

スーパーコンピュータ「京」におけるアプリケーションの高並列化と高性能化

2012年5月

独立行政法人理化学研究所
次世代スーパーコンピュータ開発実施本部 開発グループ
アプリケーション開発チーム チームリーダー

南 一生
minami_kaz@riken.jp



RIKEN Advanced Institute for Computational Science

SACSIS 2012

アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



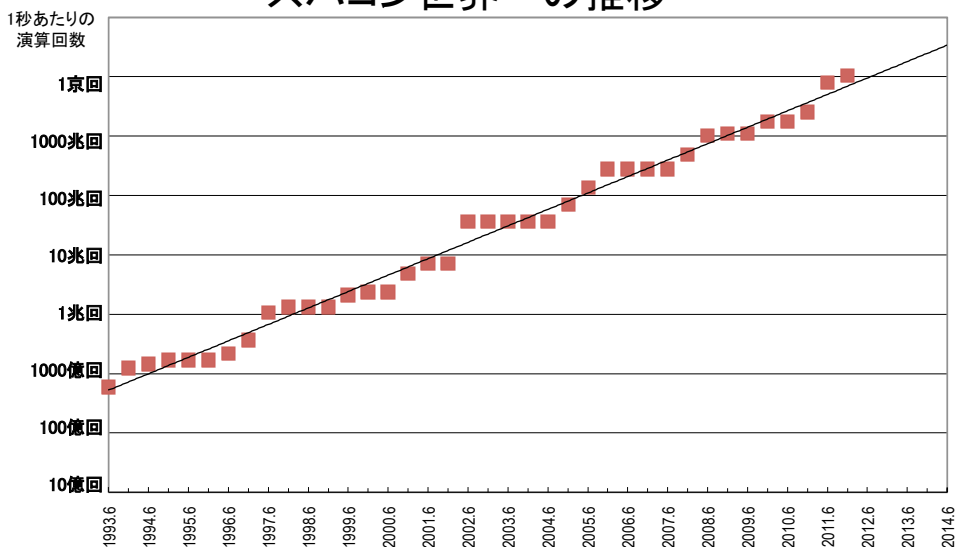
アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



スパコン世界一をめぐる競争

スパコン世界一の推移



平均すると1年に約1.9倍の性能向上！

現在世界最速のスパコンは、世界初のスパコンCRAY-1(1976年)の約7000万倍



ちなみに

世界初のスパコンCRAY-1の演算性能は、約160メガフロップス
 一方、iPhone4Sの演算性能は、約140メガフロップス



1976年



SACSIS 2012

2011年



4



これまでの日本のスパコンプロジェクト

各国の最速マシンのTOP500リストにおける順位

数値風洞@JAXA



提供:JAXA

CP-PACS@筑波大学



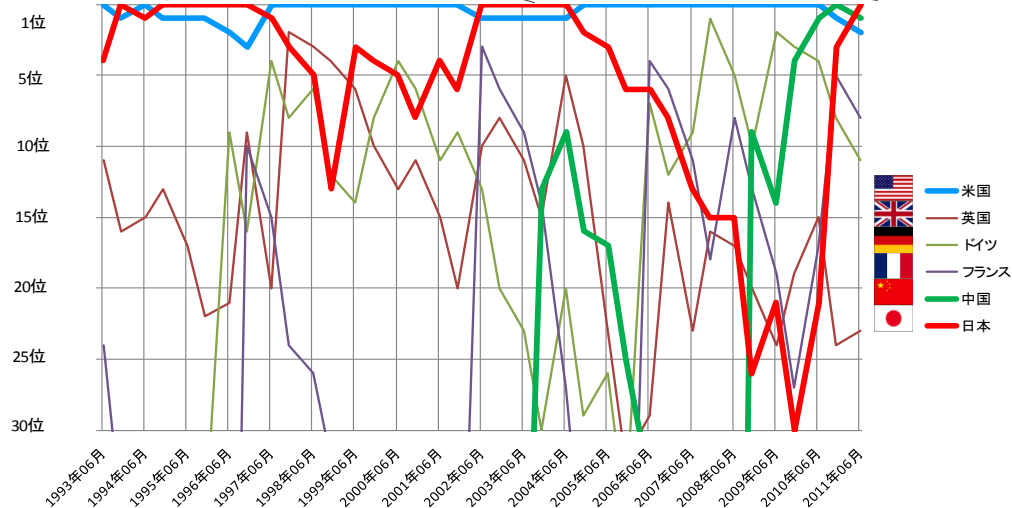
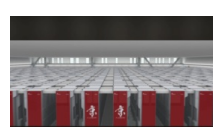
提供:筑波大学

地球シミュレータ@海洋研究開発機構



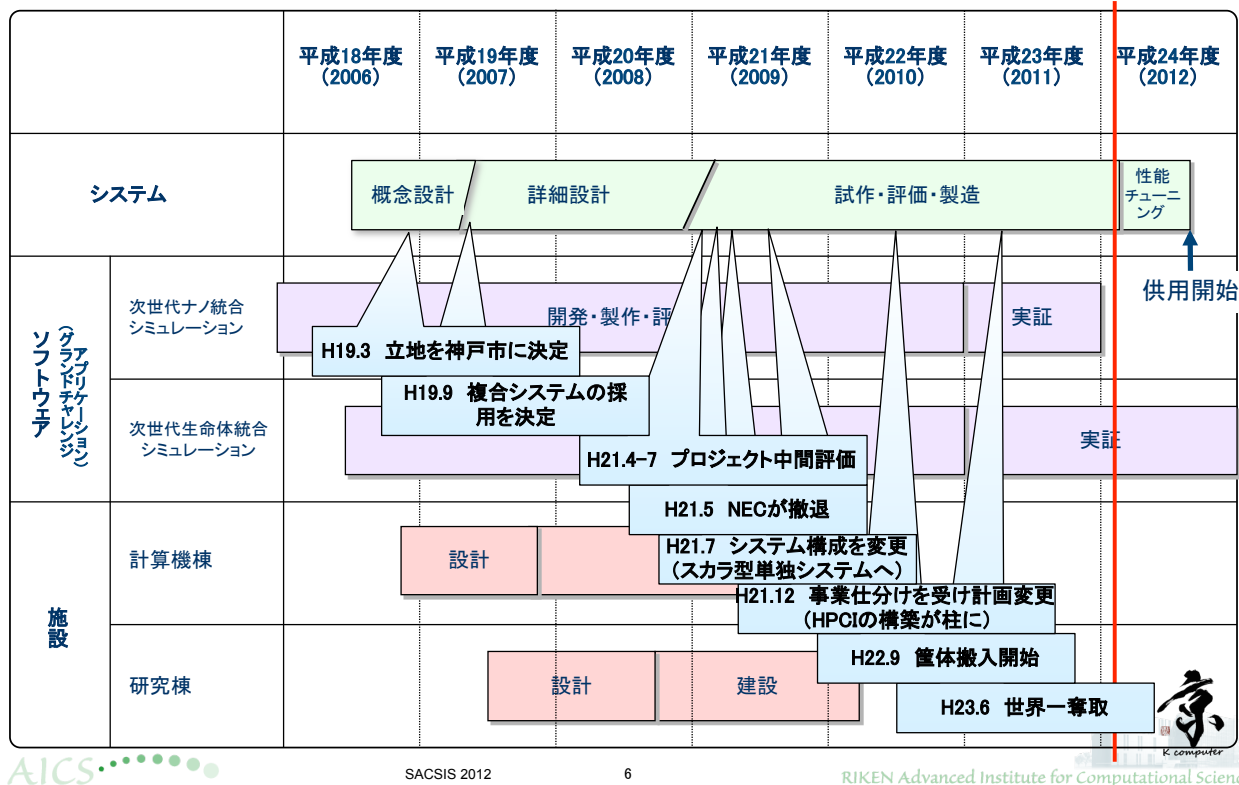
提供:JAMSTC

京@計算科学研究機構



開発スケジュール(平成18年度－平成24年度)

現在



「京」の最近の成果

(※)TOP500リストとは？

LINPACKベンチマークの実行性能を指標とした、世界のスパコン上位500位までのランキング。年に2回、6月と11月に更新される

(※)LINPACK(リンパック)とは？

大規模連立一次方程式を解くベンチマークプログラムで、世界のスパコンランキングであるTOP500でスパコンの性能指標として利用されている

✓ 2011/11/15発表の第38回TOP500リストで再び第一位を獲得.

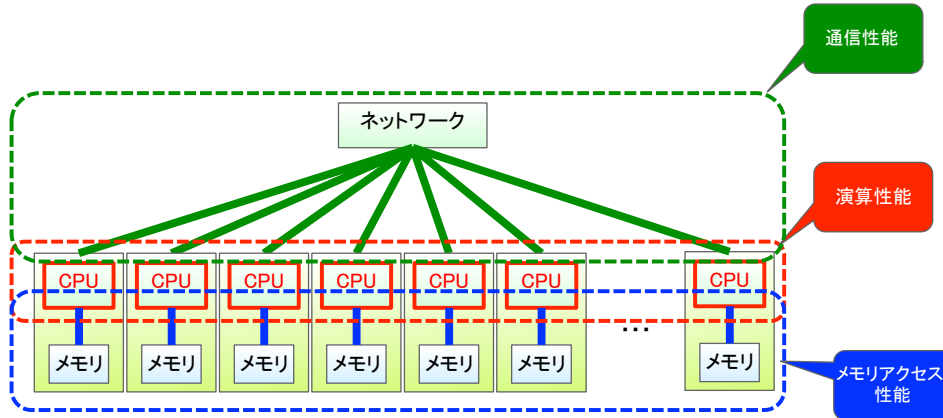
	K computer (Oct 2011)	K computer (Jun 2011)
R_{max}	10.51PFLOPS	8.162PFLOPS
R_{peak}	11.28PFLOPS	8.774PFLOPS
efficiency	93.2%	93.0%
N_{max}	11,870,208	10,725,120
Execution time	29h28m	27h59m



ランキング	国	システム名	演算性能 (ペタフロップス)	実行効率 (%)	1ワットあたりの演算性能	実行時間 (時間)
1	Japan	K computer	10.510	93	824.56	29.47
2	China	Tianhe-1A	2.566	55	635.15	3.37
3	US	Jaguar	1.759	75	253.07	17.27
4	China	Nebulae	1.271	43	492.64	1.91
5	Japan	TSUBAME2.0	1.192	52	852.27	2.40

演算性能だけでなく、高効率、低消費電力、高信頼性を同時に実証

スパコンの性能とは？



演算性能を上げるのは、比較的容易
メモリアクセス性能と通信性能を上げるのは困難

TOP500の指標となるLINPACKベンチマークは
演算性能のみで決まる
(メモリアクセス性能、通信性能はほとんど考慮されない)



「京」の最近の成果

HPCチャレンジアワード: 科学技術計算で多用される計算パターンから抽出した28項目の処理性能によって、スパコンの総合的な性能を評価するHPCチャレンジベンチマークプログラムから、特に重要な4つのベンチマークをHPCチャレンジアワードとして、毎年11月のSCにて表彰

- Global HPL (大規模な連立1次方程式の求解における演算速度) → 演算性能
- Global RandomAccess (並列プロセス間でのランダムメモリアクセス性能) → 通信性能
- EP STREAM (Triad) per system (多重負荷時のメモリアクセス速度) → メモリアクセス性能
- Global FFT (高速フーリエ変換の総合性能) → 演算性能, 通信性能

2011/11/16発表の2011年HPCチャレンジアワードの4部門すべてで第一位を獲得

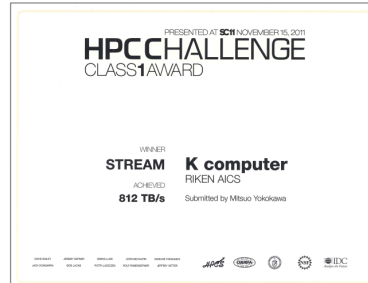
Global HPL	Performance (TFLOP/s)	System	Institutional Facility
1 st place	2,118	K computer	RIKEN
1 st runner up	1,533	Cray XT5	ORNL
2 nd runner up	736	Cray XT5	UTK
Global RandomAccess	Performance (GUPS)	System	Institutional Facility
1 st place	121	K computer	RIKEN
1 st runner up	117	IBM BG/P	LLNL
2 nd runner up	103	IBM BG/P	ANL
EP STREAM (Triad) per system	Performance (TB/s)	System	Institutional Facility
1 st place	812	K computer	RIKEN
1 st runner up	398	Cray XT5	ORNL
2 nd runner up	267	IBM BG/P	LLNL
Global FFT	Performance (TFLOP/s)	System	Institutional Facility
1 st place	34.7	K computer	RIKEN
1 st runner up	11.9	NEC SX-9	JAMSTEC
2 nd runner up	10.7	Cray XT5	ORNL





HPCC

HPC wire



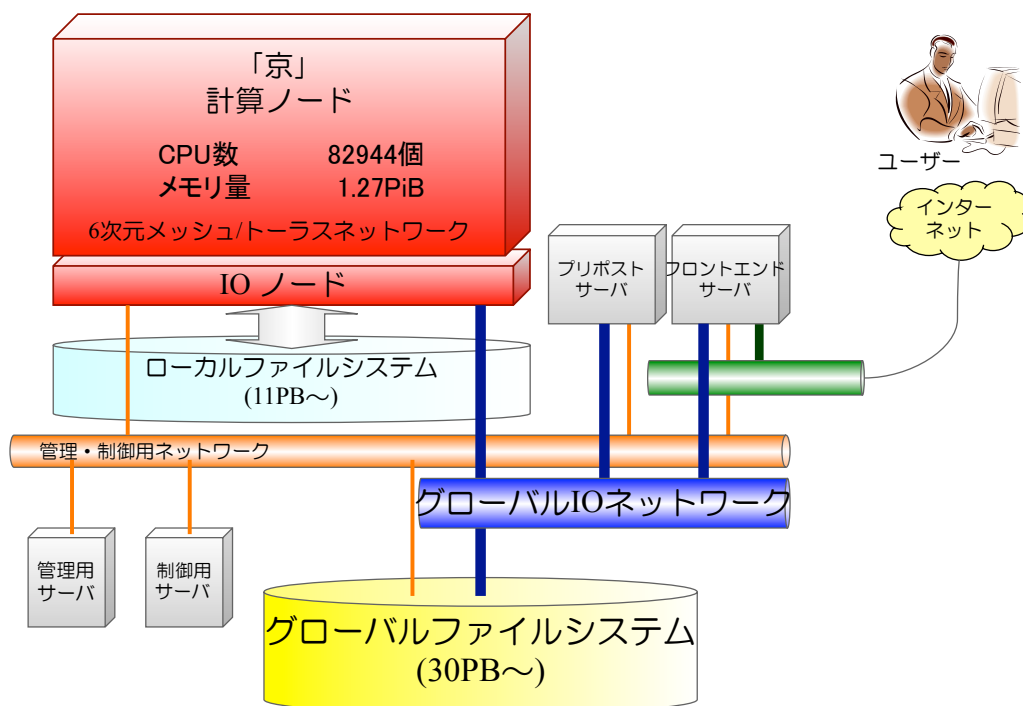
SACSIS 2012

10

アジェンダ

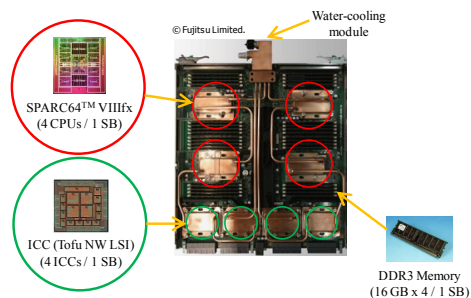
1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ

「京」の概要



CPUの詳細

	諸元
演算性能 (ピーク)	128 GFLOPS (16 GFLOPS x 8 cores)
コア数	8
クロック周波数	2.0 GHz
浮動小数点演算器	乗加算ユニット x 4 (2 SIMD) 除算器 x 2
レジスタ数	浮動小数点レジスタ (64bit) : 256 汎用レジスタ (64bit) : 188
キャッシュ	L1I\$: 32 KB (2way) L1D\$: 32 KB (2way) L2\$: Shared 6 MB (12way)
メモリ帯域	64 GB/s (0.5B/F)



45nm CMOS process
チップサイズ: 22.7mm x 22.6mm
トランジスタ数: 760M
Power: 58W (TYP, 駆動温度30°C), 水冷

