逐次解析と並列解析

2013年3月25日

第1回FrontISTR研究会

Makefile.confの詳細(1)

```
##################################################
#     Setup Configulation File for FrontISTR                    #
##################################################

# MPI
MPIDIR        =
MPIBINDIR     =
MPILIBDIR     =
MPIINCDIR     =
MPILIBS       =

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

# Metis
METISDIR      = $(HOME)/Metis-4.0
METISLIBDIR   = $(METISDIR)
METISINCDIR   = $(METISDIR)/Lib
```

MPI環境が組み込まれている場合、設定不要

/usr/local/binなどを設定してもよい

現在は使用していない

このままでよい(Metis-5.0は現時点で不可)

Makefile.confの詳細(2)

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

# Refiner
REFINERDIR       = $(HOME)/REVOCAP_Refiner-1.1.0
REFINERINCDIR    = $(REFINERDIR)/Refiner
REFINERLIBDIR    = $(REFINERDIR)/lib/x86_64-linux

# Coupler
REVOCAPDIR       = $(HOME)/REVOCAP_Coupler-1.6.2
REVOCAPINCDIR    = $(REVOCAPDIR)/librcap
REVOCAPLIBDIR    = $(REVOCAPDIR)/librcap

# MUMPS
MUMPSDIR         = $(HOME)/MUMPS_4.10.0
MUMPSINCDIR      = $(MUMPSDIR)/include
MUMPSLIBDIR      = $(MUMPSDIR)/lib
```

現在は使用していない

ダウンロードしたバージョン番号に修正

インストール時の設定に合わせる

ダウンロードしたバージョン番号に修正

このバージョン以前は不可

Makefile.confの詳細(3)

```
# C compiler settings
CC              = mpicc
CFLAGS          =
LDFLAGS         = -lm
OPTFLAGS        = -O3

# C++ compiler settings
CPP             = mpic++
CPPFLAGS        = -DMPICH_IGNORE_CXX_SEEK -I$(HOME)/include
CPPLDFLAGS      =
CPPOPTFLAGS     = -O3

# Fortran compiler settings
F90             = mpif90
F90FLAGS        =
F90LDFLAGS      = -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5
                  -lmkl_scalapack_lp64 -lmkl_blacs_openmpi_lp64
F90OPTFLAGS     = -O2

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

インストール計算機環境に合わせる

インストール計算機環境に合わせる

hecmw2をコンパイルするためのおまじない(このまま)

インストール計算機環境に合わせる

Intel MKLをリンクする

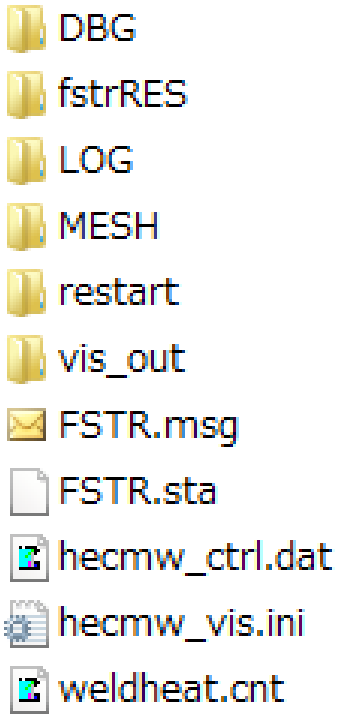
MUMPSで必要なBLAS・SCALAPACKをIntel MKLからリンクする

標準的なLinuxであればこのままでよい

hecmw2に必要なBoostライブラリがない場合
(1) Boostライブラリをインストールする
(2) includeファイルのみどこかから調達して指定する

hecmw_ctrl.datの!SUBDIR

- 並列解析時には、!SUBDIRを設定することを推奨
- 通常のLinuxでは、ディレクトリあたりのファイル数（サブディレクトリ数）が1万？を超えるとOSがハングアップ
- 「京」や東京大学のFX10は、6万5千でもトラブルなし
- !SUBDIRを設定すると、メッシュファイルや解析結果ファイルをサブディレクトリに自動的に格納
- さらにfstrRESはステップごとに分割
- ただし、ステップ数の管理はユーザー責任（リスタートをかけること）



