

SR8000 性能モニター機能の仕様改善のご報告

(株)日立製作所

1. はじめに

本「スーパーコンピューティングニュース」の以下の記事において、SR8000 の性能モニター機能についてご紹介させて頂きました。

「SR8000 性能モニター機能の利用法」 Vol.1, No.4 1999

「SR8000 性能モニターとログメッセージを用いたチューニング」 Vol.2, No.1 2000

今般、性能モニター機能の仕様を改善し、5 月初めにご提供する予定です。ここでは、その改善仕様を中心にご紹介致します。

2. 性能モニター機能改善仕様の概要

今般、実施致します性能モニター機能の改善ポイントを以下に示します。

(1) オプション名の統一

次に示す項目について、性能モニター (Performance Monitoring) を表すキーワード「pm」を統一して使用します。

(a) コンパイルオプション : -pmfunc または -pmpar

(b) リンケージオプション : コンパイルオプションと同一

(c) 出力ファイル名称 : pm_ロードモジュール名_日付_時間_node ノード番号
(バイナリーデータ)

性能モニターが出力するファイル名称に、日付と時間を使用するようにします。また今まで出力ファイル名に使用していたプロセス番号は廃止し、ノード番号を 3 桁表示に変更しました。これらの変更により、ls コマンドでファイル一覧を表示した際に、性能モニター情報ファイルが実行単位に整列するようになります。

(d) ポストプロセスコマンド : pmpr (性能モニター情報出力コマンド)

(2) コンパイル実行時の前提オプションの自動仮定

-pmfunc または -pmpar を指定すると -O4 (最適化レベル 4) を仮定します。

(3) 性能モニター情報出力内容の改善

性能モニター情報の出力内容を改善致します。詳細は後述致します。

(4) 性能モニター情報出力処理のポストプロセス化

プログラム実行時に生成する性能モニター情報ファイルをバイナリーファイル化し、このバイナリーファイルを入力としてポストプロセスを実行することにより、性能モニター情報を取り出す方式にしました。このポストプロセス化により、次の利点があります。

- ・異なる出力形式 (テキスト形式や CSV 形式、日本語や英語) の性能モニター情報を取得するために、プログラムを再実行しなくても良い。
- ・今後のポストプロセスの改善により出力形式や出力内容が変更になった場合でも、プログラムを再実行せずに、新しい形式で性能モニター情報を取り出せる。

(5) SR8000 モデル E1 / F1 のサポート

実行プロセッサが SR8000 モデル E1 / F1 の場合、データキャッシュミス数が表示でき

るようになります。

3. 性能モニター機能の概要

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います。性能モニター機能は、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、プログラム実行時の性能モニター情報ファイルを出力します。

性能モニター機能は、FORTRAN コンパイラーおよび C 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モニター情報を読み取る関数（性能モニター関数）をオブジェクトコードに自動挿入し、性能モニター関数を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。

つまり、ソースコードを修正する必要がなく、コンパイル・リンケージオプションの追加のみで性能モニター機能を利用することができます。また、性能モニター機能を使用しても、コンパイルの最適化レベルが変わる事はありません。更にソースファイル毎に性能モニター情報採取の ON/OFF を指定することができます。

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、実行性能モニター情報が集計処理されたファイルを出力します。

(1) 性能モニター機能を利用するために指定するコンパイル・リンケージオプション

プログラムに性能モニター関数を自動挿入するためのコンパイル・リンケージオプションとして、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

性能モニター関数の挿入位置	コンパイルオプション	リンケージオプション (*1)
関数 / サブルーチン単位	-pmfunc	-pmfunc
要素並列化単位	-pmpar	-pmpar
関数 / サブルーチン単位、さらに要素並列化単位	-pmfunc -pmpar	-pmfunc 又は -pmpar

(*1) makefile 等でリンケージオプションをコンパイルオプションとは別に指定する場合は、リンケージオプションにもコンパイルオプションと同様にオプション指定して下さい。

オプション -pmfunc を指定した場合の、関数 / サブルーチンに対する性能モニター関数挿入位置は、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。性能モニター情報は性能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。

また、オプション -pmpar を指定した場合の、要素並列化単位に対する性能モニター関数挿入位置は、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。