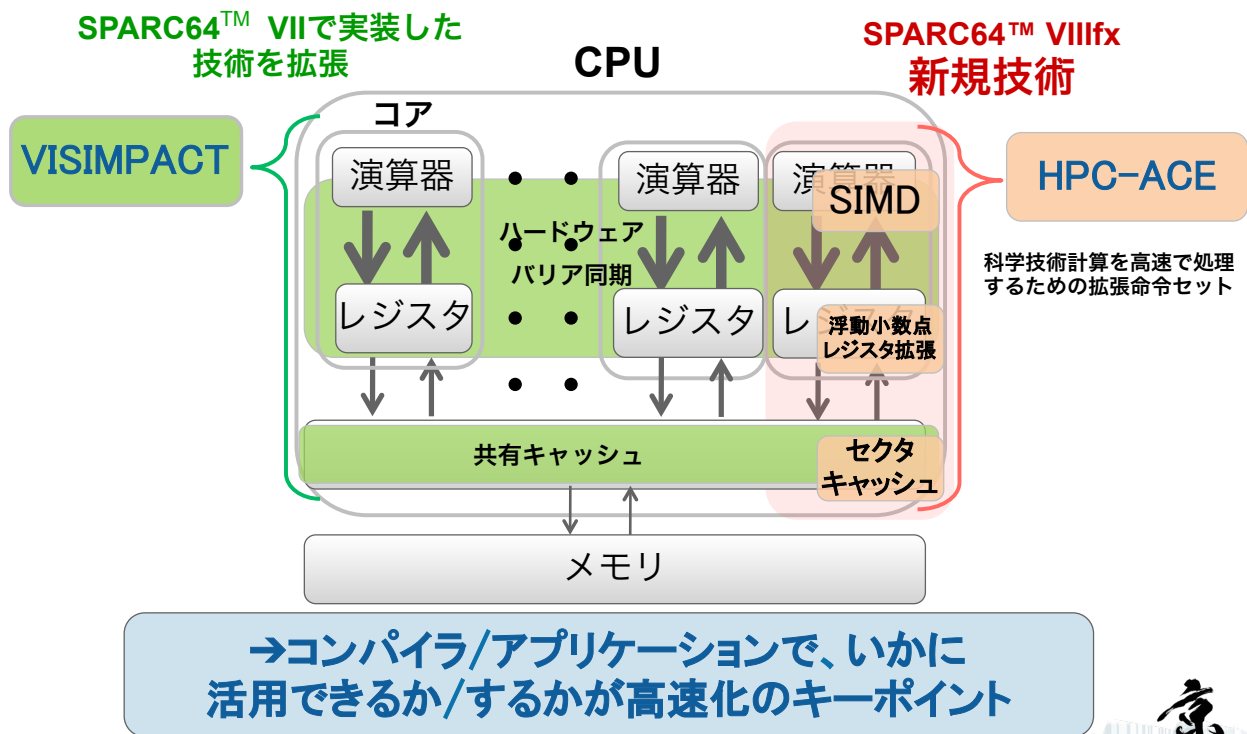
他のチップとの比較

Vendor	Name	Core	Process rule (nm)	Peak performance (GFLOPS)	Cache (MB)	Power (W)	GF/W	System (w/planned)
IBM	PowerPC A2	16	45	204.80	32	55	3.72	Sequoia (BlueGene/Q)
Intel	E3-1260L	4	32	105.60	8	45	2.35	
Fujitsu	SPARC64VIIIfx	8	45	128.00	6	58	2.21	K computer
IBM	Power7	8	45	256.00	32	200	1.28	
AMD	Opteron 6172	12	45	100.80	12	80	1.26	XE6, etc.
Intel	Xeon X5670	6	32	79.92	12	95	0.84	TSUBAME2.0, etc.

高性能かつ低消費電力



SPARC64™ VIIIfx 高速化技術



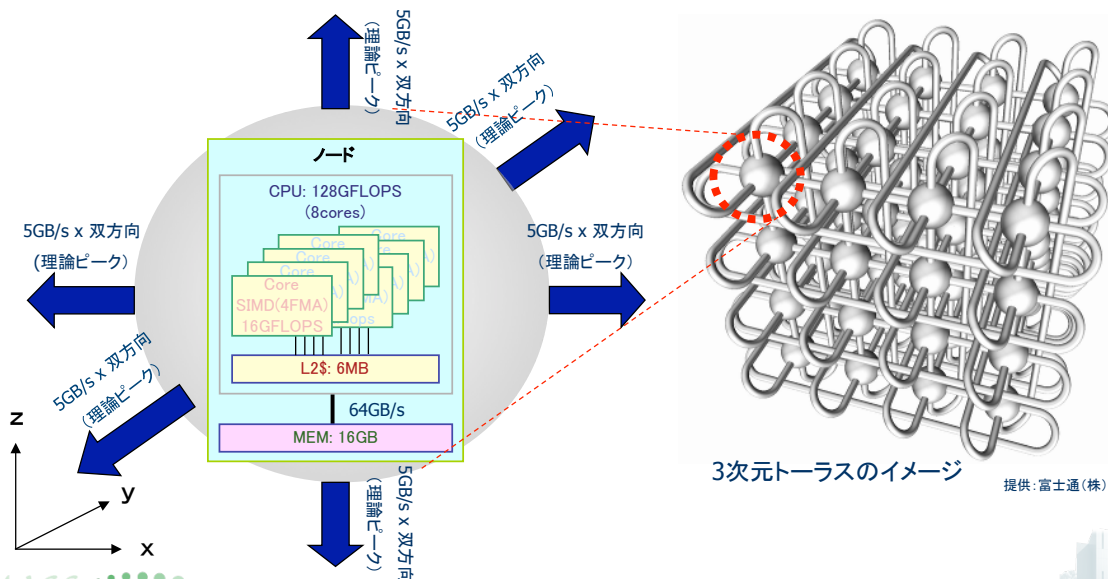
計算ノードとインターコネクトの構成

■ 計算ノードの構成

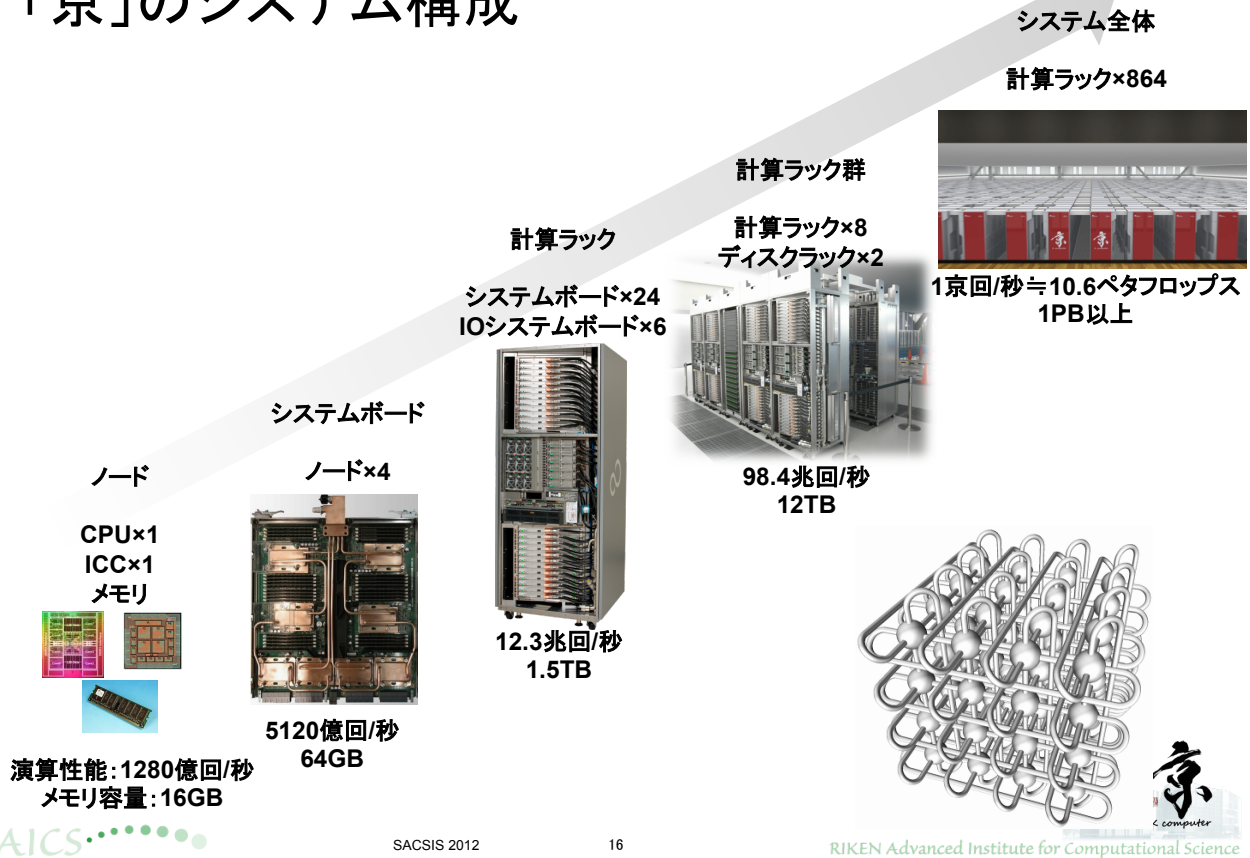
- CPU (8コア) : 1個
- ICC (インターコネクト用LSI) : 1個
- メモリ : 16GB

■ インターコネクトの構成

- ユーザービューは3次元トラス
- 帯域: 3次元の正負各方向にそれぞれ 5GB/s x 2 (双方向)【理論ピーク】
- ケーブル: 約200,000本, 約1000km



「京」のシステム構成



システムの特長

✓ 世界トップクラスの演算性能と汎用性(使いやすさ)の両立

- ✓ LINPACK^(※) 10ペタフロップス(1秒間に1京回)
- ✓ ペタフロップス級のアプリケーション実効性能
- ✓ 広範囲のアプリケーションに対応可能
 - ✓ 高帯域のメモリアクセスとインターコネクト

✓ 高性能と低消費電力の両立

- ✓ CPU:128ギガフロップス, 58W(LINPACK時、駆動温度30°C)
- ✓ システム全体:824.56メガフロップス/ワット
 - ✓ 省エネスパコンランキング(GREEN500)2011年6月版で第6位
 - ✓ 大規模システムとしてはトップクラス

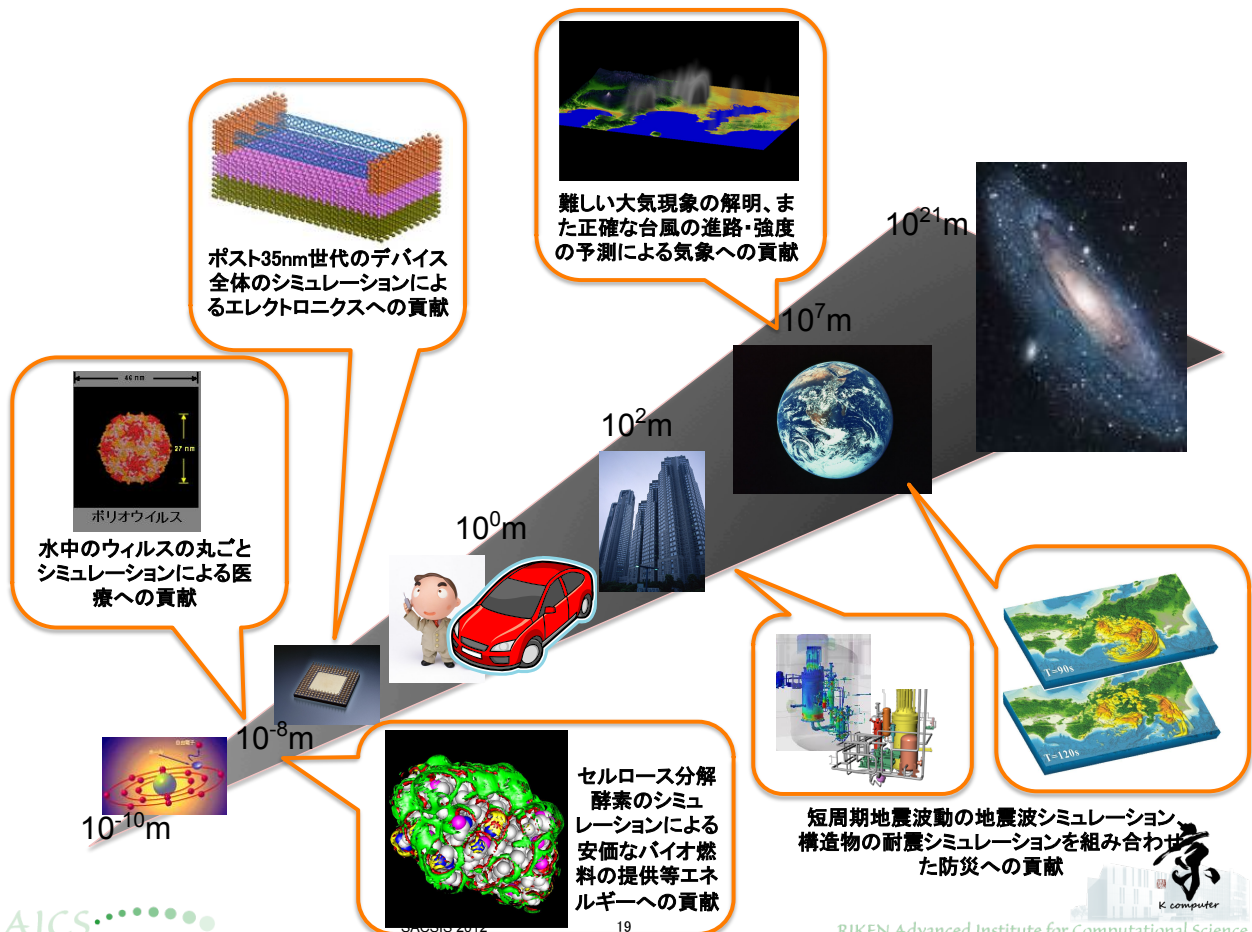
✓ 高い信頼性の確保

- ✓ 「壊れない」、「壊れても全てが止まらない」、「壊れた部分はすぐ直せる」
- ✓ ネットワークの高信頼性化: 自動代替経路, 自動再構成機能
- ✓ 約30時間の高負荷連続運転(LINPACK計測)を実証



アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ

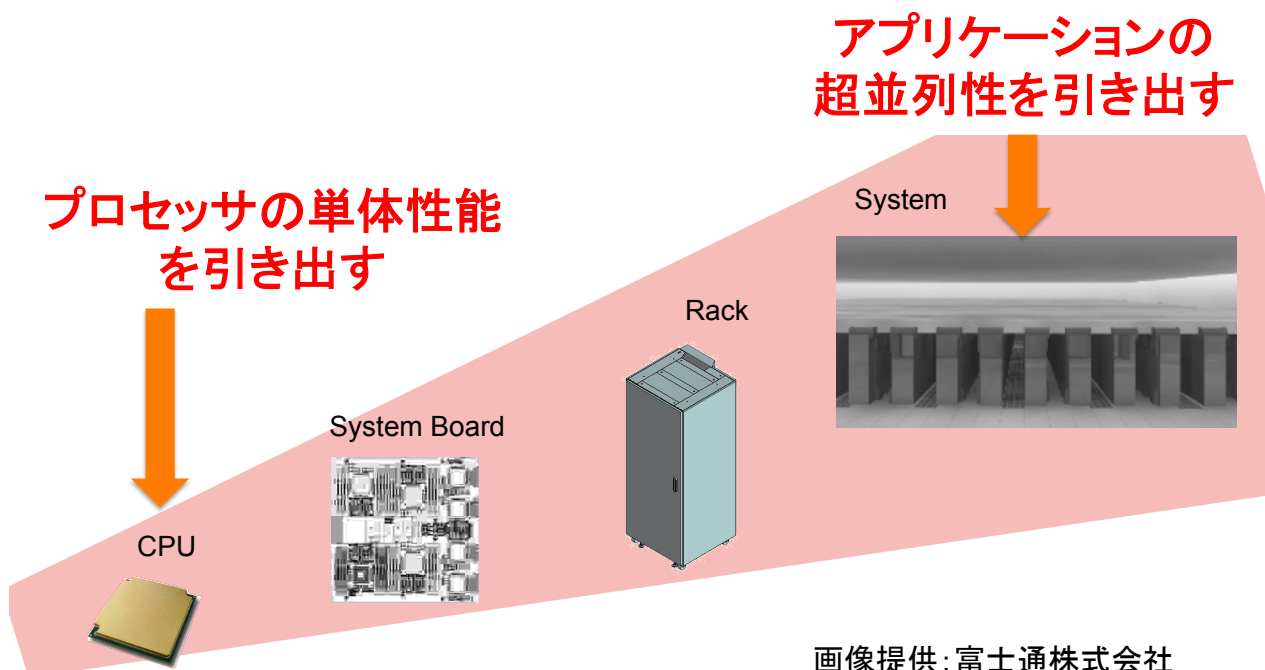


アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



現代のスパコン利用の難しさ



アジェンダ

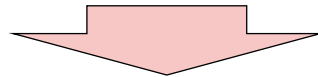
1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



