UNEDF project: some perceived software issues and collaboration opportunities

K. Roche
Future Technologies Group
Computer Science and Mathematics Division
Oak Ridge National Laboratory

*comments for: CSciADS Workshop on Libraries and Algorithms for Petascale Applications; Snowbird, Utah ; July 30 – August 2

*I am not able to represent all the efforts happening in this project.

UNEDF := Universal Nuclear Energy Density Functional

•SciDAC2 Application Project

•http://www.scidac.gov/physics/unedf.html
•http://www.unedf.org
•PI, George Bertsch (University of Washington)
•ASCR, NNSA, NP :: $3M per year , 5 years
Simple example: HF for fermionic system

\[ H \] = \sum_{i=1}^{N} (t_i + u_i) + \frac{1}{2} \sum_{i \neq j} v(x_i, x_j) 

• approximate 2-particle operator by effective 1-particle potential

\[ H_{HF} = \sum_{i=1}^{N} \left\{ H_{Li} - \frac{1}{2} \left( N \right)^{-\frac{1}{4}} \det \left[ \sum_{i} \sum_{j} (x_i \times y_j) \right] \right\} \]

• do not know HF potential \( \rightarrow \) can’t compute 1-particle orbitals
• apply Ritz’ variational theorem and minimize total energy of system wrt 1-particle orbitals

\[ \sum_{i=1}^{N} \int_{\text{dy}} \left( \psi_{HF}(y) \right)^* \left( \phi_{HF}(y) \right) = 0 \]

• Lagrange multipliers enforce normalized 1-particle orbitals

\[ \psi(y) = \phi(y) \]

\[ \int_{\text{dy}} \left( \psi(y) \right)^* \left( \phi(y) \right) \]

\[ \int_{\text{dy}} (x \times y) \psi(x) \phi(y) \]
Simple example: HF for fermionic system (continued)

• Self-consistent iteration
  • prescribe the 1-particle orbitals $\phi_i$, $i=1,N$ (in chosen basis)
  • (loop) compute HF potential $v_{HF}$ from density matrix $\rho(x,y)$
  • solve eigenequation $(t + v_{HF}) \psi = E \psi$
  • choose $N$ orbitals $\phi_i$ with $N$ lowest eigenvalues $E_i$
  • loop until self-consistent convergence

• Approximate HF total energy of the system from the occupied 1-particle orbitals
  • $E_{HF} = \frac{1}{2} \sum_{i=1,N} \langle i | t + v | i \rangle$ (2nd quantized notation)

Kohn-Sham Theory:

• take the variational functions to be the HF orbitals
• calculate kinetic energy and density w/ orbitals
• introduce the exchange correlation energy, $E_{xc}$
• Minimize the resulting functional $E_{KS} = \langle \phi | t + u + v_{xc} | \phi \rangle$
• depends on nonlocal term, $\left| \nabla \phi_i \right|^2$

Some computational issues:

• Solve the poisson equation
  • periodic lattice, 3D Fourier transforms ~ $O(N \log N)$
  • can use multigrid on laplacian in coordinate space

• Solve Schroedinger equation $\rightarrow$ multiply $H_{KS}$ on each wavefunction
• Split operator
  • $H_{KS} \psi = \mathbf{F}^{-1} \frac{k^2}{2m} \mathbf{F} \psi$ $\rightarrow$ $V_{KS}(x) \phi_i$
• Iterative refinement of wavefunctions
  • Imaginary time
  • Conjugate gradient (large KE $\rightarrow$ preconditioning)
DFT Challenges for nuclear physics: LDA, GGA, SLDA, …

• Two kinds of fermions (isospin)
• Need to include pairing
• Broken symmetry of self bound systems

• Hartree Fock Bogoliubov—a generalized single particle model

• HFB equation has the following symplectic structure (no time to dissect this)

\[
\begin{pmatrix}
  h_{11} - \mu & h_{12} & 0 & \Delta \\
  h_{21} & h_{22} - \mu & -\Delta & 0 \\
  0 & -\Delta^* & -h_{11} + \mu & -h_{12} \\
  \Delta^* & 0 & h_{21} & -h_{22} + \mu
\end{pmatrix}
\begin{pmatrix}
  U^+ \\
  U^- \\
  V^+ \\
  V^-
\end{pmatrix} = E
\begin{pmatrix}
  U^+ \\
  U^- \\
  V^+ \\
  V^-
\end{pmatrix}
\]

Current software dependencies for the HFB based (SLDA) TDDFT:

• Work between coordinate and k spaces in a plane wave basis
• Each MPI process has a local copy of both lattices
• Quasiparticle wavefunctions are distributed over each MPI process
• 3D FFTs \texttt{\rightarrow} FFTW3.1.2
  • Used extensively each time step on the single particle orbitals
• Formation of density based terms depends only on 1-particle qpwf
  • Requires reduction over all processes each time step for the normal density, kinetic energy density, density gradients
• Compute gradients of both kinetic and normal densities as corrections to the edf
• Time evolution is now with ABM PMC technique
  • Could use Crank-Nicholson
  • Loss of orthonormality is an issue here \texttt{\rightarrow} Gram-Schmidt
**Discussion of memory concerns:**

