

PROGINF/FTRACE ユーザーズガイド (G2AT03)

SX-Aurora TSUBASA

輸出する際の注意事項

本製品(ソフトウェアを含む)は、外国為替および外国貿易法で規定される規制貨物(または役務)に該当することがあります。

その場合、日本国外へ輸出する場合には日本国政府の輸出許可が必要です。

なお、輸出許可申請手続きにあたり資料等が必要な場合には、お買い上げの販売店またはお近くの当社営業拠点にご相談ください。

は し が き

本書は、SX-Aurora TSUBASA システムの性能情報解析ツール PROGINF/FTRACE について説明したものです。

本書の対象読者

本書は主に一般利用者およびプログラマを対象として書かれており、読者が Fortran コンパイラ (nfort)、C コンパイラ(nc)、C++コンパイラ(nc++)、NEC MPI の基本的な知識を有していることを前提としています。

本書の読み進め方

本書は次の章から構成されています。表の右側の対象読者を参考にして読み進めてください。

章	タイトル	内容	対象読者
第 1 章	PROGINF	PROGINF の使い方について説明しています。	一般利用者 プログラマ
第 2 章	FTRACE	FTRACE の使い方について説明しています。	一般利用者 プログラマ

関連説明書

本書に関連する説明書として次のものがあります。

- Fortran コンパイラ(**nfort**)の説明
Fortran コンパイラ ユーザーズガイド
- C コンパイラ(**ncc**)と C++コンパイラ(**nc++**)の説明
C/C++コンパイラ ユーザーズガイド
- NEC MPI の説明
NEC MPI ユーザーズガイド

表記上の約束

本書では次の表記規則を使用しています。

- ハイフン(-)に続く太字の文字列
コマンドのオプションを表します。
- コマンド実行例のドル記号(\$)
特に指定がない限り、**bash**(Bourne-Again シェル)のプロンプトを表します。
- **ncc/nc++/nfort/mpincc/mpinc++/mpinfort**
本書ではこれらコンパイルおよびリンクを行うコマンドをコンパイラと呼びます。

目次

第1章	PROGINF	1
1.1	使い方	1
1.2	オプション	7
1.2.1	コンパイラオプション	7
1.2.2	環境変数	7
1.3	注意事項	8
第2章	FTRACE	11
2.1	使い方	11
2.2	ユーザ指定リージョン	20
2.3	オプション	22
2.3.1	コンパイラオプション	22
2.3.2	測定プログラム実行時の環境変数	22
2.3.3	ftrace コマンドオプション	22
2.4	注意事項	23
付録A	改訂履歴	25

第1章 PROGINF

PROGINF はプログラム実行全体の解析情報を出力する機能です。本機能は NEC コンパイラを使用してプログラムをリンクすることで利用できます。

1.1 使い方

はじめに、PROGINF ライブラリをリンクしたプログラムを作成します。PROGINF ライブラリは既定でリンクされますので、特別なオプション指定は不要です。

例 ncc を使用して C プログラムをリンクする場合

```
$ ncc source.c
```

例 nc++ を使用して C++ プログラムをリンクする場合

```
$ nc++ source.cc
```

例 nfort を使用して Fortran プログラムをリンクする場合.

```
$ nfort source.f90
```

つぎに、環境変数 VE_PROGINF に YES、または DETAIL を指定し、プログラムを実行します。

例 bash の場合

```
$ export VE_PROGINF=DETAIL  
$ /opt/nec/ve/bin/ve_exec ./a.out
```

例 csh の場合

```
$ setenv VE_PROGINF DETAIL  
$ /opt/nec/ve/bin/ve_exec ./a.out
```

プログラムの実行終了時にプログラム実行解析情報が標準エラー(stderr)に出力されます。

```
***** Program Information *****
Real Time (sec)           :           33.060014
User Time (sec)          :           109.886525
Vector Time (sec)        :           104.243699
Inst. Count              :           77145324923
V. Inst. Count           :           16268966755
V. Element Count         :           4164804790539
V. Load Element Count    :           1921282387058
FLOP Count               :           1281280040326
MOPS                     :           42021.575083
MOPS (Real)              :           137499.248243
MFLOPS                   :           11844.518710
MFLOPS (Real)           :           38756.577193
A. V. Length             :           255.996884
V. Op. Ratio (%)         :           98.660787
L1 Cache Miss (sec)      :           0.414586
CPU Port Conf. (sec)     :           0.000000
V. Arith. Exec. (sec)    :           35.594797
V. Load Exec. (sec)      :           68.634671
VLD LLC Hit Element Ratio (%) :           0.001852
FMA Element Count        :           2002000063
Power Throttling (sec)   :           0.000000
Thermal Throttling (sec) :           0.000000
Max Active Threads       :                4
Available CPU Cores      :                8
Average CPU Cores Used   :           3.323850
Memory Size Used (MB)    :           7884.000000
Non Swappable Memory Size Used (MB) :           179.000000

Start Time (date)       :           Mon Feb 19 04:43:34 2018 JST
End Time (date)         :           Mon Feb 19 04:44:08 2018 JST
```