解析結果データのバイナリー出力

- 解析結果（res）データのファイル出力はバイナリー形式を推奨

  !RESULT, NAME=fstrRES, IO=OUT, TYPE=BINARY

- アスキー形式に戻すには、rconvを使用

  mpirun –n 16 rconv –o newFistrModel.res

- REVOCAP_PrePostへ渡す統合ファイルは、rmergeで作成

  rmerge –o text newFistrModel.res

メッシュデータにおける留意事項

- 材質パラメータは、解析制御（cnt）データで指定することが基本
- ただし、以下の制約がある
  - 材料名は、必ず!MATERIALに記述、材質パラメータはダミーでよい
  - !SECTIONも必須
  - 熱伝導解析の材質パラメータだけは、メッシュデータに記述
- EGROUP、NGROUP（SGROUPは除く）には“ALL”がデフォルトで作成されている

  ＝＞　何もしないで、解析制御データで使用可能
- !EQUATIONによるアセンブリ構造解析は検証済み、ただし自分でEQUATIONデータを作成する必要あり

hecmw_part_ctrl.datのパラメータ

- 領域分割手法（METHOD）は3通りある
  - RCB
    GEOFMプロジェクトで開発された手法（高速）
    x,y,z方向の分割のため、簡単な形状に有効
  - PMETIS
    節点数が等しくなるように分割
    小規模並列で有効（配布元の推奨）
  - KMETIS
    エッジカット（＝通信量）最小の条件を加えて分割
    大規模並列で有効（配布元の推奨）
- 最近の計算機環境では、大規模もPMETISがよさそう（ただし大差はない）
- UCD指定で出力されるUCD形式ファイルはAVSのみで表示可能

hecmw_part1の並列実行

- 大規模メッシュを数百～数千分割すると、非常に時間がかかる
- マニュアルには記述していないが、hecmw_part1の並列実行が可能（任意の並列数で追加指定なし）
- ただし、通常のNFSファイルシステムではメッシュファイルの同時読み込みで異常に遅くなるので注意
  （10分で終了する5GBファイルの読み込みが、6プロセスでは数時間たっても終わらないという例あり）
- 「京」や東京大学のFX10のように、ランクごとディレクトリにステージインできればトラブルなし

!SOLVERのパラメータ

- 反復法ソルバーはCG法で特に問題ない
  （というより、ソルバー間の性能比較未実施が本音）
- 収束判定閾値は10^{-8}程度が適切、動解析では10^{-6}程度に落としても問題なさそう
- 前処理には、ILUと対角スケーリングがあり、どちらが速いかは解析問題による。収束までの反復回数は大きく異なるが、経過時間に劇的な差はない。
- Additive Schwarzの繰り返し回数＝1（推奨値は2）を試してみてもよい
  （＝0は前処理なしになる）

fistr1並列実行の際の留意事項

- 4GBメモリの1プロセスで実行可能な節点数は、百万弱
- マルチコアによるノード内並列処理性能はリニアではない
- 1ノードをベースとする複数ノード並列処理性能は、小規模であればほぼリニア
- 大規模の場合、プロセスあたりの処理節点数により並列処理性能は大きく変わる

