

ppOpen-ATにおけるOpenFOAM® 自動最適化への取り組み

伊東 聰

東京大学 情報基盤センター
スーパーコンピューティング研究部門
sito@cc.u-tokyo.ac.jp

内容

- 背景 : ppOpen-HPCと自動チューニング機構
- OpenFOAMの行列ベクトル積チューニング
 - アンローリング
 - 行列格納形式変更
- 準構造格子用AT機構の提案
 - snappyHexMeshと準構造格子
 - SSG-AT
- まとめ

背景

- JST 戦略的創造研究推進事業 CREST
領域名「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」
研究課題名「自動チューニング機構を有するアプリケーション開発・実行環境」プロジェクト
 - 本プロジェクトの特徴の一つに自動チューニング機構 (ppOpen-AT) の導入がある
 - ユーザの多い、普及しているソフトウェアでの効果実証により、ppOpen-ATの有用性を示す

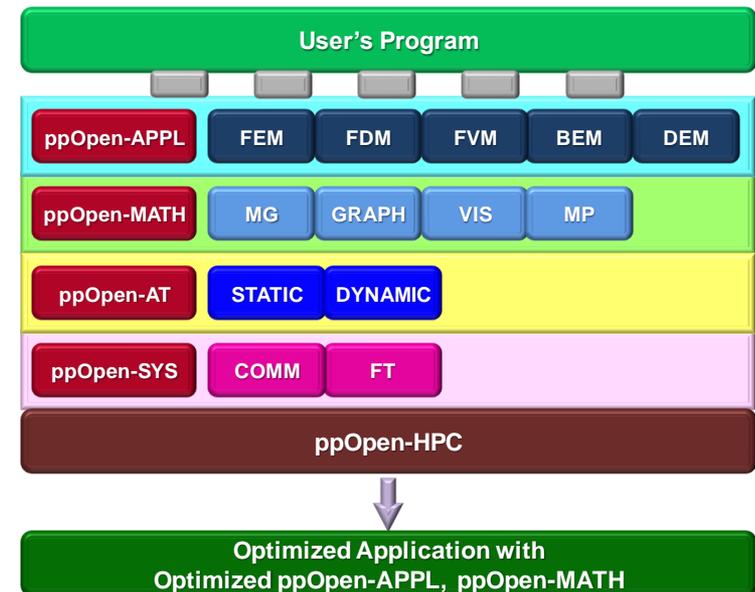


Fig.1 ppOpen-HPC whole image

OpenFOAMのAT機構開発

- ターゲットアプリ
 - icoFoam: シンプル, 工学上重要な非圧縮流れ
 - ホットスポット: 線形代数ソルバ(疎行列ベクトル積)
- 開発方針
 - 手動チューニングによる最適化法の模索
 - 準構造格子AT(SSG-AT)
 - 予備実験結果

OpenFOAM の行列ベクトル積(1/4)

- 赤色部分が行列ベクトル積(SpMV)
 - 行列格納形式: COO
 - 対角, 上三角, 下三角に分解して格納
 - SpMVの際に, 間接参照を2回(上下で計4回)必要

```
register const label nCells = diag().size();
for (register label cell=0; cell<nCells; cell++)
{
    ApsiPtr[cell] = diagPtr[cell]*psiPtr[cell];
}

register const label nFaces = upper().size();

for (register label face=0; face<nFaces; face++)
{
    ApsiPtr[uPtr[face]] += lowerPtr[face]*psiPtr[lPtr[face]];
    ApsiPtr[lPtr[face]] += upperPtr[face]*psiPtr[uPtr[face]];
}

// Update interface interfaces
updateMatrixInterfaces
(
    interfaceBouCoeffs,
    interfaces,
    psi,
    Apsi,
    cmpt
);
```

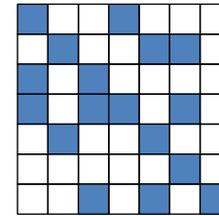
Fig.2 Partial source of lduMatrixATmul

行列ベクトル積のチューニング(2/4)

