並列コンピュータ TX7/i9610 の自動並列化コンパイラ

左近 彰一 [1]　山本 秀喜 [2]

1. はじめに

本稿では, スカラ並列コンピュータ TX7/i9610 システムで使用できる Fortran コンパイラの自動並列化機能について, その概要を説明します.

2. 自動並列化とは

Fortran コンパイラの自動並列化機能とは, コンパイラが Fortran ソースプログラムから自動的に並列実行可能なループや文の集まりを抽出し, 並列処理用の目的プログラムを生成するものです. 自動並列化を使用することにより, ユーザはソースプログラムを修正することなく複数の CPU を利用してプログラムの実行時間を短縮することができます. また, コンパイラが自動的に並列化可能かどうかを判断することが難しい場合でも, ユーザが簡単な指示行をソースプログラムに加えることによって並列化することもできます. 指示行は, スーパーコンピュータ SX-7 および SX-7C の Fortran コンパイラである FORTRAN90/SX との互換性を備えており, SX シリーズとソースプログラムを共通化して並列化を行うことができます.

3. 自動並列化の対象

Fortran の DO ループ, および配列構文が自動並列化の対象です.

表1. Fortran 自動並列化対象

対象となる構造	DO ループ, 配列構文
対象となるデータの型	整数型, 論理型, 単精度, 倍精度, 四倍精度の実数型･複素数型, (文字型は対象外)
ループ中に許される文	代入文, IF 文, GOTO 文, CASE 構文, WHERE 文 (call 文, 入出力文は対象外)
対象となる演算	加減乗除算, 論理演算, 関係演算, 型変換, 組込み関数 (ユーザ関数呼び出しは対象外)

注:call 文やユーザ関数を含む場合でもそれらがインライン展開されれば並列化される場合があります.

[1] 日本電気株式会社 第一コンピュータソフトウェア事業部
[2] NECシステムテクノロジー株式会社 サーバソフトウェア事業部

4. 並列化の基本方針

プログラム中に現れるループが多重ループである場合は，基本的に最も外側のループが並列化されます．これは，そうすることよって同期処理などの並列実行のオーバーヘッドの回数を最も減らすことが出来るためです．そして，最内側ループはソフトウエアパイプライニング（SWP）されます．もし，元のループが一重ループである場合は，そのループを二重ループに分割し，出来た外側ループを並列化，内側ループを SWP 化します．

配列構文の場合，ループに展開して同様に並列化されます．

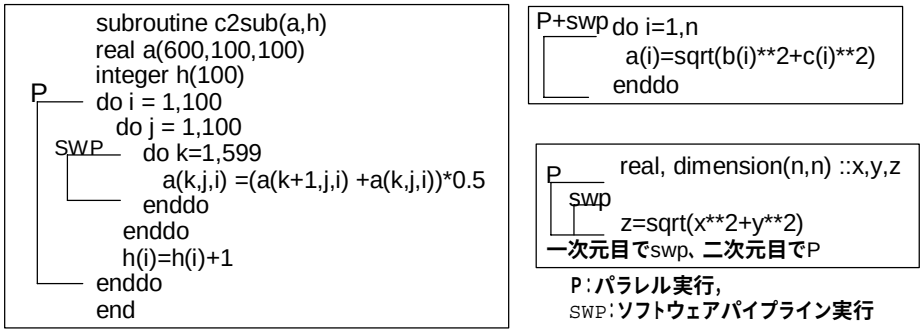

図1　多重ループ，一重ループ，配列構文の並列化

5. 並列化条件

ループが自動並列化できるためには，以下の条件を満たす必要があります．これらの条件が満たされていない場合，ループは並列化できません．

ループ構造

- ループからの途中飛び出し，飛び込みがない．

ループ内データ依存関係（以下の条件のどれかを満たす）

- 配列要素の定義，参照がループの繰り返しに閉じている．
- 配列要素が参照のみである．
- 整数型スカラ変数がインデックス変数（i=i+k の形）である．（ただし if 文の下で定義されていないこと）
- スカラ変数で，ループ内の出現が定義後参照である．
- スカラ変数が，総和・内積の形である(S=S+A(i)*b(i))
- データ依存関係がコンパイラには不明であるが，実際にはループの繰り返しをまたがる依存関係がない場合，ソースプログラム中に並列化指示行(parallel)を指定すること

により, 並列化可能となる.

