



# fhpfコンパイラの 無償配布について

富士通(株) ソフトウェア事業本部

岩下英俊

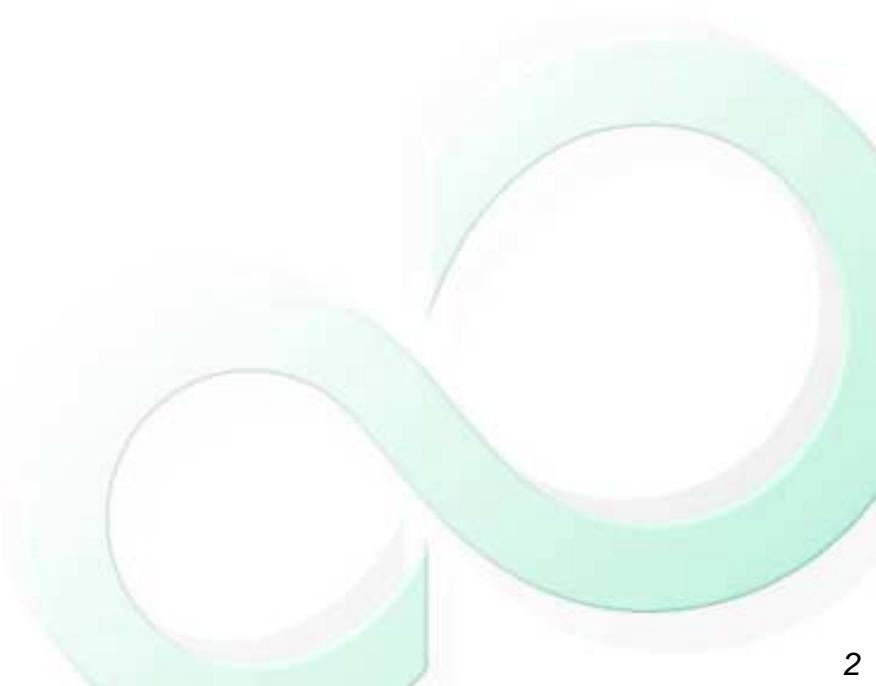
2004年12月15日



# 内容

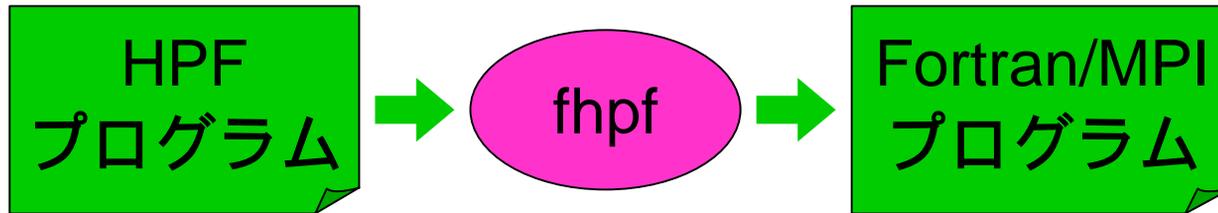
---

- fhpfとはどういうものか
- 現在の機能範囲
- ライセンスに関して
- まとめ



# HPFトランスレータfhpf

- ソース-to-ソースのコンパイラ



- Linux版とSolaris版
- 通信ライブラリにはMPIを使用
  - FortranとMPI1.1があれば、どこでも実行可能
  - MPIプログラムの自動生成ツール