各項目の意味は以下の通りです。

(*1) VE_PROGINF=DETAIL 時のみ出力

(*2) VE_PEOGINF=DETAIL かつマルチスレッド実行時のみ出力

項目	単位	説明
Real Time	秒	経過時間
User Time	秒	ユーザ時間
Vector Time	秒	ベクトル命令実行時間
Inst. Count	-	全命令実行数
V. Inst. Count	-	ベクトル命令実行数
V. Element Count	-	ベクトル命令実行要素数
V. Load Element Count	-	ベクトル命令ロード要素数
FLOP Count	-	浮動小数点データ実行要素数
MOPS	-	ユーザ時間 1 秒あたりに実行された演算数(100 万単位)
MOPS (Real)	-	経過時間 1 秒あたりに実行された演算数(100 万単位)
MFLOPS	-	ユーザ時間 1 秒あたりに処理された浮動小数点データ 実行要素数(100 万単位)
MFLOPS (Real)	-	経過時間 1 秒あたりに処理された浮動小数点データ要 素数(100 万単位)
A. V. Length	-	平均ベクトル長
V. Op. Ratio	%	ベクトル演算率 ベクトル命令を使用して演算が行われた割合
L1 Cache Miss	秒	L1 キャッシュミス時間
CPU Port Conf.	秒	CPU ポート競合時間(*1)
V. Arith Exec.	秒	ベクトル演算実行時間(*1)
V. Load Exec.	秒	ベクトルロード実行時間(*1)
VLD LLC Hit Element Ratio	%	ベクトル命令によりロードされた要素のうち、LLC か らロードされた要素の比率

項目	単位	説明
FMA Element Count	-	FMA 命令実行要素数 (*1)
Power Throttling	秒	電力要因による HW 停止時間(*1)
Thermal Throttling	秒	温度要因による HW 停止時間(*1)
Max Active Threads	-	同時にアクティブだったスレッドの最大数(*2)
Available CPU cores	-	利用可能な CPU コアの個数(*2)
Average CPU Cores Used	-	平均 CPU コア使用率(*2)
Memory Size Used	メガバイト	メモリの最大使用量
Non Swappable Memory Size Used	メガバイト	Partial Process Swapping 機能でスワップアウトできないメモリの最大使用量
Start Time (date)	-	プログラムの実行開始時刻
End Time (date)	-	プログラムの実行終了時刻

PROGINFはAuroraハードウェアの性能カウンタを採取し解析情報を出力しています。採取する性能カウンタはVE_PERF_MODE環境変数で変更することができ、これによりPROGINFが出力する項目が切り替わります。上記の出力はVE_PERF_MODEが未指定、またはVE_PERF_MODEにVECTOR-OPを指定した場合の出力で主にベクトル演算に関連する情報が得られます。VE_PERF_MODEにVECTOR-MEMを指定した場合は以下の出力となり、主にベクトルとメモリアクセスに関連する情報が得られます。

***** Program Information *****	
Real Time (sec)	: 41.899895
User Time (sec)	: 167.316072
Vector Time (sec)	: 92.309990
Inst. Count	: 175930159073
V. Inst. Count	: 16278237491
V. Element Count	: 564896066618
V. Load Element Count	: 167991674972
FLOP Count	: 432838957109
MOPS	: 4956.779900
MOPS (Real)	: 19821.081454
MFLOPS	: 2583.413548
MFLOPS (Real)	: 10330.507186
A. V. Length	: 34.702533
V. Op. Ratio (%)	: 80.776073
L1 I-Cache Miss (sec)	: 0.256811
L1 O-Cache Miss (sec)	: 32.683728
L2 Cache Miss (sec)	: 32.884266
FMA Element Count	: 105937492420
Required B/F	: 5.773960
Required Store B/F	: 2.195626
Required Load B/F	: 3.578334
Actual V. Load B/F	: 0.404243
Power Throttling (sec)	: 0.000000
Thermal Throttling (sec)	: 0.000000
Max Active Threads	: 4
Available CPU Cores	: 8
Average CPU Cores Used	: 3.993234
Memory Size Used (MB)	: 1028.000000
Non Swappable Memory Size Used (MB)	: 179.000000
Start Time (date)	: Sun Dec 15 17:03:50 2019 JST
End Time (date)	: Sun Dec 15 17:04:32 2019 JST

VE_PERF_MODE 未指定時、または VE_PERF_MODE=VECTOR-OP と比べて、L1 Cache Miss、CPU Port Conf.、V. Arith Exec.、V. Load Exec.、VLD LLC Hit Element Ratio の代わりに以下が出力されます。

(*1) VE_PROGINF=DETAIL 時のみ出力

(*2) 100 以上の値は切り捨て

項目	単位	説明
L1 I-Cache Miss	秒	L1 命令キャッシュミス時間
L1 O-Cache Miss	秒	L1 オペランドキャッシュミス時間
L2 Cache Miss	秒	L2 キャッシュミス時間
Required B/F	-	ロード命令とストア命令に指定されたバイト数から算出した B/F (*1)(*2)
Required Store B/F	-	ストア命令に指定されたバイト数から算出した B/F (*1)(*2)
Required Load B/F	-	ロード命令に指定されたバイト数から算出した B/F (*1)(*2)
Actual V. Load B/F	-	ベクトルロード命令により実際に発生したメモリアクセスのバイト数から算出した B/F (*1)(*2)

