

GPUを用いた大規模電子状態計算プログラムProteinDFの高速化

東大生研

○平野敏行, 佐藤文俊

GPU Acceleration of large scale electronic state calculation program: ProteinDF

○Toshiyuki Hirano, Fumitoshi Sato

Institute of Industrial Science, The University of Tokyo, Japan

【Abstract】

In developing the electronic structure calculation program, ProteinDF, for large scale metalloprotein, we proposed the third-generation density functional (3rd Gen DF) calculation method suitable for distributed memory parallel computer. In the 3rd Gen DF calculation method, electronic structure calculations can be achieved only by matrix operations during the SCF iterative calculation. The GPU is excellent in massively parallel computing of large data and low power consumption per computing performance, and it is expected that efficiency of the large amount of matrix computation can be improved by using the GPU. In this research, we have aimed to replace the matrix operation part of the ProteinDF with GPU code, to speed up the calculation of protein electronic structure calculation and to confirm the usefulness of the 3rd Gen DF calculation method in heterogeneous computing environment.

【序】

世界のスパコン性能ランキングである TOP500 [1]のリストにおいて、近年 GPU を始めとするアクセラレータ搭載型が増加・流行している。GPU は、大規模データの超並列演算および計算性能あたりの低消費電力の点で優秀であり、大規模計算を扱うシミュレーションプログラムにおいて GPU の利用は不可欠である。

我々は、大規模金属タンパク質の電子状態計算プログラム ProteinDF [2]の開発にあたり、分散メモリ型並列計算機に適合する第三世代密度汎関数計算法を提案した[3]。第三世代密度汎関数計算法では、計算律速な分子積分を SCF 繰り返し計算前に一度だけ行い、SCF 繰り返し計算中は行列演算のみで Kohn-Sham 行列を得ることができる。すなわち、計算機に最適化された線形演算ライブラリを利用することにより、SCF 繰り返し計算中は当該計算機の良い実行性能が容易に得られることを意味する。

これまで ProteinDF では、線形演算ライブラリとして BLAS/LAPACK [4], ScaLAPACK [5]を利用した開発を行ってきた。本研究では、ProteinDF の行列演算部を GPU コードに置き換え、タンパク質電子状態計算の高速化を図るとともに、ヘテロジニアスな環境における第三世代密度汎関数計算法の有用性を確認することを目的とした。

【方法 (実験・理論)】

行列演算ライブラリとして、BLAS/LAPACK の他, Eigen [6], ViennaCL [7]を採用した。BLAS/LAPACK は、Fortran で実装された線形演算ライブラリであり、ハードに最適化された実装が数多く提供され、デファクトスタンダードとなっている。Eigen は C++におけるテンプレートを活用したライブラリである。OpenMP を利用したスレッド並列はもちろん、AVX, FMA, AVX512 など SIMD を利用したベクトル化も可能であ

る。ViennaCLはOpenCLを使用した線形演算ライブラリである。Eigen同様、テンプレートライブラリであり、前もってインストールは不要である。OpenCLはCPUやGPU、その他アクセラレータが混在するヘテロジニアスな環境において、これらの計算資源をより汎用的な演算に利用するための規格・ライブラリである。

Kohn-Sham 行列の生成や対角化などの電子状態計算に必要な行列演算操作において、その行列のプログラム上での実装は関係がない。すなわち、行列がCPUやGPU、およびどの線形演算ライブラリ上で演算されようとも、行列演算操作は普遍である。ProteinDFが開発言語として用いているオブジェクト指向言語C++では、クラスを利用して各行列演算ライブラリを抽象化した。本研究では、将来的に他の線形演算ライブラリにも対応できるようにするため、Bridgeデザインパターンを採用した。本研究で開発したクラス図を図1に示した。Bridgeデザインパターンは、実装とインターフェースを分離するため、新たな実装が追加されても、Kohn-Sham 行列の生成や対角化などに関するインターフェースは変更されない。Bridgeデザインパターンにより、コードの省力化・クラスの追加を容易にすることができた。

【結果・考察】

本研究で開発したクラスを用いた倍精度密行列の行列積・対角化のベンチマーク結果を図2に示した。共にCPUコードであるLAPACK(BLAS)とEigenライブラリはほぼ同様な結果が得られた。GPUを利用する同じViennaCLライブラリでは、行列積に有効なAMD社製と、対角化に有効なnVidia社製とで違いが見られた。第三世代密度汎関数計算法において、計算律速となる分子積分はSCF繰り返し計算前に計算する。今後、分子積分部のGPUコード化も検討したい。

【参考文献】

- [1] TOP500: <https://www.top500.org/>
- [2] ProteinDF: <https://proteindf.github.io/>
- [3] T. Hirano, F. Sato, *Phys. Chem. Chem. Phys.*, **16**, 14496-14503 (2014).
- [4] LAPACK: <http://www.netlib.org/lapack/>
- [5] ScaLAPACK: <http://www.netlib.org/scalapack/>
- [6] Eigen: <http://eigen.tuxfamily.org/>
- [7] ViennaCL: <http://viennacl.sourceforge.net/>

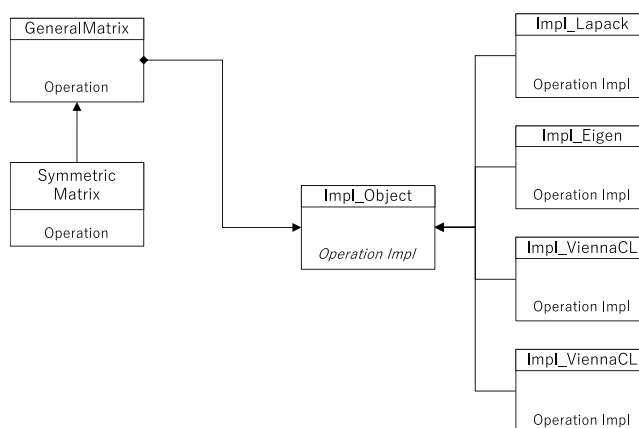


Fig. 1. Class diagram for matrices

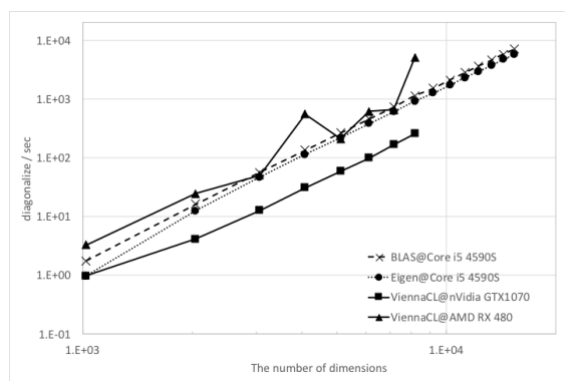
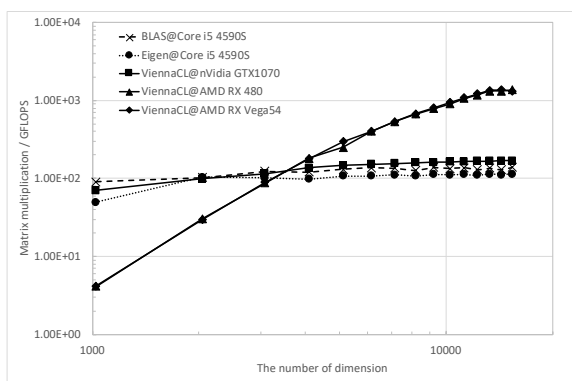


Fig. 2. Benchmarks of dense matrix-matrix multiplication (left) and diagonalization (right)