FUJITSU

Fujitsu Software
Technical Computing Suite V4.0L20

Development Studio
プロファイラ使用手引書

J2UL-2568-01Z0(13)
2024年3月

まえがき

本書の目的

本書は、富士通製CPU A64FXを搭載したシステム向けのプロファイラの機能および使用方法について説明します。

本書の読者

本書は、プロファイラを使用してアプリケーションのチューニングを行う方が対象です。本書を読むためには、以下の知識が必要です。

- Linux(R)オペレーティングシステムにおけるプログラム開発作業、および、それに伴う基本的なLinuxオペレーティングシステムのコマンド操作に関する知識
- Microsoft(R) Excel(R)に関する知識

本書の構成

本書は、以下の構成になっています。

第1章 プロファイラの概要

プロファイラの概要について説明します。

第2章 基本プロファイラ

基本プロファイラについて説明します。

第3章 詳細プロファイラ

詳細プロファイラについて説明します。

第4章 CPU性能解析レポート

CPU性能解析レポートについて説明します。

第5章 注意事項

プロファイラ使用時の注意事項について説明します。

付録A トラブルシューティング

プロファイラに関するトラブルシューティングについて説明します。

付録B メッセージ一覧

プロファイラが出力する代表的なメッセージについて説明します。

本書の位置付け

本書は、以下のマニュアルと関係があります。必要に応じて参照してください。

- “Fortran文法書”
- “Fortran使用手引書”
- “Fortran使用手引書別冊 COARRAY”
- “Fortran翻訳時メッセージ”
- “C言語使用手引書”
- “C++言語使用手引書”
- “C/C++最適化メッセージ説明書”
- “Fortran/C/C++実行時メッセージ”
- “MPI使用手引書”

上記以外に、以下の関連ソフトウェアのマニュアルも必要に応じて参照してください。

- “ジョブ運用ソフトウェア”

・“FEFS/LLIO”

構文表記記号

構文表記記号とは、構文を記述するうえで特別な意味で定められた記号であり、以下のものがあります。

記号名	記号	説明
選択記号	{ }	この記号で囲まれた項目の中から、どれか1つを選択することを表します。
	|	この記号を区切りとして、複数の項目を列挙することを表します。
省略可能記号	[ ]	この記号で囲まれた項目を省略してよいことを表します。また、この記号は選択記号"{ }"の意味を含みます。
反復記号	...	この記号の直前の項目を繰り返して指定できることを表します。

輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

商標

- Linux(R)は米国及びその他の国におけるLinus Torvaldsの登録商標です。
- OpenMPは、OpenMP Architecture Review Boardの商標です。
- Microsoft、Windowsまたはその他のマイクロソフト製品の名称および製品名は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。
- Macは、米国およびその他の国で登録されたApple Inc.の商標です。
- Microsoft Corporationのガイドラインに従って画面写真を使用しています。
- Arm is trademark or registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
- そのほか、本マニュアルに記載されている会社名および製品名は、それぞれ各社の商標または登録商標です。
- 本資料に掲載されているシステム名、製品名などには、必ずしも商標表示(TM、(R))を付記しておりません。

出版年月および版数

版数	マニュアルコード
2024年 3月 第1.13版	J2UL-2568-01Z0(13)
2023年 3月 第1.12版	J2UL-2568-01Z0(12)
2022年 9月 第1.11版	J2UL-2568-01Z0(11)
2022年 3月 第1.10版	J2UL-2568-01Z0(10)
2021年11月 第1.9版	J2UL-2568-01Z0(09)
2021年 7月 第1.8版	J2UL-2568-01Z0(08)
2021年 3月 第1.7版	J2UL-2568-01Z0(07)
2021年 1月 第1.6版	J2UL-2568-01Z0(06)
2020年12月 第1.5版	J2UL-2568-01Z0(05)
2020年11月 第1.4版	J2UL-2568-01Z0(04)
2020年 9月 第1.3版	J2UL-2568-01Z0(03)
2020年 6月 第1.2版	J2UL-2568-01Z0(02)
2020年 3月 第1.1版	J2UL-2568-01Z0(01)
2020年 2月 初版	J2UL-2568-01Z0(00)

変更履歴

変更内容	変更箇所	版数
注意を修正。	3.1.1.1 3.1.1.2	第1.13版
基本ソフトウェアからWindows 8.1を削除。	1.1	第1.12版
注意を追加。	2.1.2 2.1.3	
「mpiexecコマンド」を追加。	5.1.5	
注意を追加。	2.1.3.1.2	第1.11版
eventの説明を修正。	3.2.3.2.4	
説明を追加。	5.2.1	
説明を追加。	5.2.9	
説明を追加。	A.1.5	
注意を修正。	4.2.2.3.2	第1.10版
説明文の見直し。	5.2.7 5.2.9	
説明を追加。	A.1.1	
説明を追加。	A.1.5	
-Ibalanceオプションの出力を見直し。	2.2.2.4.1 2.2.2.4.2	第1.9版
出力項目の表の項目および説明を見直し。	4.2.2.4.1 4.2.2.4.2 4.2.2.4.3 4.2.2.5.1 4.2.2.5.2	
CPU性能解析レポートのトラブルシューティングを追加。	A.3	
CPU周波数の説明を追加。	3.2.2.1	第1.8版
「終了ステータス」を追加。	5.1.7	
「LD_PRELOAD」を追加。	5.1.8	
以下の説明を追加。 ・ 翻訳時オプション -g ・ 翻訳時オプション -Nlineまたは-ffj-line	5.2.1	
「サンプリング数」を追加。	5.2.13	
「サンプリングによるシグナル割込み」を追加。	5.2.14	
メッセージの追加と見直し。	付録B	
説明文の見直し。	-	
プロファイル結果の操作環境とCPU性能解析レポートファイルを取得する環境を追記。	1.2 4.1.6	第1.7版
注意を追加。	2.1.1.1	

変更内容	変更箇所	版数
fippコマンドに-Mオプションを追加。	2.1.4	
注意を追加。	2.2.2.6	
注意を追加。	3.1.1.1	
-Mオプションの注意事項を追加。	5.2.12	
メッセージを追加。	B.1	
基本プロファイラのコスト情報出力時のインライン関数コスト計上先の変更。	5.2.7	第1.6版
CPUバインドを変更。	5.3.5	第1.5版
プロファイルデータ計測環境情報の出力形式を変更。	3.2.2.1	第1.4版
CPU性能解析レポートの出力結果のグラフを変更。	4.2.1 4.2.2.2.1 4.2.2.2.2 4.2.2.3.1 4.2.2.3.2 4.2.2.5.1 4.2.2.5.2	
CPU性能解析レポートの出力結果の注意事項を追加。	4.2.2.2.1 4.2.2.2.2 4.2.2.4.1 4.2.2.4.2 4.2.2.4.3	
翻訳時オプションのポイントを追加。	2.1.3.1	第1.3版
CPU動作状況の出力項目の説明を改善。	2.2.2.3	
CPU性能解析レポートの出力結果の注意事項を追加。	4.2.2.2.1 4.2.2.2.2 4.2.2.4.1 4.2.2.4.2 4.2.2.4.3 4.2.2.7.1 4.2.2.7.2	
COARRAY使用時の注意事項を追加。	5.1.1	
翻訳時オプションによる影響を追加。	5.1.2	
MPIプロファイリングインターフェース利用による影響を追加。	5.1.5	
MPIプログラムの言語間結合を追加。	5.1.6	
翻訳時オプションによる影響を追加。	5.2.1	
コールグラフ情報を変更。	5.2.9	
fjprof_spawn_dir_nameの注意事項を追加。	2.1.3.2	第1.2版
frequencyの注意事項を追加。	2.2.3.2.2 3.2.3.2.2	
cntfrqの注意事項を追加。	3.2.3.2.2	
Data Transfer CMGsの図を変更。	4.2.2.10	
-uオプションの注意事項を追加。	5.2.11	
CPUバインドを変更。	5.3.5	
メッセージの追加。	B.2 B.3	
説明文の見直し。	-	

変更内容	変更箇所	版数
fjprof_spawn_dir_nameの注意事項を追加。	2.1.3.2	第1.1版
fipppxコマンドおよびfippコマンドに-b{max|total}オプションを追加。	2.1.5 2.2.2.1 2.2.2.4 2.2.2.4.1 2.2.2.4.2 2.2.2.4.3	
fipppxコマンドおよびfippコマンドに-tcsvオプションを追加。	2.1.5	
fipppxコマンドおよびfippコマンドに-uオプションを追加。	2.1.5 5.2.11	
プロファイルデータ計測環境情報に“CPU周波数”を追加。	2.2.2.1 2.2.3.2.2	
CPU動作状況に“メモリースループットの理論値に対する実測値の比率”を追加。	2.2.2.3 2.2.3.2.5	
frequencyの注意事項を追加。	2.2.3.2.2 3.2.3.2.2	
計測区間指定ルーチンに関する注意事項を追加。	3.1.1.1 3.1.1.2	
詳細プロファイラが動的プロセス生成に対応。 ・ fappコマンドに-W{spawn|nospawn}オプションを追加。 ・ fapppxコマンドおよびfappコマンドの-pオプションに引数を追加。 ・ プロファイル結果(TEXT形式)の出力内容を変更。	3.1.4 3.2.2.1 3.2.2.2 3.2.2.3	
fapppxコマンドおよびfappコマンドの-Iオプションの注意事項を変更。	3.1.5	
詳細プロファイラのプロファイル結果(TEXT形式)にCPU性能解析情報を追加。 ・ fapppxコマンドおよびfappコマンドの-Iオプションの説明を変更。 ・ プロファイル結果(TEXT形式)に“CPU性能解析情報”を追加。	3.1.5 3.2.2.4	
詳細プロファイラにプロファイル結果(XML形式)を追加。 ・ fapppxコマンドおよびfappコマンドに-txmlオプションを追加。 ・ fapppxコマンドおよびfappコマンドの-pオプションの説明を変更。 ・ プロファイル結果(XML形式)を追加。	3.1.5 3.2.3 3.2.3.1 3.2.3.2 3.2.3.2.1 3.2.3.2.2 3.2.3.2.3 3.2.3.2.4 3.2.3.2.5	
CPUバインドに関する注意事項を変更。	4.1.6.1 5.3.5	
メッセージの追加。	B.2 B.3	
説明文の見直し。	-	

目　次

第1章 プロファイラの概要

プロファイラは、プログラムの実行性能を向上させるためのチューニングに有用な性能情報を計測および出力します。一般的に、プログラムの中で実行時間が長くかかっている区間(以降、高コスト演算区間と呼びます)を見つけ出してチューニングすることで、実行時間の短縮につなげることができます。

1.1 プロファイラの構成

プロファイラは、“基本プロファイラ”、“詳細プロファイラ”、“CPU性能解析レポート”の3機能で構成しています。

基本プロファイラ

基本プロファイラは、再翻訳なしにプログラム全体の性能傾向を把握するために使用するプロファイラです。大規模な並列プログラムでも低オーバーヘッドで性能情報を計測できます。基本プロファイラでは時間統計情報、CPU動作状況、コスト情報、コールグラフ情報、およびソースコード情報を出力します。詳細については“第2章 基本プロファイラ”を参照してください。

注意

プログラムが特定の条件を満たす場合、基本プロファイラが正しく計測できないためソースコードの修正または再翻訳が必要となる場合があります。詳細については“5.1 プロファイラ共通の注意事項”および“5.2 基本プロファイラの注意事項”を参照してください。また、計測区間を指定する場合もソースコードの修正および再翻訳が必要です。計測区間の指定については“2.1.1 計測区間指定ルーチンの追加”を参照してください。

詳細プロファイラ

詳細プロファイラは、特定区間の詳細な性能を把握するために使用するプロファイラです。詳細プロファイラを使用する場合、ソースコードの修正および再翻訳が必要です。詳細プロファイラでは時間統計情報、MPI通信コスト情報、およびCPU性能解析情報を出力します。詳細については“第3章 詳細プロファイラ”を参照してください。

CPU性能解析レポート

CPU性能解析レポートは、詳細プロファイラで計測したCPU性能解析情報を集約し、表やグラフを用いてわかりやすく可視化します。詳細については“第4章 CPU性能解析レポート”を参照してください。

注意

CPU性能解析レポートを使用する場合、Microsoft Excelおよび同ソフトウェアが動作する基本ソフトウェアについて以下のいずれかの組み合わせでなければいけません。

基本ソフトウェア	Microsoft Excel
Microsoft Windows 10 (64bit)	Microsoft Excel 2016 for Windows 64bit
macOS Catalina	Microsoft Excel 2016 for Mac

1.2 チューニングの流れ

プロファイラを使用した基本的なチューニングの流れについて説明します。

1. 基本プロファイラを使用して、高コスト演算区間の特定およびプログラム全体の性能傾向を把握します。
2. 1.で特定した高コスト演算区間に対して、ソースコードに詳細プロファイラ用の計測区間指定ルーチンを追加し再翻訳します。
3. 詳細プロファイラまたはCPU性能解析レポートを使用して、高コスト演算区間の詳細な情報を分析します。どちらの機能を使用するかは参照したい性能情報によって異なります。
   - 時間統計情報およびMPIに関する性能情報を参照したい場合、詳細プロファイラを推奨します。

ー CPUに関する詳細な性能情報を参照したい場合、CPU性能解析レポートを推奨します。

各機能で実施する操作の一覧を以下に示します。

操作	実行環境	第2章 基本プロファイラ	第3章 詳細プロファイラ	第4章 CPU性能解析レポート
計測区間指定ルーチンの追加 (基本プロファイラは任意)	任意のマシン	2.1.1 計測区間指定ルーチンの追加	3.1.1 計測区間指定ルーチンの追加	
環境変数の指定	ログインノードおよび計算ノード	2.1.2 環境変数の指定	3.1.2 環境変数の指定	
翻訳	ログインノードまたは計算ノード	2.1.3 翻訳	3.1.3 翻訳	
プロファイルデータの計測	計算ノード	2.1.4 プロファイルデータの計測	3.1.4 プロファイルデータの計測	
プロファイル結果の出力	ログインノードおよび計算ノード	2.1.5 プロファイル結果の出力	3.1.5 プロファイル結果の出力	
CPU性能解析レポートの作成	WindowsまたはmacOSマシン(注)	なし		4.1.6 CPU性能解析レポートの作成

注: CPU性能解析レポートファイルはログインノードに格納されています。

第2章 基本プロファイラ

この章では、基本プロファイラについて説明します。

基本プロファイラはサンプリング解析でプログラム全体の統計情報を計測および出力します。サンプリング解析のため、実行時間が約1秒以下のプログラムは基本プロファイラでは計測できません。基本プロファイラはプロファイルデータを計測するfippコマンドと、計測したデータからプロファイル結果を出力するfipppxコマンドの2つから構成されます。基本プロファイラが出力可能な統計情報は以下のとおりです。

時間統計情報

プログラムの経過時間、ユーザーCPU時間、およびシステムCPU時間を出力します。

CPU動作状況

メモリースループットや命令数、演算数などCPUの動作状況に関係する情報を表示します。

コスト情報

プログラム実行中にサンプリングした回数を手続、ループ、および行単位のコストとして出力します。

コールグラフ情報

手続の呼出し経路および手続の呼出し経路ごとのコストを出力します。

ソースコード情報

ソースコードの各行にコストを付加して出力します。

注意

fippコマンドはプロファイルデータを計測するコマンドとプロファイル結果を出力するコマンドの両方で使用可能ですが、本使用手引書ではプロファイルデータを計測するコマンドの意味で使用します。

基本プロファイラ使用時の注意事項については“5.1 プロファイラ共通の注意事項”および“5.2 基本プロファイラの注意事項”を参照してください。

ポイント

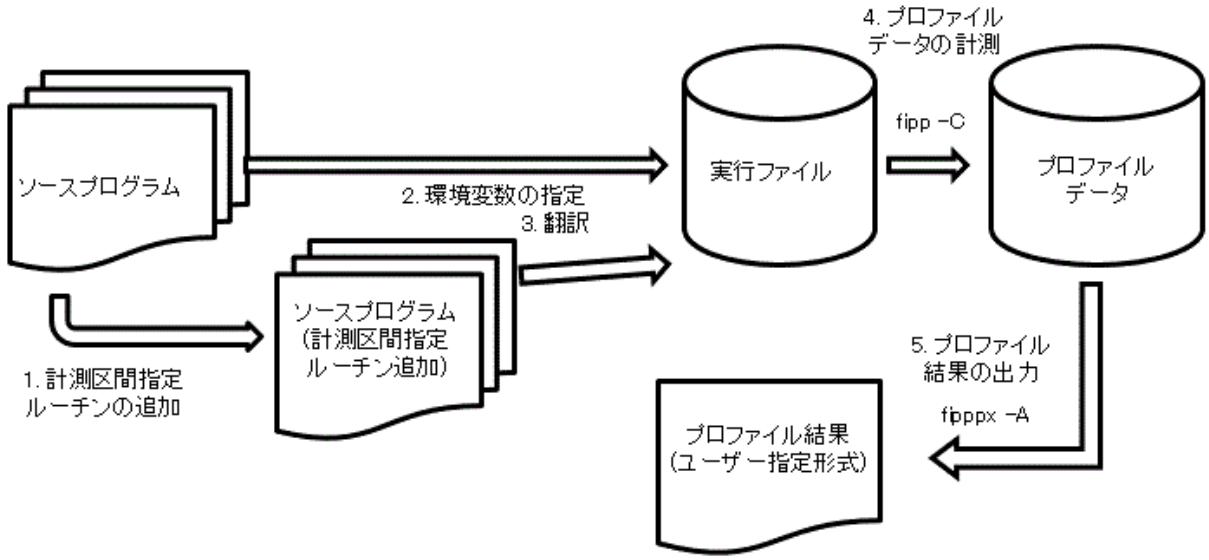
通常、基本プロファイラはプログラム全体の統計情報を計測および出力しますが-Sregionオプションを使用することで特定区間のみ計測することが可能です。ファイル入出力部分を除いて演算処理部分だけを計測したい場合などに使用します。ただし、-Sregionオプションを使用するためにはソースコードに計測区間指定ルーチンを追加する必要があります。-Sregionオプションについては“2.1.4 プロファイルデータの計測”を、計測区間指定ルーチンについては“2.1.1 計測区間指定ルーチンの追加 ”を参照してください。

MPIプログラムでMPI_COMM_SPAWNルーチンまたはMPI_COMM_SPAWN_MULTIPLEルーチンを使用して動的に並列プロセスを生成した場合、ジョブ運用ソフトウェアは生成されたプロセスに番号を割り振ります。この割り振られた番号のことを本使用手引書では“spawn番号”と呼びます。詳細は“MPI使用手引書”を参照してください。

2.1 基本プロファイラの使用手順

基本プロファイラの使用手順を以下に示します。

図2.1 基本プロファイラの使用手順



各操作の詳細を以下に示します。

2.1.1 計測区間指定ルーチンの追加

注意

fippコマンドに-Sregionオプションを指定せずにプロファイルデータを計測する場合、計測区間指定ルーチンを追加する必要はありません。-Sregionオプションについては“2.1.4 プロファイルデータの計測”を参照してください。

プロファイルデータを計測する区間(“計測開始位置”および“計測終了位置”)の指定に必要な計測区間指定ルーチンをソースコードに追加します。

基本プロファイラ向け計測区間指定ルーチンの詳細を以下に示します。

2.1.1.1 fipp_start / fipp_stopサブルーチン (Fortran)

書式

CALL fipp_start

CALL fipp_stop

機能説明

基本プロファイラによるプロファイルデータの計測を開始または終了します。これらのサブルーチンはfippコマンドに-Sregionオプションを指定した場合のみ有効になります。-Sregionオプションについては“2.1.4 プロファイルデータの計測”を参照してください。

fipp_start

基本プロファイラによるプロファイルデータの計測を開始します。

fipp_stop

基本プロファイラによるプロファイルデータの計測を終了します。

例

計測区間指定ルーチンの使用例

```
program main
...
do i=1,10000
    ...
    call fipp_start   ! 計測を開始する
    do j=1,10000
        ...
    end do
    call fipp_stop    ! 計測を終了する
end do
end program main
```

注意

- これらのサブルーチンを複数回呼び出す場合、必ずfipp_start、fipp_stopの順番で呼び出してください。fipp_stopを呼び出す前に再度fipp_startを呼び出した場合、またはfipp_startを呼び出す前にfipp_stopを呼び出した場合、警告メッセージを出力してその呼び出しを無視します。また、fipp_stopを呼び出さずにプロセスが終了した場合、その区間のプロファイルデータは計測しません。
- これらのサブルーチンを複数回呼び出した場合、指定したすべての計測区間の結果を合算します。
- プロセス並列プログラムの場合、計測対象にしたいすべてのプロセスでこれらのサブルーチンを呼び出してください。呼び出しが行われなかったプロセスのプロファイルデータは計測しません。

例

すべてのプロセスを計測対象とする例(mpi_initサブルーチンを呼び出す前に計測を開始する)

```
call fipp_start         ! 計測を開始する
call mpi_init(err)
  ...
call mpi_finalize(err)
call fipp_stop          ! 計測を終了する
```

