J. Plasma Fusion Res. Vol.82, No.9 (2006) 638-641

連載コラム High Performance Fortran で並列計算を始めよう

2. 使ってみよう HPF

岩下英俊，林　康晴[1)]，石黒静児[2)]
富士通　ソフトウェア事業本部，[1)]NEC　第一コンピュータソフトウェア事業部，[2)]核融合科学研究所

（原稿受付：2006年7月3日）

2.1 はじめに

今回は，文法やプログラミング作法の話はおくことにして，まずはHPFを使ってみましょう．フリーのHPFコンパイラをインストールし，サンプルプログラムをコンパイルして実行するまでの手順を紹介します．他に必要なソフトウェアも，大体フリーのもので揃います．うまくいけば，今日からでも並列実行の効果を目にすることができるでしょう．並列数を増やせば計算スピードが上がっていく様子を，ぜひ実際に確認してください．

以下では，まずHPFコンパイラの役割と，HPFコンパイラfhpfを紹介します（2.2章）．続いて，fhpfのインストールと（2.3章）実行環境を（2.4章）説明します．そして，サンプルプログラムを使って，コンパイルから実行までを具体的に示します（2.5章）．

2.2 HPF コンパイラ

2.2.1 HPF コンパイラの仕事

PCクラスタ，ブレードサーバや多くのスーパーコンピュータは，分散メモリ型の計算機です．HPFでは，この分散された単位（CPUとメモリのペア）をプロセッサと呼びます．分散メモリ型計算機を使って並列計算を行うためには，1プロセッサだけで計算する場合とは違い，プロセッサ間のデータのやりとり（通信）や実行の進み具合の調整（同期）が必ず必要になります．

分散メモリ型の計算機で並列プログラムを記述するには，今のところMPI（Message Passing Interface）ライブラリ[1]が最も多く使われています．これは，FortranやCのプログラムの中で，ライブラリ呼出しの形で，すべての通信や同期を直接指示する方法です．非常に手間がかかる上に間違いやすく，技術的にも計算機の専門家でなければ難しい作業です．

HPFコンパイラの一番大きな仕事は，このように煩わしく難しい通信や同期の指示を，自動的に生成することにあります．

2.2.2 HPF を MPI と比べてみる

わかりやすい例として，2.5章でサンプルプログラムとして取り上げる姫野ベンチマークで比較してみましょう．

Table 1　姫野ベンチのソース行数の比較．

	プログラム全体	MPI呼出しを含む行	HPF指示行
Fortran版	135		
Fortran+MPI版	487	40	
HPF版サンプルV1	154		13
HPF版サンプルV2	146		13
HPF版3次元分割	159		30

コメント行と空行を除くソースプログラムの行数（単位：行）

姫野ベンチマークは，ポアソン方程式をJacobi反復法で解く計算パターンを元にして作成された性能評価プログラムですので，サイズは小さくても流体系の実プログラムに近いものであるといえます．Table 1に，Fortran，MPI，およびHPFで書かれた姫野ベンチマークプログラムの規模を示します．Fortran版（himenoBMTxp_m.f）とFortran+MPI版（himenoBMTxpr.f）は姫野ベンチマークのサイト[2]からダウンロードできます．HPF版はFortran版を基に我々が作成したプログラムで，サンプルV1とV2はHPF推進協議会のサイト[3]からダウンロードできます．3次元分割版は，fhpfのソフト一式に付属しています．

FortranプログラムをMPIで並列化すると，全体の行数が3.6倍になっています．これに比べて，HPFではわずかな指示行の追加とプログラムの修正ですむことがわかります．煩雑な通信や同期の生成は，HPFコンパイラが自動でやってくれるからです．

