

オープンソース大規模構造解析プログラム FrontISTR v4.6のインストール (CentOS 7.3)

ここでは、FrontISTR を CentOS 7.3 でビルドする方法を紹介します。Redhat Enterprise Linux へも同様にインストールすることが出来ます。

はじめに

FrontISTRは[東京大学 奥田研究室](#)で開発されている、本格的な構造解析プログラムです。[FrontISTR研究会](#)から、研究会の案内や資料、ソフトウェアが入手できます。FrontISTRはMITライセンスで配布され、自由にソースを改変することが出来ます。

また、これまでにスパコンからWindowsパソコンでの動作実績があるのも特徴の一つです。

準備

予め CentOS 7.3 をインストールし、OSが動作可能な状態にしてください。また、安全のため最新の状態にしておいた方が良いでしょう。

```
$ su
# yum update
```

ビルドに必要なツールとパッケージのインストール

最初にFrontISTRをビルドするのに必要なツールとパッケージをインストールします。

```
$ su
# yum group mark install "Development Tools"
# yum update
# yum install openmpi-devel cmake
# exit
```

ここで一般ユーザに戻り、openmpiの環境設定をします。

```
$ cd $HOME
$ module avail
----- /etc/modulefiles -----
mpi/openmpi-x86_64

$ cd $HOME
$ vi .bash_profile
```

.bash_profileの一番下に

```
module purge
module load mpi/openmpi-x86_64
```

を追加しファイルを保存して、一度ログアウトしてから再度ログインをして下さい。

ライブラリのビルドとインストール

FrontISTRのビルドに必要なライブラリをビルドしインストールします。必要なライブラリは

- OpenMPI (MPI)
- OpenBLAS (LAPACK)
- Metis-5
- MUMPS
- Scalapack
- Trilinos ML

です。この内、OpenMPIは前の手順でインストールしました。

ダウンロード

以下のソフトを入手してください。

ソフトウェア名	ダウンロード先	備考
OpenBLAS-0.2.20	http://www.openblas.net	リファレンスのLAPACKよりも速い
metis-5.1.0	http://glaros.dtc.umn.edu/gkhome/metis/metis/download	scotch-metisも利用可
scalapack-2.0.2	http://www.netlib.org/scalapack/	MUMPSで利用
MUMPS_5.1.1	http://mumps.enseeiht.fr/	要ユーザ登録、直接法ソルバ
trilinos-12.10.1	https://trilinos.org/download/	要ユーザ登録、反復法前処理に利用
REVOCAP_Refiner-1.1.04	http://www.multi.k.u-tokyo.ac.jp/FrontISTR	要ユーザ登録、FrontISTRリザーバの「最新版FrontISTRおよびREVOCAP_PrePost」よりダウンロード
		要ユーザ登録、FrontISTRリザーバの「最

ライブラリのビルド

作業をするディレクトリは `$HOME/Software`、ライブラリをインストールするディレクトリは `$HOME/.local` として説明をします。

```
$ cd $HOME
$ mkdir Software
$ mkdir .local
```

PATH 環境変数を確認してください。

```
$ echo $PATH
/usr/lib64/openmpi/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/home/user/.local/bin:/home/user/bin
```

REVOCAP_Refinerのビルド

REVOCAP_Refinerをビルドします。

```
$ cd $HOME/Software
$ tar xvf REVOCAP_Refiner-1.1.04.tar.gz
$ cd REVOCAP_Refiner-1.1.04
$ make
```

OpenBLASのビルド

OpenBLASは、高速なLAPACK/BLASのフリーの実装です。

OpenMPを有効にしたライブラリをビルドします。

```
$ cd $HOME/Software
$ tar xvf OpenBLAS-0.2.20.tar.gz
$ cd OpenBLAS-0.2.20
$ make BINARY=64 NO_SHARED=1 USE_OPENMP=1
$ make PREFIX=$HOME/.local install
```

インストール時に

```
install: `libopenblas_haswellp-r0.2.20.so' を stat できません : そのようなファイルやディレク  
トリーはありません  
make[1]: *** [install] エラー 1  
make[1]: ディレクトリ `/home/user/Software/OpenBLAS-0.2.20' から出ます  
make: *** [install] エラー 2
```

