



HPF実習I

HPF/ES for PC Clusterの使い方

2009年7月

NEC 第一コンピュータソフトウェア事業部

林 康晴

HPF/ES for PC Clusterの特徴

- **自動分散並列化機能**
 - ◆ 集計演算を含む並列ループの自動認識
 - ◆ シャドウ領域(袖領域)の自動割付け
- HPF2.0仕様に加え、さまざまな**拡張仕様**をサポート
 - ◆ 主要なHPF 公認拡張仕様、HPF/JA拡張仕様
 - ◆ 不規則問題(有限要素法等)向け独自拡張仕様
- HPFプログラム向け**デバッグ、チューニング**機能
 - ◆ 並列化情報リスト、並列化情報診断メッセージ
 - ◆ 様々なデバッグ用オプション
- **上級者向け機能**
 - ◆ 部分的に**MPI**を利用したチューニングが可能

HPF/ES for PC Clusterによるプログラム開発

```
DIMENSION A(100)
!HPF$ DISTRIBUTE A(BLOCK)

DO I=1, 100
  S = S + A(I)
END DO
WRITE(*,*) S
END
```

Fortranプログラム



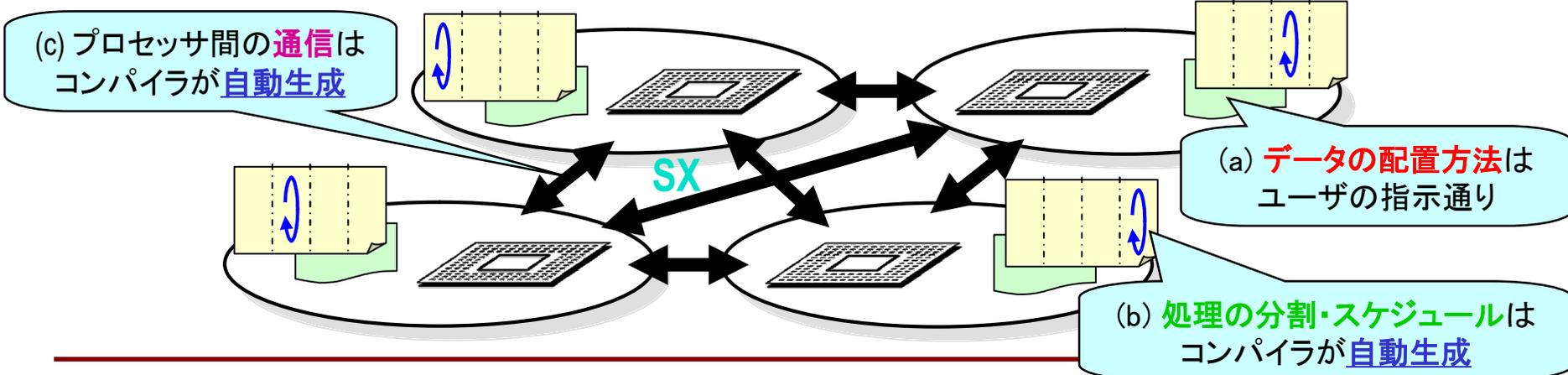
① データの配置方法をユーザが追加記述

② HPFコンパイラにかける

```
%>hpf test.hpf -o test
```

Fortran+MPIコードが自動生成され、更にFortranコンパイラにより翻訳、リンク

③ 実行 (4並列実行の場合)
mpirun_rsh -np 4 \${NQSII_MPIOPTS} ./test



コンパイル方法

■ 環境設定

◆ cshの場合

```
% source /opt/NEChpf/bin/hpfvars.csh
```

◆ shの場合

```
% source /opt/NEChpf/bin/hpfvars.sh
```

■ hpfコマンド(コンパイラ)の起動方法

```
% hpf [オプション] ソースファイル名 ...
```

■ 入力ファイル名による既定の動作

	前処理なし	前処理あり
固定形式	*.hpf *.f *.for	*.F
自由形式	*.f90	*.F90

※ 翻訳時オプションで
変更可能

※ .fの場合、-Mftnオプ
ションは使用できない。

主なコンパイルオプション1

オプション	意味
-Mextend	ソースの1行を132文字までとする(固定形式の場合に有効)
-Mfixed	ソースを固定形式として扱う(既定値)
-Mfree	ソースを自由形式として扱う
-Mpreprocess	プリプロセッサによる前処理を行う
-Minfo	診断メッセージを出力する
-Mlist2	並列化情報リストを出力する
-Msubchk	範囲外アクセス検出(デバッグ用)
-Mcommonchk	共通ブロックの手続間での矛盾検出(デバッグ用)。指定する場合は、すべての手続に指定する必要あり。
-Mnoindependent	INDEPENDENT指示文を無視(デバッグ用)。自動並列化のみよって並列化を行う。

