

モンテカルロ殻模型計算のマルチGPGPUへの 適用と開発状況

富樫智章¹, 清水則孝¹ (代表), 吉田亨¹, 宇都野穰^{1,2},
阿部喬³, 大塚孝治^{1,3}

¹東大原子核科学研究センター, ²JAEA, ³東大理

第6回「学際計算科学による新たな知の発見・統合・創出」シンポジウム
学際共同利用 成果発表 平成26年度中間報告

2014.10.22

殻模型(shell model)計算とは

多体のSchrödinger方程式: $H|\Psi\rangle = E|\Psi\rangle$

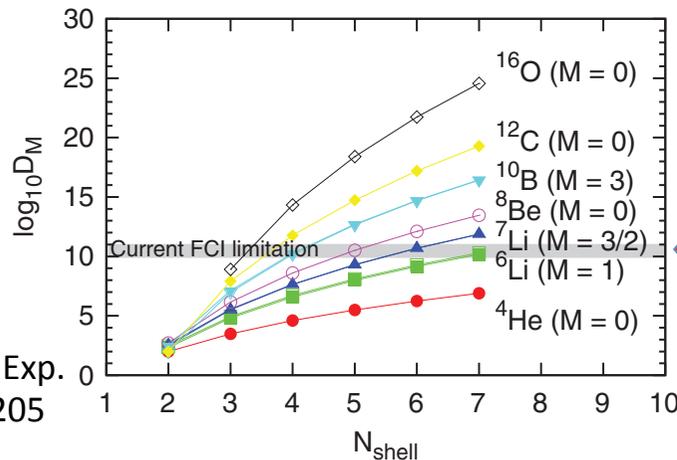
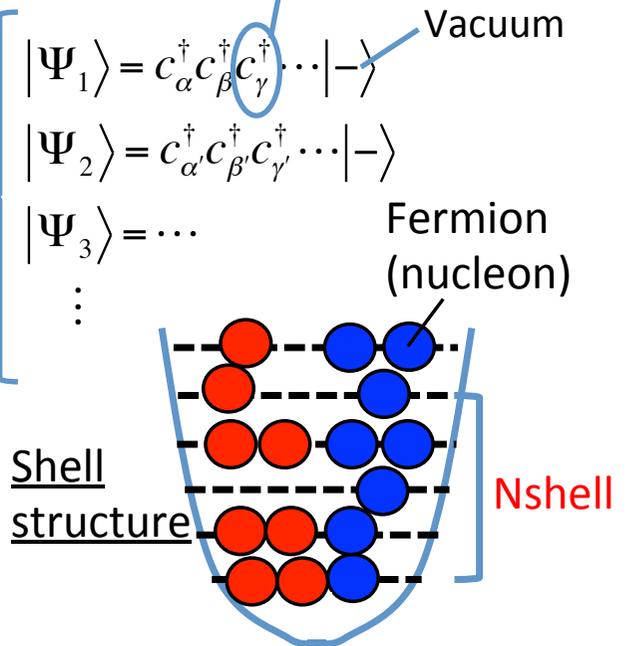


Shell model (Configuration Interaction, CI)

$$\begin{pmatrix} H_{11} & H_{12} & H_{13} & H_{14} & H_{15} & \cdots \\ H_{21} & H_{22} & H_{23} & H_{24} & & \\ H_{31} & H_{32} & H_{33} & & & \\ & H_{41} & H_{33} & \ddots & & \\ H_{51} & & & & & \\ \vdots & & & & & \end{pmatrix} \begin{pmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \\ \Psi_5 \\ \vdots \end{pmatrix} = \begin{pmatrix} E_1 & & & & & 0 \\ & E_2 & & & & \\ & & E_3 & & & \\ & & & \ddots & & \\ 0 & & & & & \end{pmatrix} \begin{pmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \\ \Psi_4 \\ \Psi_5 \\ \vdots \end{pmatrix}$$

次元数: $\sim \mathcal{O}(10^{10})$ (非ゼロ行列要素数: $\sim \mathcal{O}(10^{13-14})$) **疎行列**

殻模型軌道(0s,0p,...)の生成演算子



モンテカルロ殻模型では
通常の殻模型計算では計算できない
次元に対応する計算が可能

モンテカルロ殻模型計算によるクラスター構造研究 (by 吉田)

