

ABCLib Working Notes No.10

# *AutoTuned-RB*

*Version 1.00*

## 利用マニュアル

2005年1月17日

電気通信大学大学院情報システム学研究科

木下 靖夫

## AutoTuned-RB のライブラリインタフェース

AutoTuned-RB は、密行列積用のインストール時チューニングソフトウェアです。  
使用できる関数は以下の RB\_DGEMM です。

RB\_DGEMM ( TransA, TransB, M, N, K, a, A, lda, B, ldb, b, C, ldc )

各引き数の意味は以下のとおりです。L3BLAS ライブラリの DGEMM を同様なフォーマットです。

$$(C = a \times \text{Trans}(A) \times \text{Trans}(B) + b \times C) \quad (1)$$

**TransA:** 行列 A が、ニュートラルなら “ n ”、転置なら “ t ” (char 型)

**TransB:** 行列 B が、ニュートラルなら “ n ”、転置なら “ t ” (char 型)

**M:** 行列 A、C の行の次元数(int 型)

**N:** 行列 B、C の列の次元数(int 型)

**K:** 行列 A の列の次元数、行列 B の行の次元数(int 型)

**a:** 上の式(1)における定数 a(double 型)

**A:** A 行列の要素の配列ポインタ(double 型)

**lda:** 行列 A の第一次元要素数(int 型)

**B:** B 行列の要素の配列ポインタ(double 型)

**ldb:** 行列 B の第一次元要素数(int 型)

**b:** 上の式(1)における定数 b(double 型)

**C:** C 行列の要素の配列ポインタ(double 型)

**ldc:** 行列 C の第一次元要素数(int 型)

RB\_DGEMM は void 型です。

## 実行オプション

実行オプションの指定はありません。

## はじめに

本ライブラリは以下のような特徴をもっています。

インストール時最適化の枠組みで、密行列積をおこなう際の性能安定性に関するパラメタを自動設定します。具体的には密行列を再帰分割し、再帰段数の自動設定を行っています。計算コアでは、BLAS ライブラリをコールしています。特徴として、1CPU マシンにおいては複数のサンプリング点、SMP マシンにおいては以下の3つのサンプリング点を使用しています。

- (1) L1 キャッシュより大きい最小のサンプリングデータサイズ
- (2) L3 キャッシュより小さく最大のサンプリングデータサイズ
- (3) L3 キャッシュより大きく最小のサンプリングデータサイズ

この3点から、SMP マシン向けの自動チューニングを実現しています。

SMP マシンは、以下のチューニング効果が期待されます。

- (1) 並列化による性能向上
- (2) 行列サイズに依存する性能不安定性の改善

1CPU マシンには、(1)のチューニング効果が期待されます。

上記で述べた自動チューニング方式を、**AutoTuned-RB (Automatically Tuned Recursive BLAS)** と呼びます。

AutoTuned-RB では、本来全ての BLAS に関してチューニングする事が可能ですが、トライアル版（試用版）として、ATLAS BLAS のみ対象となっています。

使用環境は以下の通りです。

- (1) 1CPU、SMP マシン
- (2) BLAS が使用できる OS
- (3) C コンパイラ
- (4) Posix Thread

L3 キャッシュを搭載しているマシン向けに最適化をかけていますが、L1、L2 キャッシュ搭載マシン向けの最適化も行っています。

なお本試用版は、多くのユーザに利用してもらうことで、ユーザビリティの改善、バグの発見と修復、および新規開発事項への反映、を目的にリリースされるものです。

したがってマニュアルにおいての説明不備、低いコードのリーダビリティ、および突然の仕様変更、などの問題が生じる可能性があることをご理解ください。

また何かお気づきの点がありましたら、お気軽に著作者までご一報くださいませうお願い申し上げます。

## 自動チューニング項目

ここでは、本ライブラリで実装されているチューニング方式について説明します。

AutoTuned-RB のチューニング方式は以下の方式を取っています。

- インストール時チューニング  
ライブラリをインストールする際にチューニングを実行

SMP マシン、1CPU マシン共にチューニングパラメタは再帰段数です。

## インストール手順

手順は、以下の3つです。

- (1) Makefile の設定
- (2) make config の実行
- (3) Config file 作成後 make install

```
#####
```

### Makefile の設定例

```
#####
```

```
#
```

```
# AutoTuned-RB Install Makefile
```

```
#
```

```
#
```

```
#
```

```
#
```

```
#####
```

```
SHELL = /bin/sh
```

```
ARCH = Linux_P4SSE2
```