以降、実測例を紹介

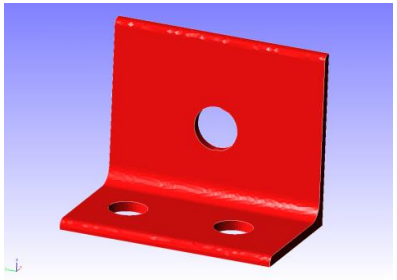
FrontISTRの並列処理性能（ノード内）

解析対象

静応力解析（四面体二次要素）

要素数：49，871

節点数：84，056



使用計算機

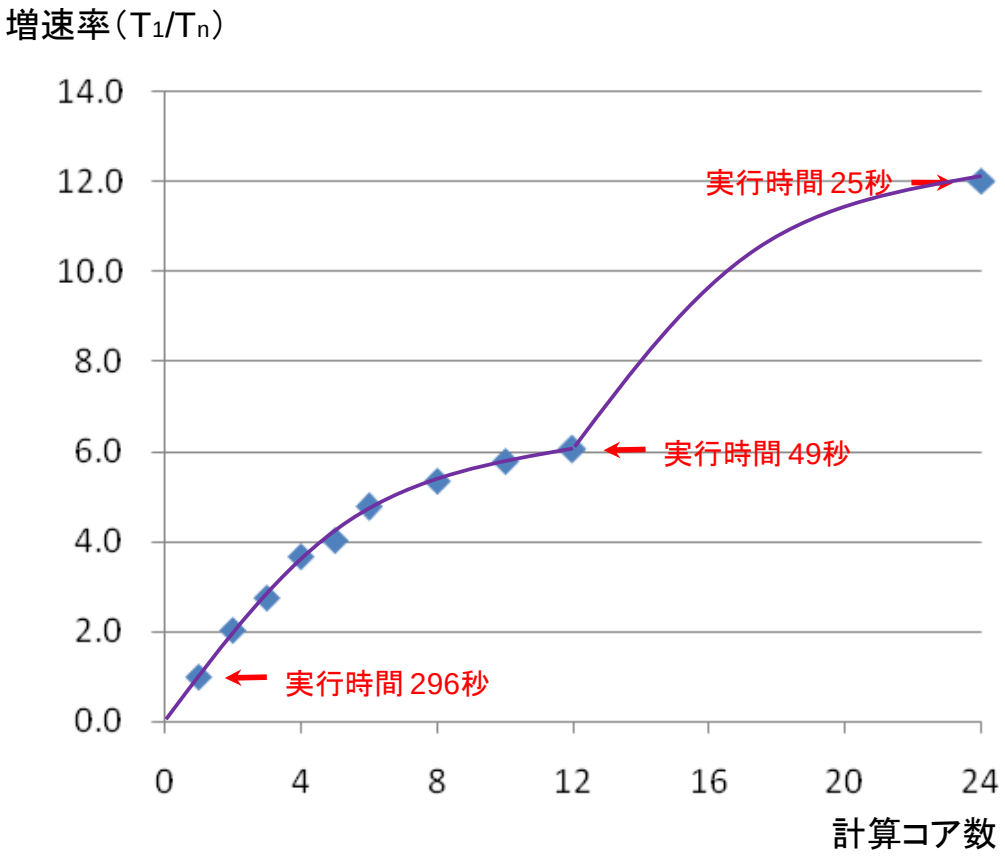
FOCUSスパコン

1～12コア：1ノード内

24コア：2ノード

注：Intel Xeon L5640(2.26GHz)