では性能はどうかというと，このような流体系のアプリケーションに対しては，最近のHPFコンパイラはMPIとほとんど差がなくなってきています．Table 1のサンプルV2とFortran+MPI版を実測した結果をFig. 1に示しています．計算機は決して高価なマシンではなく，670 MHzのPentium III CPUとイーサーネットの多段結合を使ったブレードサーバです．HPFのコンパイルには今回紹介するfhpfコンパイラを使い，他には富士通のLinux Parallel Language Package V1.0のFortranと，フリーのMPIであるMPICH V1.2.6を使用しました．

なお，Fortran+MPI版は3次元分割が可能ですが，ここ

Let Us Start Parallel Processing Using High Performance Fortran ! 2. Let Us Use HPF

IWASHITA Hidetoshi, HAYASHI Yasuharu and ISHIGURO Seiji
corresponding author's e-mail: iwashita.hideo@jp.fujitsu.com

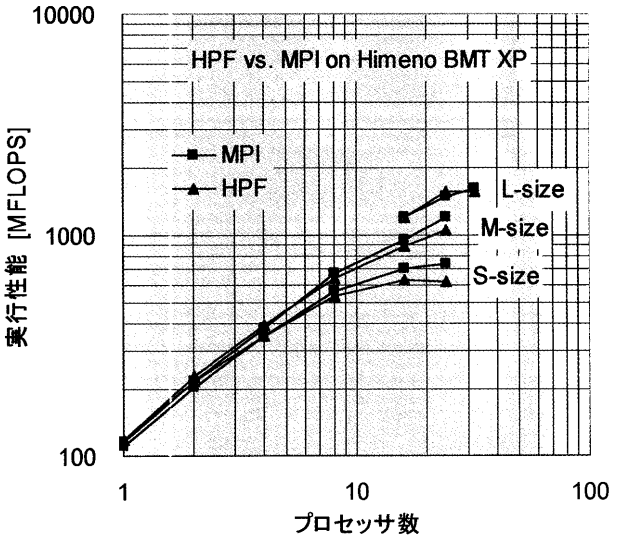

Fig. 1 姫野ベンチの性能比較（1次元分割）.

ではZ軸方向の1次元分割だけで測定しました．3次元分割を使えばさらに高並列まで性能が伸びます．HPF版の3次元並列でも同様です．シリーズの今回と続く2回は，入門編として1次元分割を中心に取り上げていきます．

2.2.3 HPF コンパイラ fhpf

fhpfは，HPFプログラムを入力として，Fortranプログラムを出力する，ソース-to-ソースのコンパイラ（トランスレータ）です[4]．MPIを使用して通信や同期を行うため，出力のFortranプログラムはMPIライブラリの呼出しを含みます．つまりfhpfは，Fortran+MPIのプログラムを自動生成するツール，と見ることもできます．

fhpfを使ってHPFプログラムをコンパイルし実行する様子をFig. 2に示します．fhpfの出力プログラムは，MPI以外の特別なライブラリの呼出しを含みませんので，標準的なFortranとMPI さえサポートされていれば，どのような計算機上でも実行することができます．

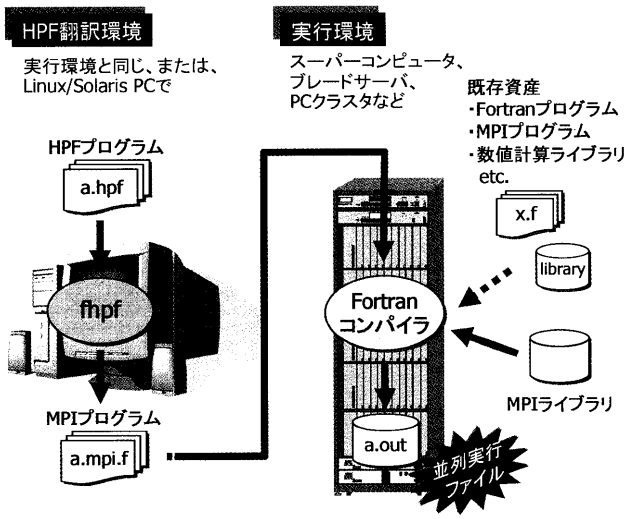

Fig. 2 fhpfによる翻訳・実行の流れ.