アーキテクチャ名(例:ATLAS にインストールした際に作成されたライブラリが入っているフォルダ名)

#####

TOPdir = /home/kinoshita  
ATLdir =\$(TOPdir)/ATLAS  
ATLlibdir = \$(ATLdir)/lib/\$(ARCH)

#####

(AutoTune-RB をインストール先の指定)  
ATRBdir = \$(TOPdir)/ATRB  
ATRBobjdir = \$(ATRBdir)/source/\$(ARCH)  
ATRBsrcdir = \$(ATRBdir)/source  
ATRBincdir = \$(ATRBdir)/include/\$(ARCH)  
ATRBlibdir = \$(ATRBdir)/lib/\$(ARCH)  
#ATRBbindir = \$(TOPdir)/bin/\$(ARCH)

#####

CC = cc  
CCFLAGS = -O3  
NM = -o  
OJ = -c

#####

ARCHIVER = ar  
ARFLAGS = r

#####

LATL = -latlas  
LPTH = -lpthread  
#Math = -lm

```
#####
```

```
all:install
```

```
config:
```

```
rm -f cofig
$(CC) -lm config.c $(NM) config
./config
mkdir -p $(ATRBincdir)
mv Config.h $(ATRBincdir)/
rm -f config
```

```
install:$(ATRBsrcdir)/ABCLib_BLAS_Src.c $(ATRBsrcdir)/ABCLib_BLAS_Rec
ursive.c
```

```
rm -f $(ATRBlibdir)/*.a
$(CC) $(ATRBsrcdir)/ABCLib_BLAS_Src.c $(NM) ¥
ABCLib_BLAS_Src $(CCFLAGS) -I$(ATRBincdir) -L$(ATLibdir) ¥
$(LATL) $(LPTH)
./ABCLib_BLAS_Src
mv Mtdev.h $(ATRBincdir)/
rm -f ABCLib_BLAS_Src
$(CC) $(OJ) -I$(ATRBincdir) $(ATRBsrcdir)/ABCLib_BLAS_Recursive.c
mkdir -p $(ATRBobjdir)
mv $(ATRBdir)/ABCLib_BLAS_Recursive.o $(ATRBobjdir)/
mkdir -p $(ATRBlibdir)
$(ARCHIVER) $(ARFLAGS) $(ATRBlibdir)/libatr.a $(ATRBobjdir)/¥
ABCLib_BLAS_Recursive.o
```

```
.PHONY : cleanall cleanbin cleanobj cleanlib cleanhead
```

```
cleanall: cleanbin cleanobj cleanlib cleanhead
```

```
cleanbin:
```

```
rm -f $(ATRBdir)/ABCLib_BLAS_Src
rm -f $(ATRBdir)/config
```

```
cleanobj:
```



```
rm -f $(ATRBobjdir)/*.o
rmdir $(ATRBobjdir)
```

cleanlib:

```
rm -f $(ATRBlibdir)/*.a
rmdir $(ATRBlibdir)
```

cleanhead:

```
rm -f $(ATRBincdir)/*.h
rmdir $(ATRBincdir)
```

```
#####
```

Makefile 設定後に、コマンド `make config` を実行します。ここで入力するデータは、

- (1) SMP サポート
- (2) CPU 台数
- (3) L3 キャッシュサポート
- (4) L1 キャッシュサイズ
- (5) L3 キャッシュサイズ(L2 キャッシュサイズ)
- (6) メインメモリサイズ

の6つです。これらを入力すると `Configure file` が作成されます。

`Configure file` 作成後にコマンド `make install` を実行すると `AutoTuned-RB` のチューニングが実行されます。インストール時間は計算機環境に依存しますが、経験的には SMP マシンで 10 分程度、1CPU マシンでは 2 時間程度です。