理研で進めているアプリ高性能化

目的

- ✓ 次世代スパコンの運用に先立ち、システム性能を実証する

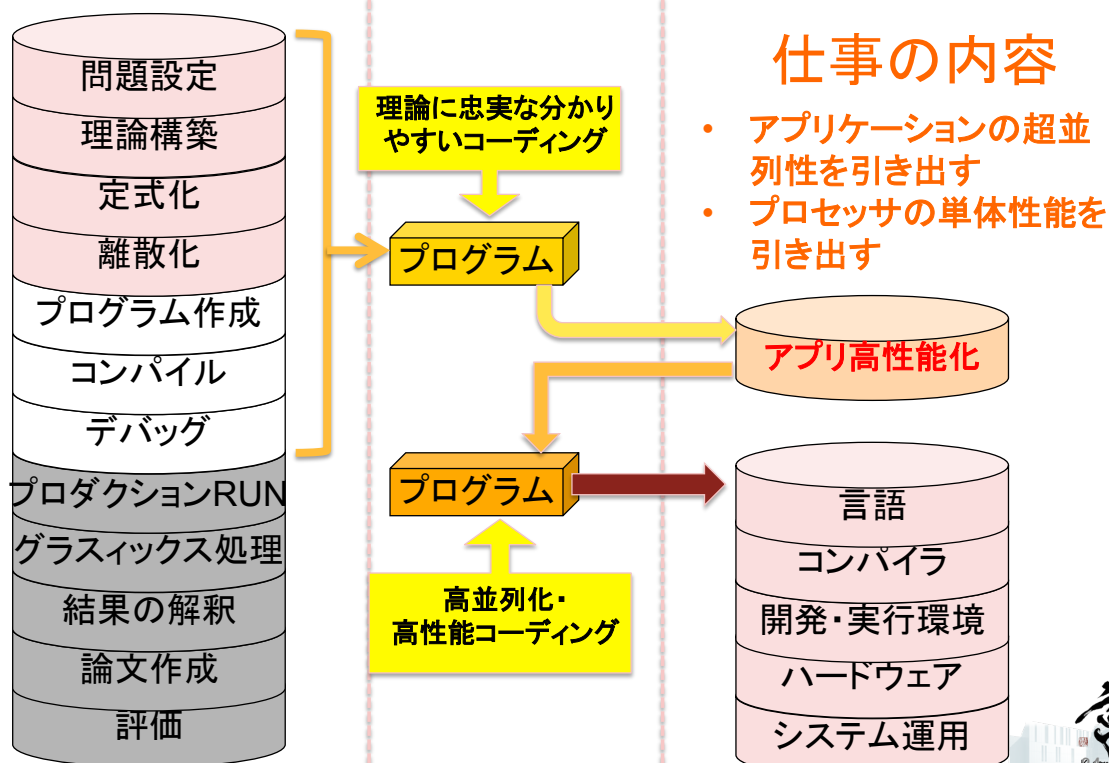


- 並列性能が見込めるコード複数本を対象
- 次世代スパコンの汎用性を踏まえ分野バランスを考慮
- 次世代スパコンの汎用性を踏まえ計算特性のバランスを考慮
 - 並列化手法が比較的**理解しやすい**・かなり複雑
 - 高い**単体性能**が得やすい・得にくい



計算科学

計算機科学



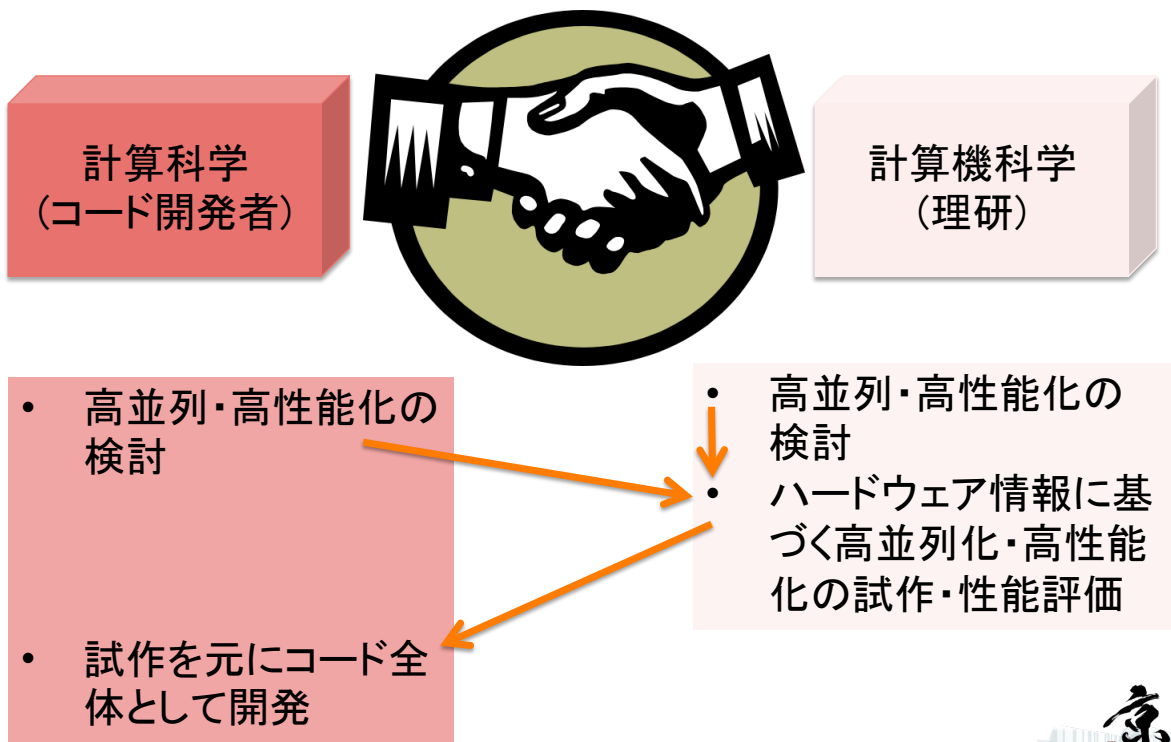
仕事の内容

- アプリケーションの超並列性を引き出す
- プロセッサの単体性能を引き出す

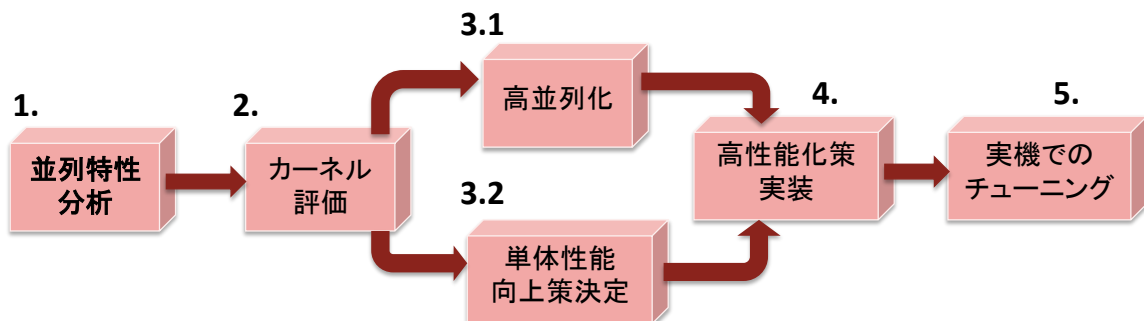
対象アプリケーション

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
NICAM	地球科学	全球雲解像度大気大循環モデル	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。	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。	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。	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。	FEM (流体)
RSDFT	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。	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。	QCD

役割分担



アプリ高性能化のステップ

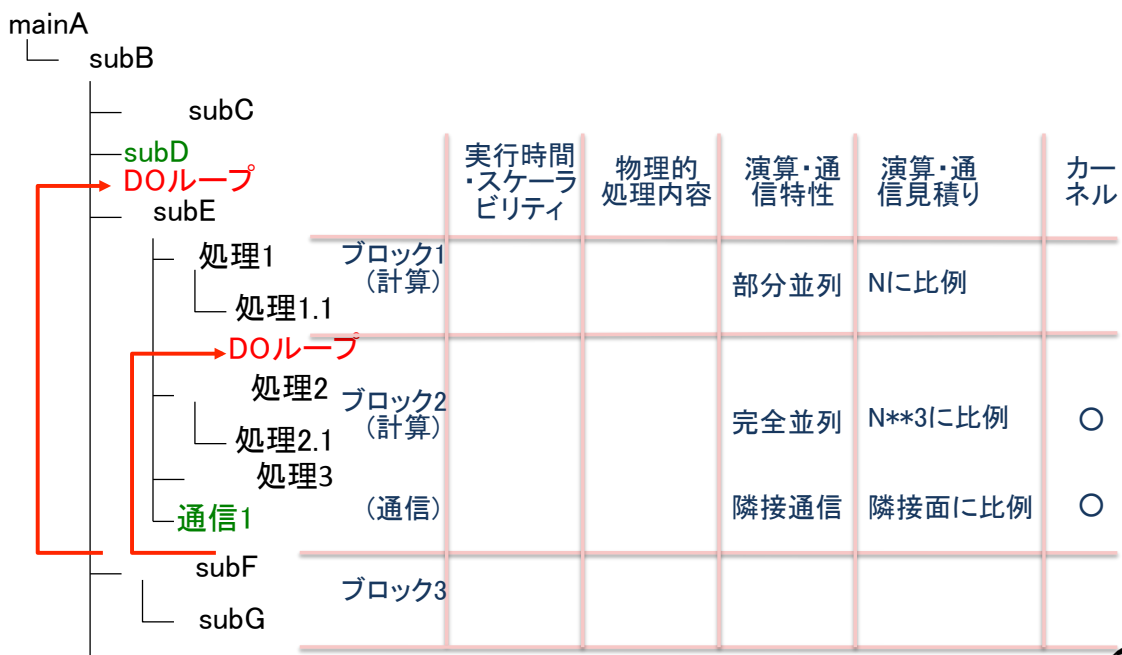


アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. **高並列化のための重要点**
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



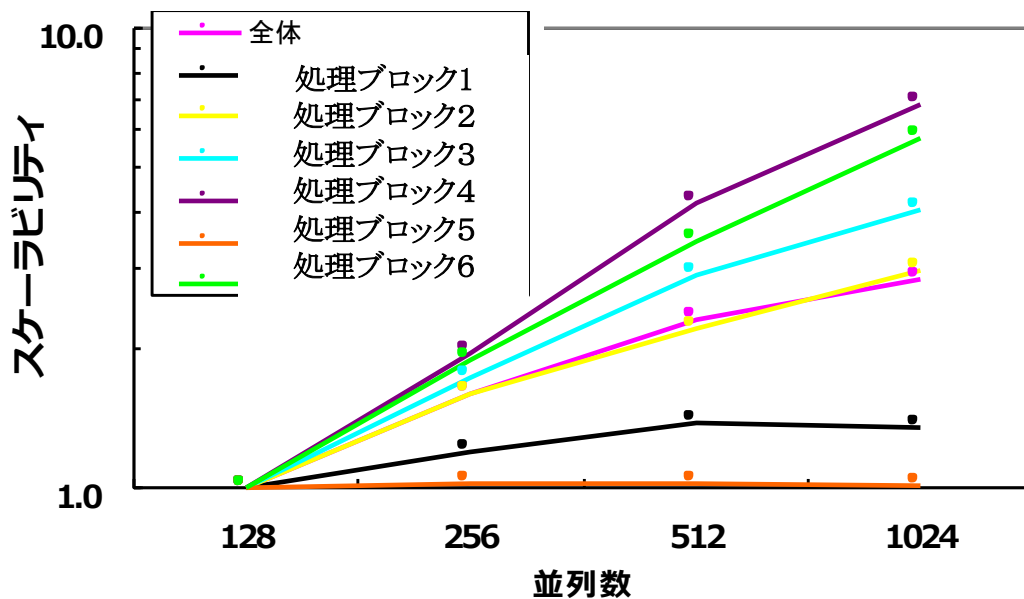
高並列化については、高並列を阻害する要因を洗い出す事が一番重要と云える。そのための並列特性の分析が重要。



ブロック毎の実行時間とスケーラビリティ評価(例)

→従来の評価はサブルーチン毎・関数毎等の評価が多い

→サブルーチン・関数は色々な場所で呼ばれるため正しい評価ができない



超高並列を目指した場合の留意点-ブロック毎に以下を評価する

- 非並列部が残っていないか？残っている場合に問題ないか？
- ロードインバランスが超高並列時に悪化しないか？
- 隣接通信時間が超高並列時にどれくらいの割合を占めるか？
- 大域通信時間が超高並列時にどれくらい増大するか？

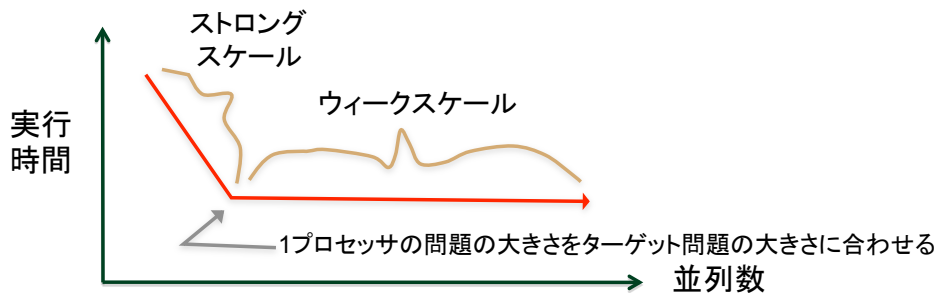
➡ これらの評価が重要

ストロングスケーリング: 全体の問題規模を一定にして並列数を増やし測定する方法

ウィークスケーリング: 1プロセッサで実行する問題規模を一定にし並列数を増やし測定する方法

そのために

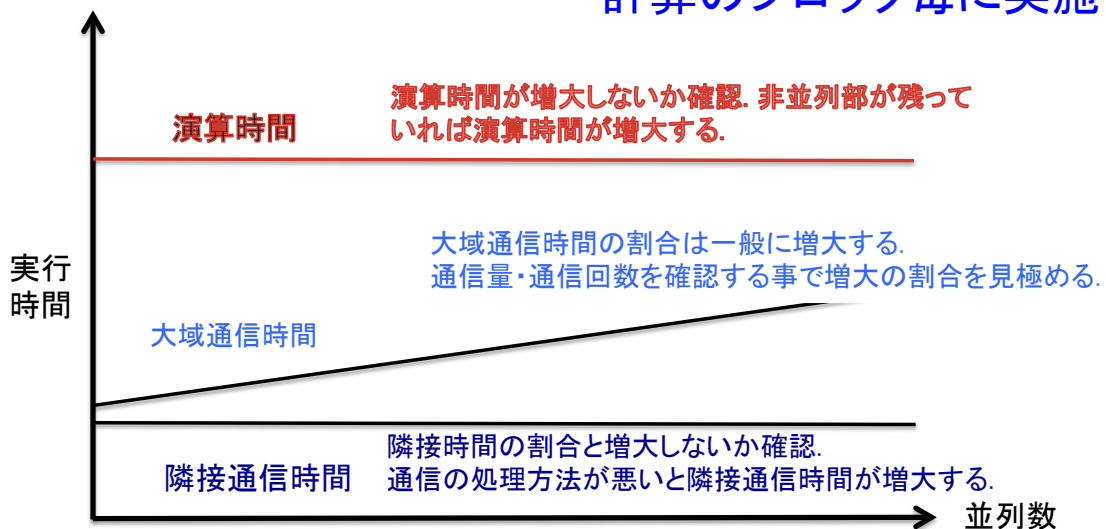
- ターゲット問題を定める
- 1プロセッサの問題規模がターゲット問題と同程度となるまでは、ストロングスケールで実行時間・ロードインバランス・隣接通信時間・大域通信時間を測定・評価する(100から数百並列まで)
- 上記の測定・評価で問題があれば解決する
- 問題なければ並列度を上げてウィークスケールで大規模並列の挙動を測定する



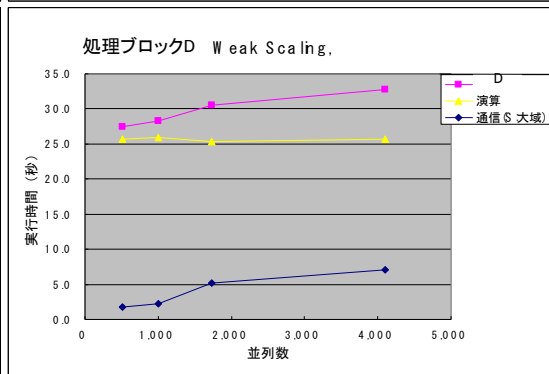
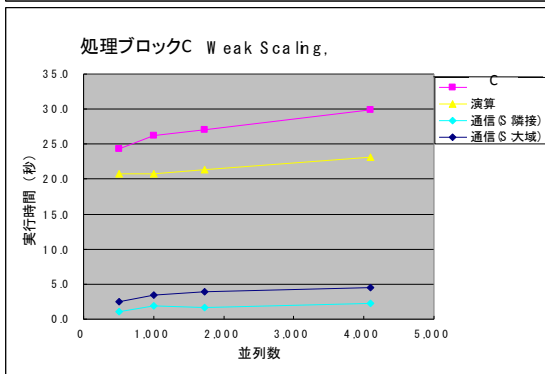
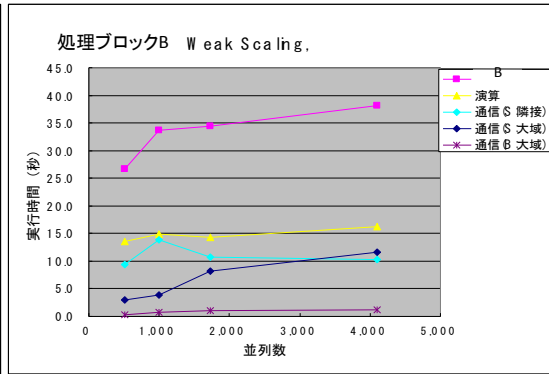
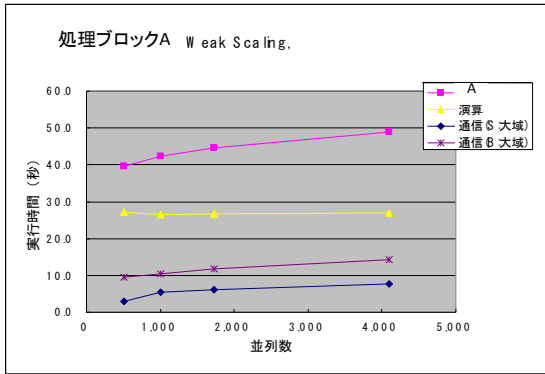
ウィークスケール評価

- 現状使用可能な実行環境を使用し100程度/1000程度/数千程度と段階を追ってできるだけ高い並列度で並列性能を確認する(ウィークスケール測定).
- ウィークスケールが難しいものもあるが出来るだけ測定したい. 難しい場合は、演算時間をモデル化して実測とモデルとの一致を評価する.