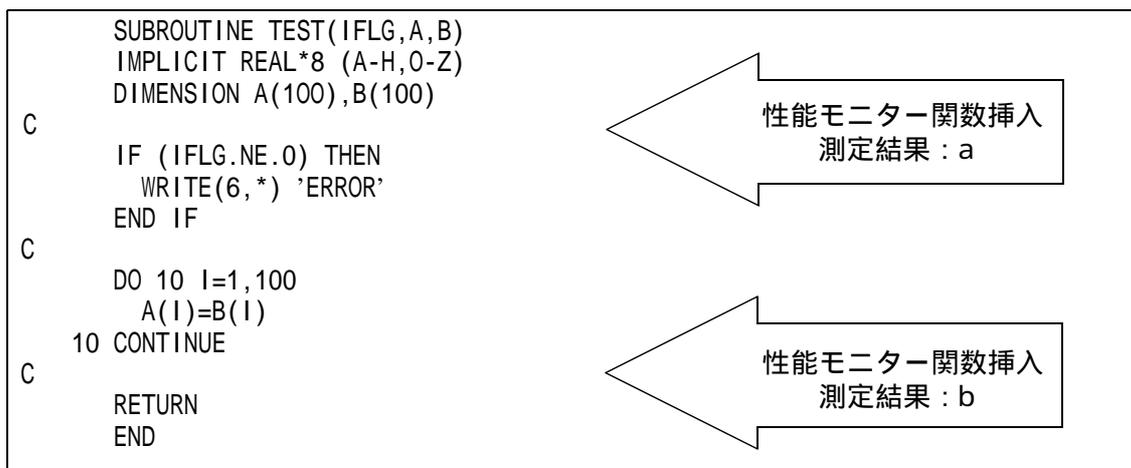


図 3.1 関数 / サブルーチンに関する性能モニター関数の挿入位置

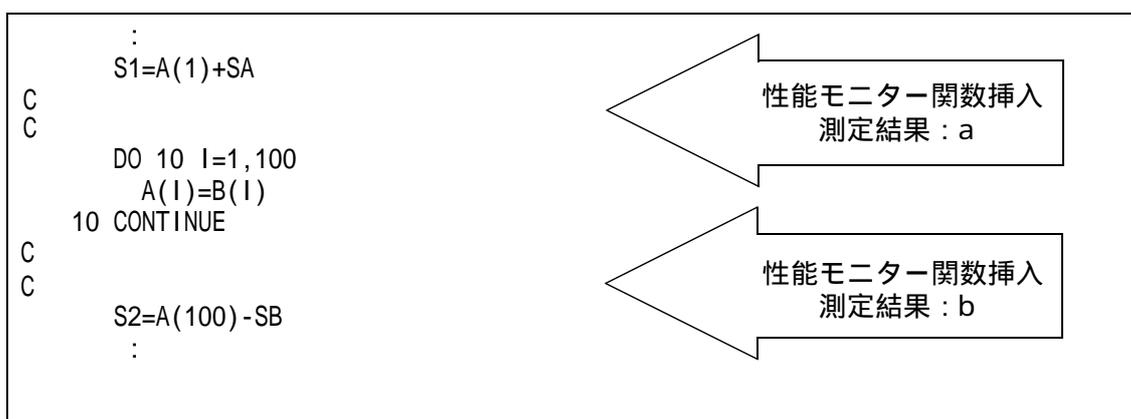


図 3.2 要素並列化単位に関する性能モニター関数の挿入位置

コンパイル・リンケージ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。

通常は、関数 / サブルーチン単位、及び要素並列化単位の両方の性能モニター情報を取り、関数 / サブルーチン単位の性能モニター情報から処理の重い関数 / サブルーチンの特定、要素並列化単位の性能モニター情報から要素並列処理部での IP (Instruction Processor) 間の負荷分散の度合いを調べます。しかし、要素並列化されたループの中から関数 / サブルーチンの呼び出しが頻繁に行われるプログラムでは、関数 / サブルーチンの呼び出し毎に性能モニター関数も実行されることになるため、性能モニター関数のオーバーヘッドが性能モニター情報に影響を与える場合があります。この場合は、関数 / サブルーチン単位毎、要素並列化単位毎に性能を分けて取ることや、要素並列化されたループからの呼び出しを受ける関数 / サブルーチンは性能モニター機能を OFF にしてコンパイルする等、対策してください。