いくつかの例を示します.

[例 1] ループからの飛び出しがある ： 並列化できない.

```
      do j = 1,n
        do i = 1,n
          a(i,j) = sqrt( b(i,j) )
          if (a(i,j) .ge. del ) go to 100
          if (c(i,j) .ge.0 ) then
            b(i,j) = c(i,j) - a(i,j)
          else
            b(i,j) = c(i,j) + a(i,j)
          end if
        end do
      end do
100 continue
```

[例 2] 配列要素(B)の定義参照がループの１つの繰り返しに閉じていない ： 並列化できない

```
      do i=1, n
        a(i) = b(i+1)
        b(i) = c(i)
      end do
```

[例 3] スカラ変数(T)の定義と引用が１つの繰り返しに閉じていない ： 並列化できない

```
      do i=1, n
        c(i) = t
        t = a(i)+c(i)
      end do
```

[例 4] スカラ変数 (T) が1つの繰り返しに閉じている ： 並列化できる

```
      do i=1, n
        t = b(i)
        c(i) = t
```

```
end do
s = t
```

6. コンパイラオプション

自動並列化関係のコンパイルオプションには以下のものがあります.

-parallel

- 自動並列化を行う指定です.

-par_report[1|2|3]

- 自動並列化メッセージの出力を制御します.
  -par_report1(既定値) : 並列化されたループにメッセージを出力します
  -par_report2:並列化されたループとされていないループにメッセージを出力します
  -par_report3: 2 に加えて並列化不可要因 (依存関係等) を表示します

-par_threshold[n]

- 自動並列化の性能向上しきい値 (確実度) を設定 (n は 0 から 100 まで) します.
  - 0: 性能向上が見込めなくても, 並列化可能なループは常に並列化します
  - 100 (既定値) : 性能向上が確実なループのみ並列化します

-force_parallel

- 強制並列化を行う指定です. 強制並列化指示行が指定されたループを並列化します. 通常は自動並列化を行うオプション –parallel と同時に指定します. -parallel オプションを同時に指定しない場合, 強制並列化指示行の指定が無いループは並列化されません. また, このオプションは-openmp オプションと併用できません.

[例 6] 自動並列化メッセージ

```
      subroutine sub(a,b,c,n,l)
       real a(n),b(n+1),c(n),d(n)
       do k=1, n
         a(k) = b(k+l)
         b(k) = c(k)
       end do
       end
$f95 -parallel -par_report3 sub.f
    external subroutine SUB
    procedure: sub
```

```
serial loop: line 3
    anti data dependence assumed from line 4 to line 5, due to "b"
    flow data dependence assumed from line 5 to line 4, due to "b"
```

7. 指示行

指示行は，ソースプログラムの DO ループの直前に記述し，ループの並列化を制御するものです．以下の指示行がサポートされています．

- !dir$ ivdep

  データ依存関係がコンパイラに不明で並列化できないループを並列化可能とします．
- !dir$ parallel

  多重ループの場合，指定のあるループが並列化可能なら，そのループを並列化します．
- !dir$ noparallel

  ループを並列化しません．
- !dir$ parallel do

  あるループを必ず並列化したい場合に指定します（強制並列化指示）．コンパイラはデータ依存関係や性能向上しきい値などのチェックを行わず，指定のループを並列化します．本指示行を有効にするためには、コンパイルオプション -force_parallel の指定が必要です．本指示行は OpenMP の!$omp parallel do 指示行とほぼ同じ意味を持ちますが，OpenMP 指示行との違いはコンパイルオプション -parallel を併用して指定した時，手続き内の他のループが並列化対象となることです．

[例7] ivdep 指示行

```
      subroutine sub(a,m,n)
      dimension a(n),m(n)
!dir$ ivdep
      do i=1,n
        a(m(i))=a(m(i))+1
      enddo
      end
```

