

# 富士通 PRIMEHPC FX10 チューニング連載講座

## 6. 数値計算ライブラリの利用

片桐 孝洋

東京大学情報基盤センター 准教授

### 1. はじめに

本稿では、富士通 PRIMEHPC FX10（以降、FX10 と記載）で利用できる数値計算ライブラリについて紹介します。本稿では特に、FX10 で利用できる数値計算ライブラリの紹介と、密行列用の数値計算ライブラリ BLAS(Basic Linear Algebra Subprograms)の FX10 での性能の紹介を目的に執筆いたします。

#### 1.1 FX10 で提供されている数値計算ライブラリの概要

FX10 では、富士通社製の数値計算ライブラリのほかに、フリーソフトウェアもいくつか提供しております。

富士通社製の数値計算ライブラリとして、以下のライブラリを提供しています。

- SSL II (Scientific Subroutine Library II)
- C-SSL II
- SSL II/MPI

富士通社製の数値計算ライブラリ以外のライブラリとしては、以下のライブラリを提供しています。

- BLAS、LAPACK、ScaLAPACK
- FFTW
- SuperLU、SuperLU\_DIST
- PETSc
- METIS
- Parallel NetCDF
- OpenFOAM

METIS については、並列版の ParMETIS が利用可能です。

2013 年 10 月現在、表 1 バージョンのソフトウェアが利用可能です。

表 1 FX10 で利用できる数値計算ライブラリとそのバージョン

ライブラリ名	バージョン
BLAS	-
LAPACK	3.2.2 <sup>1</sup>

<sup>1</sup> Netlib で公開されている LAPACK の最新版は、Version 3.4.2 になります。

ScaLAPACK	1. 8
FFTW	3. 3
boost	1. 47. 0
METIS	5. 0. 2
ParMETIS	4. 0. 2
PETSc	3. 3-p4
SuperLU	4. 3
SuperLU_DIST	3. 1
Parallel NetCDF	1. 3. 1
REVOCAP Refiner	1. 0. 0
NetCDF (C 言語版)	4. 2. 1. 1
NetCDF (Fortran、C++ 版)	4. 2
OpenFOAM	2. 2. 1

現在インストールされている最新のバージョン情報については、ログインノードにおいて、以下の module コマンドを入力することで確認できます。各自、ご確認ください。

なお、インタラクティブノードにインストールされている数値計算ライブラリについても、同様に確認できます。

```
$module avail
```

## 1. 2 OpenFOAM について

OpenFOAM は、流体解析を行うためのオープンソースの CAE (Computer Aided Engineering) ツールとして、最近注目を集めています。OpenFOAM の性能については、富士通社が C++ コンパイラの最適化能力の向上などを行い、FX10 で提供されている OpenFOAM の性能改善を行っています。東京大学情報基盤センターの FX10 においては、性能改善された OpenFOAM ライブラリが利用できます。OpenFOAM の FX10 での性能や最適化情報については、[1]を参照ください。

## 1. 3 東京大学情報基盤センター開発の数値計算ライブラリについて

東京大学情報基盤センターが開発した数値計算ライブラリがあります。これらは、いずれもフリーソフトウェアとしてソースが公開されています。FX10 での動作が確認されており、ユーザは自らライブラリを FX10 上にインストールをすることができます。以下にその数値計算ライブラリを記載します。

- **自動チューニング機能付き疎行列反復解法ライブラリ Xabclib**

( <http://www.abc-lib.org/Xabclib/index-j.html> )

自動チューニング機能付き数値計算ライブラリパッケージです。Xabclib は自動チューニング機能に関し、すでに開発されている ABCLib に対して、以下の拡張を実装したものです。

汎用化された自動チューニングインターフェースライブラリ OpenATLib を設計・実装します。ユーザが望む自動チューニング方針である<AT ポリシー>が記述できる、機能・方式・API (Application Programming Interface) が実装されています。

