

FrontISTR を ubuntu-13.04 へインストール

FrontISTR はソースからコンパイルしてインストールするが、必要となるライブラリ類は出来る限り ubuntu のバイナリパッケージを利用する。

1. ダウンロードが必要なソフトウェア

ubuntu には用意されていないソースパッケージを予めダウンロードしておく。

FrontISTR v42b

<http://www.multi.k.u-tokyo.ac.jp/FrontISTR/>

の「データリザバー」⇒「FrontISTR」から最新版をダウンロード(要ユーザ登録)

metis-4.0.3

<http://glaros.dtc.umn.edu/gkhome/fsroot/sw/metis/OLD>

ubuntu のバイナリパッケージはバージョンが古いためソースコードからビルド。

metis の最新バージョンは、現在の所 5.1.0 だが、FrontISTR は未だ最新版には対応していないので、ひとつ前のバージョン 4.0.3 をダウンロード

REVOCAP_Refiner-1.1.01

CISS(革新的シミュレーション研究センター)のページからダウンロード

2. 依存バイナリパッケージのインストール

バイナリパッケージで用意されている物は出来るだけ利用する。OS インストール直後に必要なバイナリパッケージは以下の通り。

```
% sudo apt-get install build-essential (gcc-4.7.3 等がインストールされる)
% sudo apt-get install gfortran (4.7.3 がインストールされる)
% sudo apt-get install ruby (1.9.3 がインストールされる)
% sudo apt-get install doxygen
% sudo apt-get install swig
% sudo apt-get install libopenmpi-dev (1.6 がインストールされる)
% sudo apt-get install openmpi-bin blcr-dkms
% sudo apt-get install libmumps-dev (4.10.0 がインストールされる)
(scalapack が同時にインストールされる)
```

```
% sudo apt-get install libboost-all-dev (1.49.0がインストールされる)
```

3. 依存ソースパッケージのインストール

REVOCAP Refiner のビルド

ダウンロードした REVOCAP Refiner をビルドする。

```
% tar xvf REVOCAP_Refiner-1.1.01.tgz  
% cd REVOCAP_Refiner-1.1.01  
% vi MakefileConfig.in
```

MakefileConfig.in の中でコンパイル時のオプションを編集するが、`gfortran` の場合そのままでも OK。

REVOCAP Refiner のソースファイル中に、一部記述が足りない部分があるので編集。

```
% vi RevocapIO/kmbHecmwIO_V3.cpp  
#include <cstdlib>  
を追加
```

ビルド

```
% make
```

metis のビルド

ダウンロードした metis をビルドする。

```
% tar xvf metis-4.0.3.tar.gz  
% cd metis-4.0.3  
% vi Makefile.in  
通常はこのままでも構わないが、好みに応じて編集  
% make
```

4. FrontISTR のビルド

FrontISTR 本体をビルドする

```
% tar xvf FrontISTR_V42b.tar.gz
% cd FrontISTR
```

ビルドのための設定をする

```
cp Makefile.conf.org Makefile.conf
vi Makefile.conf
```

これまで設定した環境の場合、Makefile.conf は以下の記述になる

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

```
# MPI
MPIDIR      = /usr
MPIBINDIR   = $(MPIDIR)/bin
MPILIBDIR   = $(MPIDIR)/lib
MPIINCDIR   = $(MPIDIR)/include
MPILIBS     = -lmpi -lmpi_cxx -lmpi_f77 -lmpi_f90
```

バイナリでインストールした
OpenMPI を指定する。

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

FrontISTR をインストールする
場所を指定する。

```
# Metis
METISDIR    = $(HOME)/Software/metis-4.0.3
METISLIBDIR = $(METISDIR)
METISINCDIR = $(METISDIR)/Lib
```

ソースからビルドした metis
を指定する

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

現在は使われていないので、特に指定
する必要は無い。

ソースからビルドした Refiner を指定
する。x86_64-linux の記述を忘れる事
があるので注意

```
# Refiner
REFINERDIR      = $(HOME)/Software/REVOCAP_Refiner-1.1.01
REFINERINCDIR  = $(REFINERDIR)/Refiner
REFINERLIBDIR  = $(REFINERDIR)/lib/x86_64-linux
```

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

今回は利用しないので、特に指定する必要は無い。

```
# MUMPS
MUMPSDIR       = /usr
MUMPSINCDIR    = $(MUMPSDIR)/include
MUMPSLIBDIR    = $(MUMPSDIR)/lib
```

バイナリでインストールした **mumps** を指定する。

```
# C compiler settings
CC             = mpicc
CFLAGS        =
LDFLAGS       = -lm
OPTFLAGS      = -Ofast -msse -msse2 -march=native -mtune=native -mfpmath=sse -f
bounds-check  -Wuninitialized
```

MPI 版をビルドするので **mpiXX** を指定する。

```
# C++ compiler settings
CPP           = mpic++
CPPFLAGS     =
CPPLDLAGS    =
CPPOPTFLAGS  = $(OPTFLAGS)
```

配列の境界チェックも念の為指定。
x86 の CPU なら **-msse** 等を指定すると浮動小数点演算が少し速くなる。

```
# Fortran compiler settings
F90          = mpif90
F90FLAGS     =
F90LDLAGS    =
F90OPTFLAGS  = -O1 -msse -msse2 -march=native -mtune=native -mfpmath=sse -fbou
nds-check -Wuninitialized -fbcktrace
```

配列の境界チェックも念の為指定。

```
MAKE         = make
AR           = ar ruv
CP           = cp -f
```

gfortran の場合、**-O2** 以上の最適化オプションを指定すると **Segmentation Fault** が発生する。

```
RM          = rm -f
MKDIR       = mkdir -p
```

Makefile.conf の編集が終わったら、以下のシェルスクリプトを実行して Makefile 等を生成する。

```
% ./setup.sh -g -p --with-tools --with-refiner --with-metis --with-mumps --with-paracon
```

シェルスクリプトを実行したらビルドをする。

```
% make
% make install
```

fistr1 がコピーされたディレクトリにパスを通すと使いやすい。

今回の場合、\$(HOME)/FrontISTR/bin なので、.profile 等に

```
export PATH=$(HOME)/FrontISTR/bin:$PATH
```

を記述する。

テスト実行

シリアル版

```
% cd FrontISTR/tutorial/01_elastic_hinge
% fistr1
```

コマンド実行後、正常終了すると

```
FrontISTR Completed !!
```

と表示される。

MPI 版

MPI 版は、最初にメッシュを分割する。例題は 4 分割。

```
% cd FrontISTR/tutorial/02_elastic_hinge_parallel
% hecmw_part1
```

以下が表示されると、分割されたメッシュデータが生成される。

```
Aug 01 13:48:33 Info: Reading mesh file...
Aug 01 13:48:33 Info: Starting domain decomposition...
Aug 01 13:48:34 Info: Creating local mesh for domain #0 ...
Aug 01 13:48:34 Info: Creating local mesh for domain #1 ...
Aug 01 13:48:34 Info: Creating local mesh for domain #2 ...
Aug 01 13:48:34 Info: Creating local mesh for domain #3 ...
Aug 01 13:48:34 Info: Domain decomposition done
```

生成したら、FrontISTR を実行する

```
% mpirun -np 4 fistr1
```

正常終了すると

FrontISTR Completed !!

が表示される。

海洋研究開発機構 地球シミュレータセンター 小川道夫