Density plots of Be isotopes (0_1^+) Preliminary

A = 8

A = 9

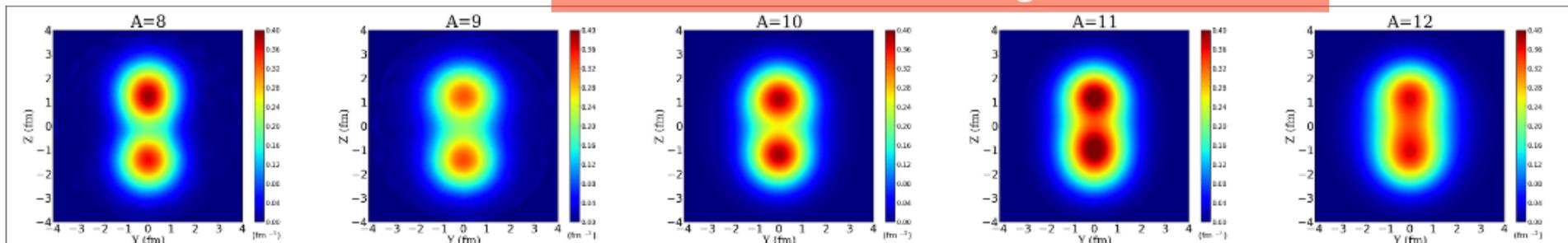
A = 10

A = 11

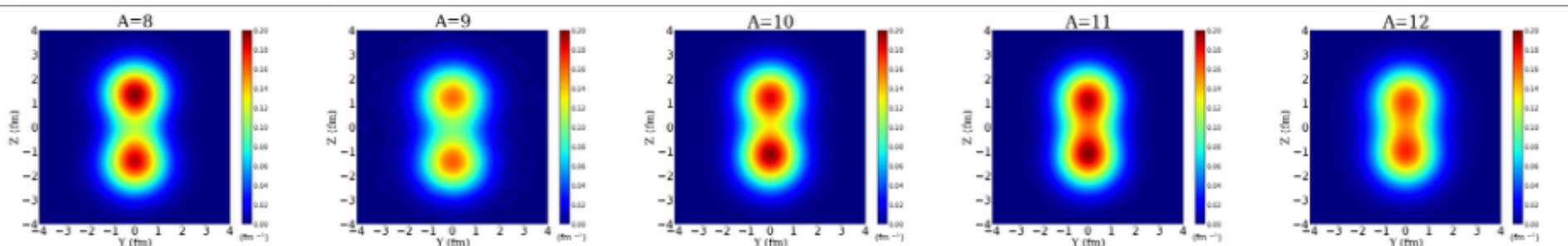
A = 12

Matter density

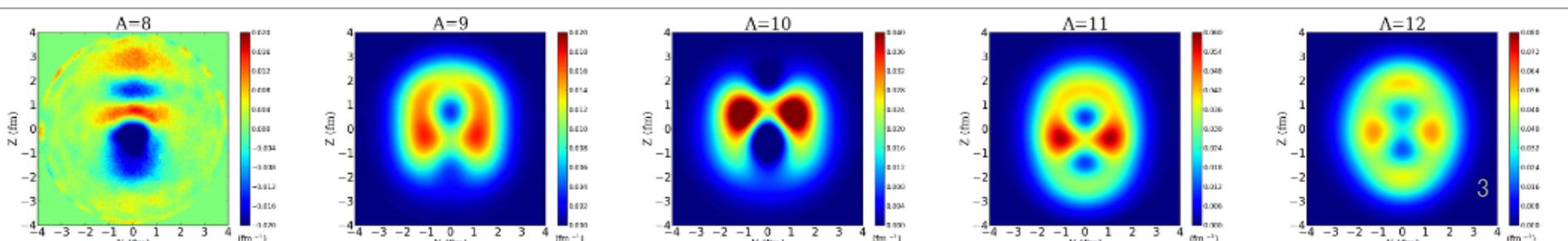
2- α structure is vanishing as A increases



Proton density



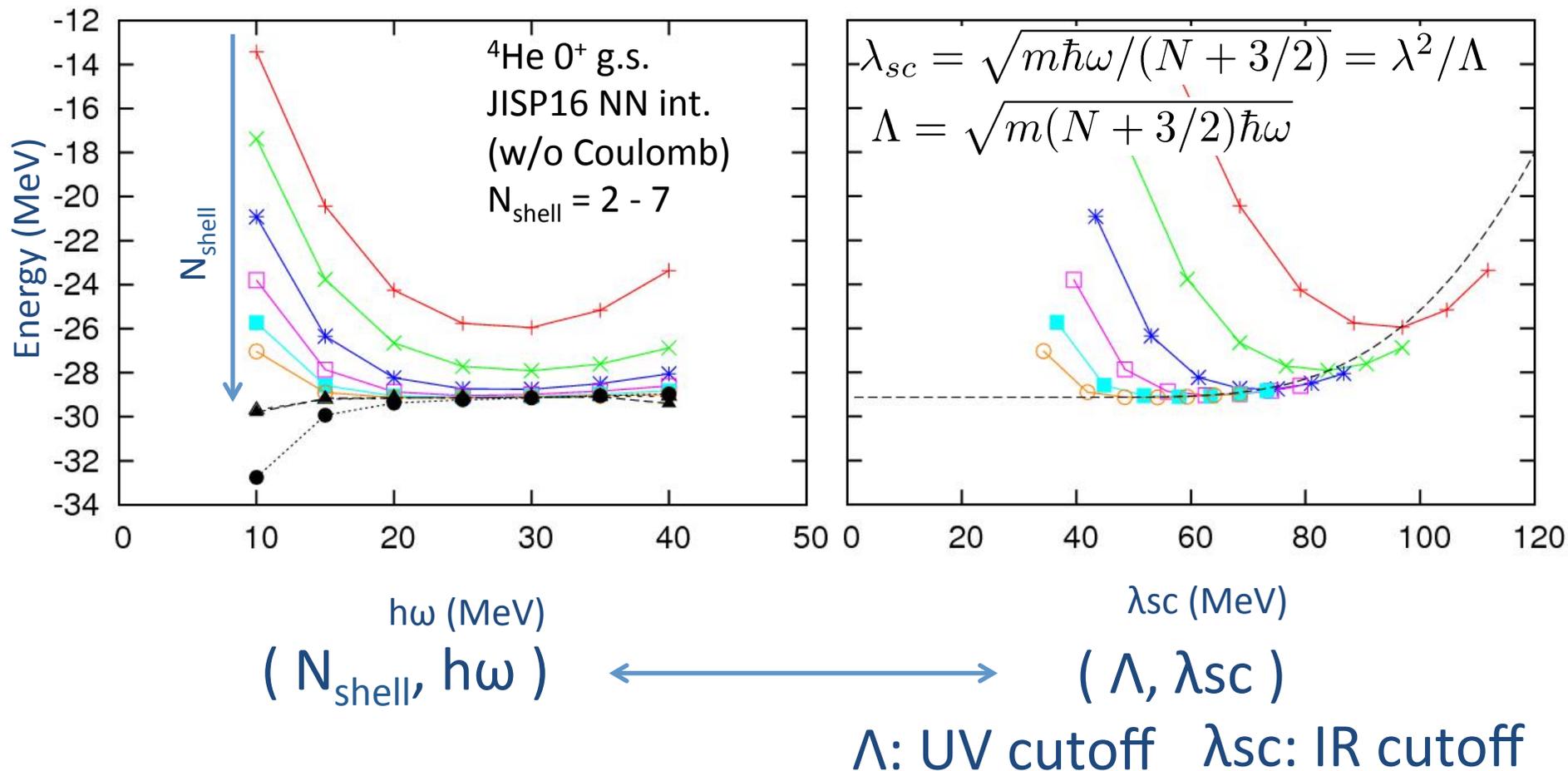
Neutron density – Proton density



模型空間(Nshell)に対する外挿法 (by 阿部)

Preliminary

Empirical & IR-cutoff extrapolations



※ モンテカルロ殻模型計算において、より大きな模型空間(Nshell)におけるより効率的な計算が必要となる。

モンテカルロ殻模型計算の概要

通常の殻模型計算

$$H = \begin{pmatrix} * & * & * & * & * & \dots \\ * & * & * & * & & \\ * & * & * & & & \\ * & * & & \ddots & & \\ * & & & & & \\ \vdots & & & & & \end{pmatrix} \xrightarrow{\text{対角化}} \begin{pmatrix} E_0 & & & & & 0 \\ & E_1 & & & & \\ & & E_2 & & & \\ & & & \ddots & & \\ & & & & & \\ 0 & & & & & \end{pmatrix}$$