と表示されますが、今回は共有ライブラリを作成していないので無視してください。

METISのビルド

グラフパーティショナのライブラリです。

OpenMPを有効にしたライブラリをビルドします。

```
$ cd $HOME/Software  
$ tar xvf metis-5.1.0.tar.gz  
$ cd metis-5.1.0  
$ make config prefix=$HOME/.local openmp=1 cc=gcc  
$ make  
$ make install
```

Scalapackのビルド

scalapackは、この後説明をするMUMPSのビルドの際必要となります。

```
$ cd $HOME/Software  
$ tar xvf scalapack-2.0.2.tgz  
$ cd scalapack-2.0.2  
$ mkdir build  
$ cd build  
$ cmake -DCMAKE_INSTALL_PREFIX=$HOME/.local \  
        -DCMAKE_EXE_LINKER_FLAGS="-fopenmp" \  
        -DBLAS_LIBRARIES=$HOME/.local/lib/libopenblas.a \  
        -DLAPACK_LIBRARIES=$HOME/.local/lib/libopenblas.a \  
        ..  
$ make  
$ make install
```

MUMPSのビルド

MUMPSは直接法のソルバです。

```
$ cd $HOME/Software  
$ tar xvf MUMPS_5.1.1.tar.gz  
$ cd MUMPS_5.1.1
```

Make.inc/Makefile.inc.genericを元にMakefile.incを環境に合わせた内容へ書き換えます。

```
% cp Make.inc/Makefile.inc.generic Makefile.inc
% vi Makefile.inc
LMETISDIR = $(HOME)/.local
IMETIS    = -I$(LMETISDIR)/include
LMETIS    = -L$(LMETISDIR)/lib -lmetis

ORDERINGSF = -Dmetis -Dpord

CC        = mpicc -fopenmp
FC        = mpifort -fopenmp
FL        = mpifort -fopenmp

LAPACK =

SCALAP    = -L$(HOME)/.local/lib -lscalapack

INCPAR =

LIBPAR    = $(SCALAP)

LIBBLAS   = -L$(HOME)/.local/lib -lopenblas
```

ファイルの修正が済んだらライブラリをビルドします。

```
$ make
$ cp lib/*.a $HOME/.local/lib
$ cp include/*.h $HOME/.local/include
```

Trilinos MLのビルド

Trilinosには多くのライブラリ含まれていますが、FrontISTRに必要なのはMLとzoltanだけです。

```
$ cd $HOME/Software
$ tar xvf trilinos-12.10.1-Source.tar.gz
$ cd trilinos-12.10.1-Source
$ mkdir build
$ cd build
$ cmake \
  -DCMAKE_INSTALL_PREFIX=$HOME/.local \
  -DTrilinos_ENABLE_ML=ON \
  -DTrilinos_ENABLE_Zoltan=ON \
  -DTrilinos_ENABLE_OpenMP=ON \
```

```
-DTrilinos_ENABLE_ALL_OPTIONAL_PACKAGES=OFF \  
-DTPL_BLAS_LIBRARIES=$HOME/.local/lib/libopenblas.a \  
-DTPL_LAPACK_LIBRARIES=$HOME/.local/lib/libopenblas.a \  
-DTPL_ENABLE_MPI=ON \  
..  
$ make  
$ make install
```

FrontISTRのビルド

必要なライブラリの導入が済んだらFrontISTRをビルドします。

FrontISTRで有効にする機能は

- hecmw_part1領域分割ツールなどのツール群生成
- REVOCAP_Refiner (メッシュの細分化機能)
- MPI (PCクラスタ等の並列実行環境)
- OpenMP (マルチコアCPUを利用)
- METIS (メッシュの領域分割のために必要)
- MUMPS (直接法ソルバー)
- LAPACK (CG法やGMRES法を用いた前処理適用後の条件数推定機能を利用)
- Trilinos-ML (代数マルチグリッド法に基づく前処理)

です。これらの機能を有効にしたバイナリをビルドします。

```
$ cd $HOME/Software  
$ tar xvf FrontISTR_V46.tar.gz  
$ cd FrontISTR_4.6
```

