



HPF開発計画

2003.9.25

NEC

NECのHPFコンパイラ

■ 同一言語仕様、プラットフォーム別に4種類

□ HPF/SX V2 (SXシリーズ用)

- HPF2.0準拠、HPF/JA1.0準拠(一部)
- ベクトル化・自動並列化機能、
不規則問題向け拡張 ほか独自拡張

□ HPF/ES (地球シミュレータ用)

- HPF/SX V2 + α (並列ファイルシステム利用機能など)

□ HPF/EX (AzusA / AsAmA(通称)用)

- HPF/SX V2 - α (ベクトル化 / 共有並列化機能)

□ HPF/ES for PC Cluster (PCクラスタ用)

- 無償で利用可能(条件つき)

(注) バックエンドFortranコンパイラの違いによる差異あり。

NEC版HPFのサポートする範囲

- ポインタの分散
- 構造体成分の分散

公認拡張

- INDIRECT

HPF2.0
(コア)

- ON
- GEN_BLOCK
- SHADOW
- 再分散

- タスク並列
- etc.

NEC版HPF

- HALO
- ベクトル化指示行
- 自動並列化
- 並列I/O
- MPI I/F

- LOCAL & REFLECT
- Reduction種別
- 通信スケジュール再利用

- 非同期通信
- etc.

HPF/JA

目標: HPF/JA
フルサポート

2003年9月25日

開発計画

- 開発のサイクルは基本的に3ヶ月
 - 要望の強いもの、緊急性の高いものは臨機応変に対応
- 開発の考え方
 - 機能の品揃えよりも、性能のチューニングを優先
 - 最低限必要な機能はほぼ揃った
 - 使用実績を積み重ね、「枯れた」コンパイラを作ることが急務
 - ユーザの生の声(要望)を優先
 - 本当に必要なものから採り入れてゆく

最近のコンパイラ機能強化

- ループ変形による最適化処理の強化
 - 不要なコピーの削除
 - ループ内依存の識別
 - 並列化基準配列の選択方法改善
- 手続き呼び出しの性能改善
- 集団通信 (ALL-GATHER型通信) の高速化

コンパイラ機能強化

- ループ変形による最適化処理強化(1)
 - 不要なコピーの削除

```
INTEGER A(100),B(100)
!HPF$ DISTRIBUTE(BLOCK) ::A,B
DO I=1, N
  A(I) = B(N)+1
ENDDO
```

改善前

ループ中、常に同一要素しか参照しない配列であっても、配列全体を一時的にコピー

改善後

参照する部分だけをコピー(スカラーデータ使用)

コンパイラ機能強化

- ループ変形による最適化処理強化(2)
 - ループ内依存の識別

```
INTEGER A(100)
!HPF$ DISTRIBUTE(BLOCK) ::A
DO I=1, N-1
    A(I) = A(N)+1
ENDDO
```

改善前

参照配列の添字がDO変数を用いた一次式でなければ、最適化された並列化をしない

改善後

DO変数を用いた一次式でなくとも、依存関係がなければ最適化する

コンパイラ機能強化

■ ループ変形による最適化処理強化(3)

□ 並列化基準配列の選択方法改善

```
INTEGER NT
PARAMETER(NT=1500)
REAL SINTHT(NT), STB(2:NT-1)
!HPF$ DISTRIBUTE (BLOCK) :: SINTHT
!HPF$ ALIGN(I) WITH SINTHT (I) :: STB
DO J= NT/2+1, NT
  JOP = NT - J + 1
  SINTHT(J) = SINTHT(JOP)
  IF (J<NT) STB(J) = STB(JOP)
END DO
```

改善前の基準配列はSTB,改善後はSINTHT

改善前

基準配列の求め方に問題があり、基準配列より大きな配列がある場合にはコピー・通信が発生

改善後

基準配列の選別方法として、配列サイズを意識するよう変更

コンパイラ機能強化

- 手続き呼び出し効率改善
 - 同一分散の認識

```
REAL*8 A(N,N,N)
!HPF$ TEMPLATE T(N)
!HPF$ DISTRIBUTE (BLOCK) :: T
!HPF$ ALIGN (*,*,I) WITH T(I) ::A
CALL SUB(A,N)
END
SUBROUTINE SUB(A,N)
INTEGER N
REAL*8 A(N,N,N)
!HPF$ TEMPLATE T(N)
!HPF$ DISTRIBUTE (BLOCK) :: T
!HPF$ ALIGN (*,*,I) WITH T(I) ::A
```

改善前

関数の引数が整列されているデータの場合、親子間の分散が一致してもコピー・通信が発生

改善後

親子間の分散が一致しているときにはコピー・通信をしないよう変更

コンパイラ機能強化

- 集団通信 (ALL-GATHER通信)の高速化
 - 通信スケジュールの改善

```
        INTEGER A(100,100,100)
!HPF$ DISTRIBUTE(*,BLOCK,*) ::A
        .....
!HPF$ REDISTRIBUTE(*,*,*) ::A
```

改善前

分散から非分散へ配列
を変更する場合の通信
スケジュールが非効率

改善後

通信スケジュールの見
直しを実施し高速化

コンパイラ機能強化

■ 性能評価結果

- 測定コンピュータ: NEC内試作機
- MPIプロセス数: 8
- 測定プログラム: 各評価用TP

性能評価結果	旧版 (s)	新版 (s)	性能比
ループ変形 (TP1)	4.37	0.201	21.7
ループ変形 (TP2)	4.75	0.232	20.5
ループ変形 (TP3)	1.85	0.0152	121
外来手続き呼び出し	13.75	0.444	30.9
ALL-GATHER通信	2.101	1.511	1.39

現在ご要望いただいている機能


- コンパイラ生成コード／ランタイムの改良
 - 割り付け配列を多次元イメージのまま扱う (→2004/3)
 - 配列へのリダクション演算の高速化 (→2003/12)
 - Fortranコンパイラと連携し、ベクトル・共有並列化を促進 (→2004/3)
 - 取り扱えるF90/SX、F90/ES指示行の範囲拡大
 - HPFコンパイラが積極的にこれら指示行を出力
 - インライン展開 (→未定)

現在ご要望いただいている機能

■ コンパイル時メッセージの改良

□ コンパイラが選択したHOME配列の出力 (→2003/9)

```
1:      real a(10,10),b(10,10)
2: !hpf$ distribute b(block,block)
3:
4:      a = 0
5:      b = 1
6:
7:      do i=2,9
8:          do j=2,9
9:              a(j,i) = b(j,i) + b(j-1,i)+b(j+1,i)+b(j,i+1)+b(j,i-1)
10:         enddo
11:     enddo
12:
13:     write(*,*)a
14:     end
```

- 
- 7, Independent loop
Array a not aligned with home array; array copied
Independent loop parallelized : **b(j,i)**
communication is generated: array copy
 - 8, Independent loop
Independent loop parallelized : **b(j,i)**

現在ご要望いただいている機能

■ ライブラリ等

□ 並列乱数 (→2004/3)

- メルセンヌ・ツイスター法
- 合同法 など実証実験中

□ ソート (→2004後半以降)

□ パック／アンパック (→2004後半以降)

- あらゆる場合(分散状態ほか)に効率を求めるのは困難なため、選別が必要

■ その他

□ コンパイル時メッセージの解説 (→2003/9)

- マニュアルに記述を追加(オンライン・図書)