全配位に対する行列要素

次元数 $\sim O(10^{10})$

モンテカルロ殻模型計算

Review: T. Otsuka, M. Honma, T. Mizusaki, N. Shimizu, Y. Utsuno, Prog. Part. Nucl. Phys. 47, 319 (2001)

$$H \sim \begin{pmatrix} * & * & \dots \\ * & \ddots & \\ \vdots & & \end{pmatrix} \xrightarrow{\text{対角化}} \begin{pmatrix} E'_0 & & 0 \\ & E'_1 & \\ 0 & & \ddots \end{pmatrix}$$

モンテカルロ的+変分計算
で基底を選択

次元数 $\sim O(100)$

エネルギー変分で求める

$$|\Phi(q, J^\pi, M, K)\rangle = P_{MK}^{J^\pi} |\Phi(q)\rangle$$

射影(projection)による
スピン・パリティ π 状態の回復

$$|\Phi(q)\rangle = \prod_i^{N_f} a_i^\dagger(q) |-\rangle = \prod_i^{N_f} \left(\sum_l^{N_s} D(q)_{li} c_l^\dagger \right) |-\rangle$$

殻模型軌道の生成演算子

Hamiltonian行列要素の計算が計算時間の大部分を占める($\sim 95\%$)

Hamiltonian行列要素の計算

Hamiltonian行列要素は密度行列 ρ と相互作用行列との行列演算となる

射影前の基底: $|\Phi(q)\rangle = \prod_i^{N_f} a_i^\dagger(q)|-\rangle = \prod_i^{N_f} \left(\sum_l^{N_s} \underline{D(q)_{li} c_l^\dagger} \right) |-\rangle$

※基底の情報は数値的には N_s (軌道数) \times N_f (粒子数) の行列で表される。



密度行列: $\rho(q', q) = D(q) \cdot (D(q')^\dagger \cdot D(q))^{-1} \cdot D(q')^\dagger$



Hamiltonian行列要素:

$$H(q', q) = \frac{N(q', q)}{\det(D(q')^\dagger \cdot D(q))} \left(\sum_{l_1 l_2}^{N_s} \underline{t_{l_1 l_2} \rho_{l_2 l_1}(q', q)} + \frac{1}{2} \sum_{l_1 l_2 l_3 l_4}^{N_s} \rho_{l_3 l_1}(q', q) \underline{\bar{v}_{l_1 l_2, l_3 l_4} \rho_{l_4 l_2}(q', q)} \right)$$

1体相互作用 2体相互作用

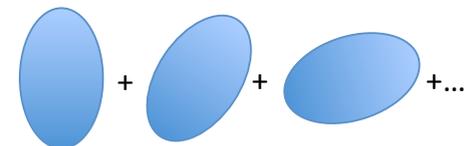
スピン(J)・パリティ(π)射影: $P_{MK}^{J\pi} |\Phi(q)\rangle$



有限角度 Ω^r の3次元回転を行った基底の重ね合わせ(角分点 N_m 個)に対応

$$H(q', q) = \sum_r^{N_m} \underline{W^r} H(q', q^r)$$

重み関数 (EulerのD関数)



ボトルネック部分の効率化

2体相互作用部分の演算量が一番多い(ボトルネック)

$$\frac{1}{2} \sum_{l_1 l_2 l_3 l_4}^{N_s} \rho_{l_3 l_1}(q', q^r) \underbrace{\bar{v}_{l_1 l_2, l_3 l_4}}_{\text{疎行列}} \rho_{l_4 l_2}(q', q^r)$$

行列 × 行列 × 行列

$$= \frac{1}{2} \sum_{\Delta m} \sum_{k' k} \tilde{\rho}(-\Delta m, q', q^r)_{k'} \underbrace{\tilde{v}(-\Delta m, \Delta m)_{k' k}}_{\text{密ブロック行列}} \tilde{\rho}(\Delta m, q', q^r)_k$$

ベクトル × 行列 × ベクトル

スピンの射影では
角分点ごとにこの演算を $H(q', q) = \sum_r W^r H(q', q^r)$
行うことになる

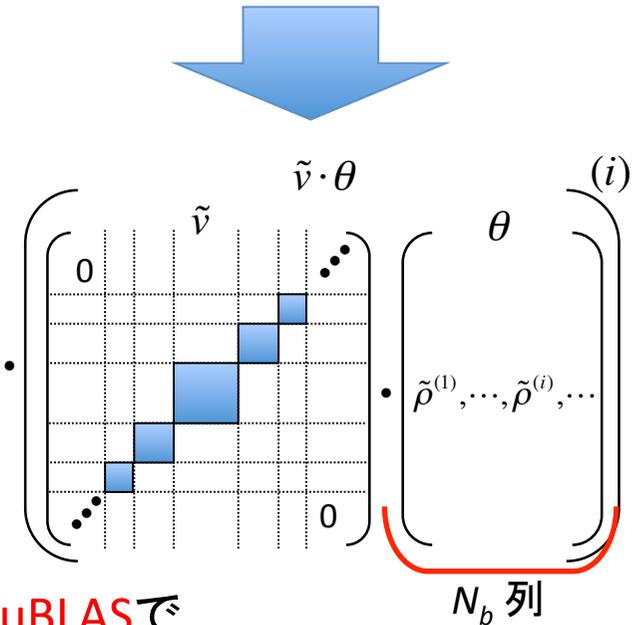
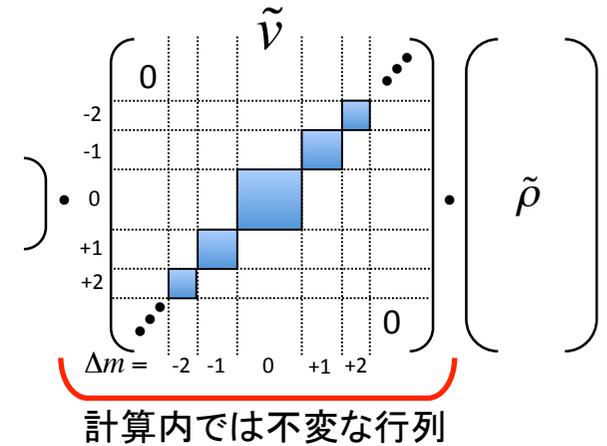
ある程度の数(N_b)の角分点の密度行列をまとめて一つの行列 θ として、2体相互作用行列との行列積を計算