## ● 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 ppOpen-HPC

( <http://ppopenhpc.cc.u-tokyo.ac.jp/wordpress/?lang=en> )

ポストペタスケールシステムの処理能力を十分に引き出す科学技術アプリケーションの効率的な開発、安定な実行に資する数値計算ライブラリ（数値ミドルウェア）です。対象離散化手法は、有限要素法、差分法、有限体積法、境界要素法、個別要素法です。各手法の特性に基づき、ハードウェアに依存しない共通インタフェースを有するアプリケーション開発用ライブラリ群を提供します。

Xabclib および ppOpen-HPC は、今後、センターからライブラリの提供がされる予定です。また、センターによる利用講習会の実施を予定しております。詳しくは、東京大学情報基盤センターの講習会の HP<sup>2</sup>をご覧ください。

## 2. BLAS の性能

### 2. 1 概要

BLAS とは、Basic Linear Algebra Subprograms の略で、基本線形代数副プログラム集のことです。線形代数計算で用いられる、基本演算を標準化（API 化）したものです。

BLAS は、**密行列用**の線形代数計算用の基本演算の副プログラムを指します。疎行列用の BLAS は、**スパース BLAS** というものがあります。しかし通常 BLAS というとき、密行列の演算ルーチンであることを注意してください。

BLAS では以下のように分類し、サブルーチンの命名規則を統一しています。

1. 演算対象のベクトルや行列の型（整数型、実数型、複素型）
2. 行列形状（対称行列、三重対角行列）
3. データ格納形式（帯行列を二次元に圧縮）
4. 演算結果が何か（行列、ベクトル）

また、演算性能から、以下の 3 つに演算を分類しています。

1. **レベル 1 BLAS** : ベクトルとベクトルの演算
2. **レベル 2 BLAS** : 行列とベクトルの演算
3. **レベル 3 BLAS** : 行列と行列の演算

### 2. 2 各 BLAS の性能

**レベル 1 BLAS** の特性について解説します。

- ▶ ベクトル内積、ベクトル定数倍の加算、などです。
  - ▶ 例： $y \leftarrow \alpha x + y$
- ▶ データの読み出し回数、演算回数がほぼ同じです。
- ▶ データの再利用（キャッシュに乗ったデータの再利用によるデータアクセス時間の短縮）がほとんどできません。実装による性能向上が、あまり期待できません。ほとんど、計算機ハードウェアの演算性能となります。
- ▶ レベル 1 BLAS のみで演算を実装すると、演算が本来持っているデータ再利用性がなくなることがあります。
  - ▶ 例：行列-ベクトル積を、レベル 1 BLAS で実装するなど。

<sup>2</sup> <http://www.cc.u-tokyo.ac.jp/support/kosyu/>

レベル2 BLAS の特性について解説します。

- ▶ 行列-ベクトル積などの演算です。
  - ▶ 例:  $y \leftarrow \alpha A x + \beta y$
- ▶ 前進/後退代入演算、 $T x = y$  ( $T$ は三角行列) を  $x$  について解く演算、を含みます。
- ▶ レベル1 BLAS のみの実装による、データ再利用性の喪失を回避する目的で提案されました。
- ▶ 行列とベクトルデータに対して、データの再利用性があります。データアクセス時間を、実装法により短縮可能です。実装法により性能向上がレベル1 BLAS に比べしやすいですが、十分ではありません。

レベル3 BLAS の特性について解説します。

- ▶ 行列-行列積などの演算です。
  - ▶ 例:  $C \leftarrow \alpha A B + \beta C$
- ▶ レベル2 BLAS では、性能向上が十分に達成できませんでした。並列化により1CPU当たりのデータ量が減少するためです。より大規模な演算をとり扱わないと、再利用の効果が出ません。
- ▶ 行列-行列積では、行列データが  $O(n^2)$  に対し、演算は  $O(n^3)$  なので、データ再利用性が原理的に高いです。行列積は、アルゴリズムレベルでもブロック化でき、さらにデータの局所性を高めることができるので、最適化の効果が高いです。

