SR8000 性能モニター機能の利用法

（株）日立製作所

１．はじめに

SR8000 は、図 1.1 に示すように複数のノードを多次元クロスバーネットワークで接続した並列計算機です。1 つのノードは 8 個の RISC プロセッサー（Instruction Processor：本文では IP と略し、8 個のそれぞれを IP0、IP1、…、IP7 と書きます）と主記憶メモリーから構成されています。

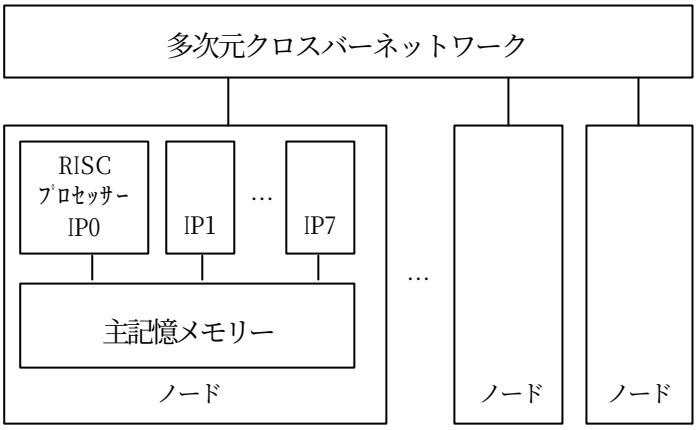

図 1.1　SR8000 の構成

1 つのノードに注目すると、SR8000 は以下に示す 2 つの大きな特徴を持っています。

(1) 擬似ベクトル化機能

データーを主記憶メモリーから IP 内のレジスターやキャッシュメモリーに先読みすることで、擬似的なベクトル処理をする機能です。この機能により、Load 命令によって後続の演算命令が待たされることがなくなります。

(2) 要素並列化機能

1 つのノード内にある 8 個の IP が DO ループやスレッド（手続き呼び出し等の文の集まり）を分担処理する機能です。要素並列化処理をする部分を持つプログラムの場合、親プロセッサー（IP0）が逐次処理部と要素並列処理部を、子プロセッサー（IP1 から IP7 の 7 個）が要素並列処理部をそれぞれ担当することになります（図 1.2）。要素並列処理部は 8 個の IP で処理することになるので要素並列処理部分がプログラム中に占める割合が大きいほど計算時間は短縮されます。

これら 2 つの機能はコンパイルオプションの指定で利用可能です。FORTRAN コンパイラーでのオプションの例を表 1.1 に示します。

図 1.2　逐次処理部と要素並列部での各 IP の動作

表 1.1　擬似ベクトル化と要素並列化機能を利用するオプションの例

指定するオプション	最適化レベル	擬似ベクトル化	要素並列化
-O4 -pvec -parallel=3 -procnum=8	4	-pvec の指定により ON	-parallel=3 で並列化レベル 3 -procnum=8 で 8 つの IP で実行
-Os -procnum=8	S	-Os で自動的に ON	-Os により並列化レベル 3 -procnum=8 で 8 つの IP で実行
-Oss -procnum=8	SS	-Oss で自動的に ON	-Oss により並列化レベル 4 -procnum=8 で 8 つの IP で実行

表 1.1 のコンパイルオプションを指定することにより計算時間の短縮がはかれますが、更なる性能向上のために、

・計算時間の長い部分（処理の重い部分）を効率よく特定する。

・実際に擬似ベクトル化と要素並列化機能が有効に働いているか確認する。

などを行い、必要に応じて対策することになります。そこでここでは、性能向上対策を効率よく支援する SR8000 の性能モニターを紹介します。

２．SR8000 性能モニター

SR8000 は、プログラム実行に要した CPU 時間、実行命令数、浮動小数点演算数をはじめとする実行時の各種性能情報を取得する為のハードウェア機構を有してます。

SR8000 性能モニターは、このハードウェア機構を介して実行時の性能情報を読み取り、集計処理を行った後、実行性能情報をファイル出力します。

性能モニターは、FORTRAN コンパイラーの機能の一部です。コンパイル・リンク時に特定のオプションを指定することにより、コンパイラーが性能情報を読み取るモニター関数をオブジェクトコードに自動挿入し、モニター関数が読み込んだ情報を集計してファイル出力する処理を追加します。こうして作成したロードモジュールを実行することにより、プログラム終了時に、実行性能情報が集計処理されたファイルが出力されます。