行列積をBLASにより計算(従来)

Y. Utsuno, N. Shimizu, T. Otsuka, T. Abe,
Comp. Phys. Comm. 184, 102 (2013)

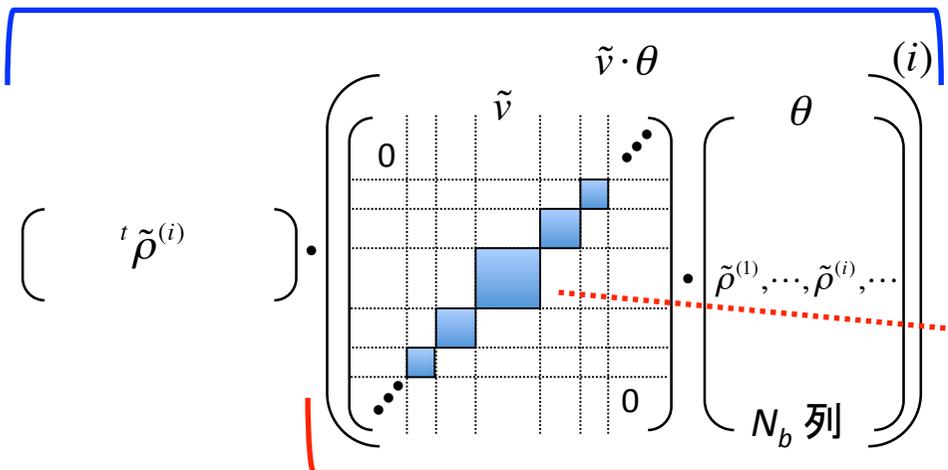


行列積を cuBLAS で
GPUにより計算

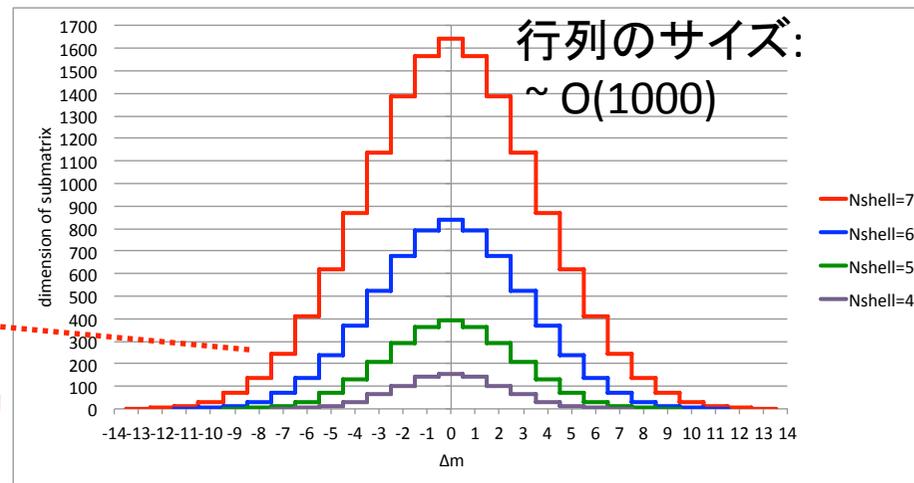


cuBLASによる演算について

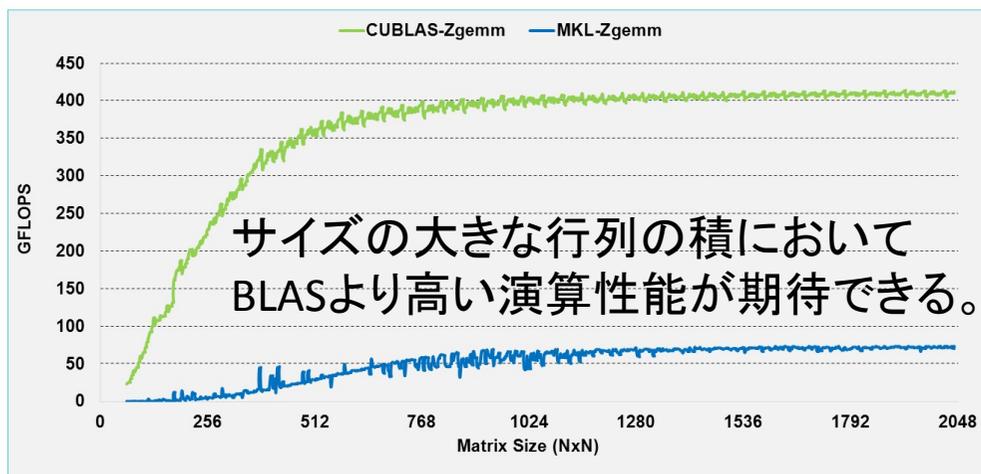
(ベクトル) x (ベクトル) を OpenACC により行う



Nshellごとの密ブロック行列のサイズ



ブロックごとに(行列) x (行列) を cuBLAS で行う



• cuBLAS 4.1 on Tesla M2090, ECC on
• MKL 10.2.3, TYAN FT72-B7015 Xeon x5680 Six-Core @ 3.33 GHz
• Performance may vary based on OS ver. and motherboard config.

Ref: NVIDIA Developer Zone (<https://developer.nvidia.com/cuBLAS>)

1行列要素の演算について

$$H(q', q) = \sum_r^{N_m} W^r H(q', q^r)$$

- 角分点の数: $N_m = 36,000$
- Chunk サイズ: $N_b = 768$
(Nshell=9では 512)
- > 47 回 chunk 分上記の演算を行う
- 1 chunk 当たり cuBLAS が呼ばれる回数: Ex) 106 in Nshell=7
- > 1 行列要素 当たり cuBLAS が呼ばれる回数: $106 \times 47 = 4,982$

GPGPU適用に対するチューニングポイント

1. CPU-GPU ハイブリッド計算 ⇒ 小さなサイズの行列演算への対応

- 密度行列 $\rho : \rho(q', q) = \underbrace{D(q)} \cdot (D(q')^\dagger \cdot D(q))^{-1} \cdot D(q')^\dagger$

N_s (軌道数): $\sim O(100) \times N_f$ (粒子数): $\sim O(10)$



密度行列 ρ は主にCPUで計算を行う

2. GPUへのデータ転送

- 計算内で不変な行列(1体,2体相互作用部分)は最初にGPUにデータを転送しておく
- 角分点ごとに可変な密度行列 ρ に関しては計算途中の行列を1つにまとめて転送する