すべてのプロセスを計測対象とする例(mpi_initサブルーチンを呼び出した直後に計測を開始する)

```
call mpi_init(err)
call fipp_start         ! 計測を開始する
  ...
call fipp_stop          ! 計測を終了する
call mpi_finalize(err)
```

プロセス0のみ計測対象とする例

```
call mpi_init(err)
call mpi_comm_rank(mpi_comm_world,rank,err)
if(rank==0) then
    call fipp_start   ! プロセス0のみ、計測を開始する
end if
    ...
if(rank==0) then
    call fipp_stop    ! プロセス0のみ、計測を終了する
end if
call mpi_finalize(err)
```

- 翻訳時オプション-mldefault=cdeclを有効にしてFortranプログラムを翻訳する場合、Fortranプログラム内で指定する計測区間指定ルーチンの名前を、以下のように変更してください。

変更前	変更後
fipp_start	fipp_start_
fipp_stop	fipp_stop_

- 翻訳時オプション-AUが有効な場合、計測区間指定ルーチンの名前はすべて小文字で記述してください。

2.1.1.2 fipp_start関数 / fipp_stop関数 (C言語およびC++言語)

書式

```
#include "fj_tool/fipp.h"

void fipp_start();
void fipp_stop();
```

機能説明

基本プロファイラによるプロファイルデータの計測を開始または終了します。これらの関数はfippコマンドに-Sregionオプションを指定した場合のみ有効になります。-Sregionオプションについては“2.1.4 プロファイルデータの計測”を参照してください。

fipp_start()

基本プロファイラによるプロファイルデータの計測を開始します。

fipp_stop()

基本プロファイラによるプロファイルデータの計測を終了します。

例

計測区間指定ルーチンの使用例

```
#include "fj_tool/fipp.h"       // ヘッダーファイルのインクルード
...
int main(void)
{
    int i,j;
    for(i=0;i<10000;i++){
        ...
        fipp_start();            // 計測を開始する
        for(j=0;j<10000;j++){
            ...
        }
        fipp_stop();             // 計測を終了する
    }
    return 0;
}
```

注意

- これらの関数を複数回呼び出す場合、必ずfipp_start()、fipp_stop()の順番で呼び出してください。fipp_stop()を呼び出す前に再度fipp_start()を呼び出した場合、またはfipp_start()を呼び出す前にfipp_stop()を呼び出した場合、警告メッセージを出力してその呼び出しを無視します。また、fipp_stop()を呼び出さずにプロセスが終了した場合、その区間のプロファイルデータは計測しません。
- これらの関数を複数回呼び出した場合、指定したすべての計測区間の結果を合算します。

- プロセス並列プログラムの場合、計測対象にしたいすべてのプロセスでこれらの関数を呼び出してください。呼び出しが行われなかったプロセスのプロファイルデータは計測しません。

例

すべてのプロセスを計測対象とする例(MPI_Init関数を呼び出す前に計測を開始する)

```
fipp_start();              // 計測を開始する
MPI_Init(&argc, &argv);
  ...
MPI_Finalize();
fipp_stop();               // 計測を終了する
```

すべてのプロセスを計測対象とする例(MPI_Init関数を呼び出した直後に計測を開始する)

```
MPI_Init(&argc, &argv);
fipp_start();              // 計測を開始する
  ...
fipp_stop();
MPI_Finalize();            // 計測を終了する
```

プロセス0のみ計測対象とする例

```
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
if(rank==0){
    fipp_start();  // プロセス0のみ、計測を開始する
}
    ...
if(rank==0){
    fipp_stop();   // プロセス0のみ、計測を終了する
}
MPI_Finalize();
```

2.1.2 環境変数の指定

プロファイラ使用時に必要な環境変数を指定します。

環境変数名	設定値
PATH	/*製品インストールパス*/bin
LD_LIBRARY_PATH	/*製品インストールパス*/lib64

“*製品インストールパス*”については、システム管理者にお問い合わせください。

注意

- 名前が“FIPP_”、“FAPP_”、“PROF_”、“FJPROF_”、“FPROF_”のいずれかから始まる環境変数は、プロファイラが使用します。プロファイラ使用時は、これらの環境変数を指定しないでください。
- mpiexecコマンドに“-x LD_LIBRARY_PATH=value”を指定してプログラムを実行する場合、“5.1.5 mpiexecコマンド”を参照してください。

2.1.3 翻訳

プログラムを翻訳します。プロファイラを利用するためのプログラム翻訳方法について、以下に示します。

注意

- プログラムの翻訳時に-rpathオプションまたは環境変数LD_RUN_PATHを指定する場合、動的セクション属性DT_RPATHに“/*製品インストールパス*/lib64”を含まないようにしてください。DT_RPATHに“/*製品インストールパス*/lib64”が含まれている場合、基本プロファイラ使用時にプロファイルデータの計測が失敗します。readelf -dコマンドの出力から、プログラムに指定されたDT_RPATHの値を確認できます。出力された情報のType“(RPATH)”を参照してください。“*製品インストールパス*”については、システム管理者にお問い合わせください。
- fork、vfork、popenシステムコール、system関数などプログラム内でプロセスを生成するシステムコールまたは関数を使用しないでください。使用した場合、プロファイルデータを正しく計測することができません。
- プロファイラを使用する場合、プログラムに対してstripコマンドを使用しないでください。また、ユーザーの共有ライブラリのコストを計測する場合には、共有ライブラリに対してstripコマンドを使用しないでください。stripコマンドを使用してシンボルを削除した場合、プロファイルデータを正しく計測することができません。
- プロファイラを使用する場合、富士通コンパイラの翻訳コマンドを使用してプログラムの翻訳およびリンクを行ってください。GNUコンパイラなどの他のコンパイラの翻訳コマンドを使用した場合、プロファイラを使用することはできません。

2.1.3.1 翻訳時オプション

プロファイラを利用するためにはツールライブラリを結合したプログラムを作成する必要があります。
frtpx、fccpx、FCCpx、mpifrtpx、mpifccpx、mpiFCCpxの各コマンドに、ツールの動作を指示する翻訳時オプションを指定します。翻訳時オプションの指定方法については“Fortran使用手引書”、“C言語使用手引書”、“C++言語使用手引書”、または“MPI使用手引書”を参照してください。

ポイント

- frtpx、fccpx、FCCpx、mpifrtpx、mpifccpx、mpiFCCpxの各コマンドを使用する場合、デフォルトでツールライブラリを結合したプログラムを作成します。そのため、通常は本オプションを意識する必要はありません。ただし、プロファイラはMPI関数をフックするため、プロファイラとMPIプロファイリングインターフェースを併用することはできません。MPIプロファイリングインターフェースを利用する場合、下記の翻訳時オプションを使用してツールライブラリを結合しないようにしてください。詳細については“5.1.6 MPIプロファイリングインターフェース利用による影響”を参照してください。
- 各コマンドの翻訳時オプションには、プロファイラの動作に影響のあるオプションが含まれています。詳細については“5.1.2 翻訳時オプションによる影響”を参照してください。

言語ごとのプロファイラ向け翻訳時オプションを以下に示します。

2.1.3.1.1 Fortran

Fortranプログラムを翻訳する際に使用するプロファイラ向け翻訳時オプションを以下に示します。

-N{fjprof|nofjprof}

ツールライブラリを結合するかどうかを指示します。本オプションはリンク時に有効になります。本オプションを指定しない場合、-Nfjprofオプションが有効になります。

-Nfjprof

ツールライブラリを結合します。プロファイラを使用できます。

-Nnofjprof

ツールライブラリを結合しません。プロファイラを使用できません。

2.1.3.1.2 C言語およびC++言語

C言語およびC++言語の場合、tradモードとclangモードという2種類のモードが存在します。tradモードとclangモードの違いについては“C言語使用手引書”または“C++言語使用手引書”を参照してください。それぞれのモードでプログラムを翻訳する際に使用するプロファイラ向け翻訳時オプションを以下に示します。

tradモード

C言語およびC++言語のプログラムをtradモードで翻訳する際に使用するプロファイラ向け翻訳時オプションを以下に示します。

-N{fjprof|nofjprof}

ツールライブラリを結合するかどうかを指示します。本オプションはリンク時に有効になります。本オプションを指定しない場合、-Nfjprofオプションが有効になります。

-Nfjprof

ツールライブラリを結合します。プロファイラを使用できます。

-Nnofjprof

ツールライブラリを結合しません。プロファイラを使用できません。

clangモード

C言語およびC++言語のプログラムをclangモードで翻訳する際に使用するプロファイラ向け翻訳時オプションを以下に示します。

-f{fj-fjprof|fj-no-fjprof}

ツールライブラリを結合するかどうかを指示します。本オプションはリンク時に有効になります。本オプションを指定しない場合、-ffj-fjprofオプションが有効になります。

-ffj-fjprof

ツールライブラリを結合します。プロファイラを使用できます。

-ffj-no-fjprof

ツールライブラリを結合しません。プロファイラを使用できません。

 注意

tradモードとclangモードではコンパイラが異なるため、翻訳対象のプログラムが同じであっても、tradモードで翻訳した場合とclangモードで翻訳した場合とでは、得られるプロファイル結果が有意に異なることがあります。

2.1.3.2 infoキー fjprof_spawn_dir_name

MPI_COMM_SPAWNルーチン、またはMPI_COMM_SPAWN_MULTIPLEルーチンを実行して動的にプロセスを生成する場合、MPI_Infoに対し以下のinfoキーを設定することで、特定の動的に生成したプロセスのプロファイルデータをfippコマンドの-dオプションに指定したディレクトリとは別のディレクトリへと抽出することができます。fippコマンドの-dオプションについては“2.1.4 プロファイルデータの計測”を参照してください。動的プロセス生成および動的プロセス生成に関連するinfoキーの詳細については“MPI使用手引書”を参照してください。

表2.1 infoキー "fjprof_spawn_dir_name"

infoキー	設定値	説明
fjprof_spawn_dir_name	出力先ディレクトリパス	動的プロセスに対するプロファイルデータの出力先を指定する

 例

infoキー "fjprof_spawn_dir_name"使用例

```
#include "mpi.h"
#include <stdio.h>
#include <stdlib.h>

#define NUM_SPAWNS 1
int main( int argc, char *argv[] )
{
    int np = NUM_SPAWNS;
    int errcodes[NUM_SPAWNS];
    MPI_Comm intercomm;
```

```
    MPI_Info info;

    MPI_Init( &argc, &argv );
    MPI_Info_create(&info);
    MPI_Info_set(info, "fjprof_spawn_dir_name", "./spawn_data");
    MPI_Comm_spawn( "./spawn.out", MPI_ARGV_NULL, np, info, 0, MPI_COMM_WORLD, &intercomm, errcodes );

    MPI_Finalize();
    return 0;
}
```

注意

- ディレクトリを相対パスで指定した場合、プロファイルデータはMPI_COMM_SPAWNルーチンまたはMPI_COMM_SPAWN_MULTIPLEルーチンが呼び出された時点のカレントディレクトリに生成します。また、カレントディレクトリはinfoキー"wdir"の影響を受けます。詳細については“MPI使用手引書”を参照してください。
- infoキー"fjprof_spawn_dir_name"に指定するディレクトリは、fippコマンドの-dオプションに指定するものとは異なるディレクトリを指定してください。同じディレクトリを指定した場合、動作は保証しません。
- 指定したディレクトリが存在する場合、そのディレクトリは空ディレクトリでなければいけません。指定したディレクトリが存在しない場合、新規にディレクトリを作成します。
- infoキー"fjprof_spawn_dir_name"に指定した出力先にプロファイルデータを作成できない場合、MPI_COMM_SPAWNルーチン、またはMPI_COMM_SPAWN_MULTIPLEルーチンを実行したプロセスにおいてsegmentation fault等のエラーが発生することがあります。動的にプロセスを生成するプログラムが異常終了した場合、以下のいずれかのメッセージが出力されていないか確認してください。
  - The files had already existed in the specified value of fjprof_spawn_dir_name key.
  - The specified value of fjprof_spawn_dir_name key is not directory.
  - The specified value of fjprof_spawn_dir_name key is permission denied.

2.1.4 プロファイルデータの計測

fippコマンドを使用して計測します。本操作は計算ノードで実施します。

注意

- fippコマンドで計測したプロファイルデータに対して以下を行った場合、動作保証しません。
  - プロファイルデータの編集
  - プロファイルデータの追加、削除、リネーム
- プロファイルデータの計測中にプログラムが中断した場合、不完全なプロファイルデータが残ることがあります。
- fippコマンドを使用する場合、環境変数“FLIB_FASTOMP”にTRUEを指定してください。指定しない場合、fippコマンドは正しく動作しません。詳細については“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。
- fippコマンドには-Aオプションも指定可能です。ただし、fippコマンドに-Aオプションを指定した場合、“2.1.5プロファイル結果の出力”のfipppxコマンドとして扱うため使用方法が異なります。詳細は“2.1.5プロファイル結果の出力”を参照してください。fippコマンドの-Aオプションと-Cオプションは同時に指定できません。同時に指定した場合、エラーメッセージを出力しプログラムの実行を終了します。

fippコマンドの形式

```
fipp -C -d profile_data [ -I{{call|nocall}|{cpupa|nocpupa}|{mpi|nompi}} ] [ -H[mode={all|user}] ]
[ -L{shared|noshared} ] [ -M{inlined|noinlined} ] [ -P{userfunc|nouserfunc} ] [ -S{all|region} ]
[ -W{spawn|nospawn} ] [ -i interval ] [ -l limit ] [ -m memsize ] exec-file [exec_option ...]
```

fippコマンドのオプション

ポイント

オプションの説明文に「○○できません」「○○でなければいけません」という制限事項がある場合、制限事項に違反するとエラーメッセージを出力して実行を終了します。

相反するオプションを複数指定した場合、最後に指定したオプションが有効となります。例えば、共有ライブラリの詳細情報の計測方法を指定する-L{shared|noshared}オプションを“-Lshared -Lnoshared”の順番で指定した場合、-Lnosharedオプションが有効になります。

-C

プロファイルデータの計測を指示します。本オプションは省略できません。本オプションを指定しない場合、エラーメッセージを出力し収集コマンドの処理を終了します。

-d *profile_data*

プロファイルデータを格納するディレクトリを指定します。本オプションは省略できません。本オプションを指定しない場合、エラーメッセージを出力し収集コマンドの処理を終了します。
*profile_data*は省略できません。*profile_data*にはプロファイルデータを格納するディレクトリ名を相対パスまたは絶対パスで指定します。指定したディレクトリが存在する場合、そのディレクトリは空ディレクトリでなければいけません。指定したディレクトリが存在しない場合、新規にディレクトリを作成します。*profile_data*に“-”から始まるディレクトリ名を指定する場合、絶対パスで指定するかまたはカレントディレクトリ(“./”)を含む相対パスで指定してください。実行中にカレントディレクトリを移動するプログラムを解析する場合、*profile_data*は絶対パスで指定してください。

-I{{call|nocall}|{cpupa|nocpupa}|{mpi|nompi}}

基本プロファイラで計測する項目を指定します。-I{call|nocall}オプション省略時は-Inocallオプションが有効になります。-I{cpupa|nocpupa}オプション省略時は-Icpupaオプションが有効になります。-I{mpi|nompi}オプション省略時の指定はプログラムの種別によって異なります。MPIプログラムの場合、-Impiオプションが有効になります。非MPIプログラムの場合、-Inompiオプションが有効になります。-Icpupaオプションが有効で-Hオプションを指定していない場合、-Hmode=all オプションが有効になります。-Iオプションにはコンマ(,)を区切りとしてサブオプションを複数指定することができます。例えば、-Icall,nocpupaのような指定ができます。

call

コールグラフ情報を計測します。

nocall

コールグラフ情報を計測しません。

cpupa

CPU動作状況を計測します。

nocpupa

CPU動作状況を計測しません。

mpi

MPIコスト情報を計測します。本引数を非MPIプログラムに指定した場合、エラーメッセージを出力して収集コマンドの処理を終了します。

nompi

MPIコスト情報を計測しません。

-H[mode={all|user}]

CPU動作状況の計測内容について指定します。サブオプションmode=にはallまたはuserのどちらか1つを指定します。本オプションまたはサブオプションmode={all|user}省略時は、mode=allが有効になります。-Inocpupaオプションを指定した場合、警告メッセージを出力して本オプションを無効にします。

mode=all

カーネルモードおよびユーザーモードにおける計測を行います。

mode=user

ユーザーモードにおける計測を行います。

-L{shared|noshared}

翻訳時オプション-Nlineまたは-ffj-lineを指定して生成した共有ライブラリ(以降、行情報付き共有ライブラリと呼びます)の計測方法を指定します。翻訳時オプションについては“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。本オプション省略時は-Lnosharedオプションが有効になります。

shared

行情報付き共有ライブラリ内の以下の情報を計測します。

- 手続の開始行番号
- 手続の終了行番号
- ループコスト分布情報
- 行コスト分布情報

noshared

行情報付き共有ライブラリ内の以下の情報を計測しません。

- 手続の開始行番号
- 手続の終了行番号
- ループコスト分布情報
- 行コスト分布情報

-M{inlined|noinlined}

プロファイルデータ計測対象の実行ファイルにおいて、インライン展開された関数のコストを計上する先を切り替えます。

本オプション省略時は、-Mnoinlinedオプションが有効になります。

inlined

インライン展開された関数のコストは、その関数自身のコストとして計上します。

noinlined

インライン展開された関数のコストは、その関数の呼出し元手続のコストとして計上します。このとき、以下の行番号は、インライン展開された関数が記載されているソースコードの行番号となります。

- ループコスト分布情報のループの開始行番号とループの終了行番号
- 行コスト分布情報の行番号

また、ソースコード情報の表示における、インライン展開された関数のコストは、その関数自身のコストとして計上します。

-P{userfunc|nouserfunc}

翻訳時オプション-Nlineまたは-ffj-lineを指定したオブジェクト(以降、行情報付きオブジェクトと呼びます)と翻訳時オプション-Nnolineまたは-ffj-no-lineを指定したオブジェクト(以降、行情報なしオブジェクトと呼びます)が混在している場合の手続コストの計上方法を指定します。標準ライブラリまたは-Lnoshared指定時の共有ライブラリは行情報なしオブジェクトとして扱います。翻訳時オプションについては“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。本オプション省略時は-Pnouserfuncオプションが有効になります。

userfunc

行情報なしオブジェクトの手続にコストが計上された場合、コールグラフ情報から行情報なしオブジェクトの手続を呼び出した手続を遡ります。行情報付きのオブジェクトの手続が存在した場合、その手続にコストを計上します。行情報付きのオブジェクトの手続が存在しなかった場合、コストは計上しません。-Puserfuncオプション指定時は、-Icallオプションも同時に指定しなければなりません。-Icallオプションを指定しない場合、エラーメッセージを出力し収集コマンドの処理を終了します。

nouserfunc

行情報なしオブジェクトの手続にコストが計上された場合、その手続にコストを計上します。ただし、手続開始行、手続終了行は出力しません。

-S{all|region}

プロファイルデータの計測区間を指定します。本オプション省略時は-Sallオプションが有効になります。

all

プログラム全体を計測します。

region

計測区間指定ルーチンで指定した区間内を計測します。ソースコード内に計測区間指定ルーチンを挿入する必要があります。

-W{spawn|nospawn}

動的に生成されたプロセスに対する計測方法を指定します。-W{spawn|nospawn}オプション省略時の指定はプログラムの種別によって異なります。MPIプログラムの場合、-Wspawnオプションが有効になります。非MPIプログラムの場合、-Wnospawnオプションが有効になります。

spawn

動的に生成されたプロセスの統計情報を計測します。特定の動的に生成されたプロセスに対してinfoキー“fjprof_spawn_dir_name”を指定している場合、infoキーを指定した動的に生成されたプロセスのプロファイルデータはinfoキー“fjprof_spawn_dir_name”に指定したディレクトリ内に格納されます。infoキー“fjprof_spawn_dir_name”を指定していない場合、動的に生成されたプロセスのプロファイルデータは-dオプションで指定したディレクトリ内に格納されます。本引数を非MPIプログラムに指定した場合、エラーメッセージを出力して収集コマンドの処理を終了します。

nospawn

動的に生成されたプロセスの統計情報を計測しません。ただし、特定の動的に生成されたプロセスに対してinfoキー“fjprof_spawn_dir_name”を指定している場合、infoキーを指定した動的に生成されたプロセスについては計測が行われ、計測したプロファイルデータはinfoキー“fjprof_spawn_dir_name”に指定したディレクトリ内に格納されます。

-i *interval*

プロファイルデータを計測するサンプリング間隔を指定します。*interval*にはサンプリング間隔を整数値(ミリ秒単位)で指定します。本オプション省略時は-i 100オプションが有効になります。*interval*は省略できません。*interval*には10～3,600,000の範囲の整数値を指定します。*interval*に範囲外の値を指定した場合、警告メッセージを出力して-i 100オプションが有効になります。

-l *limit*

手続情報の計測件数を指定します。出力件数以上の手続情報は、“__other__”として合算して計測します。本オプション省略時は-l 0オプションが有効になります。*limit*は省略できません。*limit*には0～2,147,483,647の範囲の整数値を指定します。*limit*に0を指定した場合、全件を計測します。*limit*に範囲外の値を指定した場合、警告メッセージを出力して-l 0オプションが有効になります。

-m *memsize*

計測時に使用する作業用メモリーサイズを指定します。作業用メモリーの領域はスレッドごとに確保します。本オプション省略時は-m 3000オプションが有効になります。*memsize*には作業用メモリーサイズを整数値(KB単位)で指定します。*memsize*は省略できません。*memsize*には1～2,147,483の範囲の整数値を指定します。*memsize*に範囲外の値を指定した場合、警告メッセージを出力して-m 3000オプションが有効になります。

exec-file [*exec_option* ...]

