



シミュレーションが 未来をひらく

チュートリアル資料

京コンピュータにおけるアプリケーションの 高並列化および高性能化 —アプリケーションの実例—

2014年8月5日



独立行政法人理化学研究所
計算科学研究機構 運用技術部門
ソフトウェア技術チーム チームヘッド

南 一生
minami_kaz@riken.jp



RIKEN ADVANCED INSTITUTE FOR COMPUTATIONAL SCIENCE

理研で進めた性能最適化

6本のターゲットアプリ

プログラム名	分野	アプリケーション概要	期待される成果	手法
NIGAM	地球科学	全球雲解像大気大循環モデル	大気大循環のエンジンとなる熱帯積雲対流活動を精緻に表現することでシミュレーションを飛躍的に進化させ、現時点では再現が難しい大気現象の解明が可能となる。(開発 東京大学,JAMSTEC,RIKEN AICS)	FDM (大気)
Seism3D	地球科学	地震波伝播・強震動シミュレーション	既存の計算機では不可能な短い周期の地震波動の解析・予測が可能となり、木造建築およびコンクリート構造物の耐震評価などに応用できる。(開発 東京大学地震研究所)	FDM (波動)
PHASE	ナノ	平面波展開第一原理電子状態解析	第一原理計算により、ポスト35nm世代ナノデバイス、非シリコン系デバイスの探索を行う。(開発 物質・材料研究機構)	平面波DFT
FrontFlow/Blue	工学	Large Eddy Simulation (LES)に基づく非定常流体解析	LES解析により、エンジニアリング上重要な乱流境界層の挙動予測を含めた高精度な流れの予測が実現できる。(開発 東京大学生産技術研究所)	FEM (流体)
RSDFE	ナノ	実空間第一原理電子状態解析	大規模第一原理計算により、10nm以下の基本ナノ素子(量子細線、分子、電極、ゲート、基盤など)の特性解析およびデバイス開発を行う。(開発 東京大学)	実空間DFT
LatticeQCD	物理	格子QCDシミュレーションによる素粒子・原子核研究	モンテカルロ法およびCG法により、物質と宇宙の起源を解明する。(開発 筑波大)	QCD



理研で進めた性能最適化



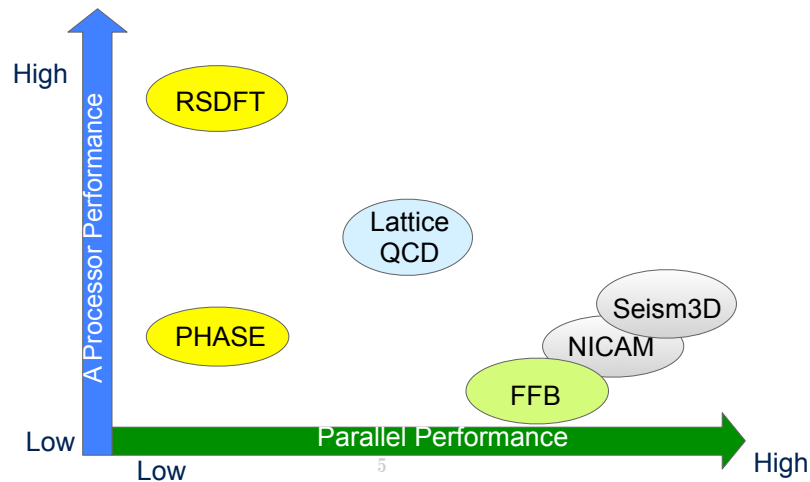
理研で進めた性能最適化

コラボレーション



理研で進めた性能最適化

6本のターゲットアプリの計算機科学的な位置づけ



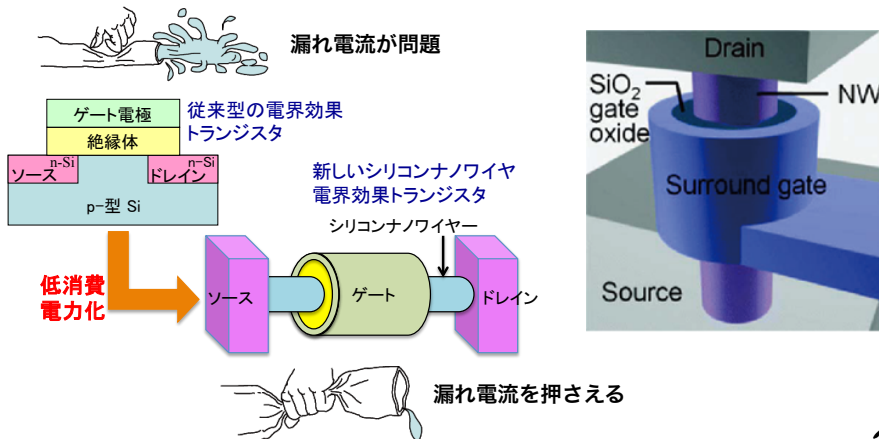
RSDFTの性能最適化



RSDFTとは

- ナノスケールでの量子論的諸現象を第一原理に立脚して解明し新機能を有するナノ物質・構造を予測

- 例えば...



RSDFTの原理

Kohn-Sham方程式

$$n(\mathbf{r}) = \sum_i |\varphi_i(\mathbf{r})|^2$$

$$\left[-\frac{1}{2}\nabla^2 + v_{\text{nucl}}(\mathbf{r}) + \int \frac{n(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}' + \frac{\delta E_{\text{xc}}[n]}{\delta n(\mathbf{r})} \right] \varphi_i(\mathbf{r}) = \varepsilon_i \varphi_i(\mathbf{r})$$

波動関数

ハミルトニアン

φ_i : 電子軌道 (=波動関数)

ε_i : 電子準位 (=エネルギーバンド)

\mathbf{r} : 空間離散点 (=空間格子)

$$H\varphi_i(\mathbf{r}) = \varepsilon_i \varphi_i(\mathbf{r}) \quad \text{固有値方程式}$$

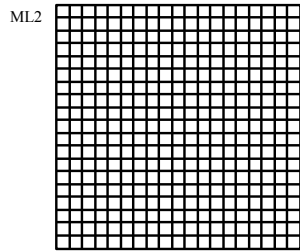


RSDFTの原理

実空間法

$$H\varphi_i(r) = \varepsilon_i\varphi_i(r) \quad \text{固有値方程式}$$

Kohn-Sham方程式を3次元格子に
離散化し差分方程式として解く



各次元方向をML1,ML2,ML3等分して格子を生成

ML(=ML1×ML2×ML3)次元のエルミート行列の固有値問題

ユニットセル (実際は3次元)



RSDFTの計算フロー

Self-Consistent Field procedure

- 1 (CG) 共役勾配法
- 2 (GS) Gram-Schmidt規格直交化
- 3 密度とポテンシャルの更新
- 4 (SD) 部分対角化

SCF計算

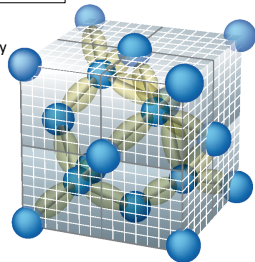


RSDFTの並列化

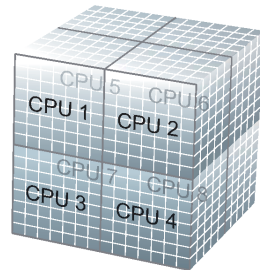
J.-I. Iwata *et al.*, J. Comp. Phys. (2010)

Real space

Blue : Si atom
Yellow: electron density



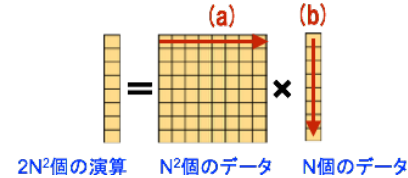
CPU space



RSDFTのCPU単体性能の向上

スレッド並列化 キャッシュの有効利用

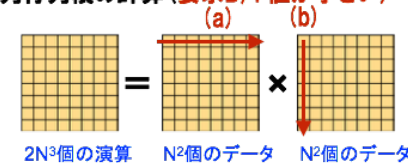
行列ベクトル積の計算 (要求B/F値が大きい)



$$\begin{aligned} \text{B/F値} &= \text{移動量(Byte)}/\text{演算量(Flop)} \\ &= (N^2+N)/2N^2 \\ &\approx 1/2 \end{aligned}$$

原理的には1/Nより大きな値

行列行列積の計算 (要求B/F値が小さい)



$$\begin{aligned} \text{B/F値} &= \text{移動量(Byte)}/\text{演算量(Flop)} \\ &= 2N^2/2N^3 \\ &= 1/N \end{aligned}$$

原理的にはNが大きい程小さな値



RSDFTのCPU単体性能の向上

RSDFT

- 実空間差分法
- 空間並列

計算コアの最適化

- 行列積化
ターゲット計算機：PACS-CS, T2K-Tsukuba

スレッド並列の実装

ターゲット計算機：PACS-CS, T2K-Tsukuba



RSDFTのCPU単体性能の向上

GramSchmidt直交化の行列積化

$$\begin{aligned} \psi'_1 &= \psi_1 \\ \psi'_2 &= \psi_2 - \langle \psi_1 | \psi_2 \rangle \psi_1 \\ \psi'_3 &= \psi_3 - \langle \psi_1 | \psi_3 \rangle \psi_1 - \langle \psi_2 | \psi_3 \rangle \psi_2 \\ \psi'_4 &= \psi_4 - \langle \psi_1 | \psi_4 \rangle \psi_1 - \langle \psi_2 | \psi_4 \rangle \psi_2 - \langle \psi_3 | \psi_4 \rangle \psi_3 \\ \psi'_5 &= \psi_5 - \langle \psi_1 | \psi_5 \rangle \psi_1 - \langle \psi_2 | \psi_5 \rangle \psi_2 - \langle \psi_3 | \psi_5 \rangle \psi_3 - \langle \psi_4 | \psi_5 \rangle \psi_4 \\ \psi'_6 &= \psi_6 - \langle \psi_1 | \psi_6 \rangle \psi_1 - \langle \psi_2 | \psi_6 \rangle \psi_2 - \langle \psi_3 | \psi_6 \rangle \psi_3 - \langle \psi_4 | \psi_6 \rangle \psi_4 - \langle \psi_5 | \psi_6 \rangle \psi_5 \\ \psi'_7 &= \psi_7 - \langle \psi_1 | \psi_7 \rangle \psi_1 - \langle \psi_2 | \psi_7 \rangle \psi_2 - \langle \psi_3 | \psi_7 \rangle \psi_3 - \langle \psi_4 | \psi_7 \rangle \psi_4 - \langle \psi_5 | \psi_7 \rangle \psi_5 - \langle \psi_6 | \psi_7 \rangle \psi_6 \\ \psi'_8 &= \psi_8 - \langle \psi_1 | \psi_8 \rangle \psi_1 - \langle \psi_2 | \psi_8 \rangle \psi_2 - \langle \psi_3 | \psi_8 \rangle \psi_3 - \langle \psi_4 | \psi_8 \rangle \psi_4 - \langle \psi_5 | \psi_8 \rangle \psi_5 - \langle \psi_6 | \psi_8 \rangle \psi_6 - \langle \psi_7 | \psi_8 \rangle \psi_7 \\ \psi'_9 &= \psi_9 - \langle \psi_1 | \psi_9 \rangle \psi_1 - \langle \psi_2 | \psi_9 \rangle \psi_2 - \langle \psi_3 | \psi_9 \rangle \psi_3 - \langle \psi_4 | \psi_9 \rangle \psi_4 - \langle \psi_5 | \psi_9 \rangle \psi_5 - \langle \psi_6 | \psi_9 \rangle \psi_6 - \langle \psi_7 | \psi_9 \rangle \psi_7 - \langle \psi_8 | \psi_9 \rangle \psi_8 \\ &\vdots \end{aligned}$$