フローチャート(単体ノード, 1GPUの場合)

(OpenMP + GPGPU) TT, N.Shimizu, Y.Utsuno, T.Abe, T.Otsuka, Procedia Computer Science 29, 1711 (2014)

$D(q'), D(q)$ が与えられる

角分点のメッシュ:



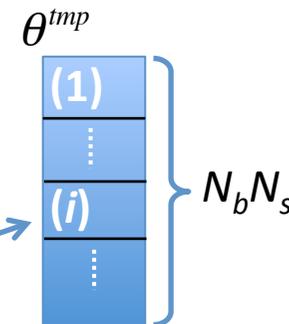
※相互作用のデータ \tilde{t}, \tilde{v} は
予めGPUに転送

(1), (2), ..., (i), ..., (N_b)のループ

OpenMPで
スレッド並列化

$D(q)$ から(i) 番目の回転角に対応した $D(q^{(i)})$ を生成し

$D(q^{(i)}) \cdot (D(q')^\dagger \cdot D(q^{(i)}))^{-1}$ まで計算を行い θ^{tmp} に保存



(GPU computing)

配列 $\theta^{tmp}, D(q')^\dagger$ をGPUに転送

※配列をまとめて一度に転送

1. $\theta^{tmp} \cdot D(q')^\dagger$ を計算し、密行列積に対応した形に置換: $\tilde{\theta} = (\tilde{\rho}^{(1)}, \dots, \tilde{\rho}^{(i)}, \dots, \tilde{\rho}^{(N_b)})$
2. Hamiltonian 行列要素 $H^0(q', q^{(i)})$ を計算: $\underbrace{(\tilde{t} \cdot \tilde{\theta})^{(i)}}_{1\text{体演算子}} + \underbrace{t \tilde{\rho}^{(i)} \cdot (\tilde{v} \cdot \tilde{\theta})^{(i)}}_{2\text{体演算子}}$

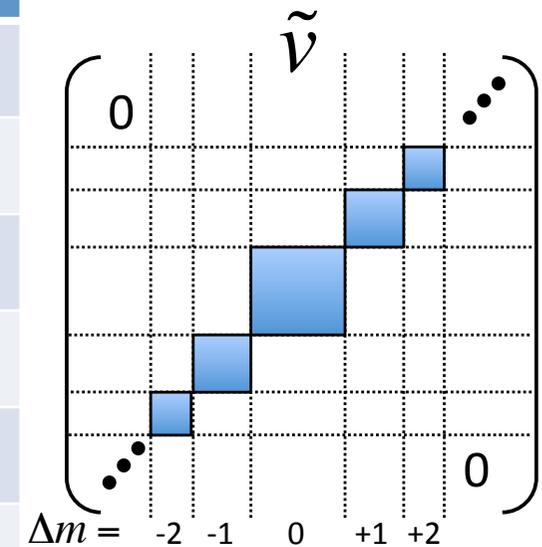


計算結果 $H^0(q', q^{(1)}), \dots, H^0(q', q^{(i)}), \dots, H^0(q', q^{(N_b)})$ をホスト側に転送

密ブロック行列(2体相互作用部分)に必要なメモリ容量

相互作用: JISP16 (A.M.Shirokov *et al.*, PLB644, 33 (2007))

Nshell	bytes
4	3M
5	22M
6	125M
7	547M
8	2.0G
9	6.4G

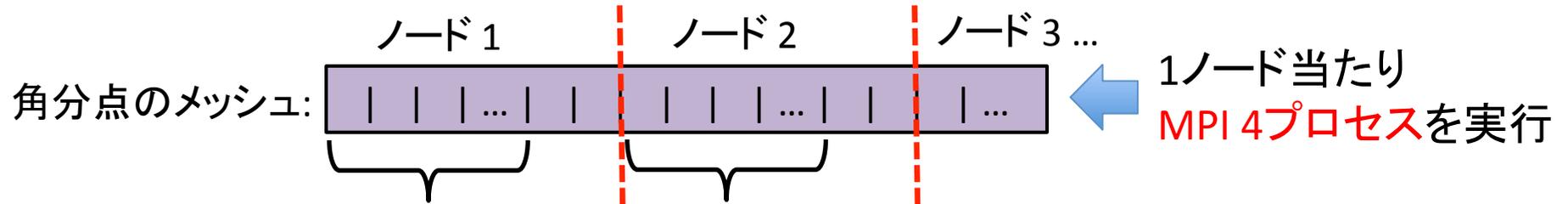


HA-PACSで使用されているGPU(NVIDIA M2090, K20X)のメモリ容量が**6GB**なので Nshell=9 以上では密ブロック行列の分割化が必要となる。

⇒ マルチGPU環境を利用した密ブロック行列の分割化を行いつつ、MPI並列によりさらなる効率化を試みる。

フローチャート(1ノード当たり4GPU) (OpenMP + MPI + GPGPU)

ノードごとに角分点メッシュを分割しそれぞれ実行

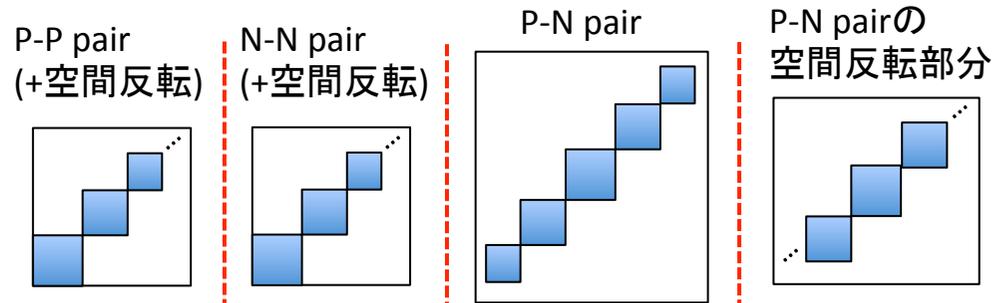
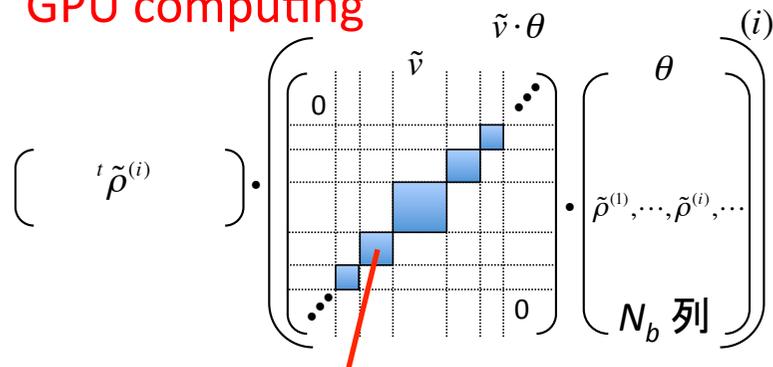


