

京コンピュータでの計算情報

H P C I 戦略分野 5 課題 2 では、モンテカルロ殻模型コードを京で走らせています。

京向けのチューニング作業では、以下の作業を主におこないました。

京に限らず一般のスパコン向けにもあてはまる内容です。

1. アルゴリズムを可能な限り密行列積で記述し、チューニングされている BLAS ライブラリ/DGEMM を用いることにより、単体ノードでの実行性能をあげる。
2. OpenMP において、スレッド中、Fortran の「自動配列」などにより動的にメモリー確保するコストが京に関しては特に大きく、パフォーマンス悪化の要因となった。これを排除する。
3. 大規模並列向けに、並列化する階層を増やす。

項目 1. に関しては、下記文献に発表した手法を中心にしてコードのあらゆる箇所で、密行列積にできるところは可能な限り書き換えた。(場合によっては遅くなるので、段階的にテストしていく必要があります。)

これにより、ボトルネック部分のベストケースでは理論性能比で実行性能は 80% だすことが可能。

コード全体では、ベストケースで 40% 近くまでだすことができた。

Y. Utsuno, N. Shimizu, T. Otsuka, and T. Abe, "Efficient computation of Hamiltonian matrix elements between non-orthogonal Slater determinants", *Comp. Phys. Comm.* 184, 102 (2013).

DGEMM/BLAS ライブラリは、京では富士通 SSLII ライブラリにふくまれている。

京テスト運用時の富士通 SSLII ライブラリの性能が小次元行列の場合に悪く、対応策を検討していたが、一般供用開始までに SSLII ライブラリの性能が向上し、ほぼ問題なくなった。

項目 2. に関しては、OpenMP スレッドから呼ばれる関数の working area の確保が、サイズが確定せず、自動配列で確保するような場合に問題となる。

Intel Fortran ではほとんどコストがかからない場合でも、京・SPARC 上の Fujitsu Fortran では大きなコストがかかることが多い。

対策としては、スレッド外であらかじめ確保するか、サイズを parameter 文によって静的に指定する。

項目 3. に関しては、大規模並列に対応するため、

各ハミルトニアン行列要素の計算=シリアル

1 行列要素を計算するための積分計算=並列

にしていた 2 階層のものをまとめて、行列要素の計算から並列にするようコードの書き換えをおこなった。これにより数万並列までの対応が可能となった。

上記 3 点以外にも、(プロファイルにより時間がかかっている部分の発見+その部分のアルゴリズムの検討・改良) のサイクルを多数繰り返しています。