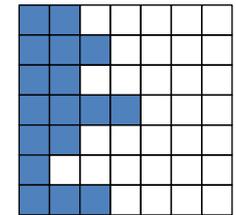
- チューニング方法
 - アンローリング
 - 行列格納形式変更
 - CRS(LDU分離型)
 - CRS2(LDU一体型)
 - FDS(5重対角専用)
- チュートリアルデータ利用
 - キャビティー流れ
 - (疑似)2次元
 - $100^2 \sim 500^2$
- テスト環境
 - T2K
 - AMD Opteron 8356
2.3GHz (Barcelona)
 - L1 : 64KB
 - L2 : 512KB/core
 - L3 : 2MB/processor
 - IntelCompiler+OpenMPI

行列ベクトル積のチューニング(3/4)

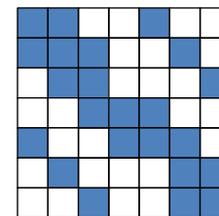
- COO
 - COOrdinate storage format
 - 非ゼロ成分の行・列番号を記憶
- CRS
 - Compressed Row Storage
 - 行方向に圧縮後, 1次元配列に
 - 各行の先頭までの個数も保持
- FDS
 - Fivefold Diagonal Storage
 - 5本の1次元配列に分解・記憶



COO



CRS



FDS

Fig.5 Various storage methods of Sparse M

行列ベクトル積のチューニング(4/4)