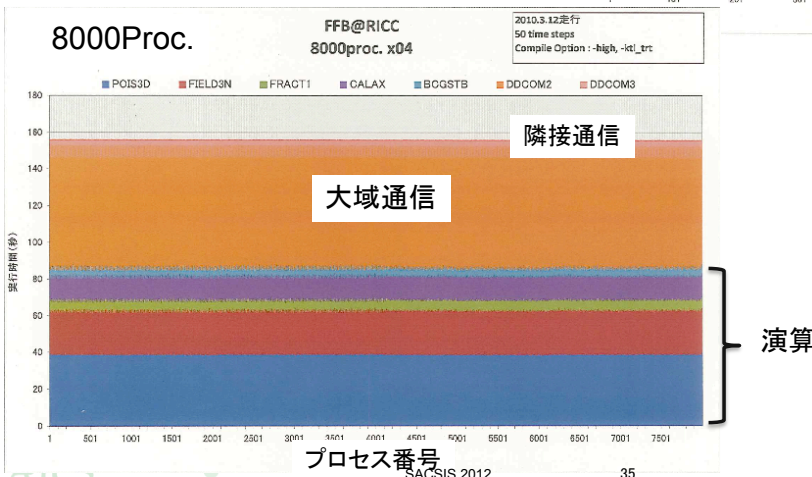
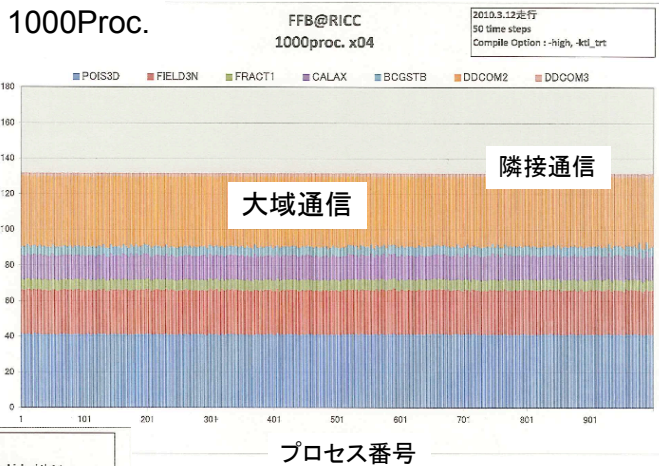
計算のブロック毎に実施



Weak Scaling測定例



Weak Scaling測定例 (横軸をランク番号としてロードインバランスをチェック)



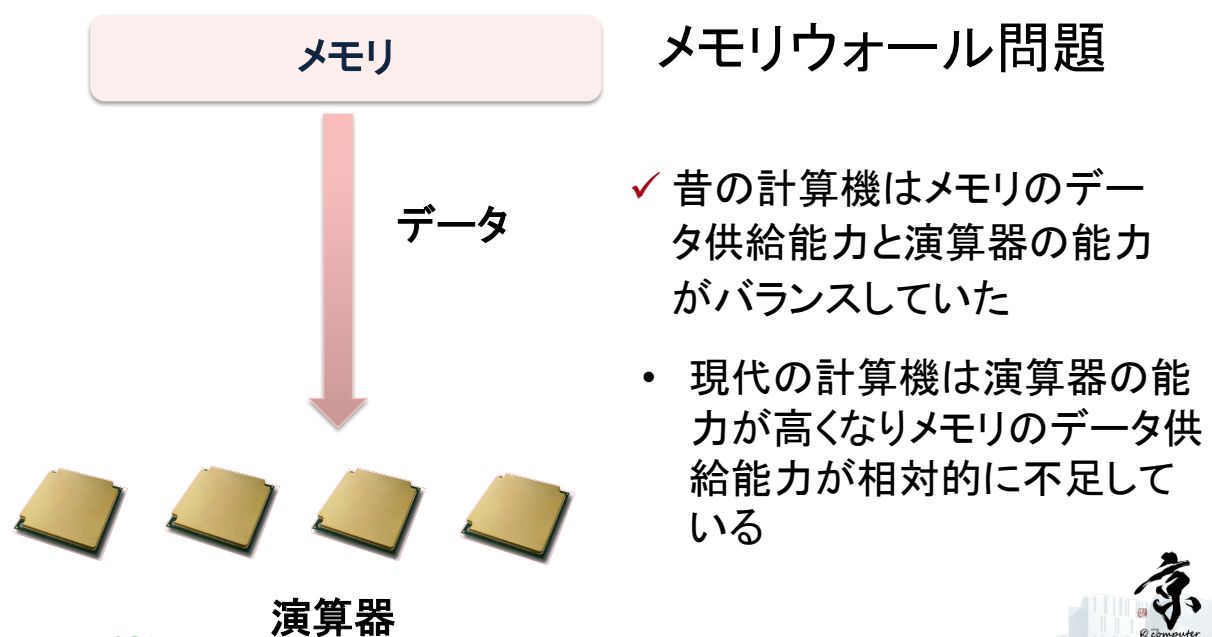
アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



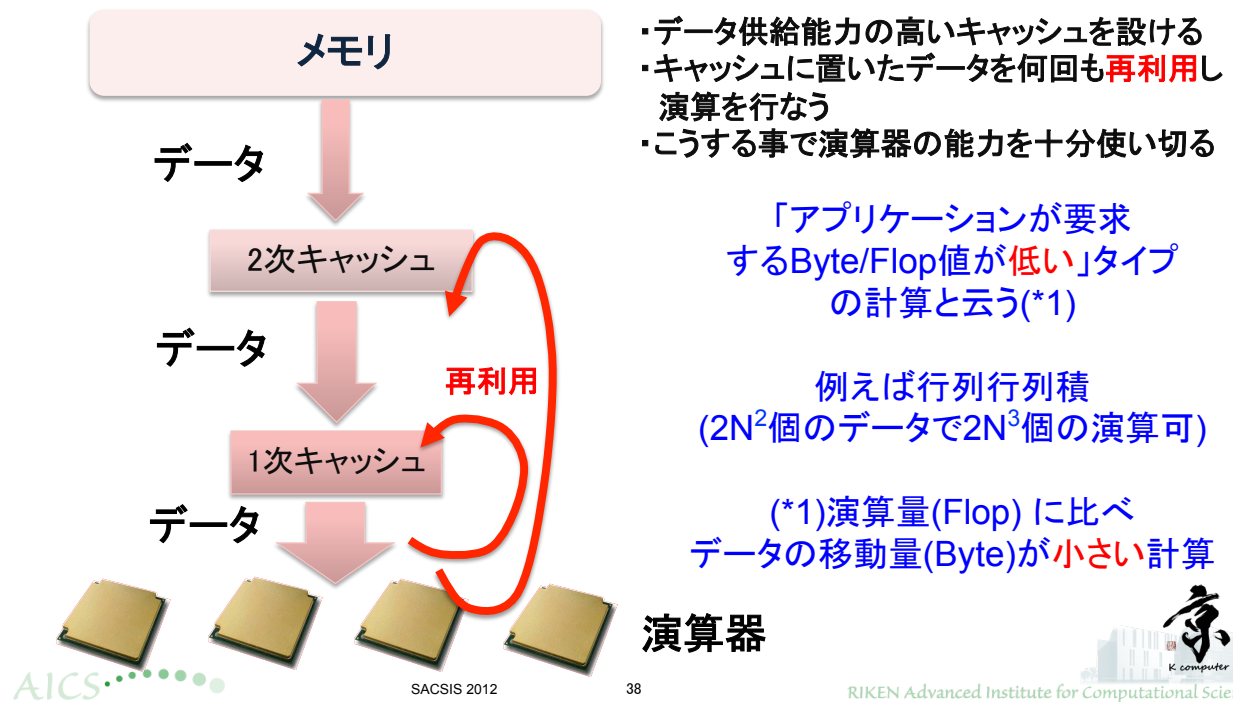
プロセッサの単体性能を引き出す(1)

- かつては研究者やプログラマーは物理モデル式に忠実に素直にプログラミングすることが一般的であった



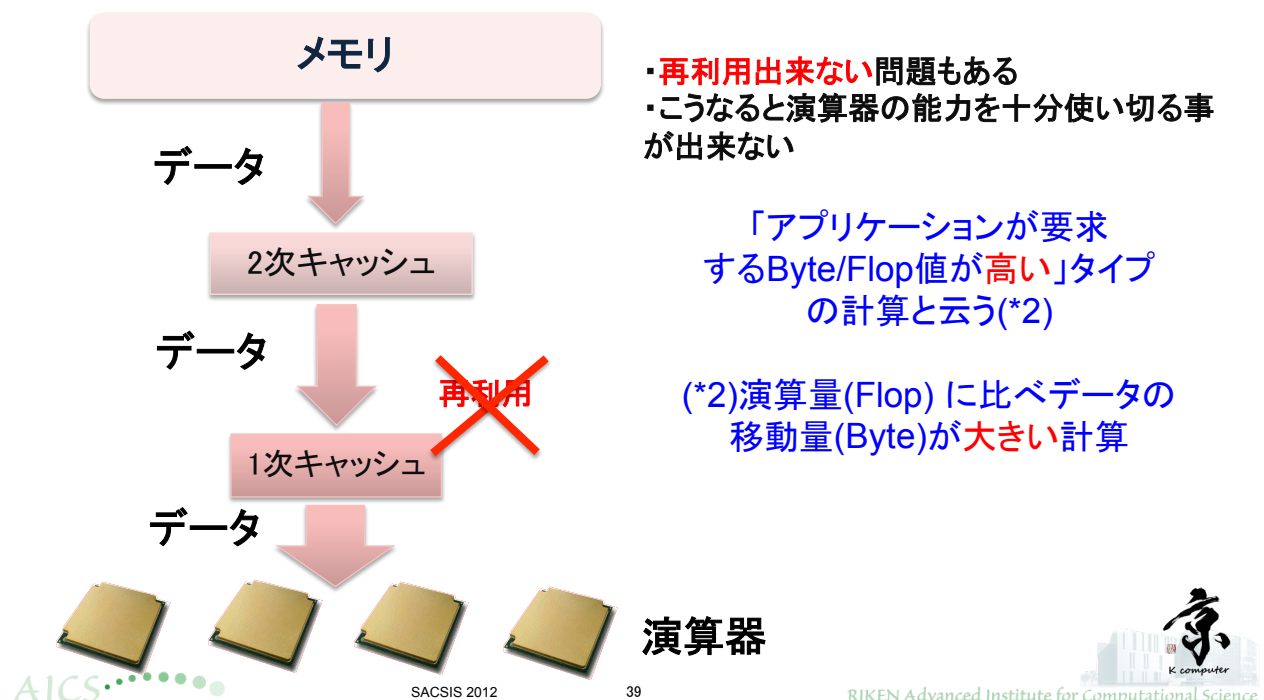
プロセッサの単体性能を引き出す(2)

メモリウォール問題への対処



プロセッサの単体性能を引き出す(3)

とは言っても……



プロセッサの単体性能を引き出す(4)

- ✓ メモリ構造、とりわけ1次キャッシュ、2次キャッシュといった多階層メモリ構造を意識してプログラミングを行う必要が顕在化。
- ✓ こうした現代の高性能計算機(HPC: High Performance Computing)においては、あらかじめ実行性能を考えたデータ構造・ループ構造等をプログラミング時に採用しなければならなくなった。