2.3 HPF翻訳環境の作り方

並列計算機を比較的自由に使える方は，fhpfを実行環境にインストールすれば便利でしょう．そうでない方は，手近なPCなどへのインストールをお薦めします．手近な計算機でfhpfを使い，実行は並列計算機へ送って行います．

2.3.1 fhpfの動作環境

fhpfコンパイラにはLinux版とSolaris版があり，それぞれ以下のような条件を満たす計算機環境で動作します．WindowsPCやMacintoshをお使いの方は，VMWare[5]を介してLinuxを実装し，その上で使う方法もあります．

	Linux版	Solaris版
CPU	Intel IA32系	SPARC系
オペレーティングシステム	Linuxカーネルバージョン2.4以降	Solaris 7(SunOS 5.7)以降

どちらの版でも，生成されるFortranプログラムは同じになります．実行環境とは関係なく，日頃使いやすい計算機を利用してください．

2.3.2 fhpfの入手とインストール

fhpfは，HPF推進協議会の正会員・準会員に無償で配布されていますが，本連載の読者の方々へのサービスとして，期間限定でダウンロードできるようにしています．HPF推進協議会のホームページ[3]から辿ってください．

インストール方法は，同時に提供されるインストールガイドをご覧ください．通常は適当なディレクトリに展開するだけです．

2.4 実行環境の作り方

どんな計算機上でも，標準的なFortranとMPIさえサポートされていれば，実行することができます．PCクラスタやブレードサーバはもちろん，SMPクラスタならノード内にもノード間にも利用可能です．1CPUの計算機でも並列実行の実験はできます（性能は期待できませんが）．

FortranまたはMPIが使えない計算機では，以下に紹介するように入手してインストールしてください．

2.4.1 Fortran コンパイラ

Fortran90仕様をサポートしたものが望ましいです．市販のIntel製と富士通製のFortranではfhpfとの組み合せに問題ないことを確認しています．フリーのものではGNUのgfortranとg95があります．

GNUのg77はLinuxの大抵のディストリビューションに入ってきますが，FORTRAN77仕様です．fhpfはオプション指定でFORTRAN77コンパイラにも対応しますが，機能が制限されますので，Fortran90コンパイラの利用をお薦めします．詳しくはfhpfのユーザーガイドを参照してください．

2.4.2 MPI ライブラリ

fhpfが使用するMPI仕様はVersion 1.1の範囲ですので，大抵のMPIライブラリなら使用可能です．フリーのものではMPICH[6]やLAM/MPI[7]があり，Intelや富士通などから市販されているものもあります．

MPICHはソースコードで提供されるので利用者環境で

makeが必要です．使用するFortranへのパスの指定が必要ですが，それ以外は特に難しくないと思います．詳しくはMPICHのドキュメントを参照してください．

2.5 コンパイルから実行まで

fhpfを使って，HPFのサンプルプログラムをコンパイルして，並列実行してみましょう．

2.5.1 サンプルプログラムの入手

HPF推進協議会のホームページのSample Codesに，姫野ベンチマークのHPF版が2つリンクされています．これらをダウンロードしてください．

問題サイズはMとしています．このプログラムは，予備実行で実行時間を見積った後，本実行が1分程度で終了するようにできていますが，予備実行に非常に時間がかかるとか，メモリが不足するようなら，プログラムに手を入れてサイズを小さくしてください．V1ではプログラム中に，V2ではインクルードファイル中に，以下のように計算空間のサイズを決めるPARAMETER文があります．

```
C  PARAMETER (mimax=1025,mjmax=513,mkmax=513)
C  PARAMETER (mimax=513,mjmax=257,mkmax=257)
   PARAMETER (mimax=257,mjmax=129,mkmax=129)
C  PARAMETER (mimax=129,mjmax=65,mkmax=65)
C  PARAMETER (mimax=65,mjmax=33,mkmax=33)
```

