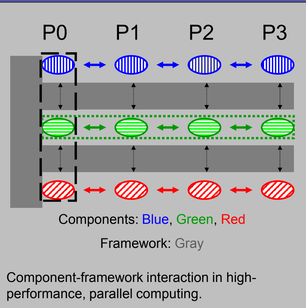
SAND2006-6513P

Component Architectures for Quantum Chemistry: Forging New Capabilities and Insights

CCA
Common Component Architecture

Joseph P. Kenny,[1] Curtis L. Janssen,[1] Ida M. B. Nielsen,[1] Manojkumar Krishnan,[2] Vidhya Gurumoorthi,[3] Edward F. Valeev,[4] Theresa L. Windus,[5]

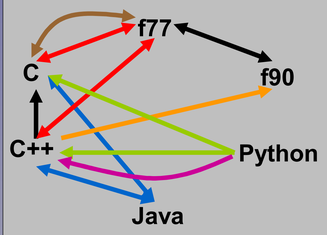
[1]Scalable Computing Research and Development, Sandia National Laboratories/CA
[2]Computational Science and Mathematics, Pacific Northwest National Laboratory
[3]Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory
[4]Department of Chemistry, Virginia Tech
[5]Department of Chemistry, Iowa State University



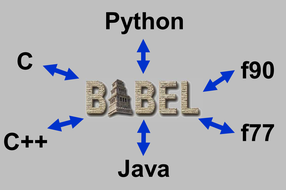
Component-framework interaction in high-performance, parallel computing.

Common Component Architecture (CCA)

- Components for high performance computing
- SPMD for distributed memory
  - Framework handles per-process component composition
  - No interference with interprocess communication – standard parallel models
- Support for distributed/grid computing (RMI) and MPMD



- Babel provides a scalable solution to language interoperability
- Generates glue code to link clients and servers in any supported language
- Supports scientific data-types (complex, array, etc) and object model

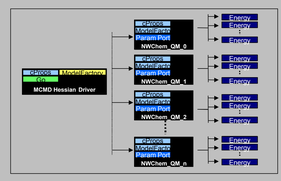


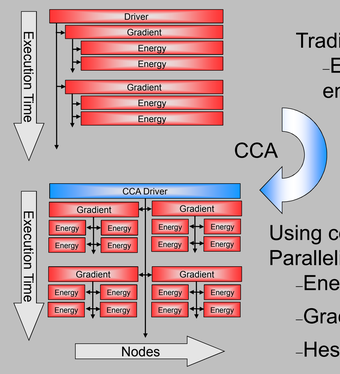
Component-based Software Engineering

- Components are objects which exist within a runtime environment or *framework* which provides services
  - Application composition
  - Performance evaluation
  - Computational quality of service
- Component approaches facilitate interface standardization and code interoperability
- Inserting "glue code" at component boundaries solves language interoperability (Babel)

Managing Multilevel Parallelism for Improved Machine Utilization

- Combining SPMD and MPMD Paradigms – MultiComponent Multiple Data
- MCMD driver launches multiple instances of QM components on subsets of processors (CCA)
- Each QM (gradient) component does multiple energy computations on subgroups (GA)

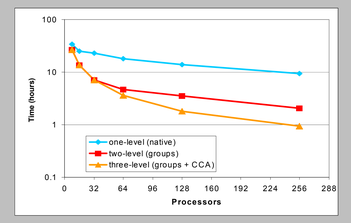




Traditional parallelization scheme:
- Each step executes on the entire machine

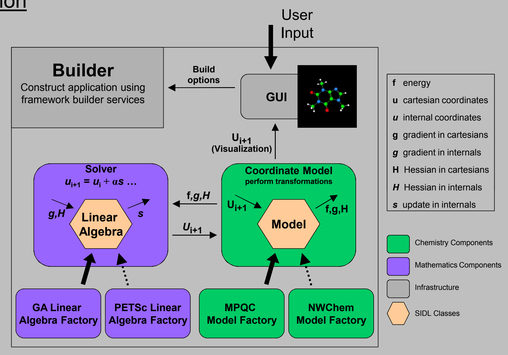
Using components, a Three-level Parallelism scheme was implemented:
- Energy-Level: Native NWChem code
- Gradient-Level: Global Array groups
- Hessian Level: Task-based using CCA

- Distributing work for maximum efficiency nontrivial and machine dependent
- Order of magnitude improvement:



High Level Components for Structure Optimization

- Evaluate component technology as an approach for high-performance software development, both within and outside the chemistry domain
- Gain interoperability between chemistry packages
- Incorporate and evaluate generic mathematics packages (TAO, PETSc, GA)
  - What improvements are needed for good performance of generic mathematics routines in the chemistry domain?
  - Will state-of-the-art mathematics routines provide better performance than application specific routines written by chemistry package developers?