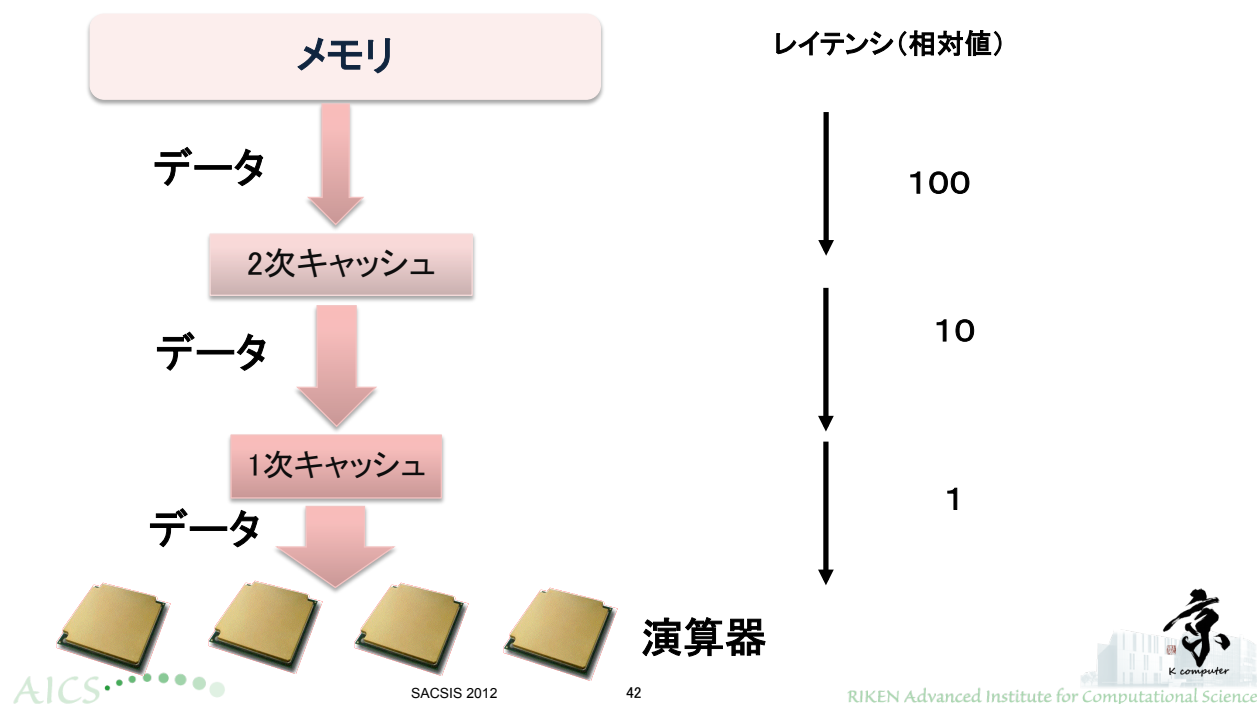
CPU単体性能をあげるためには？

CPU内の複数コアでまずスレッド並列する事は前提として

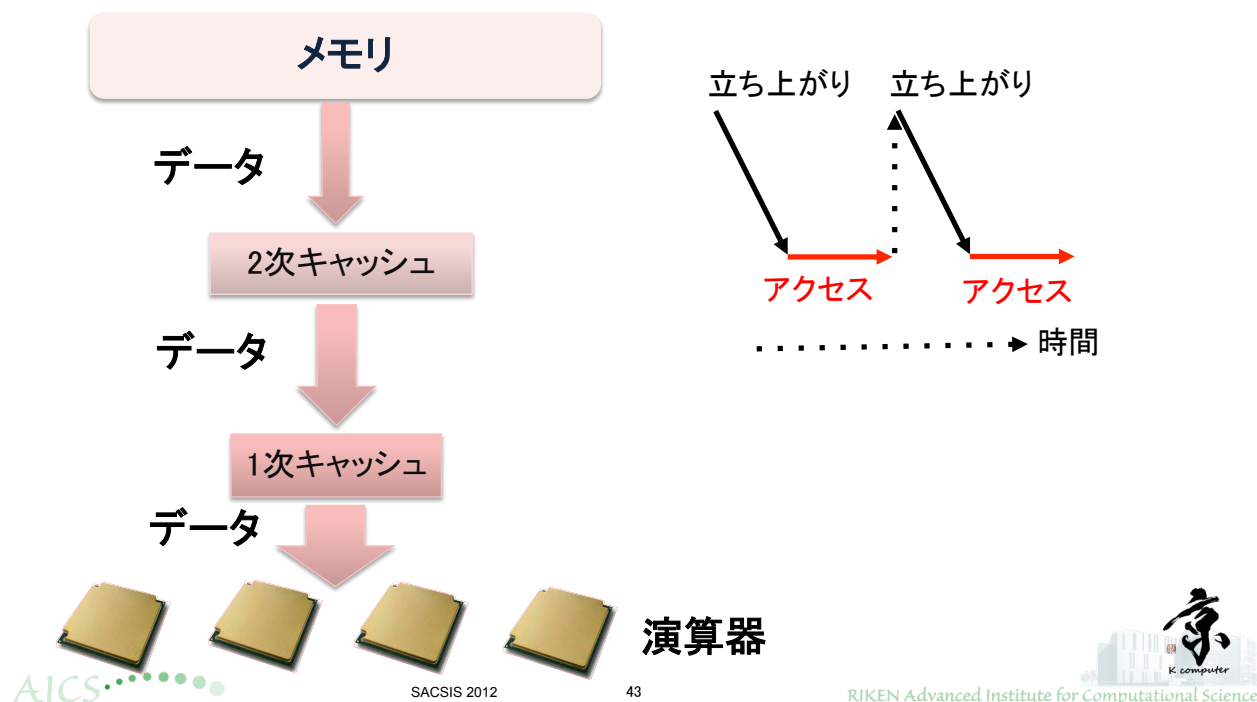
- (1)プリフェッチの有効利用
- (2)ラインアクセスの有効利用
- (3)キャッシュの有効利用
- (4)効率の良い命令スケジューリング
- (5)演算器の有効利用



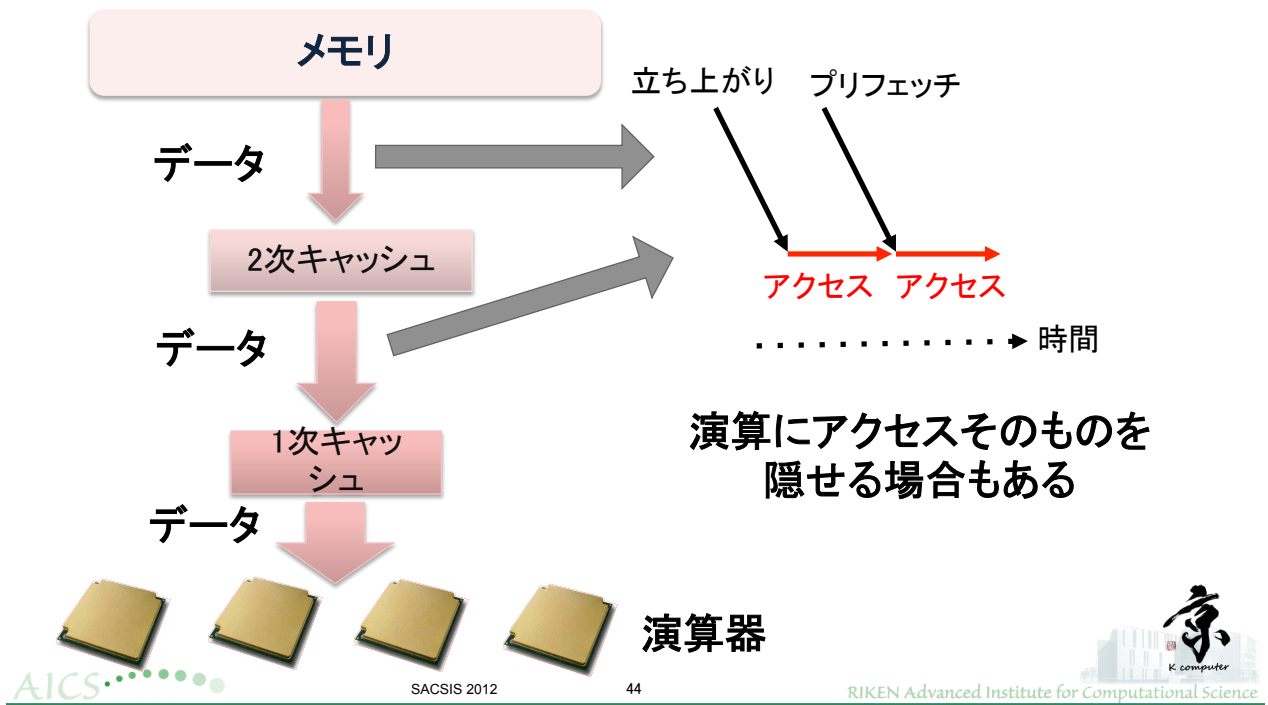
レイテンシ(アクセスの立ち上がり)



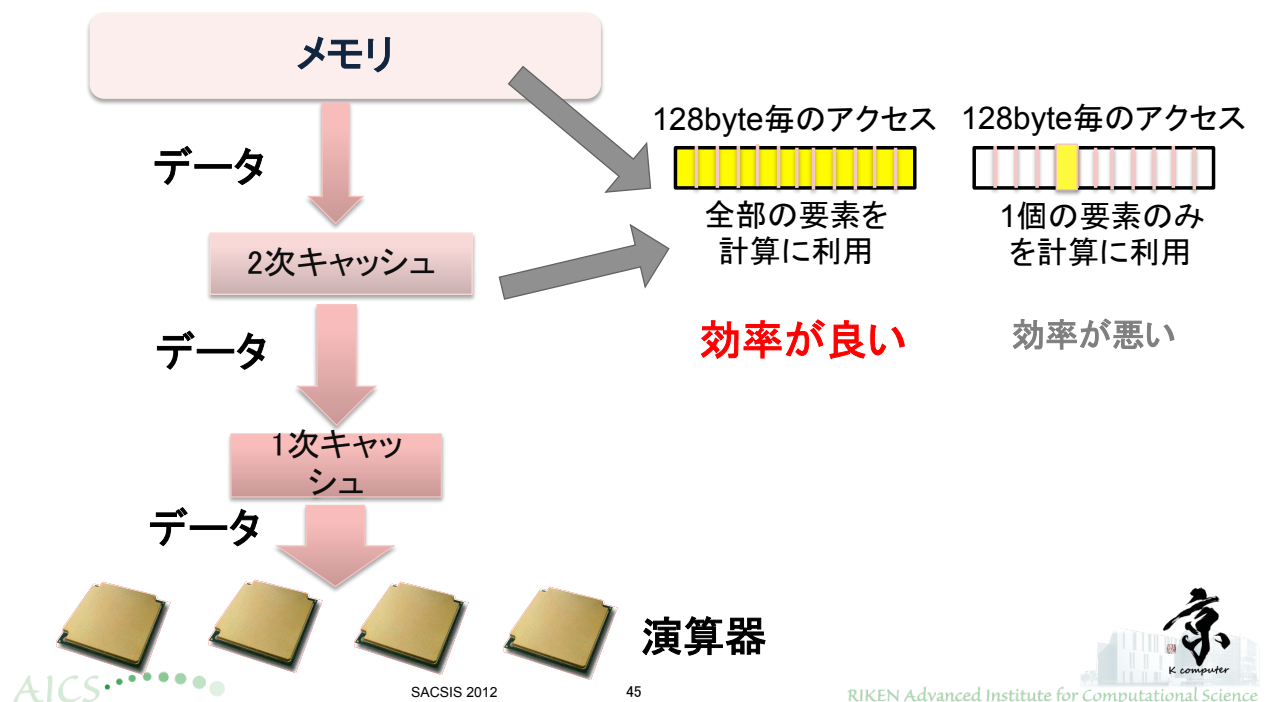
レイテンシ(アクセスの立ち上がり)



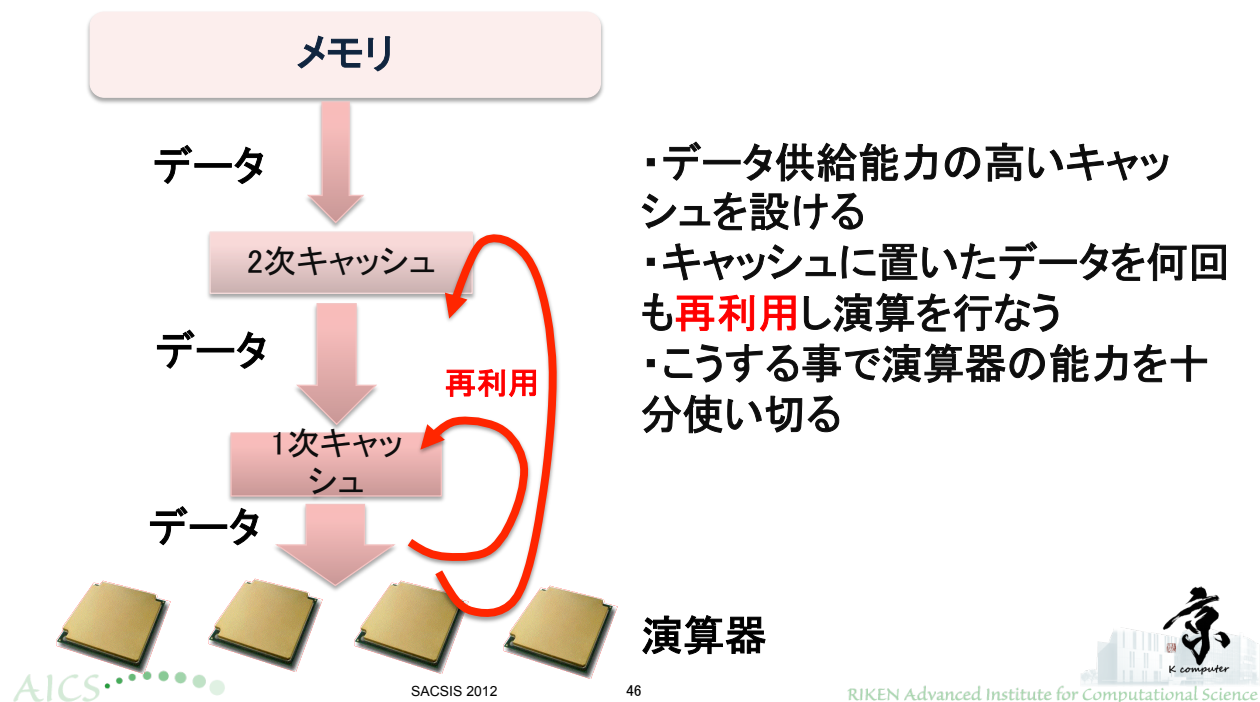
(1)プリフェッチの有効利用



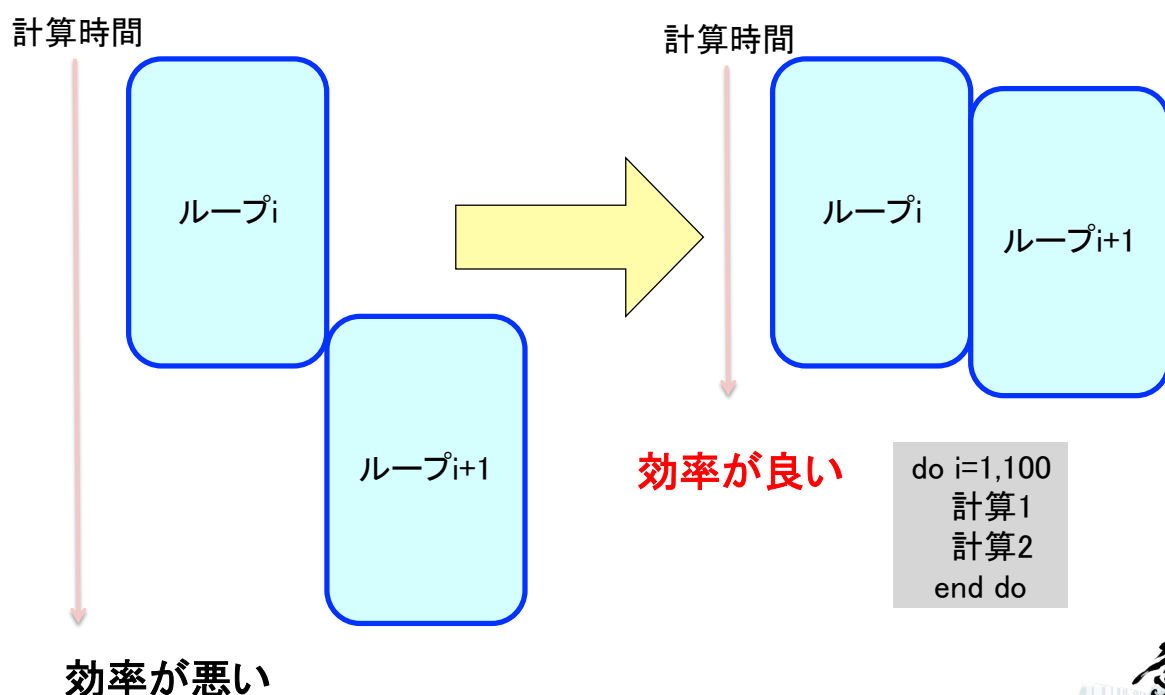
(2)ラインアクセスの有効利用



(3) キャッシュの有効利用



(4) 効率の良い命令スケジューリング



(5)演算器の有効利用

乗算と加算を4個同時に計算可能

$$(1 + 1) \times 4 = 8$$

この条件に
近い程高効率

1コアのピーク性能: 8演算×2GHz = 16G演算/秒



要求B/F値 と性能の関係



高い性能を得るための要素と 要求B/F値の関係

- 要求するB/Fが**大きい**アプリケーションについて
- ・メモリバンド幅を使い切る事が大事
 - ・一番重要なのは(1)(2)
 - ・次にできるだけオンキャッシュする(3)が重要
 - ・これら(1)(2)(3)が満たされ計算に必要なデータが演算器に供給された状態で、それらのデータを十分使える程度に(4)のスケジューリングができて、さらに(5)の演算器が有効に活用できる状態である事が必要



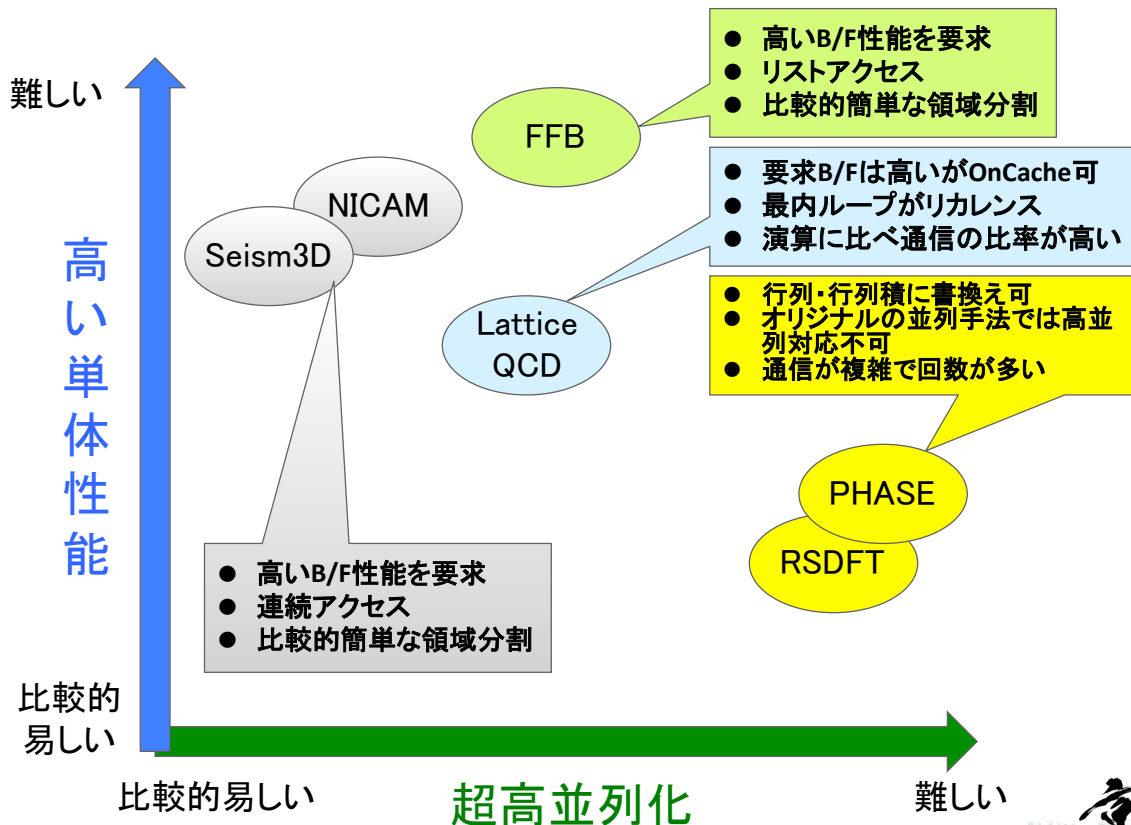
高い性能を得るための要素と 要求B/F値の関係

- 要求するB/Fが**小さい**アプリケーションについて
- ・原理的にキャッシュの有効利用が可能
 - ・まずデータをオンキャッシュにするコーディング:(3)が重要
 - ・つぎに2次キャッシュのライン上のデータを有効に利用するコーディング:(2)が重要
 - ・それが実現できた上で(4)(5)が重要



アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



RSDFT

●概要

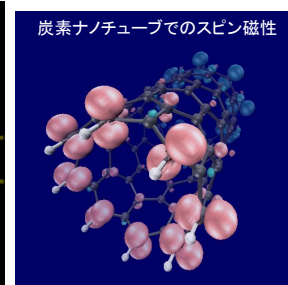
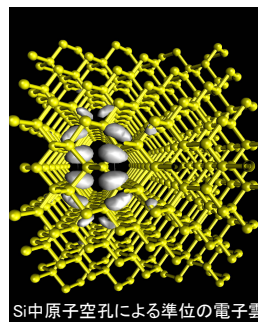
- ナノスケールでの量子論的諸現象を、第一原理に立脚して解明し、新機能を有するナノ物質・構造を予測
- 密度汎関数法の基本方程式を実空間差分法によって解き、構造安定性、電子構造、ダイナミクスを解明

●アルゴリズム

- 実空間高次差分法
- Fortran90

●どんなことが期待されるか？

- ポストスケーリング・Siテクノロジーでの材料探索を促進
- 次世代テクノロジーをブーストする新機能物質の探索
- 量子論に立脚したものづくりの指導原理を確立
- ナノおよびバイオ物質を共通の量子論的基盤で取り扱うことによる学際的学問分野を創出



どんな計算か？

対象とする系（分子・固体）に含まれる電子の数だけの波動関数 ϕ を求める

➔ 電子状態（バンド構造, 電荷密度, 状態密度）がわかる

密度汎関数法（DFT: Density Functional Theory）に基づき
Kohn-Sham方程式を解く

Kohn-Sham方程式

$$\left[-\frac{1}{2} \nabla^2 + V_{\text{LOC}}[\rho(\phi)](r) + V_{\text{non-LOC}} \right] \phi_n(r) = \varepsilon_n \phi_n(r)$$

ハミルトニアン

位置ベクトル $r(|r|^2 = x^2 + y^2 + z^2)$



$$H\phi_n(r) = \varepsilon_n \phi_n(r) \quad \text{固有値問題}$$

どんな計算か？

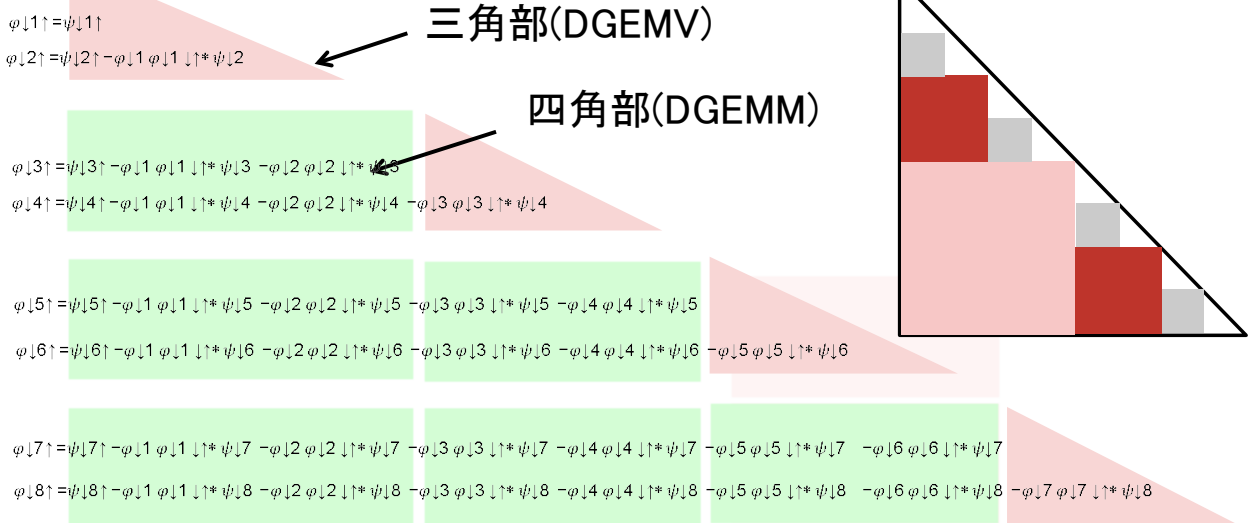
計算手順

1. 波動関数 ϕ の初期値を与える
2. **CG法**により、波動関数 ϕ を更新する
3. 波動関数 ϕ を**規格直交化（グラム-シュミット）**する
4. 局所ポテンシャル V_{LOC} を更新する。更新前後で変化が無ければ計算終了
5. ハミルトニアンを更新
6. **部分対角化**（MB×MB空間で）を行い、1.に戻る

計算コアの行列積化 – Gram-Schmidt -

ベクトル積を行列積に変換

再帰分割法



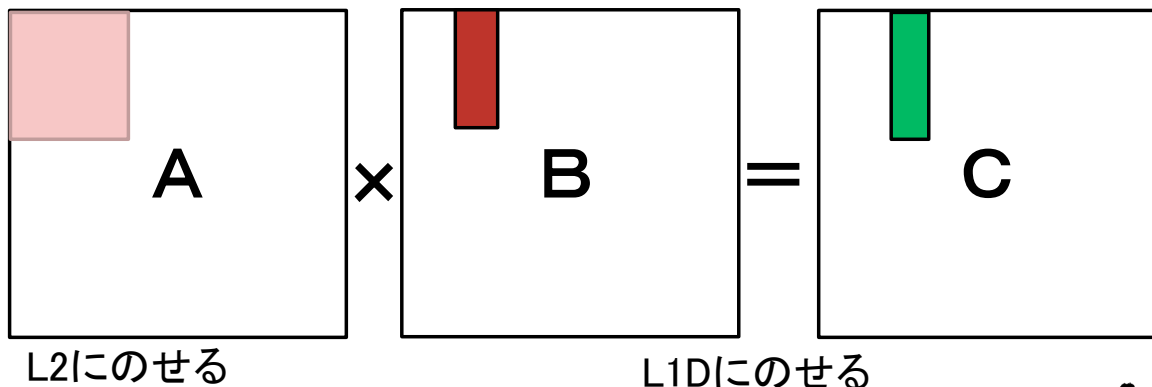
- 依存関係のある部分とない部分にブロック化して計算
- 再帰的にブロック化することで四角部を多く確保



DGEMMの性能

- ✓ 実行効率: 96.6%
- ✓ セクタキャッシュの効果: 12%程度の性能向上

行列積(C=AB) 行列をブロックに分け、L1D、L2キャッシュにのせる

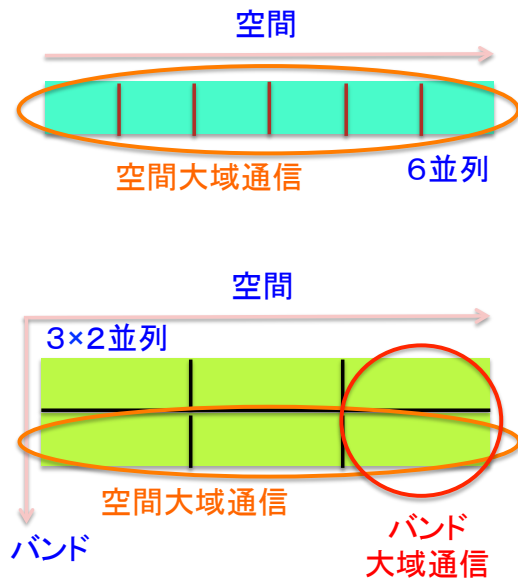


セクタキャッシュ機能で追い出しを防止



RSDFTに対する高性能化(高並列化)

- ✓ 空間並列から空間+エネルギーバンド並列(2軸並列)へ書換
- ✓ 並列軸を増やす事で10万並列レベルに対応可能となった
- ✓ 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
- ✓ 通信時間が増大を招く
- ✓ 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



RSDFTに対する高性能化(単体性能)

ML:格子数, MB:バンド数

処理ブロック	処理内容		演算量	単体性能向上手法
DTCG	ML x 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 $\frac{\langle \psi_m H_{KS} \psi_n \rangle}{\langle \psi_n \psi_n \rangle} \rightarrow \text{minimize}$	$O(ML \times ML)$ $\rightarrow O(N^2)$	CG法の計算 行列・ベクトル積
GramSchmidt	規格直交化	$H_{m,n} = \langle \psi_m H_{KS} \psi_n \rangle$	$O(ML \times MB^2)$ $\rightarrow O(N^3)$	行列・行列積へ書換
DIAG	ML次元の部分空間に限ってハミルトニアン の対角化をする.			
	行列要素生成 (MatE)	$\psi'_n = \psi_n - \sum_{m=1}^{n-1} \psi_m \langle \psi_m \psi_n \rangle$	$O(ML \times MB^2)$ $\rightarrow O(N^3)$	行列・行列積へ書換
	固有値求解 (pdsyevd)	$\begin{pmatrix} H_{N \times N} \\ \end{pmatrix} \begin{pmatrix} \vec{c}_n \\ \end{pmatrix} = \epsilon \begin{pmatrix} \vec{c}_n \\ \end{pmatrix}$	$O(MB^3) \rightarrow O(N^3)$	行列対角化 高速ライブラリ導入
	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・行列積へ書換



RSDFTコードのテスト性能

「京」の96筐体、9216プロセッサ、73728コアを使用し性能測定を実施

ピーク性能1.18Pflopsに対し**31.4%(370Tflops)**という
高い実行性能を実現

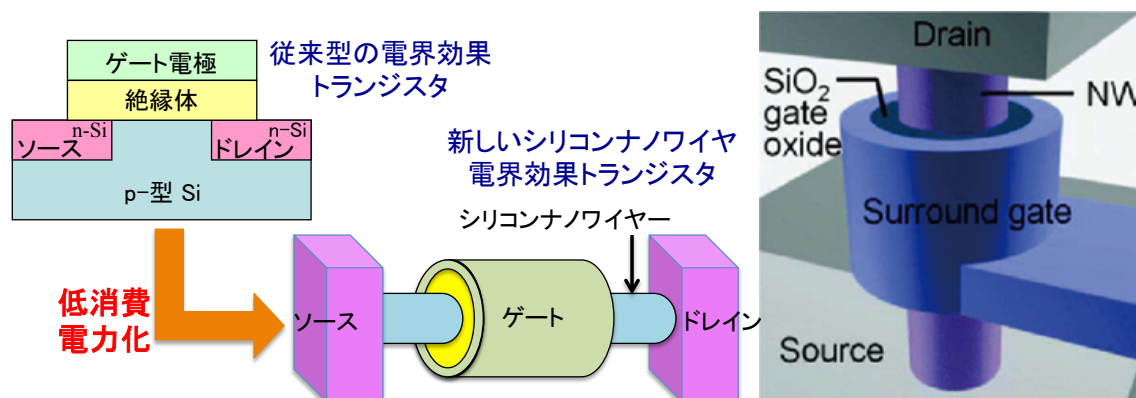
区間	全体(秒)	演算(秒)	通信時間(秒)			性能 (GFLOPS)
			隣接/空間	大域/空間	大域/バンド	
DIAG	322.490	272.488	4.042	41.723	4.237	38.381
DTCG	80.408	20.868	36.381	23.153	0.006	0.913
GramSchmidt	110.375	74.866	—	18.696	16.813	74.300
Total	513.272	368.333	40.423	83.572	21.056	40.235 ピーク性能比 31.4%

格子数 576 × 576 × 40、原子数 25236、バンド数 53280
空間分割数 32 × 48 × 2、バンド分割数 3、9216プロセッサ、73728コア



ゴードンベル賞の計算

- シリコンナノワイヤの第一原理シミュレーション
 - 次世代半導体材料として期待されているシリコンナノワイヤの電子状態、エネルギー等をシミュレーションで求めることが目的。
 - 576ラック(7.07ペタフロップス)を用いた計算を実施し高い性能を実現。
 - 世界で初めて約40,000原子の電子状態計算に成功。



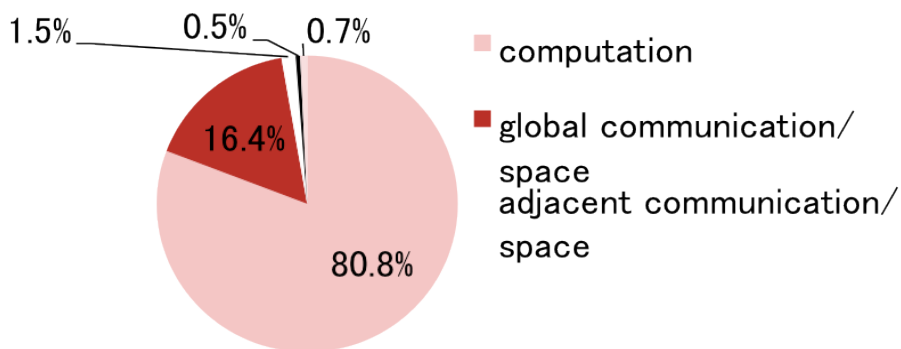
(※)ゴードン・ベル賞 (ACM Gordon Bell Prize) は、並列計算技術の向上を目的として、米国計算機学会 (ACM) にて運営され、毎年、ハードウェアとアプリケーションの開発において最高の成果をあげた論文に付与される賞である。毎年11月に開催される米国スーパーコンピュータ会議 (SC: Supercomputing Conference) にて表彰式が行われる。



Challenge to 100,000 atoms simulation

- Sustained performance is **3.08 PFLOPS** /SCF.
- **43.6 %** efficiency to the peak performance.
- Communication cost is 19.0% of all execution times.
- One iteration time of SCF is 5,500 sec. (1.5 hours)

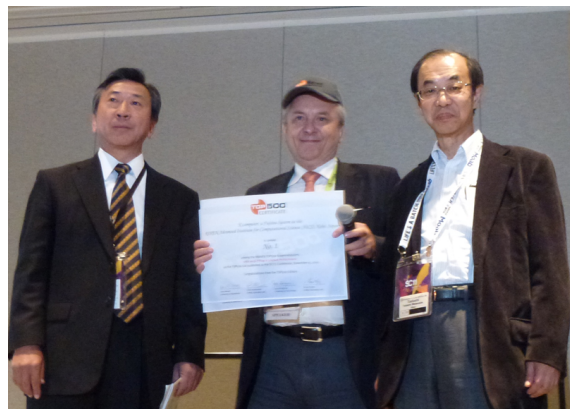
Distribution of computational cost



SACSIS 2012

64

TOP500 SC11(2011年11月)



ゴードンベル賞



アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



PHASE

擬ポテンシャルと密度汎関数法によるナノ材料第一原理分子動力学プログラムである。局在基底ではなく平面波基底を用いることにより、分子から固体まで多くの物質に対して高精度な電子状態計算が可能である。

第一原理分子動力学計算 = 厳密かつパラメータが入らない

…電子の振る舞いを量子力学Schrödinger方程式を用いて解く

$$\left\{ -\frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{r}) \right\} \Psi_i(\mathbf{r}) = E_i \Psi_i(\mathbf{r})$$

厳密には解けないので、近似する(密度汎関数法)

…電子の海(電場)中の電子の運動におきかえる(多電子問題→1電子近似)。

$$\left\{ -\frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{r}) + V_{xc}[\rho(\mathbf{r})] \right\} \varphi_i(\mathbf{r}) = \varepsilon_i \varphi_i(\mathbf{r})$$
$$\rho(\mathbf{r}) = \sum_i \varphi_i(\mathbf{r})^2$$