OpenMP (1), (2), ... (i), ... (N_b)のループ
1角分点当たりの
密度行列ρに関する計算

Chunkサイズ N_b

MPIプロセス x 4 ※ MPI 1プロセスに 1GPU が割り当てられる

GPU computing



密ブロック行列を分割して演算を行う → 演算量を可能な限り均等するように 4つのパートに分割する (HA-PACSでは1ノード当たり4GPUあるので)

ベンチマークテスト

対象とする原子核

^{16}O (陽子数:8, 中性子数:8), スピン・パリティ: 0^+

主殻(Nshell) : 4-9

相互作用: JISP16

→ 5基底のHamiltonian行列要素の計算(15要素)を行う

計算環境

HA-PACS base部を使用 (CPU 16core, GPU×4)

CPU: Intel E5 (SandyBridge-EP) 8core x2 -> 理論性能: 20.8 GFLOPS/core

GPU: NVIDIA M2090 -> 理論性能: 665 GFLOPS

コンパイラ:

GPU使用の場合:

PGI Accelerator Fortran 13.9 (NVIDIA CUDA 5.5 使用オプション), mvapich2 1.8.1

CPUのみを使用の場合:

Intel Composer XE 2013 SP1 14.0.3 (Intel Math Kernel Library 11.1.3を使用)

