

Fault tolerance for quantum chemistry methods

Curtis Janssen
Matt Leininger
Ida Beck Nielsen

Sandia National Laboratories

Goals

- ❖ Motivation
 - ❖ Beginning project to develop an high accuracy quantum chemistry program for large molecular systems
 - ❖ Requires large scale parallel computing
 - ❖ Fault tolerance will be an issue
- ❖ Outline
 - ❖ Background
 - ❖ Expected available computing resources in ~2004
 - ❖ Overview of several quantum chemistry methods
 - Computational requirements
 - Communication requirements and patterns
 - Thoughts on application level fault tolerance

MPQC: The massively parallel quantum chemistry program

- ❖ Computes properties of atoms and molecules from first principles using the time independent Schrödinger equation
- ❖ Closed shell and open shell Hartree-Fock and density functional theory energies and gradients
- ❖ Second order open shell perturbation theory (OPT2[2]) and Z-averaged perturbation theory (ZAPT2) energies.
- ❖ Second order closed shell Møller-Plesset perturbation theory energies and gradients.
- ❖ Object oriented design in C++
- ❖ Designed to run on parallel machines *ab initio*

Target parallel machine: Sample specification

Processors	10648
Processor flop rate	2.8 Gflop/s
Memory	768 MB/processor
Processors per node	1 to 4
Bidirectional bandwidth	4.2 GB/s per processor
Topology	3D mesh
Total flop rate	30 Tflop/s
Bisection bandwidth	2 TB/s
Global disk I/O rate	150 GB/s
OS	Lightweight
MTBI	50 hours

Goal of quantum chemistry

- ❖ Time-independent Schrödinger equation for molecules:

$$H\Psi = E\Psi$$

$$H = -\frac{1}{2} \sum_i^n \nabla_i^2 - \sum_i^n \sum_a^N \frac{q_a}{r_{ia}} + \sum_{i>j}^n \frac{1}{r_{ij}} + \sum_{a>b}^N \frac{q_a q_b}{r_{ab}}$$

- ❖ Solve using spectral methods:
 - ❖ Expand Ψ as product of one particle functions, ϕ
 - ❖ Expand ϕ as sum of atomic functions, χ

Methods of quantum chemistry: Hartree-Fock and density functional theory

- ❖ Hartree-Fock theory (density functional theory similar):

$$F|i\rangle = \varepsilon_i|i\rangle$$

$$F_{ij} = \sum_A^{\text{atoms}} \langle i | \frac{1}{r_A} | j \rangle + \sum_{kl} D_{kl} \left(2 \langle ik | \frac{1}{r_{12}} | jl \rangle - \langle ij | \frac{1}{r_{12}} | kl \rangle \right)$$

- ❖ Characteristics of parallel Hartree-Fock programs:
 - ❖ Iterative
 - ❖ Matrices can be replicated or distributed; dimension few 100 to few 1000; dense, but sparse in future

Task	Communication	% time
Compute integrals	Only if dynamic load balancing	90-95%
Linear algebra	yes	5-10%

Methods of quantum chemistry: Møller-Plesset theory

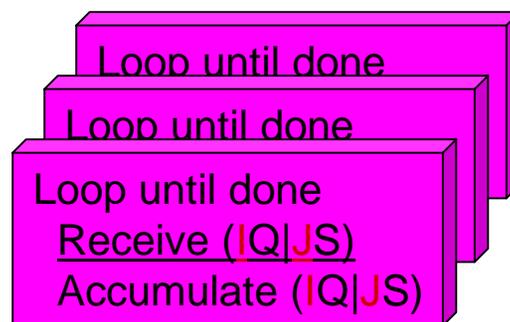
$$E = \frac{1}{4} \sum_{ijab} \frac{(\langle ij|ab\rangle - \langle ij|ba\rangle)^2}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}$$

$$\langle ij|ab\rangle = \sum_{pqrs} C_{pi} C_{qj} C_{ra} C_{sb} \langle pq|rs\rangle$$