これはハミルトニアンという微分演算子に対する固有値問題である。ハミルトニアンが決まれば、エネルギーと波動関数が求まり、波動関数が決まれば電子分布が変わるため、ハミルトニアンも変化する。この収束計算を行う(SCF)。



■カーネルの抽出

PHASEにて抽出された、カーネルは以下の11区間である。

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 A 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 $H\psi$ 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

3種類に分類

種類	区間番号
行列-行列積に書き換え可能	2,4,5,8,9,10
FFTを含む	1,3,6,7,8,11
対角化	9

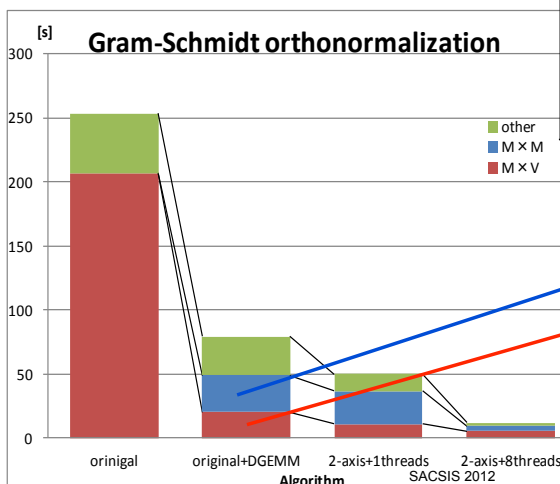
- 赤字は性能評価したカーネル
- オリジナルコードはハイブリッド並列に対応していない
- 「京」での性能比較データは, 1threadsの評価結果である,



■BLAS Level3適用

- RSDFTと同様にGram-Schmidtの直交化を行列・行列積に書換え
- Gram-Schmidtの直交化を含むカーネル(区間5)にBLAS Level3を適用
- FX1,「京」16並列で測定
- HfSiO₂ 768原子アモルファス系にて性能評価を実施

「京」



サブルーチン	FX1			「京」		
	時間 [sec]	比率 [%]	演算効率 [%]	時間 [sec]	比率 [%]	演算効率 [%]
区間5	512.7	100.0	28.23	11.8	100.0	23.07
m_ES_F_transpose_r	105.3	20.5	0	0.0	0.1	0.00
m_ES_W_transpose_r	15.7	3.1	0	0.1	0.5	0.00
WSW_t	15.2	3.0	0.036	1.2	9.8	0.10
normalize_bp_and_psi_t	0.9	0.2	3.25	0.4	3.3	0.13
W1SW2_t_r	49.2	9.6	5.46	3.7	31.8	2.30
modify_bp_and_psi_t_r	50.8	9.9	4.45	1.7	14.8	3.75
W1SW2_t_r_block	162.0	31.6	41.89	2.3	19.4	56.82
modify_bp_and_psi_t_r_block	96.2	18.8	74.65	2.1	17.6	60.90
m_ES_W_transpose_back_r	14.1	2.8	0	0.0	0.3	0.00
m_ES_F_transpose_back_r	1.3	0.3	0	0.0	0.1	0.00

FX1(左), 「京」(右)上でのGram-Schmidt直交化のBLAS Level3適用結果

□ : RSDFTと同様の四角計算部

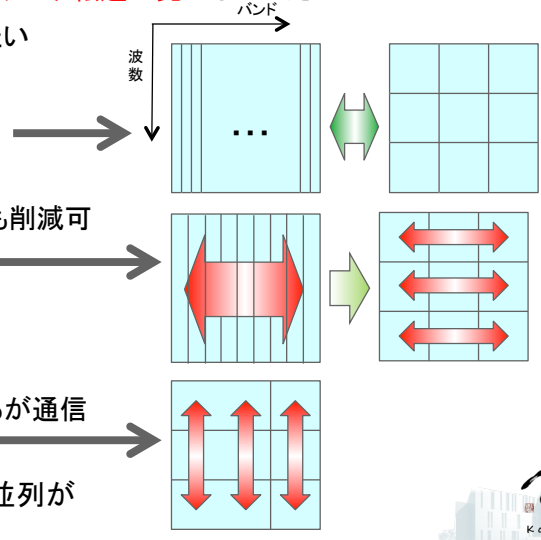
△ : RSDFTと同様の三角計算部

演算効率4~5%→40~70%を達成した。



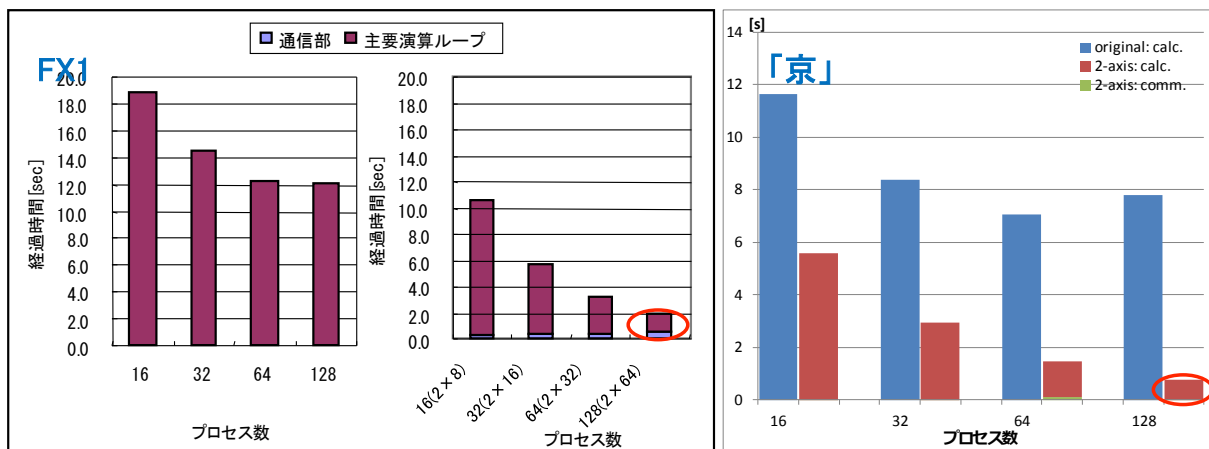
■二軸並列化

- PHASEのような量子力学計算では以下の2つの並列軸がある
 - **バンド方向**: 粒子(バンド)インデックス方向で, 電子数分ループ長がある.
 - **波数方向**: 波動関数の離散化軸である. PHASEではフーリエ級数展開係数方向
- PHASEではバンド並列を基本とし一部波数並列を行う**一軸並列**を採用している
- 一軸並列では10万原子系以上でない**と64万コアの並列度に対応できない**
- またバンド並列とは数並列の間にall to allの**データ転送が発生していた**
- 現実**はより小さい問題を短い時間で計算したい**
- **10,000原子系を64万コアで解きたい**
- このため**バンドと波数の二軸並列化を実装**
- 二軸並列の長所
 - 高並列に対応可能となりall to allの転送も削減可
 - **バンド方向の通信時間が減る**
- 二軸並列の短所
 - コード改変量が膨大
 - **波数方向のFFTについて通信が発生するが通信はバンドのグループ内に閉じる**
- 上記の長所・短所を定量的に見積り二軸並列が**有利と判断した**



■BLAS Level3の二軸並列化の効果

- BLAS Level3化されたカーネルに(区間2)についての二軸並列化の並列性能を測定した(HfSiO₂ 384原子アモルファス系). 左はオリジナルのバンド並列時の並列性能, 右は二軸並列時の並列性能である.

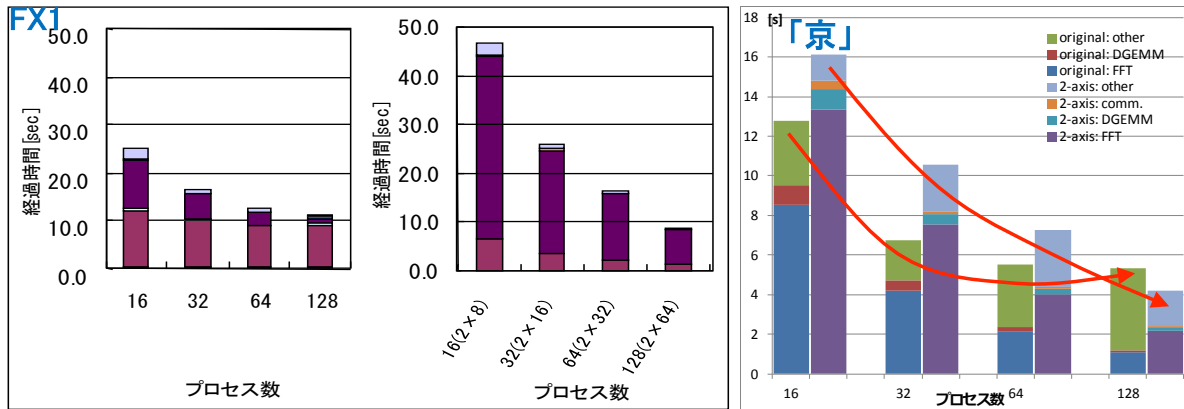


二軸並列化に伴い, **わずかに新たな通信は発生しているが, 並列性能が向上している.**

■FFTの二軸並列化

■ 評価結果

FFTを含むカーネルに(区間8)についての二軸並列化の並列性能を測定した(HfSiO₂ 384原子アモルファス系). 左は一軸並列時の並列性能, 右は二軸並列時の並列性能である. FFTは, FFTWを用いている.



低並列時には, **波数方向の分割数が減少**したため, オリジナルのものより性能が低い, 高並列になるにつれ**並列性能**が向上する.

■対角化計算の並列化

■ 高速並列対角化ルーチンの導入

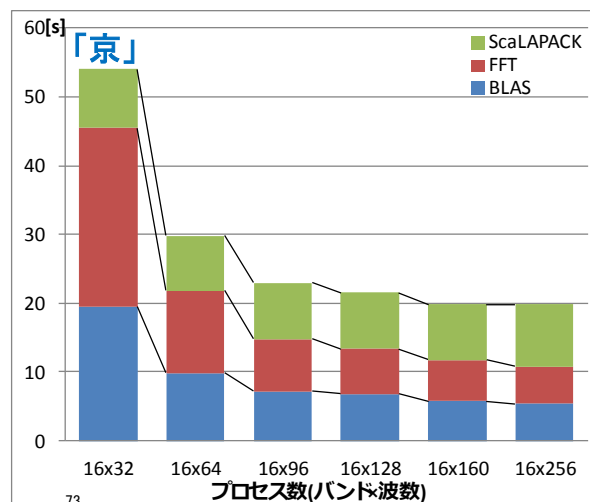
量子力学計算は多原子の計算が困難であったため, 対角化に必要な元数はバンド数程度(1,000程度)であり, 従来は並列化が不要な箇所であった. しかし超並列での計算では, 原子数が10,000を越えるため, 並列化を実施した.

■ 高速対角化ルーチンの分割数の固定

3,000~6,000原子程度の系(元数=10,000~20,000)にて, 性能評価を行うと, 分割数=16x16=256以上はオーバーヘッドが見られたため分割数を16x16(=256)に固定した.

■ 総合性能

- 現在トータル数ペタを使用した性能測定を実施しながら最終チューニングを実施中
- FFTを使用したコードとしては高性能である対ピーク性能比20%以上を得られる見通し
- 数千原子の計算を1~2万ノードを使用し計算できる見通しが



アジェンダ

1. 次世代スーパーコンピュータプロジェクトについて
2. 京速コンピュータ「京」の概要について
3. 「京」の応用分野
4. 現代のスパコン利用の難しさ
5. 理研で進めているアプリ高性能化
6. 高並列化のための重要点
7. 高い単体性能を得るための重要点
8. 理研重点アプリの特徴
9. RSDFTの高性能化
10. PHASEの高性能化
11. Seism3D,FFBの高性能化
12. まとめ



Seism3D,FFB

- ◆ Seism3DとFFBは基本的に疎行列とベクトルの積であるので、その観点で議論する

疎行列とベクトルの積の問題点

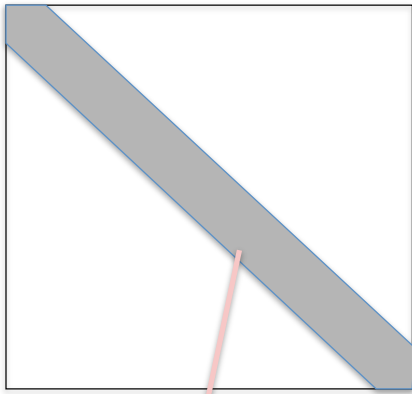
- ◆ CPU単体性能チューニングにおいては、対象とするコーディングの限界性能値が分からない
- ◆ 要求B/F値が高いコーディングに対するスカラチューニング

ここでの議論の狙い

- ◆ どの段階までチューニング作業を進めるかの判断基準を設ける事を目的に疎行列とベクトルの積のコーディングから性能を予測する手法を提案
- ◆ Seism3D及びFFBを対象に「京」において更なる性能向上のための具体的なチューニング手法を提案



疎行列とベクトルの積の特徴



- ベクトルは一般的に3次元配列である
- しかしコーディング的には1次元や2次元配列で表現されている場合もある
- ベクトルは行列の列幅程度の再利用性がある
- したがって量的には行列の列幅分の1程度の大きさである
- メモリバンド幅を消費するが再利用性を生かせるかが重要
- 一次元の量としてリストアクセスする場合がある
- その場合はリストは行列と同じ量が必要となりメモリバンド幅を消費する

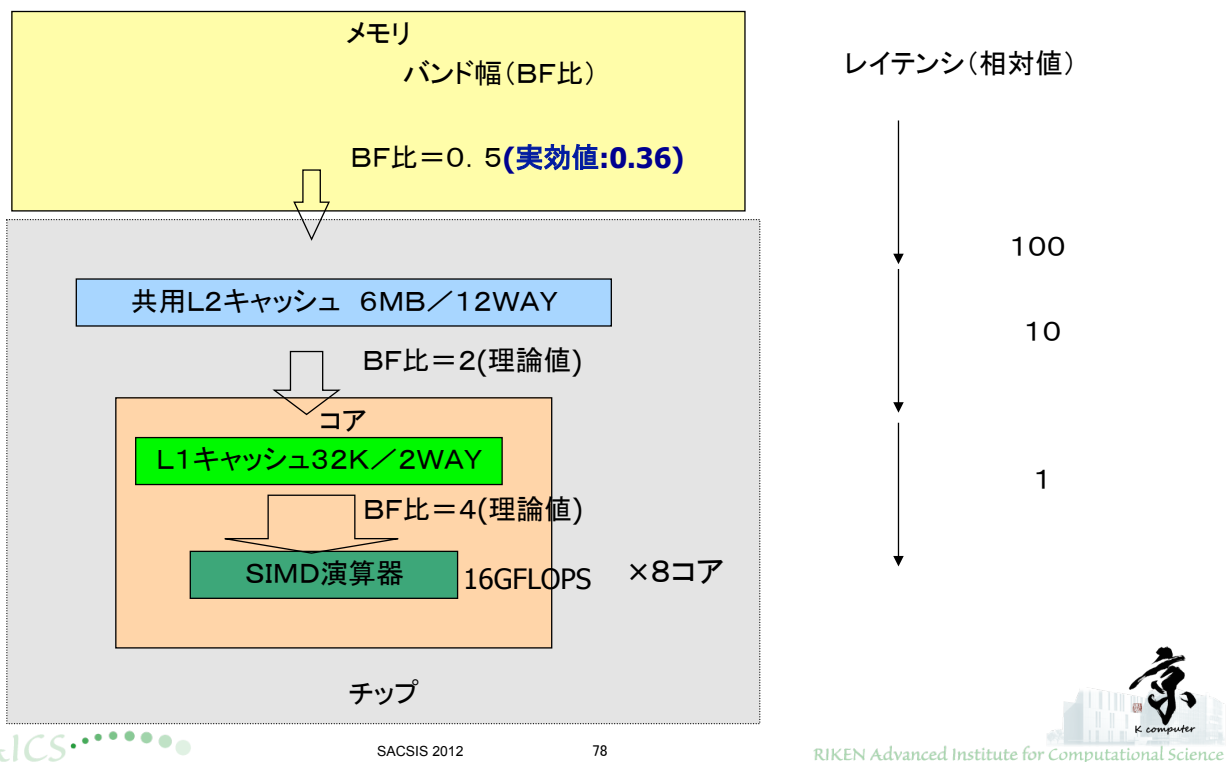
- 行列は一般的に3次元の配列である
- しかしコーディング的には1次元や2次元配列で表現されている場合もある
- ただ量的には大きいのでメモリバンド幅を消費する場合が一般的
- しかし物理的に2次元の量である場合やスカラー量である場合がある