1ノード当たりの性能比較

CPUのみ(16core), 1GPU (+OpenMP: CPU 16core使用),

4GPU (+MPI 4プロセス × OpenMP: CPU 4 core) で性能を比較

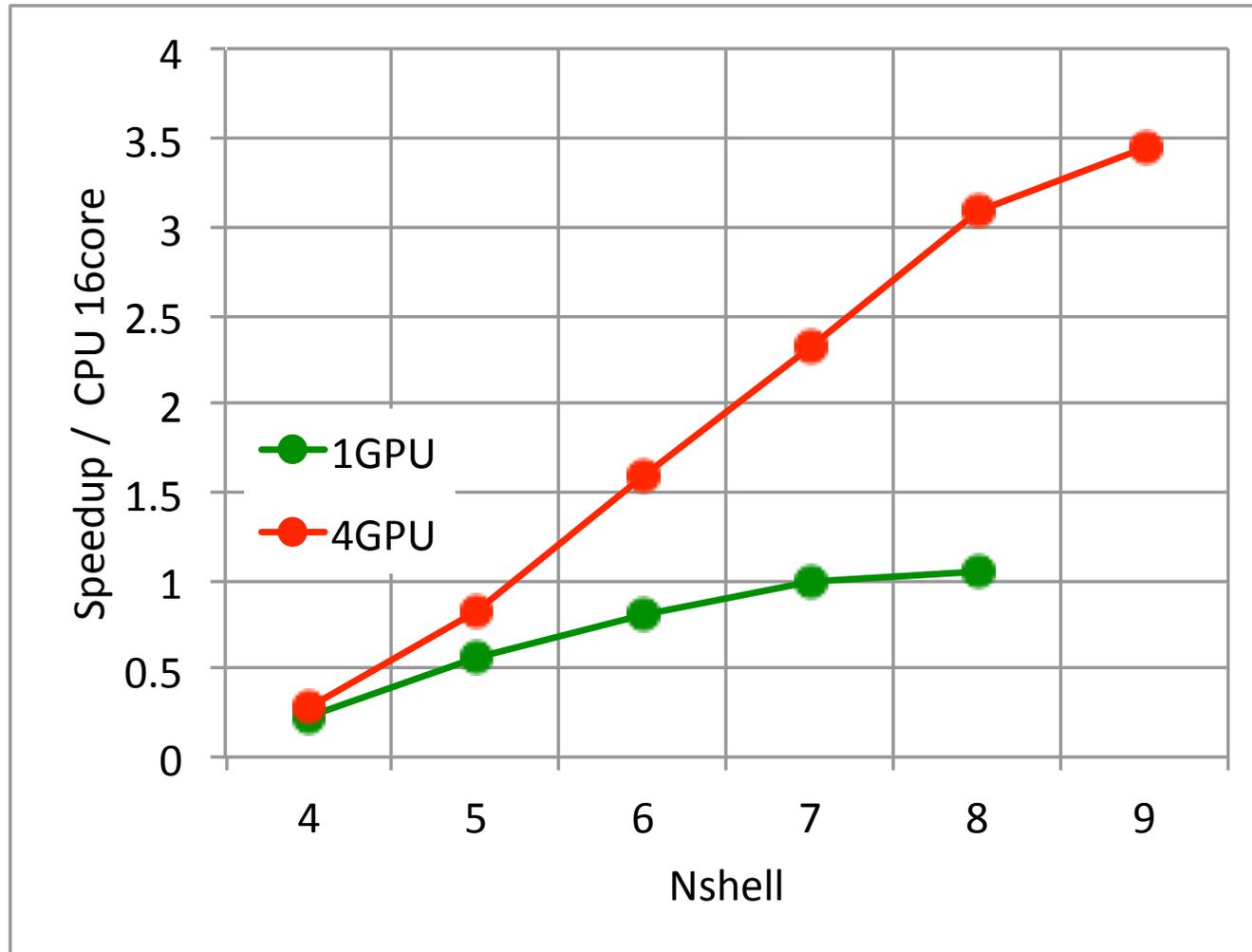
並列性能

1ノード4GPUを使用した Nshell=9 で 32ノードまでの並列性能(strong scale)を計測

1ノード計算での性能比較

CPUのみ(16core)と比較して

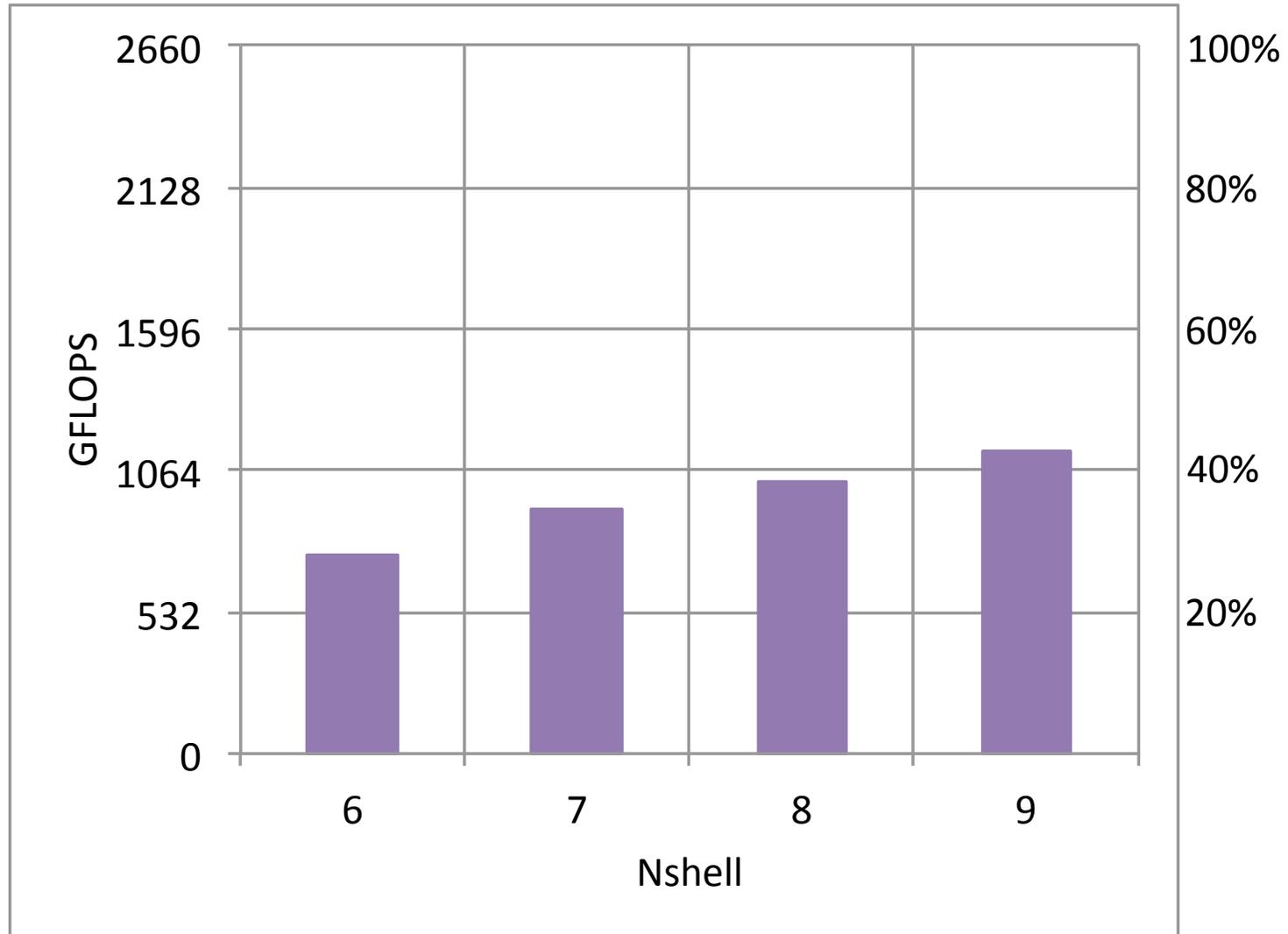
1GPU (+OpenMP:16core), 4GPU (+MPI 4プロセス × OpenMP:4core) での性能向上を図示