オリジナルは行列ベクトル積



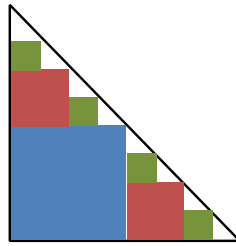
RSDFTのCPU単体性能の向上

GramSchmidt直交化の行列積化

ベクトル積を行列積に変換

再帰分割法

$$\begin{aligned} \varphi_1 &= \psi_1 \\ \varphi_2 &= \psi_2 - \varphi_1 \langle \varphi_1 | \psi_2 \rangle \\ \varphi_3 &= \psi_3 - \varphi_1 \langle \varphi_1 | \psi_3 \rangle - \varphi_2 \langle \varphi_2 | \psi_3 \rangle \\ \varphi_4 &= \psi_4 - \varphi_1 \langle \varphi_1 | \psi_4 \rangle - \varphi_2 \langle \varphi_2 | \psi_4 \rangle - \varphi_3 \langle \varphi_3 | \psi_4 \rangle \\ \varphi_5 &= \psi_5 - \varphi_1 \langle \varphi_1 | \psi_5 \rangle - \varphi_2 \langle \varphi_2 | \psi_5 \rangle - \varphi_3 \langle \varphi_3 | \psi_5 \rangle - \varphi_4 \langle \varphi_4 | \psi_5 \rangle \\ \varphi_6 &= \psi_6 - \varphi_1 \langle \varphi_1 | \psi_6 \rangle - \varphi_2 \langle \varphi_2 | \psi_6 \rangle - \varphi_3 \langle \varphi_3 | \psi_6 \rangle - \varphi_4 \langle \varphi_4 | \psi_6 \rangle - \varphi_5 \langle \varphi_5 | \psi_6 \rangle \\ \varphi_7 &= \psi_7 - \varphi_1 \langle \varphi_1 | \psi_7 \rangle - \varphi_2 \langle \varphi_2 | \psi_7 \rangle - \varphi_3 \langle \varphi_3 | \psi_7 \rangle - \varphi_4 \langle \varphi_4 | \psi_7 \rangle - \varphi_5 \langle \varphi_5 | \psi_7 \rangle - \varphi_6 \langle \varphi_6 | \psi_7 \rangle \\ \varphi_8 &= \psi_8 - \varphi_1 \langle \varphi_1 | \psi_8 \rangle - \varphi_2 \langle \varphi_2 | \psi_8 \rangle - \varphi_3 \langle \varphi_3 | \psi_8 \rangle - \varphi_4 \langle \varphi_4 | \psi_8 \rangle - \varphi_5 \langle \varphi_5 | \psi_8 \rangle - \varphi_6 \langle \varphi_6 | \psi_8 \rangle - \varphi_7 \langle \varphi_7 | \psi_8 \rangle \end{aligned}$$



三角部(DGEMV)

四角部(DGEMM)

- 依存関係のある三角部とない四角部にブロック化して計算
- 再帰的にブロック化することで四角部を多く確保



RSDFTの並列特性分析(処理・演算量)

ルーチン	処理内容	演算量	ML格子数、MB/バンド数	
			高並列化性能	単体性能
DTCG	ML×ML対称行列の固有値、固有ベクトルを共役勾配法で固有値の小さいものから順にMB本求める。 レイリー商 $\frac{\langle \psi_n H_{KS} \psi_n \rangle}{\langle \psi_n \psi_n \rangle}$ → minimize	$O(N^2)$		
GramSchmidt	規格直交化 $H_{m,n} = \langle \psi_m H_{KS} \psi_n \rangle$	$O(ML \times MB^2)$ $\rightarrow O(N^3)$		
DIAG	ML次元の部分空間に限定してハミルトニアンの対角化をする。			
	行列要素生成 (MatE)	$\psi'_n = \psi_n - \sum_{m=1}^{n-1} c_{n,m} \psi_m$	$O(ML \times MB^2)$ $\rightarrow O(N^3)$	
	固有値求解 (lodsvevd)	$\begin{pmatrix} H_{N \times N} \\ \end{pmatrix} \begin{pmatrix} \vec{c}_n \\ \end{pmatrix} = \epsilon \begin{pmatrix} \vec{c}_n \\ \end{pmatrix}$	$O(MB^3) \rightarrow O(N^3)$	
回転 (RotV)	$\psi'_n(r) = \sum_{m=1}^N c_{n,m} \psi_m(r)$	$O(ML \times MB^2)$ $\rightarrow O(N^3)$		



RSDFTの並列特性分析 (コスト)

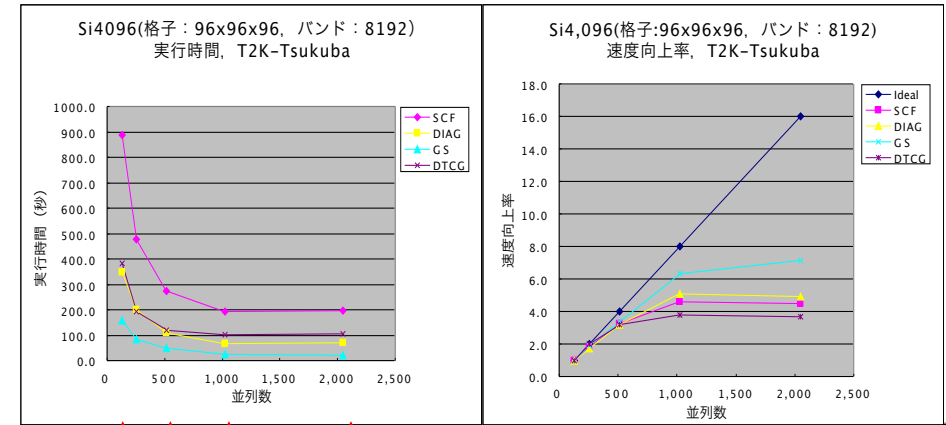
計算機 : RICC
 8,000原子 : 格子数120x120x120, バンド数16,000
 並列数 : 8x8x8 (空間方向のみ)
 SCFループ1回実行の実測データからSCFループ100回として実行時間を推定

処理内容	コスト	演算	プロセス間通信
初期化 パラメータの読み込み 全プロセスへの転送	0.4%		Bcast, lsend/recv
SCF部 DO SCFループ(100回と仮定)	99.6%	O(IN ³)	
DIAG	30.5%	O(IN ³) DGEMM中心	行列生成部: Reduce, lsend/recv (HPSI) 固有値ノルバー部: PDSYEV内(Bcast) ローテーション:部分Bcast, 部分Reduce
DTCG	27.4%	O(IN ²) 演算<ロード	スカラー値のallreduce中心 lsend/recv(ノンローカル項/HPSI) lsend/recv(境界データ交換/BCSET)
GramSchmidt	38.6%	O(IN ³) DGEMM中心	Allreduce (内積, 規格化変数)
Mixing, 途中結果の出力	3.1%		途中結果出力は毎SCFではないのでコストは もった少
ENDDO			

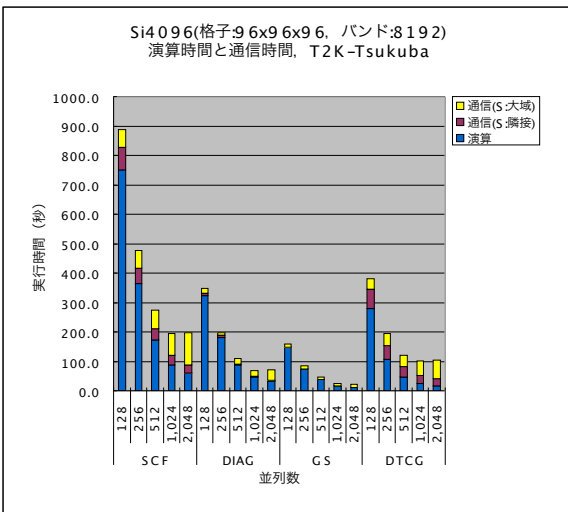


RSDFTの並列特性分析 (ブロック毎のスケラビリティ)

計算機 : T2K-Tsukuba
 コンパイラ&ライブラリ : PGI + mvapich2-medium
 S方向の分割数:128, 256, 512, 1024, 2048



RSDFTの並列特性分析 (ブロック毎のスケラビリティ)



単位: 秒

	並列数	演算	通信(S:隣接)	通信(S:大域)
SCF	128	749.472	77.435	60.057
	256	363.281	52.715	59.732
	512	172.218	38.797	64.133
	1,024	87.206	33.345	73.494
DIAG	2,048	59.728	28.637	108.404
	128	322.212	10.005	14.931
	256	181.852	6.831	9.690
	512	86.939	4.492	16.882
GS	1,024	46.392	4.061	17.659
	2,048	31.597	4.229	34.866
	128	148.354	0.000	9.629
	256	73.964	0.000	10.070
DTCG	512	37.479	0.000	9.969
	1,024	16.900	0.000	8.008
	2,048	11.451	0.000	10.687
	128	278.906	67.429	35.497
DTCG	256	107.465	45.884	39.972
	512	47.800	34.305	37.283
	1,024	23.914	29.284	47.827
	2,048	16.679	24.408	62.851

通信時間の増加が問題

※PDSYEVの通信は演算部に含まれている



並列性能上のボトルネック

- 今まで示した調査を実施することにより処理ブロック毎に並列性能上の問題がある事が発見される。
- それらを分析するとだいたい以下の6点に分類されると考える。

1	アプリケーションとハードウェアの並列度のミスマッチ (アプリケーションの並列度不足)
2	非並列部の残存
3	大域通信における大きな通信サイズ、通信回数の発生
4	フルノードにおける大域通信の発生
5	隣接通信における大きな通信サイズ、通信回数の発生
6	ロードインバランスの発生



RSDFTの並列特性分析 (並列・単体性能)

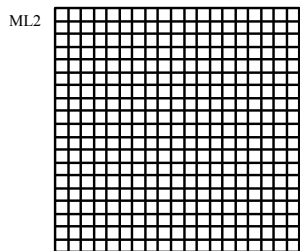
ルーチン	処理内容	演算量	高並列化性能	単体性能	
DTCG	ML×ML対称行列の固有値、固有ベクトルを共役な配法で固有値の小さいものから順にMB本求める。 → minimize $\frac{\langle \psi_n H_{KS} \psi_n \rangle}{\langle \psi_n \psi_n \rangle}$	$O(ML \times ML) \rightarrow O(N^2)$	通信時間増大 演算時間と逆転 並列度の不足	行列ベクトル積性能は悪い	
GramSchmidt	規格直変化 $H_{m,n} = \langle \psi_m H_{KS} \psi_n \rangle$	$O(ML \times MB^2) \rightarrow O(N^3)$	通信時間減少せず 演算時間と同程度 並列度の不足	行列積化で良好	
DIAG	ML次元の部分空間に 限ってハミルトニアン の対角化する。				
	行列要素生成 (MatE)	$\psi_n' = \psi_n - \sum_{m=1}^{n-1} \langle \psi_m \psi_n \rangle \psi_m$	$O(ML \times MB^2) \rightarrow O(N^3)$	通信時間増大 演算時間と同程度 並列度の不足	行列積化で良好
	固有値求解 (pdsyevd)	$H_{N \times N} \begin{pmatrix} \tilde{c}_n \\ \vdots \\ \tilde{c}_n \end{pmatrix} = \epsilon \begin{pmatrix} \tilde{c}_n \\ \vdots \\ \tilde{c}_n \end{pmatrix}$	$O(MB^3) \rightarrow O(N^3)$	Scalapackの性能 が悪い ラビリシティが悪い	Scalapackの性能 が悪い
	回転 (RotV)	$\psi_n'(r) = \sum_{m=1}^N c_{n,m} \psi_m(r)$	$O(ML \times MB^2) \rightarrow O(N^3)$		行列積化で良好

RSDFTの高並列化

固有値方程式

$$H\varphi_i(r) = \epsilon_i \varphi_i(r)$$

φ_i : 電子軌道 (=波動関数)
 i : 電子準位 (=エネルギーバンド)
 r : 空間離散点 (=空間格子)



ユニットセル (実際は3次元)