上から順に問題サイズXL，L，M，S，XSに対応します．どれか1つを残して他をコメントアウト(頭にCを付ける)することで選択できます．V1ではプログラム中に3ヶ所ありますので，必ず同じ選択にしてください．

2.5.2 サンプルV1

V1は使いやすさを重視した版です．一度コンパイルしたら，同じ実行ファイルで並列数を変えて実行できます．

(1) HPF翻訳環境で

himenoBMTxp_HPFv1．fをHPF翻訳環境の作業ディレクトリに置き，次のようにタイプすると，MPI呼出しを含むFortranプログラムが生成されます．

```
fhpf himenoBMTxp_HPFv1.f
```

生成されるファイルの名前は，入力のサフィックス".f"を".mpi.f90"に変えたhimenoBMTxp_HPFv1.mpi.f90になります．fhpfの変換に興味のある人は，エディタで入力プログラムと見比べてみてください．

(2) 実行環境で

himenoBMTxp_HPFv1.mpi.f90を，実行環境の作業ディレクトリにコピーし，MPIプログラムとしてコンパイルします．このときのコマンドは使用するFortranとMPIによって違います．例えば，コマンドmpif90では，次のようにタイプします．

```
mpif90 -o hime1 himenoBMTxp_HPFv1.mpi.f90
```

詳しくは使用するMPIのマニュアルを確認してください．実行は並列数を指定して起動します．例えば4並列で実行させる場合，MPICHでは，

```
mpirun -np 4 hime1
```

富士通のMPIでは，

```
mpiexec -mode limited -n 4 hime1
```

などとタイプします．富士通MPIではこのように，高速なlimitedモードが選択できます．詳しくはそれぞれのマニュアルを参照してください．

実行が正常に終了すると，色々なメッセージに続いて，以下のように性能値がMFLOPSの単位で出力されます．

```
Loop executed for 622 times
Gosa : 9.25013679E-04
MFLOPS: 1450.47632  time(s): 58.7940636
Score based on Pentium III 600MHz : 17.5093708
```

サンプルV1では，同じ実行ファイルで，-npや-nで指定する並列数を変えて実行することができます．並列数の増加に沿って，性能が上がっていく様子を確認してください．並列数が実装されているCPUの数を超えるとそれ以上性能は上がりませんが，正常に実行されます．

また，1プロセッサではメモリ不足で実行できなくても，並列数を上げれば実行できる場合があります．問題サイズを大きくしてそれを確認するのも面白いと思います．

2.5.3 サンプルV2

V2はV1よりも性能を重視した版です．プログラム中でプロセッサの数が静的に宣言されていますので，並列数が固定されたプログラムです．

(1) HPF翻訳環境で

himenoBMTxp_HPFv2.fとparam_HPFv2.hをHPF翻訳環境の作業ディレクトリに置いてください．プロセッサ数はparam_HPFv2.hのNPEで定義されています．

```
    PARAMETER (NPE=2)
```

8並列で実行するには，この値を8と修正してください．HPF翻訳では，オプション-oで生成するFortranプログラムの名前を変更できますので，例えば以下のように，名前にプロセッサ数を反映しておくのがよいでしょう．

```
fhpf himenoBMTxp_HPFv2.f -o hime2_8.f90
```

この手順を繰り返して，hime2_4.f90，hime2_2.f90，…などとプロセッサを変えたMPIプログラムを用意します．

(2) 実行環境で

それぞれのMPIプログラムについてコンパイルを行います．コマンドはV1と同じです．

```
mpif90 -o hime2_8 hime2_8.f90
mpif90 -o hime2_4 hime2_4.f90
mpif90 -o hime2_2 hime2_2.f90
…
```

実行もV1と同様ですが，注意としてV2では，-npや-nで指定する並列数が，プロセッサ数NPEの値と必ず一致しなければなりません．

```
mpirun -np 8 hime2_8
mpirun -np 4 hime2_4
mpirun -np 2 hime2_2
…
```

