TX7/AzusA Linux コンパイラの自動並列化機能

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　　左近　彰一
山本　秀喜

概要

TX7/AzusA Linux Fortran および C/C++コンパイラの自動並列化機能の利用方法について説明する。

1. はじめに

TX7/AzusA システムは 1 ノード 16 個の CPU で構成されており、これらの CPU を同時に利用してプログラムの実行時間を短縮化する並列処理を行うことができます。AzusA での並列処理機能としては、OpenMP や MPI に加えて、自動並列化がありますので、以下にその使用方法を説明します。

2. 自動並列化とは

自動並列化とはコンパイラが並列化可能なループを検索し、そのループを自動的に並列化する機能です。ユーザは、ソースプログラムを修正する必要が無く、手軽に並列化を行うことが出来ます。また、コンパイラが並列化の可否を自動的に判断できないループに対しても、簡単な指示行を挿入するだけで並列化の指定を行うことが出来ます。また 1 つのプログラムでサブルーチン・関数単位に OpenMP との共存が可能であり、きめ細かな性能改善を行いたいサブルーチンに対してだけ OpenMP で並列化を行い、他の部分は自動並列化を使用するといった使い方も出来ます。

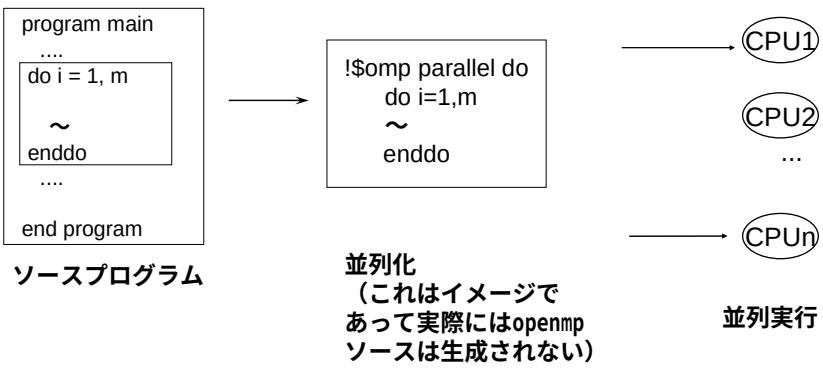

図 1　自動並列化

3. 自動並列化の対象

Fortran の場合、DO ループが自動並列化の対象です。C/C++の場合は、for 文・while 文・do 文でループ長が入り口で決定できるものが対象になります。

表１．Fortran 自動並列化対象

対象となるループ	DO ループ
対象となるデータの型	整数型、論理型、短精度および倍精度の実数型・複素数型（四倍精度実数、四倍精度複素数、文字型は対象外）
ループ中に許される文	代入文、IF 文、GOTO 文、CONTINUE 文、CASE 構文 （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

注：複素数型の組込み関数は対象外です。

表 2. C/C++ 自動並列化対象

対象となるループ	for、while、do ループ（ループ長が決定できるもの）
対象となるデータの型	文字型、すべての整数型・単精度・倍精度の実数型
ループ中に許される文	代入、if、while、do、for、goto 文等 （switch 文と関数呼び出しは対象外）
対象となる演算	加減乗除算、論理演算、関係演算、シフト、型変換、数学関数

注：条件演算子 ”?:” は対象外です。

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、並列化が可能なループのネストの内、最も外側のループが並列化されます。これは、粒度が最も大きくなり、並列化のオーバーヘッドができるだけ小さくなるためです。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、ループが一重ループである場合は、そのループを２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。

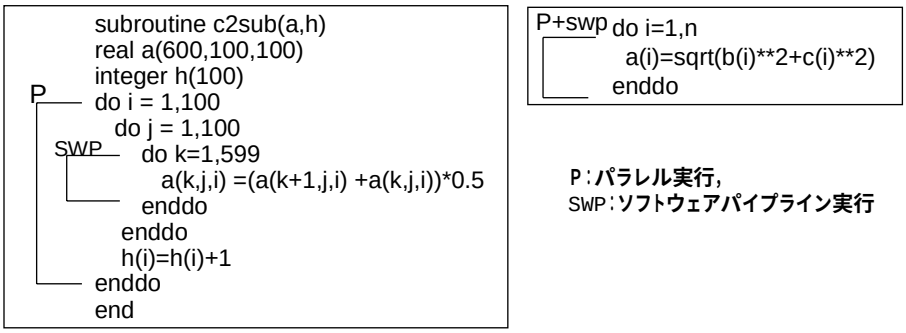

図２　多重ループと一重ループの並列化

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（以下の条件をすべて満たす）

- ループからの途中飛び出し、飛び込みがない。
- ループの繰り返し回数がループの入り口で決まっている

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。
- スカラー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）
- スカラー変数で、ループ内の出現が定義後参照である。(この変数は private 化される）。ループを出た後、参照されていてもよい。
- データ依存関係が不明な場合
    1. 実際にはループの繰り返しをまたがる依存関係がないことが分かっている場合、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。
    2. C のポインタ変数が使用されている場合、通常は依存関係不明で並列化できないが、実行時に重なりをチェック出来る場合にはチェックするコードを自動的に生成し、並列化する。