- 各手法での性能評価

- アンローリング

- ほとんど効果なし

- CRS

- LDU分離型は性能劣化
#最内ループ長が1~2
 - LDU一体型はCOOと同程度
#分離型と同じ理由

- FDS

- 10~20倍の高速化
 - 境界条件部分にゼロpaddingを含んでいる
 - n重対角にしか使えない

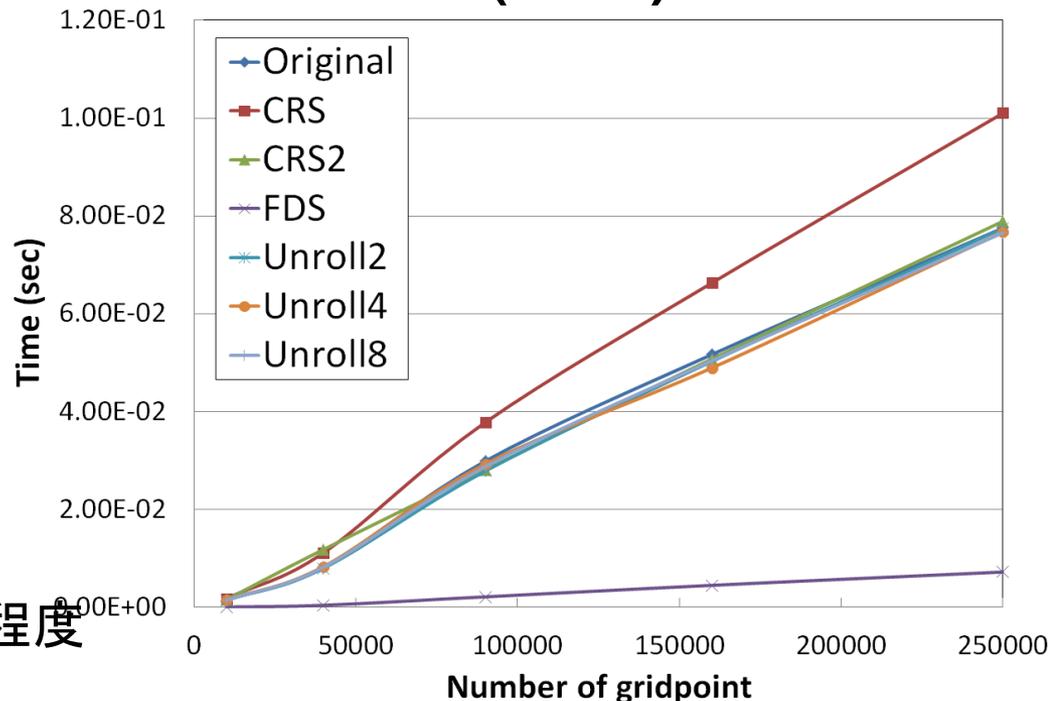


Fig.3 The results of each tuning method

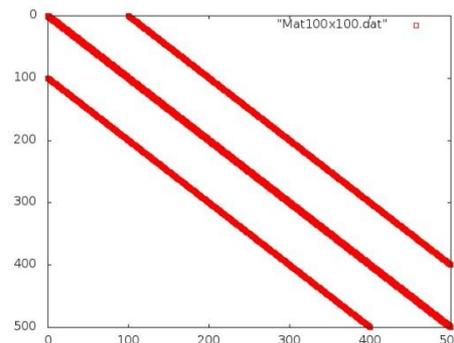


Fig.4 The structure of non-zero elements in a matrix (Fivefold diagonal)

内容

- 背景 : ppOpen-HPCと自動チューニング機構
- OpenFOAMの行列ベクトル積チューニング
 - アンローリング
 - 行列格納形式変更
- **準構造格子用AT機構の提案**
 - snappyHexMeshと準構造格子
 - SSG-AT
- まとめ

準構造格子用AT機構(1/5)

- OpenFOAMで使われる計算格子
 - CFD native
 - 流体用メッシュジェネレータで作成されたもの
 - 構造格子
 - OpenFOAM utility
 - blockMesh: 構造格子
 - snappyHexMesh
 - STL+Bounding box
 - 多面体セルが入る: 非構造格子
 - 多面体セルはSTL近傍のみ

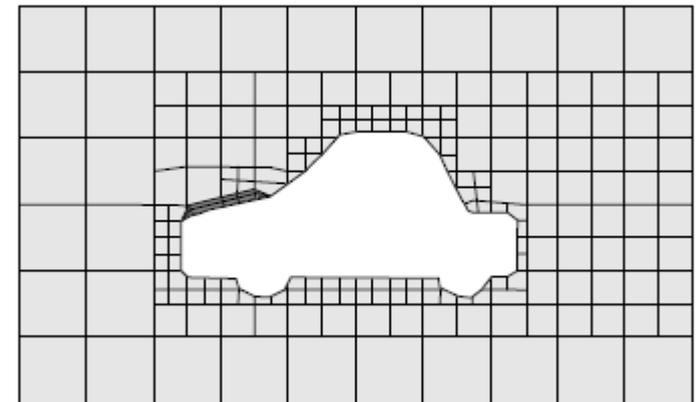


Fig.5 A image of mesh generated by snappyHexMesh

準構造格子として最適化できないか？

→ FDS/CRSハイブリッドフォーマット

準構造格子用AT機構(2/5)