Number of energy/gradient evaluations required to determine minimum energy molecular structures

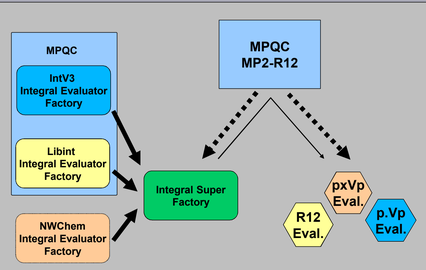
QC Package	MPQC	MPQC	NWChem	NWChem	NWChem
Solver Package/Algorithm	MPQC/BFGS	TAO/LMVM	NWChem/BFGS	NWChem/BFGS	TAO/LMVM
Line Search	no	yes	no	0.5*unit	yes
Guess Hessian	unit	scaled unit	0.5*unit	0.5*unit	scaled unit
Glycine ($C_2H_5NO_2$)	26/26	19/19 +27%	33/33	65/33	19/19
Isoprene (C_5H_{10})	75/75	43/43 +43%	56/56	89/45	45/45
Phosphoserine ($C_3H_8NO_6P$)	85/85	62/62 +27%	79/79	121/61	67/67
Acetylsalicylic Acid ($C_9H_8O_4$)	54/54	48/48 +21%	43/43	83/42	51/51
Cholesterol ($C_{27}H_{46}O$)	27/27	30/30 -11%	33/33	—/—	30/30

Stand-alone MPQC/NWChem | TAO Solver Component

- MPQC/NWChem solvers use the standard Broyden, Fletcher, Goldfarb, Shanno (BFGS) update
- The TAO package provides a limited-memory variable-metric (LMVM) solver
  - Uses a subset of correction vectors – may be more appropriate for long optimizations
  - Identical to BFGS with all correction vectors
- The TAO solver outperforms the solvers provided by both stand-alone QC packages – the TAO line search is well suited to molecular structure optimizations

Low Level Components for Extended Capabilities

- Integrals programs do not implement all integral types
- Ability to share integrals and combine packages
  - Enables new science
  - Permits selection of most efficient package for each machine



Component architecture for integral evaluation

Now possible to combine three corrections into one:
δ[core] + δ[rel.] + δ[basis] → δ[core+rel.+basis]

	$Cr(OH)_6$ Reaction 1		$CrO(OH)_4$ Reaction 2	
ΔE_{rxn}[HF]	−29.66		23.58	
δ[MP2]	+9.81		−7.47	
δ[CCSD]	+5.04		+8.45	
δ[CCSD(T)]	+9.94		+0.96	
	MP2	MP2-R12	MP2	MP2-R12
δ[basis]	−3.55		−2.39	
δ[core]	−0.32		−0.79	
δ[rel]	+2.08		+3.44	
δ[basis+core+rel]		−1.85		+0.32
δ[ZPVE]	−8.07	−8.07	−4.99	−4.99
$\Delta H^\circ_{rxn,0}$	−14.73	−14.79	20.79	20.85
$\Delta H^\circ_{f,0}$	−232.98	−232.92	−211.40	−211.45
$\Delta H^\circ_{f,298.15}$	−239.77	−239.71	−215.94	−216.00

- Low-level components tend to be finer grained with more function call overhead. For Hartree-Fock (execution time in seconds):

		MPQC	CCA	Overhead
H_2O cc-pVQZ	energy	19.9	21.0	5.5%
	gradient	39.5	41.3	4.6%
C_5H_{10} cc-pVDZ	energy	86.3	93.7	8.6%
	gradient	204.8	219.1	7.0%

Argonne National Laboratory, University of Chicago
Ames Laboratory
ORNL Oak Ridge National Laboratory
Pacific Northwest National Laboratory, Operated by Battelle for the U.S. Department of Energy
Sandia National Laboratories