- i はエネルギーバンド量子数
- i についての依存関係はない
- 空間(s)に加えエネルギーバンド(B)の並列を実装
- 万を超える並列度を確保



RSDFTの高並列化

RSDFT

- 実空間差分法
- ベクトルの内積計算が基本
- 空間並列

計算コアの最適化

- 行列積化

ターゲット計算機: PACS-CS, T2K-Tsukuba

スレッド並列の実装

ターゲット計算機: PACS-CS, T2K-Tsukuba

超並列向けの実装

- バンド並列の拡張
- EIGENライブラリ*の適用

ターゲット計算機: K computer

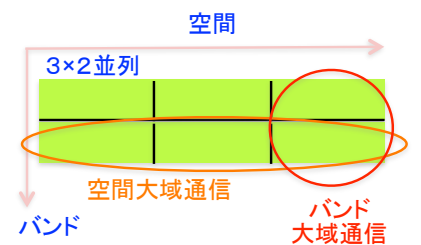
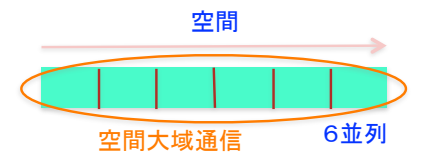
*高速固有値ライブラリ
Imamura et al. SNA+MC2010 (2010)



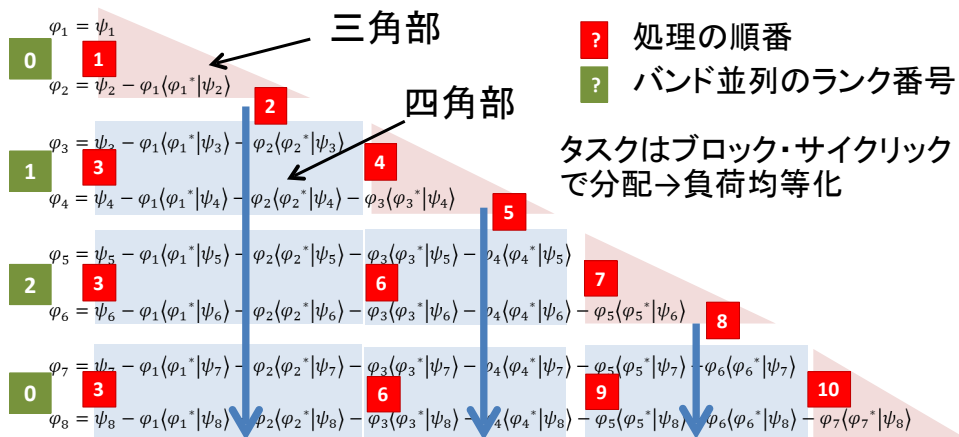
RSDFTの高並列化

並列軸拡張の効果

- 並列軸を増やす事で空間の分割粒度を増やすことができる
- 10万並列レベルに対応可能
- 空間並列のみの場合は全プロセッサ間の大域通信が必要
- 通信時間の増大を招く
- 2軸並列への書換で空間に対する大域通信が一部のプロセッサ間での通信とできる
- バンドに対する大域通信も同様
- 大域通信の効率化が実現可



RSDFTの高並列化- Gram-Schmidtの実装 -



- (1) 三角部の計算
- (2) 計算した値を四角部に転送 (バンド方向の各プロセッサに分配)
- (3) 四角部を並列に計算



RSDFTの高並列化 - 効果の確認 -

Weak Scaling 測定

タスクサイズ/プロセスを固定する。
 格子サイズ: 12x12x12, バンドサイズ: 2,400
 バンド方向の並列数は8で固定。
 空間方向を並列数に比例して増加させる。

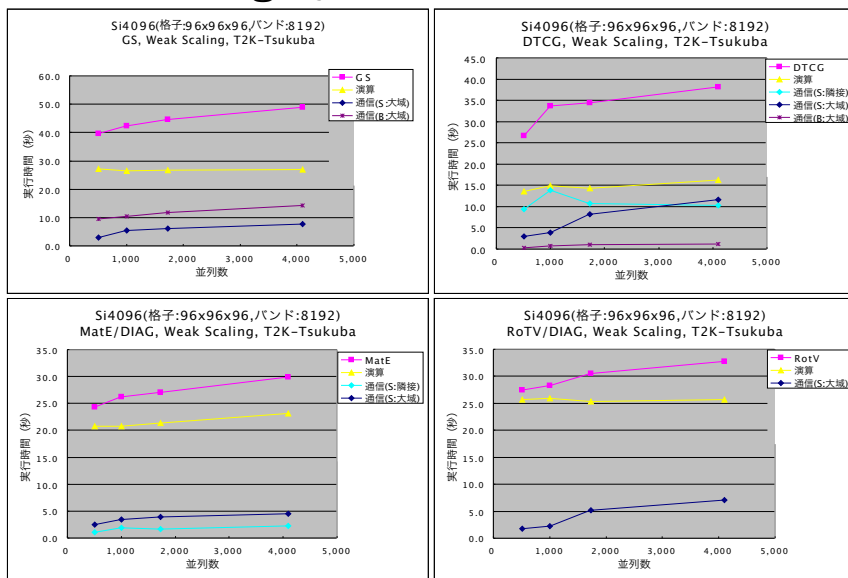
	原子数	格子数	バンド数	並列数
パターン1	512	48x48x48	19,200	512 (4x4x4x8)
パターン2	1,000	60x60x60	19,200	1,000 (5x5x5x8)
パターン3	1,728	72x72x72	19,200	1,728 (6x6x6x8)
パターン4	4,096	96x96x96	19,200	4,096 (8x8x8x8)
パターン5	8,000	120x120x120	19,200	8,000 (10x10x10x8)

T2K-Tsukubaで測定



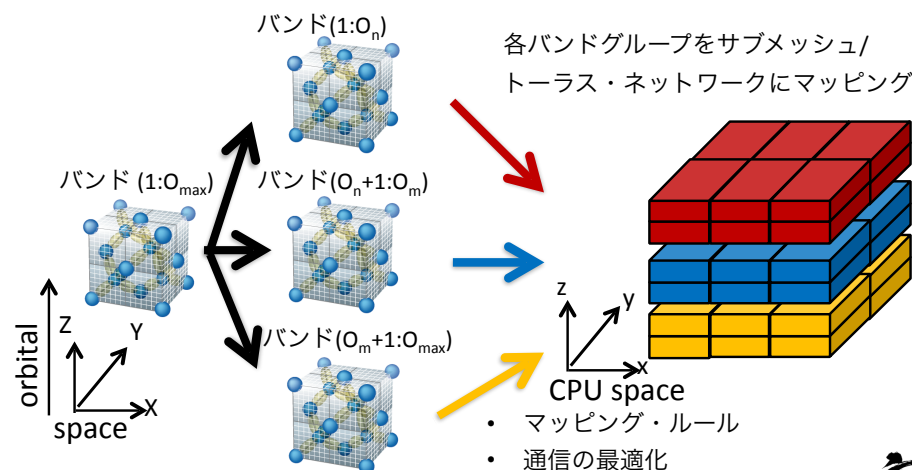
RSDFTの高並列化 - 効果の確認 -

Weak Scaling 測定



RSDFTの高並列化 - Tofuネットワークへのマッピング -

空間並列 空間並列+バンド並列 Tofuネットワークへのマッピング

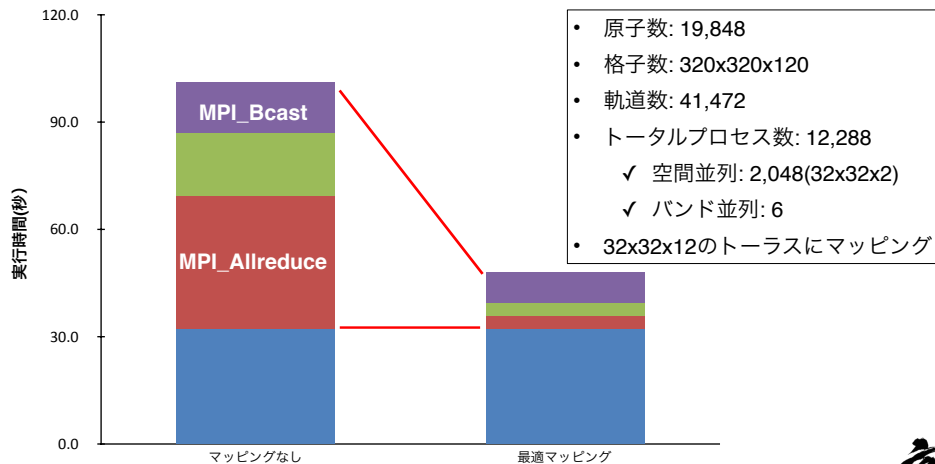


サブメッシュ/トーラス内で通信が閉じられる



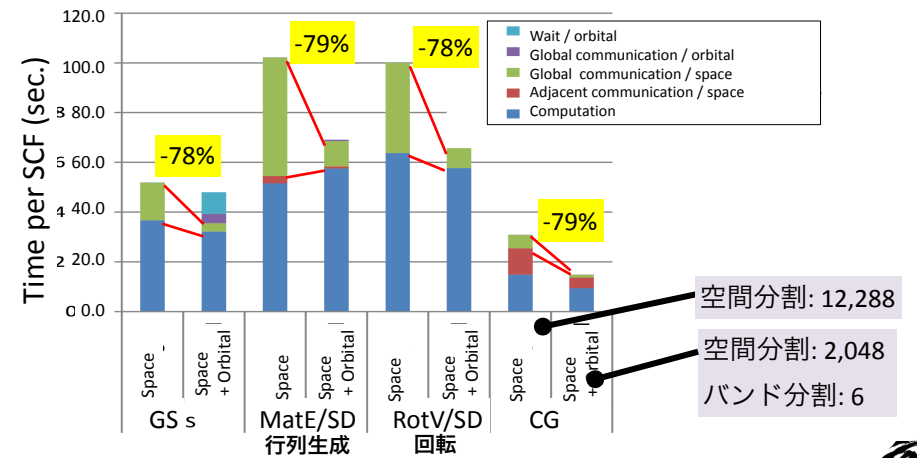
RSDFTの高並列化 -Gram-Schmidtマッピングの効果-

最適マッピング → サブコミュニケーター間のコンフリクトが発生しない
MPI通信でTofu向けアルゴリズムが選択される



RSDFTの高並列化 -二軸並列の効果-

SiNW, 19,848 原子, 格子数:320x320x120, バンド数:41,472
トータル並列プロセス数は12,288で固定



大域通信時間を大幅に削減



RSDFTの高並列化-スケラビリティ-

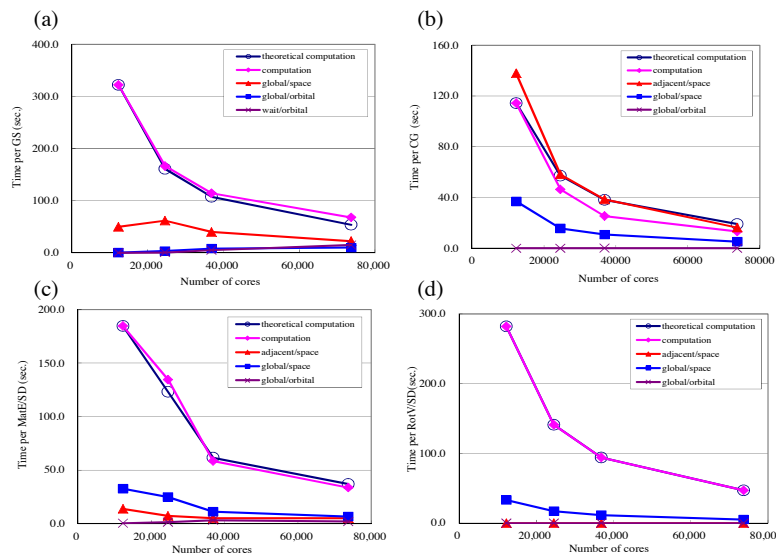


Figure 6. Computation and communication time of (a) GS, (b) CG, (c) MatE/SD and (d) RotV/SD for different numbers of cores.



RSDFTの高並列化

総合性能

Table 2. Distribution of computational costs for an iteration of the SCF calculation of the modified code.

