



# 数値計算ライブラリの観点 ～「京」世代、次々世代、 エクサ世代～

片桐孝洋

東京大学情報基盤センター

戦略的高性能計算システム開発に関するワークショップ

日時：2010年8月2日13:10～21:30

アプリケーション&ライブラリ（座長：中島研吾@東大）13:40～13:50

キャッスルイン金沢 1階ピア

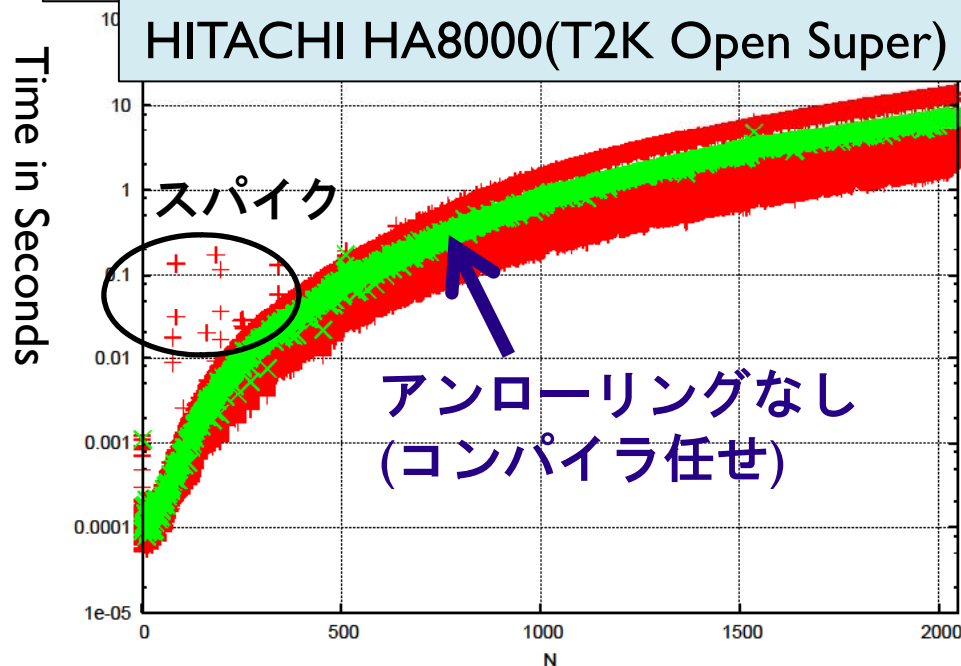
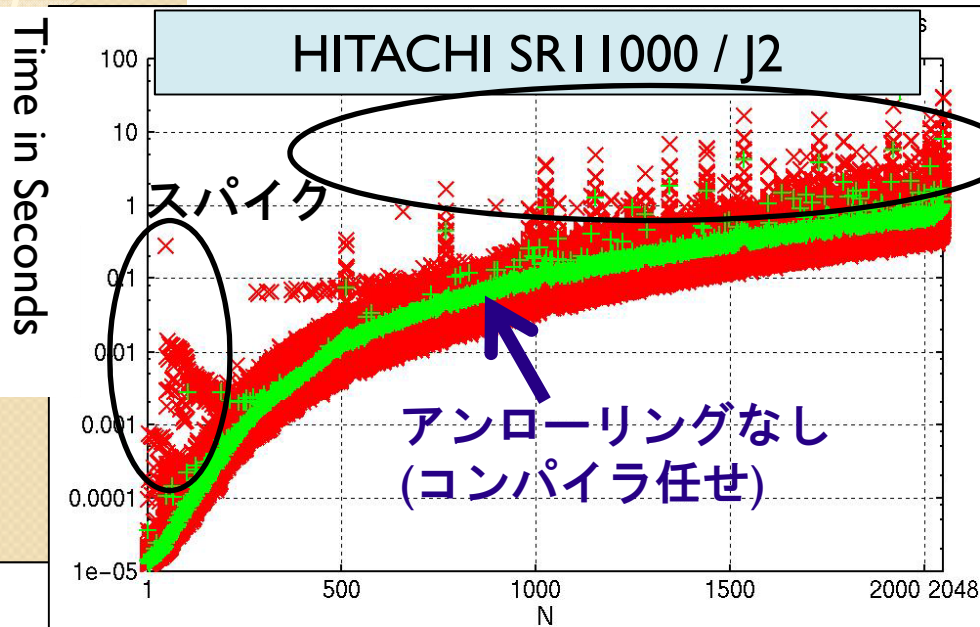
# 自己紹介