３．性能モニターの主な特徴と活用法

(1) 特徴

ソースコードを修正する必要がなく、コンパイル・リンクオプションの追加のみで利用可能です。また、性能モニター機能を使用しても、コンパイル時の最適化レベルが変わる事はありません。

(2) 必要とするオプション

プログラムに性能モニター関数を自動挿入するオプションとして、表 3.1 に示す 2 種類を提供しています。

表 3.1　性能モニター関数自動挿入オプション

モニター関数の挿入位置	コンパイルオプション
関数／サブルーチン単位に挿入	-Xfuncmonitor
関数／サブルーチン単位、さらに要素並列化単位に挿入	-Xparmonitor

オプション-Xfuncmonitor を指定した場合の、関数／サブルーチンに対するモニター挿入位置は、図 3.1 に示すように処理の先頭、および末尾（RETURN 文直前）です。性能情報はモニター関数によって得られた測定結果 a と測定結果 b の差分をもとに出力しています。これにより、関数／サブルーチン単位に、CPU 時間、浮動小数点演算性能などの情報が得られます。

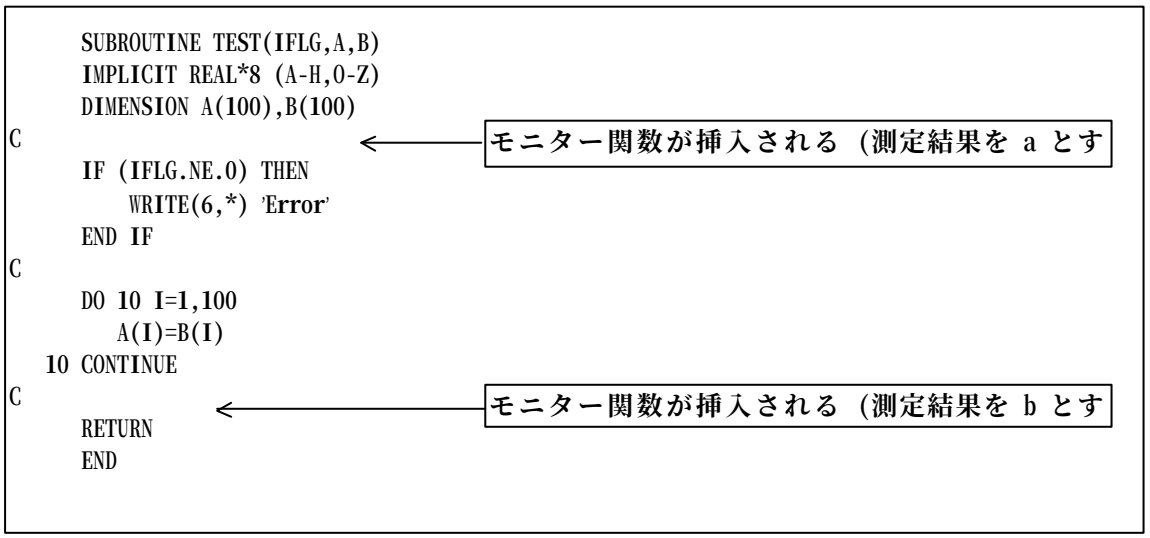

図 3.1　関数／サブルーチンに関するモニター挿入位置

要素並列化単位に対するモニター挿入位置は、図 3.2 に示すように要素並列化をしている DO ループの直前及び END DO（CONTINUE）文直後です。

これにより、要素並列化単位に、CPU 時間、浮動小数点演算性能などの情報が得られます。

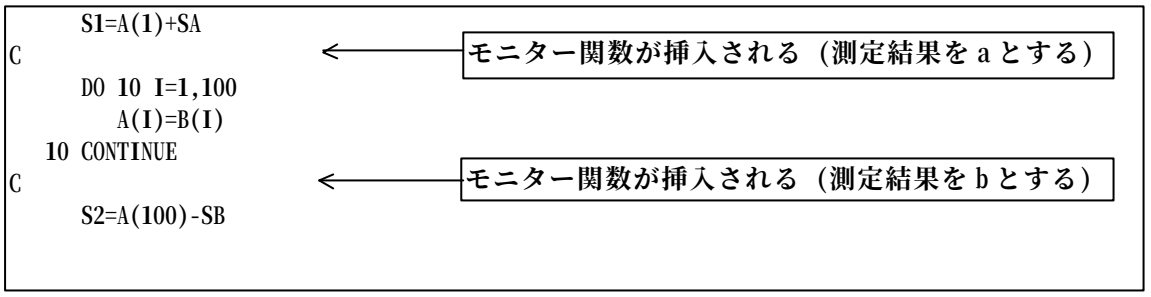

図 3.2　要素並列化単位に関するモニター挿入位置

また、性能モニター機能はコンパイルオプションで指定するので、ソースファイル単位に指定できます。

例えば、ソースファイル A.f 中に MAIN プログラムが記述され、ソースファイル B.f にサブルーチン SUB1 が、ソースファイル C.f 中にサブルーチン SUB2 と SUB3 の記述がされているとします。この場合、コンパイルオプションと、そのオプションを指定するファイルの組み合わせにより、表 3.2 に示すような情報を得ることができます。

表 3.2　コンパイルオプションをソースファイル単位に指定する例

オプションの指定	得られる情報
A.f に-Xfuncmonitor を指定	プログラム（プロセス）単位の性能情報
A.f、B.f、C.f に-Xfuncmonitor を指定	サブルーチン単位の性能情報
A.f、C.f に-Xfuncmonitor を指定 B.f に-Xparmonitor を指定	サブルーチン単位の性能情報 SUB1 は要素並列化単位の性能情報

コンパイルオプションについては、次のような使い分け方をお奨めします。