×2CPU(計12コア)/ノード



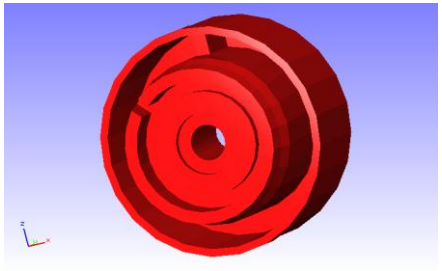
FrontISTRの並列処理性能（小規模）

解析対象

静応力解析（四面体二次要素）

要素数：684，807

節点数：1，008，911



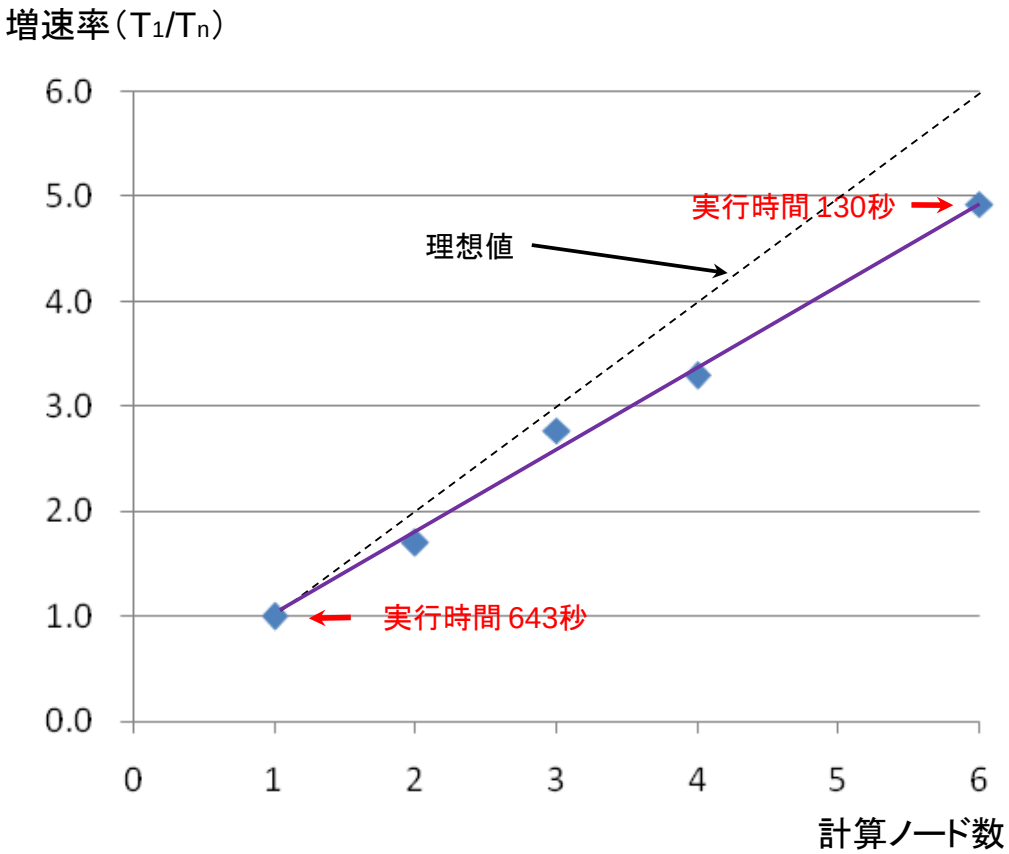
使用計算機

FOCUSスパコン

1～6ノード

注：Intel Xeon L5640(2.26GHz)

×2CPU(計12コア)/ノード



FrontISTRの並列処理性能（大規模）

解析対象

静応力解析（四面体二次要素）

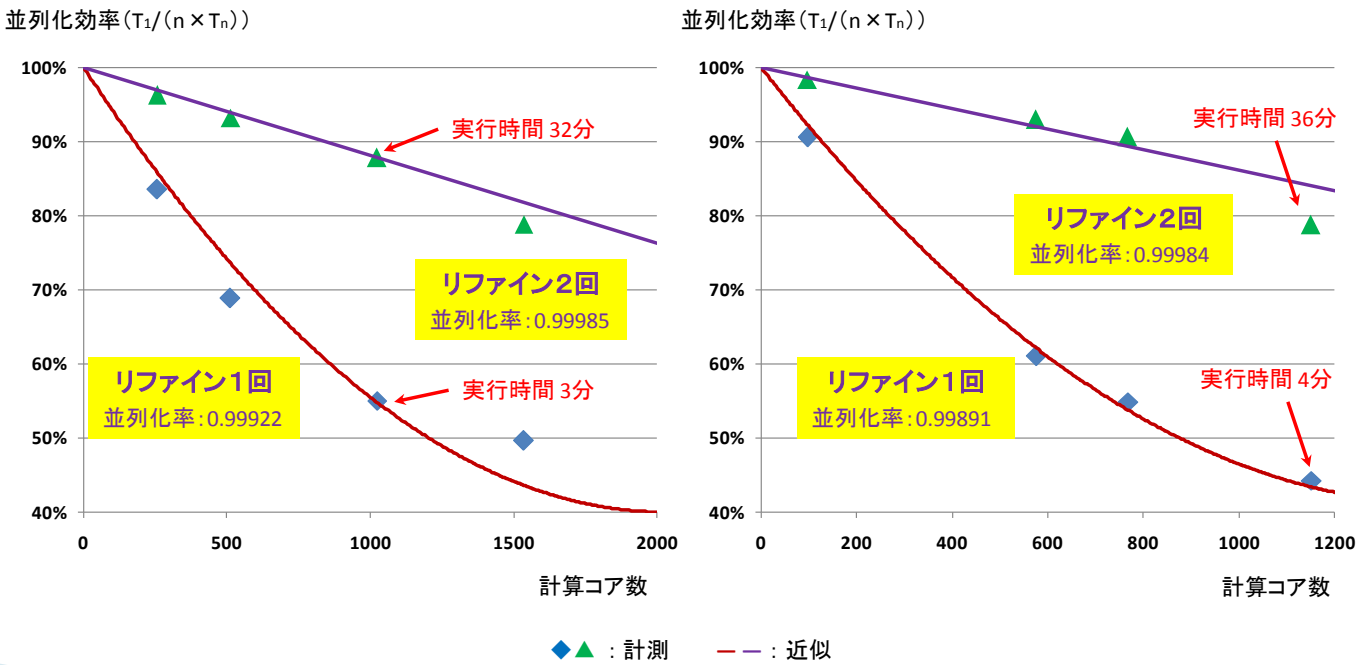
リファイン	要素数	節点数
なし	684,807	1,008,911
1回	5,478,456	7,707,758
2回	43,827,648	60,089,084

東京大学FX10

SPARC64 Ixfx(1.848 GHz)
×1CPU（16コア）/ノード

FOCUSスパコン

Intel Xeon L5640(2.26GHz)
×2CPU（計12コア）/ノード



Refiner使用時のポスト処理

- Refiner使用時、解析結果（res）データは細分化された節点・要素についても物理量が出力される
- ポスト処理への連携は、オリジナルメッシュへのマッピング（＝間引き処理）を方針としている
  （現状、節点物理量のみを対象）
- 解析終了後、rmerge（Refiner使用は内部判定）を使用して、resファイルを統合
- この後、hemw_vis1でUCDファイル変換も可能