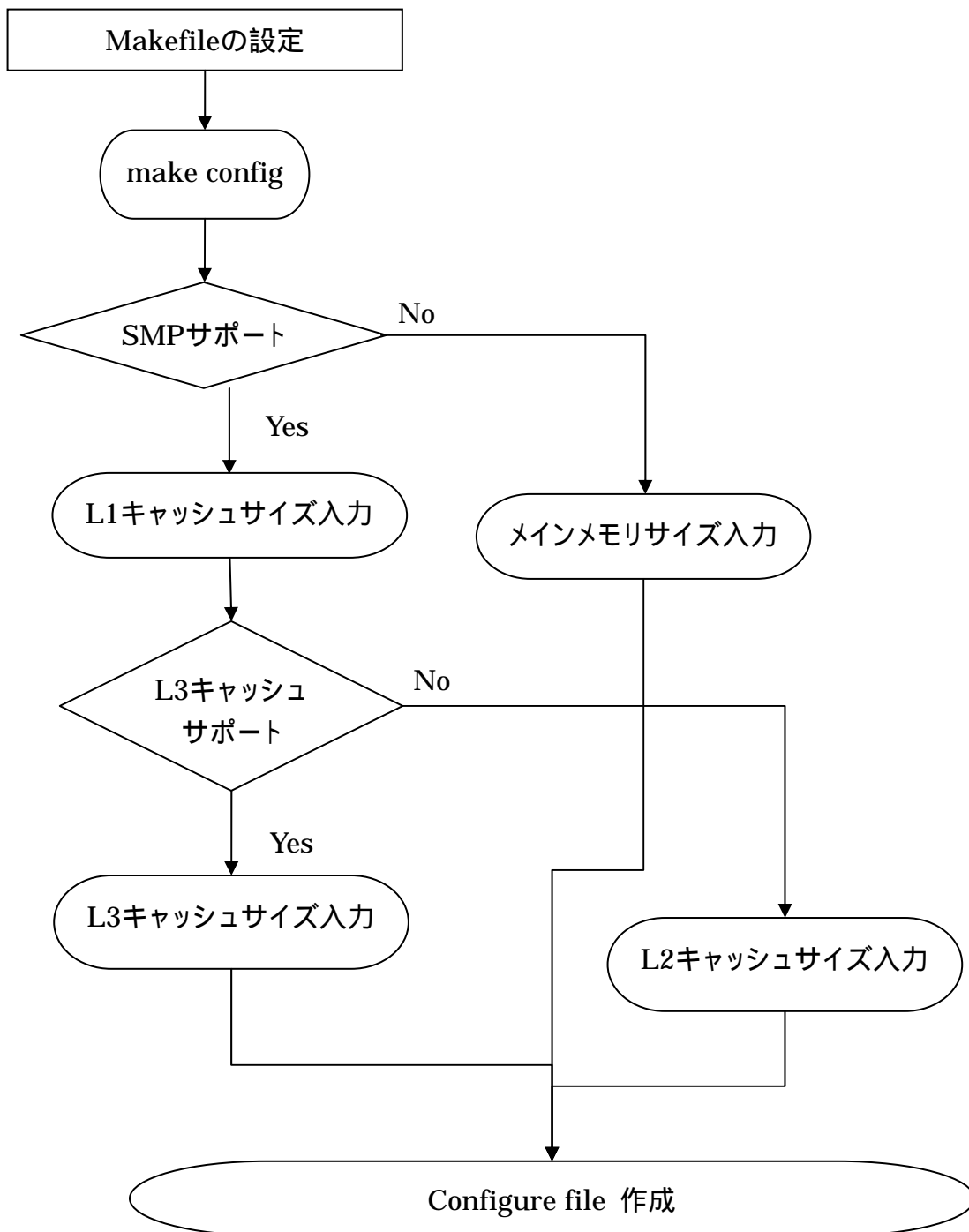


図 Configure file作成フロー

## ディレクトリ構造

- ATRB/  
Makefile  
Config.c
- ATRB/source  
ABCLib\_BLAS\_Recursive.c  
ABCLib\_BLAS\_Src.c
- ATRB/source/<arch>  
ABCLib\_BLAS\_Recursive.o  
Recursive\_p.o
- ATRB/include/<arch>  
Config.h  
Mtdev.h
- ATRB/lib  
Libatr.a
- ATRB/test  
Makefile  
ABCLib\_BLAS\_Test.c