まず､関数／サブルーチン単位の性能情報を取得するオプション-Xfuncmonitor を使うことによって、各関数／サブルーチンの CPU 時間を取得して、この値に注目し、処理の重い部分を特定します。次に、要素並列化単位まで性能情報を取得するオプション-Xparmonitor を指定することで、要素並列処理部での IP 間での負荷分散が適当であるか等を調べます。この結果によりソースファイルを書き換えたり、ディレクティブを挿入するなどして問題を解決します。

(3) 採取可能な性能情報

性能モニターにより得られる性能情報の主な項目を表 3.3 に示します。

表 3.3　採取可能な性能情報

情報項目名称	内容
Execution count	関数／サブルーチンが実行された回数
User Time [sec]	ユーザーCPU 時間[秒]
LD/ST	LoaD と STore 命令数
Instruction count	実行命令数
Floating-point arithmetic count	浮動小数点演算実行数
MIPS	1秒間に実行される命令数を 100 万命令単位で示したもの（Million Instructions Per Second）
MFLOPS	1 秒間に実行される浮動小数点演算を 100 万回単位で示したもの（Mega FLoating point Operations Per Second）

(4) 出力ファイル

実行性能情報を出力するファイルの形式はテキスト形式と CSV 形式の 2 種類を指定することができます。

ファイルの出力形式は、プログラム実行時に環境変数 APDEV_OUTPUT_TYPE を指定することにより選択します。 環境変数 APDEV_OUTPUT_TYPE を設定せずに、ロードモジュールを実行した場合のファイルの出力形式はテキスト形式になります。

表 3.4　出力ファイル形式の選択方法（C シェルの場合）

環境変数の設定法	出力される形式とファイル名
setenv APDEV_OUTPUT_TYPE TEXT	テキスト形式 pl_wwwww_xxxx_yy.txt
setenv APDEV_OUTPUT_TYPE CSV	CSV 形式 pl_wwwww_xxxx_yy.csv
setenv APDEV_OUTPUT_TYPE BOTH	テキスト形式と CSV 形式両方 pl_wwwww_xxxx_yy.txt と.csv

wwwww はロードモジュール名、xxxx はプロセス番号、yy はノード番号です。

テキスト形式の出力ファイルは、エディター等で参照が簡単にできる為、以下の場合に使用するとよいでしょう。

- ロードモジュールが比較的少数の関数／サブルーチンから構成される場合
- 実行性能情報を採取する箇所を局所化した場合（コンパイルオプション指定の有無をソースファイル毎に切り替えてロードモジュールを作成した場合です。表 3.2 を参照してください。）

CSV 形式の出力ファイルは、パソコンの表計算プログラム等によって内容を参照すると、データーの加工や編集が簡単にできる為、以下の場合に使用するとよいでしょう。

- 関数、サブルーチン、要素並列化箇所が多数ある場合で、ユーザーCPU 時間や実行回数等の測定された項目をキーとしたソートをする場合
- 各性能情報に関して、グラフ作成をする場合