Procedure block	Execution time (s)	Computation time (s)	Communication time (s)				Performance (PFLOPS/%)
			Adjacent/grids	Global/grids	Global/orbitals	Wait/orbitals	
SCF	2903.10	1993.89	61.73	823.02	12.57	11.89	5.48/51.67
SD	1796.97	1281.44	13.90	497.36	4.27	-	5.32/50.17
MatE/SD	525.33	363.18	13.90	143.98	4.27	-	6.15/57.93
EigenSolve/SD	492.56	240.66	-	251.90	-	-	0.01/1.03
RotV/SD	779.08	677.60	-	101.48	-	-	8.14/76.70
CG	159.97	43.28	47.83	68.85	0.01	-	0.06/0.60
GS	946.16	669.17	-	256.81	8.29	11.89	6.70/63.10

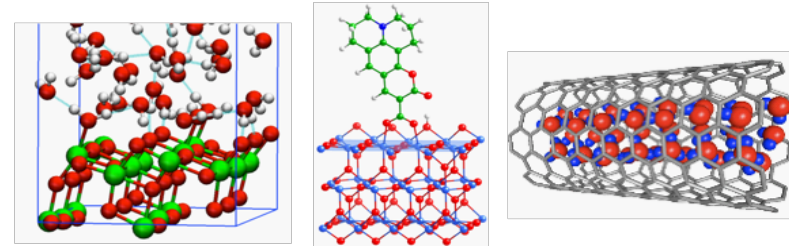
The test model was a SiNW with 107,292 atoms. The numbers of grids and orbitals were $576 \times 576 \times 180$, and 230,400, respectively. The numbers of parallel tasks in grids and orbitals were 27,648 and three, respectively, using 82,944 compute nodes. Each parallel task had 2160 grids and 76,800 orbitals.

Performance evaluation of ultra-large-scale first-principles electronic structure calculation code on the K computer

Yukihiro Hasegawa et al., *International Journal of High Performance Computing Applications* published online 17 October 2013



- ナノスケールでの量子論的諸現象を第一原理に立脚して解明し新機能を有するナノ物質・構造を予測。この点はRSDFTと同じ。
- 例えば以下のような用途に用いる。
- 繰り返し構造を持つ結晶等の解析が得意。



電子状態計算(デバイス特性, エネルギー問題, 反応・拡散), 構造緩和

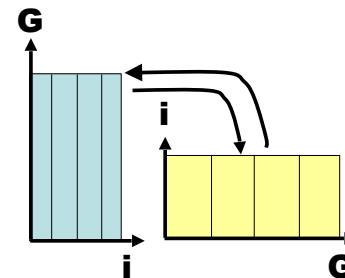
PHASEの性能最適化



PHASEの並列化

$$H\phi_{ik}(G) = \varepsilon_i\phi_{ik}(G)$$

- i はエネルギーバンド量子数
- 基本的にエネルギーバンドについて並列化されている
- 一部, 波数: G について並列化されている
- G 並列の前にエネルギーバンド並列されている波動関数を G 並列可能なようにトランスバース転送が発生
- G 並列後に G 並列されている波動関数をエネルギーバンド並列に戻すためのトランスバース転送が発生
- このトランスバース転送のコストが大



PHASEの原理

Kohn-Sham方程式

$$H\phi_i(r) = \varepsilon_i\phi_i(r)$$



← 波数: G による展開
 求めたい波動関数は未知の関数のため、既知の関数の線形結合で記述する。PHASEでは平面波基底を用いる。

$$H\phi_{ik}(G) = \varepsilon_i\phi_{ik}(G)$$

- ϕ_{ik} : 電子軌道 (=波動関数)
- i : 電子準位 (=エネルギーバンド量子数)
- G : 波数格子
- k : k 点



PHASEの並列特性分析 (処理・演算量)

■カーネルの抽出

種類	区間番号	
行列-行列積に書き換え可能	2,4,5,9,10	$O(N^3)$
FFTを含む	1,3,6,7,8,11	$O(N^2 \log N)$
対角化	9	$O(N^3)$

抽出されたカーネルは以下の11区間。

区間1: V_{local} の逆FFT

区間2: $V_{nonlocal}$ を波動関数 ψ_i と β の内積 f_i に作用

区間3: V_{local} を波動関数 ψ_i に作用, 波動関数の修正値 $H\psi_i$ を計算

区間4: $f_{ijt} = \beta \cdot \psi$ の計算

区間5: Gram-Schmidtの直交化

区間6: 固有値計算, 波動関数 ψ_i と f_i のバンド方向並べ替え

区間7: 電荷密度計算

区間8: V_{local} の逆FFT

区間9: 行列対角化計算, 波動関数 ψ_i の修正

区間10: $f_{ijt} = \beta \cdot \psi$ の計算

区間11: 電荷密度, ポテンシャル, 全エネルギー計算

以上のカーネルを計算特別に分類すると3つに分類が可能である。



PHASEの並列特性分析 (ブロック毎のスケラビリティ)

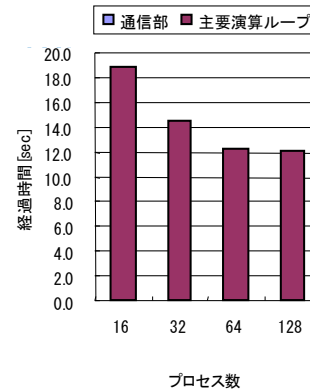
■行列積カーネル

区間2: $V_{nonlocal}$ を波動関数 ψ_i と β の内積 f_i に作用

区間4: $f_{ijt} = \beta \cdot \psi$ の計算

区間10: $f_{ijt} = \beta \cdot \psi$ の計算

- PHASEの処理ブロック:区間2を例に示す。
- 低並列でストロングスケールで測定。
- HfSiO₂ 384原子アモルファス系を測定
- すでにこの並列度でスケールしていない。
- 原因は非並列部の残存。
- 区間4, 10も同様



```

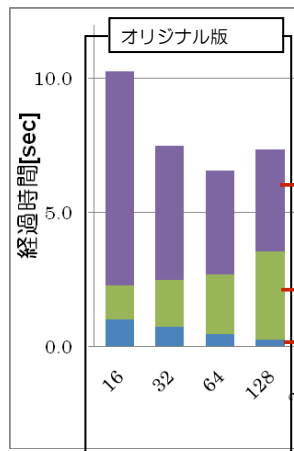
subroutine m_es_vnonlocal_w(ik,iksnl,ispin,switch_of_eko_part)
+call tstatc0_begin
loop ntyp: do it = 1, ntyp
loop natm : do ia = 1, natm -----原子数のループ
+call calc_phase
T-do lmt2 = 1, lmt(it)
+call vnonlocal_w_part sum over lmt1
+call add_vnlph_l without_eko_part
subroutine add_vnlph_l without_eko_part()
T-if (king == 1) then
T-do i = 1, np_e -----エネルギーバンド並列部
V-end do
V-end do
T-do i = 1, iba(ik) -----エネルギーバンド並列部
V-end do
V-end do
V-end if
end subroutine add_vnlph_l without_eko_part
V-end do
V-end do loop natm
V-end do loop ntyp
end subroutine m_es_vnonlocal_w
    
```



PHASEの並列特性分析 (ブロック毎のスケラビリティ)

■行列積カーネル

区間5: Gram-Schmidtの直交化



- PHASEの処理ブロック:区間5を例に示す。
- Si512原子の結果。
- 低並列でストロングスケールで測定。
- すでにこの並列度でスケールしていない。
- 演算もスケールしないが通信が増大
- トランスバース転送が原因

演算処理

通信処理

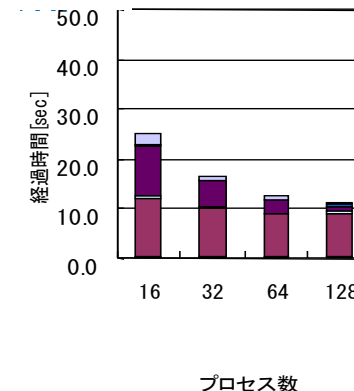
トランスバース処理
(通信以外)



PHASEの並列特性分析 (ブロック毎のスケラビリティ)

■FFTカーネル

区間8: V_{local} の逆FFT



- PHASEの処理ブロック:区間8を例に示す。
- 低並列でストロングスケールで測定。
- すでにこの並列度でスケールしていない。
- 384原子, 800程度のエネルギーバンド数。
- エネルギーバンド並列のみでは128並列にも到達しない。



PHASEの高並列化・高性能化の結果

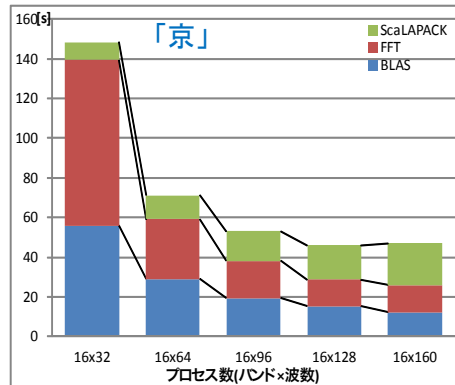
■対角化カーネル

- 対角化カーネル(区間9)
- HfSiO₂ 1,536原子, 5,120元アモルファス系で測定.
- 500並列以上で並列オーバーヘッドが測定された.

「京」

ScaLAPACKを含むカーネルの並列特性.

カーネル	512 (16x32)	1024 (16x64)	1536 (16x96)	2048 (16x128)
区間9(秒)	11.4	13.8	16.5	18.8
通信	0.0	0.0	0.0	0.0
BLAS	2.1	1.1	0.7	0.5
ScaLAPACK	8.8	12.0	15.1	17.5
他	0.5	0.7	0.7	0.8



PHASEの高並列化・高性能化

固有値方程式

$$H\varphi_{ik}(\mathbf{G}) = \varepsilon_i \varphi_{ik}(\mathbf{G})$$

φ_{ik} : 電子軌道 (=波動関数)

i : 電子準位 (=エネルギーバンド量子数)

\mathbf{G} : 波数格子

k : k 点

- エネルギーバンド(B)に加え波数(G)の並列を実装
- 完全な2軸並列とする
- RSDFTと同様の行列積化も実装



PHASEの高並列化・高性能化

■二軸並列化

■行列積カーネル

- 従来の非並列部を並列化できる.
- グラムシュミットの直交化処理のトランスバース転送が削減できる.

```

subroutine m_es_vnlocal_w(ik,iksnl,ispin,switch_of_eko_part)
+call tstatc0 begin
loop_ntyp: do it = 1, ntyp
loop_natm: do ia = 1, natm -----原子数のループ
+call calc_phase
T-do lmt2 = 1, ilmt(it)
+call vnlocal_w_part sum over lmt1
+call add_vnlp1_without_eko_part
subroutine add_vnlp1_without_eko_part()
T-if(kimg == 1) then
T-do ib = 1, np_e -----エネルギーバンド並列部
T-do i = 1, iba(ik)
V-end do
V-end do
--else
T-do ib = 1, np_e -----エネルギーバンド並列部
T-do i = 1, iba(ik)
V-end do
V-end do
V-enddo
V-end if
end subroutine add_vnlp1_without_eko_part
V-end do
V-end do loop_natm
V-end do loop_ntyp
end subroutine m_es_vnlocal_w
    
```

非並列部が波数で並列化できる

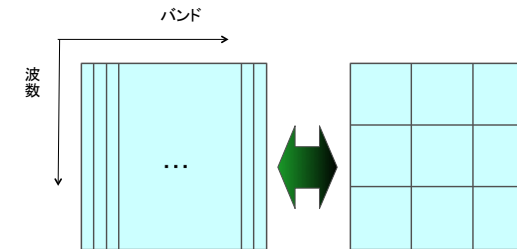


PHASEの高並列化・高性能化

■二軸並列化

■行列積カーネル

- 分割粒度が大きくなる.
- ループの回転長が増えることで、並列性能が高まる.
- cf. バンド方向のループ長が1/9から1/3と3倍に増える.

