

# OpenFOAM勉強会 「C++プログラム相談」のご案内

2012.12.15

オープンCAEシンポジウム2012

金田 誠 (OpenFOAM勉強会 for beginner@関東)

# C++学習法

- C++プログラマを目指す人
- OpenFOAMの解析者



学習法が同じであって良いはずがない

解析者はC++ばかりに時間を割いていけない

# 「本を紹介して」と言われると悩む

## ● Effective C++

Scott Meyers (著)

C++プログラマを目指す人が、入門書の次に読む本として優れた名著。

ただし、OpenFOAMの解析者にお勧めして良いものか、悩ましい。

高度なことも書かれている。勉強しすぎかも。

# そうは言っても、OpenFOAMは難しい

- OpenFOAMのソースコードは容赦ない  
templateや多重継承のような上級者向けの機能を、  
容赦なく使っている。

細かい話は避けませんが、OpenFOAMの多重継承の  
使い方は行儀が悪い(と思う)。



余計に難しい。

ちゃんとC++を勉強しないと歯が立たない？

C++を知っている人に聞いてしまえば楽！

OpenFOAM勉強会@関東で

「C++プログラム相談」を始めました。

少し自己紹介

金田 誠

- C++プログラマ出身の流体解析エンジニア
- 業務では宇宙輸送機周りの極超音速流れの解析
- 職場では格子職人と呼ばれています

# C++プログラム相談 事例紹介

いままでのご相談事例から、3件ご紹介します。

- 1件目：連成計算の通信部分を並列計算に対応させたい
- 2件目：連成計算がFOCUSスパコンで動かない
- 3件目：ソースカスタマイズしたソルバを実行すると実行時エラーで落ちる

# 1件目:連成計算の通信部分を 並列計算に対応させたい

オープンCAEワークショップ2012 - オープンCAE学会 - Mozilla Firefox

ファイル(E) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

オープンCAEワークショップ2012 - オープンCAE... +

← www.opencae.jp/wiki/オープンCAEワークショップ2012

## オープンCAEワークショップ2012 / OpenCAE Workshop

多くの方にご参加いただき、まことにありがとうございました。

### 講演会プログラム

- **13:00-13:10 開会式** 司会:田辺正孝(オープンCAE学会)
  - 13:00-13:10「開会の挨拶」 吉田正典(オープンCAE学会会長)
- **13:10-13:50 基調講演** 司会:今野雅(東京大学)
  - 13:10-13:50「オープンソースCAEソフトウェアの役割と課題」(R)
- **13:50-14:00 休憩**
- **14:00-15:40 一般講演セッション** 司会:中川慎二(富山県立大学)
  - 14:00-14:20「OpenFOAMにおけるImmersed Boundary法を
  - **14:20-14:40「OpenFOAM®と空調シミュレータの連成解析」**
  - 14:40-15:00「ppOpen-ATにおけるOpenFOAM®自動最適化
  - 15:00-15:20「OpenFOAM®による衝撃波伝播計算の計算精
  - 15:20-15:40「OpenCAE学会活動報告(Khronos Group、KIT

## 2件目：連成計算がFOCUSスパコンで動かない

- 空調シミュレータとOpenFOAMを連成。
- 空調シミュレータ：単一プロセス
- OpenFOAM：フラットMPIで並列計算
- 空調シミュレータとOpenFOAMの間はソケット通信
  - 手元のワークステーションでは動く
  - FOCUSスパコンで通信できないというご相談
  - ソケット通信をMPIで書き換えてみた



# 前提知識：並列処理のモデル

- SPMD【Single Program Multiple Data】
  - ・ 同じプログラムで並列処理を行う
  - ・ MPIのプログラムは、ほとんどがSPMDモデル（OpenFOAMも）
- MPMD【Multiple Program Multiple Data】
  - ・ 複数のプログラムを協調させて並列処理を行う
    - 今回はこっち。私は経験がなかった。