# 翻訳から実行まで

## (1) HPF翻訳

- fhpfコンパイラを使用

```
% fhpf a.hpf b.hpf
```

## (2) Fortran/MPI翻訳

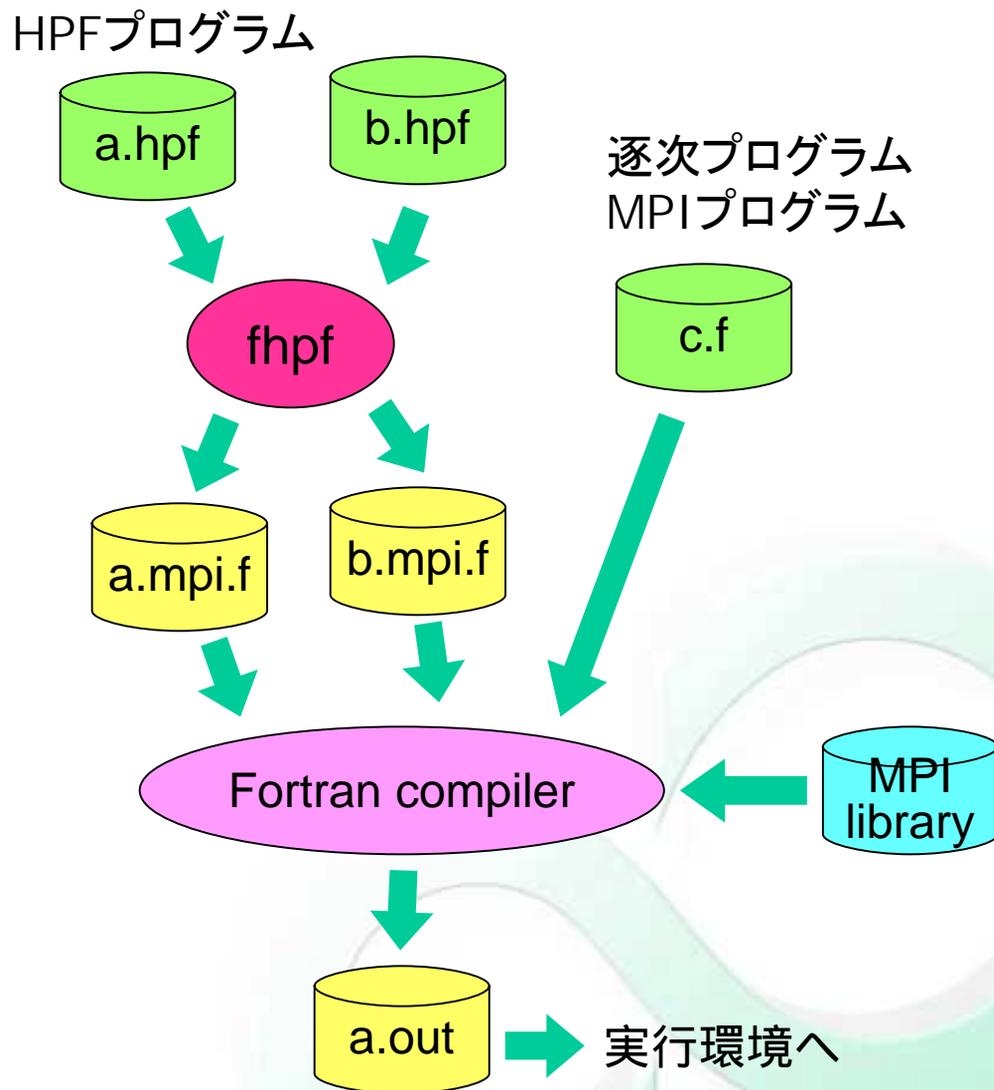
- お手持ちのFortranコンパイラとMPIを使用

```
% mpif77 a.mpi.f b.mpi.f c.f
```

## (3) MPIプログラム実行

- お手持ちの実行環境で

```
% mpirun -np 4 a.out
```



# fhpfの変換の実例

Input file: block.hpf

```
integer A(100)
!hpf$ processors P(4)
!hpf$ distribute A(block) onto P

!hpf$ independent
do i=n1,n2
  A(i)=i
enddo
end
```

Command lines

```
% fhpf block.hpf -f77
fhpf V1.2/MPI -- HPF translator for Linux system
block.hpf -> block.mpi.f
%
```