- 数値計算屋   ！＝数値解析屋
- 詳細な分類
  1. 数値解析屋（誤差解析、**頭の中の並列性**に興味）
  2. 数値ライブラリ屋（数値アルゴリズムとソフトウェア利便性に興味）
  3. カーネル屋（計算機実装に興味）
    - **1の下部～3の上部** に興味
- 数値解析屋が「逐次処理」と思っている数値計算演算を
  - **数式レベルの並列性**
  - **データ並列性**
  - **命令レベル並列性**を駆使して並列化してしまう人



# ○ 計算機システムの課題

# ノード内での性能不安定性



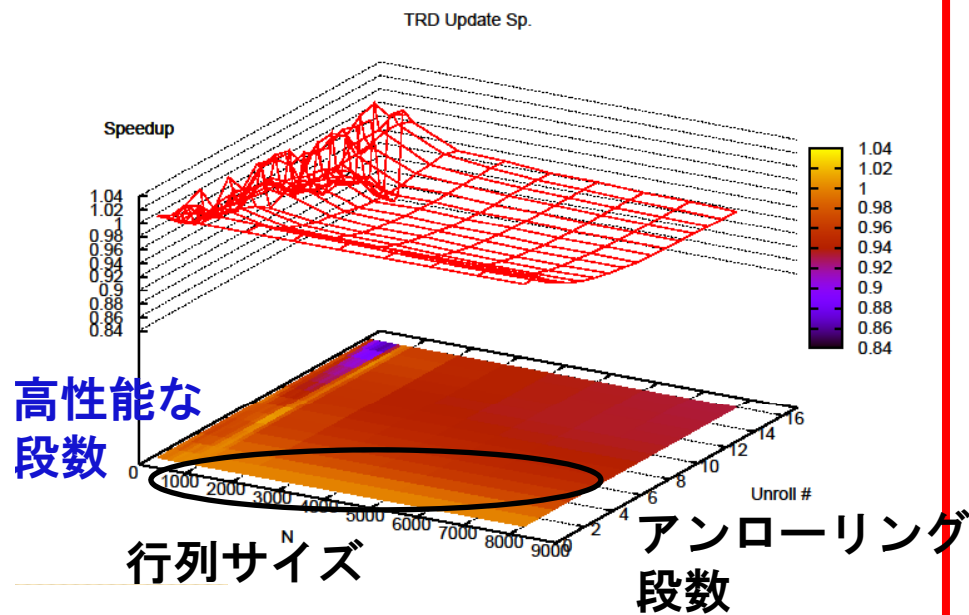
- 密行列の行列-行列積  
BLASを用いていない単純コード
- 3重ループ (i,j,k), アンローリング1段~4段
- 次元Nに関し 4\*4\*4=64種類の実装
- 1 から2048次元まで1刻みのデータ.
- コンパイラ HITACHI Optimized Fortran90.  
オプション: -Oss
- 自動並列化 (ノード内)
- 計算機構成:
  - HITACHI SRI 1000/J2
  - HA8000 (T2K Open Supercomputer (Todai Combined Cluster))
  - Installed in Information Technology Center, The University of Tokyo.
  - 16コア/ノード.

- コンパイラ任せは遅い
- 常時<特定の実装>が速くない
- 10倍ぐらい実行時間がぶれる (実行時間の<不安定性>)
- 場合により100倍も遅い!