1.2 オプション

PROGINF 機能に関するコンパイラオプションおよび環境変数は以下の通りです。

1.2.1 コンパイラオプション

オプション	説明
-proginf	PROGINF ライブラリをリンクします。(既定値)
-no-proginf	PROGINF ライブラリをリンクしません。

1.2.2 環境変数

環境変数	説明
VE_PROGINF	プログラムの実行終了時にプログラム実行解析情報を出力するかを制御します。 NO: 解析情報を出力しません。(既定値) YES: 基本的な項目について解析情報を出力します。 DETAIL: 詳細な項目について解析情報を出力します。
VE_PPROGINF_CO MPAT	プログラム実行解析情報の出力形式を制御します。 0: 最新の形式で出力します。(既定値) 1: 旧形式で出力します。 本形式では性能項目 “Non Swappable Memory Size Used” は出力されません。
VE_PROGINF_USE_ SIGNAL	シグナル SIGUSR1 を使ってスレッドの性能情報を採取します。本環境変数を指定する場合は、1.3(1)の注意事項をご参照ください。 NO: シグナルを使用しません。 YES: シグナルを使用します。(既定値)

環境変数	説明
VE_PERF_MODE	<p>採取する HW 性能カウンタの種類を変更します。これにより PROGINF の出力項目が切り替わります。</p> <p>VECTOR-OP: 主にベクトル演算に関連する情報を採取します。(既定値)</p> <p>VECTOR-MEM: 主にベクトルとメモリアクセスに関連する情報を採取します。</p>

1.3 注意事項

- (1) PROGINF は既定でスレッドにシグナル SIGUSR1 を発行して性能情報を採取しますが、環境変数 VE_PROGINF_USE_SIGNAL を NO に設定することで、シグナル発行を抑止し、例えばデバッガにシグナルを捕捉されないようにできます。シグナルを使用しない場合、PROGINF は代わりにコンパイラの自動並列および OpenMP によるワーカースレッドのみを対象にスレッドを停止して性能情報を採取するため、コンパイラの自動並列および OpenMP によるワーカースレッド以外のスレッドの性能情報が採取できず、プログラム実行解析情報に正しく表示されない場合があります。
 - マルチスレッド実行時に有効となる項目が表示されない
 - 情報が採取できなかったスレッドの性能情報が各項目に反映されない
(User Time、Real Time、Memory Size Used および Non Swappable Memory Size Used を除く)
- (2) プログラムと pthread ライブラリとのリンクにはオプション **-pthread** を使用する必要があります。オプション **-pthread** でなく **-lpthread** を使用した場合、スレッドの性能情報が採取できず、(1)と同様にプログラム実行解析情報が正しく表示されない場合があります。
- (3) コンパイラオプション **-fopenmp** や **-mparallel** を指定して作成した共有ライブラリを実行可能ファイルにリンクした場合、スレッドの性能情報が採取できず、(1)と同様にプログラム実行解析情報が正しく表示されない場合があります。
- (4) **-pthread** オプションを指定して作成した共有ライブラリを使用する場合、実行可能ファイルへのリンク時にも **-pthread** オプションを指定してください。**-pthread** オプションが指定されない場合、スレッドの性能情報が採取できず、(1)と同様にプログラム実行解析情報が正しく表示されない場合があります。

(5) PROGINF は MPI 以外のマルチプロセスプログラムをサポートしていません。

第2章 FTRACE

FTRACE は NEC コンパイラを使用してコンパイルおよびリンクすることで、プログラムの関数ごと、ユーザの指定した任意の区間(ユーザ指定リージョン)ごとの CPU 時間やベクトル化に関する詳細な性能情報を得る機能です。

2.1 使い方

はじめに、NEC コンパイラに **-ftrace** オプションを指定してプログラムをコンパイルおよびリンクし、性能測定用のプログラムを作成します。

例 ncc を使用して C プログラムをコンパイル・リンクする場合

```
$ ncc -ftrace source.c
```

例 nc++ を使用して C++ プログラムをコンパイル・リンクする場合

```
$ nc++ -ftrace source.cc
```

例 nfort を使用して Fortran プログラムをコンパイル・リンクする場合

```
$ nfort -ftrace source.f90
```

MPI プログラムの場合は **mpincc**, **mpinc++**, **mpinfort** コマンドを使用します。

例 mpincc を使用して MPI C プログラムをコンパイル・リンクする場合

```
$ mpincc -ftrace source.c
```

例 mpinc++ を使用して MPI C++ プログラムをコンパイル・リンクする場合

```
$ mpinc++ -ftrace source.cc
```

例 mpinfort を使用して MPI Fortran プログラムをコンパイル・リンクする場合

```
$ mpinfort -ftrace source.f90
```

つぎに、性能測定用プログラムを実行し、解析情報ファイルを採取します。このプログラムは、通常のプログラムと同じように実行できます。性能測定用プログラムは、プログラム終了時にプログラムの実行ディレクトリに解析情報ファイルを出力します。解析情報ファイルは、非 MPI プログラムの場合は一つ、MPI プログラムの場合は MPI プロセスの数だけ作成されます。