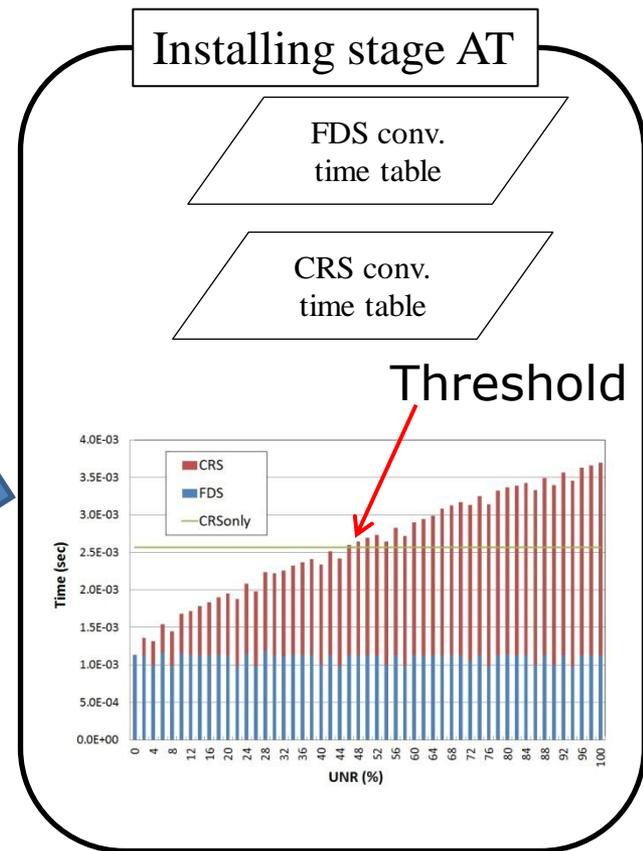
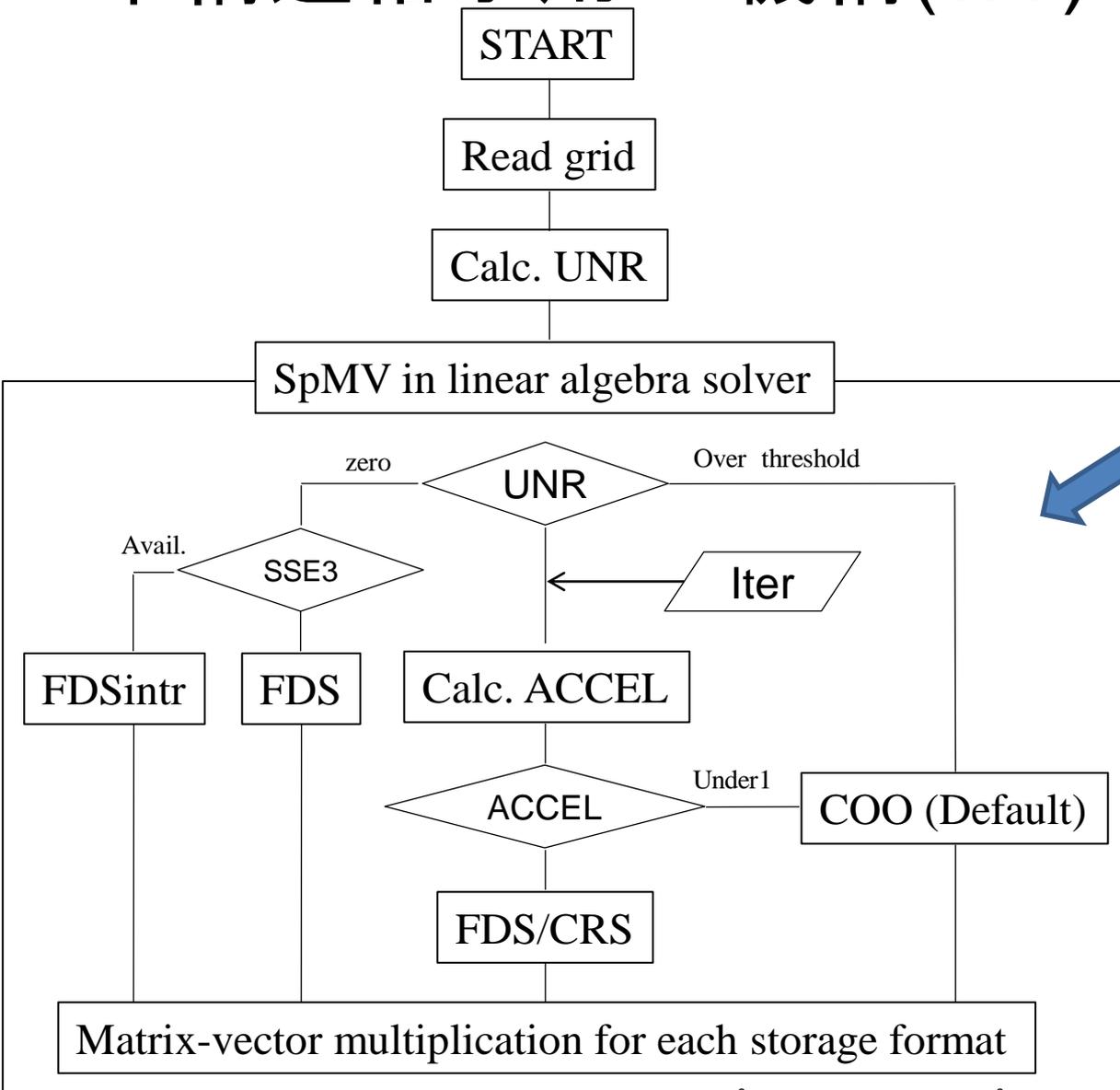
- 非構造格子の割合で格納形式を切り替える
 - UNR:UNstructured Ratio
 - N_{all} : 行列内の全非ゼロ要素数
 - N_{str} : 行列内のn重対角非ゼロ要素数