さて，Fig. 1のような並列化効果が確認できたでしょうか．

2.5.4 備考：V1とV2の違いについて

サンプルV1とV2のプログラム上の大きな違いを，Table 2にまとめました．プロセッサ数無依存と固定の使い勝手の違いは，実際に比べて分かってもらえたと思います．プロセッサ数無依存の方が便利ですが，プログラムの書き方によっては，性能が落ちる場合があります．そのため，V1では手続き間のデータ受渡しをオリジナルのFortran版から変更しています．また，V1はコンパイラの自動並列化に頼った書き方，V2は全く頼らない書き方となっています．

こういった，性能を考慮したプログラミングや，自動並列化機能との付き合い方については，シリーズの次回以降で解説していきたいと思います．

2.6 最後に

HPF推進協議会では，講習会などの活動も行っています．HPFとその実用化拡張であるHPF/JA言語[8]，そして将来の並列言語に興味をお持ちの方は，ホームページ[3]をご覧の上，この機会にぜひご入会ください．

fhpfに関するお問合せ，ご意見・ご要望など，すべてfhpfの問合せ窓口fhpf@hpfpc.orgで受け付けています．お気軽に電子メールでお寄せください．お試しいただいた結果なども，今後の活動のため大変参考になります．

連載の次回と次々回は，「作ってみようHPFプログラム」と題して，HPFプログラミングを基礎から解説します．

Table 2 サンプルプログラムの主な違い．

	サンプルV1	サンプルV2
プロセッサ数	無依存	固定
手続き間データ受渡し	引数	common変数
自動並列化機能	使用する	使用しない
インクルードファイル	使用しない	使用する

謝辞

姫野ベンチマークプログラムの利用にあたっては，理化学研究所姫野龍太郎博士のご理解とご協力をいただきました．

参考文献

[1] Message Passing Interface Forum (http://www.mpi-forum.org/).
[2] Himeno ベンチマークプログラム (http://accc.riken.jp/HPC/HimenoBMT/).
[3] HPF 推進協議会 (http://www.hpfpc.org/).
[4] 岩下英俊，青木正樹：HPF トランスレータ fhpf における分散種別を一般化したコード生成手法，情報処理学会論文誌コンピューティングシステム(ACS15)，2006年8月（掲載予定）．
[5] VMWare (http://www.vmware.com/jp/).
[6] MPICH (http://www-unix.mcs.anl.gov/mpi/mpich1/).
[7] LAM/MPI (http://www.lam-mpi.org/).
[8] High Performance Fortran Forum 著，富士通，日立，日本電気訳，High Performance Fortran 公式マニュアル（シュプリンガー・フェアラーク東京，1999年）(付録にHPF/JA言語仕様書)．

岩下英俊（いわした ひでとし）

1963年生まれ．1986年愛媛大学工学部電子工学科卒業．1988年同大学大学院工学研究科修士課程修了．同年㈱富士通研究所入社．1997年より富士通㈱．入社以来，主にスーパーコンピュータの言語処理系に関わる研究開発に従事．HPC向け並列プログラミング言語の設計，並列化コンパイラの開発などを行ってきた．博士（工学）．

林 康晴（はやし やすはる）

1992年京都大学理学部卒業．同年NEC入社．以来ベクトル・並列コンピュータ向けソフトウェア，特に分散メモリマシンのための並列化コンパイラの研究開発に従事．現在第一コンピュータソフトウェア事業部主任．

石黒静児（いしぐろ せいじ）

自然科学研究機構核融合科学研究所理論・シミュレーション研究センター教授．総合研究大学院大学物理科学研究科核融合科学専攻教授(併任)．名古屋大学大学院理学研究科博士課程修了後，東北大学大学院工学研究科助手，核融合科学研究所助教授を経て，2004年4月より現職．粒子シミュレーションにより，主としてプラズマ中電位構造形成・制御，高強度電磁波とプラズマの相互作用などのプラズマ基礎特性に関する研究に従事している．