(2) 採取可能な性能モニター情報

性能モニター機能により得られる性能モニター情報の項目を表 3.2 に示します。

表 3.2 性能モニター情報の項目

表示項目	内容	-pmfunc 指定時		-pmpar 指定時	
		プログラム全体	関数単位	プログラム全体	要素並列化単位
Date	プログラム実行開始日時		-		-
Node no	論理ノード番号		-		-
Process id	プロセス番号		-		-
Execution module	ロードモジュール名称		-		-
Input/Output count	I/O 回数		-		-
Input/Output quantity	I/O 量		-		-
Function/Subroutine	関数 / サブルーチン名称	-		-	
File	ソースファイル名称	-		-	
Line	開始行番号	-		-	
Times	実行回数	-		-	
CPU-Time	CPU 時間				
LD/ST	LoaD/STore 回数		2		2
Inst	総実行命令数		2		2
FLOP	浮動小数点演算数		2		2
MIPS	Million Instructions Per Second		2		2
MFLOPS	Mega FLOating point Operations Per Second		2		2
D-cache 1	データキャッシュミス回数	1	1	1	1
CPU-Time[%]	実行比率 (時間ベース)	-		-	
Element parallelizing rate	IP 負荷分散比率 (CPU 時間と MFLOPS)	-	2	-	2

: 表示する項目 - : 表示しない項目

1: 実行プロセッサが SR8000 E1 / F1 モデルの場合に表示する

2: 要素並列化された (IP 毎に情報を持つ) 場合に表示する

(3) 性能モニター情報ファイル

性能モニターにより得られる性能モニター情報の出力ファイル (性能モニター情報ファイル) は、下記のファイル名規則に従い作成されます。ファイルはバイナリー形式であり、(4) で述べる方法でテキスト形式に変換し、参照します。

・プリフィックス: pm_

・ロードモジュール名、プログラム実行開始日時、論理ノード番号 (3 桁) を使用

【例】ロードモジュール名が a.out、プログラム実行開始日時が 4 月 26 日の 12 時 34 分、論理ノード番号が 1 の場合

pm_a.out_Apr26_1234_node001

(4) 性能モニター情報の出力方法

性能モニター関数を取り込んだロードモジュールを通常の手順で実行することにより、性能モニターは性能モニター情報をバイナリー形式で出力します。このバイナリー形式のファイルをテキスト化するツールとして、性能モニター情報出力コマンド (pmpr) があります。pmpr は、このバイナリー形式の性能モニター情報ファイルを入力とし、テキスト化して標準出力に出力するコマンドです。使い方の例を以下に示します。

【例 1】性能モニター情報をテキスト形式で表示する。

```
pmpr pm_a.out_Apr26_1234_node001
```

【例 2】性能モニター情報を区切り文字コンマ(",")で区切った形式で表示する。

```
pmpr -c pm_a.out_Apr26_1234_node001
```

【例 3】性能モニター情報を区切り文字タブコードで区切った形式で表示する。

```
pmpr -s pm_a.out_Apr26_1234_node001
```

【例 4】性能モニター情報を任意の区切り文字（この場合はコロン）で区切った形式で表示する。

```
pmpr -s ':' pm_a.out_Apr26_1234_node001
```