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 配列要素の定義参照がループ内に閉じていない：並列化できない

```
      DO K=1, N
        A(K) = B(K+1)
        B(K) = C(K)
      END DO
```

例 3 スカラー変数の定義と引用が閉じていない：並列化できない

```
       DO I=1, N
         C(I) = T
         T = B(I)
      END DO
```

例 4 スカラー変数ループを出た後で参照されている：並列化できる

```
      DO I=1, N
         T = B(I)
         C(I) = T
      END DO
      S = T
```

例 5 C のポインタ。実行時に重なりをチェックする：並列化できる

```
      sub(double *a, double *b, int n)
           int i;
           for (i=0; i< n; i++)
              a[i]=b[i]+1.0
```

6. 制限事項

以下のものがループ内にあるとループは並列化されません。これは将来のリリースで解除する予定です。

- 総和（リダクション）計算
- Fortran の複素数型組込み関数、指数が整定数(-10～10)でないべき乗
- C の条件演算子
- 添字式中の仮引数である変数の参照、ループ内で仮引数である変数の定義、仮引数であるインデックス変数の使用
- 8 バイト整数型のインデックス変数

7. コンパイルオプション

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[0|1|2|3]

- 自動並列化メッセージの出力を制御します。

  -par_report0: メッセージを出力しません

  -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]

- 自動並列化のしきい値（確実度）を設定（n は 0 から 100 まで）します
- 0: 並列化可能なループは常に並列化します
- 100：性能向上が確実なループのみ並列化します
- 75（既定値）：高速化する確率が 75%以上のループを並列化します

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_
```

```
      Reason: Assumed ANTI dependence for "b" from line 4 to line 5.
      Reason: Assumed FLOW dependence for "b" from line 5 to line 4.
    SERIAL LOOP: Line 3
```

このメッセージは、配列Bのデータ依存関係に関するものです。最初のメッセージは、4行目のBと5行目のBの間に反依存関係(参照の後定義) があるとコンパイラが仮定したこと、２番目のメッセージは、5行目のBと繰り返しをまたいだ4行目のBの間にフロー依存関係(定義の後参照)があると仮定したことを示しています。このようにループの繰り返しをまたいだデータ依存関係の可能性がある場合、ループは並列化されません。

メッセージに表示される依存関係の種類は、フロー依存関係(FLOWdependence)、反依存関係(ANTI dependence)以外に、出力依存関係(OUTPUTdependence)があります。これはデータが定義後また定義される関係を表します。

8. 並列化指示行

並列化指示行は、ソースプログラムの DO ループ(Fortran)または for ループ等(C/C++)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。

Fortran

- !dir$ parallel
  多重ループの指定のループが並列化可能なら、並列化する。
  データ依存関係がコンパイラに不明な場合、並列化可能とする。
- !dir$ noparallel
  ループを並列化しない。

C/C++

- #pragma parallel
  多重ループの指定のループが並列化可能なら、並列化する。
  データ依存関係がコンパイラに不明な場合、並列化可能とする。
- #pragma noparallel
  ループを並列化しない。

例７　並列化指示行

```
            subroutine sub(a,m,n)
            dimension a(n),m(n)
      !dir$ parallel
```

```
      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、並列化しませんが、実際には m(i)に重なりがないことが分かっている時は、並列化指示行を指定することにより並列化されます。

例 8　並列化指示行

```
      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。

9. 環境変数

以下の環境変数が使用できます。これらは OpenMP と共通です。

- OMP_NUM_THREADS : スレッド数を指定（既定値：構成 CPU 数）
- OMP_SCHEDULE：ループのスケジューリング（既定値：static）
- KMP_STACKSIZE：スタックサイズを指定。
- 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します。既定値は 1000　（1 秒）。1s と秒指定も可能。
- KMP_LIBRARY: serial | turnaround | throughput
  - serial: シリアル実行用ライブラリ
  - 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）

10．OpenMP と自動並列化の関係

OpenMP と自動並列化は、プログラム中で共存可能です。手続き単位にどちらを使用するかは以下のように選択されます。

- コンパイルオプション -openmp のみ指定
  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
  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自動並列化されます。

11．今後の強化項目

NEC では今後も継続して自動並列化機能および性能の強化を行っていく予定です。強化予定項目は下記のものがあります。

制限の解除
– 仮引数による並列化制限の解除
– Fortran べき乗および複素数型組み込み関数の並列化
並列化対象の拡大
– 添字式中の変数による依存関係が不明な場合、実行時にチェック
– ループ融合、ループ入れ換えして並列化
– 総和の並列化

12．おわりに

TX7/AzusA Linux コンパイラの自動並列化機能について簡単にご紹介いたしました。自動並列化の利用の一助になれば幸いです。