４．性能モニターの使用方法

以下、簡単な例題プログラムを用いて、性能モニターを使った実行性能情報の採取方法を紹介します。

(1) 例題プログラム sample.f の紹介

図 4.1 に示すような MAIN 部分とサブルーチン KEISAN 部分からなる sample.f を例として取り上げて説明します。サブルーチン KEISAN 部分は DO 10、20、30 の 3 つの DO ループからなります。このうち、ループ DO 20 は配列 A についてループの繰り返し間に依存があるので、要素並列化が適用されません。

```
 1        PROGRAM MAIN
 2        IMPLICIT REAL*8 (A-H,O-Z)
 3  C
 4        WRITE(6,*) 'Sample Program Start'
 5        CALL KEISAN(SUM)
 6        WRITE(6,*) 'Sum of A =',SUM
 7        WRITE(6,*) 'Sample Program End'
 8  C
 9        STOP
10        END
11  C
12        SUBROUTINE KEISAN(SUM)
13        IMPLICIT REAL*8 (A-H,O-Z)
14        DIMENSION A(400000),B(400000),C(400000)
15  C
16        DO 10 I=1,400000
17           A(I)=1.0D0*I
18           B(I)=2.0D0+I
19           C(I)=A(I)+B(I)
20     10 CONTINUE
21  C
22        DO 20 I=1,399999
23           A(I+1)=A(I)+C(I)
24     20 CONTINUE
25  C
26        SUM=0.0
27        DO 30 I=1,400000
28           SUM=SUM+A(I)*B(I)
29     30 CONTINUE
30  C
31        RETURN
32        END
```

図 4.1　例題プログラム sample.f

(2) 使用例１：関数／サブルーチン単位の情報を採取する場合

関数／サブルーチン単位に性能情報を取得する場合は、３章で説明したように、コンパイルオプション-Xfuncmonitor を追加してコンパイルし、実行します。

処理の重い関数／サブルーチンを特定するために各関数の実行時間を採取することを目的とするオプションです。

以下、コンパイル、リンク、プログラム実行方法、および、出力される性能情報ファイルの見方について、順を追って説明します。

(2-1) コンパイル方法

オプション-Xfuncmonitor を追加してコンパイルします。

なお、性能モニターを使用する為には、最適化レベル４（-O4）以上の最適化オプションの指定があらかじめ必要です。今回の例では、あらかじめ、-Oss -procnum=8 を指定しています（図 4.2）。実行後、オブジェクトファイル sample.o が作成されます。

手順 ： f77△コンパイルオプション△ソースファイル名△-Xfuncmonitor

（△は空白）

```
prompt> f77 -Oss -procnum=8 -c sample.f -Xfuncmonitor
```

図 4.2　例題プログラムのコンパイル

(2-2) リンク方法

オプション -lpl -parallel を追加してリンクします。

なお、要素並列化しないロードモジュールを作成する場合でも -parallel オプションの指定が必要です（図 4.3）。実行後、ロードモジュール sample が作成されます。

手順 ： f77△オブジェクト名△リンクオプション△-lpl△-parallel

（△は空白）

```
prompt> f77  sample.o  -o sample  -lpl -parallel
```

図 4.3　例題プログラムのリンク

(2-3) ロードモジュール実行（実行性能情報ファイル取得）方法

通常のロードモジュールと同様な手順で実行します。

手順 ： prun△-p△パーティション名△ロードモジュール名

（△は空白）

```
prompt> prun -p ALL sample
```

図 4.4　例題プログラムの実行

ロードモジュール実行終了後、関数／サブルーチン単位の実行性能情報ファイルである pl_wwwww_xxxx_yy.txt が作成されます。ここで、wwwww はロードモジュール名、xxxx はプロセス番号、yy はノード番号です（MPI などを使用して並列化したプログラムを、例えば 4 ノードで実行した場合、yy=0、1、2、3 の計 4 つのファイルが作成されます）。