各 BLAS の性能挙動は、図1のようになるのが普通です。

図1より、可能であればレベル3 BLAS3 演算を用いる演算にするほうが、理論ピーク性能に対する実効効率の観点から、高い性能を保つことができます。

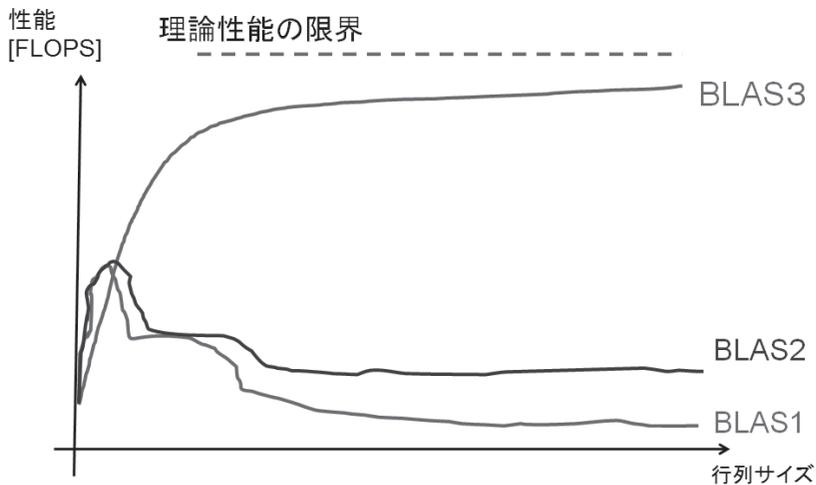


図1 各 BLAS の性能挙動について

### 2. 3 FX10 における BLAS ライブラリの利用について

東京大学情報基盤センターで提供している BLAS (Basic Linear Algebra Subprograms)、LAPACK (Linear Algebra PACKage)、および ScaLAPACK (Scalable LAPACK) は、FX10 (SPARC64 IXfx) 向けにチューニングしたライブラリを提供しています。詳細については、利用手引書を

ご覧ください。

東大の FX10 で、BLAS、LAPACK、および ScaLAPACK を利用する場合、富士通コンパイラを利用し、以下のコンパイルオプションを指定してライブラリをリンクしてください。なお、富士通社製 C/C++ コンパイラにて数学ライブラリを使用する場合には、数学ライブラリの製品マニュアルに記載されている注意事項も合わせてご覧ください。

表 2 BLAS、LAPACK、ScaLAPACK の利用方法（富士通コンパイラを利用のこと）

ライブラリ名		コンパイラ オプション	備考
BLAS	逐次版	-SSL2	-
	スレッド並列版	-SSL2BLAMP	-
LAPACK	逐次版	-SSL2	-
	スレッド並列版	-SSL2BLAMP	-
ScaLAPACK	MPI 並列	-SCALAPACK	利用する BLAS/LAPACK に応じて（つまり、逐次版かスレッド並列版か）、-SSL2 または -SSL2BLAMP を指定する

## 2. 4 FX10 における BLAS ライブラリの性能

ここでは、FX10 の 1 ノード（16 コアまで）を用いる場合における、レベル 3BLAS の dgemm 関数（倍精度演算、行列 - 行列積演算）の性能を示します。

関数 dgemm の呼び出しは、行列 - 行列積演算を行う場合には、実行環境に依存せず、以下のような実装になります。

### ■ レベル 3 BLAS の呼び出し例 (Fortran 言語)

```
double precision ALPHA, BETA
ALPHA=1.0d0
BETA=1.0d0
CALL DGEMM('N', 'N', n, n, n, ALPHA, A, n, B, n, BETA, C, n)
```