- 非 MPI プログラムの場合
ftrace.out
- MPI プログラムの場合
ftrace.out.グループ ID.ランク番号
グループ ID、ランク番号はそれぞれ NEC MPI の環境変数 MPIUNIVERSE、MPIRANK の値

例 非 MPI プログラムの場合

```
$ /opt/nec/ve/bin/ve_exec ./a.out  
$ ls ftrace.out  
ftrace.out
```

例 MPI プログラムの場合

```
$ mpirun -np 4 /opt/nec/ve/bin/ve_exec ./a.out  
$ ls ftrace.out.*  
ftrace.out.0.0 ftrace.out.0.1 ftrace.out.0.2 ftrace.out.0.3
```

最後に、解析情報ファイルを指定して **ftrace** コマンドを実行します。

例 非 MPI プログラムの場合

```
$ ftrace -f ftrace.out
```

例 MPIプログラムの場合

```
$ ftrace -f ftrace.out.*
```

例 MPIプログラムでユニバース番号0 ランク番号0のプロセスの結果のみを出力する場合

```
$ ftrace -f ftrace.out.0.0
```

測定内容が集計され、測定結果(解析リスト)が標準出力(stdout)に出力されます。

```

*-----*
FTRACE ANALYSIS LIST
*-----*

Execution Date : Sat Feb 17 12:44:49 2018 JST
Total CPU Time : 0:03'24"569 (204.569 sec.)

FREQUENCY  EXCLUSIVE      AVER.TIME      MOPS   MFLOPS  V.OP  AVER.      VECTOR L1CACHE  CPU PORT  VLD LLC  PROC.NAME
            TIME[sec]( %)      [msec]
            TIME      MISS      CONF HIT E.%

    1012    49.093( 24.0)    48.511  23317.2  14001.4  96.97  83.2    42.132  5.511    0.000  80.32 funcA
  160640    37.475( 18.3)     0.233  17874.6   9985.9  95.22  52.2    34.223  1.973    2.166  96.84 funcB
  160640    30.515( 14.9)     0.190  22141.8  12263.7  95.50  52.8    29.272  0.191    2.544  93.23 funcC
  160640    23.434( 11.5)     0.146  44919.9  22923.2  97.75  98.5    21.869  0.741    4.590  97.82 funcD
  160640    22.462( 11.0)     0.140  42924.5  21989.6  97.73  99.4    20.951  1.212    4.590  96.91 funcE
53562928   15.371(  7.5)     0.000   1819.0    742.2   0.00   0.0     0.000  1.253    0.000   0.00 funcG
      8    14.266(  7.0)   1783.201  1077.3    55.7   0.00   0.0     0.000  4.480    0.000   0.00 funcH
  642560    5.641(  2.8)     0.009   487.7     0.2  46.45  35.1     1.833  1.609    0.007  91.68 funcF
    2032    2.477(  1.2)     1.219   667.1     0.0  89.97  28.5     2.218  0.041    0.015  70.42 funcI
      8    1.971(  1.0)     246.398  21586.7  7823.4  96.21  79.6     1.650  0.271    0.000   2.58 funcJ

-----
54851346   204.569(100.0)    0.004  22508.5  12210.7  95.64  76.5    154.524  17.740   13.916  90.29 total

```

各項目の意味は以下の通りです。

項目	単位	説明
Execution Date	-	実行終了時刻
Total CPU Time	-	全ての関数の CPU 時間の合計
PROC.NAME	-	関数名
FREQUENCY	-	関数の呼び出し回数
EXCLUSIVE TIME	秒 (%)	関数の実行に要した EXCLUSIVE な CPU 時間と全関数の CPU 時間に対する比率

項目	単位	説明
AVER.TIME	ミリ秒	関数の1回の実行に要した EXCLUSIVE な CPU 時間の平均
MOPS	-	“EXCLUSIVE TIME” 1秒あたりに実行された演算数(100万単位)
MFLOPS	-	“EXCLUSIVE TIME” 1秒あたりに処理された浮動小数点データ実行要素数(100万単位)
V.OP RATIO	%	ベクトル演算率 ベクトル命令を使用して演算が行われた割合
AVER.V.LEN	-	平均ベクトル長
VECTOR TIME	秒	ベクトル命令実行時間
L1CACHE MISS	秒	L1 キャッシュミス時間
CPU PORT CONF.	秒	CPU ポート競合時間
VLD LLC HIT E.%	%	ベクトル命令によりロードされた要素のうち、LLC からロードされた要素の比率