この例の場合、ロードモジュール sample 実行時のプロセス番号が、18933 であったとする

と、実行終了後に作成されるファイル名は pl_sample_18933_0.txt となります。

(2-4) 実行性能情報ファイル pl_sample_18933_0.txt の説明

関数／サブルーチン実行性能情報ファイル pl_sample_18933_0.txt を図 4.5 に示します。

図 4.5 において、<Process Information> 以降がプロセス全体の性能情報、<Function Information> 以降が関数／サブルーチン単位の性能情報をそれぞれ示しています。

```
(Exe.R %)    :Execution ratio
(Par.E %)    :Parallel efficiency
(Max. total):Total of the Max User Time for each info (=CPU elapse time).
< Process Information >        …(a)
  Node no.                        :                  0        …(b)
  Process no.                     :              18933        …(c)
  Load file name                  : sample       …(d)
  User Time [sec]    (Max. total) :           0.011628 (Par.E  12.50 %)        …(e)
                        (Total) :           0.011628       …(f)
                          (Max) :           0.011628       …(g)
                          (0-2) :           0.011628             0.000000          0.000000
                          (3-5) :           0.000000             0.000000          0.000000
                          (6-7) :           0.000000             0.000000        …(h)
  LD/ST                           :            793909 (Par.E  12.50 %)
                          (Max) :            793909
                          (0-2) :            793909                    0                 0
                          (3-5) :                 0                    0                 0
                          (6-7) :                 0                    0
  Instruction count               :           2181246 (Par.E  12.50 %)
                          (Max) :           2181246
                          (0-2) :           2181246                    0                 0
                          (3-5) :                 0                    0                 0
                          (6-7) :                 0                    0
  Floating-point arithmetic count :            650081 (Par.E  12.50 %)
                          (Max) :            650081
                          (0-2) :            650081                    0                 0
                          (3-5) :                 0                    0                 0
                          (6-7) :                 0                    0
  MIPS                            :         187.588688
                          (0-2) :         187.588688             0.000000          0.000000
                          (3-5) :           0.000000             0.000000          0.000000
                          (6-7) :           0.000000             0.000000
  MFLOPS                          :          55.907423
                          (0-2) :          55.907423             0.000000          0.000000
                          (3-5) :           0.000000             0.000000          0.000000
                          (6-7) :           0.000000             0.000000

< Function Information >       …(i)
  Function name                   : MAIN         …(j)
  Source file name                : sample.f     …(k)
  Line no.                        :                  4       …(l)
  Execution count                 :                  1       …(m)
                          (0-2) :                  1                    0                 0
                          (3-5) :                  0                    0                 0
                          (6-7) :                  0                    0         …(n)
  User Time [sec]           (Max) :           0.004052 (Exe.R  34.84 %) (Par.E  12.50 %)     …(o)
                          (Total) :           0.004052 (Exe.R  34.84 %)
                          (0-2) :           0.004052             0.000000          0.000000
                          (3-5) :           0.000000             0.000000          0.000000
                          (6-7) :           0.000000             0.000000
```

```
～ 途中省略 ～

< Function Information >
  Function name                 : KEISAN
  Source file name              : sample.f
  Line no.                      :                16
  Execution count               :                 1
                         (0-2) :                 1                 0                 0
                         (3-5) :                 0                 0                 0
                         (6-7) :                 0                 0
  User Time [sec]          (Max) :          0.007576 (Exe.R  65.16 %) (Par.E  12.50 %)    …(p)
                         (Total) :          0.007576 (Exe.R  65.16 %)
                         (0-2) :          0.007576          0.000000          0.000000
                         (3-5) :          0.000000          0.000000          0.000000
                         (6-7) :          0.000000          0.000000

～ 以下省略 ～
```

図 4.5　pl_sample_18933_0.txt の内容

以下、各項目（図 4.5 における(a)～(p)）について説明します。

(a) この行以降の情報が、実行したプロセス全体の情報であることを示します。

(b) 実行時のノード番号を示します。

この例では、1 ノードしか使用していないので出力されるファイルは pl_sample_18933_0.txt のみです。2 ノード使用したときには、同時に pl_sample_18933_1.txt が出力されます。

(c) 実行時のプロセス番号です。

(d) 実行したロードモジュール名です。

(e) プロセスの実行にかかった CPU 時間[秒]です。-Xfuncmonitor を指定した場合、IP0 でプロセスの実行にかかった CPU 時間[秒]になります。ただし、手続き間要素並列を行なった場合には各 IP の情報が出力されます。

(f) プロセスの実行にかかった各 IP の CPU 時間[秒]を合計した値です。

(g) プロセスの実行にかかった各 IP の CPU 時間[秒]の中での最大値です。

(h) プロセスの実行にかかった各 IP での CPU 時間[秒]を示します。

(0-2)は IP0、1、2、(3-5)は IP3、4、5、(6-7)は IP6、7 の CPU 時間[秒]をそれぞれ示します。

図 4.5 では省略しましたが、LD/ST、Instruction count、Floating-point arithmetric count、MIPS、MFLOPS も同様の内容で表示されます。