Output file: block.mpi.f

```
PROGRAM ORG__MAIN__
INCLUDE 'mpif.h'
EXTERNAL ORG__MAIN__
INTEGER*4 ORG_RANK
INTEGER*4 ORG_SIZE
INTEGER*4 ierrX0
COMMON /ORG_COMMON/ORG_RANK,ORG_SIZE
CALL mpi_init(ierrX0)
CALL mpi_comm_rank(MPI_COMM_WORLD,ORG_RANK,ierrX0)
CALL mpi_comm_size(MPI_COMM_WORLD,ORG_SIZE,ierrX0)
CALL ORG__MAIN__()
CALL mpi_finalize(ierrX0)
END
SUBROUTINE ORG__MAIN__()
INCLUDE 'mpif.h'
INTEGER*4 a(0:24)
INTEGER*4 i
INTEGER*4 n1
INTEGER*4 n2
INTEGER*4 ORG_RANK
INTEGER*4 ORG_SIZE
INTEGER*4 spmdX0
INTEGER*4 spmdX1
INTEGER*4 spmd_startX0
INTEGER*4 spmd_endX0
INTEGER*4 spmd_stepX0
INTEGER*4 ierrX0
COMMON /ORG_COMMON/ORG_RANK,ORG_SIZE
spmdX0 = mod(ORG_RANK,4)
spmdX1 = 25*spmdX0
IF (spmdX0.LE.(n1-1)/25) THEN
  spmd_startX0 = n1-1-spmdX1
ELSE
  spmd_startX0 = 0
ENDIF
IF (spmdX0.LT.(n2-1)/25) THEN
  spmd_endX0 = 24
ELSE
  spmd_endX0 = n2-1-spmdX1
ENDIF
spmd_stepX0 = 1
DO i=spmd_startX0,spmd_endX0,1
  a(i) = i+spmdX1+1
ENDDO
END
```

# 現在 (V1.2) の機能範囲 (1/3)

## • データ分散

- 全分散種別に対応
  - block, cyclic, block-cyclic, gen\_block, indirect
- フルshadowは未対応
  - 必須とは考えていない。
- DISTRIBUTE 指示文の ONTO 以下省略は不可
  - 当面 number\_of\_processors を推奨。
- ALIGN 指示文に制限
  - データの複製など(使う?)  
`distribute T(block) onto p`  
`align A(*,*) with T(*)`
- 文字変数のマッピングは不可

## • 手続間結合

- 指令的、記述的分散に対応。どちらも通信なしを実行時判定
- INHERIT 指示文、転写的分散は未対応
- EXTRINSIC 接頭辞は対応不完全だが、明示的引用仕様宣言のないサブプログラムはHPFでもFortranでも呼出し可能
  - 記述子でなくデータ実体を、分割後のイメージで引数に乗せるため。
- HPF\_LOCAL libraryは未対応

# 現在 (V1.2) の機能範囲 (2/3)

## • ループ並列

- INDEPENDENT指示の省略 (自動並列化可能判定) は、あまり強くない。
  - オプション指定時のみ機能
- ON指示の省略 (自動負荷分散) は、ある程度使える。
  - 並列化されないメッセージが出た場合には、必ず指定を。
  - shadow領域のアクセスには、ON+LOCAL指示が必須
- ON指示はIndependentループのbody全体を囲む形のみ有効

## • 通信

- ループ内通信の自動一括化は、未対応
  - 書換えを推奨。F90記述またはASYNCHRONOUS構文へ。

```
do j=1,N
  do i=1,M
    A(i,j)=B(i,j)
  end do
end do
```

↓

```
A(1:M,1:N)=B(1:M,1:N)
```

- ASYNCHRONOUS構文に対応
- REFLECT構文に対応 (非同期も)
- INDEX\_REUSEは無効

# 現在 (V1.2) の機能範囲 (3/3)

## •タスク並列

- TASK\_REGIONは未対応
  - 3月末対応予定

## •動的再分散

- REDISTRIBUTE指示文、REALIGN指示文は未対応
  - 1月末対応予定

## •Fortran90機能

- ポインタ、割付け配列の分散は未対応
  - 1月末対応予定
- 引数キーワードは未対応
- 総称指定のある引用仕様宣言は未対応
- WHERE構文は未対応

## •FORTRAN77以前の機能

- ASSIGN文と割当て型GOTO文、ENTRY文、選択戻り指定子は未対応

# fhpf V1.2 無償配布版



# 配布を受ける人、使う人

- 管理者（ソフトを受け取る人）
  - HPF推進協議会の会員・準会員の方々
  - インストールガイドに沿ってインストールし、使用者に使用させてください。
- 使用者
  - 管理者を含め、管理者の目の届く範囲の人
  - ユーザーガイドを参照して、本ソフトウェアを使用してください。ユーザーガイドのコピーは自由です。