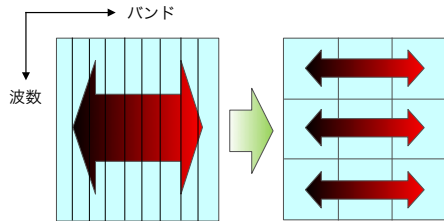


PHASEの高並列化・高性能化

■二軸並列化

■行列積カーネル

- バンド方向の大域通信が一部のプロセッサに閉じるため通信時間の短縮が図れる。

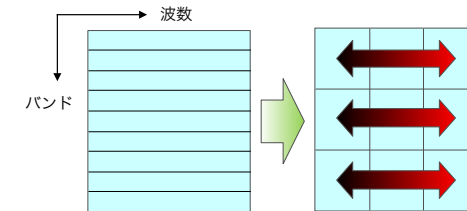


PHASEの高並列化・高性能化

■二軸並列化

■FFTカーネル

- オリジナルのFFTカーネルは波数方向に並列化されていなかった。
- そのためFFTに関する通信は発生していなかった。
- 二軸並列化に伴いFFTに関する通信が発生する。
- FFT通信の問題は全プロセッサ間の転置通信。
- 二軸並列化では通信は全プロセッサでなく一部のプロセッサに閉じる。



PHASEの高並列化・高性能化

■二軸並列化

- この分野では小規模問題を短時間で計算したいという科学的な要求が高い。
- バンド計算(エネルギー準位など)：1万原子の1回SCF収束で良い~100SCF程度。
- 構造緩和(MD)や反応経路探索：外側に原子核の緩和に関するループ構造~100step程度。
- 10,000原子を10PFシステム(80,000ノード), また1,000原子を10,000ノードで計算する事を目指せる。
- ただし二軸並列はメリットとデメリットがあるため実施前に効果が期待できるか詳細な評価を実施した。



PHASEの高並列化・高性能化の結果

スレッド並列化

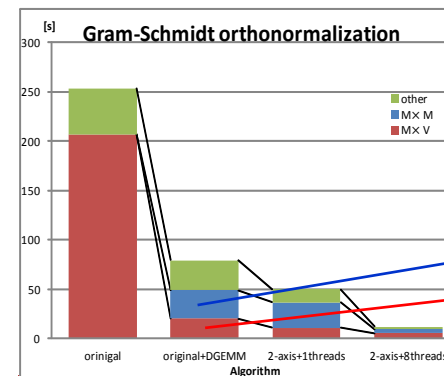
キャッシュの有効利用-行列積化

「京」

サブルーチン	時間 [sec]	比率 [%]	演算効率 [%]
区間5	11.8	100.0	23.07
m_ES_F_transpose_r	0.0	0.1	0.00
m_ES_W_transpose_r	0.1	0.5	0.00
WSW_t	1.2	9.8	0.10
normalize_bp_and_psi_t	0.4	3.3	0.13
W1SW2_t_r	3.7	31.4	2.30
modify_bp_and_psi_t_r	1.7	14.8	3.75
W1SW2_t_r_block	2.3	19.4	56.82
modify_bp_and_psi_t_r_block	2.1	17.6	60.90
m_ES_W_transpose_back_r	0.0	0.3	0.00
m_ES_F_transpose_back_r	0.0	0.1	0.00

Gram-Schmidt直交化のBLAS Level3適用結果

「京」



- 行列積
- 行列ベクトル積

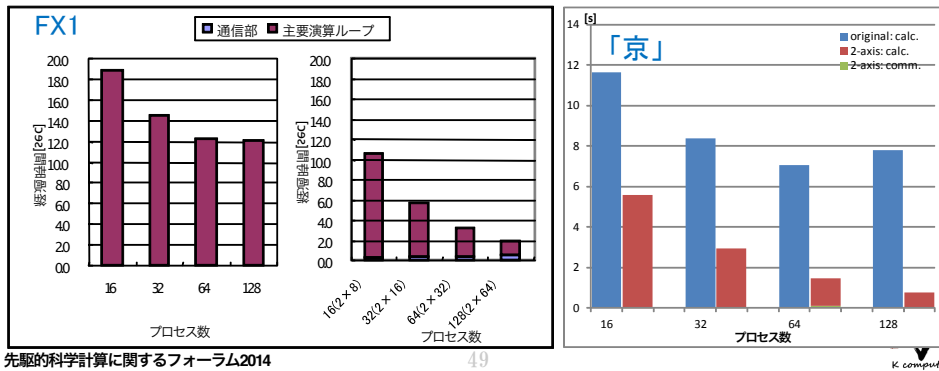


PHASEの高並列化・高性能化の結果

■二軸並列化

■行列積カーネル

- 行列積化されたカーネルに(区間2)についての結果.
- HfSiO₂ 384原子アモルファス系のデータ.
- 大幅な性能向上を達成.

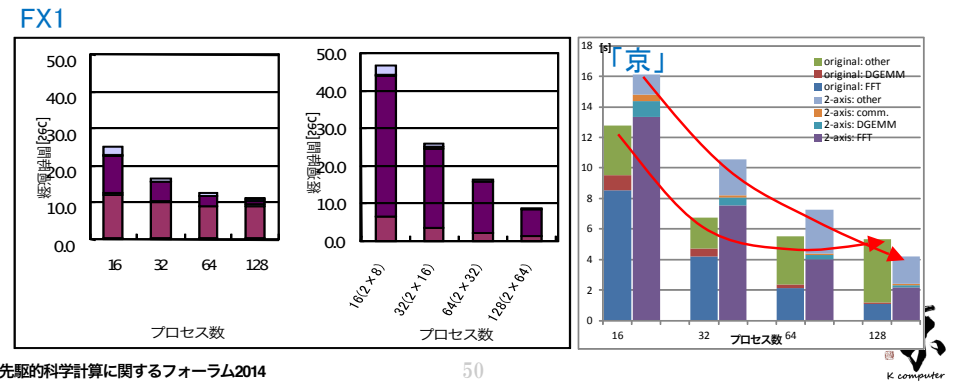


PHASEの高並列化・高性能化の結果

■二軸並列化

■FFTカーネル

- FFTを含むカーネルに(区間8)についての結果.
- HfSiO₂ 384原子アモルファス系のデータ.
- 性能向上を達成.

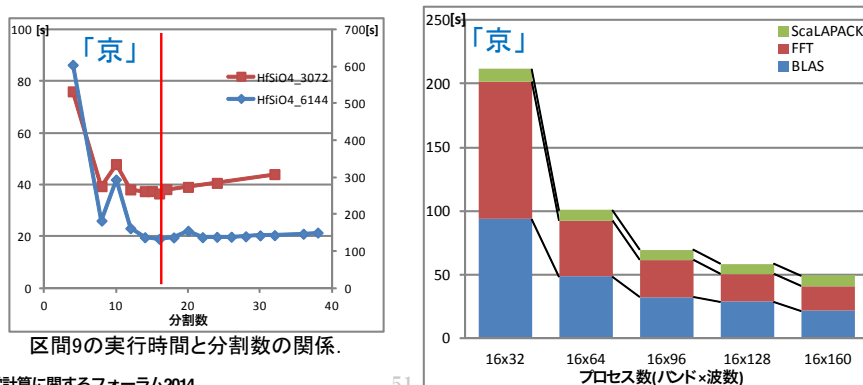


PHASEの高並列化・高性能化の結果

■Scalapack分割数の固定

■対角化カーネル

- 対角化はエネルギーバンド数の元を持つ行列が対象
- 行列の大きさに比べて分割数が多すぎる
- 分割数を16x16=256に固定



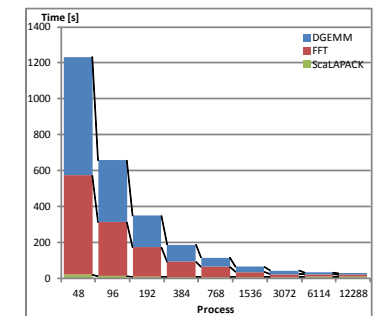
PHASEの高並列化・高性能化の結果

総合性能

- 「京」3,072並列にて、SiC 4,096原子計算にて、構造緩和(263MD, 2days).
- 「京」82,944並列にて、SiC 20,440原子計算にて、MSDソルバー効率 20.2 % (2.1 PFLOPS) 達成.

「京」で測定した並列性能(SiC 3,800原子系)

Kernel	Time [sec]	Efficiency of theoretical Peak
SCF	39.78	20.11%
DGEMM	13.19	49.73%
FFT	14.46	7.86%
ScaLAPACK	12.31	3.88%



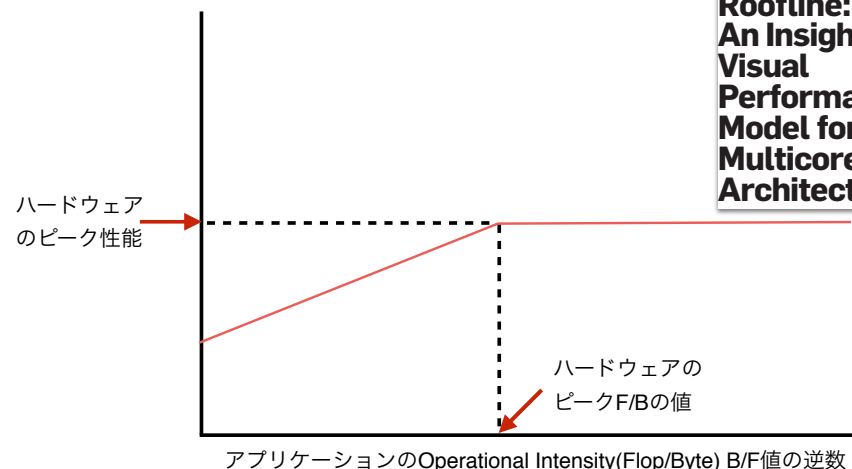
性能予測手法 (要求B/F値が大きい場合)



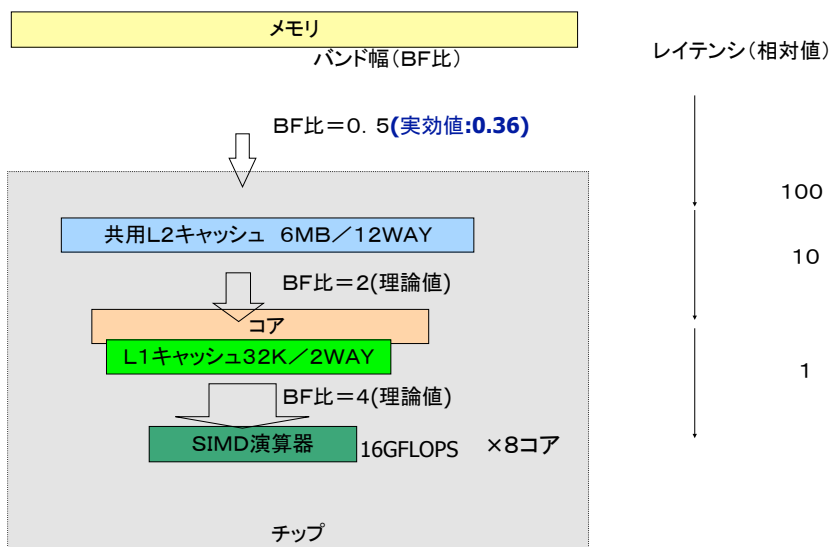
ルーフラインモデル

The Roofline model offers insight on how to improve the performance of software and hardware.
BY SAMUEL WILLIAMS, ANDREW WATERMAN, AND DAVID PATTERSON

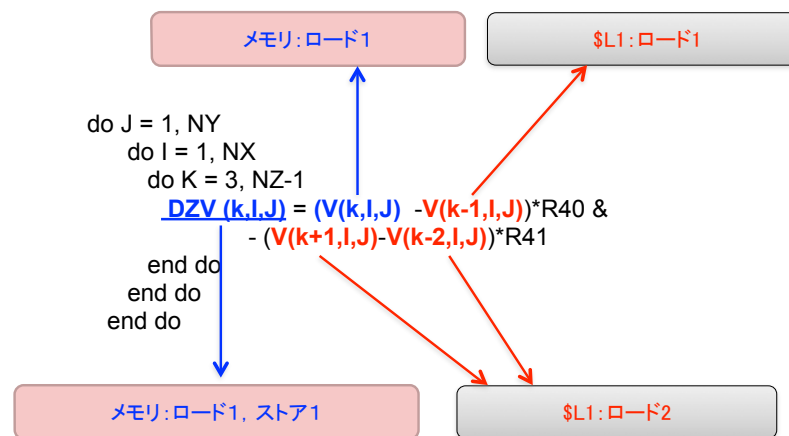
Roofline:
An Insightful
Visual
Performance
Model for
Multicore
Architectures