プロファイルデータの計測対象となる実行ファイルとオプションを指定します。MPIプログラムの場合、mpiexecから指定します。*exec-file*に“-”で始まる実行ファイルを指定する場合、カレントディレクトリ(“./”)を含んだ相対パス、または絶対パスで指定してください。*exec-file*にシェルスクリプトを指定することはできません。実行ファイル名に続く文字列(*exec_option* ...)は、実行ファイルへのオプションと見なします。

例

fippコマンドでCPU動作状況とコールグラフの情報を計測する例

```
fipp -C -d ./tmp -Icall ./a.out
```

2.1.5 プロファイル結果の出力

fippコマンドで計測したプロファイルデータの結果を出力します。本操作は使用するノードによって異なるコマンドを使用します。

ログインノードの場合

fipppxコマンドを使用します。

計算ノードの場合

fippコマンドを使用します。

fipppxコマンドまたはfippコマンドの形式

```
{fipppx|fipp} -A [ -I{{balance|nobalance}|{call|nocall}|{cpupa|nocpupa}|{mpi|nompi}|
{src[:path ]|nosrc}} ] [ -Tt_no ] [ -b{max|total} ] [ -f func_name ] [ -l limit ]
[ -o outfile ] [ -pp_no ] [ -t{csv|text|xml} ] [ -u ] [ -d ] profile_data
```

fipppxコマンドまたはfippコマンドのオプション

ポイント

オプションの説明文に「○○できません」「○○でなければいけません」という制限事項がある場合、制限事項に違反するとエラーメッセージを出力して実行を終了します。

相反するオプションを複数指定した場合、最後に指定したオプションが有効となります。例えば、プロファイル結果の出力対象とする項目を指定する-I{call|nocall}オプションを“ -Icall,nocall”の順番で指定した場合、-Inocallオプションが有効になります。

-A

プロファイル結果の出力処理を指示します。本オプションは省略できません。本オプションを指定しない場合、エラーメッセージを出力し解析コマンドの処理を終了します。

-I{{balance|nobalance}|{call|nocall}|{cpupa|nocpupa}|{mpi|nompi}|{src[:*path*]|nosrc}}

プロファイル結果の出力対象とする項目を指定します。-Iオプションにはコンマ(,)を区切りとしてサブオプションを複数指定することができます。例えば、-Icall,cpupaのような指定ができます。-I{balance|nobalance}オプション省略時は-Inobalanceオプションが有効になります。-I{call|nocall}オプション省略時は-Inocallオプションが有効になります。-I{cpupa|nocpupa}オプション省略時は-Icpupaオプションが有効になります。-I{mpi|nompi}オプション省略時の指定はプロファイルデータ計測時の対象プログラムの種別によって異なります。プロファイルデータの計測対象がMPIプログラムの場合、-Impiオプションが有効になります。プロファイルデータの計測対象が非MPIプログラムの場合、-Inompiオプションが有効になります。-I{src[:*path*]|nosrc}オプション省略時は-Inosrcオプションが有効になります。-Icall,cpupa,mpiの各オプションを指定する場合、fippコマンドで該当する項目を計測している必要があります。計測していない情報を出力対象に指定した場合、エラーメッセージを出力し解析コマンドの処理を終了します。

balance

コスト情報にコストバランス情報(並列実行単位間のコストを比較した情報)を出力します。ただし、逐次プログラムの場合、-tcsvオプションまたは-txmlオプションを指定した場合、-Ibalanceオプションを指定してもコストバランス情報は出力されません。

nobalance

コスト情報にコストバランス情報を出力しません。

call

コールグラフ情報を出力します。

nocall

コールグラフ情報を出力しません。

cpupa

CPU動作状況を出力します。

nocpupa

CPU動作状況を出力しません。

mpi

コスト情報にMPIコスト情報を出力します。

nompi

コスト情報にMPIコスト情報を出力しません。

src[:*path*] ...

ソースコード情報と行ごとのコストを出力します。行ごとのコストには-Impiオプションで出力するコストは含みません。*path*には、ソースコードが存在するディレクトリパスを指定します。*path*を複数指定する場合、コロン(:)で区切って指定してください。*path*を省略した場合、プログラム翻訳時に指定したディレクトリパスを参照します。-tcsvオプションまたは-txmlオプションを指定した場合、-Isrcオプションを指定してもソースコード情報は出力されません。

nosrc

ソースコード情報を出力しません。

-T*t_no*

プロファイルデータの出力対象とするスレッドを指定します。*t_no*には*N*, limit=*n*, allの中からいずれか1つ以上を指定します。本オプション省略時は-Tallオプションが有効になります。*t_no*は省略することができません。-Tオプションにはコンマ(,)を区切りとして*t_no*を複数指定することができます。例えば、-T3,5,limit=10のような指定ができます。

N[,*N*] ...

*N*に指定したスレッド番号の情報を先頭に出力します。*N*に指定したスレッドの情報が存在しない場合、指定を無視します。*N*を複数指定した場合、指定した順番に出力します。

limit=*n*

コストが高い順に*n*件分のスレッド情報を出力します。*n*に0または総スレッド数を超える値を指定した場合、全スレッドの情報を出力します。

all

全スレッドの情報を出力します。-Tlimit=0オプションを指定した場合と同じです。サブオプションlimit=*n*を指定しない場合、こちらが有効になります。

-b{max|total}

プロファイル結果をTEXT形式で出力する場合のコスト情報の出力方法を指定します。-b{max|total}オプション省略時は-btotalオプションが有効になります。本オプションは-ttextオプション指定時のみ有効です。

max

TEXT形式で出力するコスト情報を“最大スレッドコスト基準”で出力します。“最大スレッドコスト基準”の場合、プロセス内で最もコストが高いスレッドの値をそのプロセスのコスト情報として出力します。また、プログラム内で最もコストが高いプロセスの値をそのプログラムのコスト情報として出力します。本サブオプションを指定した場合、-Icpupaオプションおよび-Icallオプションを指定してもCPU動作状況およびコールグラフ情報を出力しません。また、CPU周波数を出力しません。本サブオプションは、-uオプションと同時に指定できません。同時に指定した場合、エラーメッセージを出力し解析コマンドの処理を終了します。

total

TEXT形式で出力するコスト情報を“合計スレッドコスト基準”で出力します。“合計スレッドコスト基準”の場合、プロセス内の全スレッドの合計値を、そのプロセスのコスト情報として出力します。また、プログラム内の全プロセスの合計値をそのプログラムのコスト情報として出力します。

-f *func_name*

特定の手続を出力することを指定します。*func_name*にはプログラムが使用している手続名を指定します。*func_name*のコストが-lオプションで指定した出力件数の範囲外であっても、*func_name*に関する情報を出力します。ただし、以下のいずれかに該当する場合、-f *func_name*を指定しても情報は出力されません。

- fippコマンドで、*func_name*の手続に関する情報を計測していない場合
- *func_name*の手続コストが0の場合

-l *limit*

出力する手続情報の出力件数を指定します。本オプション省略時の値は-t{csv|text|xml}オプションの指定によって異なります。-ttextオプションが有効な場合、-l 10オプションが有効になります。-tcsvオプションまたは-txmlオプションが有効な場合、-l 0オプションが有効になります。*limit*には0～2,147,483,647の範囲の整数値を指定します。*limit*に0を指定した場合、全件を出力します。*limit*に範囲外の値を指定した場合、-l 0オプションが有効になります。

-o *outfile*

プロファイル結果の出力先を指定します。*outfile*には、出力先ファイル名を相対パスまたは絶対パスで指定するか、またはstdoutを指定します。本オプション省略時は-ostdoutオプションが有効になります。*outfile*にstdoutを指定した場合、プロファイル結果を標準出力に出力します。*outfile*に“-”で始まるファイル名を指定する場合、絶対パスまたはカレントディレクトリ(“./”) を含む相対パスで指定してください。

-p*p_no*

プロファイル結果に出力するプロセスを指定します。*p_no*には*N*@*M*, input=*n*, limit=*m*, allの中からいずれか1つ以上を指定します。本オプション省略時の値は-t{csv|text|xml}オプションの指定によって異なります。-ttextオプションが有効な場合、-pinput=0,limit=16オプションが有効になります。-tcsvオプションまたは-txmlオプションが有効な場合、-pallオプションが有効になります。*p_no*は省略することができません。-pオプションにはコンマ(,)を区切りとして*p_no*を複数指定することができます。例えば、-p3,5,limit=10のような指定ができます。

N[@*M*]...

*N*に指定したプロセス番号の情報を先頭に出力します。spawn番号*M*に所属するプロセス番号*N*を指定する場合、*N*@*M*のように指定します。*N*[@*M*]に指定したプロセス番号の情報が存在しない場合、指定を無視します。*N*[@*M*]は複数指定できます。*N*[@*M*]を複数指定した場合、指定した順番に出力します。[@*M*]は-ttextオプションが有効な場合のみ有効になります。

input=*n*

コストが上位*n*件のプロセスの情報を読み込みます。読み込むファイル数が減るため処理が高速になりますが、読み込まなかったプロセスの情報は比率計算をする際の分母に含まれません。*n*に0またはプロセス数を超えた値を指定した場合、全プロセスの情報を読み込みます。本サブオプション省略時はinput=0が有効になります。サブオプションinput=*n*とallを同時に指定した場合、オプションの指定順にかかわらず、サブオプションinput=*n*が有効になります。サブオプションinput=*n*とサブオプションlimit=*m*は同時に指定できます。

limit=*m*

コストが上位*m*件のプロセスの情報を出力します。-ttextオプションが有効な場合、コストが高い順に出力します。出力しなかったプロセスの情報は比率計算をする際の分母に含まれます。*m*に0、または入力となるプロセス数を超える値を指定した場合、全プロセスの情報を出力します。本サブオプション省略時の値は-t{csv|text|xml}オプションの指定によって異なります。-ttextオプションが有効な場合、limit=16が有効になります。-tcsvオプションまたは-txmlオプションが有効な場合、limit=0が有効になります。

all

全プロセスの情報を読み込み、出力します。-ttextオプションが有効な場合、コストが高い順に出力します。-pinput=0,limit=0オプションを指定した場合と同じです。サブオプションinput=*n*および limit=*m*のいずれも指定しない場合、こちらが有効になります。

-t{csv|text|xml}

プロファイル結果の出力形式を指定します。本オプション省略時は-ttextが有効になります。

csv

プロファイル結果をCSV形式で出力します。

text

プロファイル結果をTEXT形式で出力します。

xml

プロファイル結果をXML形式で出力します。

-u

スレッド並列プログラムの生成手続名の出力方法を指定します(生成手続名に関しては“表2.6 スレッド並列プログラムの生成手続名”を参照してください)。本オプションを指定した場合、手続と生成手続のコスト情報を合算し手続名として出力します。本オプションを指定しない場合、手続と生成手続を分けて出力します。本オプションは-tcsvオプションまたは-ttextオプション指定時のみ有効です。本オプションと-bmaxオプションは同時に指定できません。同時に指定した場合、エラーメッセージを出力し解析コマンドの処理を終了します。

-d *profile_data*

プロファイルデータを格納したディレクトリ名を、*profile_data*に相対パスまたは絶対パスで指定します。本オプションは省略できません。ただし、*profile_data*の指定をオプション並びの最後にする場合に限り、“-d”は省略できます。*profile_data*に“-”で始まるディレクトリ名を指定する場合、絶対パス、またはカレントディレクトリ(“./”)を含む相対パスで指定してください。

例

fippコマンドの計測結果./tmp から、CPU動作状況およびMPIコスト情報を出力する例

```
fipppx -A -Inobalance,cpupa,mpi,nocall,nosrc -d ./tmp
```

2.2 プロファイル結果

fipppxコマンドまたはfippコマンドが出力するプロファイル結果の内容について説明します。

2.2.1 プロファイル結果の概要

プロファイル結果は以下の統計情報から構成されています。各情報は、fipppxコマンドまたはfippコマンドの-Iオプションによって出力を制御できます。-Iオプションの詳細については“2.1.5 プロファイル結果の出力”を参照してください。以下の情報を出力します。

- プロファイルデータ計測環境情報
- 時間統計情報
- CPU動作状況
- コスト情報(手続コスト分布情報、ループコスト分布情報、行コスト分布情報)
- コールグラフ情報
- ソースコード情報 (TEXT形式のみ)

詳細については、“2.2.2 プロファイル結果の詳細(TEXT形式)”または“2.2.3 プロファイル結果の詳細(XML形式)”配下を参照してください。

ポイント

基本プロファイラは情報集計レベルApplication、Process、およびThread単位でプロファイル結果を出力しますが、ProcessまたはThreadが1つしかない場合、該当する情報集計レベルの出力を省略します。

2.2.2 プロファイル結果の詳細(TEXT形式)

“2.1.5 プロファイル結果の出力”で-ttextオプションを指定した場合、TEXT形式による出力を行います。TEXT形式のフォーマットについて説明します。

2.2.2.1 プロファイルデータ計測環境情報

プロファイルデータ計測環境情報は、プロファイルデータを計測した時の環境情報を出力します。

プロファイルデータ計測環境情報の出力形式

```
------------------------------------------------------------------------------------------
Fujitsu Instant Performance Profiler Version @vl
   Measured time                : @date
   CPU frequency                : Process      @pno  @frequency (MHz) @sno
   Type of program              : @type
   Average at sampling interval : @interval (ms)
   Measured region              : @region
   Virtual coordinate           : (@x, @y, @z)
------------------------------------------------------------------------------------------
```

表2.2 プロファイルデータ計測環境情報の出力項目

出力項目	出力項目の意味
@vl	プロファイラの版数

出力項目	出力項目の意味
@date	プロファイルデータの計測日時
@pno	プロセス番号
@frequency	CPU周波数 **注意** 以下のいずれかの条件に該当する場合、CPU周波数は“--”を出力します。 ・“2.1.4 プロファイルデータの計測”または“2.1.5 プロファイル結果の出力”で、-Inocpupaオプションを指定した場合 ・“2.1.4 プロファイルデータの計測”で、-Hmode=userオプションを指定した場合 ・“2.1.4 プロファイルデータの計測”で-Sregionオプションを指定し、かつ“2.1.1 計測区間指定ルーチンの追加”で指定した区間の計測が一度も実施されなかった場合 ・“2.1.5 プロファイル結果の出力”で、-bmaxオプションを指定した場合
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・計測対象がMPIプログラムである ・fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・動的に生成されたプロセスである
@type	プログラム実行形態 “SERIAL”：逐次 “Thread (AUTO)”：自動並列のみ “Thread (OpenMP)”：OpenMPのみ “Thread (OpenMP & AUTO)”: OpenMP,自動並列併用 “MPI”：MPIのみ “MPI & Thread (AUTO)”：MPI,自動並列 “MPI & Thread (OpenMP)”：MPI,OpenMP “MPI & Thread (OpenMP & AUTO)”: MPI,OpenMP,自動並列併用
@interval	サンプリング間隔 実際のサンプリング間隔の平均値を出力します。
@region	計測区間種別 All:プログラム全体 Specified:指定した計測指定区間
(@x, @y, @z)	MPIプログラム実行時の論理形状 **ポイント** データ計測対象がMPIプログラムの場合のみ出力します。 また、動的に生成したプロセスの論理形状は出力しません。

ポイント

プロセス(動的に生成されたプロセスを含みます)ごとにCPU周波数が異なる場合、CPU frequency を複数行出力します。ただし、同じCPU周波数のプロセスが連続する場合、CPU frequency は以下のとおり1行にまとめて出力します。

```
CPU frequency                  : Process      @spno - @lpno @frequency (MHz)   (Spawn @ssno - @lsno )
```

連続するプロセスのうち*@spno*には最小のプロセス番号、*@lpno*には最大のプロセス番号が入ります。*@ssno*には最小のspawn番号、*@lsno*には最大のspawn番号が入ります。fipppxコマンドまたはfippコマンドの-pinput=*n*オプションで読み込むプロセス情報を限定している場合、読み込んだプロセスのうち同じCPU周波数の範囲に含まれる最小、および最大のプロセス番号またはspawn番号が入ります。-pinput=*n* オプションについては“2.1.5 プロファイル結果の出力”を参照してください。

2.2.2.2 時間統計情報

時間統計情報は、Application、Process、およびThreadごとにプログラムの経過時間を出力します。また、ApplicationおよびProcessごとにユーザーCPU時間、およびシステムCPU時間を出力します。

時間統計情報の出力形式

```
Time statistics

        Elapsed(s)        User(s)       System(s)
  ------------------------------------------------
          @elapse         @user           @system  @level  @pno  @sno
  ------------------------------------------------
          @elapse         @user           @system  @level  @pno  @sno
```

表2.3 時間統計情報の出力項目

出力項目	出力項目の意味
@elapse	経過時間(s)
@user	ユーザーCPU時間(s) “2.2.2.1 プロファイルデータ計測環境情報”の@type が“SERIAL”または“MPI”以外の場合、または@levelが“Thread”の場合、この出力項目は“--”固定です。
@system	システムCPU時間(s) “2.2.2.1 プロファイルデータ計測環境情報”の@type が“SERIAL”または“MPI”以外の場合、または@levelが“Thread”の場合、この出力項目は“--”固定です。
@level	情報集計レベル(Application、Process、Thread)
@pno	プロセス番号またはスレッド番号
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがProcessである ・ 動的に生成されたプロセスである

2.2.2.3 CPU動作状況

CPU動作状況は、メモリースループット、命令数、演算数などCPUの動作状況に関係する情報を表示します。この情報は、-Icpupaオプションが有効な場合に出力します。CPU動作状況は次の単位で集計して出力します。

- Application
- Process
- Thread

CPU動作状況の出力形式

```
Performance monitor event : statistics

  ********************************************************************************
     @level  @pno  @sno    - performance monitors
  ********************************************************************************

    Execution                      Floating-point  Mem throughput  Mem throughput
          time(s)          GFLOPS   peak ratio(%)          (GB/s)   peak ratio(%)
  --------------------------------------------------------------------------------
            @time          @value1         @value2         @value3         @value4   @level   @pno   @sno
  --------------------------------------------------------------------------------
            @time          @value1         @value2         @value3         @value4   @level   @pno   @sno

    Effective       Floating-point  SIMD inst.      SVE operation
       instruction       operation         rate(%)         rate(%)
  --------------------------------------------------------------------------------
          @value5         @value6         @value7         @value8                   @level   @pno   @sno
  --------------------------------------------------------------------------------
          @value5         @value6         @value7         @value8                   @level   @pno   @sno

              IPC             GIPS
  --------------------------------------------------------------------------------
          @value9        @value10                                                   @level   @pno   @sno
  --------------------------------------------------------------------------------
          @value9        @value10                                                   @level   @pno   @sno
```

表2.4 CPU動作状況の出力項目

出力項目	出力項目の意味
@level	情報集計レベル(Application、Process、Thread)
@pno	プロセス番号またはスレッド番号
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがProcessである ・ 動的に生成されたプロセスである
@time	計測対象区間の命令実行に要した時間(秒) 最上段の値は、各スレッドにおける計測対象区間の命令実行の最長時間です。
@value1	1秒あたりに実行された浮動小数点演算数 Process: 最上段の値は、そのプロセスにおける1秒あたりに実行された浮動小数点演算の総数です。 Application: 最上段の値は、そのアプリケーションにおける1秒あたりに実行された浮動小数点演算の総数です。 **注意** GFLOPS値はすべてActive elementとして算出しています。そのため、Inactive elementの多いプログラムでは本来のGFLOPS値より高い値を出力します。
@value2	浮動小数点演算性能の理論値に対する実測値の比率(%)