# ソースカスタマイズの要件

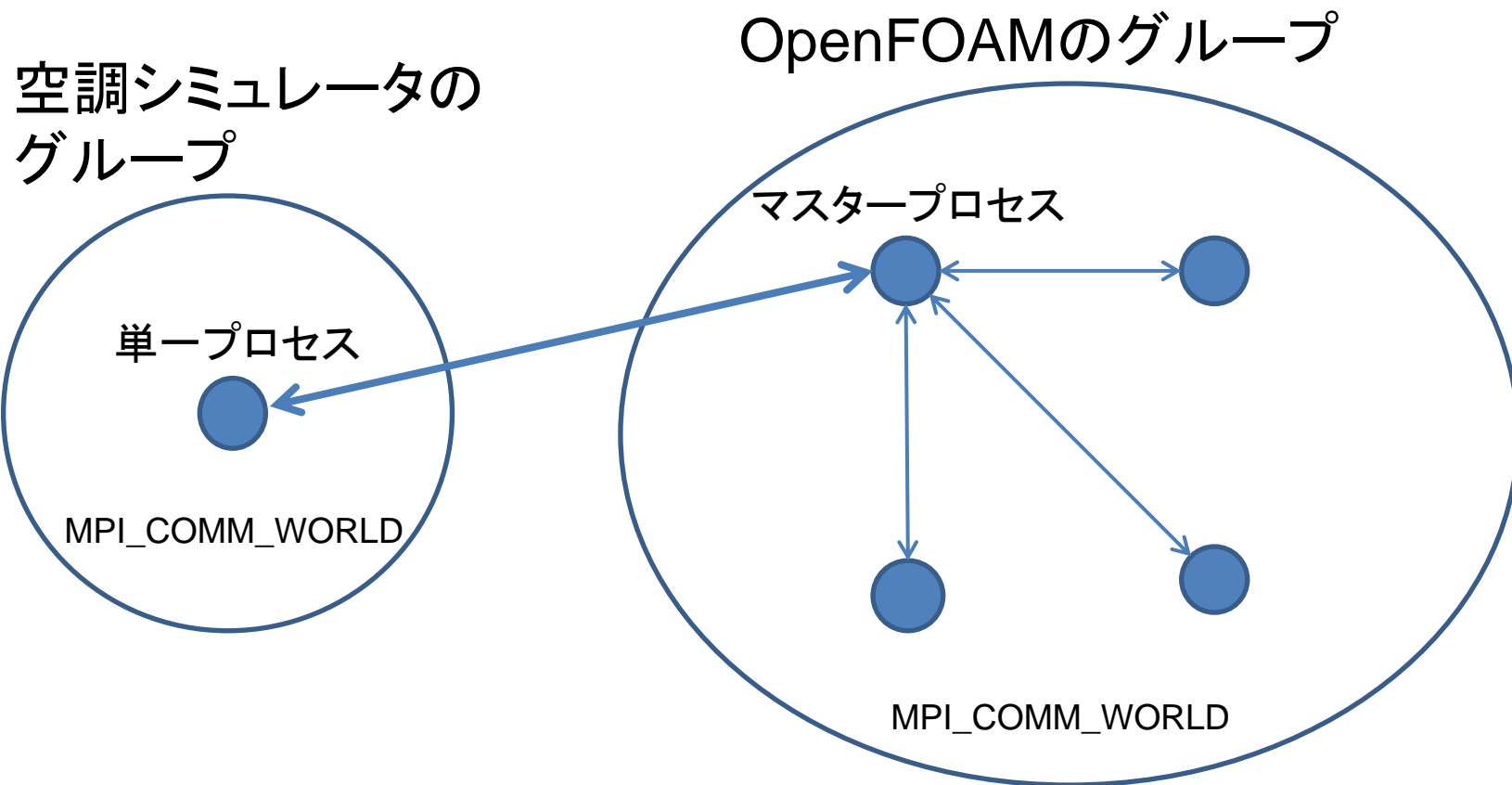
OpenFOAMは、可能ならライブラリに手を加えたくない。

ソルバのみのカスタマイズに留めたい。



MPIの機能で、空調シミュレータとOpenFOAMを別グループに分けることにより実現。

# グループ間通信



# グループ間通信

- MPIのグループ間コミュニケータを使って通信
  - ・ 今回は、OpenFOAM側は、マスタープロセスのみが空調シミュレータと通信するようにした。



マスタープロセスが受け取ったデータを、  
各ワーカープロセスに分配。

# カスタマイズ後の動作確認

- DEXCS上では動いた。
- FOCUSスパコンでは、まだ動作確認していないそうです。
  - 動作確認できたら、いずれ発表できるかも。

# プログラムの起動方法 ( OpenMPIの場合 )

(1) ネームサーバーを立ち上げる。

```
$ ompi-server -d --report-uri $HOME/ompi-uri.dat
```

※ \$HOME/ompi-uri.datは任意のファイル名にして良い。

連成するプログラム間を動的に接続するとき、  
サービス名の解決に使用。

システムにompi-serverがないときの最終手段は？

→ OpenMPIをソースからビルド

# プログラムの起動方法 ( OpenMPIの場合 )

(2) 空調シミュレータを起動する。

```
$ mpiexec -mpi-server file:$HOME/mpi-uri.dat -np 1
```

逐次処理でも mpiexecを使って動かす必要がある。

(3) OpenFOAMを並列で起動する。

```
$ mpiexec -mpi-server file:$HOME/mpi-uri.dat -np 4 aaaFoam -parallel
```