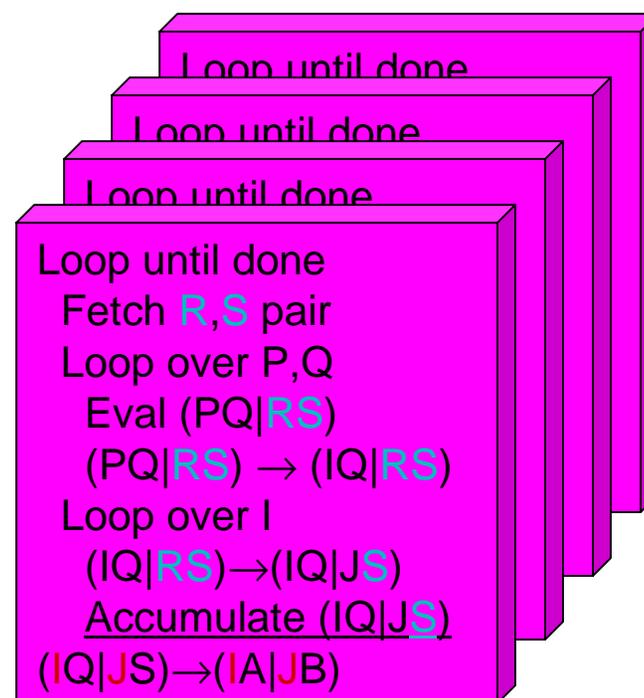
- ❖ Noniterative
- ❖ Distributed data
- ❖ Communication and computation overlaid

In terms of the number of electron pairs, n :

Storage	$160 n^4$ bytes
Flops	$1300 n^5$
Communication	$320 n^5$ bytes



Communication Threads



Computation Threads

Methods of quantum chemistry: Coupled cluster theory

$$\Psi = e^T \Phi$$

$$e^{-T} H e^T = E \Phi$$

- ❖ Solve nonlinear equations for t iteratively:

$$0 = \left(v_{ab}^{ij} - \frac{1}{2} v_{ba}^{ij} \right) t_{uj}^{\beta b} \left(2t_{iv}^{a\gamma} - t_{iv}^{\gamma a} \right) + \text{c.c.}$$

- ❖ Rough needs per iteration in terms of number of electron pairs, n :

	Small n	Large n ($n > 40$)
Storage for t	$12 n^4$	$644 \times 10^6 n$ bytes
Flops	$670 n^6$	$24 \times 10^{12} n$ flop/s
Communication	$4800 n^4$	$1.3 \times 10^9 n$ bytes

Approach to fault tolerance

- ❖ Standard checkpoint to disk but avoid requeuing job:
 - ❖ Detect errors
 - ❖ Redistribute work (possibly to spares)
- ❖ Use RAID 3 ideas, but to memory instead of disk
 - ❖ For example, use Hamming(7,4) to support multiple node errors
 - ❖ Detection and redistribution still issues

Coupled cluster theory: Specific example on 2004 machine

❖ Requirements for $n = 2000$ on full sample machine:

	Global	Per processor
Storage	1.3 TB	120 MB
Time/iter	1600 s	
Comm/iter	2.6 TB	244 MB
Comm time/iter	0.10s	
Disk checkpt time	9 s	
Mem checkpt time	0.4 s	

- ❖ Memory checkpointing ~10 times faster than disk
- ❖ Storage availability can be guaranteed

Issues for handling faults at application level

- ❖ Difficult verifying code works in exceptional circumstances
 - ❖ Automated verification suite
 - ❖ Reproducibility of bugs
- ❖ MPI is primary communication library for labs apps
 - ❖ Does not handle faults well
 - ❖ Need specified behavior and recovery methods
- ❖ ASCI supports vendor, open source, and 3rd party MPI
 - ❖ Can influence evolution of MPI

Conclusions

- ❖ Fault tolerance important on ASCI machines
- ❖ Application programmers need
 - ❖ Portable standards to deal with faults
 - ❖ Methods for verifying fault tolerant codes
- ❖ In-memory redundancy could be useful approach for certain distributed memory applications.