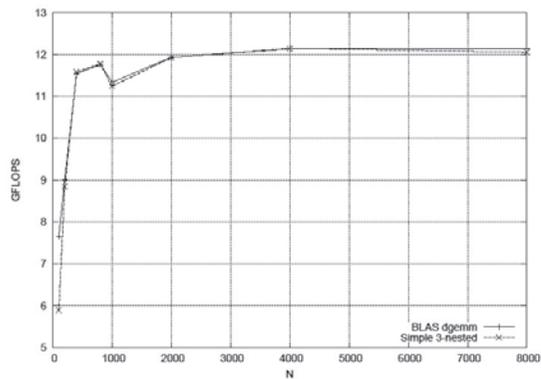
一方、BLAS を用いず、自ら行列 - 行列積演算を実装する場合は、OpenMP を用いてスレッド並列実行することが可能です。その場合の実装例を、以下に示します。

■行列 - 行列積演算を実装する場合(手書き実装) (Fortran 言語)

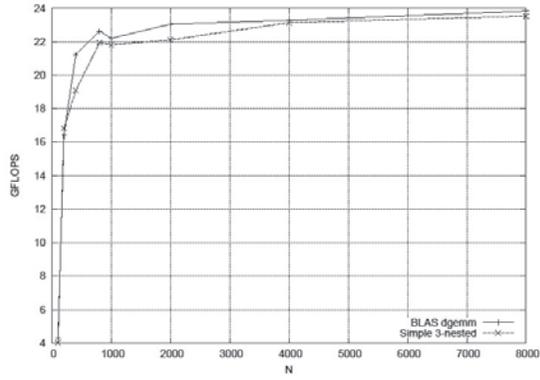
```
!$omp parallel do private(i, j, k)
  do i=1, n
    do j=1, n
      do k=1, n
        C(i, j) = C(i, j) + A(i, k) * B(k, j)
      enddo
    enddo
  enddo
!$omp end parallel do
```

以上のレベル 3 BLAS の性能と、手書きした 3 重ループによる行列 - 行列積の性能の[GfLOPS]値を、問題サイズ  $N=100$  から  $N=1000$  までは 200 刻みで、 $N=1000$  から  $N=8000$  までは、 $N=2000$ 、 $N=4000$ 、 $N=8000$  について測定したものを図 2 に示します。

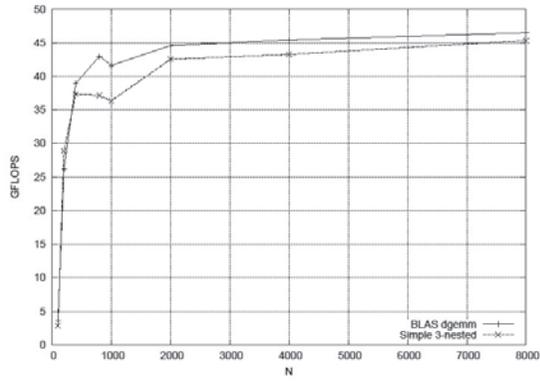
なお対象ルーチンは、 $N=1000$  までは 100 回連続で呼び出し、1 回当たりの平均時間から GfLOPS 値を算出しています。 $N=2000$  以上は 4 回連続で呼び出し、1 回当たりの平均時間から GfLOPS 値を算出しています。したがって、配列データをキャッシュに載せた上での性能計測 (**ホット・キャッシュ性能**) である点に注意してください。