関数内部の並列化リージョンの情報取得は後述する-Xparcmonitor を使用します。

(i) この行以降の情報が、(j)に名前が示されている関数／サブルーチン単位の情報であることを示します。

(j) 関数／サブルーチンの名前です。

(k) 関数／サブルーチン(j)が記述されている FORTRAN ソースファイル名を示します。

(l) ソースファイル(k)における、関数／サブルーチン(j)の記述開始行を示します。

この例の場合、MAIN はソースファイル sample.f の 4 行目から始まることを示していま

す。

(m) 関数／サブルーチン(j)が各 IP で実行された回数の合計を示します。

(n) 関数／サブルーチン(j)が各 IP で実行された回数を示します。

(0-2)は IP0、1、2、(3-5)は IP3、4、5、(6-7)は IP6、7 で実行された回数をそれぞれ示します。

(o) Exe.R（Execution ratio）は、プロセス全体の実行にかかった CPU 時間(e)に対して関数／サブルーチン(j)の実行比率を示します。この値をみることにより、処理の重たい部分を特定することができます。

この例の場合、プロセス全体では、

User Time [sec] (Max) = 0.011628

であり、MAIN においては、

User Time [sec] (Max) = 0.004052

なので、

0.004052÷0.011628×100 = 34.84%

となります。

プログラム sample は MAIN とサブルーチン KEISAN から構成されているので、

（MAIN の CPU 時間）+（KEISAN の CPU 時間）=（プロセスの CPU 時間）

すなわち、

（User Time(o)）+（User Time(p)）=（User Time(e)）

となります。MAIN はサブルーチン KEISAN の上位階層にありますが、MAIN に対する性能情報には、サブルーチン KEISAN の分は含まれません。一般に下位の関数にモニター関数を挿入した場合、上位の関数／サブルーチンには、下位の関数／サブルーチン部分の性能情報は含まれないので注意して下さい。

(3) 使用例２：要素並列化単位の性能情報まで採取する場合

さらに詳細に、要素並列化単位の性能情報まで採取する場合には、オプション-Xparmonitor を追加してコンパイルします。

コンパイル、リンク、実行の流れは、オプション名-Xfuncmonitor を-Xparmonitor に変更する以外、(2)の場合とまったく同じです。ロードモジュール sample 実行後、作成されたファイル名が pl_sample_67929_0.txt であるとして、このファイルの内容を説明します。