⇒ 4GPUの場合にNshell=6以上で性能向上が見られる

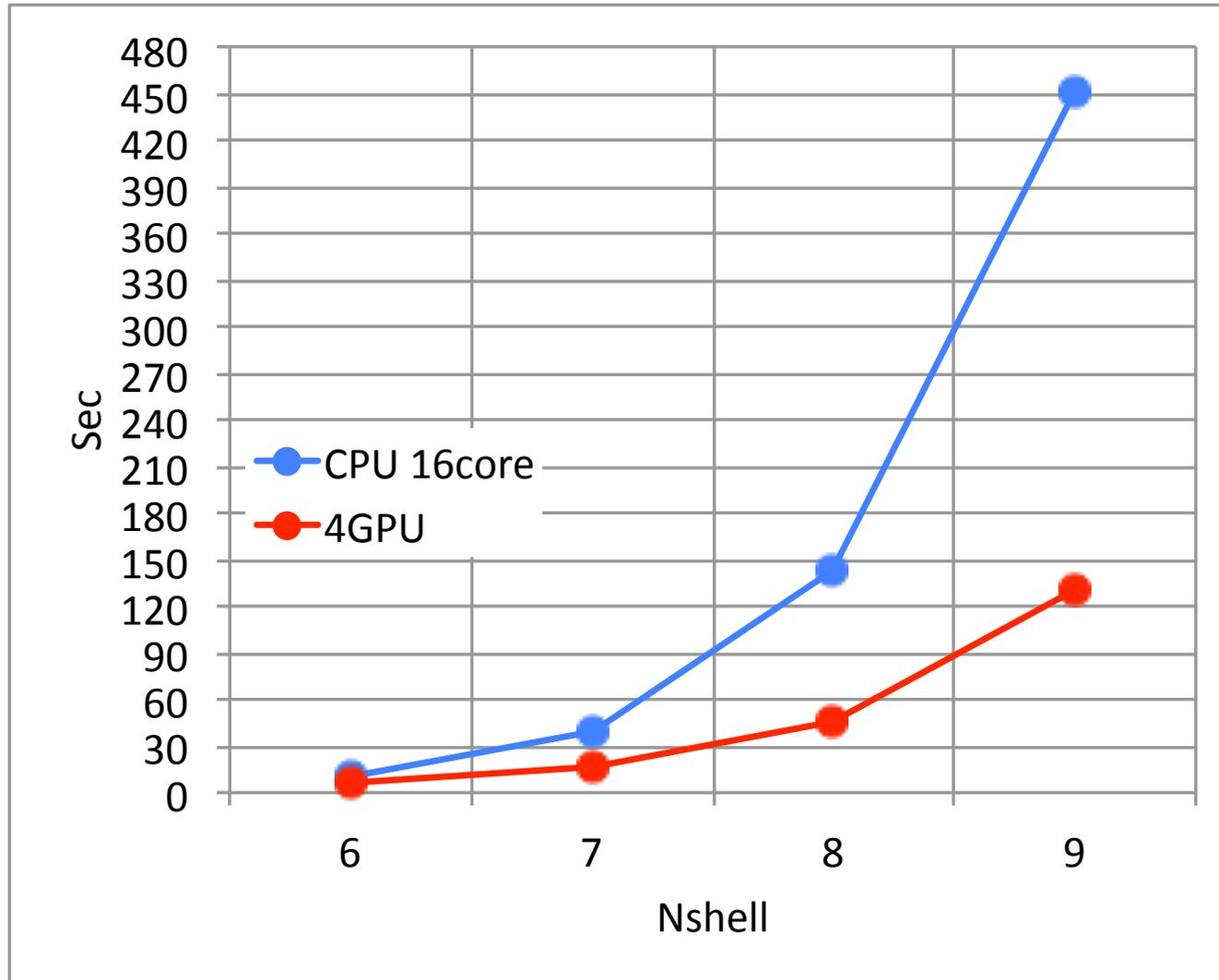
2体相互作用(ボトルネック)部分の性能計測

1ノード4GPUによる2体相互作用部分計算の性能(GFLOPS)を図示
理論ピーク性能: $665 \text{ GFLOPS} \times 4 = \underline{2,660 \text{ GFLOPS}}$



実時間計測(1ノード計算)

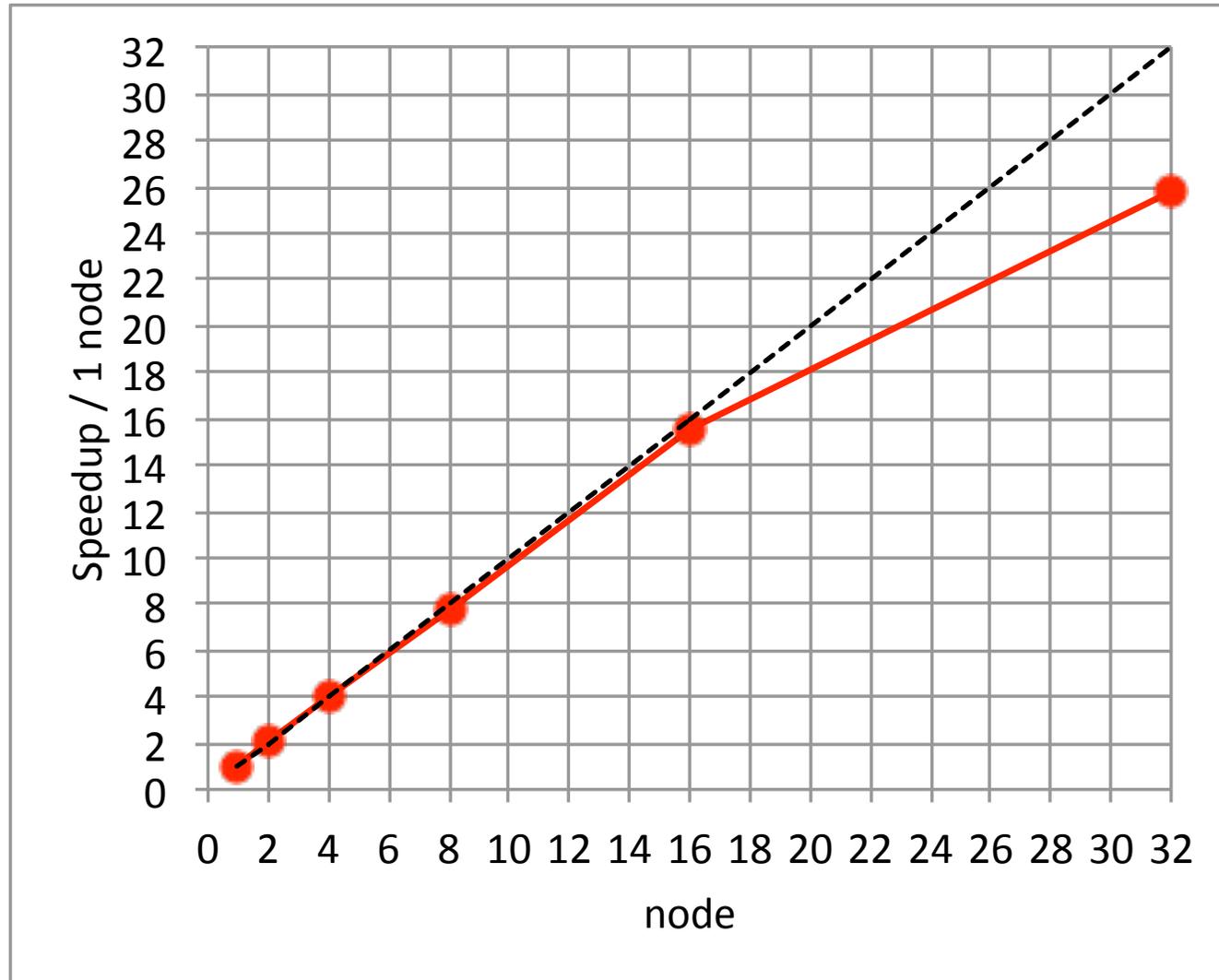
1行列要素当たりの計算にかかった実時間を図示



Nshell=6以降: CPUのみ -> Nshellが増えるごとに~3倍で増加
4GPU -> Nshellが増えるごとに~2倍で増加

並列性能 (1ノード4GPU, Nshell=9)

1ノードの場合と比較した性能向上を図示 (ノード数2,4,8,16,32で計測)



⇒ 16ノードまでのスケーリングを確認

まとめ

- マルチGPU環境を利用した2体相互作用部分の密ブロック行列の分割化を行うことで、GPGPUでNshell=9までの計算が可能となった。
- 1ノードでの計算においては、4GPUを用いたOpenMP + MPI + GPGPUでCPU 16coreの~3.5倍の性能、2体相互作用部分で~40%の実行効率が得られた(Nshell=9)
- 並列性能として16ノードまでのスケーリングを確認した。



実計算コードの実装へ

謝辞

本研究は、HPCI戦略プログラム分野5「物質と宇宙の起源と構造」
研究開発課題2「大規模量子多体計算による核物性解明とその応用」と
関連したものとなっており、ここでの成果によって上記の課題の研究も
大きく推進されることが期待されます。