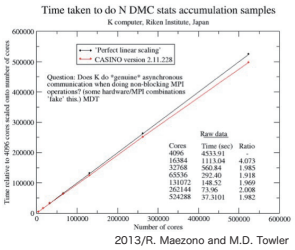
固体系量子モンテカルロ電子状態計算のAPUを用いた高速化

（北陸先端科学技術大学院大学・情報科学研究科 ）○今村 光良、前園涼

I. Introduction

量子モンテカルロ(QMC)計算のMPI並列性能は十分

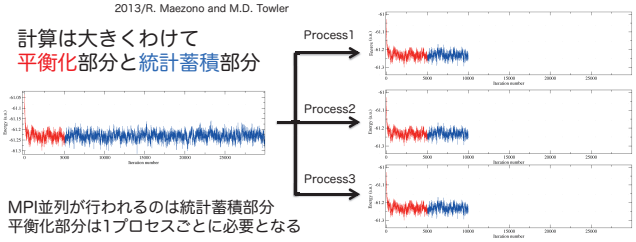


2013/R. Maezono and M.D. Towler

50万並列でも90% 以上の
並列化効率を維持している
※Weak Scaling

しかし超並列計算の実現によって
ボトルネックが変わる

計算は大きくわけて
平衡化部分と統計蓄積部分



MPI並列が行われるのは統計蓄積部分
平衡化部分は1プロセスごとに必要となる

新たに計算律速箇所となった部分を高速化することが課題

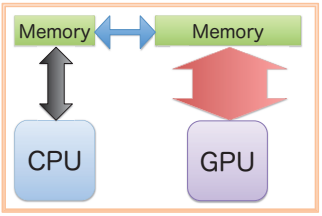
$$E \approx \frac{1}{M}\sum_{i}^{M} \Psi_{trial}^{-1}(\vec{R}_i)\hat{H}\Psi_{trial}(\vec{R}_i)$$

既存のMPI並列を活かした
1プロセスあたりの計算性能の向上

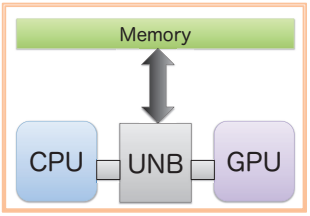
II. Acceleration Device

本研究ではMPI並列と共存できる並列化として、GPGPUを扱う
また、利用するGPUは従来の外付けのものではなく、
CPUコアと一緒にパッケージされたCPU内蔵のGPUを用いる

外付け型



内蔵型※



内蔵型のGPUは外付け型のGPUと比べ、メモリバンド幅こそ劣るが、
UNB(Unified North Bridge)という機能を利用することで、CPUと同一のメモリを扱い、
PCIeを介さず、直接データのやりとりができるという利点がある
※本研究ではAMD社のAPUを指す

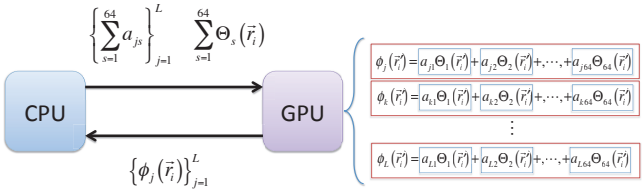
III. Implementation

平衡化部分の律速1つとして
試行関数を構成する一体軌道関数$\phi(\vec{r}_i')$の再計算部分がある

$$\left\{\phi_j(\vec{r}_i')\right\}_{j=1}^{L} = \left\{\sum_{s=1}^{64} a_{js}\cdot\Theta_s(\vec{r}_i')\right\}_{j=1}^{L}$$

軌道: j と s のループを並列計算可能

再計算部分をGPUで計算することで高速化を実現する



$$\left\{\sum_{s=1}^{64} a_{js}\right\}_{j=1}^{L} \quad \sum_{s=1}^{64}\Theta_s(\vec{r}_i')$$

$$\left\{\phi_j(\vec{r}_i')\right\}_{j=1}^{L}$$

$$\phi_j(\vec{r}_i') = a_{j1}\Theta_1(\vec{r}_i') + a_{j2}\Theta_2(\vec{r}_i') + \cdots + a_{j64}\Theta_{64}(\vec{r}_i')$$

$$\phi_k(\vec{r}_i') = a_{k1}\Theta_1(\vec{r}_i') + a_{k2}\Theta_2(\vec{r}_i') + \cdots + a_{k64}\Theta_{64}(\vec{r}_i')$$

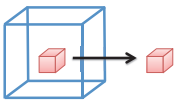
$$\phi_L(\vec{r}_i') = a_{L1}\Theta_1(\vec{r}_i') + a_{L2}\Theta_2(\vec{r}_i') + \cdots + a_{L64}\Theta_{64}(\vec{r}_i')$$

OpenCL1.2を用いて以下の通り実装した

Packaging Vector

$\sum_{s=1}^{64}\Theta_s(\vec{r}_i')$ $\vec{r}_i'$ に対して4×4×4の64個から成る各Blip基底関数で
評価した値をGPUで計算するように準備する

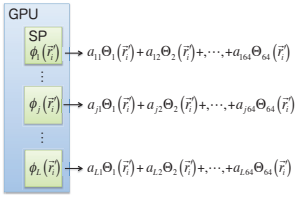
Packaging Coefficient



各軌道におけるBlip基底関数の係数を準備する

GPUで処理しやすいすいように
計算部分の係数のみでデータを再構成する

Calculating



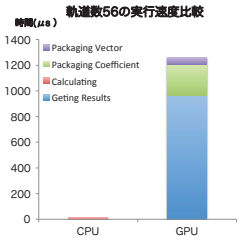
軌道の並列数を利用して、
SP(ストリームプロセッサ)で
並列処理をおこなっていく

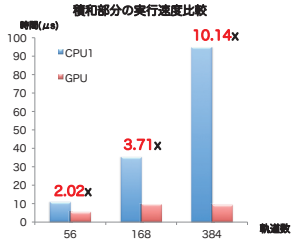
Getting Results

※本実行環境では倍精度扱えないため、データの準備段階で倍精度から単精度に、
結果の取得段階で単精度から倍精度とする動作を含んでいる

IV. Results

・実行環境
CPU : AMD Fusion APU A6-3670 2.7GHz
GPU : HD 6350D 443MHz SP 320基
試行回数 ： 10000





・計算部分が10μs以下なのに対して各Packagingでは数十倍、
結果の取得に関しては数百倍の時間を要している
- Packagingに関して言えば、今回作成したプログラムでは、
計算ルーチンにはいたびにpackagingを行っていたが、
原理的に係数部分はpackagingを1回にすることが可能
- 基底関数計算部分も計算ルーチンにはいる前の計算段階でpackagingが可能
- 倍精度が利用できれば精度を戻す作業が不要

・積和算を行うCalculating部分のみで比較すると最大で約10倍の高速化

V. Summary

・**成果**
内蔵型のGPUを利用することで最大約10倍の高速化が可能

・**今後の課題**
利用するデータの扱いを最適化する必要がある

References

"GPGPU for orbital function evaluation with a new updating scheme",
Y. Uejima and R. Maezono, J. Comput. Chem. 34, 83–94 (2013).

"Acceleration of a QM/MM-QMC simulation using GPU",
Y. Uejima, T. Terashima and R. Maezono, J. Comput. Chem. 32, 2264-2272 (2011).