指示行が無いと，コンパイラは m(i)に重なりが無いことが分からないので，並列化しません．し

かし，実際にはm(i)に重なりがないことが分かっている時は，ivdep指示行を指定することにより並列化されます．

[例 8] parallel 指示行

```
        subroutine sub2(a,n,m)
        real*8 a(n,m)
        do j=1,m
!dir$ parallel
          do i=1,n
            a(i,j)=sin(a(i,j))
          enddo
        enddo
        end
```

指示行がない場合外側ループであるdo jで並列化されますが，たとえばループ長mが極端に短い場合は，do j の内側で並列化したほうが性能が良い場合があります．そのような場合に，parallel 指示行を内側ループに指定します．

[例9] parallel do 指示行(強制並列化)

```
        subroutine sub2(a,n,m)
        real*8 a(n,m)
        do j=1,m
!dir$ parallel do
          do i=1,n
            a(k(i),j)=a(k(i),j)+1
          enddo
        enddo
        end
```

do i のループで強制並列化します．コンパイラはデータ依存関係のチェックを行わないので，並列化しても問題がない場合にだけ使用できます．なお，コンパイルオプション-force_parallel の指定が必要です。

8. 環境変数

自動並列化プログラムでは, 以下の環境変数が使用できます. これらは OpenMP 並列処理と共通です.

- OMP_NUM_THREADS : スレッド数を指定 (既定値:利用可能 CPU 数)
- OMP_SCHEDULE:ループのスケジューリング (既定値:static)
- KMP_STACKSIZE:各スレッドのスタックサイズを指定(既定値:4MB)
- KMP_BLOCKTIME: パラレルリージョン終了後, sleep するまでの時間をミリ秒単位で指定(既定値は 200ms)
- KMP_LIBRARY: turnaround | throughput
  - turnaround: 占有環境でのターンアラウンド優先. 待ち状態で sleep せず busy loop しつづけます.
  - throughput:共用環境でのスループット優先. KMP_BLOCKTIME で指定した時間待つとタスクが sleep します (既定値) .

9. OpenMP との関係

OpenMP と自動並列化は, 一つのプログラム中で共存可能です. どちらの並列化が有効になるかはコンパイルオプションの指定によって, 手続き (関数, サブルーチン) 単位に決まります.

- コンパイルオプション -openmp のみ指定

  ソースプログラムの OpenMP 指示行が有効となります.
- コンパイルオプション -parallel のみ指定

  ソースプログラムの OpenMP 指示行は無視され, 自動並列化が行われます.
- -openmp と -parallel の両方指定

  手続き内に OpenMP 指示行がある場合, それが有効となり, 自動並列化は行われません. 手続き内に OpenMP 指示行がない場合は, その手続きは自動並列化されます.

また強制並列化との関係は以下の通りです.

- コンパイルオプション-force_parallel のみ指定

  強制並列化指示行(!dir$ parallel do)が指定されたループだけが並列化されます.
- -force_parallel と -parallel の両方指定

  強制並列化指示行を指定されたループは並列化され, その他のループは並列化可能であれば自動並列化されます.
- -force_parallel と -openmp の両方指定

このオプションの組み合わせは指定できません.

10. FORTRAN90/SX との互換性

スカラ並列コンピュータのコンパイラは, スーパーコンピュータ SX-7 および SX-7C の Fortran コンパイラである FORTRAN90/SX の指示行との互換機能を持っており, ＳＸの以下の形式の指示行を使用することが出来ます.

!CDIR コンパイラ指示オプション

!CDIR の代わりに FORTRAN77/SX の旧形式 (*VDIR, *PDIR, *ODIR) 指示行も使用できます. コンパイラ指示オプションには, 以下のものが使用できます.

- 並列コンピュータで指定できる指示オプション(ivdep, parallel, noparallel, parallel do)
- nodep (ivdep と等価)
- loopcnt=ループ長

この!CDIR 形式の SX 互換の指示行機能は, コンパイラオプション –sx_directive[-]で有効 (既定値) /無効を切り替えることが可能です.

11. おわりに

スカラ並列コンピュータ TX7/i9610 コンパイラの自動並列化機能と使用方法について簡単にご紹介いたしました. 自動並列化によるプログラム実行の高速化の一助になれば幸いです.

参考文献

[1] 左近彰一, 山本秀喜: TX7/AzusA Linux コンパイラの自動並列化機能, SENAC Vol.35 No.1, (2002)