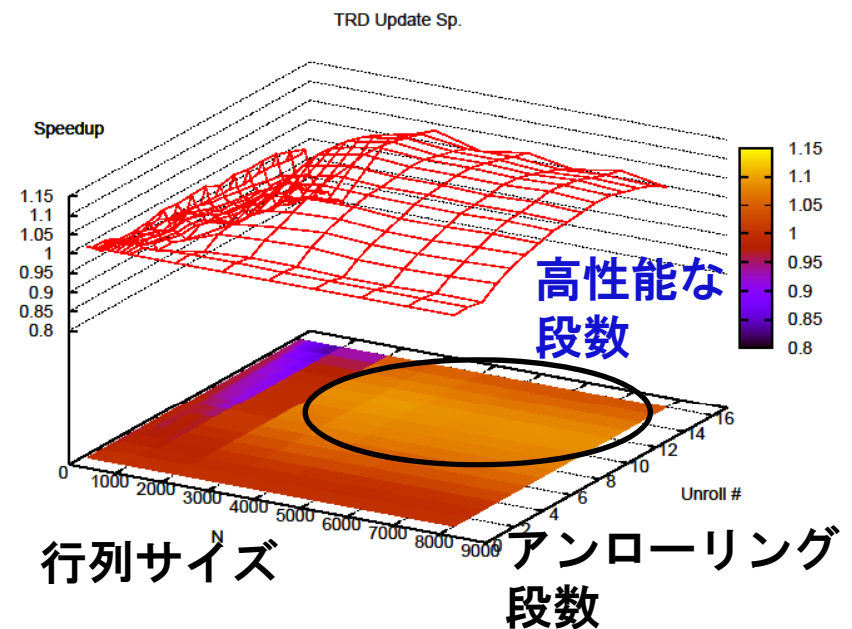
# マルチコア実行時利用形態による 性能挙動変化 (T2Kオープンスパコン)

## 対称固有値ソルバの三重対角化 (行列更新部分)

4コア (4コア / 1ソケット実行)



4コア (1コア / 1ソケット実行)



ソケットごとの共有L3キャッシュの影響

→ (動的な) 実行コア数により最適実装が変わる!