FTRACE は性能測定用プログラムの実行時に採取した Aurora ハードウェアの性能カウンタを元に解析情報を出力しています。採取する性能カウンタは性能測定用プログラムの実行時に VE_PERF_MODE 環境変数で変更することができます。上記は VE_PERF_MODE が未設定、または VE_PERF_MODE に VECTOR-OP を指定して採取した解析情報ファイルを出力した例です。この場合、主にベクトル演算に関する項目が出力されます。VE_PERF_MODE に VECTOR-MEM を指定した場合、以下の出力が得られます。この場合、主にベクトルとメモリアクセスに関する項目が出力されます。

```

*-----*
FTRACE ANALYSIS LIST
*-----*

Execution Date : Sun Dec 15 21:51:48 2019 JST
Total CPU Time : 0:00'39"862 (39.862 sec.)

FREQUENCY  EXCLUSIVE          ...  L1ICACHE L1OCACHE L2CACHE  REQ.  REQ.ST  REQ.LD  ACT.VLD  FLOP COUNT  FMA ELEM.  PROC.NAME
          TIME[sec]( % ) ...  MISS    MISS    MISS    B/F    B/F    B/F    B/F
-----
15562     17.311( 43.4) ...  0.005   4.441   4.442   5.12   1.34   3.78   0.10   16689871512  4486524600 funcA
15562     17.235( 43.2) ...  0.009   4.007   4.009   5.12   1.34   3.78   0.01   16689871512  4187422960 funcB
  253      1.886(  4.7) ...  0.003   0.049   0.051   6.53   1.55   4.97   0.73   27492387186  9261379732 funcC
15562     1.392(  3.5) ...  0.010   0.013   0.021   6.77   3.42   3.35   0.14   21847989784  3469579024 funcD
15562     1.074(  2.7) ...  0.006   0.073   0.077   6.85   3.53   3.32   0.13   19036465492  3469579024 funcE
3147416   0.618(  1.6) ...  0.000   0.030   0.000   8.39   2.41   5.98   0.00   387132168    0 funcF
  2      0.211(  0.5) ...  0.000   0.000   0.000  39.45  21.54  17.91  0.00   46735724    0 funcG
  2      0.099(  0.2) ...  0.005   0.003   0.012  13.41   5.38   8.03   3.82   897306183   299101640 funcH
  1      0.020(  0.1) ...  0.000   0.002   0.002  50.15  26.41  23.75  0.00   3574991     0 funcI
  1      0.005(  0.0) ...  0.000   0.000   0.000   4.26   1.04   3.22   0.17   156349803   58463060 funcJ
  2      0.003(  0.0) ...  0.000   0.000   0.000   0.00   0.00   0.00   0.00     0         0 funcK
31124     0.003(  0.0) ...  0.000   0.002   0.005   0.00   0.00   0.00   0.00     0         0 funcL
31124     0.002(  0.0) ...  0.000   0.000   0.000   0.00   0.00   0.00   0.00     0         0 funcM
  4      0.001(  0.0) ...  0.000   0.000   0.000   2.74   0.04   2.70   0.96   14376844    4766560 funcN
  1      0.000(  0.0) ...  0.000   0.000   0.000 100.00  57.77  87.24  0.00     2292     0 funcO
  1      0.000(  0.0) ...  0.000   0.000   0.000 100.00 100.00 100.00  0.00     265     0 funcP
  1      0.000(  0.0) ...  0.000   0.000   0.000   0.00   0.00   0.00   0.00     0         0 funcQ
  1      0.000(  0.0) ...  0.000   0.000   0.000 100.00 100.00 100.00  0.00     14     0 funcR
  4      0.000(  0.0) ...  0.000   0.000   0.000 100.00 100.00 100.00  0.00     8         0 funcS
  1      0.000(  0.0) ...  0.000   0.000   0.000   2.60   0.26   2.34   0.10   685289     225700 funcT
 44      0.000(  0.0) ...  0.000   0.000   0.000   0.00   0.00   0.00   0.00     0         0 funcU
  2      0.000(  0.0) ...  0.000   0.000   0.000 100.00 100.00 100.00  0.00     4         0 funcV
  2      0.000(  0.0) ...  0.000   0.000   0.000   0.00   0.00   0.00   0.00     0         0 funcW
  1      0.000(  0.0) ...  0.000   0.000   0.000  58.60  50.00   8.60   0.00     20     0 funcX
  1      0.000(  0.0) ...  0.000   0.000   0.000   0.00   0.00   0.00   0.00     0         0 funcY
  2      0.000(  0.0) ...  0.000   0.000   0.000   0.00   0.00   0.00   0.00     0         0 funcZ
-----
3272238   39.862(100.0) ...  0.038   8.620   8.620   6.26   2.29   3.97   0.30  103262749091  25237042300 total

```

(...部分は出力を省略しています)

VE_PERF_MODE 未指定時、または VE_PERF_MODE に VECTOR-OP 指定した場合と比べて、L1ICACHE MISS、CPU PORT CONF、VLD LLC HIT E.%の代わりに以下が出力されます。

(*1) 100 以上の値は切り捨て

項目	単位	説明
L1ICACHE MISS	秒	L1 命令キャッシュミス時間
L1OCACHE MISS	秒	L1 オペランドキャッシュミス時間
L2CACHE MISS	秒	L2 キャッシュミス時間
REQ. B/F	-	ロード命令とストア命令に指定されたバイト数から算出した B/F (*1)
REQ. ST B/F	-	ストア命令に指定されたバイト数から算出した B/F (*1)
REQ. LD B/F	-	ロード命令に指定されたバイト数から算出した B/F (*1)
ACT. VLD B/F	-	ベクトルロード命令により実際に発生したメモリアクセスのバイト数から算出した B/F (*1)
FLOP COUNT	-	浮動小数点データ実行要素数
FMA ELEM.	-	FMA 命令実行要素数