ベースとなる性能値



メモリとキャッシュアクセス(1)



性能見積り

```

do J = 1, NY
  do I = 1, NX
    do K = 3, NZ-1
      DZV (k,I,J) = (V(k,I,J) - V(k-1,I,J))*R40 &
        - (V(k+1,I,J)-V(k-2,I,J))*R41
    end do
  end do
end do
    
```

- 最内軸(K軸)が差分
- 1ストリームでその他の3配列は\$L1に載っており再利用できる。

要求Byteの算出:

1store,2loadと考える

4x3 = 12byte

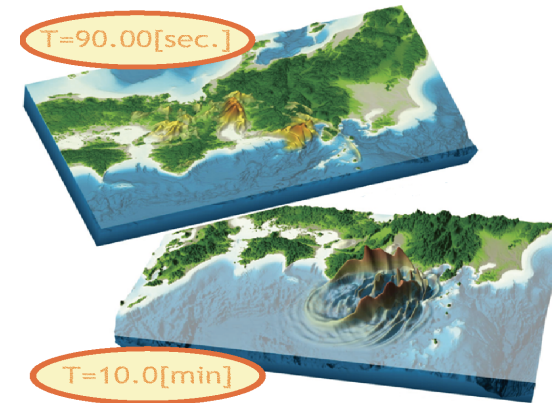
要求flop:

add : 3 mult : 2 = 5

要求B/F	12/5 = 2.4
性能予測	0.36/2.4 = 0.15
実測値	0.153



Seism3Dの性能最適化



Seism3D

- 地震伝播を解くアプリケーション
- 有限差分法により数値的に粘弾性方程式を時間発展させる
- 現在は地震伝播と津波を連動して解く事が可能
- 大規模な並列化に対応しているアプリケーション
- 以下の6つの計算部分より構成される

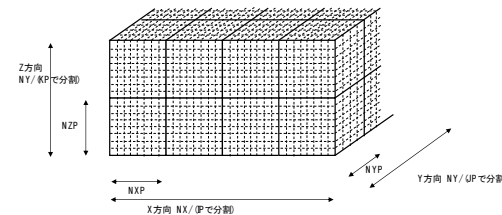


- 応力空間微分計算
- 速度空間微分計算
- 応力時間積分計算
- 応力時間積分吸収計算
- 速度時間積分計算
- 速度時間積分吸収計算



Seism3Dの並列化

3次元の領域分割

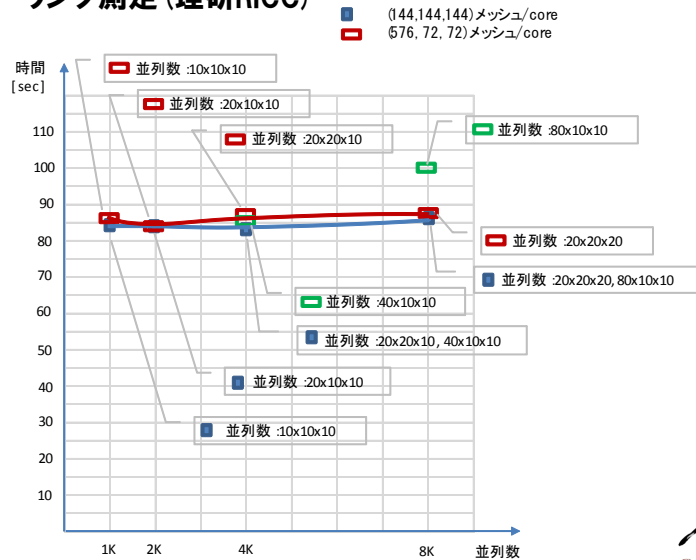


当初の評価アプリ



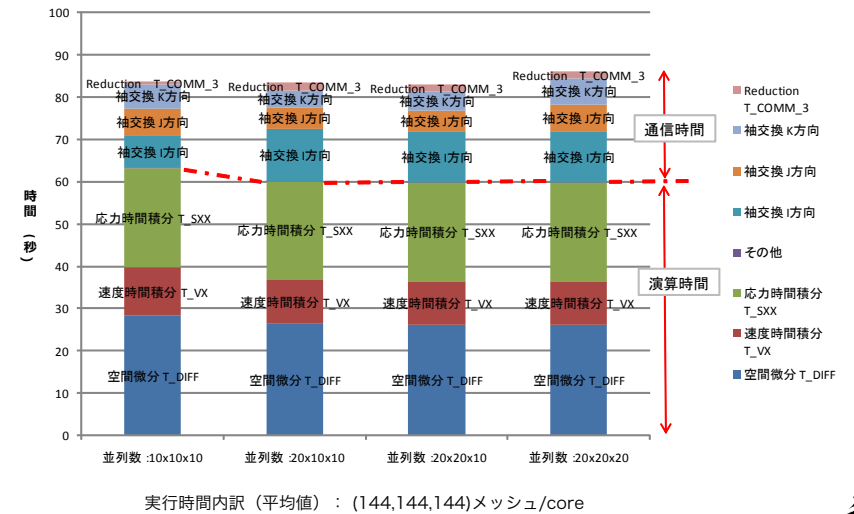
Seism3Dの並列特性分析

ウィークスケーリング測定 (理研RICC)



Seism3Dの並列特性分析

ウィークスケーリング測定 (理研RICC)



Seism3Dの並列特性分析

当初の評価アプリについて

- a) 応力空間微分計算
- b) 速度空間微分計算
- c) 応力時間積分計算
- d) 応力時間積分吸収計算
- e) 速度時間積分計算
- f) 速度時間積分吸収計算

演算量 高並列性能

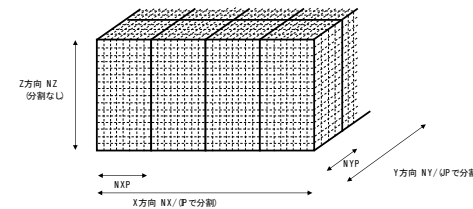
O(N)
O(N)
O(N)
O(N)

完全並列
ウィークスケールしている
隣接通信時間が通信量に比べると大域通信が並列数に応じ増大傾向



Seism3Dの並列化

2次元の領域分割



新しい評価アプリ



- ・ 今後計算体系の上面のみで実行される津波の計算を鑑み2次元分割が計画された
- ・ 2次元分割での通信時間を評価
- ・ 通信量は4-5倍になるが通信回数は2/3になると評価
- ・ 通信のパッキングは2次元が有利
- ・ 通信時間は演算時間の数%ですむと評価



空間微分Z方向の計算a) b) (1次元目の差分)

```
do J = 1, NY
  do I = 1, NX
    do K = 3, NZ-1
      DZV (k,I,J) = (V(k,I,J) -V(k-1,I,J))*R40 &
        - (V(k+1,I,J)-V(k-2,I,J))*R41
    end do
  end do
end do
```

要求Byteの算出:

1store,2loadと考える

4x3 = 12byte

要求flop:

add : 3 mult : 2 = 5

要求B/F	12/5 = 2.4
性能予測	0.36/2.4 = 0.15
実測値	0.153



空間微分X方向の計算a) b) (2次元目の差分)

```
do J = 1, NY
  do I = 1, NX
    do K = 1, NZ
      DXV (k,I,J) = (V(k,I,J) -V(k,I-1,J))*R40&
        - (V(k,I+1,J)-V(k,I-2,J))*R41
    end do
  end do
end do
```

- 第2軸(I軸)が差分
- 1ストリームでその他の3配列は \$L1or\$L2に載っており再利用できる
- 従って1次元目が差分のパターンと同じ性能になる

要求Byteの算出:

P12より、メモリコストだけを考慮する。

1store,2loadと考える

4x3 = 12byte

要求flop:

add : 3 mult : 2 = 5

要求B/F	12/5 = 2.4
性能予測	0.36/2.4 = 0.15
実測値	0.135

- 実測値が13.5%と少し低い
- 14.0%に比較すれば13.5%は良い値
- 実測したメモリバンド幅は42.9GB/sec
- L2キャッシュ負荷の増大によりメモリバンド幅が下がった可能性(京特有の現象)
- この値で性能予測をすると14.0%となる



空間微分Y方向の計算a) b) (3次元目の差分)

```
do J = 1, NY
  do I = 1, NX
    do K = 1, NZ
      DYV (k,I,J) = (V(k,I,J) -V(k,I,J-1))*R40 &
        - (V(k,I,J+1)-V(k,I,J-2))*R41
    end do
  end do
end do
```

- 第3軸が差分 → 再利用性なし

要求flop:

add : 3 mult : 2 = 5

要求Byteの算出:

1store/5loadより

(5+1) * 4byte = 24

要求B/F	24/5 = 4.8
性能予測	0.36/4.8 = 0.075
実測値	0.076



空間微分Y方向の計算a) b) (3次元目をcyclicでスレッド並列化)

```
!$OMP DO SCHEDULE(static,1),PRIVATE(I,J,K)
do J = 1, NY
  do I = 1, NX
    do K = 1, NZ
      DYV (k,I,J) = (V(k,I,J) -V(k,I,J-1))*R40 &
        - (V(k,I,J+1)-V(k,I,J-2))*R41
    end do
  end do
end do
```

キャッシュに載せる

- 第3軸をcyclic分割 → 1ストリームで3配列がL2に乗る(説明次項)
- 性能が2倍になる

要求Byteの算出:

1store,2loadと考える

4x3 = 12byte

要求flop:

add : 3 mult : 2 = 5

要求B/F	12/5 = 2.4
性能予測	0.36/2.4 = 0.15
実測値	0.136



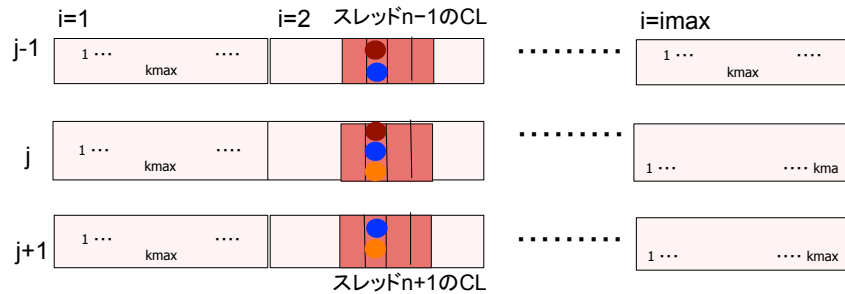
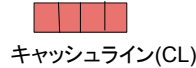
(cyclic分割スレッド並列の説明)

プログラム例

```

Do j=1,jmax
do i=1,imax
do k=1,kmax
a(k,i,j)=...c0*v(k,i,j-1)+c1* v(k,i,j)+c2* v(k,i,j+1)...
end do
end do
end do
    
```

- :スレッドn-1で参照するデータ
- :スレッドnで参照するデータ
- :スレッドn+1で参照するデータ



空間微分Y方向の計算 a) b) (ZXYループ融合cyclicスレッド並列)

```

!$OMP DO SCHEDULE(static,1)
do J = 1, NY
do I = 1, NX
do K = 3, NZ-1
DZV (k,l,J) = (V(k,l,J) -V(k-1,l,J))*R42 &
- (V(k+1,l,J)-V(k-2,l,J))*R43
DXV (k,l,J) = (V(k,l,J) -V(k,l-1,J))*R40&
- (V(k,l+1,J)-V(k,l-2,J))*R41
DYV (k,l,J) = (V(k,l,J) -V(k,l,J-1))*R40 &
- (V(k,l,J+1)-V(k,l,J-2))*R41
end do
end do
end do
    
```