# 次世代システムへの要求（1）

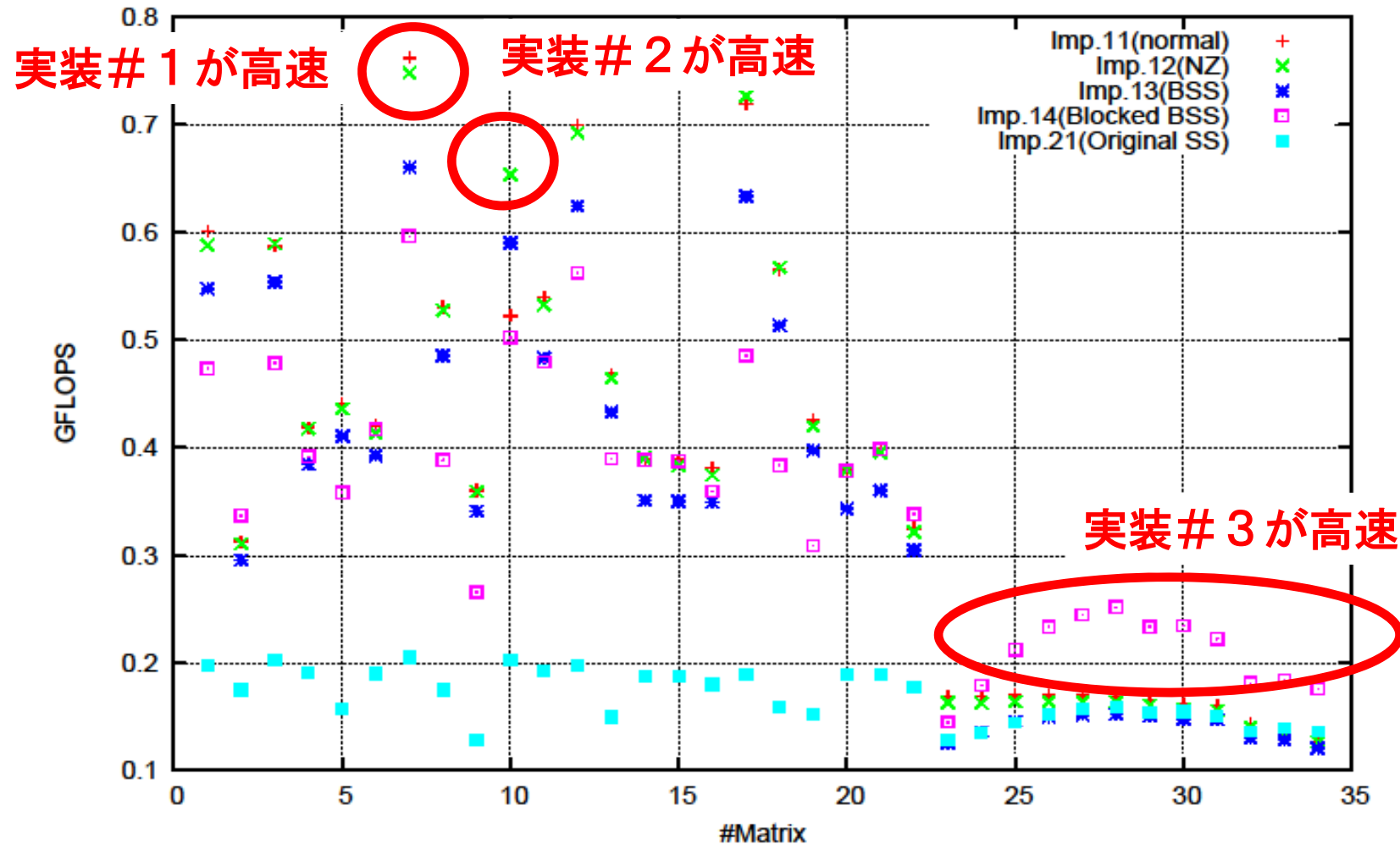
1. ノード内、ノード間で性能安定化する方法の実現
    - **現在でも性能は不安定（性能ヘテロ）**
      - 階層化キャッシュ
      - メモリエラーチェックによる、特定ノードの予測できない性能低下
  2. ジョブが割り当てられた物理コア番号、および物理ノード番号を知る方法の実装
    - ハードウェア、OSカーネルレベルでの実現
    - ユーザプログラムへのAPIの標準化
- 実行時におこなう
- 動的負荷分散
  - 実装方式の選択
- を実現する **自動チューニング機能の実装が必須**



# ● 数値アルゴリズムの課題

# 入力データ依存(疎行列-ベクトル積)

- HITACHI SRI6K/VLI(核融合研 設置)
- 128コア(SMT)/ノード
- フロリダ行列21種+独自行列 (非対称)
- 1コア実行





# アルゴリズムパラメタ自動調整効果 (Ver.Alpha)

Matrices	Fixed Restart Frequency			Auto-tuning			Speedup with AT
	M	#Restart	time(sec)	M	#Restart	time(sec)	
chem_master1	30	22	2.55	52	14	2.25	1.13
torso2	30	1	0.68	7	2	0.31	2.19
torso1	30	1	2.54	2	1	0.72	3.53
torso3	30	12	33.57	32	14	34.11	0.98
memplus	30	5	0.25	22	10	0.20	1.25
ex19	30	1000	収束せず	100	60	26.23	収束
poisson3Da	30	3	0.48	17	7	0.54	0.89
airfoil_2d	30	7	0.73	22	14	0.83	0.88
poisson3Db	30	7	10.38	17	14	11.03	0.94
viscoplastic2	30	19	2.93	37	15	1.70	1.72
xenon1	30	30	20.16	62	19	16.18	1.25
xenon2	30	40	92.96	72	20	64.29	1.45
wang4	30	5	0.37	17	9	0.29	1.28
ecl32	30	1000	収束せず	92	22	11.61	収束
sme3Da	30	670	215.49	100	90	101.33	2.13
sme3Db	30	1000	収束せず	100	120	377.29	収束
sme3Dc	30	1000	収束せず	100	122	575.63	収束
epb1	30	11	0.38	32	14	0.35	1.09
epb2	30	3	0.22	12	9	0.21	1.05
epb3	30	11	3.22	42	14	3.02	1.07