出力項目	出力項目の意味
	Process: 最上段の値は、浮動小数点演算性能の理論値に対するそのプロセスの@value1の比率です。 Application: 最上段の値は、浮動小数点演算性能の理論値に対するそのアプリケーションの@value1の比率です。 **注意** 浮動小数点演算性能の理論値は倍精度演算として算出しています。そのため、単精度や半精度の演算の場合、実際の割合の2倍や4倍の値を出力します。
@value3	メモリースループット(GB/s) Process: 最上段の値は、そのプロセスにおける@timeに対するメモリーアクセスの総量です。 Application: 最上段の値は、そのアプリケーションにおける@timeに対するメモリーアクセスの総量です。
@value4	メモリースループットの理論値に対する実測値の比率(%) Process: 最上段の値は、メモリースループットの理論値に対するそのプロセスにおける@value3の比率です。 Application: 最上段の値は、メモリースループットの理論値に対するそのアプリケーションにおける@value3の比率です。
@value5	実行された命令の総数 Process: 最上段の値は、そのプロセスで実行された命令の総数です。 Application: 最上段の値は、そのアプリケーションで実行された命令の総数です。 **注意** 実行された命令の総数には、MOVPRFX命令を含みません。
@value6	実行された浮動小数点演算の総数 Process: 最上段の値は、そのプロセスで実行された浮動小数点演算の総数です。 Application: 最上段の値は、そのアプリケーションで実行された浮動小数点演算の総数です。
@value7	実行された命令の総数に対するSIMD命令数の比率(%) Process: 最上段の値は、そのプロセスにおける@value5に対する実行されたSIMD命令の総数の比率です。 Application: 最上段の値は、そのアプリケーションにおける@value5に対する実行されたSIMD命令の総数の比率です。
@value8	実行された浮動小数点演算の総数に対するSVE演算数の比率(%) Process: 最上段の値は、そのプロセスにおける@value6に対するSVE演算の総数の比率です。 Application: 最上段の値は、そのアプリケーションにおける@value6に対するSVE演算の総数の比率です。
@value9	1サイクルあたりに実行された命令数 Process: 最上段の値は、そのプロセスにおけるサイクルの総数で@value5を除算した値です。 Application: 最上段の値は、そのアプリケーションにおけるサイクルの総数で@value5を除算した値です。
@value10	1秒あたりに実行された命令数 Process: 最上段の値は、そのプロセスにおける@timeで@value5を除算した値です。 Application: 最上段の値は、そのアプリケーションにおける@timeで@value5を除算した値です。

注意

@timeおよび各@valueの出力結果は12桁以内を想定しています。そのため、出力結果が13桁を超える場合、見出しと出力結果にズレが生じます。

2.2.2.4 コスト情報

コスト情報は、以下の情報で構成されています。

- 手続コスト分布情報
- ループコスト分布情報
- 行コスト分布情報

並列プログラムの場合には、手続コスト分布情報およびループコスト分布情報にコストバランス情報を出力することができます。コスト情報の出力形式として“合計スレッドコスト基準”または“最大スレッドコスト基準”のいずれかをfippppxコマンドの-bオプションによって指定します。詳細については“2.1.5 プロファイル結果の出力”を参照してください。

2.2.2.4.1 手続コスト分布情報

手続コスト分布情報は、Application、Process、およびThreadごとに、手続コスト、スレッド間同期待ちコスト、手続の開始行番号、手続の終了行番号、および手続名を出力します。手続コスト分布情報は常に出力します。MPIコスト情報は、-Impiオプションが有効な場合のみ出力します。コストバランス情報は-Ibalanceオプションが有効な場合にのみ出力します。

手続コスト分布情報の出力形式(合計スレッドコスト基準)

常に出力

```
Procedures profile (Total thread cost basis)

   ********************************************************************************************
       @level @pno @sno - procedures
         Application and Process outputs the total value of the cost of each thread.
         Procedure outputs the total value of the procedure cost of each thread.
   ********************************************************************************************
              Cost          %  Operation (s)        Barrier              %    Start      End
   --------------------------------------------------------------------------------------------
            @cost @cost-rate           @ope        @barrier @barrier-rate       --       --   @level  @pno  @sno
   --------------------------------------------------------------------------------------------
            @cost @cost-rate           @ope        @barrier @barrier-rate   @start     @end   @name
```

-Impiオプション有効時に追加出力

```
              MPI           %  Communication (s)    Start      End
   ------------------------------------------------------------
            @mpi   @mpi-rate             @comm       --       --   @level  @pno  @sno
   ------------------------------------------------------------
            @mpi   @mpi-rate             @comm   @start     @end   @name
```

-Ibalanceオプション有効時に追加出力

```
   __                          __
   __  Parallel balance of cost  __

   @name  @start - @end
      +----------------------+----------------------+
      |                      |                      |   @balance  @cost2  @pno  @sno
      +----------------------+----------------------+
```

手続コスト分布情報の出力形式(最大スレッドコスト基準)

常に出力

```
Procedures profile (Max thread cost basis)

   ********************************************************************************************
       @level @pno @sno - procedures
         Application and Process outputs the max value of the cost of each thread.
         Procedure outputs the max value of the procedure cost of each thread.
```

```
  ********************************************************************************

 Spawn  Process   Thread     Cost          %  Operation (s)   Barrier            %    Start      End
--------------------------------------------------------------------------------------------------------
@spawn @process  @thread    @cost @cost-rate          @ope  @barrier @barrier-rate       --       --   @level @pno @sno
--------------------------------------------------------------------------------------------------------
@spawn @process  @thread    @cost @cost-rate          @ope  @barrier @barrier-rate   @start     @end   @name
```

-Impiオプション有効時に追加出力

```
 Spawn  Process   Thread          MPI          %  Communication (s)    Start      End
-------------------------------------------------------------------------------------
@spawn @process  @thread        @mpi  @mpi-rate             @comm       --       --   @level    @pno @sno
-------------------------------------------------------------------------------------
@spawn @process  @thread        @mpi  @mpi-rate             @comm   @start     @end   @name
```

-Ibalanceオプション有効時に追加出力

```
__                  __
__  Parallel balance of cost  __

@name  @start - @end
   +-----------------------+-----------------------+
   |                       |                       |   @balance  @cost2  @pno  @sno
   +-----------------------+-----------------------+
```

表2.5 手続コスト分布情報の出力項目

出力項目	出力項目の意味
@level	情報集計レベル(Application、Process、Thread)
@pno	プロセス番号またはスレッド番号
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがProcessである ・ 動的に生成されたプロセスである
@spawn	最大コストとして選ばれたspawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。条件を満たさない場合、“--”を出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがProcessである ・ 動的に生成されたプロセスである
@process	最大コストとして選ばれたプロセス番号
@thread	最大コストとして選ばれたスレッド番号
@cost	手続のコスト(@barrierのコストを含む)
@cost-rate	@levelまたは@nameのコストが情報集計レベルのコストに占める割合(%)

出力項目	出力項目の意味
@ope	演算時間(s)
@barrier	スレッド間同期待ちコスト **注意** スレッド並列プログラム以外の場合、本項目は“--”固定です
@barrier-rate	スレッド間同期待ちコストが情報集計レベルのコストに占める割合(%) **注意** スレッド並列プログラム以外の場合、本項目は“--”固定です
@mpi	MPIコスト
@mpi-rate	MPIコストが情報集計レベルのコストに占める割合(%)
@comm	通信時間(s)
@start	手続の開始行番号
@end	手続の終了行番号
@name	手続名
@balance	手続のプロセス間またはスレッド間コストバランス(%) @balance = (@cost2 - (@cost2の平均値)) / (@cost2の平均値) * 100 @balanceは±100%を超えることがあります。 @balanceが4桁以上の場合表示カラムがずれます。
@cost2	プロセスまたはスレッドの手続コスト(スレッドバリアコストを除く)

スレッド並列プログラムの場合、並列化された部分を生成手続の情報として出力します。生成手続名には、手続名の後ろに生成手続の種別に応じた識別子を付加します。C言語/C++言語の場合、翻訳時に使用したコンパイラのモードによって生成手続名が異なります。詳細については、“C言語使用手引書”または“C++言語使用手引書”の内部関数名に関する記事を参照してください。生成手続の種別ごとのスレッド並列プログラム生成手続名を下表に示します。

表2.6 スレッド並列プログラムの生成手続名

言語種別	生成手続の種別	生成手続名
Fortran	自動並列手続	*手続名*._PRL_*数字*_
	OpenMP並列手続	*手続名*._OMP_*数字*_
	TASK構文	*手続名*._TSK_*数字*_
C言語/C++言語 (tradモード)	自動並列手続	*手続名*._PRL_*数字*
	OpenMP並列手続	*手続名*._OMP_*数字*
	TASK構文	*手続名*._TSK_*数字*
C言語/C++言語 (clangモード)	OpenMP並列手続	*手続名*.omp_outlined._debug__*数字* (注1)
	TASK構文	*手続名*.omp_task_entry.*数字*

注1:clangモードには自動並列手続はありません。また、“_debug__”は、翻訳時オプション-gが指定されている場合に付加します。

参考

TASK構文のコスト情報は、集計レベル「Application」の手続コスト分布情報から確認することができます。また、TASK構文のスレッドへの散らばり具合は、情報集計レベル「Thread」の手続コスト分布情報またはコストバランス情報から確認することができます。ただし、基本プ

ロファイラによる手続情報の出力件数はデフォルトでは上位10件となっており、コストの小さいTASK構文情報を出力しない可能性があります。そのため、TASK構文のコストを確認する場合は“2.1.5 プロファイル結果の出力”の-lオプションを使用して、手続情報の出力件数を変更することを推奨します。

2.2.2.4.2 ループコスト分布情報

ループコスト分布情報は、Application、Process、およびThreadごとに、ループコスト、スレッド間同期待ちコスト、ネストレベル、ループ種別、ループの翻訳種別、ループの開始行番号、ループの終了行番号、およびループの属する手続名を出力します。ループコスト分布情報は常に出力します。MPIコスト情報は、-Impiオプションが有効な場合のみ出力します。コストバランス情報は-Ibalanceオプションが有効な場合にのみ出力します。

ループコスト分布情報の出力形式(合計スレッドコスト基準)

常に出力

```
Loops profile (Total thread cost basis)

   ********************************************************************************
       @level @pno @sno - loops
         Application and Process outputs the total value of the cost of each thread.
         Procedure outputs the total value of the loop cost of each thread.
   ********************************************************************************

    Cost           %  Operation (s)    Barrier              %    Nest    Kind     Exec    Start      End
   ------------------------------------------------------------------------------------------------------
    @cost @cost-rate          @ope    @barrier @barrier-rate      --      --       --      --       --  @level  @pno  @sno
   ------------------------------------------------------------------------------------------------------
    @cost @cost-rate           @ope    @barrier @barrier-rate   @nest   @kind    @exec   @start     @end   @name
```

-Impiオプション有効時に追加出力

```
              MPI          %  Communication (s)    Nest    Kind     Exec    Start      End
   ------------------------------------------------------------------------------------
             @mpi  @mpi-rate             @comm      --    --       --        --       --   @level  @pno  @sno
   ------------------------------------------------------------------------------------
             @mpi  @mpi-rate             @comm   @nest   @kind    @exec   @start     @end   @name
```

-Ibalanceオプション有効時に追加出力

```
   __                          __
   __  Parallel balance of cost  __

   @name  @start - @end
      +-----------------------+-----------------------+
      |                       |                       |   @balance  @cost2 @pno  @sno
      +-----------------------+-----------------------+
```

ループコスト分布情報の出力形式(最大スレッドコスト基準)

常に出力

```
Loops profile (Max thread cost basis)

   ********************************************************************************
       @level @pno @sno - loops
         Application and Process outputs the max value of the cost of each thread.
         Procedure outputs the max value of the loop cost of each thread.
   ********************************************************************************

 Spawn  Process   Thread   Cost          %  Operation (s)    Barrier              %  Nest  Kind  Exec  Start  End
-------------------------------------------------------------------------------------------------------------------
@spawn @process  @thread  @cost @cost-rate          @ope   @barrier @barrier-rate    --    --    --     --   --  @level
@pno @sno
```

```
-------------------------------------------------------------------------------------------------------------------
@spawn @process  @thread  @cost @cost-rate          @ope   @barrier @barrier-rate @nest @kind @exec @start @end   @name
```

-Impiオプション有効時に追加出力

```
 Spawn  Process   Thread    MPI          %  Communication (s)  Nest  Kind   Exec    Start    End
-------------------------------------------------------------------------------------------------------------
@spawn @process  @thread   @mpi  @mpi-rate             @comm    --    --     --       --     --   @level    @pno @sno
-------------------------------------------------------------------------------------------------------------
@spawn @process  @thread   @mpi  @mpi-rate             @comm @nest @kind  @exec   @start   @end   @name
```

-Ibalanceオプション有効時に追加出力

```
__                          __
__  Parallel balance of cost  __

 @name  @start - @end
    +------------------------+------------------------+
    |                        |                        |   @balance  @cost2  @pno  @sno
    +------------------------+------------------------+
```

表2.7 ループコスト分布情報の出力項目

出力項目	出力項目の意味
@level	情報集計レベル(Application、Process、Thread)
@pno	プロセス番号またはスレッド番号
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがProcessである ・ 動的に生成されたプロセスである
@spawn	最大コストとして選ばれたspawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。条件を満たさない場合、“--”を出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがProcessである 動的に生成されたプロセスである
@process	最大コストとして選ばれたプロセス番号
@thread	最大コストとして選ばれたスレッド番号
@cost	ループのコスト（@barrierのコストを含む）
@cost-rate	@levelまたは@nameのコストが情報集計レベルのコストに占める割合(%)
@ope	演算時間(s)
@barrier	スレッド間同期待ちコスト

出力項目	出力項目の意味
	注意 スレッド並列プログラム以外の場合、本項目は“--”固定です。
@barrier-rate	スレッド間同期待ちコストが情報集計レベルのコストに占める割合(%) 注意 スレッド並列プログラム以外の場合、本項目は“--”固定です。
@mpi	MPIコスト
@mpi-rate	MPIコストが情報集計レベルのコストに占める割合(%)
@comm	通信時間(s)
@nest	ネストレベル
@kind	ループ種別(DO、WHILE、UNTIL、ARRAY、FOR、GOTO、OTHER)
@exec	ループの翻訳種別(SERIAL:逐次、OpenMP:OpenMP、AUTO:自動並列)
@start	ループの開始行番号
@end	ループの終了行番号
@name	手続名
@balance	ループのプロセス間またはスレッド間コストバランス(%) @balance = (@cost2 - (@cost2の平均値)) / (@cost2の平均値) * 100 @balanceは±100%を超えることがあります。 @balanceが4桁以上の場合表示カラムがずれます。
@cost2	プロセスまたはスレッドのループコスト(スレッドバリアコストを除く)

2.2.2.4.3 行コスト分布情報

行コスト分布情報は、Application、Process、およびThreadごとに、行コスト、行番号、および行の属する手続名を出力します。行コスト分布情報は常に出力します。MPIコスト情報は、-Impiオプションが有効な場合のみ出力します。

行コスト分布情報の出力形式(合計スレッドコスト基準)

常に出力

```
Lines profile (Total thread cost basis)

   ********************************************************************************
       @level @pno @sno - lines
         Application and Process outputs the total value of the cost of each thread.
         Procedure outputs the total value of the line cost of each thread.
   ********************************************************************************

              Cost          %  Operation (s)        Barrier              %    Line
   ------------------------------------------------------------------------------
            @cost @cost-rate           @ope        @barrier @barrier-rate      --   @level  @pno  @sno
   ------------------------------------------------------------------------------
            @cost @cost-rate           @ope        @barrier @barrier-rate   @line   @name
```

-Impiオプション有効時に追加出力

```
              MPI          %  Communication (s)    Line
   ------------------------------------------------------
```

```
          @mpi   @mpi-rate             @comm      --   @level  @pno  @sno
-------------------------------------------------
          @mpi   @mpi-rate             @comm   @line   @name
```

行コスト分布情報の出力形式(合計スレッドコスト基準)

常に出力

```
Lines profile (Max thread cost basis)

  ********************************************************************************
      @level @pno @sno - lines
        Application and Process outputs the max value of the cost of each thread.
        Procedure outputs the max value of the line cost of each thread.
  ********************************************************************************

      Spawn  Process   Thread     Cost          %  Operation (s)     Barrier             %     Line
  ------------------------------------------------------------------------------------------------------
     @spawn @process  @thread    @cost @cost-rate          @ope    @barrier @barrier-rate       --   @level  @pno @sno
  ------------------------------------------------------------------------------------------------------
     @spawn @process  @thread    @cost @cost-rate          @ope    @barrier @barrier-rate    @line   @name
```

-Impiオプション有効時に追加出力

```
            MPI          %  Communication (s)     Line
-------------------------------------------------
          @mpi  @mpi-rate             @comm       --   @level    @pno @sno
-------------------------------------------------
          @mpi  @mpi-rate             @comm    @line   @name
```

表2.8 行コスト分布情報の出力項目

出力項目	出力項目の意味
@level	情報集計レベル(Application、Process、Thread)
@pno	プロセス番号またはスレッド番号
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがProcessである ・ 動的に生成されたプロセスである
@spawn	最大コストとして選ばれたspawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。条件を満たさない場合、“--”を出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがProcessである 動的に生成されたプロセスである
@process	最大コストとして選ばれたプロセス番号

出力項目	出力項目の意味
@thread	最大コストとして選ばれたスレッド番号
@cost	行のコスト
@cost-rate	@levelまたは@nameのコストが情報集計レベルのコストに占める割合(%)
@ope	演算時間(s)
@barrier	スレッド間同期待ちコスト 注意 スレッド並列プログラム以外の場合、本項目は“--”固定です
@barrier-rate	スレッド間同期待ちコストが情報集計レベルのコストに占める割合(%) 注意 スレッド並列プログラム以外の場合、本項目は“--”固定です
@mpi	MPIコスト
@mpi-rate	MPIコストが情報集計レベルのコストに占める割合(%)
@comm	通信時間(s)
@line	行番号
@name	手続名

2.2.2.5 コールグラフ情報

コールグラフ情報は、手続の呼出し経路および手続の呼出し経路ごとのコストを出力します。

この情報は、-Icallオプションが有効な場合に出力します。

コールグラフ情報の出力形式

```
Call graph

    Process @pno  @sno  - Thread @thno
    --------------------+
                        | @rate % <@nest> @name [@cost/@accumulation]
```

表2.9 コールグラフ情報の出力項目

出力項目	出力項目の意味
@pno	プロセス番号
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ 動的に生成されたプロセスである
@thno	スレッド番号
@rate	手続コストがスレッド全体コストに占める割合(%)

出力項目	出力項目の意味
@nest	手続呼出しのネストレベル
@name	手続名
@cost	手続のコスト
@accumulation	呼出し先手続のコストを含めた手続のコスト

注意

- 入出力文の処理を実行中に基本プロファイラのサンプリングによる割込みが発生した場合、コールグラフ情報を正しく出力しない場合があります。
- コールグラフ情報のネストレベルに“<???>”と出力された場合、以下のいずれかに該当します。
  - 手続の呼出し経路が不明である
  - 手続呼出しのネストレベルが128以上である
  - 最適化により呼び出し経路が辿れなくなった

2.2.2.6 ソースコード情報

ソースコード情報は、ソースコードの各行にコストを付加して出力します。この情報は、-Iscrcオプションが有効な場合に出力します。

ソースコード情報の出力形式

```
Sources profile
  -----> @ file-name
    ------------------------------------------------------------------------
      Line           Costs
    ------------------------------------------------------------------------
     @line          @cost     @source-code
```

表2.10 ソースコードの出力項目

出力項目	出力項目の意味
@file-name	ソースコードファイル名
@line	行番号
@cost	行のコスト
@source-code	ソースコード

注意

-Mnoinlinedオプションを使用した場合、ソースコード情報の表示において、インライン展開された関数と呼出し元の手続きのうち、どちらか一方、または両方が表示されない場合があります。

2.2.3 プロファイル結果の詳細(XML形式)

“2.1.5 プロファイル結果の出力”で-txmlオプションを指定した場合、XML形式による出力を行います。XML形式のフォーマットについて説明します。

2.2.3.1 XML形式の構成

プロファイラが出力するXML形式出力の構成について説明します。

プロファイラが出力するXML形式出力は全体を<profile>要素で囲みます。<profile>要素は<environment>および<information>の2つの要素から構成されます。

XML形式の構成

```
<?xml version="1.0" encoding="utf-8"?>                          XML宣言
<profile type="@type" version="@vid" output_version="@oid">     プロファイル情報
  <environment>                                                 プロファイルデータ計測環境情報
    ......
  </environment>
  <information item="@item">                                    性能情報
    ......
  </information>
  <information item="@item">                                    性能情報
    ......
  </information>
  ......
</profile>
```

要素の説明

要素名	要素の説明	概要
profile	プロファイル情報	プロファイラが出力したXML形式出力であることを示します。
environment	プロファイルデータ計測環境情報	TEXT形式の以下に相当する内容を格納します(※)。本要素は一つだけ出力します。 ・“2.2.2.1 プロファイルデータ計測環境情報”
information	性能情報	TEXT形式の以下に相当する内容を格納します(※)。本要素は複数回出力します。 ・“2.2.2.2 時間統計情報” ・“2.2.2.3 CPU動作状況” ・“2.2.2.4 コスト情報” ・“2.2.2.5 コールグラフ情報”