# ライセンスに関して

## •インストールガイド「1. ライセンスに関して」

本ソフトウェアは、HPF推進協議会の会員および準会員の皆様への試供品です。

基本的にはバイナリフリーですが、会員・準会員の方とその周辺に限定したサービスとし、二次配布はお断りしています。コピーをご要望の方には、この機会にHPF推進協議会にご参加くださるようお勧めください。

- 二次配布をお断りする理由・・・知財権の問題ではなく
  - 利用者の声を聞きたい。
  - 開発元からの情報を届けたい。
  - HPFPCに参加して欲しい。

# 知的財産権に関する注意事項

## •インストールガイド「1.1 注意事項」

本ソフトウェアおよび付属する関連資料に関わる知的財産権は、富士通(株)(以下、当社と呼びます)に帰属します。

…(略)…

管理者は以下を遵守し、また、利用者にも遵守させてください。

- 本ソフトウェアまたはその複製を、利用者以外の第三者に使用させてはなりません。
- 本ソフトウェアまたはその複製を、第三者に販売、貸与または譲渡してはなりません。
- 本ソフトウェアに含まれるバイナリコードについて、逆アセンブル、リバースエンジニアリング、改造または別のソフトウェアへの組み込みを行ってはなりません。

# ご協力のお願い

fhpf専用窓口  
fhpf@hpfpc.org

## •インストールガイド「1.2 お願い」

本ソフトウェアを使用して得た結果（出力コード）については、利用者の所有物としてご自由にお使いいただけます。ただし、成果を論文や口頭発表の形で公開する場合には、以下の点についてできるだけ協力していただくよう、利用者にご指導ください。

- fhpfを使用した旨と版数を、本文、脚注または他のどこかに明記するか、口頭で伝えるようにしてください。
- 発表先と発表内容について、簡単でも結構ですので、管理者を通して窓口までお知らせください。差し障りのない範囲で結構です。

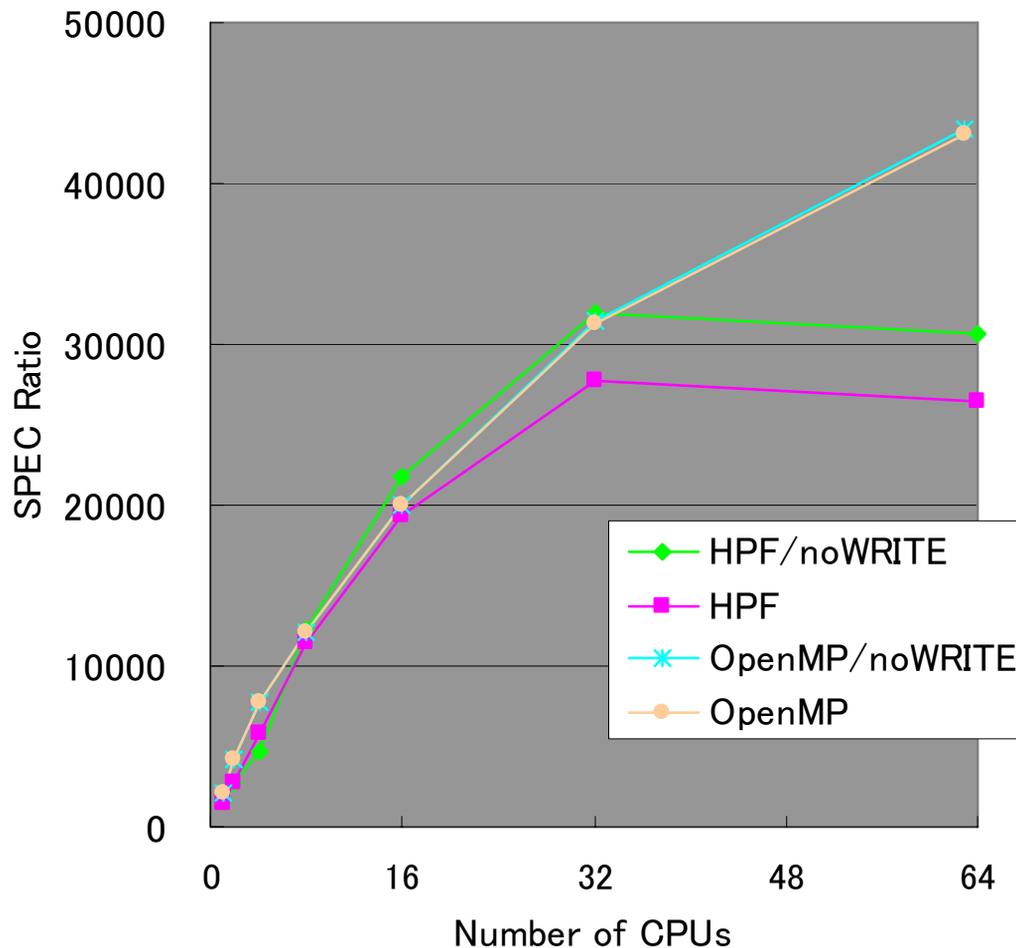
皆様のご意見・ご要望は、今後の開発にできるだけ反映していきたいと思っています。ぜひ窓口までお寄せください。