## 次世代システムへの要求（２）


- 多くの場合、密行列は何とかなる
  - 疎行列で、以下の実行時最適化機能が必須
    - 疎行列形状をもとに、以下を最適化
      1. 演算アルゴリズム
      2. 実装方式
      3. 通信方式
    - 疎行列の数値特性をもとに、以下を最適化
      1. 数値アルゴリズム上のパラメタ  
(リスタート周期、前処理方式、収束判定条件)
      2. 数値解法自体の自動選択
- いずれも、従来の自動チューニング研究で対象にしてきた内容



● その他の考慮すべき課題

# その他の課題

- **多倍長計算**
  - 問題：大規模化すると、反復回数が増える
  - 部分的多倍長化による**解法の安定化**
    - 例：反復解法における内積計算部分
- **精度保証計算**
  - 問題：大規模化すると、精度が劣化する
  - 数百万次元のHPLの精度 $\rightarrow O(1)$
  - 問題の数値特性、丸め誤差を考慮し、**解の理論的な存在範囲を知る手法**の必要性が増す
- **フォールトトレラントを考慮した、新しい数値アルゴリズムの開発**
  - 問題：数万ノードでは、実行中にノードが必ず壊れる
  - 複数パラメタ同時実行による冗長化
  - ノード故障により、解ベクトルの一部が欠損しても、**計算の続行が可能なアルゴリズム**（できれば、数理的に復元可能なアルゴリズム）
    - 定常反復解法？



○ 将来の計算機システム  
と数値計算ライブラリ

# 次次世代マシン(2015頃)

- アクセレータ (GPGPU) とマルチコアの混合機
- マルチコア
  - 周波数は現状維持 (~5GHz)
  - コア数は微増 (~64コア)
  - コア当たりメモリ量は現状維持 (数GB)
  - **Byte per Flops → 0.2台**
- この時代の数値計算ライブラリ
  - 密行列はアクセレータで高効率実行が可能
    - アクセレータ用のBLASなどが整備される
  - 疎行列の性能はピーク性能比 **数%台**  
→ 疎行列の性能向上が課題となる
    - 自動チューニング技術が進展し、実用化される
  - 超大規模・超並列実行により、演算精度の問題が表面化
    - 部分的多倍長化、精度保証技術が進展
  - 何らかのフォールトトレラント機能が数値計算ライブラリレベルで実装される

# エクサスケールマシン(2018~頃)

- 複数アクセラータ、低電力化複数マルチコアの混合機
- マルチコアの数：128超
- アクセラータに現在のGPUは無くなっている
  - 以下の複数の汎用処理用に分割し搭載
    - 分子動力学用 (GRAPE)
    - 密行列用(アレイ・プロセッサ、BLAS3、線形分解(LU, QR)など)
    - 疎行列用 (ランダム間接参照)
    - **線形ソルバ用 (疎行列直接解法用、固有値問題用)**
- マルチコアは「超低電力化」指向に
  - 周波数は大幅低下 (数百MHz~数GHz)
  - コア数は増大 (128コア超)
  - コア当たりのメモリ量 (数MB)
  - **Byte per Flops →0.0x台**
  - マルチコアはByte per Flopsが高いものと低いものの複数に分割し、複数のマルチコアが実装されているかも
- **新しい機能：**  
FPGA搭載で、コンパイル時にユーザの個別処理に特化した回路を自動生成する機構

# エクサ時代の数値計算ライブラリ

- **マルチコア上で、密行列(BLAS3)ですら高効率実行ができない**
- **以下の2つに分離**
  1. **キャッシュに収まる「小規模・超並列」実行**  
→超低電力マルチコアで実行
  2. **アクセレータを用いた「大規模」実行**  
→アクセレータ群で実行
  - 以上の切り替えを自動化する、自動チューニング技術が実用化
- **大規模計算時の精度保証技術が確立**
  - 数値計算結果は、演算結果、誤差の範囲（上限、下限）の3つが同時計算される
  - 部分的多倍長化も自動化
    - コンパイラレベル、数値計算ライブラリレベル
- **フォールトトレラント機能を実装**
  - 主要な数値計算ライブラリレベル