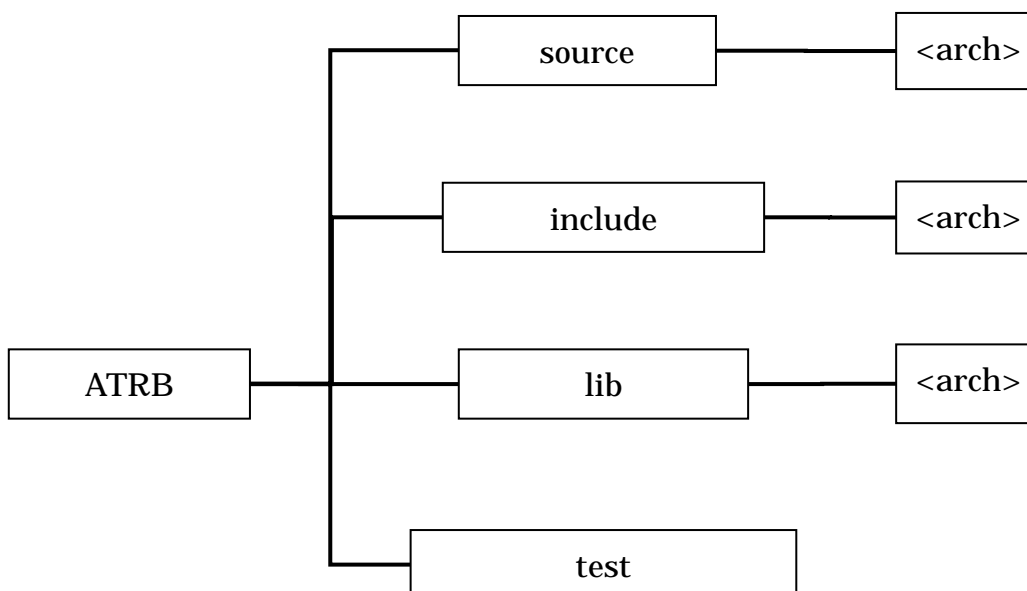


図 AutoTuned-RB ディレクトリ構造

## サンプルプログラム

密行列積を求めるためのサンプルプログラム( `test/ABCLibTestBLAS.c` )は、以下のとおりです。MATRIXSIZE を変更することで計算する行列の次元を変更することができます。

```

/*****/
/*                                     */
/*      Posix Pthread version         */
/*      Recursive BLAS Matrix Mutmal  */
/*                                     */
/*      Yasuo Kinoshita               */
/*                                     */
/*****/

#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <unistd.h>
#include <time.h>

#define MATRIXSIZE 4000

int main(int argc、 char **argv)
{
    int M, N, K;
    double *C, *A, *B;
    char TransA='n', TransB='n';
    double a=1.0, b=0.0;
    int lda, ldb, ldc;
    int i, j, x;
    struct timeval t1, t2;
    double soltime, sec, usec;
    time_t seed;
    time(&seed);

```

```

srand(seed);
M=MATRIXSIZE;
N=M;
K=M;
lda = K;
ldb = N;
ldc = M;

printf("MatrixC:%d*%d ", M, N);
printf("MatrixA:%d*%d ", M, K);
printf("MatrixB:%d*%d\n", K, N);

C = (double *)malloc(sizeof(double)*(M*N));
A = (double *)malloc(sizeof(double)*(M*K));
B = (double *)malloc(sizeof(double)*(K*N));

x=10;

for(i=0 ;i<M ;i++ ){
    for(j=0 ;j<N ;j++ ){
        *(C+j+i*N) = 0.0;
    }
}

for(i=0 ;i<M ;i++ ){
    for(j=0 ;j<K ;j++ ){
        *(A+j+i*K) =(double)(rand()%x);
    }
}

for(i=0 ;i<K ;i++ ){
    for(j=0 ;j<N ;j++ ){
        *(B+j+i*N) =(double)(rand()%x);
    }
}

```

```
gettimeofday(&t1, NULL);
RB_DGEMM(TransA, TransB, M, N, K, a, A, lda, B, ldb, b, C, ldc);
gettimeofday(&t2, NULL);
```

```
sec = t2.tv_sec - t1.tv_sec;
usec = t2.tv_usec - t1.tv_usec;
soltime = (sec + usec/1000000.0);
printf("Solve time = %0.3lf\n", soltime);
printf("   flops   = %0.3lf\n", 2*M*M*(M/soltime)/1000000);
```

```
free(C);
free(A);
free(B);
```

```
return 0;
```

```
}
```

```
#####
```

## 実行例

以下に、AutoTuned-RB のインストール実行例をのせます。  
なお以下の出力例は、バージョン変更等により形式が変わることがありますので、ご注意ください。  
この例でのインストールは、2 CPU の SMP マシンで行いました。

```
#####
```

```
kinoshita@opt01:~/ATRB> make config  
rm -f cofig  
cc -lm config.c -o config  
./config
```

```
#####
```

```
          ABCLib-BLAS  
        version   ver.1.0  
    composed by Yasuo Kinoshita
```

```
Graduate School of Information Systems,  
The University of Electro-Communications  
  /JAPAN SCIENCE AND TECHNOLOGY AGENCY  
      2004/01/16
```

```
AutoTuned-RB Configure
```

```
#####
```

```
===== make Config.h =====
```

```
SMP SUPPORT[y/n]: y
```

```
Input Number of CPU : 2
```

```
===== Sampling point Setting =====
```

Input L1Cache Size[KByte]: 64

L3Cache Machine?[y/n]: n

Input L2CacheSize[KByte]: 1024

Configuration Completed!!