# まとめ

- fhpfは、
  - HPFからFortran+MPIを生成する、トランスレータ
  - Linuxクラスタ、ブレードサーバからスパコンまで
- 機能範囲
  - 動的再マッピング、割付け配列の分散を開発中
  - HPF/JA仕様の概ねサポート完は3月予定
  - 今後、自動化技術開発を継続
    - 通信の一括化、自動並列化可能判定
- フリーウェア化
  - 今回、会員・準会員へのbinary無償配布まで
  - 段階的にオープン化

# 性能評価(1/3) OpenMPとの比較

SPEC OMP swim M-model/base



- Fujitsu PRIMEPOWER HPC2500 1.5GHz (SMPノード内)
- コンパイラ
  - OpenMP
    - Parallelnavi Fortran2.3
  - HPF
    - fhpf V1.1.3
    - Parallelnavi Fortran2.3
    - Fujitsu MPI6.1

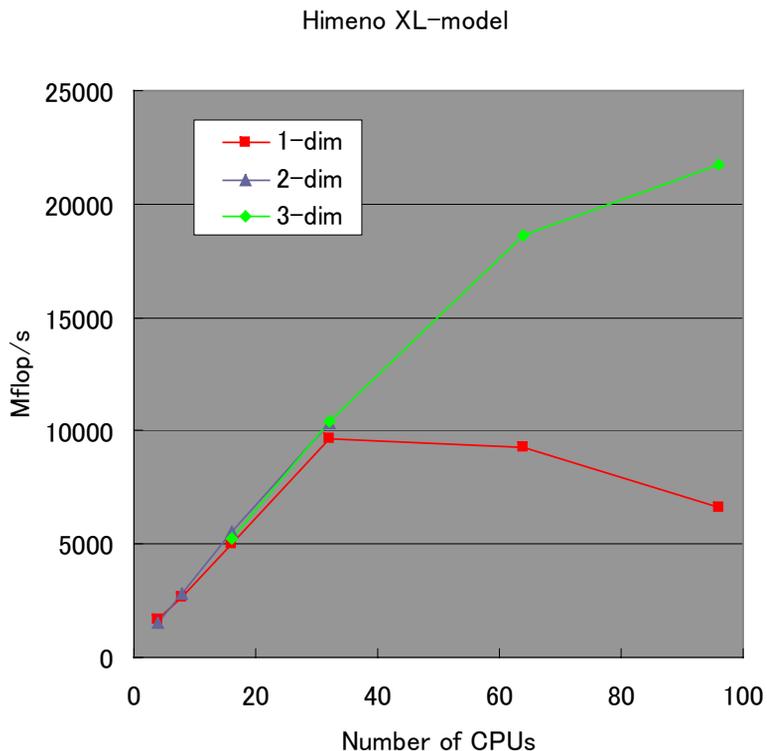
# 性能評価(2/3) 多次元分散の効果

Himeno benchmark XL-model (1024x512x512 grid)

(<http://w3cic.riken.go.jp/HPC/HimenoBMT/>)