4. 性能モニター機能の使用方法

性能モニター機能の使用方法について、コンパイルから性能モニター情報出力までの一連の流れを図 4.1 に示します。

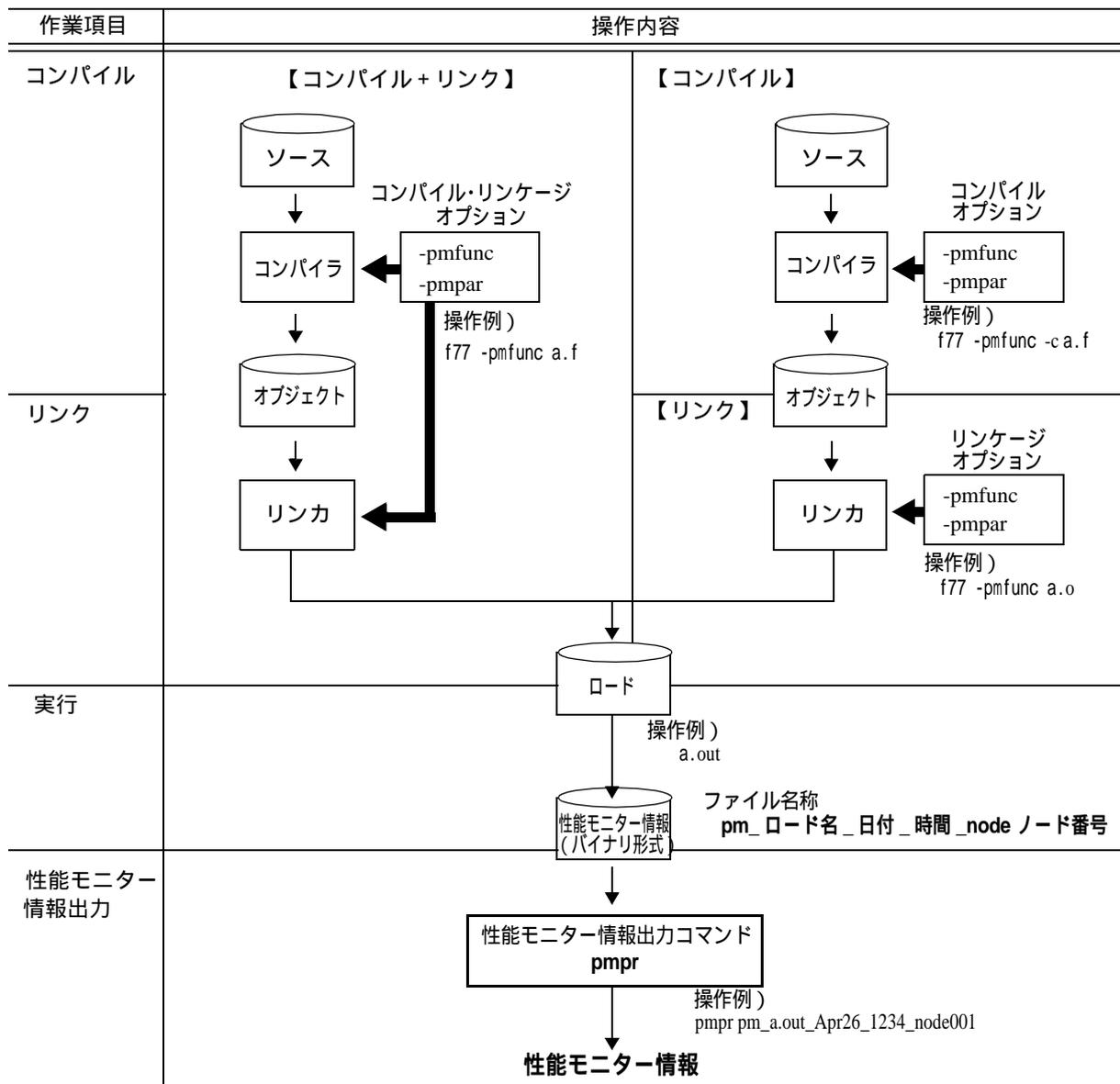


図 4.1 性能モニター機能の使用手順

5 . 性能モニター出力情報の詳細内容

3章(2)に述べましたように、使用するコンパイル・リンケージオプションにより、以下の性能モニター情報を得ることができます。

-pmfunc オプション指定時

- ・プログラム全体の性能モニター情報(サマリー情報)
- ・関数/サブルーチン単位の性能モニター情報

-pmpar オプション指定時

- ・プログラム全体の性能モニター情報(サマリー情報)
- ・要素並列化単位の性能モニター情報

-pmfunc および -pmpar オプションの両方を指定した時

- ・プログラム全体の性能モニター情報(サマリー情報)
- ・関数/サブルーチン単位の性能モニター情報
- ・要素並列化単位の性能モニター情報

各性能モニター情報の内容の詳細を以下に示します。

(1) プログラム全体の性能モニター情報(Execution summary)

・プログラム実行開始日時(Date)

プログラムの実行を開始した日時を出力します。

出力形式は、「曜日 月 日 時:分:秒 タイムゾーン 西暦」です。

・論理ノード番号(Node no)

プログラムを実行した論理ノード番号を出力します。

・プロセス番号(Process id)

実行したプログラムの識別番号(プロセス番号)を出力します。

・ロードモジュール名称(Execution module)