$$UNR = 1 - \frac{N_{str}}{N_{all}}$$

- ACCEL:Expected value of ACCELeration
 - ハイブリッドフォーマットを使った時のスピードアップの期待値
 - ITER:反復回数(予測値)
 - T_{**} :計算時間 or フォーマット変換時間

$$ACCEL = T_{COO} * I_{ter} / (T_{HB} * I_{ter} + T_{FDS} + T_{CRS})$$

準構造格子用AT機構(3/5)



インストール時に取得しておくサンプルデータ

準構造格子用AT機構(4/5)

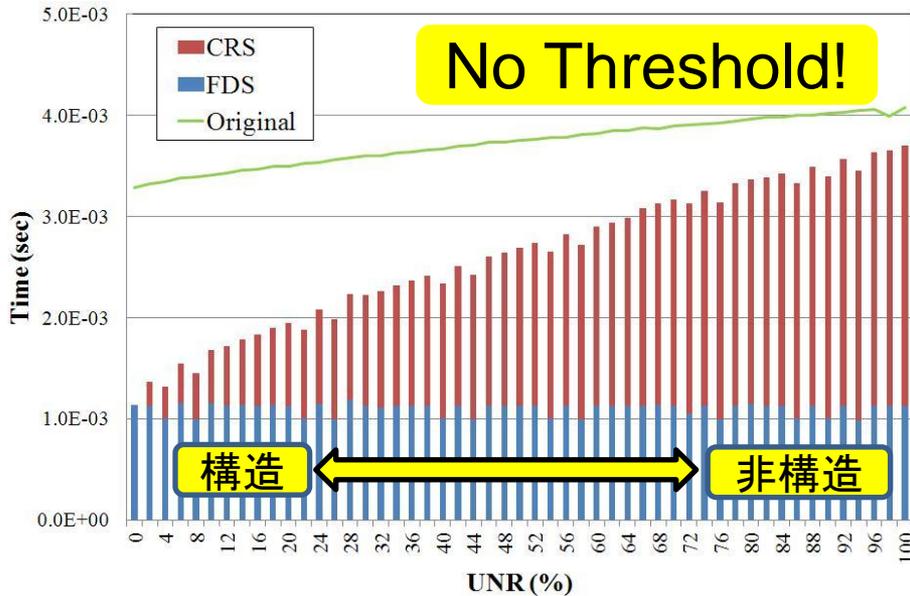


Fig.6 SpMV time for each UNR on T2K

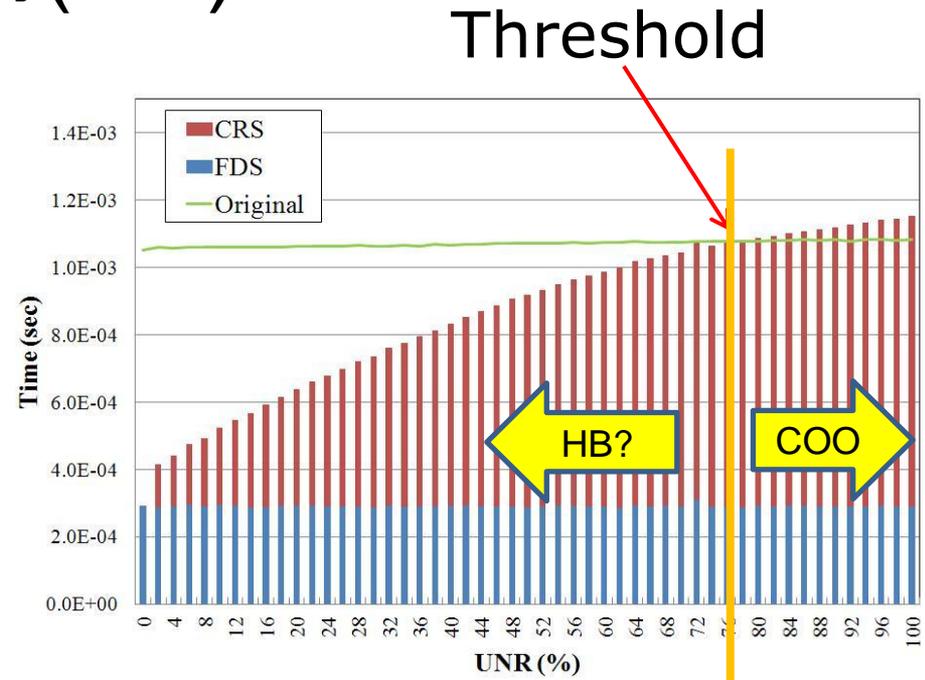


Fig.7 SpMV time for each UNR on C1

- AMD Opteron 8356 2.3GHz (T2K)
 - L2 : 512KB/core
 - L3 : 2MB/processor
 - IntelCompiler+OpenMPI