MPI プログラムの場合、追加のリストが以下の形式で出力されます。MPI プロセスごとの情報は関数に続く(MPIUNIVERSE 番号.MPIRANK 番号)の行に出力されます。

ELAPSED TIME[sec]	COMM.TIME [sec]	COMM.TIME / ELAPSED	IDLE TIME [sec]	IDLE TIME / ELAPSED	AVER.LEN [byte]	COUNT	TOTAL LEN [byte]	PROC.NAME
12.444	0.000		0.000		0.0	0	0.0	funcA
9.420	0.000		0.000		0.0	0	0.0	funcB
7.946	0.000		0.000		0.0	0	0.0	funcG
7.688	0.000		0.000		0.0	0	0.0	funcC
7.372	0.000		0.000		0.0	0	0.0	funcH
5.897	0.000		0.000		0.0	0	0.0	funcD
5.653	0.000		0.000		0.0	0	0.0	funcE
1.699	1.475		0.756		3.1K	642560	1.9G	funcF
1.073	1.054		0.987		1.0M	4064	4.0G	funcI
0.704	0.045		0.045		80.0	4	320.0	funcK

FREQUENCY	EXCLUSIVE TIME[sec] (%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	L1CACHE MISS	CPU PORT	VLD CONF	LLC HIT	PROC.NAME
1012	49.093(24.0)	48.511	23317.2	14001.4	96.97	83.2	42.132	5.511	0.000	80.32	funcA	
253	12.089	47.784	23666.9	14215.9	97.00	83.2	10.431	1.352	0.000	79.40	0.0	
253	12.442	49.177	23009.2	13811.8	96.93	83.2	10.617	1.406	0.000	81.26	0.1	
253	12.118	47.899	23607.4	14180.5	97.00	83.2	10.463	1.349	0.000	79.36	0.2	
253	12.444	49.185	23002.8	13808.2	96.93	83.2	10.622	1.404	0.000	81.26	0.3	
...												
54851346	204.569(100.0)	0.004	22508.5	12210.7	95.64	76.5	154.524	17.740	13.916	90.29	total	

ELAPSED TIME[sec]	COMM.TIME [sec]	COMM.TIME / ELAPSED	IDLE TIME [sec]	IDLE TIME / ELAPSED	AVER.LEN [byte]	COUNT	TOTAL LEN [byte]	PROC.NAME
12.444	0.000		0.000		0.0	0	0.0	funcA
12.090	0.000	0.000	0.000	0.000	0.0	0	0.0	0.0
12.442	0.000	0.000	0.000	0.000	0.0	0	0.0	0.1
12.119	0.000	0.000	0.000	0.000	0.0	0	0.0	0.2
12.444	0.000	0.000	0.000	0.000	0.0	0	0.0	0.3
...								

各項目の意味は以下の通りです。

項目	単位	説明
PROC.NAME	-	関数名、または MPI プロセス (MPIUNIVERSE.MPIRANK)
ELAPSED TIME	秒	経過時間
COMM.TIME	秒	MPI 通信の送受信に要した経過時間
COMM.TIME / ELAPSED	-	関数内での経過時間に対する MPI COMM.TIME の比率
IDLE TIME	秒	MPI 通信の通信を行うまでの待ち時間、および 同期待ちに要した経過時間
IDLE TIME / ELAPSED	-	関数内での経過時間に対する MPI IDLE.TIME の比率
AVER.LEN	バイト	MPI 通信 1 回当たりの平均通信量(TOTAL LEN / COUNT)。バイト、キロバイト、メガバイト、 ギガバイト、テラバイト、ペタバイト単位で表示
COUNT	-	MPI 通信回数
TOTAL LEN	バイト	MPI 通信の通信量。バイト、キロバイト、メガ バイト、ギガバイト、テラバイト、ペタバイト単 位で表示

コンパイラの自動並列化機能、または OpenMP により並列化された関数は、それぞれの関数に続いてスレッドごとの性能値が出力されます。スレッドごとの情報は -thread0, -thread1, -thread2 ... 行に出力されます。MPI プログラムで本出力を行う場合、1 プロセスの解析情報ファイルを ftrace コマンドに入力してください。

FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	L1CACHE MISS	CPU CONF	PORT HIT	VLD E.%	LLC	PROC.NAME
40160	40.688(26.8)	1.013	3101.6	0.0	0.00	0.0	0.000	0.262	0.000	0.000	0.00	funcE	
40160	40.191(26.4)	1.001	3105.9	0.0	0.00	0.0	0.000	0.264	0.000	0.000	0.00	funcD	
40160	9.340(6.1)	0.233	17931.1	10016.3	95.20	52.2	8.572	0.435	0.541	95.92	funcB		
40160	8.217(5.4)	0.205	20558.4	11385.3	95.48	52.8	7.828	0.099	0.636	92.91	funcC		
160640	8.090(5.3)	0.050	257.6	0.0	31.45	6.9	5.275	0.645	0.002	97.92	funcF		
13390732	7.926(5.2)	0.001	1167.1	359.9	0.00	0.0	0.000	2.520	0.000	0.00	funcG		
160640	6.176(4.1)	0.038	42375.0	21745.2	98.02	98.3	5.084	0.603	1.147	98.13	funcD\$1		
40160	1.558(1.0)	0.039	41992.4	21549.0	98.02	98.3	1.274	0.158	0.287	98.28	-thread0		
40160	1.536(1.0)	0.038	42609.0	21865.3	98.02	98.3	1.264	0.149	0.287	98.41	-thread1		
40160	1.540(1.0)	0.038	42495.2	21806.8	98.02	98.3	1.268	0.152	0.287	97.92	-thread2		
40160	1.543(1.0)	0.038	42408.7	21762.4	98.02	98.3	1.279	0.144	0.287	97.92	-thread3		
1012	5.122(3.4)	5.061	18482.6	11129.8	96.24	83.5	4.122	0.688	0.000	96.46	funcA\$2		
253	1.259(0.8)	4.975	18802.5	11322.4	96.24	83.5	1.021	0.161	0.000	96.46	-thread0		
253	1.292(0.8)	5.106	18320.1	11031.9	96.24	83.5	1.033	0.180	0.000	96.46	-thread1		
253	1.294(0.9)	5.115	18288.9	11013.1	96.24	83.5	1.035	0.181	0.000	96.46	-thread2		
253	1.277(0.8)	5.049	18528.0	11157.1	96.24	83.5	1.032	0.167	0.000	96.46	-thread3		
...													
14048364	152.037(100.0)	0.011	9379.8	4107.7	74.34	76.3	46.350	8.509	3.477	90.34	total		
ELAPSED TIME[sec]	COMM.TIME [sec]	COMM.TIME / ELAPSED	IDLE TIME [sec]	IDLE TIME / ELAPSED	AVER.LEN [byte]	COUNT	TOTAL LEN [byte]	PROC.NAME					
50.531	0.000	0.000	0.000	0.000	0.0	0	0.0	funcD					
50.415	0.000	0.000	0.000	0.000	0.0	0	0.0	funcE					
11.919	0.000	0.000	0.000	0.000	0.0	0	0.0	funcG					
9.376	0.000	0.000	0.000	0.000	0.0	0	0.0	funcB					
8.591	0.000	0.000	0.000	0.000	0.0	0	0.0	funcH					
8.254	0.000	0.000	0.000	0.000	0.0	0	0.0	funcC					
8.138	7.782	0.956	5.987	0.736	3.1K	160640	490.2M	funcF					
3.863	2.628	0.680	2.557	0.662	1.0M	1016	1017.1M	funcI					
2.105	0.000	0.000	0.000	0.000	0.0	0	0.0	funcA					
1.570	0.000	0.000	0.000	0.000	0.0	0	0.0	funcD\$1					
1.570	0.000	0.000	0.000	0.000	0.0	0	0.0	-thread0					
1.549	0.000	0.000	0.000	0.000	0.0	0	0.0	-thread1					
1.552	0.000	0.000	0.000	0.000	0.0	0	0.0	-thread2					
1.557	0.000	0.000	0.000	0.000	0.0	0	0.0	-thread3					
...													

2.2 ユーザ指定リージョン

FTRACE はユーザが指定した区間(ユーザ指定リージョン)の性能情報を解析できます。以下の形式で区間の開始と終了をプログラム中に指定すると、その区間の性能情報が解析リストに出力されます。

関数プロトタイプ

```
// Beginning of user-specified region
extern int ftrace_region_begin(const char *id);

// End of user-specified region
extern int ftrace_region_end(const char *id);
```

引数

リージョン名を表す文字列(ID)を指定します。各ユーザ指定リージョンは、開始と終了に同じ ID を指定する必要があります。また、この ID は解析リストの PROC.NAME 列に出力されます。

戻り値

Name	Description
FTRACE_OK	正常終了
FTRACE_EINVAL	引数が不正
FTRACE_ENOBEGIN	対応するリージョンの開始が見つからない
FTRACE_EMAXNUM	利用できるリージョンの上限(200)に達した

プログラム例

```
#include <ftrace.h>
...
(void) ftrace_region_begin("loop#1"); // outside region begin

for (i = 0; i < n; i++) {
    ...
}

(void) ftrace_region_begin("loop#2"); // inside region begin

for (j = 0; j < n; j++) {
    ...
}

(void) ftrace_region_end("loop#2"); // inside region end

(void) ftrace_region_end("loop#1"); // outside region end
```

出力例

FREQUENCY	EXCLUSIVE TIME[sec](%)	AVER.TIME [msec]	MOPS	MFLOPS	V.OP RATIO	AVER. V.LEN	VECTOR TIME	L1CACHE MISS	CPU CONF	PORT HIT	VLD E.%	LLC	PROC.NAME
1012	49.093(24.0)	48.511	23317.2	14001.4	96.97	83.2	42.132	5.511	0.000		80.32		funcA
160640	37.475(18.3)	0.233	17874.6	9985.9	95.22	52.2	34.223	1.973	2.166		96.84		funcB
160640	30.515(14.9)	0.190	22141.8	12263.7	95.50	52.8	29.272	0.191	2.544		93.23		funcC
160640	23.434(11.5)	0.146	44919.9	22923.2	97.75	98.5	21.869	0.741	4.590		97.82		funcD
160640	22.462(11.0)	0.140	42924.5	21989.6	97.73	99.4	20.951	1.212	4.590		96.91		funcE
53562928	15.371(7.5)	0.000	1819.0	742.2	0.00	0.0	0.000	1.253	0.000		0.00		funcG
8	14.266(7.0)	1783.201	1077.3	55.7	0.00	0.0	0.000	4.480	0.000		0.00		funcH
642560	5.641(2.8)	0.009	487.7	0.2	46.45	35.1	1.833	1.609	0.007		91.68		funcF
2032	2.477(1.2)	1.219	667.1	0.0	89.97	28.5	2.218	0.041	0.015		70.42		funcI
8	1.971(1.0)	246.398	21586.7	7823.4	96.21	79.6	1.650	0.271	0.000		2.58		funcJ