# 処理の流れ(空調シミュレータ側)

```
// クライアント(OpenFOAM)が接続するポートを生成する
MPI_Open_port(MPI_INFO_NULL, port_name);

// ポート名をサービス名に関連付ける。サービス名はプログラマが指定できる
MPI_Publish_name(const_cast<char*>(service_name.c_str()), MPI_INFO_NULL, port_name);

// クライアントの接続要求を受け付ける
MPI_Comm_accept(port_name, MPI_INFO_NULL, 0, MPI_COMM_SELF, &client_comm);

// 処理

// ポート名とサービス名の関連を切り離す
MPI_Unpublish_name(const_cast<char*>(service_name.c_str()), MPI_INFO_NULL, port_name);

// ポートを閉じる
MPI_Close_port(port_name);

// クライアントを切り離す
MPI_Comm_disconnect(&client_comm);
```



# 処理の流れ (OpenFOAM側)

```
// サービス名を渡して、特定のポート名に接続する
MPI_Lookup_name(const_cast<char*>(service_name.c_str()), MPI_INFO_NULL, port_name);

// サーバー(空調シミュレータ)との接続を確立する
MPI_Comm_connect(port_name, MPI_INFO_NULL, 0, MPI_COMM_SELF, &server_comm);

// 処理

// サーバーを切り離す
MPI_Comm_disconnect(&server_comm);
```

# 参考文献

コンピューター一般

■ [実践MPI-2 メッセージパッシング・インタフェースの上級  
者向け機能](#)

和書 at 2007/8/31 11:16:18

2005 | 5,000~10,000円



訳: 畑崎 隆雄

著者: William Gropp, Ewing Lusk, Rajeev Thakur

刊行: 2002年10月

ページ数: 376

税込価格: 5,040円

ISBN10: 4894714442

ISBN13: 9784894714441

# 3件目: ソースカスタマイズしたソルバを実行すると、実行時エラーで落ちる

```
#0 Foam::error::printStack(Foam::Ostream&) in  
  "/opt/OpenFOAM/OpenFOAM-  
  2.1.1/platforms/linux64GccDPOpt/lib/libOpenFOAM.so"  
#1 Foam::sigSegv::sigHandler(int) in "/opt/OpenFOAM/OpenFOAM-  
  2.1.1/platforms/linux64GccDPOpt/lib/libOpenFOAM.so"  
#2  
  at sigaction.c:0  
#3 typeinfo for Foam::HashTable<Foam::regIOobject*, Foam::word,  
  Foam::string::hash> in "/opt/OpenFOAM/OpenFOAM-  
  2.1.1/platforms/linux64GccDPOpt/lib/libOpenFOAM.so"
```

セグメンテーション違反です (コアダンプ)

# 乱流モデル変数のコンストラクタ呼び出し

```
PtrList<incompressible::RASModel> turbulence(fluidRegions.size());
```

(中略)

乱流モデル変数を格納するコンテナ

```
forAll(fluidRegions, i) {
```

forAllループのブロック開始

```
singlePhaseTransportModel laminarTransport(UFluid[i], phiFluid[i]);
```

```
turbulence.set (
```

```
  i,
```

```
  autoPtr<incompressible::RASModel> (
```

```
    incompressible::RASModel::New (
```

```
      UFluid[i],
```

```
      phiFluid[i],
```

```
      laminarTransport
```

```
    )
```

```
  ).ptr()
```

繰り返しの度に、  
laminarTransport変数の  
コンストラクタが呼ばれる。

乱流モデルクラスのコンストラクタに参照が渡されて、乱流モデル内部から参照される

```
);
```

(中略)

```
}
```

forAllループのブロック終了。

繰り返しの度に、laminarTransport変数のデストラクタが呼ばれる。

# 結論

- turbulence変数(テナ)は、forAllループの外で定義されているため、forAllループが終わっても生き続ける。
- laminarTransport変数は、forAllループの中でローカル変数として定義されているため、forAllループの繰り返しのたびに生成→破壊が行われる。
- その後、turbulence変数の要素から、寿命の尽きたlaminarTransport変数を参照したときに、ソルバが実行時エラーで落ちる。

# C++プログラム相談、受付中

OpenFOAM勉強会内の活動として、

C++プログラム相談を受け付けております。

- 名前: 金田 誠
- メール: kaneda@makoto.nifty.jp
- メールまたは勉強会@関東で直接お声掛けください。
- 勉強会でのご相談は無料ですが、**ビジネスの話も歓迎します**。その場合も、まず勉強会の無料相談をお試しいただけます。