要求B/F値を下げる
キャッシュに載せる

- K,l,J軸差分のループを融合することにより、V(K,l,J)のロードを共通化でき、プログラムの要求B/F比を下げる。

要求Byteの算出:

Store 3 + 4 load と考えると、

$(3+4)*4 = 28\text{byte}$

要求flop:

add : 9 mult : 6 = 15

要求B/F	28/15 = 1.86
性能予測	0.36/1.86 = 0.19
実測値	0.177



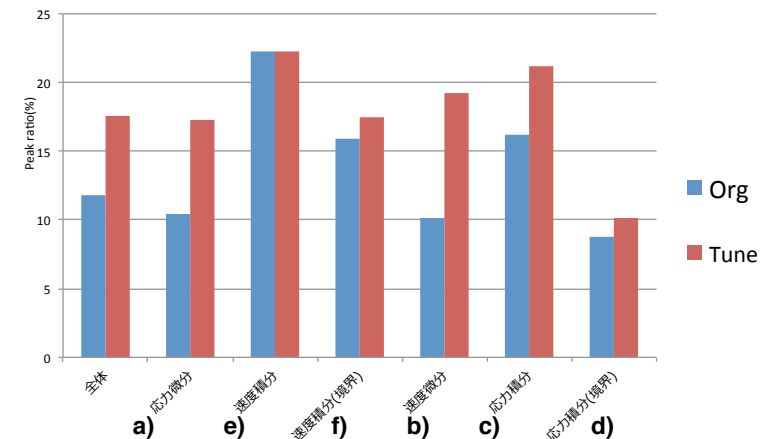
速度時間積分の計算 e)

オリジナルコードの結果

要求B/F	72/52=1.38
性能予測	0.36/1.38 = 0.26
実測値	0.240



CPU単体性能チューニングの結果



Seism3Dの総合性能

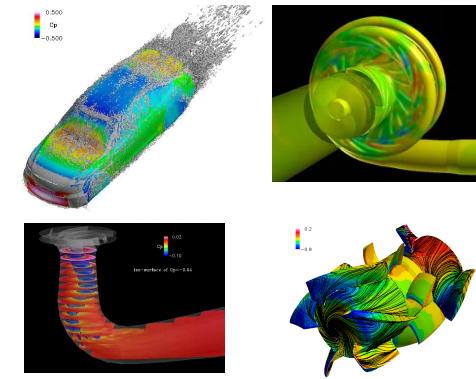
(*)通信を含む性能

The number of node	Elapse time(sec)	Ratio to peak performance
16	48.8	17.1%
256	48.8	17.6%
4096	48.9	17.7%
16384	48.8	17.8%
36864	48.6	17.9%
64512	48.5	17.9%
82944	48.5	17.9%

- 8万ノードまでの良好なウィークスケールラビリティを得られた
- フルノードでトータル性能**1.9Pflops**を達成



FrontFlow/blueの性能最適化

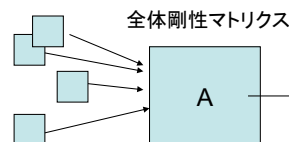


FrontFlow/blue (FFb)

- 有限要素法を用いた流体計算のプログラム
- 有限要素法には2つのタイプの計算方法がある
 - 全体剛性マトリクスを構築するタイプ
 - 全体構成マトリクスを構築せずに要素剛性マトリクスのみで計算を進めるタイプ(エレメント・バイ・エレメント法)
- FFbは新バージョンにおいて両方のソルバに対応

全体剛性マトリクスを使用する方法(陰解法の場合)

要素剛性マトリクス



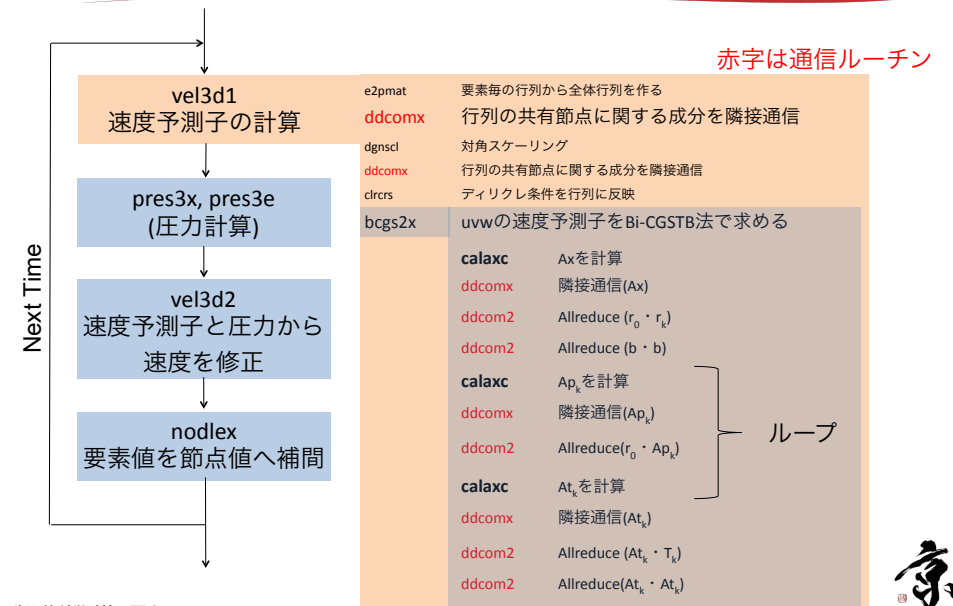
ソルバーを用いて解く

$$Ax=b$$

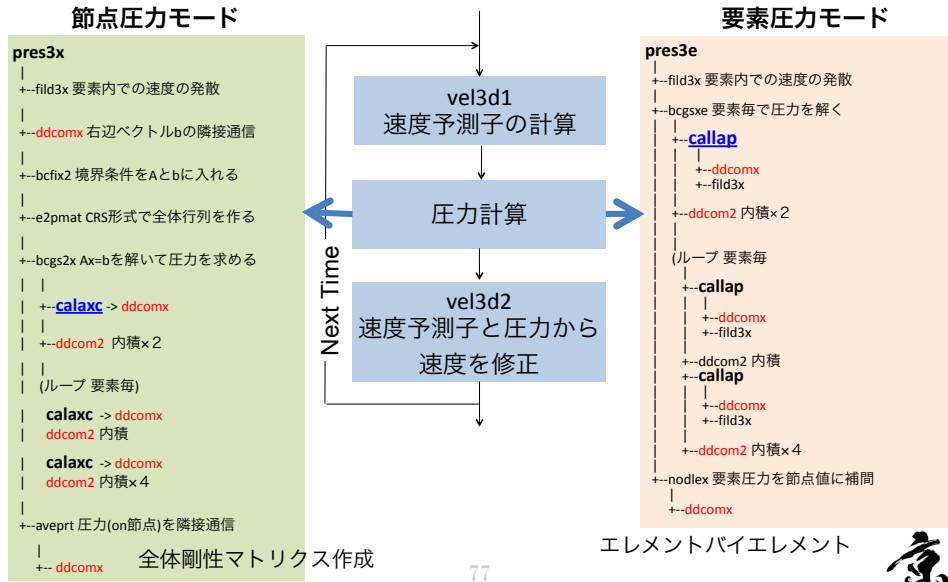
- ・Xは節点上の値となりAは周りの節点を関連づける粗行列となる
- ・ソルバーの演算は疎行列のマトリクス/ベクトル積が中心となる



FFbの主要部の構造

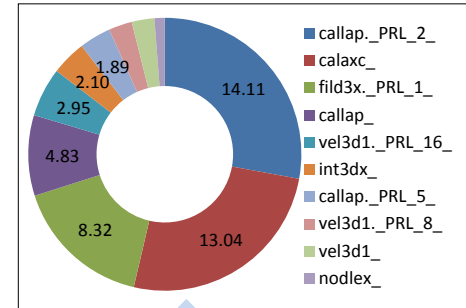


FFbの主要部の構造



FFbの主要計算

要素圧力モードでの処理時間内訳, 上位10件

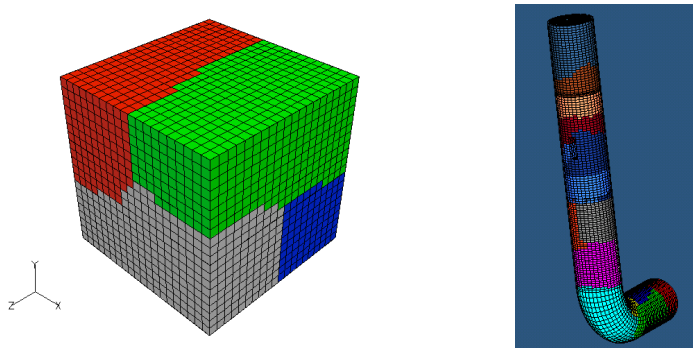


- 10万個の四面体要素+α
- 14.11s(30%)が四面体要素の勾配計算
- 13.04s(27%)が疎行列ベクトル積
- 8.32s (17%)が四面体要素の発散計算



FFbの並列化

3次元の領域分割 (非構造格子)

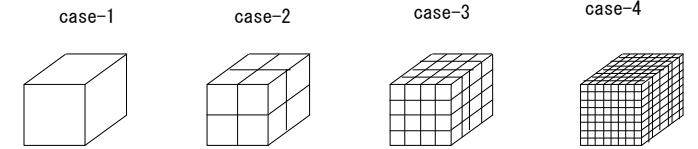
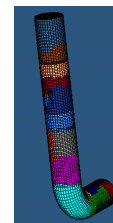


FFbの並列特性分析

初期のウィークスケール評価

評価モデルについて

メッシュ規模拡大イメージ(メッシュを細かくしただけで、評価モデルは同じもの)



	基本モデル	基本モデル×8	基本モデル×64	基本モデル×512
領域分割数 (コア数)	2	16	128	1024
要素数	53,460	427,680	3,421,440	27,371,520
節点数	60,680	456,342	3,535,646	27,827,454
要素数/コア	26,730	26,730	26,730	26,730
節点数/コア	30,340	28,521	27,622	27,175

規模を拡大しても、1コア当たりの計算規模は同等としている



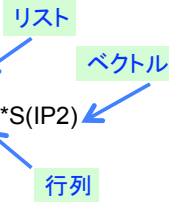
節点圧力カーネル (calaxc相当) の性能見積り

オリジナルコード(疎行列とベクトルの積)

```

ICRS=0
DO 110 IP=1,NP
  BUF=0.0E0
  DO 100 K=1,NPP(IP)
    ICRS=ICRS+1
    IP2=IPCRS(ICRS)
    BUF=BUF+A(ICRS)*S(IP2)
  100 CONTINUE
  AS(IP)=AS(IP)+BUF
110 CONTINUE
    
```

- CRS格納形式の行列ベクトル積
- ベクトルアクセスがリストアクセスとなる
- ベクトルの部分がL1キャッシュに載っていると**仮定した場合**
- ベクトルのメモリへのアクセスを全く無視してよい
- メモリからのロードは行列とリストのみ



要求B/F	8/2 = 4
性能予測	0.36/4 = 0.09

要求Byteの算出:

単精度 : 2 load なので

2*4 = 8byte

要求flop:

add : 1 mult : 1 = 2

(スレッド並列を仮定しピーク性能128Gflopsに対して)



節点圧力カーネル (calaxc相当) の実測性能

(スレッド並列なし : 1コア)

- オリジナルコードはスレッド並列されていなかった
- この状態の推定性能を前頁と同様な方法で見積もる
- メモリバンド幅を1コアで占有する場合のSTREAMベンチマークの結果は20GB/秒
- 1コアの理論ピーク性能は16GFLOPS
- 従って理論的なB/F値は20GB/16GFLOPで1.25

要求Byteの算出 : 2loadより 2* 4byte = 8

要求flop : 1(add)+1(mult) = 2

要求B/F	8/2 = 4
性能予測	1.25/4 = 0.313
実測値	0.059(六面体) 0.024(四面体)

- ベクトルがリストアクセス
- 連続アクセスでないためプリフェッチが効きにくい
- メモリアccessのレイテンシが見える
- 最悪1キャッシュラインのうち1要素しか使用できない事による大きなペナルティが発生
- 著しい性能低下が発生
- L2オンキャッシュでも同様のペナルティが発生