実行したプログラムのファイル名を出力します。

・CPU時間(CPU-Time)

プログラムの実行に掛かったCPU時間(IP番号0での情報)を秒単位でマイクロ秒の値まで出力します。

・MFLOPS

1秒あたりの浮動小数点演算数をM(百万)単位で小数点以下3桁まで出力します。

・MIPS

1秒あたりの実行命令数をM(百万)単位で小数点以下3桁まで出力します。

・入出力情報(Input count、Output count、Input quantity、Output quantity)

プログラム中で実行されたファイル入力と出力に対して、回数とデータ量をそれぞれ出力します。

・IP毎の詳細情報

IP毎の浮動小数点演算数(FLOP)、総実行命令数(Inst)、LoadD/STore回数(LD/ST)およびデータキャッシュミス回数(D-cache)を出力し、その合計(TOTAL)も出力します。ただし、データキャッシュミス回数は、実行プロセッサがSR8000 E1 / F1モデルの場合で

のみ出力します。

出力は、出力幅 7 桁で、以下の規則に従って出力します。

- 0 ~ 999999 : そのまま表示 (0 ~ 999999)
- 1000000 ~ 999999999 : キロ単位で表示 (1000K ~ 999999K)
- 1000000K ~ 999999999K : メガ単位で表示 (1000M ~ 999999M)
- 1000000M ~ 999999999M : ギガ単位で表示 (1000G ~ 999999G)
- 1000000G ~ 999999999G : テラ単位で表示 (1000T ~ 999999T)

ただし、各 IP の合計値の形式に合わせて出力します。出力単位の 1 の位を下回った場合で、値が真のゼロは 0、そうでない場合は小数点以下 1 桁まで表示します。

また、IP 中での最大値と最小値のマークを数値の右に付けます。最大値は >、最小値は < のマークで示します。最大値もしくは最小値が複数存在する場合は、それぞれの値にマークを付けます。ただし、全てがゼロの場合はマークは付けません。また、全てが等しい場合は全ての値に最大値のマークが付きます。

IP 毎の性能モニター情報は、スカラープログラム (使用 IP 数が 1) の場合は、IP0 での結果のみ出力し、合計 (TOTAL) は出力しません。また、最大値、最小値のマークも出力しません。

```
#####  
## Execution summary ##  
#####  
Date : Thu Feb 10 HH:MM:SS JST 2000  
Node no : 0  
Process id : xxxx  
Execution module : a.out  
CPU-Time : xx.xxxxxx[sec]  
MFLOPS : mmm.mmm  
MIPS : nnn.nnn  
Input count : xx  
Output coun : xx  
Input quantity : yyyyyyy[byte]  
Output quantity : yyyyyyy[byte]  
-----  
          FLOP   Inst   LD/ST D-cache  
-----  
IP0  ffffffff  iiiiii  ssssss< xxxxxxx  
IP1  ffffffff  iiiiii  ssssss> xxxxxxx  
IP2  ffffffff  iiiiii  ssssss  xxxxxx<  
IP3  ffffffff> iiiiii  ssssss  xxxxxx  
IP4  ffffffff  iiiiii> ssssss  xxxxxx>  
IP5  ffffffff  iiiiii< ssssss  xxxxxx  
IP6  ffffffff  iiiiii  ssssss  xxxxxx  
IP7  ffffffff< iiiiii  ssssss  xxxxxx  
-----  
TOTAL ffffffff  iiiiii  ssssss  xxxxxx
```

図 5.1 出力例 (サマリー情報の部分)

(2) 関数 / サブルーチン単位の性能モニター情報 (Function/Subroutine)

関数 / サブルーチン (以下、関数と略します) 単位の性能モニター情報には、(a) 関数の実行に要した CPU 時間を大きい順にソートしたランキング情報と (b) 要素並列化により複数 IP で実行された関数の詳細情報があります。

(a) ランキング情報 (Function/Subroutine Ranking)

・順位

オプション -pmfunc を指定してコンパイルされた関数すべてを、実行にかかった CPU 時間の降順に出力します。

・CPU 時間 (CPU-Time)

実行に要した CPU 時間 (IP の中での最大値) を出力幅 8 桁で以下の規則に従って出力します。

1 未満 : マイクロ秒まで表示 (0.000000 ~ 0.999999)