Type "make install" if you continue install

```
mkdir -p /home/kinoshita/ATRB/include/Linux_UNKNOWNNSSE2_2
mv Config.h /home/kinoshita/ATRB/include/Linux_UNKNOWNNSSE2_2/
rm -f config
```

```
kinoshita@opt01:~/ATRB> make install
```

```
rm -f /home/kinoshita/ATRB/lib/Linux_UNKNOWNNSSE2_2/*.a
```

```
cc /home/kinoshita/ATRB/source/ABCLib_BLAS_Src.c -o ¥
```

```
ABCLib_BLAS_Src -O3 -I/home/kinoshita/ATRB/include/Linux_UNKNOWNNSSE2_2
-L/home/kinoshita/AutoTuned-RB/ATLAS/lib/Linux_UNKNOWNNSSE2_2 -latlas
-lpthread
```

```
./ABCLib_BLAS_Src
```

```
#####
```

```
ABCLib-BLAS
```

```
version ver1.0
```

```
composed by Yasuo Kinoshita
```

```
Graduate School of Information Systems,
```

```
The University of Electro-Communications
```

```
/JAPAN SCIENCE AND TECHNOLOGY AGENCY
```

```
2005/01/16
```

```
AutoTuned-RB Install-time Optimization
```

```
#####
```

```
##### SMP Machine Tuning #####
```



##### Near L1 Cache size #####

MATRIX SIZE 123 \* 123

RNUM	TIME	MFLOPS
------	------	--------

=====

0	0.002	1542.6
1	0.002	1602.1
2	0.003	1487.1
3	0.003	1400.2

##### In L3 Cache size #####

MATRIX SIZE 395 \* 395

RNUM	TIME	MFLOPS
------	------	--------

=====

0	0.065	1906.9
1	0.039	3197.5
2	0.041	3021.7
3	0.043	2899.4
4	0.050	2448.9
5	0.065	1901.1

Matrixsize	OptiNum
------------	---------

=====

123	1
395	1

Tuning Completed !

```
mv Mtdev.h /home/kinoshita/ATRB/include/Linux_UNKNOWNSSSE2_2/
```

```
rm -f ABCLib_BLAS_Src
```

```
cc -c -I/home/kinoshita/ATRB/include/Linux_UNKNOWNSSSE2_2
```

```
/home/kinoshita/ATRB/source/ABCLib_BLAS_Recursive.c
```

```
mkdir -p /home/kinoshita/ATRB/source/Linux_UNKNOWNSSSE2_2
```

```
mv /home/kinoshita/ATRB/ABCLib_BLAS_Recursive.o
```

```
/home/kinoshita/ATRB/source/Linux_UNKNOWNSSSE2_2/
```

```
mkdir -p /home/kinoshita/ATRB/lib/Linux_UNKNOWNNSSE2_2
ar      r      /home/kinoshita/ATRB/lib/Linux_UNKNOWNNSSE2_2/libatr.a
/home/kinoshita/ATRB/source/Linux_UNKNOWNNSSE2_2/ABCLib_BLAS_Recursive.
o
kinoshita@opt01:~/ATRB>
```

```
#####
```

## サンプルプログラム( test/ABCLibTestBLAS.c )

### 行列 1000 × 1000 の実行例

```
kinoshita@opt01:~/ATRB/test> make dgemm
cc      ABCLib_BLAS_Test.c      -o      ABCLib_BLAS_Test      -O3
-L/home/kinoshita/AutoTuned-RB/ATLAS/lib/Linux_UNKNOWNNSSE2_2 ¥
-L/home/kinoshita/ATRB/lib/Linux_UNKNOWNNSSE2_2 -latr -latlas -lpthread -lm
kinoshita@opt01:~/ATRB/test> ./ABCLib_BLAS_Test
MatrixC:1000*1000 MatrixA:1000*1000 MatrixB:1000*1000
s = 2
Solve time =  0.509
      Mflops   =  3926.897
kinoshita@opt01:~/ATRB/test>
```

注) test/makefile の内容を適宜変更して使用してください。

## おわりに

本マニュアルでは、BLAS3 用の再帰 BLAS における自動チューニングソフトウェア AutoTuned-RB の利用法について説明しました。

なお、以下のオンラインマニュアルでも、本マニュアルの内容を閲覧できます。

<http://www.abc-lib.org/online/abclib.htm>