※ TEXT形式とXML形式では一致しない項目も存在します。

2.2.3.2 XML形式出力の詳細

XML形式出力で使用する各要素について説明します。

2.2.3.2.1 プロファイル情報 <profile>

プロファイラが出力したXML形式出力であることを示します。

要素名	説明
profile	<profile type="@type" version="@vid" output_version="@oid"> </profile> プロファイラが出力したXML形式出力であることを示します。 @type にはプロファイラの種別が入ります。“fipp”固定です。 @vid にはプロファイラの版数が入ります。 @oid には出力フォーマットの版数が入ります。

2.2.3.2.2 プロファイルデータ計測環境情報 <environment>

プロファイルデータを計測した時の環境情報を出力します。

フォーマット

```
<environment>
  <measured_time unit="date">@date</measured_time>
  <type_of_program program="@program"/>
  <measured_region region="@region"/>
  <coordinate x="@x" y="@y" z="@z"/>
  <spawn id="@id">
    <process id="@id">
      <sampling_interval unit="ms">@interval</sampling_interval>
      <frequency unit="MHz">@frequency</frequency>
    </process>
  </spawn>
</environment>
```

要素の説明

要素名	説明
environment	<environment> </environment> プロファイルデータの計測環境情報であることを示します。
measured_time	<measured_time unit="date"> @date </measured_time> プロファイルデータの計測日時を示します。 @date にはプロファイリングデータの計測日時が YYYY-MM-DDThh:mm:ss 形式で入ります。
type_of_program	<type_of_program program="@program "/> プログラム実行形態を示します。 @program には以下のいずれかが入ります。 “SERIAL”：逐次 “Thread(AUTO)”：自動並列のみ “Thread(OpenMP)”：OpenMPのみ “Thread(OpenMP+AUTO)”: OpenMP,自動並列併用 “MPI”：MPIのみ “MPI+Thread(AUTO)”：MPI,自動並列 “MPI+Thread(OpenMP)”：MPI,OpenMP “MPI+Thread(OpenMP+AUTO)”: MPI,OpenMP,自動並列併用
measured_region	<measured_region region="@region " /> 基本プロファイラの計測区間種別を示します。 @region には以下のどちらかが入ります。 All:プログラム全体 Specified:指定した計測指定区間
coordinate	<coordinate x="@x " y="@y " z="@z " /> MPIプログラム実行時の論理形状を示します。 @x 、@y 、@z にはそれぞれx軸、y軸、z軸の値が入ります。 **ポイント** データ計測対象がMPIプログラムの場合のみ出力します。 また、動的に生成したプロセスの論理形状は出力しません。
spawn	<spawn id="@id"> </spawn> spawn番号を示します。本要素は動的プロセスが存在する場合、複数回出力することがあります。

要素名	説明
	@id には、spawn番号が入ります。動的プロセスではない場合、@id には0が入ります。
process	<process id="@id"> </process> プロセス番号を示します。本要素はプロセスが複数存在する場合、複数回出力します。 @id にはプロセス番号が入ります。
sampling_interval	<sampling_interval unit="ms"> @interval </sampling_interval> サンプリング間隔を出力します。単位はミリ秒(unit="ms")です。 @interval にはサンプリング間隔を出力します。
frequency	<frequency unit="MHz"> @frequency </frequency> CPU周波数を出力します。単位はMHz(unit="MHz")です。 @frequency にはCPU周波数を出力します。 **注意** 以下のいずれかの条件に該当する場合、CPU周波数は“-”を出力します。 ・“2.1.4 プロファイルデータの計測”で、-Inocpupaオプションを指定した場合 ・“2.1.4 プロファイルデータの計測”で、-Hmode=userオプションを指定した場合 ・“2.1.4 プロファイルデータの計測”で-Sregionオプションを指定し、かつ“2.1.1 計測区間指定ルーチンの追加”で指定した区間の計測が一度も実施されなかった場合 ・“2.1.5 プロファイル結果の出力”で、-bmaxオプションを指定した場合

2.2.3.2.3 性能情報 <information>

プロファイラの性能情報を出力します。性能情報は@item 属性によって出力内容が異なります。詳細についてはinformation要素の説明を参照してください。本項では、性能情報共通で使用する要素について説明します。

フォーマット

```
<information item="@item">
  <spawn id="@id">
    <process id="@id">
      <thread id="@id">
          ......
      </thread>
    </process>
  </spawn>
</information>
```

要素の説明

要素名	説明
information	<information item="@item"> </information> 性能情報であることを示します。 @item には出力する性能情報にあわせて以下のいずれかを出力します。出力する性能情報が複数存在する場合、information要素を複数回出力します。@item の値によって“フォーマット”の青字部分に出力する要素が異なります。 time:“2.2.3.2.4 時間統計情報 <time>”の情報を出力します。 cpupa:“2.2.3.2.5 CPU動作状況 <cpupa>”の情報を出力します。 cost:“2.2.3.2.6 コスト情報 <cost> ”の情報を出力します。

要素名	説明
	call:“2.2.3.2.7 コールグラフ情報 <call>”の情報を出力します。
spawn	<spawn id="@id"> </spawn> spawn番号を示します。本要素は動的プロセスが存在する場合、複数回出力することがあります。 @id には、spawn番号が入ります。動的プロセスではない場合、@id には0が入ります。
process	<process id="@id"> </process> プロセス番号を示します。本要素はプロセスが複数存在する場合、複数回出力します。 @id にはプロセス番号が入ります。
thread	<thread id="@id"> </thread> スレッド番号を示します。本要素はスレッドが複数存在する場合、複数回出力します。 @id にはスレッド番号が入ります。

2.2.3.2.4 時間統計情報 <time>

スレッド単位のプログラムの経過時間を出力します。information要素の@item 属性が“time”の場合に出力します。information要素については“2.2.3.2.3 性能情報 <information>”を参照してください。プログラム単位、プロセス単位の情報は出力しません。

フォーマット

```
<time>
  <elapsed unit="s">@elapse</elapsed>
  <user unit="s">@user</user>
  <system unit="s">@system</system>
</time>
```

要素の説明

要素名	説明
time	<time> </time> 時間統計情報であることを示します。
elapsed	<elapsed unit="s" > @elapse </elapsed> スレッドごとの経過時間を示します。単位は秒(unit="s")です。 @elapse にはスレッドごとの経過時間が入ります。
user	<user unit="s"> @user </user> スレッドごとのユーザーCPU時間を示します。単位は秒(unit="s")です。本要素はtype_of_program要素の@program 属性が“SERIAL”または“MPI”の場合のみ出力します。type_of_program要素については“2.2.3.2.2 プロファイルデータ計測環境情報 <environment>”を参照してください。 @user にはスレッドごとのユーザーCPU時間が入ります。
system	<system unit="s"> @system </system> スレッドごとのシステムCPU時間を示します。単位は秒(unit="s")です。本要素はtype_of_program要素の@program 属性が“SERIAL”または“MPI”の場合のみ出力します。type_of_program要素については“2.2.3.2.2 プロファイルデータ計測環境情報 <environment>”を参照してください。 @system にはスレッドごとのシステムCPU時間が入ります。

2.2.3.2.5 CPU動作状況 <cpupa>

スレッド単位のメモリースループット、命令数、演算数などCPUの動作状況に関係する情報を出力します。information要素の@item 属性が“cpupa”の場合に出力します。information要素については“2.2.3.2.3 性能情報 <information>”を参照してください。プログラム単位、プロセス単位の情報は出力しません。この情報は、-Icpupaオプションが有効な場合に出力します。

フォーマット

```
<cpupa>
  <execution_time unit="s">@execution_time</execution_time>
  <gflops unit="GFLOPS">@gflops</gflops>
  <floating_point_peak_ratio unit="%">@floating_point_peak_ratio</floating_point_peak_ratio>
  <mem_throughput unit="GB/s">@mem_throughput</mem_throughput>
  <mem_throughput_peak_ratio unit="%">@mem_throughput_peak_ratio</mem_throughput_peak_ratio>
  <effective_instruction unit="instructions">@effective_instruction</ffective_instruction>
  <floating_point_operation unit="operations">@floating_point_operation</floating_point_operation>
  <simd_inst_rate unit="%">@simd_inst_rate</simd_inst_rate>
  <sve_operation_rate unit="%">@sve_operation_rate</sve_operation_rate>
  <ipc unit="IPC">@ipc</ipc>
  <gips unit="GIPS">@gips</gips>
</cpupa>
```

要素の説明

要素名	説明
cpupa	<cpupa> </cpupa> CPU動作状況であることを示します。
execution_time	<execution_time unit="s"> @execution_time </execution_time> 計測対象区間の命令実行に要した時間を示します。単位は秒(unit="s")です。 @execution_time には計測対象区間の命令実行に要した時間が入ります。
gflops	<gflops unit="GFLOPS"> @gflops </gflops> 1秒あたりに実行された浮動小数点演算数を示します。単位はGFLOPS(unit="GFLOPS")です。 @gflops には1秒あたりに実行された浮動小数点演算数が入ります。 **注意** GFLOPS値はすべてActive elementとして算出しています。そのため、Inactive elementの多いプログラムでは本来のGFLOPS値より高い値を出力します。
floating_point_peak_ratio	<floating_point_peak_ratio unit="%"> @floating_point_peak_ratio </floating_point_peak_ratio> 浮動小数点演算性能の理論値に対する実測値の比率を示します。単位は%(unit="%")です。 @floating_point_peak_ratio には浮動小数点演算性能の理論値に対する実測値の比率が入ります。 **注意** 浮動小数点演算性能の理論値は倍精度演算として算出しています。そのため、単精度や半精度の演算の場合、実際の割合の2倍や4倍の値を出力します。
mem_throughput	<mem_throughput unit="GB/s"> @mem_throughput </mem_throughput> メモリースループットを示します。単位はGB/秒(unit="GB/s")です。 @mem_throughput にはメモリースループットが入ります。
mem_throughput_peak_ratio	<mem_throughput_peak_ratio unit="%"> @mem_throughput_peak_ratio </ mem_throughput_peak_ratio> メモリースループットの理論値に対する実測値の比率を示します。単位は%(unit="%")です。 @mem_throughput_peak_ratio にはメモリースループットの理論値に対する実測値の比率が入ります。
effective_instruction	<effective_instruction unit="instructions"> @effective_instruction </effective_instruction>

要素名	説明
	実行された命令の総数を示します。単位は命令数(unit="instructions")です。 @effective_instruction には実行された命令の総数が入ります。 **注意** 実行された命令の総数には、MOVPRFX命令を含みません。
floating_point_operation	<floating_point_operation unit="operations"> @floating_point_operation </floating_point_operation> 実行された浮動小数点演算の総数を示します。単位は演算数(unit=" operations")です。 @floating_point_operation には実行された浮動小数点演算の総数が入ります。
simd_inst_rate	<simd_inst_rate unit="%"> @simd_inst_rate </simd_inst_rate> 実行された命令の総数に対するSIMD命令数の比率を示します。単位は%(unit="%")です。 @simd_inst_rate には実行された命令の総数に対するSIMD命令数の比率が入ります。
sve_operation_rate	<sve_operation_rate unit="%"> @sve_operation_rate </sve_operation_rate> 実行された浮動小数点演算の総数に対するSVE演算数の比率を示します。単位は%(unit="%")です。 @sve_operation_rate には実行された浮動小数点演算の総数に対するSVE演算数の比率が入ります。
ipc	<ipc unit="IPC"> @ipc </ipc> 1サイクルあたりに実行された命令数を示します。単位はIPC(unit="IPC")です。 @ipc には1サイクルあたりに実行された命令数が入ります。
gips	<gips unit="GIPS"> @gips </gips> 1秒あたりに実行された命令数を示します。単位はGIPS(unit="GIPS")です。 @gips には1秒あたりに実行された命令数が入ります。

2.2.3.2.6 コスト情報 <cost>

スレッド単位の手続コスト分布情報、ループコスト分布情報、行コスト分布情報を出力します。information要素の@item 属性が“cost”の場合に出力します。information要素については“2.2.3.2.3 性能情報 <information>”を参照してください。プログラム単位、プロセス単位の情報は出力しません。

フォーマット

```
<cost>
  <procedures>
    <procedure func="@func" start="@start" end="@end">
      <procedure_base_cost unit="hits">@procedure_base_cost</procedure_base_cost>
      <procedure_barrier_cost unit="hits">@procedure_barrier_cost</procedure_barrier_cost>
      <procedure_mpi_cost unit="hits">@procedure_mpi_cost</procedure_mpi_cost>
    </procedure>
      ......
  </procedures>
  <loops>
    <loop func="@func" start="@start" end="@end" nest="@nest" kind="@kind" exec="@exec">
      <loop_base_cost unit="hits">@loop_base_cost</loop_base_cost>
      <loop_barrier_cost unit="hits">@loop_barrier_cost</loop_barrier_cost>
      <loop_mpi_cost unit="hits">@loop_mpi_cost</loop_mpi_cost>
    </loop>
      ......
  </loops>
  <lines>
    <line func="@func" line_number="@line_number">
```

```
      <line_base_cost unit="hits">@line_base_cost</line_base_cost>
      <line_barrier_cost unit="hits">@line_barrier_cost</line_barrier_cost>
      <line_mpi_cost unit="hits">@line_mpi_cost</line_mpi_cost>
    </line>
      ......
  </lines>
</cost>
```

要素の説明

要素名	説明
cost	<cost> </cost> コスト情報であることを示します。
procedures	<procedures> </procedures> 手続コスト分布情報のまとまりを示します。本要素および子要素は手続コスト情報が存在する場合のみ出力します。
procedure	<procedure func="@func" start="@start" end="@end"> </procedure> 手続コスト分布情報を示します。出力対象の手続が複数存在する場合、複数回出力します。 @func には手続名が入ります。 @start には手続の開始行番号が入ります。本属性は対象手続の開始行番号が取得可能な場合のみ出力します。 @end には手続の終了行番号が入ります。本属性は対象手続の終了行番号が取得可能な場合のみ出力します。 **参照** スレッド並列プログラムの場合、並列化された部分を生成手続の情報として出力します。生成手続名には、手続名の後ろに生成手続の種別に応じた識別子を付加します。詳細については“表2.6 スレッド並列プログラムの生成手続名”を参照してください。
procedure_base_cost	<procedure_base_cost unit="hits"> @procedure_base_cost </procedure_base_cost> 手続のコストを示します。 @procedure_base_cost には手続のコストが入ります。
procedure_barrier_cost	<procedure_barrier_cost unit="hits"> @procedure_barrier_cost </procedure_barrier_cost> 手続のスレッド間同期待ちコストを示します。本要素は計測対象がスレッド並列プログラムの場合のみ出力します。 @procedure_barrier_cost には手続のスレッド間同期待ちコストが入ります。
procedure_mpi_cost	<procedure_mpi_cost unit="hits"> @procedure_mpi_cost </procedure_mpi_cost> 手続のMPIコストを示します。本要素はプロファイル結果の出力時に-Impiオプションが有効な場合のみ出力します。 @procedure_mpi_cost には手続のMPIコストが入ります。
loops	<loops> </loops> ループコスト分布情報のまとまりを示します。本要素および子要素はループコスト情報が存在する場合のみ出力します。
loop	<loop func="@func" start="@start" end="@end" nest="@nest" kind="@kind" exec="@exec"> </loop> ループコスト分布情報を示します。出力対象のループが複数存在する場合、複数回出力します。 @func には手続名が入ります。 @start にはループの開始行番号が入ります。

要素名	説明
	@end にはループの終了行番号が入ります。 @nest にはネストレベルが入ります。 @kind にはループの種別が入ります。以下のいずれかが入ります。 DO、WHILE、UNTIL、ARRAY、FOR、GOTO、OTHER @exec にはループの翻訳種別が入ります。以下のいずれかが入ります。 SERIAL、OpenMP、AUTO
loop_base_cost	<loop_base_cost unit="hits"> @loop_base_cost </loop_base_cost> ループのコストを示します。 @loop_base_cost にはループのコストが入ります。
loop_barrier_cost	<loop_barrier_cost unit="hits"> @loop_barrier_cost </loop_barrier_cost> ループのスレッド間同期待ちコストを示します。本要素は計測対象がスレッド並列プログラムの場合のみ出力します。 @loop_barrier_cost にはループのスレッド間同期待ちコストが入ります。
loop_mpi_cost	<loop_mpi_cost unit="hits"> @loop_mpi_cost </loop_mpi_cost> ループのMPIコストを示します。本要素はプロファイル結果の出力時に-Impiオプションが有効な場合のみ出力します。 @loop_mpi_cost はループのMPIコストが入ります。
lines	<lines> </lines> 行コスト分布情報のまとまりを示します。本要素および子要素は行コスト情報が存在する場合のみ出力します。
line	<line func="@func" line_number="@line_number"> </line> 行コスト分布情報を示します。出力対象の行が複数存在する場合、複数回出力します。 @func には手続名が入ります。 @line_number には行番号が入ります。
line_base_cost	<line_base_cost unit="hits"> @line_base_cost </line_base_cost> 行のコストを示します。 @line_base_cost には行のコストが入ります。
line_barrier_cost	<line_barrier_cost unit="hits"> @line_barrier_cost </line_barrier_cost> 行のスレッド間同期待ちコストを示します。本要素は計測対象がスレッド並列プログラムの場合のみ出力します。 @line_barrier_cost には行のスレッド間同期待ちコストが入ります。
line_mpi_cost	<line_mpi_cost unit="hits"> @line_mpi_cost </line_mpi_cost> 行のMPIコストを示します。本要素はプロファイル結果の出力時に-Impiオプションが有効な場合のみ出力します。 @line_mpi_cost は行のMPIコストが入ります。

2.2.3.2.7 コールグラフ情報 <call>

手続の呼出し経路および手続の呼出し経路ごとのコストを出力します。information要素の@item 属性が“call”の場合に出力します。information要素については“2.2.3.2.3 性能情報 <information> ”を参照してください。この情報は、-Icallオプションが有効な場合に出力します。

フォーマット

```
<call>
  <frames>
    <frame func="@func">
      <procedure_cost unit="hits">@procedure_cost</procedure_cost>
      <frame func="@func">
        <procedure_cost unit="hits">@procedure_cost</procedure_cost>
          ......
      </frame>
    </frame>
  </frames>
</call>
```

要素の説明

要素名	説明
call	<call> </call> コールグラフ情報であることを示します。
frames	<frames> </frames> コールグラフフレーム情報のまとまりを示します。
frame	<frame func="@func" unknown="true"> </frame> コールグラフフレームを示します。複数の手続がネストの関係にある場合、frame要素はネスト出力します。複数の手続が並列の関係にある場合、frame要素は並列出力します。 @func には手続名が入ります。 unknown="true" 属性はテキスト出力におけるネストレベルの表示が<???>の状態の場合のみ出力します。
procedure_cost	<procedure_cost unit="hits"> @procedure_cost </procedure_cost> 手続のコストを示します。 @procedure_cost には手続のコストが入ります。

第3章 詳細プロファイラ

この章では、詳細プロファイラについて説明します。

詳細プロファイラは、アプリケーションの指定した区間の実行性能情報の計測および出力を行います。詳細プロファイラはプロファイルデータを計測するfappコマンドと、計測したデータからプロファイル結果を出力するfapppxコマンドの2つから構成されます。詳細プロファイラが計測および出力する情報は以下のとおりです。

時間統計情報

計測対象区間の呼出し回数、経過時間、ユーザーCPU時間、およびシステムCPU時間の内訳などの情報を出力します。

MPI通信コスト情報

計測対象区間のMPI関数の実行回数、メッセージ長、実行時間および待ち時間の平均値、最大値、および最小値を出力します。

CPU性能解析情報

計測対象区間のプログラム実行時のCPU動作状況を出力します。“第4章 CPU性能解析レポート”で使用します。

注意

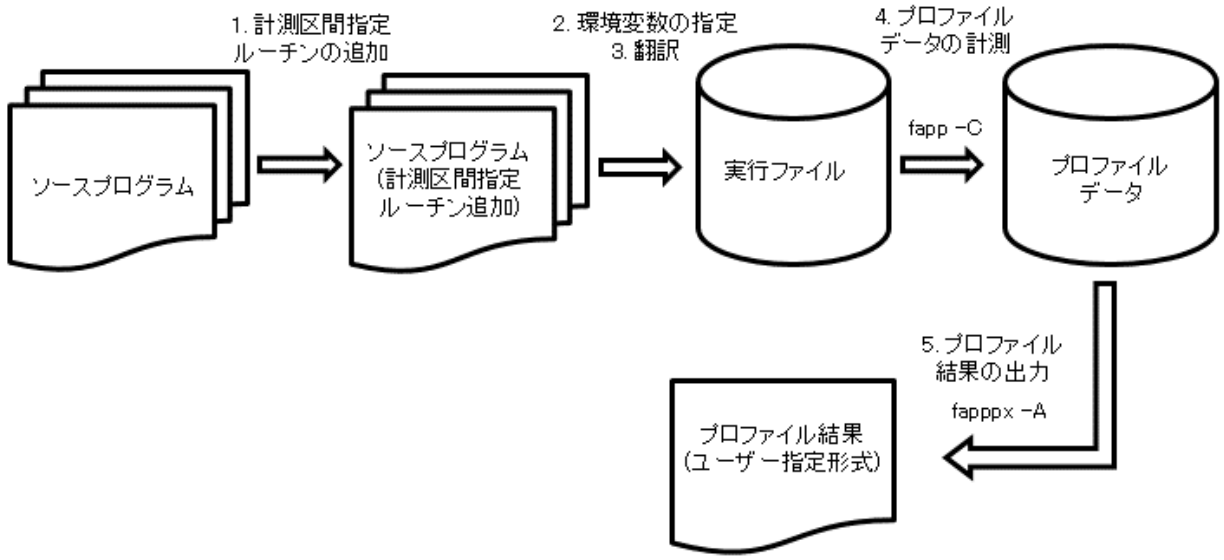
fappコマンドはプロファイルデータを計測するコマンドとプロファイル結果を出力するコマンドの両方で使用可能ですが、本使用手引書ではプロファイルデータを計測するコマンドの意味で使用します。