1 ~ 1000 未満 : ミリ秒まで表示 (1.000 ~ 999.999)

1000 ~ 100000000 未満 : 秒単位で表示 (99999999 は 1157 日分に相当します)

・時間ベースの実行比率 (CPU-Time[%])

プログラム全体の実行にかかった CPU 時間に対する各関数の CPU 時間の比率を小数点以下 2 桁まで出力します。

・実行回数 (Times)

プログラムの実行中に呼び出された回数 (IP 番号 0 での情報) を出力します。

・ソースファイル情報 (Func(File+Line))

関数名 (Func) と、関数が所属するソースファイル名 (File) 及び開始行番号 (Line) を出力します。要素並列化部分から呼び出された関数の場合には、"Func(File+Line)" の後ろにアスタリスク (*) を付加します。

(b) 各関数の詳細情報 (Function/Subroutine Detail)

この詳細情報は、要素並列化された部分から呼び出された関数 (複数 IP で実行された関数。但し、1 つの IP で実行された場合でも IP0 以外で実行された場合であれば表示対象となります) に対してのみ出力します。

詳細情報には、各 IP の CPU 時間 (CPU-Time)、浮動小数点演算数 (FLOP)、総実行命令数 (Inst)、LD/ST 命令数 (LD/ST)、データキャッシュミス回数 (D-cache。ただし、SR8000 E1 / F1 モデルのみ)、MFLOPS、MIPS、実行回数 (Times) および IP 負荷分散比率 (Element parallelizing rate) を出力します。また、それぞれの情報について、IP の中での最大値と最小値のマークを数値の右に付けて表示します。

・MFLOPS および MIPS は、以下の規則で出力します。

1000 未満 : 小数点以下 3 桁まで表示する (0 ~ 999.999)

1000 以上 : 小数点以下は表示しない (1000 ~ 9999999)

・IP 負荷分散比率 (Element parallelizing rate)

CPU 時間 (CPU-Time) と浮動小数点演算 (FLOP) について、以下の計算式により求めた値を表示します。

IP 負荷分散比率 = すべての IP の合計値 ÷ (IP の中での最大値 × 実行 IP 数)

```

#####
## Function/Subroutine                               ##
#####
=====
== Function/Subroutine Ranking                       ==
=====
      CPU-Time[%]      Times  Func(File+Line)
-----
[  1]  xxx.xxx[yy.yy]   zzzz  SUB1(nnnn.f+lll)*
[  2]  xxx.xxx[yy.yy]   zzzz  SUB2(nnnn.f+lll)
[  3]  xxx.xxx[yy.yy]   zzzz  MAIN(main.f+lll)
      :
[ nn]  0.xxxxxx[yy.yy]   zzzz  xxxxxxxx(xxxx.f+lll)
-----
TOTAL  xxx.xxx[yy.yy]
=====
== Function/Subroutine Detail                       ==
=====
[  1] Func(File+Line) : SUB1(nnnn.f+lll)
      CPU-Time      FLOP      Inst      LD/ST      D-cache      MFLOPS      MIPS      Times
-----
IP0   xxx.xxx> ffffffff  iiiiii  sssssss< xxxxxxx  mmm.mmm  nnn.nnn  ZZZZ>
IP1   xxx.xxx  ffffffff  iiiiii  sssssss> xxxxxxx  mmm.mmm  nnn.nnn  ZZZZ>
IP2   xxx.xxx  ffffffff  iiiiii  sssssss  xxxxxxx> mmm.mmm  nnn.nnn  ZZZZ>
IP3   xxx.xxx< ffffffff> iiiiii  sssssss  xxxxxxx  mmm.mmm  nnn.nnn> ZZZZ>
IP4   xxx.xxx  ffffffff  iiiiii> sssssss  xxxxxxx  mmm.mmm> nnn.nnn  ZZZZ>
IP5   xxx.xxx  ffffffff  iiiiii< sssssss  xxxxxxx  mmm.mmm< nnn.nnn  ZZZZ>
IP6   xxx.xxx  ffffffff  iiiiii  sssssss  xxxxxxx  mmm.mmm  nnn.nnn  ZZZZ>
IP7   xxx.xxx  ffffffff< iiiiii  sssssss  xxxxxxx< mmm.mmm  nnn.nnn< ZZZZ<
-----
TOTAL  xxx.xxx  ffffffff  iiiiii  sssssss  xxxxxxx  mmm.mmm  nnn.nnn  ZZZZ
-----
Element parallelizing rate : (TOTAL)/(Max * lps)
CPU-Time : xxx.xx[%] = xxx.xxx/(xxx.xxx*8)
FLOP      : xxx.xx[%] = ffffffff/(fffffff*8)