- Xeon X5550 2.67HGz (C1)
 - L2 : 256KB/core
 - L3 : 8MB
 - IntelCompiler+OpenMPI

格子点数: 90,000(300²)

準構造格子用AT機構(5/5)

- スピードアップの期待値
 - 格子点数: 90,000(300²)
 - 反復回数の予測値: 100
 - Thresholdの無かったT2KでもCOOの方が良い場合がある
 - 反復回数と変換時間の兼ね合い

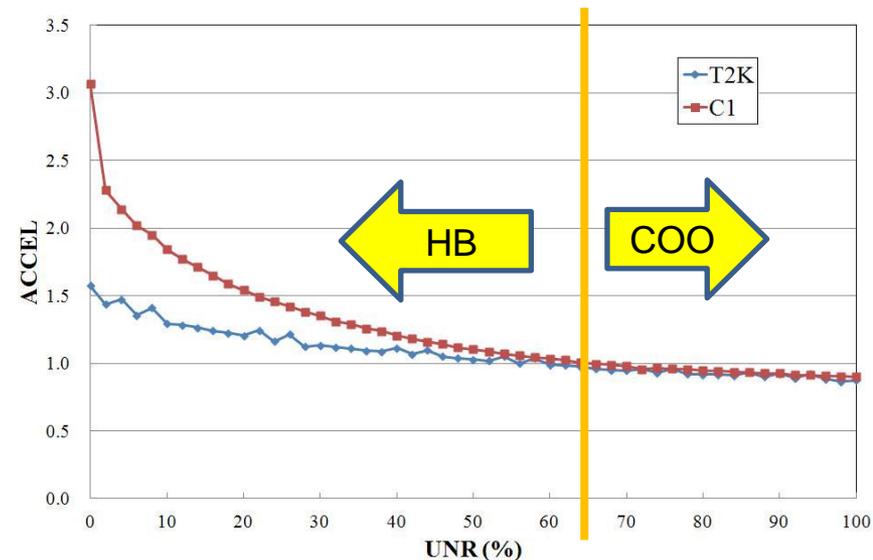


Fig.8 Expected acceleration for *Iter*=100

まとめ

- OpenFOAM(icoFOAM)に対する既存チューニング手法の適用とその結果
 - アンローリングはあまり効果が高くなく, FDSなどの専用格納形式を用いた場合に圧倒的に速くなる
- 準構造格子用AT機構の提案
 - UNR, Thresholdなどを用いた格納形式切り替え機構
 - インストール時にサンプルデータから基本性能取得
- 予備試験結果
 - 計算環境によってはThresholdがない(常にハイブリッドが速い)ような場合もある
 - その場合でも, 条件次第で最適なフォーマットが変わる