図 4.6 に、pl_sample_67929_0.txt の内容を示します。

```
(Exe.R %)    :Execution ratio
(Par.E %)    :Parallel efficiency
(Max. total):Total of the Max User Time for each info (=CPU elapse time).
< Process Information >
  Node no.                        :                  0
  Process no.                     :              67929
  Load file name                  : sample
  User Time [sec]    (Max. total) :          0.013999 (Par.E  19.80 %)      …(a)
                          (Total) :          0.022161       …(b)
                           (Max) :          0.013989       …(c)
```

```
                    (0-2) :          0.013989          0.001166          0.001166
                    (3-5) :          0.001170          0.001167          0.001169
                    (6-7) :          0.001165          0.001167       …(d)
  LD/ST                   :       1714102 (Par.E  26.97 %)       …(e)
                    (Max) :        794344       …(f)
                    (0-2) :        794344            131394            131394
                    (3-5) :        131394            131394            131394
                    (6-7) :        131394            131394       …(g)

～  途中省略  ～

< Function Information >
  Function name           : MAIN
  Source file name        : sample.f
  Line no.                :              4
  Execution count         :              1
                    (0-2) :              1                 0                 0
                    (3-5) :              0                 0                 0
                    (6-7) :              0                 0
  User Time [sec]   (Max) :       0.005879 (Exe.R  41.99 %) (Par.E  12.50 %)    …(h)
                  (Total) :       0.005879 (Exe.R  26.53 %)       …(i)
                    (0-2) :       0.005879          0.000000          0.000000
                    (3-5) :       0.000000          0.000000          0.000000
                    (6-7) :       0.000000          0.000000

～  途中省略  ～

< Function Information >
  Function name           : KEISAN
  Source file name        : sample.f
  Line no.                :             16
  Execution count         :              1
                    (0-2) :              1                 0                 0
                    (3-5) :              0                 0                 0
                    (6-7) :              0                 0
  User Time [sec]   (Max) :       0.006949 (Exe.R  49.64 %) (Par.E  12.50 %)    …(j)
                  (Total) :       0.006949 (Exe.R  31.36 %)
                    (0-2) :       0.006949          0.000000          0.000000
                    (3-5) :       0.000000          0.000000          0.000000
                    (6-7) :       0.000000          0.000000

～  途中省略  ～

< Parallel Information >      …(k)
  Function name           : KEISAN      …(l)
  Source file name        : sample.f
  Line no.                :             16      …(m)
  Execution count         :              8      …(n)
                    (0-2) :              1                 1                 1
                    (3-5) :              1                 1                 1
                    (6-7) :              1                 1
  User Time [sec]   (Max) :       0.000905 (Exe.R   6.47 %) (Par.E  99.93 %)    …(o)
                  (Total) :       0.007237 (Exe.R  32.66 %)
                    (0-2) :       0.000905          0.000905          0.000904
                    (3-5) :       0.000904          0.000905          0.000905
                    (6-7) :       0.000903          0.000905
  LD/ST                   :         600512 (Exe.R  35.03 %) (Par.E 100.00 %)
                    (Max) :          75064
                    (0-2) :          75064             75064             75064
                    (3-5) :          75064             75064             75064
                    (6-7) :          75064             75064
```

```
  Instruction count                :          2001226 (Exe.R  41.99 %) (Par.E 100.00 %)
                          (Max) :           250154
                          (0-2) :           250153          250153          250153
                          (3-5) :           250154          250154          250151
                          (6-7) :           250154          250154
  Floating-point arithmetic count :          1200032 (Exe.R  50.00 %) (Par.E 100.00 %)
                          (Max) :           150005
                          (0-2) :           150003          150003          150003
                          (3-5) :           150005          150005          150003
                          (6-7) :           150005          150005
  MIPS                             :       2210.762025
                          (0-2) :         276.373065      276.507481      276.682417
                          (3-5) :         276.577067      276.346082      276.376963
                          (6-7) :         276.894230      276.518367
  MFLOPS                           :       1325.679945
                          (0-2) :         165.725732      165.806333      165.911233
                          (3-5) :         165.849608      165.711098      165.729394
                          (6-7) :         166.039795      165.814409

～ 途中省略 ～

< Parallel Information >
  Function name                    : KEISAN
  Source file name                 : sample.f
  Line no.                         :              27
  Execution count                  :               8
                          (0-2) :                1               1               1
                          (3-5) :                1               1               1
                          (6-7) :                1               1
  User Time [sec]           (Max) :        0.000266 (Exe.R   1.90 %) (Par.E  98.49 %)    …(p)
                          (Total) :        0.002096 (Exe.R   9.46 %)
                          (0-2) :         0.000256        0.000262        0.000262
                          (3-5) :         0.000266        0.000262        0.000264
                          (6-7) :         0.000262        0.000263

～ 以下省略 ～
```

図 4.6　pl_sample_67929_0.txt **の内容**

以下、各項目（図 4.6 **における**(a)～(p)**）について説明します。**

(a) **プロセスの実行にかかった** CPU **時間[秒]を示します。**

Par.E **は** Parallel efficiency**（並列化効率）を表しており、**

Par.E =**（各** IP **での** CPU **時間の合計**(b)**）**

÷**（各** IP **での** CPU **時間の最大値**(c)×8）×100

= 0.022161÷（0.013989×8）× 100

= 19.80 [%]

で定義しています。この値が 100%**に近いほど、効率よく要素並列化されていることになります。**

(b) **プロセスの実行にかかった各** IP **の** CPU **時間[秒]を合計した値を示します。**

(c) **プロセスの実行にかかった各** IP **の** CPU **時間[秒]の中での最大値を示します。**

(d) **プロセスの実行にかかった各** IP **の** CPU **時間[秒]を示します。**

(0-2)**は** IP0**、**1**、**2**、**(3-5)**は** IP3**、**4**、**5**、**(6-7)**は** IP6**、**7 **の** CPU **時間[秒]をそれぞれ示しま

す。

(e) プロセスの実行中に各 IP で発行された LoaD 命令と STore 命令の数の合計を示します。

Par.E は

Par.E =（各 IP で発行された LD/ST 命令数の合計(e)）
÷（各 IP で発行された LD/ST 命令数の最大値(f)×8）×100
= 1714102 ÷ (794344×8) × 100
= 26.97 [%]