```

図 5.2 出力例（関数 / サブルーチン単位の性能モニター情報の部分）

(3) 要素並列化単位の性能モニター情報 (Element Parallel Region)

要素並列化単位の性能モニター情報には、(a) 実行に要した CPU 時間を大きい順にソートしたランキング情報と (b) IP 毎の詳細情報があります。

(a) ランキング情報 (Element Parallel Region Ranking)

・順位

オプション `-pmpar` を指定してコンパイルされた要素並列化単位すべてを、実行にかかった CPU 時間の降順に出力します。

・ソースファイル情報 (Func[Rank](File+Line))

要素並列化単位が所属する関数名 (Func) と、その関数が所属するソースファイル名 (File) 及び開始行番号 (Line) を出力します。Rank には、関数単位の性能モニター情報で求めた、その関数の順位が入ります。

(b) IP 毎の詳細情報 (Element Parallel Region Detail)

ランキングされた全ての要素並列単位について、IP 毎の詳細情報を出力します。出力項目については、(2)(b) の各関数の詳細情報と同様です。

```

#####
## Element Parallel Region                                ##
#####
=====
== Element Parallel Region Ranking ==
=====
      CPU-Time[%]      MFLOPS      MIPS      Times      Func[Rank](File+Line)
-----
[  1] xxx.xxx[yy.yy] mmm.mmm  nnn.nnn  zzzzz  SUB1[xx](nnnn.f+lll)
[  2] xxx.xxx[yy.yy] mmm.mmm  nnn.nnn  zzzzz  SUB2[xx](nnnn.f+lll)
[  3] xxx.xxx[yy.yy] mmm.mmm  nnn.nnn  zzzzz  MAIN[xx](main.f+lll)
:
:
:
[ nn] 0.xxxxxx[yy.yy] mmm.mmm  nnn.nnn  zzzzz  xxxxxx[xx](xx.f+lll)
-----
TOTAL  xxx.xxx[yy.yy]
=====
== Element Parallel Region Detail ==
=====
[  1] Func[Rank](File+Line) : SUB1[xx](nnnn.f+lll)
      CPU-Time      FLOP      Inst      LD/ST      D-cache      MFLOPS      MIPS      Times
-----
IP0  xxx.xxx> ffffffff  iiiiii  ssssss< xxxxxx  mmm.mmm  nnn.nnn  zzzzz>
IP1  xxx.xxx  ffffffff  iiiiii  ssssss> xxxxxx  mmm.mmm  nnn.nnn  zzzzz>
IP2  xxx.xxx  ffffffff  iiiiii  ssssss  xxxxxx> mmm.mmm  nnn.nnn  zzzzz>
IP3  xxx.xxx< ffffffff> iiiiii  ssssss  xxxxxx  mmm.mmm  nnn.nnn> zzzzz>
IP4  xxx.xxx  ffffffff  iiiiii> ssssss  xxxxxx  mmm.mmm> nnn.nnn  zzzzz>
IP5  xxx.xxx  ffffffff  iiiiii< ssssss  xxxxxx  mmm.mmm< nnn.nnn  zzzzz>
IP6  xxx.xxx  ffffffff  iiiiii  ssssss  xxxxxx  mmm.mmm  nnn.nnn  zzzzz>
IP7  xxx.xxx  ffffffff< iiiiii  ssssss  xxxxxx< mmm.mmm  nnn.nnn< zzzzz<
-----
TOTAL  xxx.xxx  ffffffff  iiiiii  ssssss  xxxxxx  mmm.mmm  nnn.nnn  zzzzz
-----
Element parallelizing rate : (TOTAL)/(Max * Ips)
CPU-Time : xxx.xx[%] = xxx.xxx/(xxx.xxx*8)
FLOP      : xxx.xx[%] = ffffffff/(fffffff*8)

```

図 5.3 出力例 (要素並列化単位の性能モニター情報の部分)

以上