詳細プロファイラ使用時の注意事項については“5.1 プロファイラ共通の注意事項”および“5.3 詳細プロファイラの注意事項”を参照してください。

3.1 詳細プロファイラの使用手順

詳細プロファイラの使用手順を以下に示します。

図3.1 詳細プロファイラの使用手順



参考

CPU性能解析レポート用のデータを計測・解析する場合にも詳細プロファイラを使用します。CPU性能解析レポートを使用する際の実行例については“4.1 CPU性能解析レポートの使用手順”を参照してください。

各操作の詳細を以下に示します。

3.1.1 計測区間指定ルーチンの追加

プロファイルデータを計測する区間(“計測開始位置”および“計測終了位置”)の指定に必要な計測区間指定ルーチンをソースコードに追加します。詳細プロファイラ向け計測区間指定ルーチンの仕様を以下に示します。

3.1.1.1 fapp_start / fapp_stopサブルーチン (Fortran)

書式

CALL fapp_start(*name*, *number*, *level*)

CALL fapp_stop(*name*, *number*, *level*)

機能説明

詳細プロファイラによるプロファイルデータの計測を開始または終了します。

fapp_start(*name*, *number*, *level*)

詳細プロファイラによるプロファイルデータの計測を開始します。
引数*name*(グループ名)と引数*number*(詳細番号)を組み合わせて計測区間名として扱います。異なる計測区間名は並行して計測できます。引数*level*はfappコマンドの-Lオプションに対して意味を持ちます。“-Lオプションの引数 *level*”>=“引数*level*”の区間のみ計測対象として有効にします。-Lオプションについては“fappコマンドのオプション”を参照してください。

fapp_stop(*name*, *number*, *level*)

詳細プロファイラによるプロファイルデータの計測を終了します。
引数*name*(グループ名)と引数*number*(詳細番号)を組み合わせて計測区間名として扱います。引数*level*はfappコマンドの-Lオプションに対して意味を持ちます。“-Lオプションの引数 *level*”>=“引数*level*”の区間のみ計測対象として有効にします。-Lオプションについては“fappコマンドのオプション”を参照してください。

引数

name

グループ名を示します。引数*number*(詳細番号)と組み合わせて計測区間名として扱います。

基本文字型スカラ。引数*name*には以下の文字を使用できます。

- 英字

  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  a b c d e f g h i j k l m n o p q r s t u v w x y z

- 数字

  0 1 2 3 4 5 6 7 8 9

- 記号

  _ (アンダースコア)

number

詳細番号を示します。引数*name*(グループ名)と組み合わせて計測区間名として扱います。

4Byteの整数型。

level

起動レベルを示します。fappコマンドの-Lオプションで使用します。

4Byteの整数型。ただし、0～2,147,483,647の整数値でなければなりません。誤った値を指定した場合、警告メッセージを出力して本ルーチンによる指定を無視します。

例

計測区間指定ルーチンの使用例

```
program main
...
call fapp_start("foo",1,0)      ! 計測区間名"foo1"の計測を開始する
do i=1,10000
    ...
    call fapp_start("bar",1,0) ! 計測区間名"bar1"の計測を開始する
    do j=1,10000
        ...
    end do
    call fapp_stop("bar",1,0)  ! 計測区間名"bar1"の計測を終了する
end do
call fapp_stop("foo",1,0)       ! 計測区間名"foo1"の計測を終了する
end program main
```

注意

- 詳細プロファイラは計測区間名ごとにプロファイルデータの計測を行います。同じ計測区間名のサブルーチンを複数回呼び出す場合、必ずfapp_start、fapp_stopの順番で呼び出してください。fapp_stopを呼び出す前に再度fapp_startを呼び出した場合、またはfapp_startを呼び出す前にfapp_stopを呼び出した場合、警告メッセージを出力し、その呼び出しを無視します。計測区間名が異なる場合、fapp_startまたはfapp_stopが連続しても問題ありません。また、fapp_stopを呼び出さずにプロセスが終了した場合、その区間のプロファイルデータは計測しません。
- 同一の計測区間名に対する計測を複数回実施した場合、すべての計測結果を合算します。
- 引数*level*の値はfapp_startとfapp_stopで同じ値を指定してください。異なる値を指定した場合、fappコマンドの-Lオプションの指定によっては意図しない結果となる可能性があります。
- スレッド並列プログラムを計測する場合は、計測範囲がスレッド並列区間毎にスレッド並列区間(自動並列化によって生成されたスレッド並列区間も含む)全体を包含するように計測区間指定ルーチンを追加してください。並列化状況については、翻訳情報を確認してください。仕様外の利用の場合は、動作保証しません。
- 以下の計測区間と詳細番号の組み合わせは指定しないでください。この組み合わせはプログラム全体を対象とするよう予約されています。

```
  name : "all"
number : 0
```

  ただし、fappコマンドによる計測時に-Hmethod=fastオプションを指定した場合、この区間の計測は実施しません。-Hmethod=fastオプションについては"3.1.4 プロファイルデータの計測"を参照してください。

- プロセス並列プログラムの場合、計測対象とするすべてのプロセスで同じ計測区間名のサブルーチンを呼び出してください。呼び出しが行われなかったプロセスのプロファイルデータは計測されません。

例

すべてのプロセスを計測対象とする例(mpi_initサブルーチンを呼び出す前に計測を開始する)

```
call fapp_start("foo",1,0)  ! 計測を開始する
call mpi_init(err)
  ...
call mpi_finalize(err)
call fapp_stop("foo",1,0)   ! 計測を終了する
```

すべてのプロセスを計測対象とする例(mpi_initサブルーチンを呼び出した直後に計測を開始する)

```
call mpi_init(err)
call fapp_start("foo",1,0)  ! 計測を開始する
```

```
  ...
call fapp_stop("foo",1,0)    ! 計測を終了する
call mpi_finalize(err)
```

プロセス0のみ計測対象とする例

```
call mpi_init(err)
call mpi_comm_rank(mpi_comm_world,rank,err)
if(rank==0) then
    call fapp_start("foo",1,0)    ! プロセス0のみ、計測を開始する
end if
    ...
if(rank==0) then
    call fapp_stop("foo",1,0)     ! プロセス0のみ、計測を終了する
end if
call mpi_finalize(err)
```

- 翻訳時オプション-mldefault=cdeclを有効にしてFortranプログラムを翻訳する場合、Fortranプログラム内で指定する計測区間指定ルーチンの名前を、以下のように変更してください。

変更前	変更後
fapp_start	fapp_start_
fapp_stop	fapp_stop_

- 翻訳時オプション-AUが有効な場合、計測区間指定ルーチンの名前はすべて小文字で記述してください。

3.1.1.2 fapp_start関数 / fapp_stop関数 (C言語およびC++言語)

書式

```
#include "fj_tool/fapp.h"

void fapp_start( const char *name, int number, int level );
void fapp_stop( const char *name, int number, int level );
```

機能説明

詳細プロファイラによる情報計測を開始または終了します。

fapp_start(const char *name, int number, int level)

詳細プロファイラによるプロファイルデータの計測を開始します。
引数*name*(グループ名)と引数*number*(詳細番号)を組み合わせて計測区間名として扱います。異なる計測区間名は並行して計測できます。引数*level*はfappコマンドの-Lオプションに対して意味を持ちます。“-Lオプションの引数 *level*”>=“引数*level*”の区間のみ計測対象として有効にします。-Lオプションについては“fappコマンドのオプション”を参照してください。

fapp_stop(const char *name, int number, int level)

詳細プロファイラによるプロファイルデータの計測を終了します。
引数*name* (グループ名)と引数*number* (詳細番号)を組み合わせて計測区間名として扱います。引数*level* はfappコマンドの-Lオプションに対して意味を持ちます。“-Lオプションの引数 *level*”>=“引数*level*”の区間のみ計測対象として有効にします。-Lオプションについては“fappコマンドのオプション”を参照してください。

引数

name

グループ名を示します。引数number (詳細番号)と組み合わせて計測区間名として扱います。

引数*name* には以下の文字を使用できます。

- 英字

  A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
  a b c d e f g h i j k l m n o p q r s t u v w x y z

- 数字

  0 1 2 3 4 5 6 7 8 9

- 記号

  _ (アンダースコア)

number

詳細番号を示します。引数*name* (グループ名)と組み合わせて計測区間名として扱います。

int型。

level

起動レベルを示します。fappコマンドの-Lオプションで使用します。

int型。ただし、0～2,147,483,647の整数値でなければなりません。誤った値を指定した場合、警告メッセージを出力して本ルーチンによる指定を無視します。

例

計測区間指定ルーチンの使用例

```
#include "fj_tool/fapp.h" // ヘッダーファイルのインクルード
...

int main(void)
{
    int i,j;
    fapp_start("foo",1,0);      // 計測区間名"foo1"の計測を開始する
    for(i=0;i<10000;i++){
        ...
        fapp_start("bar",1,0); // 計測区間名"bar1"の計測を開始する
        for(j=0;j<10000;j++){
            ...
        }
        fapp_stop("bar",1,0);  // 計測区間名"bar1"の計測を終了する
    }
    fapp_stop("foo",1,0);       // 計測区間名"foo1"の計測を終了する
    return 0;
}
```

注意

- 詳細プロファイラは計測区間名ごとにプロファイルデータの計測を行います。同じ計測区間名の関数を複数回呼び出す場合、必ずfapp_start()、fapp_stop()の順番で呼び出してください。fapp_stop()を呼び出す前に再度fapp_start()を呼び出した場合、またはfapp_start()を呼び出す前にfapp_stop()を呼び出した場合、警告メッセージを出力し、その呼び出しを無視します。計測区間名が異なる場合、fapp_start()またはfapp_stop()が連続しても問題ありません。また、fapp_stop()を呼び出さずにプロセスが終了した場合、その区間のプロファイルデータは計測しません。
- 同一の計測区間名に対する計測を複数回実施した場合、すべての計測結果を合算します。
- 引数*level*の値はfapp_start()とfapp_stop()で同じ値を指定してください。異なる値を指定した場合、fappコマンドの-Lオプションの指定によっては意図しない結果となる可能性があります。
- スレッド並列プログラムを計測する場合は、計測範囲がスレッド並列区間毎にスレッド並列区間(自動並列化によって生成されたスレッド並列区間も含む)全体を包含するように計測区間指定ルーチンを追加してください。並列化状況については、翻訳情報を確認してください。仕様外の利用の場合は、動作保証しません。

- 以下の計測区間と詳細番号の組み合わせは指定しないでください。この組み合わせはプログラム全体を対象とするよう予約されています。

```
  name  : "all"
number  : 0
```

ただし、fappコマンドによる計測時に-Hmethod=fastオプションを指定した場合、この区間の計測は実施しません。-Hmethod=fastオプションについては“3.1.4 プロファイルデータの計測”を参照してください。

- プロセス並列プログラムの場合、計測対象とするすべてのプロセスで同じ計測区間名の関数を呼び出してください。呼び出しが行われなかったプロセスのプロファイルデータは計測されません。

例

すべてのプロセスを計測対象とする例(MPI_Init関数を呼び出す前に計測を開始する)

```
fapp_start("foo",1,0);    // 計測を開始する
MPI_Init(&argc, &argv);
  ...
MPI_Finalize();
fapp_stop("foo",1,0);     // 計測を終了する
```

すべてのプロセスを計測対象とする例(MPI_Init関数を呼び出した直後に計測を開始する)

```
MPI_Init(&argc, &argv);
fapp_start("foo",1,0);    // 計測を開始する
  ...
fapp_stop("foo",1,0);     // 計測を終了する
MPI_Finalize();
```

プロセス0のみ計測対象とする例

```
MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
if(rank==0){
    fapp_start("foo",1,0);  // プロセス0のみ、計測を開始する
}
    ...
if(rank==0){
    fapp_stop("foo",1,0);   // プロセス0のみ、計測を終了する
}
MPI_Finalize();
```

3.1.2 環境変数の指定

プロファイラ使用時に必要な環境変数を指定します。詳細については“2.1.2 環境変数の指定”を参照してください。

3.1.3 翻訳

プログラムを翻訳します。詳細については、“2.1.3 翻訳”を参照してください。

3.1.4 プロファイルデータの計測

fappコマンドを使用して計測します。本操作は計算ノードで実施します。

注意

- fappコマンドで計測したプロファイルデータに対して以下を行った場合、動作保証しません。

  ー プロファイルデータの編集

ー プロファイルデータの追加、削除、リネーム

- プロファイルデータの計測中にプログラムが中断した場合、不完全なプロファイルデータが残ることがあります。
- fappコマンドを使用する場合、環境変数“FLIB_FASTOMP”にTRUEを指定してください。指定しない場合、fappコマンドは正しく動作しません。詳細については“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。
- fappコマンドには-Aオプションも指定可能です。ただし、fappコマンドに-Aオプションを指定した場合、“3.1.5 プロファイル結果の出力”のfapppxコマンドとして扱うため使用方法が異なります。詳細は“3.1.5 プロファイル結果の出力”を参照してください。fappコマンドの-Aオプションと-Cオプションは同時に指定できません。同時に指定した場合、エラーメッセージを出力しプログラムの実行を終了します。

fappコマンドの形式

```
fapp -C -d profile_data  [ -I{{cpupa|nocpupa}|{cputime|nocputime}|{mpi|nompi}} ]
[ -H{event=event|event_raw=event_raw}[,method={fast|normal},mode={all|user}] ]
[ -L level ] [ -W{spawn|nospawn} ] exec-file [exec_option ...]
```

fappコマンドのオプション

ポイント

- オプションの説明文に「○○できません」「○○でなければいけません」という制限事項がある場合、制限事項に違反するとエラーメッセージを出力して実行を終了します。
- 相反するオプションを複数指定した場合、最後に指定したオプションが有効となります。例えば、CPU性能解析情報を計測するかを指定する-I{cpupa|nocpupa}オプションを“-Icpupa -Inocpupa”の順番で指定した場合、-Inocpupaオプションが有効になります。
- CPU性能解析レポート使用時は-Icpupaオプションおよび-Hevent=*event*オプションを使用してください。詳細については“4.1.4 プロファイルデータの計測”を参照してください。

-C

プロファイルデータの計測を指示します。本オプションは必ず指定してください。本オプションを指定しない場合、エラーメッセージを出力し収集コマンドの処理を終了します。

-d *profile_data*

計測したプロファイルデータを格納するディレクトリを指定します。本オプションは必ず指定してください。本オプションを指定しない場合、エラーメッセージを出力し収集コマンドの処理を終了します。*profile_data* は省略できません。*profile_data* にはプロファイルデータを格納するディレクトリ名を相対パスまたは絶対パスで指定します。指定したディレクトリが存在する場合、そのディレクトリは空ディレクトリでなければいけません。指定したディレクトリが存在しない場合、新規にディレクトリを作成します。*profile_data* に“-”から始まるディレクトリ名を指定する場合、絶対パスで指定、またはカレントディレクトリ(“./”)を含む相対パスで指定してください。実行中にカレントディレクトリを移動するアプリケーションを解析する場合、*profile_data* は絶対パスで指定してください。

-I{{cpupa|nocpupa}|{cputime|nocputime}|{mpi|nompi}}

詳細プロファイラで計測する項目を指定します。-Iオプションにはコンマ(,)を区切りとしてサブオプションを複数指定することができます。例えば、-Inocpupa,mpiのような指定ができます。-I{cpupa|nocpupa}オプション省略時の指定は-Hオプションによって異なります。-Hオプションが指定されている場合、-Icpupaオプションが有効になります。-Hオプションが指定されていない場合、-Inocpupaオプションが有効になります。-I{cputime|nocputime}オプション省略時の指定は-I{cpupa|nocpupa}オプションによって異なります。-Icpupaオプションが有効な場合、-Inocputimeオプションが有効になります。-Inocpupaオプションが有効な場合、-Icputimeオプションが有効になります。-Icpupaオプションが有効な場合のみ、-Inocputimeオプションが指定できます。-Inocpupaオプションが有効な場合に-Inocputimeオプションを指定した場合、エラーメッセージを出力し収集コマンドの処理を終了します。-I{mpi|nompi}オプション省略時の指定はプログラムの種別によって異なります。MPIプログラムの場合、-Impiオプションが有効になります。非MPIプログラムの場合、-Inompiオプションが有効になります。-Icpupaオプションが有効で-Hオプションを指定していない場合、-Hevent=statistics,method=normal,mode=allオプションが有効になります。

cpupa

CPU性能解析情報を計測します。

nocpupa

CPU性能解析情報を計測しません。

cputime

ユーザーCPU時間およびシステムCPU時間を計測します。

nocputime

ユーザーCPU時間およびシステムCPU時間を計測しないかわりに、CPU性能解析情報の計測にかかる時間を短縮します。

mpi

MPI通信コスト情報を計測します。本引数を非MPIプログラムに指定した場合、エラーメッセージを出力して収集コマンドの処理を終了します。

nompi

MPI通信コスト情報を計測しません。

-H{event=*event* |event_raw=*event_raw* }[,method={fast|normal},mode={all|user}]

CPU性能解析情報の計測について指定します。-Inocpupaオプションを指定した場合、警告メッセージを出力して本オプションを無効にします。サブオプションevent=*event*またはevent_raw=*event_raw*はどちらか1つを必ず指定してください。サブオプションevent=*event*およびevent_raw=*event_raw*を同時に指定した場合、最後に指定したサブオプションが有効になります。サブオプションmode=の省略時はmode=allが有効になります。サブオプションmethod=の省略時はmethod=normalが有効になります。

event=*event*

CPU性能解析レポートに使用する情報を計測します。*event*は省略できません。*event*には以下のいずれか1つを指定します。pa1とstatisticsは同値です。

{ pa1 | pa2 | pa3 | pa4 | pa5 | pa6 | pa7 | pa8 | pa9 | pa10 | pa11 | pa12 | pa13 | pa14 | pa15 | pa16 | pa17 | statistics}

event_raw=*event_raw*

PMUイベント情報のイベント番号を指定して、CPU性能解析情報を計測します。*event_raw*は省略できません。*event_raw*には、CPUに対応したイベント番号を10進数または16進数(0xまたは0X)表記で指定してください。*event_raw*は、コンマ(,)で区切ることで、最大8個まで指定できます。

method=fast

CPU性能解析情報の計測方式を指定します。本サブオプションを指定した場合、ハードウェアの情報を直接計測する方式により、高精度のCPU性能解析情報の計測を行います。

method=normal

CPU性能解析情報の計測方式を指定します。本サブオプションを指定した場合、OSを経由して計測する方法によりCPU性能解析情報の計測を行います。本サブオプションを指定した場合、-Hevent_raw=*event_raw*に同じイベント番号を複数回指定することは出来ません。

mode=all

CPU性能解析情報の計測モードを指定します。本サブオプションを指定した場合、カーネルモードおよびユーザーモードにおける性能計測を行います。

mode=user

CPU性能解析情報の計測モードを指定します。本サブオプションを指定した場合、ユーザーモードにおける性能計測を行います。

-L *level*

計測対象の起動レベルを指定します。*level*には0～2,147,483,647の範囲の整数値を指定します。*level*に範囲外の値を指定した場合、警告メッセージを出力して-L 0オプションが有効になります。本オプションは計測区間指定ルーチンの第3引数*level*に対して意味を持ちます。“*level*”>=“計測区間指定ルーチンの第3引数*level*”の区間のみ計測対象として有効にします。本オプション省略時は、-L 0オプションが有効になります。

-W{spawn|nospawn}

動的に生成されたプロセスに対する計測方法を指定します。-W{spawn|nospawn}オプション省略時の指定はプログラムの種別によって異なります。MPIプログラムの場合、-Wspawnオプションが有効になります。非MPIプログラムの場合、-Wnospawnオプションが有効になります。

spawn

動的に生成されたプロセスの統計情報を計測します。特定の動的に生成されたプロセスに対してinfoキー“fjprof_spawn_dir_name”を指定している場合、infoキーを指定した動的に生成されたプロセスのプロファイルデータはinfoキー“fjprof_spawn_dir_name”に指定したディレクトリ内に格納されます。infoキー“fjprof_spawn_dir_name”を指定していない場合、動的に生成されたプロセスのプロファイルデータは-dオプションで指定したディレクトリ内に格納されます。本引数を非MPIプログラムに指定した場合、エラーメッセージを出力して収集コマンドの処理を終了します。

nospawn

動的に生成されたプロセスの統計情報を計測しません。ただし、特定の動的に生成されたプロセスに対してinfoキー“fjprof_spawn_dir_name”を指定している場合、infoキーを指定した動的に生成されたプロセスについては計測が行われ、計測したプロファイルデータはinfoキー“fjprof_spawn_dir_name”に指定したディレクトリ内に格納されます。

exec-file [*exec_option* ...]