で計算されます。

(f) プロセスの実行中に各 IP で発行された LoaD 命令と STore 命令数の中での最大値を示します。

(g) プロセスを実行中に各 IP で発行された LoaD 命令と STore 命令数を示します。

(0-2)は IP0、1、2、(3-5)は IP3、4、5、(6-7)は IP6、7 をそれぞれ示します。

図 4.6 では省略しましたが、Instruction count、Floating-point arithmetric count、MIPS、MFLOPS の項目も(e)～(g)と同様の内容で表示されます。

(h) MAIN 部分の実行にかかった各 IP の CPU 時間[秒]の中での最大値です。 < Function Information > の欄にあるので、サブルーチン KEISAN の処理部分は含まれないことに注意して下さい。またこの例では、MAIN は逐次処理のみなので IP0 の CPU 時間となります。

Exe.R は

Exe.R =（MAIN の各 IP の CPU 時間の最大値(h)）
÷（プロセスの CPU 時間(a)）×100
= 0.005879÷0.013999×100
= 41.99[%]

で計算されます。

Par.E は

Par.E =（MAIN の各 IP の CPU 時間の合計(i)）
÷（MAIN の CPU 時間の最大値(h)×8）×100
= 12.5[%]

で計算されます。逐次処理部の Par.E は 12.5[%]になります。

(i) MAIN 部分の実行にかかった各 IP の CPU 時間[秒]を合計した値です。

この例では、MAIN は逐次処理なので IP0 の CPU 時間となります。

(j) サブルーチン KEISAN 部分の実行にかかった各 IP の CPU 時間[秒]の中での最大値です。

要素並列化単位まで性能情報を取得した場合（-Xparmonitor を指定してコンパイルした場合）、関数情報にはサブルーチン内で要素並列処理されている部分を除いた情報が出力されます。この例の場合には、要素並列処理されている部分の DO10 と DO30 を除いた分が出力されます（DO 20 は逐次処理なのでここに含まれます）。

(k) 要素並列化単位の情報であることを示します。

(l) この要素並列化単位を含むサブルーチン名を示します。

この例では KEISAN 中の要素並列化部分であることを示します。

(m) この要素並列化単位のソースファイル中での開始行を示します。
この例では、16 行目なので DO 10 のループ部分であることがわかります。
(n) この要素並列化単位（DO 10）の合計実行回数を示します。
この例では、各々IP で 1 回づつ実行されているので 8 回となります。
(o) サブルーチン KEISAN のループ DO 10 の実行にかかった各 IP の CPU 時間[秒]の中での最大値です。
(p) サブルーチン KEISAN の DO 30 のループ部分の実行にかかった各 IP の CPU 時間[秒]の中での最大値です。

(4) 例題プログラムの性能情報のまとめ

例題プログラムの実行性能情報ファイルの中から、CPU 時間の項目を取り出して、まとめたものを表 4.1 に示します。

表 4.1 例題プログラム sample の実行時間内訳

計算部分	ユーザーCPU 時間	全体に占める割合（Exe.R）
MAIN	0.005879 [s]	41.99 [%]
KEISAN 逐次処理（DO 20 含）	0.006949 [s]	49.64 [%]
KEISAN DO 10	0.000905 [s]	6.47 [%]
KEISAN DO 30	0.000266 [s]	1.90 [%]

MAIN の処理時間ですが、これは出力処理なので、ここでは検討しません。また、サブルーチン KEISAN 内の要素並列化されているループ DO 10 と DO 30 は、実行命令数や浮動小数点演算数が各 IP にほぼ均等に割り振られているので問題はないと判断します。

表 4.1 より、サブルーチン KEISAN における逐次処理部分の占める割合が大きいことが分かります。これは、ループ DO 20 が要素並列化されていないためです。

例題プログラムのループ DO 20 が要素並列化されない原因は、配列 A について、ループの繰り返し間に依存があるためです。そこで、

・ディレクティブを挿入して要素並列化可能にする
・ソースコードを書き換えて要素並列化可能にする

などを検討して、性能向上を図ります。この方法については、今後ご紹介する予定です。

以上より、この性能モニターツールで、サブルーチン KEISAN の DO 20 の処理を要素並列化することで性能向上を実現できるということが確認できます。

５．おわりに

SR8000 性能モニター機能についてご紹介いたしましたが、今後も性能情報の出力方式の見直しはもとより、使い易さの向上のためのユーザーインターフェースの改善も行っていく予定です。新たな機能や操作方法につきましては、追ってご紹介いたします。

以上