76

SACSI 2012

性能予測手法

ベースとなる性能値

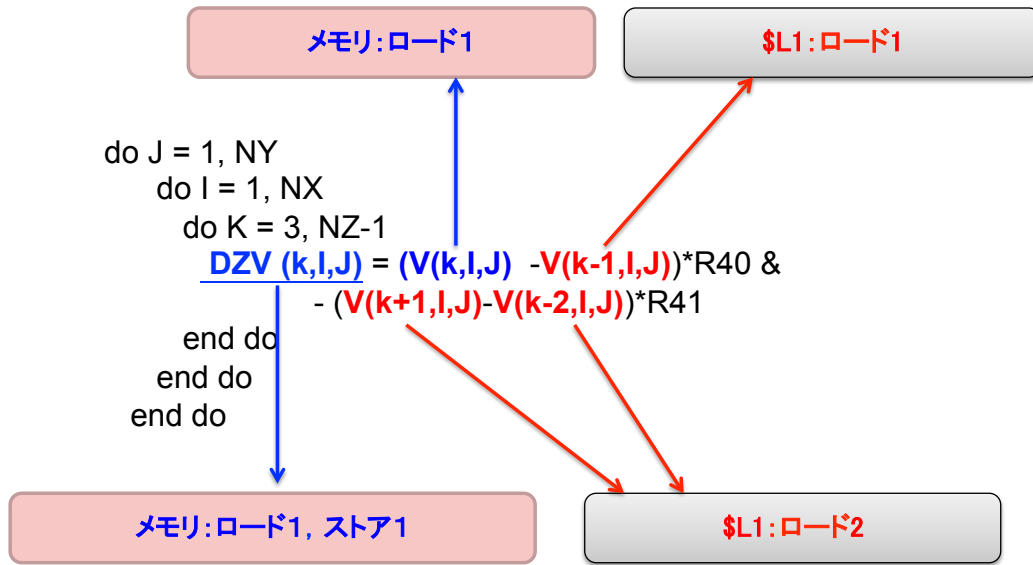


各階層の論理/実効スループット

	L1	L2	Mem
論理	64GB/s /core	256GB/s /chip	64GB/s /chip
論理B/F	4	2	0.5
実効	----	----	46GB/s/ chip(*)
実効B/F	----	----	0.36

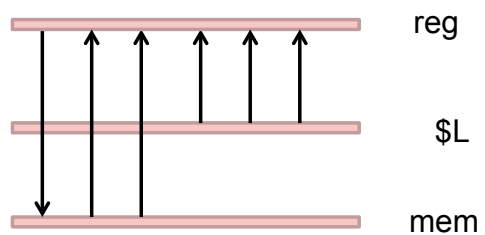
(*)STREAMベンチマークの値

メモリとキャッシュアクセス(1)



メモリとキャッシュアクセス(2)

```
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
```



	Store	Load	バンド幅比 (\$L1)	データ移動時間の比(L1)	バンド幅比(\$L2)	データ移動時間の比(L2)
\$L	1	5	11.1 (8*64G/s)	0.5= 6/11.1	5.6 (256G/s)	1.1=6/5.6
M	1	2	1(46G/s)	3=3/1	1(46G/s)	3=3/1

データ移動時間の比を見るとメモリで律速される
→ メモリアクセス変数のみで考慮すれば良い。



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:

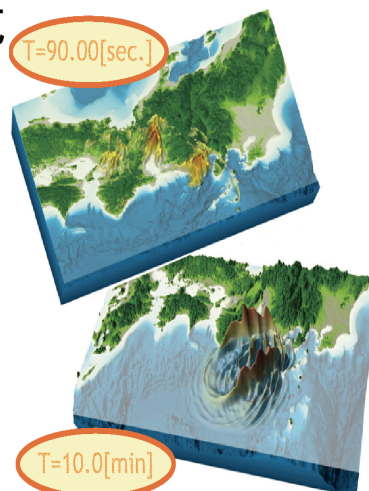
add : 3 mult : 2 = 5

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

Seism3D

- 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
- 地震伝播と津波を連動して解く
- 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

- 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 チューニング



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
```

- 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



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
```

要求B/F	12/5 = 2.4
性能予測	0.36/2.4 = 0.15
実測値	0.151

- 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



空間微分Z方向の計算a)b)(ZXループ融合)

要求B/F値を下げる

```
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))*R42 &
        - (V(k+1,I,J)-V(k-2,I,J))*R43
      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
```

要求B/F	20/10 = 2
性能予測	0.36/2 = 0.18
実測値	0.174

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

要求Byteの算出:

2store,3loadより、

(2+3)*4 = 20

要求flop:

add : 6 mult : 4 = 10



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

要求Byteの算出:

1store/5loadより

$(5+1) * 4\text{byte} = 24$



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

$4 \times 3 = 12\text{byte}$

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

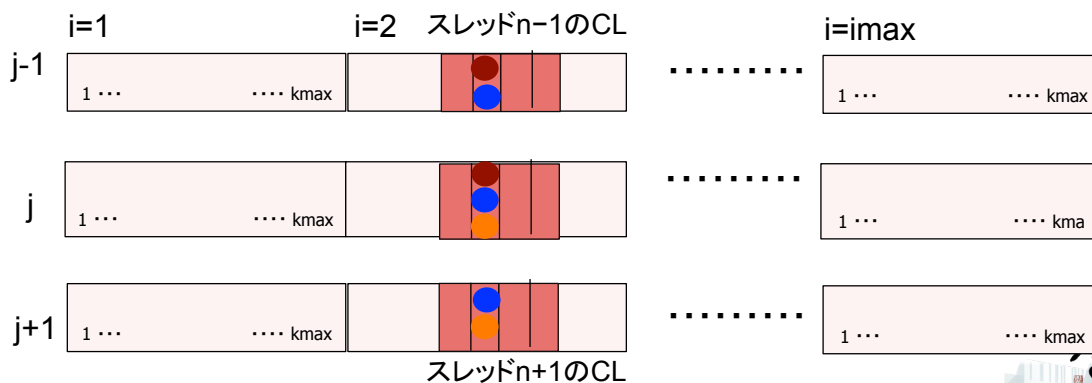
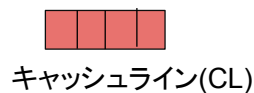


(cyclic分割スレッド並列の説明)

```

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),PRIVATE(I,J,K)
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))*R42 &
- (V(k+1,I,J)-V(k-2,I,J))*R43
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
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
    
```

**要求B/F値を下げる
キャッシュに載せる**

- K,I,J軸差分のループを融合することにより、V(K,I,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 = 28byte

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

応力時間積分の計算c)

オリジナルコードの結果

要求B/F	252/175=1.44
性能予測	0.36/1.44 = 0.25
実測値	0.174

- 実測値が低い
- 使用したメモリバンド幅の測定結果は 35GB/sec 程度
- ハードウェアプリフェッチが効率的に出ていない事が判明
- いくつかの配列を融合し配列の数を減らす事でストリームの数を削減
- ハードウェアプリフェッチの効率を向上
- その結果実測値が**0.218まで向上**
- メモリバンド幅も42.4GB/secまで向上



Seism3D全体のチューニング結果

(*)通信を含む性能

	オリジナル 時間(秒) peak比		チューニング 時間(秒) peak比	
全体(*)	75.5	10.3%	52.9	15.3%
a)応力空間微分	13.0	10.4%	9.2	14.7%
b)速度空間微分	13.4	10.1%	7.7	17.7%
c)応力時間積分	12.8	17.0%	11.5	21.8%
d)応力時間積分吸収	12.5	7.6%	10.3	10.2%
e)速度時間積分	9.7	7.5%	3.0	23.0%
f)速度時間積分吸収	7.9	15.3%	6.9	17.5%



Front Flow/blueの チューニング



Front Flow/blue(FFB)の概要

- 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
 - 全体剛性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
- 本講演の疎行列とベクトルの積は前者のソルバで使用される計算カーネル



オリジナルコードの性能予測(理想的なケース)

オリジナルコード

```

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
    
```

リスト (ICRS=0, DO 110 IP=1,NP)

ベクトル (BUF=BUF+A(ICRS)*S(IP2))

行列 (AS(IP)=AS(IP)+BUF)

- 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Byteの算出:

単精度 : 2 load なので

$$2 * 4 = 8 \text{ byte}$$

要求flop:

$$\text{add} : 1 \quad \text{mult} : 1 = 2$$

要求B/F	$8/2 = 4$
性能予測	$0.36/4 = 0.09$

(スレッド並列を仮定しピーク性能128Gflopsに対して)



オリジナルコードの性能予測と実測 (スレッド並列なし: 1コア)

- 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 $2 * 4 \text{ byte} = 8$

要求flop : $1(\text{add}) + 1(\text{mult}) = 2$

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)

(スレッド並列なしピーク性能16Gflopsに対して)

- 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
- 1ラインのうち1要素しか使用し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



チューニング1: フルアンロール

狙い:

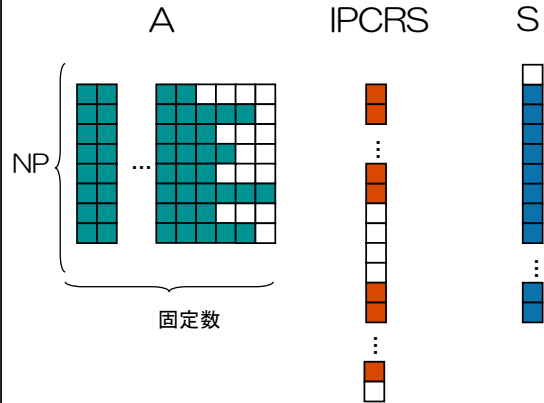
- スケジューリングの改善(演算待ちの削減)

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)

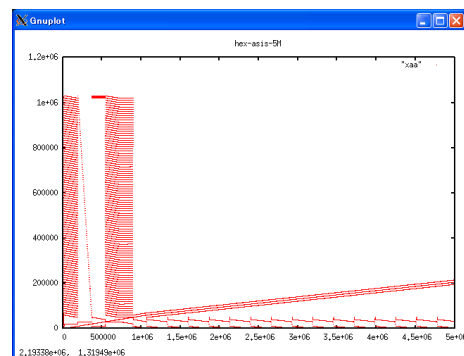
狙い:

- 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

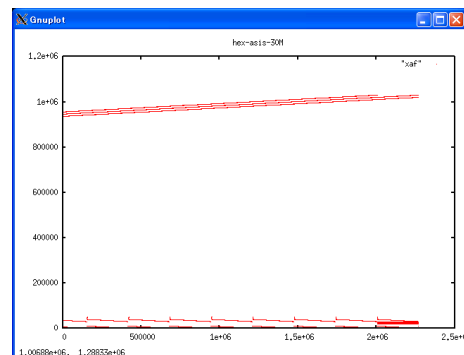
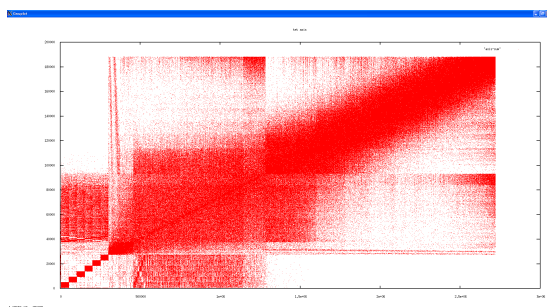
オリジナルデータの特徴:

- 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

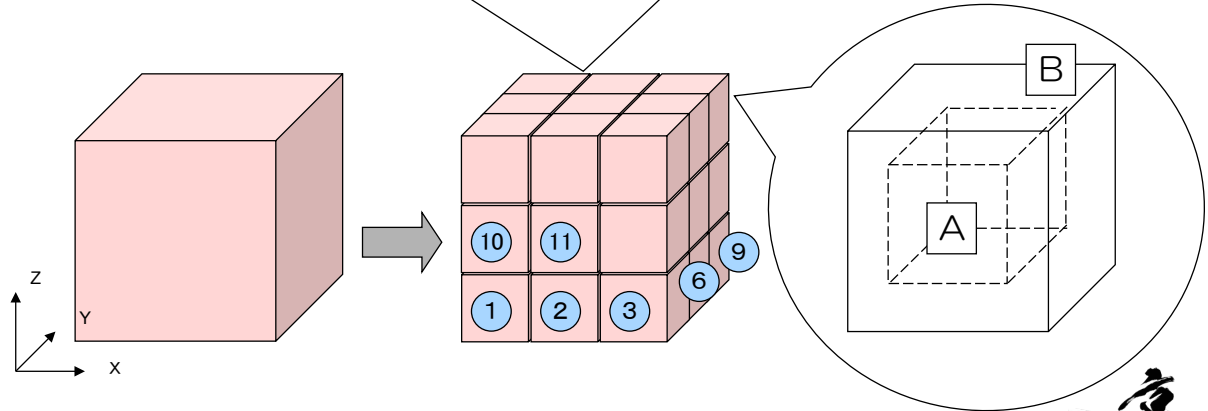


チューニング2: リオーダーリング (2/4)

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

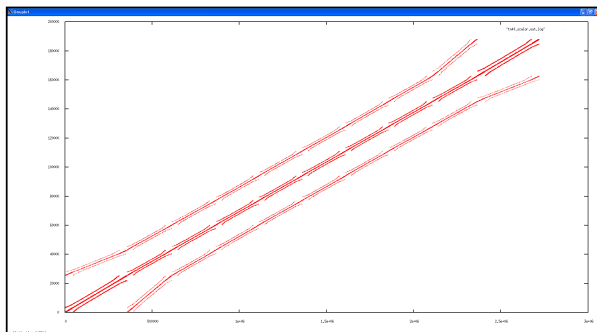
ひとつのブロックを外と内に分け、内側(A)のナンバリング後、外側(B)のナンバリングを実施(内:外の比 8:2)



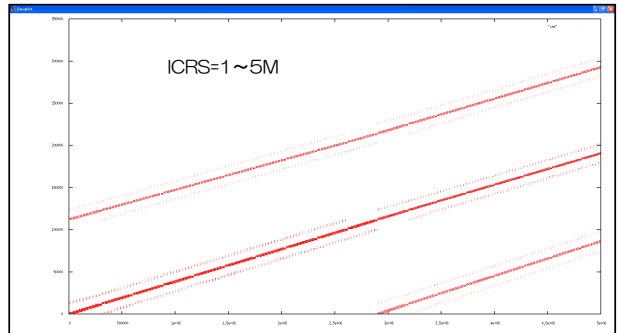
チューニング2: リオーダーリング (3/4)

- 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

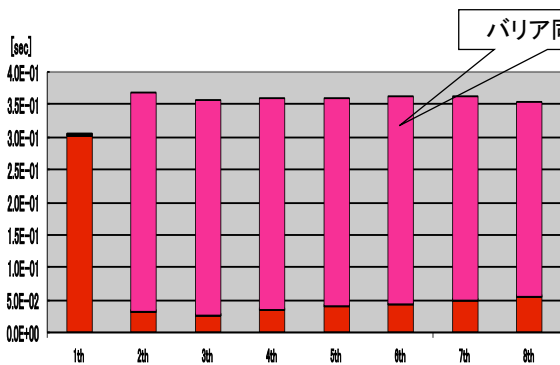
■4面体 リオーダーリング結果



■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)

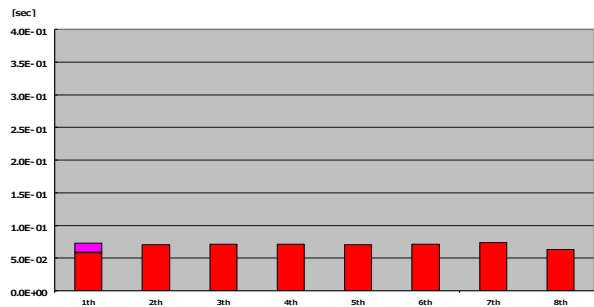


リオーダーリング前

- 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クセスコストが増大
- 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
- メモリアクセスバランスの改善と同期コストの大幅な削減



FFBカーネルの結果まとめ

	6面体	4面体
オリジナル(1core)	5.9%	2.4%
フルアンロール (1core)	10.8%	4.2%
フルアンロール (8core)	5.4%	3.0%
フルアンロール + リオーダーリング (1core)	10.2%	10.2%
フルアンロール + リオーダーリング (8core)	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

まとめ

- 次世代スーパーコンピュータプロジェクトについて概要を示した
- 京速コンピュータ「京」の概要を示した
- 理研で進めているアプリ高性能化の概要を示した
- 高並列化のための重要点を示した
- 高い単体性能を得るための重要点を示した
- 理研重点アプリの特徴を示した
- RSDFT—主に二軸並列の効果について示した
- PHASE—行列・行列積化, 二軸並列化, FFT, 対角化ルーチンに係るチューニングについて示した.
- Seism3D,FFB—疎行列とベクトルの積について, 性能推定手, チューニング手法の実例を示した.