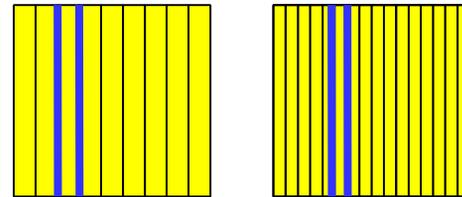
On Fujitsu PRIMEPOWER HPC2500 1.5GHz (In SMP node)

fhpf V1.2

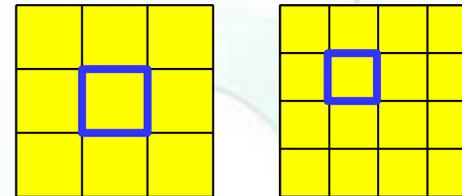


- 多次元分散で高速になる理由  
... REFLECT通信のコストオーダー

- 1次元分散:  $O(1)$



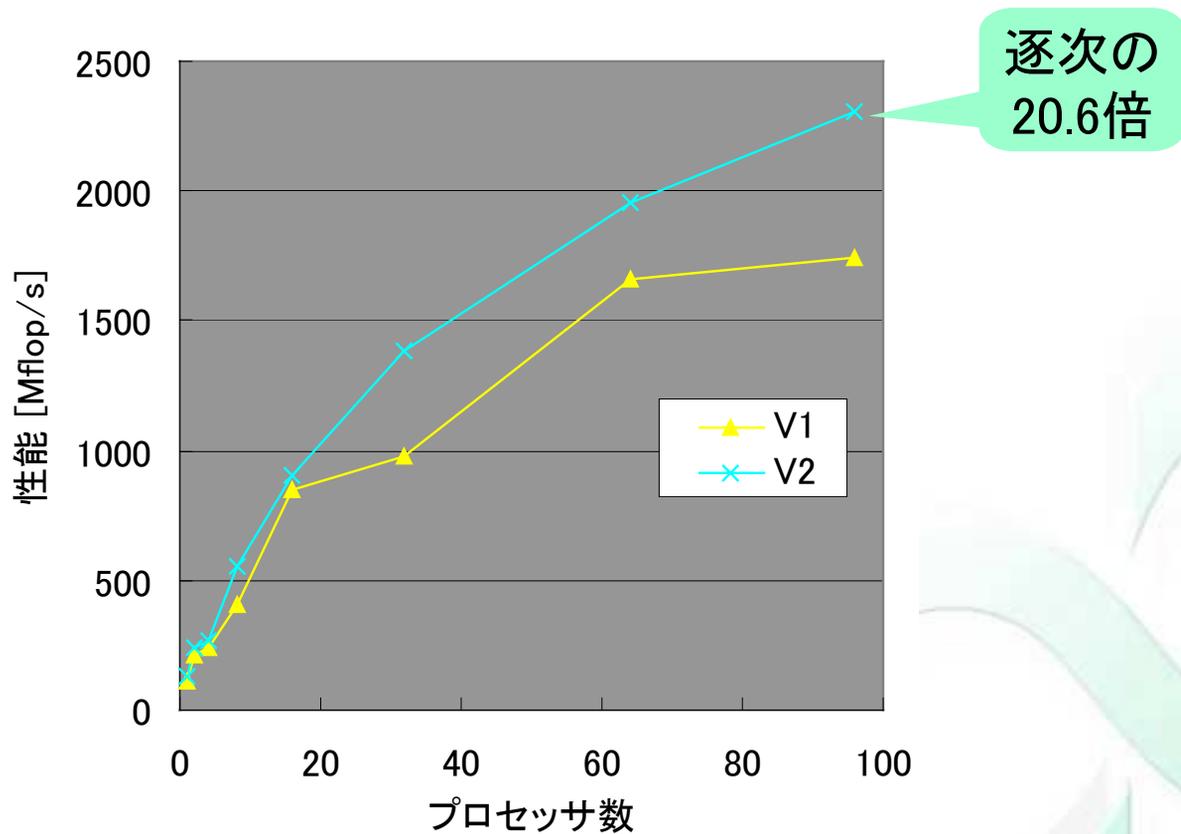
- 2次元分散:  $O(p^{-1/2})$



- 3次元分散:  $O(p^{-2/3})$

# 性能評価(3/3) 分散メモリ環境

姫野ベンチHPF版、サイズM  
Linuxブレードサーバ(Mobile Pentium III)  
fhpf V1.0, g77, LAM-MPI



逐次の  
20.6倍