プロファイルデータ計測対象の実行ファイルとオプションを指定します。MPIプログラムの場合、mpiexecを指定します。*exec-file*に“-”で始まる実行ファイルを指定する場合、カレントディレクトリ(“./”)を含んだ相対パス、または絶対パスで指定してください。*exec-file*にシェルスクリプトを指定することはできません。実行ファイル名に続く文字列(*exec_option*...)は、実行ファイルへのオプションと見なします。

例

fappコマンドでCPU性能解析情報(statistics)を計測する例

```
fapp -C -d ./tmp -Icpupa -Hevent=statistics ./a.out
```

3.1.5 プロファイル結果の出力

fappコマンドで計測したプロファイルデータの結果を出力します。また、CPU性能解析レポートに使用する入力ファイルを出力することもできます。本操作は使用するノードによって異なるコマンドを使用します。

ログインノードの場合

fapppxコマンドを使用します。

計算ノードの場合

fappコマンドを使用します。

fapppxコマンドまたはfappコマンドの形式

```
{fapppx|fapp} -A [ -I{{cpupa|nocpupa}|{mpi|nompi}} ] [ -o outfile ] [ -pp_no ] [ -t{csv|text|xml} ]
[ -d ] profile_data
```

fapppxコマンドまたはfappコマンドのオプション

ポイント

- オプションの説明文に「○○できません」「○○でなければいけません」という制限事項がある場合、制限事項に違反するとエラーメッセージを出力して実行を終了します。
- 相反するオプションを複数指定した場合、最後に指定したオプションが有効となります。例えば、プロファイル結果の出力対象とする項目を指定する-I{cpupa|nocpupa}オプションを“-Icpupa -Inocpupa”の順番で指定した場合、-Inocpupaオプションが有効になります。
- CPU性能解析レポートを使用する場合、-tcsv オプションを指定してください。

-A

プロファイル結果の出力処理を指示します。本オプションは必ず指定してください。本オプションを指定しない場合、エラーメッセージを出力し解析コマンドの処理を終了します。

-I{{cpupa|nocpupa}|{mpi|nompi}}

プロファイル結果の出力対象とする項目を指定します。-Iオプションにはコンマ(,)を区切りとしてサブオプションを複数指定することができます。例えば、-Inocpupa,mpiのような指定ができます。-I{cpupa|nocpupa}オプション省略時は-Inocpupaオプションが有効になります。-I{mpi|nompi}オプション省略時の指定はプロファイルデータ計測時の対象プログラムの種別によって異なります。プロファイルデータの計測対象がMPIプログラムの場合、-Impiオプションが有効になります。プロファイルデータの計測対象が非MPIプログラムの場合、-Inompiオプションが有効になります。-Icpupa,mpiの各オプションを指定する場合、fappコマンドで該当する項目を計測している必要があります。計測していない情報を出力対象に指定した場合、その指定を無視します。

cpupa

CPU性能解析情報を出力します。本オプションと-ttextオプションを同時に指定した場合の動作は、計測したプロファイルデータによって異なります。プロファイルデータの計測時に、-Hevent=pa1オプションまたは-Hevent=statisticsオプションを指定した場合、CPU性能解析情報をテキスト形式で出力します。それ以外のオプションを指定した場合、CPU性能解析情報をテキスト形式で出力しません。本オプションと-tcsvオプションまたは-txmlオプションを同時に指定した場合、-Hオプションの指定に関わらずCPU性能解析情報を出力します。

nocpupa

CPU性能解析情報を出力しません。

mpi

MPI通信コスト情報を出力します。

nompi

MPI通信コスト情報を出力しません。

-o *outfile*

プロファイル結果の出力先を指定します。*outfile*には、出力先ファイル名を相対パスまたは絶対パスで指定するか、またはstdoutを指定します。本オプション省略時は-ostdout オプションが有効になります。*outfile*にstdoutを指定した場合、プロファイル結果を標準出力に出力します。*outfile* に“-”で始まるファイル名を指定する場合、絶対パスまたはカレントディレクトリ(“./”) を含む相対パスで指定してください。

-p*p_no*

プロファイル結果に出力するプロセスを指定します。*p_no*には*N*@*M*, input=*n*, limit=*m*, allの中からいずれか1つ以上を指定します。本オプション省略時の値は-t{csv|text|xml}オプションの指定によって異なります。-ttextオプションが有効な場合、-pinput=0,limit=16オプションが有効になります。-tcsvオプションまたは-txmlオプションが有効な場合、-pallオプションが有効になります。*p_no*は省略することができません。-pオプションにはコンマ(,)を区切りとして*p_no*を複数指定することができます。例えば、-p3,5,limit=10のような指定ができます。

N[@*M*]...

*N*に指定したプロセス番号の情報を先頭に出力します。spawn番号*M*に所属するプロセス番号*N*を指定する場合、*N*@*M*のように指定します。*N*[@*M*] に指定したプロセス番号の情報が存在しない場合、指定を無視します。*N*[@*M*] は複数指定できます。*N*[@*M*] を複数指定した場合、指定した順番に出力します。[@*M*]は-ttextオプションが有効な場合のみ有効になります。

input=*n*

コストが上位*n*件のプロセスの情報を読み込みます。読み込むファイル数が減るため処理が高速になりますが、読み込まなかったプロセスの情報は比率計算をする際の分母に含まれません。*n*に0またはプロセス数を超えた値を指定した場合、全プロセスの情報を読み込みます。本サブオプション省略時はinput=0が有効になります。サブオプションinput=*n*とallを同時に指定した場合、オプションの指定順にかかわらず、サブオプションinput=*n*が有効になります。サブオプションinput=*n*とサブオプションlimit=*m*は同時に指定できます。

limit=*m*

コストが上位*m*件のプロセスの情報を出力します。-ttextオプションが有効な場合、コストが高い順に出力します。出力しなかったプロセスの情報は比率計算をする際の分母に含まれます。*m*に0、または入力となるプロセス数を超える値を指定した場合、全プロセスの情報を出力します。本サブオプション省略時の値は-t{csv|text|xml}オプションの指定によって異なります。-ttextオプションが有効な場合、limit=16が有効になります。-tcsvオプションまたは-txmlオプションが有効な場合、limit=0が有効になります。

all

全プロセスの情報を読み込み、出力します。-ttextオプションが有効な場合、コストが高い順に出力します。-pinput=0,limit=0オプションを指定した場合と同じです。サブオプションinput=*n* および limit=*m* のいずれも指定しない場合、こちらが有効になります。

-t{csv|text|xml}

プロファイル結果の出力形式を指定します。本オプション省略時は-ttextが有効になります。

csv

プロファイル結果をCSV形式で出力します。

text

プロファイル結果をTEXT形式で出力します。

xml

プロファイル結果をXML形式で出力します。

-d *profile_data*

プロファイルデータを格納したディレクトリ名を、*profile_data*に相対パスまたは絶対パスで指定します。本オプションは省略できません。ただし、*profile_data*の指定をオプション並びの最後にする場合に限り、“-d”は省略できます。*profile_data*に“-”で始まるディレクトリ名を指定する場合、絶対パス、またはカレントディレクトリ(“./”)を含む相対パスで指定してください。

例

fappコマンドの計測結果 ./tmp からプロファイル結果をTEXT形式で出力する例

```
fapppx -A -ttext -o tmp.txt -d ./tmp
```

3.2 プロファイル結果

fapppxコマンドまたはfappコマンドが出力する詳細プロファイラ情報の内容について説明します。

3.2.1 プロファイル結果の概要

詳細プロファイラ情報は以下の構成となっています。各情報は、fapppxコマンドまたはfappコマンドの-Iオプションによって出力を制御できます。-Iオプションの詳細については“3.1.5 プロファイル結果の出力”を参照してください。TEXT形式の場合、以下の順番で情報を出力します。

- プロファイルデータ計測環境情報
- 時間統計情報
- MPI通信コスト情報
- CPU性能解析情報

詳細については、“3.2.2 プロファイル結果の詳細(TEXT形式)”または“3.2.3 プロファイル結果の詳細(XML形式)”配下を参照してください。

3.2.2 プロファイル結果の詳細(TEXT形式)

“3.1.5 プロファイル結果の出力”で-ttextオプションを指定した場合、TEXT形式による出力を行います。TEXT形式のフォーマットについて説明します。

3.2.2.1 プロファイルデータ計測環境情報

プロファイルデータ計測環境情報は、プロファイルデータを計測した時の環境情報を出力します。

プロファイルデータ計測環境情報の出力形式

```
--------------------------------------------------------------------------------
Fujitsu Advanced Performance Profiler Version @vl
   Measured time                : @date
   CPU frequency                : Process      @pno  @frequency (MHz)  @sno
   Type of program              : @type
```

```
  Virtual coordinate            : (@x, @y, @z)
--------------------------------------------------------------------------------
```

表3.1 詳細プロファイリングデータ計測環境情報の出力項目

出力項目	出力項目の意味
@vl	プロファイラの版数
@date	プロファイルデータの計測日時
@pno	プロセス番号
@frequency	CPU周波数 プロセス毎に、以下の“システムCPU周波数ファイル”からプロファイラ終了時に採取した値です。 /sys/devices/system/cpu/cpu*N*/cpufreq/scaling_cur_freq　（cpu*N* の *N* はコア番号） “システムCPU周波数ファイル”が存在しない場合は、プロファイラ内部で計算した値になります。 **注意** 以下のいずれかの条件に該当する場合、CPU周波数は“--”を出力します。 ・“3.1.4 プロファイルデータの計測”または“3.1.5 プロファイル結果の出力”で-Inocpupaオプションが有効な場合 ・“3.1.4 プロファイルデータの計測”で-Hmethod=fastオプションを指定し、かつ“3.1.1 計測区間指定ルーチンの追加”で指定した計測区間の計測が1度も実施されなかった場合 ・“3.1.4 プロファイルデータの計測”で-Hmode=userオプションを指定した場合
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・計測対象がMPIプログラムである ・fappコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・動的に生成されたプロセスである
@type	プログラム実行形態 (@thnoはスレッド数を示します) “SERIAL” : 逐次 “Thread (AUTO) @thno” : 自動並列のみ “Thread (OpenMP) @thno” : OpenMPのみ “Thread (OpenMP & AUTO) @thno” : OpenMP,自動並列併用 “MPI” : MPIのみ “MPI & Thread (AUTO) @thno” : MPI,自動並列 “MPI & Thread (OpenMP) @thno” : MPI,OpenMP “MPI & Thread (OpenMP & AUTO) @thno” : MPI,OpenMP,自動並列併用
(@x, @y, @z)	MPIプログラム実行時の論理形状 **ポイント** データ計測対象がMPIプログラムの場合のみ出力します。 また、動的に生成したプロセスの論理形状は出力しません。

ポイント

プロセス(動的に生成されたプロセスを含みます)ごとにCPU周波数が異なる場合、CPU frequency を複数行出力します。ただし、同じCPU周波数のプロセスが連続する場合、または-Inocpupaオプションが指定されている場合、CPU frequency は以下のとおり1行にまとめて出力します。

```
CPU frequency                : Process      @spno - @lpno @frequency (MHz)   (Spawn @ssno - @lsno )
```

連続するプロセスのうち*@spno*には最小のプロセス番号、*@lpno*には最大のプロセス番号が入ります。*@ssno*には最小のspawn番号、*@lsno*には最大のspawn番号が入ります。fapppxコマンドまたはfappコマンドの-pinput=*n* オプションで読み込むプロセス情報を限定している場合、読み込んだプロセスのうち同じCPU周波数の範囲に含まれる最小、および最大のプロセス番号またはspawn番号が入ります。-pinput=*n* オプションについては“3.1.5 プロファイル結果の出力”を参照してください。

3.2.2.2 時間統計情報

時間統計情報は、計測対象区間ごとの、呼出し回数、経過時間、ユーザーCPU時間、およびシステムCPU時間の平均値、最大値、および最小値を出力します。

時間統計情報の出力形式

```
Basic profile

 ********************************************************************************
       Application
 ********************************************************************************

     Kind   Elapsed(s)      User(s)    System(s)         Call
   ------------------------------------------------------------
      AVG      @elapse        @user         @sys        @call    @name @number @sno
      MAX      @elapse        @user         @sys        @call
      MIN      @elapse        @user         @sys        @call

 ********************************************************************************
       Process @pno @sno
 ********************************************************************************

            Elapsed(s)      User(s)    System(s)         Call
   ------------------------------------------------------------
               @elapse        @user         @sys        @call    @name @number
```

表3.2 時間統計情報の出力項目

出力項目	出力項目の意味
@elapse	経過時間(s)
@user	ユーザーCPU時間(s) “3.2.2.1 プロファイルデータ計測環境情報”の@type が“SERIAL”または“MPI”以外の場合、またはプロファイルデータの計測時に-Inocputimeオプションを指定した場合、この出力項目は“--”固定です。
@sys	システムCPU時間(s) “3.2.2.1 プロファイルデータ計測環境情報”の@type が“SERIAL”または“MPI”以外の場合、またはプロファイルデータの計測時に-Inocputimeオプションを指定した場合、この出力項目は“--”固定です。
@call	呼出し回数
@name	グループ名
@number	詳細番号
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります)

出力項目	出力項目の意味
	本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fappコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ 動的に生成されたプロセスである
@pno	プロセス番号

3.2.2.3 MPI通信コスト情報

MPI通信コスト情報は、MPIルーチンの実行回数、メッセージ長、実行時間および待ち時間の平均値、最大値、および最小値を出力します。

MPI通信コスト情報の出力形式

```
MPI profile

 ****************************************************************************************
       Application
 ****************************************************************************************

     Kind   Elapsed(s)   Wait(s)   Byte      Call (   0-4K     4K-64K    64K-1024K   1024KB-)
   ------------------------------------------------------------------------------------------------------------------------------
              @elapse     @wait  ----      @call      @ma        @mb          @mc          @md   @name @number @sno
   ------------------------------------------------------------------------------------------------------------------------------
       AVG    @elapse     @wait  @byte     @call      @ma        @mb          @mc          @md   @mfunc
       MAX    @elapse     @wait  @byte     @call      @ma        @mb          @mc          @md
       MIN    @elapse     @wait  @byte     @call      @ma        @mb          @mc          @md


 ****************************************************************************************
       Process @pno @sno
 ****************************************************************************************

            Elapsed(s)   Wait(s)   Byte      Call (   0-4K     4K-64K    64K-1024K   1024KB-)
   ------------------------------------------------------------------------------------------------------------------------------
              @elapse   @wait    ----      @call      @ma        @mb          @mc          @md   @name @number
   ------------------------------------------------------------------------------------------------------------------------------
              @elapse   @wait    @byte     @call      @ma        @mb          @mc          @md   @mfunc
```

表3.3 MPI通信コスト情報の出力項目

出力項目	出力項目の意味
@elapse	経過時間(s)
@wait	待ち時間(s)
@byte	平均メッセージ長(Byte)
@call	MPIライブラリの呼出し回数
@ma	メッセージ長が0Byte以上～4KB未満の場合の、MPIライブラリの呼出し回数
@mb	メッセージ長が4KB以上～64KB未満の場合の、MPIライブラリの呼出し回数
@mc	メッセージ長が64KB以上～1024KB未満の場合の、MPIライブラリの呼出し回数
@md	メッセージ長が1024KB以上の、MPIライブラリの呼出し回数
@name	グループ名
@number	詳細番号
@sno	spawn番号

出力項目	出力項目の意味
	(Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fappコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ 動的に生成されたプロセスである
@mfunc	MPIライブラリ名
@pno	プロセス番号

注意

MPIルーチンが持続的通信要求に属する場合、MPI通信コスト情報の出力方法が変化します。“持続的通信要求”の詳細および持続的通信要求のルーチンとして扱われるルーチンの一覧については“MPI使用手引書”を参照してください。持続的通信要求の通信をMPI_STARTルーチンを使用して開始する場合と、MPI_STARTALLルーチンを使用する場合で出力方法が異なります。

1. **持続的通信要求の通信を、MPI_STARTルーチンを使用して開始する場合**

   MPI通信コスト情報には持続的通信要求ルーチンとMPI_STARTルーチンの2つを出力します。それぞれ、以下の特徴があります。

   持続的通信要求ルーチン

   - @mfuncには持続的通信要求ルーチン名が出力されます。
   - @byteは0固定です。
   - それ以外は通常の出力と同じです。

   MPI_STARTルーチン

   - @mfuncにはMPI_STARTルーチン名とMPI_STARTが開始した持続的通信要求ルーチン名を出力します。持続的通信要求ルーチン名はMPI_STARTルーチンの後ろに()付きで出力します。
   - @elapsed、@wait、および@callにはMPI_STARTルーチンの情報を出力します。
   - @byte、@ma、@mb、@mc、および@mdには開始した持続的通信要求ルーチンの情報を出力します。
   - MPI_STARTルーチンが複数存在する場合、開始した持続的通信要求ルーチンごとに集計します。
   - それ以外は通常の出力と同じです。

2. **持続的通信要求の通信を、MPI_STARTALLルーチンを使用して開始する場合**

   MPI通信コスト情報には持続的通信要求ルーチンとMPI_STARTALLルーチンの2つを出力します。それぞれ、以下の特徴があります。

   持続的通信要求ルーチン

   - @mfuncには持続的通信要求ルーチン名が出力されます。
   - @byteは0固定です。
   - それ以外は通常の出力と同じです。

   MPI_STARTALLルーチン

   MPI_STARTALLルーチンの情報は以下の2種類の行から構成されます。

   - @mfuncがMPI_STARTALLルーチンの行(以降、代表行と呼びます)
   - @mfuncが"MPI_STARTALLルーチンで開始した持続的通信要求ルーチン名"の行(以降、内訳行と呼びます)

   それぞれ、以下の特徴があります。

代表行

- @mfuncには、MPI_STARTALLルーチンを出力します。
- @elapsed、@wait、および@callにはMPI_STARTALLルーチンの情報を出力します。
- @byte、@ma、@mb、@mc、および@mdには、開始した全持続的通信要求ルーチンの集計結果を出力します。

内訳行

- 内訳行の情報は()付きで出力します。また、@mfuncはインデントを下げて出力します。
- @mfuncには、MPI_STARTALLで開始した持続的通信要求ルーチン名を出力します。複数の持続的通信要求ルーチンを開始した場合、1ルーチンに対して1行を出力します。
- @elapsedおよび@waitは"--"固定です。
- @byte、@call、@ma、@mb、@mc、および@mdには、@mfuncに該当するルーチンの集計結果を()付きで出力します。

MPI_STARTALLルーチンが複数存在した場合、開始した持続的通信要求ルーチンの種別および呼び出し順が一致した場合のみ、同一ルーチンとして集計します。ただし、同じ持続的通信要求ルーチンを複数回呼び出した場合、最初の1回のみカウントします。

例

以下に持続的通信要求ルーチンの出力例(抜粋)を示します。

MPI_START

```
0.0001    0.0000       0.0000    2  2  0  0  0  MPI_Send_init
0.0000    0.0000    3999.5000    2  1  1  0  0  MPI_Start(MPI_Send_init)
```

MPI_STARTALL

```
   0.0000    0.0000       0.0000    2  2  0  0  0  MPI_Send_init
   0.0000    0.0000       0.0000    2  2  0  0  0  MPI_Bsend_init
   0.0000    0.0000       0.0000    2  2  0  0  0  MPI_Rsend_init
   0.0000    0.0000 1091998.5000    2  0  0  0  2  MPI_Startall
       --        --    (3999.5000) (2)(1)(1)(0)(0)    (MPI_Send_init)
       --        --   (63999.5000) (2)(0)(1)(1)(0)    (MPI_Bsend_init)
       --        --  (1023999.5000) (2)(0)(0)(1)(1)    (MPI_Rsend_init)
   0.0000    0.0000       0.0000    1  1  0  0  0  MPI_Comm_rank
```

メッセージ長の計算式

持続的通信要求以外のMPIルーチンの計算式を下表に示します。持続的通信要求MPIルーチンの計算式については“メッセージ長の計算式(持続的通信要求ルーチン)”を参照してください。

表3.4 メッセージ長の計算式

MPIサブルーチン/関数名	計算式
MPI_SEND MPI_BSEND MPI_SSEND MPI_RSEND MPI_ISEND MPI_IBSEND MPI_ISSEND MPI_IRSEND	送信バッファ内の要素数*送信バッファの各要素のデータ型のサイズ
MPI_RECV	受信した要素数*受信バッファの各要素のデータ型のサイズ