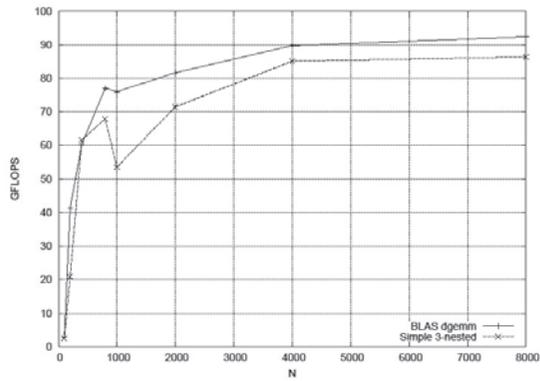
(a) 1 スレッドの性能



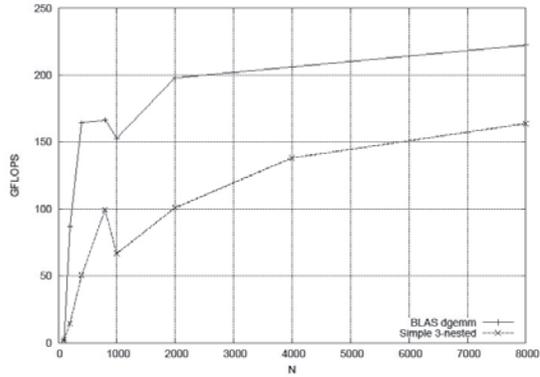
(b) 2 スレッド性能



(c) 4 スレッド性能



(d) 8 スレッド性能



(e) 16 スレッド性能

図 2 FX10 の BLAS ライブラリの性能

図 2 より、スレッド数が 1 スレッド、2 スレッドの場合は、BLAS を利用しても、手書きの 3 重ループコードに対してほぼ同等の性能となります。

しかしながら、4 スレッド以上になると、BLAS を利用することによる高速化が望めます。特に、16 スレッドでの BLAS 利用の効果が大きいです。具体的には、図 2(d) の 16 スレッド性能で、N=8000 の時、BLAS 利用で 222GFLOPS、手書き実装で 163GFLOPS となりますので、BLAS 利用により手書き実装の性能に対して、36%の速度向上になります。

次に、対ピーク性能を算出します。FX10 の 1 コアの理論性能は 14.78 GFLOPS、1 ノード (16 コア) の理論性能は 236.5 GFLOPS です。表 3 に、N=8000 の時の、スレッド並列版 BLAS 利用の時の理論性能に対する実行効率をまとめます。

表 3 スレッド並列版 BLAS 利用における実行効率 (N=8000)

	1 スレッド	2 スレッド	4 スレッド	8 スレッド	16 スレッド
<b>GFLOPS</b>	12.13	23.82	46.50	92.39	222.2
<b>理論性能</b>	14.78	29.56	59.12	118.24	236.5
<b>効率</b>	82.0%	80.5%	78.6%	78.1%	93.9%

表 3 より、16 スレッド実行の効率が 93.9%と非常に高性能です。スレッド並列版の富士通社製 BLAS は、16 スレッド実行で特に最適化された実装になっていると思われます。

ところで、1 スレッド実行では、逐次 BLAS を利用する方が、スレッド並列版 BLAS の 1 スレッド実行よりも、高性能である場合があります。そこで、表 4 に逐次 BLAS の性能を載せます。

図 3 逐次版 BLAS 利用における実行効率

問題サイズ	N=100	N=200	N=1000	N=4000	N=8000
<b>GFLOPS</b>	8.20	8.97	11.25	12.01	12.02
<b>効率</b>	55.4%	60.6%	76.1%	81.2%	81.3%

図3より、逐次版 BLAS とスレッド並列版 BLAS との性能差は、あまりないことが分かります。

### 3. おわりに

本稿では、FX10 における、数値計算ライブラリの紹介、BLAS の利用法、および、BLAS の性能を紹介しました。

FX10 では BLAS のほかに、LAPACK、ScaLAPACK などの密行列用ライブラリが提供されています。また BLAS 以外についても、いくつかのフリーソフトウェアがインストールしており、ユーザに提供されています。各自のプログラムの利用状況を判断した上で、数値計算ライブラリの利用の検討をしていただければ幸いです。

### 参 考 文 献

- [1] OpenFOAM ワークショップ— OpenFOAM を京で使おう —、2013 年 9 月 27 日（金）  
[https://www.hpci-office.jp/pages/ws\\_openfoam\\_130927](https://www.hpci-office.jp/pages/ws_openfoam_130927)