(スレッド並列なしピーク性能16Gflopsに対して)



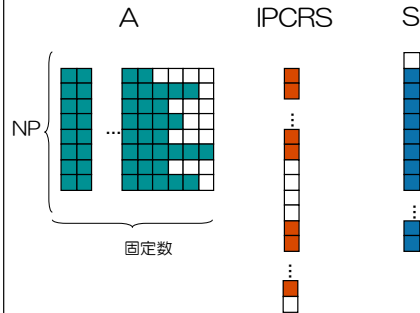
チューニング1: フルアンロール

- 狙い:
- スケジューリングの改善(演算待ちの削減)

- 変更点:
- 行列要素の配列に0を代入
 - ベクトルインデックスの配列に0を代入
 - 余分な配列同士は0*0の演算を実施

```

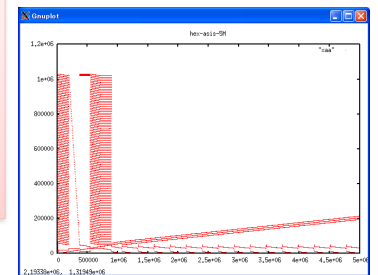
ICRS=0
DO 110 IP=1,NP
  BUF=0.0E0
  ! DO 100 K=1,NPP(IP) MAX_NZ=27
  BUF=BUF+A(ICRS+ 1)*S(IPCRS(ICRS+ 1))
  &   +A(ICRS+ 2)*S(IPCRS(ICRS+ 2))
  &   +A(ICRS+ 3)*S(IPCRS(ICRS+ 3))
  &   +A(ICRS+ 4)*S(IPCRS(ICRS+ 4))
  .....(省略).....
  &   +A(ICRS+24)*S(IPCRS(ICRS+24))
  &   +A(ICRS+25)*S(IPCRS(ICRS+25))
  &   +A(ICRS+26)*S(IPCRS(ICRS+26))
  &   +A(ICRS+27)*S(IPCRS(ICRS+27))
  ICRS=ICRS+27
  ! 100 CONTINUE
  AS(IP)=AS(IP)+BUF
110 CONTINUE
    
```



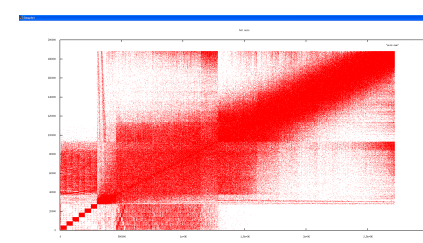
チューニング2: リオーダーリング (1/4)

- 狙い:
- ベクトルデータ(S)のブロック化によるL1,L2キャッシュミスの削減
- オリジナルデータの特徴:
- 6面体(総回転数: 約2700万)
 - 最初の1M回のSへの広範囲なランダムアクセス
 - それ以降は二極化するが、局所的アクセス
 - 4面体(総回転数: 約270万)
 - 全アクセスとも、広範囲なランダムアクセス

6面体 オリジナルデータ



4面体 オリジナルデータ

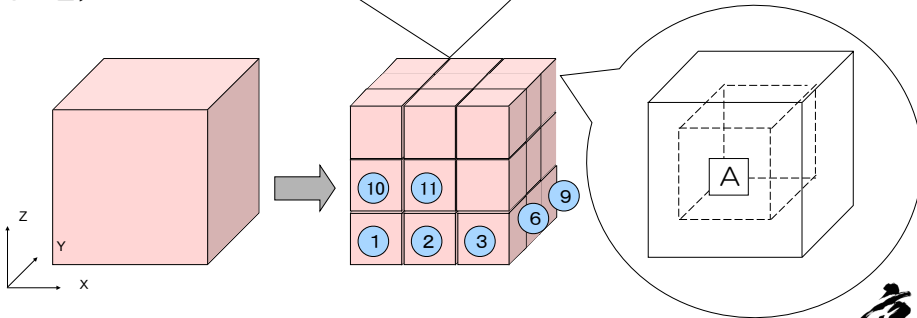


チューニング2: リオーダーリング (2/4)

節点番号のリオーダーリング:

- オリジナルデータを各軸分割しブロックを作成
- 各ブロックを外と内に分割し物理座標に基づき内側・外側の順にナンバリング

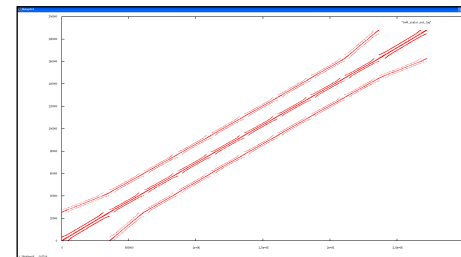
ひとつのブロックを内と外に分け、内側(A)のナンバリング後、外側(B)のナンバリングを実施(内:外の比 8:2)



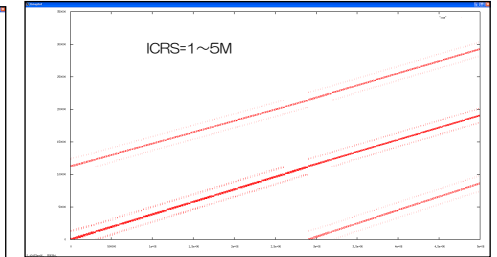
チューニング2: リオーダーリング (3/4)

- 物理的に近い節点が配列の並びとしても近い位置に配置される事を期待
- 一要素を構成する節点の番号が近くなる
- 一箱の大きさを調整することによりベクトルのリストアクセスの多くに対しL1オンキャッシュのデータを利用できる

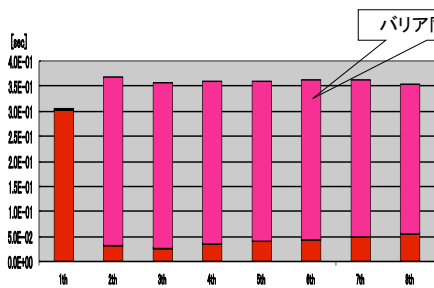
■ 4面体 リオーダーリング結果



■ 6面体 リオーダーリング結果



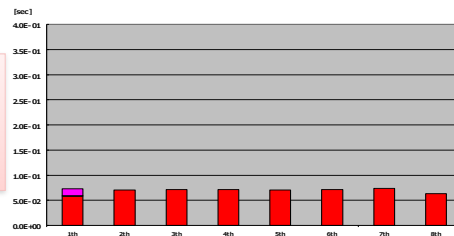
チューニング2: リオーダーリング (4/4)



バリア同期待ち

- リオーダーリング前
- マスタースレッドが担当する範囲にランダムアクセスが集中しメモリアccessコストが増大
 - スレーブスレッドが担当する範囲は局所アクセスが多く結果としてバリア同期待ちのコストが支配的

- リオーダーリング後
- 全てのスレッドが均一的に局所アクセス化
 - メモリアccessバランスの改善と同期コストの大幅な削減

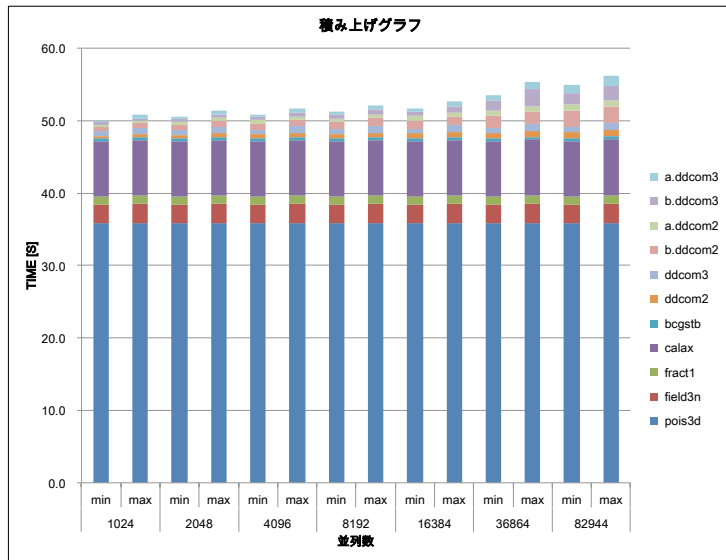


節点圧力カーネル (calaxc相当) チューニング結果

	6面体	4面体
フルアンロール (Score)	5.4%	3.0%
フルアンロール + リオーダーリング (Score)	8.1%	7.7%

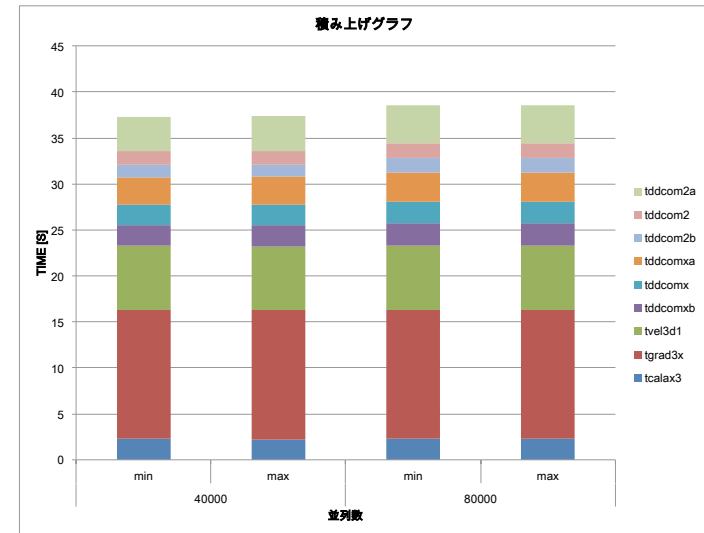
L1 オンキャッシュである時の理論性能値である9%に近い性能値を実現

FFbのウィークスケール性能 (Ver5)



FFbの総合性能 (Ver7)

現状の総合性能は80000ノードでピーク性能比：3.16%



まとめ

- 理研で進めた性能最適化
- RSDFTの性能最適化
- PHASEの性能最適化
- Seism3Dの性能最適化
- FrontFlow/blueの性能最適化

Hasegawa, Y., Iwata, J.I., Tsuji, M., Takahashi, D., Oshiyama, A., Minami, K., Boku, T., Shoji, F., Uno, A., Kurokawa, M., Inoue, H., Miyoshi, I., Yokokawa, M.: "First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer", Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis. pp. 1:1-1:11. SC '11, ACM, New York, NY, USA (2011)

黒田 明義, 長谷川 幸弘, 寺井 優晃, 井上 俊介, 市川 真一, 小松 秀実, 大井 憲行, 安藤 琢也, 山崎 隆浩, 大野 隆央, 南 一 生, "ナノ材料第一原理分子動力学プログラムPHASE の京速コンピュータ「京」上の計算性能最適化", ハイパフォーマンスコンピューティングと計算科学シンポジウム論文集, pp.144-152 (2012)

南 一 生, 井上 俊介, 堤 重信, 前田 拓人, 長谷川 幸弘, 黒田 明義, 寺井 優晃, 横川 三津夫, "「京」コンピュータにおける疎行列とベクトル積の性能チューニングと性能評価", ハイパフォーマンスコンピューティングと計算科学シンポジウム論文集, pp.23-31 (2012)

Kiyoshi Kumahata, Shunsuke Inoue, Kazuo Minami: Kernel performance improvement for the FEM-based fluid analysis code on the K computer, Procedia Computer Science, Volume 18, 2013, Pages 2496-2499, 2013 International Conference on Computational Science

井上俊介, 堤重信, 前田拓人, 南一 生, "スーパーコンピュータ「京」におけるメモリアクセシブルなアプリケーションの評価および高性能化, 先進的計算基盤システムシンポジウム SAC/SIS2013論文集, pp.123(2013)

Masaaki Terai, Eiji Tomiyama, Hitoshi Murai, Kazuo Minami and Mitsuo Yokokawa: "Kscope: a Java-based Fortran Source Code Analyzer with Graphical User Interface for Performance Improvement", Third International Workshop on Parallel Software Tools and Tool Infrastructures (PSTI2012).