- ALLOCATE (wavf(nwfip,-3:1,4,Nx,Nx,Nx),STAT=ierr)
  - IF (ierr .NE. 0) THEN
    - WRITE(6,*,'ERROR: cannot ALLOCATE wavf(l)')
  - ENDIF
- ALLOCATE (wavf_t_der(nwfip,-3:1,4,Nx,Nx,Nx),STAT=ierr)
  - IF (ierr .NE. 0) THEN
    - WRITE(6,*,'ERROR: cannot ALLOCATE wavf_t_der()')
  - ENDIF
- ALLOCATE (wavf_modifier(nwfip,-3:1,4,Nx,Nx,Nx),STAT=ierr)
  - IF (ierr .NE. 0) THEN
    - WRITE(6,*,'ERROR: cannot ALLOCATE wavf_modifier()')
  - ENDIF
- ALLOCATE (wavf_predictor(nwfip,-3:1,4,Nx,Nx,Nx),STAT=ierr)
  - IF (ierr .NE. 0) THEN
    - WRITE(6,*,'ERROR: cannot ALLOCATE wavf_predictor()')
  - ENDIF
- ALLOCATE (wavf_corrector(nwfip,-3:1,4,Nx,Nx,Nx),STAT=ierr)
  - IF (ierr .NE. 0) THEN
    - WRITE(6,*,'ERROR: cannot ALLOCATE wavf_corrector()')
  - ENDIF
- ALLOCATE (rho(Nx,Nx,Nx),STAT=ierr)
  - IF (ierr .NE. 0) THEN
    - WRITE(6,*,'ERROR: cannot ALLOCATE rho()')
  - ENDIF
- IF (np>1) THEN
  - IF (ip==0) THEN
    - ALLOCATE (trho(Nx,Nx,Nx),STAT=ierr)
      - IF (ierr .NE. 0) THEN
        - WRITE(6,*,'ERROR: cannot ALLOCATE trho()')
      - ENDIF
  - ENDIF
  - ENDIF
  - ENDIF

**Pursuits:**

- Check-point restart
- rma
- Pthread based io overlap with computation, data reorganization

128**3 ~ 2,097,152

**Coupled Cluster example:**

- `ab initio` technique
- Calculates ground-state properties of closed-shell (or sub-shell) nuclei
- Solves coupled nonlinear sets of equations (largest ~10M unknowns)
- $\text{CCSD} \sim O(\nu^2 \cdot N_0^4)$ (\nu:=unoccupied orbitals; \no:=occupied)
- $\text{CCSDT} \sim O(\nu^3 \cdot N_0^5)$

**SAMPLE TERM:** (fully reduced in 2\textsuperscript{nd} quantization notation)

For $i=1,n \sim 100$
For $j=1,n$
For $a=n+1,N \sim 1000$
For $b=n+1,N$

$$F(a,b,i,j) = \sum_{c=n+1,N} \sum_{d=n+1,N} \sum_{k=1,n} \sum_{l=1,n} V(k,l,c,d) \cdot T(c,d,i,j) \cdot T(a,b,k,l)$$
Make the thing look like a sequence of matrix multiplies.

\[
F(a,b,i,j) = \sum_{c=n+1}^{N} \sum_{d=n+1}^{N} \sum_{k=1}^{n} V(k,l,c,d) \times T(c,d,i,j) \times T(a,b,k,l)
\]

Tasks

- copy data into matrix form
  - from disk (next slide)
  - in network
- impose 2D block decomposition for larger sub-blocks
- use Goto BLAS or PBLAS *gemm() subroutines
- write products

Other

- Shared and remote memory protocols, co-arrays, global arrays
- Role of multi-core is not clear

Files on disk

1

Pre-mapped data

The pre-mapped data is brought in-core in parallel by each process in the logical process grid. Each process accesses its own file.

2

Natural data

File object - disk

Root process

Remaining nodes

The natural data is brought in-core by a chosen process (root) from the logical process grid. The data may be distributed by root in a manner imposing the mapping during direct communication with other nodes or. Root may broadcast the data as is and let the mapping be done by each process locally.

3

Natural data

File object - disk

Allocated compute nodes

The natural data is brought in-core in parallel by each process in the logical process grid. The mapping may be imposed exactly during the load through random access to the file or bulk data can be brought over the network by each process and the mapping imposed in-core.
Lesson learned:

G1 –from MPI_COMM_WORLD
G3 = G1 \ G2  (the work group)
G2 := outliers

Conclusion: (Could go on but this is a good place to stop)

• Multicore, what adjustments will be needed, enhancements gained

• Collective io is coming
  • PxQ → RxS 2d block cyclic mapping research
  • Check-pt-restart

• Believe we have a need for hybrid programming model for pthreads in an MPI environment; also language interoperability

• Many of the unedf codes could not be discussed
  • Eg, multiwavelet basis based approach is interesting effort

• Gram-Schmidt is a concern

• SVD and HOSVD are being pursued in some UNEDF codes

• Dense and sparse (both iterative and direct) solvers are critical to unedf and at the petascale

• Diagonalization of the Hamiltonian → zheevd()
  • Ax=cx ; A(x+dx)=(c+dc)(x+dx)