Makefile.amを編集し、ビルドしない部分を除外します。

```
% vi Makefile.am  
PREFIX      = @prefix@  
BINDIR      = @bindir@  
LIBDIR      = @libdir@  
INCLUDEDIR  = @includedir@  
  
MAKE        = @make@  
AR          = @ar@  
CP          = @cp@  
RM          = @rm@  
MV          = @mv@  
RANLIB      = @ranlib@  
MKDIR       = @mkdir@  
  
all:  
    @cd hecmw1 && $(MAKE)
```

```

        @cd fistr1 && $(MAKE)
#       @cd hecmw2 && $(MAKE)
#       @cd fistr2 && $(MAKE)
        @cd fistr  && $(MAKE)

clean:
        @cd hecmw1 && $(MAKE) clean
        @cd fistr1 && $(MAKE) clean
#       @cd hecmw2 && $(MAKE) clean
#       @cd fistr2 && $(MAKE) clean
        @cd fistr  && $(MAKE) clean

distclean:
        @cd hecmw1 && $(MAKE) distclean
        @cd fistr1 && $(MAKE) distclean
#       @cd hecmw2 && $(MAKE) distclean
#       @cd fistr2 && $(MAKE) distclean
        @cd fistr  && $(MAKE) distclean
        - $(RM) ./bin/*

install:
        if [ ! -d $(PREFIX) ]; then \
            $(MKDIR) $(PREFIX); \
        fi
        if [ ! -d $(BINDIR) ]; then \
            $(MKDIR) $(BINDIR); \
        fi
        @cd hecmw1 && $(MAKE) install
        @cd fistr1 && $(MAKE) install
#       @cd hecmw2 && $(MAKE) install
#       @cd fistr2 && $(MAKE) install
        @cd fistr  && $(MAKE) install

```

hecmw2およびfistr2はビルドしませんので、コメントアウトしてください。

Makefile.confの編集

```

% cp Makefile.conf.org Makefile.conf
% vi Makefile.conf

```

以下の内容でMakefile.confを編集してください。

```

#####
#                                                                    #
#   Setup Configuration File for FrontISTR                          #
#                                                                    #
#####

```

```

# MPI
MPIDIR      = /usr/lib64/openmpi
MPIBINDIR   = $(MPIDIR)/bin
MPILIBDIR   = $(MPIDIR)/lib
MPIINCDIR   = $(MPIDIR)/include
MPILIBS     = -lmpi -lmpi_cxx -lmpi_mpi fh

# for install option only
PREFIX      = $(HOME)/FrontISTR
BINDIR      = $(PREFIX)/bin
LIBDIR      = $(PREFIX)/lib
INCLUDEDIR  = $(PREFIX)/include

# Metis
METISDIR    = $(HOME)/.local
METISLIBDIR = $(METISDIR)/lib
METISINCDIR = $(METISDIR)/include
HECMW_METIS_VER= 5

# ParMetis
PARMETISDIR = $(HOME)/ParMetis-3.1
PARMETISLIBDIR = $(PARMETISDIR)
PARMETISINCDIR = $(PARMETISDIR)/ParMETISLib

# Refiner
REFINERDIR  = $(HOME)/Software/REVOCAP_Refiner-1.1.04
REFINERINCDIR = $(REFINERDIR)/Refiner
REFINERLIBDIR = $(REFINERDIR)/lib/x86_64-linux

# Coupler
REVOCAPDIR  = $(HOME)/REVOCAP_Coupler
REVOCAPINCDIR = $(REVOCAPDIR)/librcap
REVOCAPLIBDIR = $(REVOCAPDIR)/librcap

# MUMPS
MUMPSDIR    = $(HOME)/.local
MUMPSINCDIR = $(MUMPSDIR)/include
MUMPSLIBDIR = $(MUMPSDIR)/lib
MUMPSLIBS   = -ldmumps -lmumps_common -lpord -L$(HOME)/.local/lib -
lscalapack

# MKL PARDISO
MKLDIR      = $(HOME)/
MKLINCDIR   = $(MKLDIR)/include
MKLLIBDIR   = $(MKLDIR)/lib

# ML
MLDIR       = $(HOME)/.local
MLINCDIR    = $(MLDIR)/include
MLLIBDIR    = $(MLDIR)/lib

# C compiler settings