主なコンパイルオプション2

オプション	意味
-Mscalarnew	定義されるスカラ変数を全てNEW変数と扱う。NEW変数が多数ある場合に便利。
-Mnomapnew	定義され、かつマップされていない配列を全てNEW変数と扱う。
-Moverlap=size:n	既定のシャドウ幅(シフト通信用袖領域のサイズ。既定値は、n=4)を変更する。袖領域が不要の場合、メモリの削減が可能。
-Mf90	Fortranプログラムとして翻訳する。
-Mlocal	LOCAL 手続として翻訳する。-Mf90と同時に指定すれば、 Fortran言語のLOCALモデルの手続 として扱うことができる。
-Mftn	Fortranコンパイラによる翻訳を省略する(コンパイル時間の短縮が可能)。但し、サフィックスが「.f」の中間ソースが出力されるため、ソースファイル名が「~.f」の場合は使用できない。
-Wo,"....."	バックエンドコンパイラ・リンクにオプション を渡す。

実行方法(1) ジョブの投入方法

- ジョブスクリプトに実行方法を記述し、qsubコマンドで実行する

ジョブスクリプト

```
%>qsub run.sh
```

- ジョブスクリプト中のmpirun_rshコマンドで実行プログラムを起動する
 - ◆ ジョブスクリプトrun.shの例

```
#PBS -l cputim_job=00:10:00 ← 使用CPU時間を指定する
#PBS -l memsz_job=1gb      ← 使用メモリ量を指定する
#PBS -T vltmpi             ← Voltaire MPIを指定する
#PBS -l cpunum_job=1      ← 各ノードの使用CPU(コア)数を指定する
#PBS -b 4                  ← 使用ノード数を指定する
#PBS -q PCS-B              ← 使用するキューを指定する
cd run                     ← 実行ディレクトリへ移動する
mpirun_rsh -np 4 ${NQSII_MPIOPTS} ./sample ← 実行する(4並列実行の場合)
```

実行方法(2) mpirun_rshコマンドの書き方

- mpirun_rshコマンドに実行ファイルを指定して実行する

```
mpirun_rsh [MPIオプション] 実行ファイル名 [引数等] [HPFオプション]
```

例:

```
#実行ファイルa.outを、4プロセスで並列実行  
mpirun_rsh -np 4 ${NQSII_MPIOPTS} ./a.out
```

```
#HPF実行時オプション-commmsg付きで、4プロセス並列実行  
mpirun_rsh -np 4 ${NQSII_MPIOPTS} ./a.out -hpf -commmsg
```

備考: 実行時オプション-hpf -commmsg を指定して実行すると、手続境界で通信が発生した場合、対象配列と手続名が標準エラー出力に出力される。

```
“仮引数名”: Communication occurs at procedure  
boundary PROG=“手続名” ELN=“行番号”  
Called from “手続名” ELN=“行番号”
```

```
real a(100)  
!hpf$ distribute a(block)  
call sub(a)  
end  
:  
subroutine sub(a)  
real a(100)
```

ジョブの確認方法

■ 実行状況の確認方法

```
%> qstat
```

RequestID	ReqName	UserName	Queue	Pri	STT S	Memory	CPU	Elapse	R	H	M	Jobs
12467.pcj	run.sh	hayashi	PCS-A	0	RUN -	1.71M	0.00	1	Y	Y	Y	8

ジョブID

実行状態

使用ノード数

- qstat -Jでより詳細な情報(ノード別の情報)が確認できる

■ 実行結果の確認方法

```
%> less run.sh.o12467 ← 標準出力  
%> less run.sh.e12467 ← 標準エラー出力
```

ジョブID