfaces
  - Dynamic composability
- Facilitates collaboration among domain scientists, mathematicians, and computer scientists



Isoprene HF/6-311G(2df,2pd) parallel speedup in MPQC-based CCA simulations using components from the Toolkit for Advanced Optimization (TAO), a math library in the SciDAC TOPS project.

Quantum Chemistry Scientific Application Partnership:

- Construct interoperating mechanism among several leading high-performance QC codes (**NWChem**, **GAMESS**, and **MPQC**) through CCA infrastructure
- **CCA_chem_generic** package defines several interfaces for QC calculations
  - Molecule, Model, GaussianBasis, IntegralEvaluator



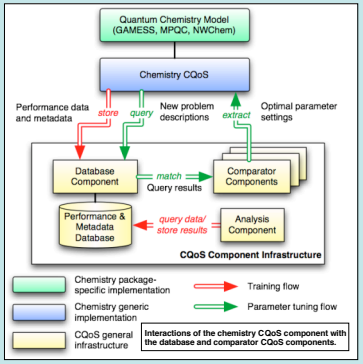
Every package implements these interfaces to create its own components … Facilitates sharing capabilities among chemists and the wider scientific community.

CQoS in Quantum Chemistry: Motivation and Approach

Motivation:

- **QCSAP Challenges:** How, during runtime, can we make the best choices for reliability, accuracy, and performance of interoperable quantum chemistry components based on NWChem, MPQC, and GAMESS?
  - When several QC components provide the same functionality, what criteria should be employed to select one implementation for a particular application instance and computational environment?

  - How do we incorporate the most appropriate externally developed components (e.g., which algorithms to employ from numerical optimization components)?
- **Initial Focus:** Parallel application configuration of QC applications so that these can run effectively on various high-performance machines
  - Eliminate guesswork or trial-and-error configuration
- **Future Work:** More sophisticated analysis to configure algorithmic parameters for particular molecular targets, calculation approaches, and hardware environments

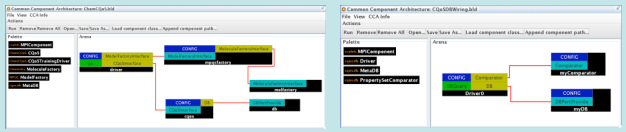


CQoS Approach:

- **Overall**: Develop infrastructure for dynamic component adaptivity, i.e., composing, substituting, and reconfiguring running CCA component applications in response to changing conditions
  - Performance, accuracy, mathematical consistency, reliability, etc.
- **Approach:** Develop CQoS tools for
  - **Analysis:** Performance monitoring, problem/solution characterization, and performance model building
  - **Control:** Interpretation and execution of control laws to modify an application's behavior
  - **CQoS database component:** Manage interactions with performance and runtime databases to facilitate analysis and decision making
  - **Leverage external tools** under development by PERI and others
- **Phases of work**:

CQoS Infrastructure and Preliminary Results for Quantum Chemistry

CQoS Database Usage in Quantum Chemistry:



CQoS database infrastructure provides a programmer-friendly component front-end to database implementations. These wiring diagrams made by the CCA Ccaffeine framework GUI (developed by SNL) represent component connections between ports. (*Uses* ports are gold; *provides* ports are blue.)

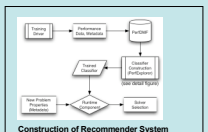
CQoS Database Usage

- **Application metadata**
  - Molecule characteristics: atom types, topology, moments of inertia
  - Algorithm parameters: tunable parameters, convergence level
- **System parameters**
  - Compilers
  - Machine info, e.g., number of nodes, threads per node, network
- **Historical performance data**
  - Execution times, iterations to convergence, etc.
  - Obtained through source instrumentation, e.g., TAU
  - Can guide configuration of related new simulations

CQoS Comparator Components

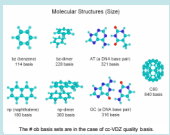
- Compare sets of parameters within the performance database
- Quantum chemistry applications can match the current application state against historical data through database queries during runtime.
- Use metadata to guide parameter selection and application configuration
  - Match molecule similarity, basis set similarity, electronic correlation approach, etc.

CQoS Analysis Using PerfExplorer and PerfDMF:



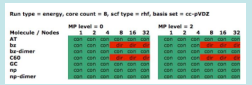
Preliminary performance evaluation of QC packages to learn effective configurations

- 7 test molecules: bz (benzene), bz-dimer, AT, np (naphthalene), np-dimer, GC, C60
- 3 run types: energy, gradient, Hessian
- 5 SCF wavefunctions: RHF, ROHF, UHF, GVB, MCSCF
- 2 MP levels: 0, 2
- 4 basis sets: cc-pVDZ, 6-31G, 6-31G*, 6-31G**
- 2 methods: direct (dir) or conventional (con)
- 672 combinations, without even considering node/core counts, machine parameters, architectures …



The # db basis sets are in the case of cc-VDZ quality basis.

Sample Results for GAMESS



basis set=cc-pVDZ, scf type=RHF, run type=energy, cores=8, vary molecule, mp level, method, and node counts
test machine: bassi.nersc.gov: IBM POWER5, 111 compute nodes with 8 cores, 32GB memory per node
Direct is faster than conventional for some molecules at higher node counts – the correlation is data dependent.

This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research and Scientific Discovery through Advanced Computing (SciDAC) initiative, Office of Science, U.S. Dept. of Energy, under Contracts DE-AC02-06CH11357, DE-AC04-94AL85000, DE-AC02-07CH11358, and DE-FG02-07ER25826.

SciDAC Scientific Discovery through Advanced Computing

AMES LABORATORY · Argonne NATIONAL LABORATORY · Sandia National Laboratories

Office of Science
U.S. DEPARTMENT OF ENERGY