```



```

CC          = mpicc -fopenmp
CFLAGS      =
LDFLAGS     = -lstdc++ -lm
OPTFLAGS    = -O3

# C++ compiler settings
CPP         = mpicxx -fopenmp
CPPFLAGS    =
CPPLDFLAGS  =
CPPOPTFLAGS = -O3

# Fortran compiler settings
F90         = mpif90 -fopenmp
F90FLAGS    =
F90LDFLAGS  = -lstdc++ -L$(HOME)/.local -lopenblas
F90OPTFLAGS = -O2
F90FPP      = -cpp
F90LINKER   = mpif90 -fopenmp

MAKE        = make
AR          = ar ruv
MV          = mv -f
CP          = cp -f
RM          = rm -f
MKDIR      = mkdir -p

```

この手順で用いるMETISはVersion5.1.0です。**HECMW_METIS_VER = 5**を指定してください。

gfortranのプリプロセッサ指定は**-cpp**です。**F90FPP = -cpp**を指定してください。

編集が出来たら**setup.sh**を実行してください。

```

$ ./setup.sh -p --with-tools --with-refiner --with-metis --with-mumps --with-
lapack --with-ml

```

setup.shの実行が済んだらビルドしてバイナリを作成します。

```

$ make
$ make install

```

ビルドしたバイナリをインストールをすると、**\$(HOME)/FrontISTR/bin**以下に

```

$ ls $HOME/FrontISTR/bin
fistr1  hec2rcap  hecmw_part1  hecmw_vis1  neu2fstr  rconv  rmerge

```

などのバイナリがコピーされ実行できる状態になります。

テスト

インストールされた FrontISTR (`fistr1`) を実行してみましょう。

```
$ export PATH=$HOME/FrontISTR/bin:$PATH
$ which fistr1
~/FrontISTR/bin/fistr1
$ cd $HOME/Software/FrontISTR_V46/tutorial
$ ls
01_elastic_hinge          07_viscoelastic_cylinder  13_dynamic_beam_nonlinear
02_elastic_hinge_parallel 08_creep_cylinder         14_dynamic_plate_contact
03_hyperelastic_cylinder  09_contact_hertz         15_eigen_spring
04_hyperelastic_spring    10_contact_2tubes        16_heat_block
05_plastic_cylinder       11_contact_2beam         17_freq_beam
06_plastic_can            12_dynamic_beam          18_cavity_flow
```

様々な例題が同梱されています。この中の `01_elastic_hinge` を実行してみましょう。

`01_elastic_hinge` ディレクトリに移動し、ディレクトリの中に格納されているファイルを見てみます。

```
$ cd 01_elastic_hinge
$ ls
hecmw_ctrl.dat  hinge.cnt  hinge.msh
```

この中には3つのファイルが格納されています。FrontISTRで解析する場合、この3つの入力ファイルが最低必要になります。

ファイル名	名前	説明
hecmw_ctrl.dat	全体制御ファイル	解析で使う入力ファイル名や、結果の出力ファイル名を指定します
hinge.cnt	解析制御データ	解析の種別や条件などを指定します
hinge.msh	メッシュファイル	解析対象のメッシュ情報や材料データを指定します

FrontISTR の実行名は `fistr1` です。引数は特にありません。3つのファイルが入ったディレクトリで起動してみてください。

```
$ fistr1
Step control not defined! Using default step=1
fstr_setup: OK
### 3x3 B-SSOR-CG(0) 1
   1  1.903375E+00
   2  1.974378E+00
   3  2.534627E+00
...
...
```

```
Start visualize PSF 1 at timestep 1
```

```
=====  
TOTAL TIME (sec) :    61.42  
  pre (sec) :    0.67  
  solve (sec) :    60.75  
=====  
FrontISTR Completed !!
```

しばらくすると解析が終了し、結果ファイルなどが出力されます。

```
$ ls  
0.log      FSTR.msg  hecmw_ctrl.dat  hinge.cnt  hinge.res.0.1  
FSTR.dbg.0 FSTR.sta  hecmw_vis.ini   hinge.msh  hinge_vis_psf.0001.inp
```

ファイルの内容については、マニュアルを参照してください。