54851346	204.569(100.0)	0.004	22508.5	12210.7	95.64	76.5	154.524	17.740	13.916		90.29		total
62248	37.709(18.4)	0.606	2200.2	1026.4	0.00	0.0	0.000	0.532	0.000		20.00		loop#1
2032	4.834(2.4)	2.379	415.8	0.0	28.61	6.3	4.098	0.246	0.000		0.00		loop#2

2.3 オプション

FTRACE機能に関するコンパイラオプションおよびftraceコマンドのオプションは以下の通りです。

2.3.1 コンパイラオプション

オプション	意味
-ftrace	ftrace機能用のオブジェクトファイルおよび実行ファイルを生成します

2.3.2 測定プログラム実行時の環境変数

環境変数	説明
VE_PERF_MODE	採取する HW 性能カウンタの種類を変更します。これにより FTRACE の出力項目が切り替わります。 VECTOR-OP: 主にベクトル演算に関連する情報を採取します。(既定値) VECTOR-MEM: 主にベクトルとメモリアクセスに関連する情報を採取します。

2.3.3 ftrace コマンドオプション

オプション	意味
-all	全ての関数の情報を出力します。
-num <i>n</i>	コストの上位 <i>n</i> 個の関数のみを出力します。-all と -num が同時に指定された場合、後に指定されたオプションが有効になります。
-f <i>file-name(s)</i>	読み込む解析情報ファイルを指定します。-f オプションが省略された場合、ftrace コマンドは既定で ftrace.out ファイルを読み込みます。複数のファイルを指定する場合、-f の後ろに空白区切りでファイルを並べます。複数のファイルが指定された場合、ftrace は既定でコストの上位 10 関数を出力します。

2.4 注意事項

- (1) FTRACE 機能は、ncc、nc++および nfort の自動並列化と OpenMP 並列化以外のマルチスレッドプログラム、および MPI 以外のマルチプロセスプログラムをサポートしません。
- (2) 自動並列化、OpenMP 並列化された関数を `-ftrace` を指定せずにコンパイルした場合、マスタスレッド以外のスレッドの性能情報は解析結果に含まれません。
- (3) NLC (NEC Numeric Library Collection) の並列化関数により生成されるマスタスレッド以外のスレッドの性能情報は解析結果に含まれません。
- (4) 測定プログラムの実行では性能情報を採取するための性能測定ルーチンを関数の入口、出口、ユーザ指定リージョンの前後で呼び出します。このため、プログラム内の関数呼出しの回数が多いとプログラムの実行時間が大幅に長くなることがあります。そのとき、性能情報が必要な関数を含むソースファイルだけを `-ftrace` でコンパイルするなどして、測定用プログラムを作成してください。
- (5) `-ftrace` を指定してコンパイルした共有ライブラリをアンロードしないでください。アンロードした場合、プログラム終了時に異常終了することがあります。

付録A 改訂履歴

版数	発行日	変更内容
初 版	2018 年 2 月	初版
第 2 版	2018 年 5 月	<ul style="list-style-type: none">• 1.3 注意事項を追加• 2.3 ユーザ指定リージョンのプログラム例を修正• 2.4 注意事項を追加
第 3 版	2018 年 12 月	<ul style="list-style-type: none">• 1.3 オプションの説明を追加• 1.4 注意事項を更新• 2.4 オプションの説明を追加• 2.5 注意事項を更新
第 4 版	2019 年 5 月	<ul style="list-style-type: none">• 2.5 注意事項を更新
第 5 版	2020 年 1 月	<ul style="list-style-type: none">• 1.1 使い方を更新• 1.2 オプションを更新• 1.3 注意事項を更新• 2.1 使い方を更新• 2.3 オプションを更新• 2.4 注意事項を更新
第 6 版	2020 年 10 月	<ul style="list-style-type: none">• 1.1 使い方を更新• 1.2 オプションを更新• 1.3 注意事項を更新
第 7 版	2021 年 12 月	<ul style="list-style-type: none">• 2.4 注意事項を更新

SX-Aurora TSUBASA システムソフトウェア

PROGINF/FTRACE ユーザーズガイド

2018年 2月 初 版

2021年 12月 第 7 版

日本電気株式会社

東京都港区芝五丁目 7 番 1 号

© NEC Corporation 2018-2021

日本電気株式会社の許可なく複製・改変などを行うことはできません。
本書の内容に関しては将来予告なしに変更することがあります。