

大規模時間密度汎関数理論計算のための MPI/OpenMP ハイブリッド並列プログラムの開発

(理研 AICS) ○河東田 道夫, 中嶋 隆人

Development of MPI/OpenMP hybrid parallel program for time-dependent density functional theory calculations

(RIKEN AICS) ○Michio Katouda, Takahito Nakajima

【序】近年のスーパーコンピュータは、マルチコア CPU を複数搭載したノードを高速なインターコネクトで接続したマルチコア超並列クラスタシステム構成が主流となっており、現在ではスーパーコンピュータ「京」のような数万ノード・数十万 CPU コアで構成された数 10 ペタフロップス級の演算性能を持つスーパーコンピュータも出現している。発表者らの研究チームでは、「京」の性能を最大限に引き出し大規模分子の計算を可能とする分子科学計算ソフトウェア NTChem[1,2]の開発を行っている。現在、大規模分子の励起状態計算を NTChem で実現すること目的として、線形応答時間密度汎関数理論(LR-TDDFT)に基づき励起状態計算を行うための計算プログラムの開発を進めている。これまでに、TDDFT random phase approximation (TDDFT/RPA) 法と TDDFT Tamm-Dancoff approximation (TDDFT/TDA)法の実装を行い、計算のボトルネックとなるフォック様行列計算部分のフラット MPI 並列化を行った。しかしながら、フラット MPI 並列化による制約のため、数百プロセス規模の並列計算を行うのが限界であり、数千ノード規模での並列計算を行うことは困難となっていた。さらに、励起状態の解を求めるための大規模行列の固有値解法として採用している Davidson 型のアルゴリズムの並列化を行っていないため、励起エネルギーの解を数十個以上求める場合には膨大な計算時間を要することも問題となっていた。本研究では、大規模分子の TDDFT/RPA 法および TDDFT/TDA 法を「京」などのマルチコア超並列クラスタシステムで高速・高並列に行うことを目的として、これらの計算システムで推奨されている MPI/OpenMP ハイブリッド並列計算プログラムの開発を行った[2]。本発表では開発したプログラムの概要および、「京」でのテスト計算結果を紹介する。

【並列実装】TDDFT 計算ではフォック様行列計算が最大の律速区間、Davidson 型のアルゴリズムによる固有値計算が第 2 の律速区間となるため、これらの箇所を中心に MPI/OpenMP 並列化を行った(図 1)。フォック様行列計算の MPI/OpenMP 並列化(図 1(1))は DFT 計算の MPI/OpenMP 並列化と同様のアルゴリズム[3]を採用して行った。Davidson 型の固有値計算の MPI/OpenMP 並列化では、試行ベクトルと残差ベクトルをアップデートする処理においてこれらのベクトルデータを各ノードに分散して保持し、ノード毎の処理を MPI 並列化、ノード内の処理を OpenMP 並列化する実装を行った(図 1(2)-(9))。さらに、相互作用行列計算(図 1(4))および、拡張した試行ベクトルの修正 Gram-Schmidt 直交化(図 1(9))の処理について、ブロック化されたデータの行列-行列積演算を BLAS ライブラリの DGEMM を用いて行うことにより高速化する実装を行った。

【結果】今回開発したアルゴリズムを分子科学計算ソフトウェア NTChem に実装し、「京」で並列性能ベンチマークテストを行った。表 1 にオリゴチオフェン dendrimer $C_{168}H_{86}OS_{43}$ の TDDFT/RPA 励起状態計算の計算時間および高速化率を示す。「京」2048CPU コアを用いた計算の高速化率を 2048 倍とした場合、16384CPU コアを用いた計算では 5026 倍の加速を達成しており、CPU コア数の増加に対して良好な並列性能を達成することが確認された。また、16384 コアを用いた場合の計算時間は

213 分であり、応用計算も十分実行可能な計算時間内であった。本研究で開発されたソフトウェアを「京」で用いることにより、300 原子 4000 原子軌道規模のナノサイズ分子の TDDFT 励起状態計算がルーチンワークとして実行可能となり、光機能性分子材料や光受容タンパク質などの光応答機構解明などへの応用が期待される。

(0) Prepare initial trial vector $t_{ia}^{[q]}$

Loop of TDDFT iteration

(1) Evaluate Fock like matrix in AO based algorithm

$$B_{\mu\nu}^{[q]} = \left[2(\mu\nu|\lambda\sigma) - c_x(\mu\lambda|\nu\sigma) + c_x^{\text{LR}}(\mu\lambda|\nu\sigma)^{\text{LR}} + \int_{\mu\nu,\lambda\sigma}^{\text{xc}} \right] t_{\lambda\sigma}^{[q]}$$

(MPI parallel: ν, λ, σ , OpenMP parallel: μ)

(2) Transform AO to MO $B_{ia}^{[q]} = \sum_{\mu\nu} C_{\mu i} C_{\nu a} B_{\mu\nu}^{[q]}$

(MPI parallel: i, a, q block, OpenMP parallel: i, a, q in a block)

(3) Evaluate matrix $(A \pm B)_{ia}^{[q]} = (\epsilon_a - \epsilon_i) t_{ia}^{[q]} \pm B_{ia}^{[q]}$

(MPI parallel: i, a, q block, OpenMP parallel: i, a, q in a block)

(4) Evaluate reduced interaction matrix $(A_R \pm B_R)_{pq} = \sum_{ia} \left\{ t_{ia}^{[p]} (A \pm B)_{ia}^{[q]} \right\}$

(MPI parallel: i, a, q block, OpenMP parallel: i, a, q in a block)

(5) Solve Hermitian eigenvalue problem $(\mathbf{A}_R - \mathbf{B}_R)^{1/2} (\mathbf{A}_R + \mathbf{B}_R) (\mathbf{A}_R - \mathbf{B}_R)^{1/2} \mathbf{Z}_R = \omega_R^2 \mathbf{Z}_R$

to get left $\tilde{L}_{qn}$ and right $\tilde{R}_{qn}$ eigenvectors and eigenvalues ω_R (OpenMP parallel)

(6) Evaluate residual vectors $W_{ia}^{[n]L} = \sum_q (A+B)_{ia}^{[q]} \tilde{R}_{qn} - \omega_n \sum_q \tilde{L}_{qn} t_{ia}^{[q]}$ and $W_{ia}^{[q]R} = \sum_q (A-B)_{ia}^{[q]} \tilde{L}_{qn} - \omega_n \sum_q \tilde{R}_{qn} t_{ia}^{[q]}$

(MPI parallel: i, a, q block, OpenMP parallel: i, a, q in a block)

(7) Check convergence of $W_{ia}^{[n]L}$ and $W_{ia}^{[n]R}$ and break this loop if converged

(8) Update trial vectors $t_{ia}^{[q]}$ adding a set of perturbed vectors $q_{ia}^{[q]}$

(MPI parallel: i, a, q block, OpenMP parallel: i, a, q in a block)

(9) Gram-Schmidt orthogonalization of $q_{ia}^{[q]}$ with respect to $t_{ia}^{[q]}$

(MPI parallel: i, a, q block, OpenMP parallel: i, a, q in a block)

End loop of TDDFT iteration

図 1. MPI/OpenMP ハイブリッド並列 TDDFT/RPA 計算アルゴリズムの概略

表 1. 「京」での MPI/OpenMP ハイブリッド並列 TDDFT/RPA 計算の計算時間と高速化率 (オリゴチオフェンデンドリマー C₁₆₈H₈₆OS₄₃ B3LYP/def2-SVP, 298 原子, 3538 原子軌道, 励起エネルギー解 20)

ノード数	CPU コア数	計算時間[秒]	高速化率
256	2048	31483	2048
512	4096	20703	3114
1024	8192	15404	4186
2048	16384	12829	5026

[参考文献] [1] NTChem2013, http://labs.aics.riken.jp/nakajimat_top/ntchem_j.html

[2] T. Nakajima, M. Katouda, M. Kamiya, and Y. Nakatsuka, Int. J. Quant. Chem. **115**, 3 (2015).

[3] K. Ishimura, K. Kuramoto, Y. Ikuta, and S. Hyodo, J. Chem. Theory Comput. **6**, 1075 (2010).