MPIサブルーチン/関数名	計算式
	注意 MPI_RECVルーチンのstatus引数に“MPI_STATUS_IGNORE”を指定した場合、“受信した要素数”が取得できず0となります。status引数に“MPI_STATUS_IGNORE”を指定しないでください。
MPI_IRECV	受信バッファ内の要素数*受信バッファの各要素のデータ型のサイズ
MPI_SENDRECV	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(受信した要素数*受信バッファの要素のデータ型のサイズ) **注意** MPI_SENDRECVルーチンのstatus引数に“MPI_STATUS_IGNORE”を指定した場合、“受信した要素数”が取得できず0となります。status引数に“MPI_STATUS_IGNORE”を指定しないでください。
MPI_SENDRECV_REPLACE	(送信兼受信バッファ内の要素数*送信兼受信バッファの要素のデータ型のサイズ)*2
MPI_BCAST MPI_IBCAST	・ ルートプロセスの場合 バッファ内の要素数*バッファの要素のデータ型のサイズ*2 ・ ルートプロセス以外の場合 バッファ内の要素数*バッファの要素のデータ型のサイズ
MPI_GATHER MPI_IGATHER	・ ルートプロセスの場合 (送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(総プロセス数*各プロセスから受信するデータの要素数*受信バッファ要素のデータ型のサイズ) ・ ルートプロセス以外の場合 送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ
MPI_GATHERV MPI_IGATHERV	・ ルートプロセスの場合 (送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(各プロセスから受信するデータの要素数の合計*受信バッファ要素のデータ型のサイズ) ・ ルートプロセス以外の場合 送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ
MPI_SCATTER MPI_ISCATTER	・ ルートプロセスの場合 (総プロセス数*各プロセスに送られる要素数*送信バッファの要素のデータ型のサイズ)+(受信バッファ内の要素数*受信バッファの要素のデータ型のサイズ) ・ ルートプロセス以外の場合 受信バッファ内の要素数*受信バッファの要素のデータ型のサイズ
MPI_SCATTERV MPI_ISCATTERV	・ ルートプロセスの場合 (各プロセスへ送信するデータの要素数の合計*送信バッファの要素のデータ型のサイズ)+(受信バッファ内の要素数*受信バッファの要素のデータ型のサイズ) ・ ルートプロセス以外の場合 受信バッファ内の要素数*受信バッファの要素のデータ型のサイズ
MPI_ALLGATHER MPI_IALLGATHER	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(総プロセス数*受信バッファの要素数*受信バッファの要素のデータ型のサイズ)
MPI_ALLGATHERV MPI_IALLGATHERV	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(各プロセスから受信するデータの要素数の合計*受信バッファの要素のデータ型のサイズ)
MPI_ALLTOALL MPI_IALLTOALL	(総プロセス数*各プロセスへ送信するデータの要素数*送信バッファの要素のデータ型のサイズ)+(総プロセス数*1つのプロセスから受信するデータの要素数*受信バッファの要素のデータ型のサイズ)

MPIサブルーチン/関数名	計算式
MPI_ALLTOALLV MPI_IALLTOALLV	(各プロセスへ送信するデータの要素数の合計*送信バッファの要素のデータ型のサイズ) +(各プロセスから受信するデータの要素の合計*受信バッファの要素のデータ型のサイズ)
MPI_REDUCE MPI_IREDUCE	・ ルートプロセスの場合 送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ*2 ・ ルートプロセス以外の場合 送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ
MPI_ALLREDUCE MPI_IALLREDUCE	送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ*2
MPI_REDUCE_SCATTER MPI_IREDUCE_SCATTER	(各プロセスへ送信するデータの要素数の合計*バッファの要素のデータ型のサイズ)+(受信要素数*バッファの要素のデータ型のサイズ)
MPI_SCAN MPI_ISCAN	入力バッファの要素数*入力バッファの要素のデータ型のサイズ*2
MPI_PUT MPI_RPUT	起点バッファ内の要素数*起点バッファの要素のデータ型のサイズ
MPI_GET MPI_RGET	ターゲットバッファ内の要素数*ターゲットバッファの要素のデータ型のサイズ
MPI_ACCUMULATE MPI_RACCUMULATE	起点バッファ内の要素数*起点バッファの要素のデータ型のサイズ
MPI_ALLTOALLW MPI_IALLTOALLW	(各プロセスの送信バッファの要素数*各プロセスの送信バッファのデータ型のサイズ)+(各プロセスの受信バッファの要素数*各プロセスの受信バッファのデータ型のサイズ)
MPI_EXSCAN MPI_IEXSCAN	入力バッファの要素数*入力バッファの要素のデータ型のサイズ*2
MPI_REDUCE_SCATTER_BLOCK MPI_IREDUCE_SCATTER_BLOCK	(バッファの要素のデータ型のサイズ*総プロセス数*ブロックごとの要素数)+(バッファの要素のデータ型のサイズ*ブロックごとの要素数)
MPI_MRECV MPI_IMRECV	各受信バッファ要素のデータ型のサイズ*受信バッファ内の要素数
MPI_COMPARE_AND_SWAP	すべてのバッファ要素のデータ型のサイズ*2
MPI_GET_ACCUMULATE MPI_RGET_ACCUMULATE	(起点バッファ内のデータ型サイズ*起点バッファのエントリ数)+(ターゲットバッファ内のデータ型のサイズ*ターゲットバッファのエントリ数)
MPI_NEIGHBOR_ALLGATHER MPI_INEIGHBOR_ALLGATHER	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(入力となる辺の数*受信バッファの要素数*受信バッファの要素のデータ型のサイズ)
MPI_NEIGHBOR_ALLGATHERV MPI_INEIGHBOR_ALLGATHERV	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(各入力となる辺の受信バッファの要素数の合計*受信バッファの要素のデータ型のサイズ)
MPI_NEIGHBOR_ALLTOALL MPI_INEIGHBOR_ALLTOALL	(出力となる辺の数*送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(入力となる辺の数*受信バッファの要素数*受信バッファの要素のデータ型のサイズ)
MPI_NEIGHBOR_ALLTOALLV MPI_INEIGHBOR_ALLTOALLV	(各出力となる辺の送信バッファ内の要素数の合計*送信バッファの要素のデータ型のサイズ)+(各入力となる辺の受信バッファの要素数の合計*受信バッファの要素のデータ型のサイズ)
MPI_NEIGHBOR_ALLTOALLW MPI_INEIGHBOR_ALLTOALLW	(各出力となる辺の送信バッファ内の要素数の合計*各出力となる辺の送信バッファの要素のデータ型のサイズ)+(各入力となる辺の受信バッファの要素数の合計*各入力となる辺の受信バッファの要素のデータ型のサイズ)

メッセージ長の計算式(持続的通信要求ルーチン)

持続的通信要求MPIルーチンの計算式を下表に示します。持続的通信要求以外のMPIルーチンの計算式については“メッセージ長の計算式”を参照してください。

表3.5 メッセージ長の計算式(持続的通信要求ルーチン)

MPIサブルーチン/関数名	計算式
MPI_SEND_INIT MPI_BSEND_INIT MPI_SSEND_INIT MPI_RSEND_INIT	送信バッファ内の要素数*送信バッファの各要素のデータ型のサイズ
MPI_RECV_INIT	受信バッファ内の要素数*受信バッファの各要素のデータ型のサイズ
MPIX_BCAST_INIT	・ ルートプロセスの場合 バッファ内の要素数*バッファの要素のデータ型のサイズ*2 ・ ルートプロセス以外の場合 バッファ内の要素数*バッファの要素のデータ型のサイズ
MPIX_GATHER_INIT	・ ルートプロセスの場合 (送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(総プロセス数*各プロセスから受信するデータの要素数*受信バッファ要素のデータ型のサイズ) ・ ルートプロセス以外の場合 送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ
MPIX_GATHERV_INIT	・ ルートプロセスの場合 (送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(各プロセスから受信するデータの要素数の合計*受信バッファ要素のデータ型のサイズ) ・ ルートプロセス以外の場合 送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ
MPIX_SCATTER_INIT	・ ルートプロセスの場合 (総プロセス数*各プロセスに送られる要素数*送信バッファの要素のデータ型のサイズ)+(受信バッファ内の要素数*受信バッファの要素のデータ型のサイズ) ・ ルートプロセス以外の場合 受信バッファの要素数*受信バッファの要素のデータ型のサイズ
MPIX_SCATTERV_INIT	・ ルートプロセスの場合 (各プロセスへ送信するデータの要素数の合計*送信バッファの要素のデータ型のサイズ)+(受信バッファ内の要素数*受信バッファの要素のデータ型のサイズ) ・ ルートプロセス以外の場合 受信バッファ内の要素数*受信バッファの要素のデータ型のサイズ
MPIX_ALLGATHER_INIT	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(総プロセス数*受信バッファの要素数*受信バッファの要素のデータ型のサイズ)
MPIX_ALLGATHERV_INIT	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(各プロセスから受信するデータの要素数の合計*受信バッファの要素のデータ型のサイズ)
MPIX_ALLTOALL_INIT	(総プロセス数*各プロセスへ送信するデータの要素数*送信バッファの要素のデータ型のサイズ)+(総プロセス数*1つのプロセスから受信するデータの要素数*受信バッファの要素のデータ型のサイズ)
MPIX_ALLTOALLV_INIT	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(各プロセスから受信するデータの要素数の合計*受信バッファの要素のデータ型のサイズ)
MPIX_ALLTOALLW_INIT	(各プロセスの送信バッファの要素数*各プロセスの送信バッファのデータ型のサイズ)+(各プロセスの受信バッファの要素数*各プロセスの受信バッファのデータ型のサイズ)
MPIX_REDUCE_INIT	・ ルートプロセスの場合 送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ*2 ・ ルートプロセス以外の場合 送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ

MPIサブルーチン/関数名	計算式
MPIX_ALLREDUCE_INIT	送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ*2
MPIX_REDUCE_SCATTER_BLOCK_INIT	(バッファの要素のデータ型のサイズ*総プロセス数*ブロックごとの要素数)+(バッファの要素のデータ型のサイズ*ブロックごとの要素数)
MPIX_REDUCE_SCATTER_INIT	(各プロセスへ送信するデータの要素数の合計*バッファの要素のデータ型のサイズ)+(受信要素数*バッファの要素のデータ型のサイズ)
MPIX_SCAN_INIT	入力バッファの要素数*入力バッファの要素のデータ型のサイズ*2
MPIX_EXSCAN_INIT	入力バッファの要素数*入力バッファの要素のデータ型のサイズ*2
MPIX_NEIGHBOR_ALLGATHER_INIT	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(入力となる辺の数*受信バッファの要素数*受信バッファの要素のデータ型のサイズ)
MPIX_NEIGHBOR_ALLGATHERV_INIT	(送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(各入力となる辺の受信バッファの要素数の合計*受信バッファの要素のデータ型のサイズ)
MPIX_NEIGHBOR_ALLTOALL_INIT	(出力となる辺の数*送信バッファ内の要素数*送信バッファの要素のデータ型のサイズ)+(入力となる辺の数*受信バッファの要素数*受信バッファの要素のデータ型のサイズ)
MPIX_NEIGHBOR_ALLTOALLV_INIT	(各出力となる辺の送信バッファ内の要素数の合計*送信バッファの要素のデータ型のサイズ)+(各入力となる辺の受信バッファの要素数の合計*受信バッファの要素のデータ型のサイズ)
MPIX_NEIGHBOR_ALLTOALLW_INIT	(各出力となる辺の送信バッファ内の要素数の合計*各出力となる辺の送信バッファの要素のデータ型のサイズ)+(各入力となる辺の受信バッファの要素数の合計*各入力となる辺の受信バッファの要素のデータ型のサイズ)

3.2.2.4 CPU性能解析情報

CPU性能解析情報は、プログラム実行時のCPU動作状況を出力します。この情報は“3.1.4 プロファイルデータの計測”で指定した-Heventオプションの引数がpa1またはstatisticsの場合のみ出力します。

CPU性能解析情報の出力形式

Applicationの出力形式

```
Performance monitor event : statistics

    ********************************************************************************
       @level
    ********************************************************************************

                 Execution                Floating-point  Mem throughput  Mem throughput
      Kind        time(s)          GFLOPS   peak ratio(%)          (GB/s)   peak ratio(%)
    --------------------------------------------------------------------------------
       AVG          @time         @value1         @value2         @value3         @value4     @name @number
       MAX          @time         @value1         @value2         @value3         @value4
       MIN          @time         @value1         @value2         @value3         @value4

                 Execution                Floating-point  Mem throughput  Mem throughput
      Kind        time(s)          GFLOPS   peak ratio(%)          (GB/s)   peak ratio(%)
    --------------------------------------------------------------------------------

       AVG          @time         @value1         @value2         @value3         @value4     @name @number @sno
       MAX          @time         @value1         @value2         @value3         @value4
       MIN          @time         @value1         @value2         @value3         @value4

           Effective       Floating-point  SIMD inst.      SVE operation
      Kind      instruction       operation         rate(%)         rate(%)
    --------------------------------------------------------------------------------
       AVG          @value5         @value6         @value7         @value8                     @name @number
```

```
  MAX          @value5         @value6         @value7         @value8
  MIN          @value5         @value6         @value7         @value8

        Effective       Floating-point  SIMD inst.      SVE operation
  Kind      instruction       operation         rate(%)         rate(%)
-------------------------------------------------------------------------------------
  AVG          @value5         @value6         @value7         @value8                 @name @number @sno
  MAX          @value5         @value6         @value7         @value8
  MIN          @value5         @value6         @value7         @value8

  Kind             IPC             GIPS
-------------------------------------------------------------------------------------
  AVG          @value9        @value10                                                 @name @number
  MAX          @value9        @value10
  MIN          @value9        @value10

  Kind             IPC             GIPS
-------------------------------------------------------------------------------------
  AVG          @value9        @value10                                                 @name @number @sno
  MAX          @value9        @value10
  MIN          @value9        @value10
```

Processの出力形式

```
*************************************************************************************
   @level @pno @sno
*************************************************************************************

             Execution               Floating-point  Mem throughput  Mem throughput
  Kind        time(s)          GFLOPS   peak ratio(%)          (GB/s)   peak ratio(%)
-------------------------------------------------------------------------------------
  AVG           @time         @value1         @value2         @value3         @value4     @name @number
  MAX           @time         @value1         @value2         @value3         @value4
  MIN           @time         @value1         @value2         @value3         @value4

             Execution               Floating-point  Mem throughput  Mem throughput
  Kind        time(s)          GFLOPS   peak ratio(%)          (GB/s)   peak ratio(%)
-------------------------------------------------------------------------------------
  AVG           @time         @value1         @value2         @value3         @value4     @name @number
  MAX           @time         @value1         @value2         @value3         @value4
  MIN           @time         @value1         @value2         @value3         @value4

        Effective       Floating-point  SIMD inst.      SVE operation
  Kind      instruction       operation         rate(%)         rate(%)
-------------------------------------------------------------------------------------
  AVG          @value5         @value6         @value7         @value8                 @name @number
  MAX          @value5         @value6         @value7         @value8
  MIN          @value5         @value6         @value7         @value8

        Effective       Floating-point  SIMD inst.      SVE operation
  Kind      instruction       operation         rate(%)         rate(%)
-------------------------------------------------------------------------------------
  AVG          @value5         @value6         @value7         @value8                 @name @number
  MAX          @value5         @value6         @value7         @value8
  MIN          @value5         @value6         @value7         @value8

  Kind             IPC             GIPS
-------------------------------------------------------------------------------------
  AVG          @value9        @value10                                                 @name @number
  MAX          @value9        @value10
  MIN          @value9        @value10

  Kind             IPC             GIPS
```

```
-------------------------------------------------------------------------------------
   AVG          @value9        @value10                                      @name @number
   MAX          @value9        @value10
   MIN          @value9        @value10
```

Threadの出力形式

```
*****************************************************************************************
   @level @pno
*****************************************************************************************

                Execution                  Floating-point  Mem throughput  Mem throughput
                 time(s)          GFLOPS    peak ratio(%)          (GB/s)   peak ratio(%)
-----------------------------------------------------------------------------------------
                   @time         @value1         @value2         @value3         @value4     @name @number
                   @time         @value1         @value2         @value3         @value4     @name @number

          Effective       Floating-point  SIMD inst.      SVE operation
             instruction       operation         rate(%)         rate(%)
-----------------------------------------------------------------------------------------
                   @value5         @value6         @value7         @value8                 @name @number
                   @value5         @value6         @value7         @value8                 @name @number

                       IPC             GIPS
-----------------------------------------------------------------------------------------
                   @value9        @value10                                              @name @number
                   @value9        @value10                                              @name @number
```

表3.6 CPU動作状況の出力項目

出力項目	出力項目の意味
@level	情報集計レベル(Application、Process、Thread)
@pno	プロセス番号またはスレッド番号
@sno	spawn番号 (Spawn @num) の形式で入ります。(@numには数字が入ります) 本項目は以下のすべての条件を満たした場合のみ出力します。 ・ 計測対象がMPIプログラムである ・ fippコマンドの-Wspawnオプションが有効である、またはプログラム内にinfoキー “fjprof_spawn_dir_name”が設定されている ・ @levelがApplicationまたはProcessである ・ 動的に生成されたプロセスである
@time	計測対象区間の命令実行に要した時間(秒)
@value1	1秒あたりに実行された浮動小数点演算数 **注意** GFLOPS値はすべてActive elementとして算出しています。そのため、Inactive elementの多いプログラムでは本来のGFLOPS値より高い値を出力します。
@value2	浮動小数点演算性能の理論値に対する実測値の比率(%) **注意** 浮動小数点演算性能の理論値は倍精度演算として算出しています。そのため、単精度や半精度の演算の場合、実際の割合の2倍や4倍の値を出力します。

出力項目	出力項目の意味
	“3.1.4 プロファイルデータの計測”で-Hmode=user オプションを指定した場合、この出力項目は“--”固定です。
@value3	メモリースループット(GB/s)
@value4	メモリースループットの理論値に対する実測値の比率(%)
@value5	実行された命令の総数 **注意** 実行された命令の総数には、MOVPRFX命令を含みません。
@value6	実行された浮動小数点演算の総数
@value7	実行された命令の総数に対するSIMD命令数の比率(%)
@value8	実行された浮動小数点演算の総数に対するSVE演算数の比率(%)
@value9	1サイクルあたりに実行された命令数
@value10	1秒あたりに実行された命令数

注意

@timeおよび各@valueの出力結果は12桁以内を想定しています。そのため、出力結果が13桁を超える場合、見出しと出力結果にズレが生じます。

3.2.3 プロファイル結果の詳細(XML形式)

“3.1.5 プロファイル結果の出力”で-txmlオプションを指定した場合、XML形式による出力を行います。XML形式のフォーマットについて説明します。

3.2.3.1 XML形式の構成

プロファイラが出力するXML形式出力の構成について説明します。

プロファイラが出力するXML形式出力は全体を<profile>要素で囲みます。<profile>要素は<environment>および<information>の2つの要素から構成されます。

XML形式の構成

```
<?xml version="1.0" encoding="utf-8"?>                          XML宣言
<profile type="@type" version="@vid" output_version="@oid">     プロファイル情報
  <environment>                                                 プロファイルデータ計測環境情報
    ......
  </environment>
  <information item="advanced">                                 性能情報
    ......
  </information>
</profile>
```

要素の説明

要素名	要素の説明	概要
profile	プロファイル情報	プロファイラが出力したXML形式出力であることを示します。
environment	プロファイルデータ計測環境情報	TEXT形式の以下に相当する内容を格納します(※)。 ・“3.2.2.1 プロファイルデータ計測環境情報”

要素名	要素の説明	概要
information	性能情報	TEXT形式の以下に相当する内容を格納します(※)。 ・“3.2.2.2 時間統計情報” ・“3.2.2.4 CPU性能解析情報” ・“3.2.2.3 MPI通信コスト情報”

※ TEXT形式とXML形式では一致しない項目も存在します。

3.2.3.2 XML形式出力の詳細

XML形式出力で使用する各要素について説明します。なお、特に説明が無い場合、各要素の出力項目の内容はテキスト出力と同様です。

3.2.3.2.1 プロファイル情報 <profile>

詳細プロファイラが出力したXML形式出力であることを示します。

要素名	説明
profile	<profile type="@type" version="@vid" output_version="@oid"> </profile> 詳細プロファイラが出力したXML形式出力であることを示します。 @type にはプロファイラの種別が入ります。“fapp”固定です。 @vid にはプロファイラの版数が入ります。 @oid には出力フォーマットの版数が入ります。

3.2.3.2.2 プロファイルデータ計測環境情報 <environment>

プロファイルデータを計測した時の環境情報を出力します。

フォーマット

```
<environment>
  <measured_time unit="date">@date</measured_time>
  <type_of_program program="@program" />
  <coordinate x="@x" y="@y" z="@z"/>
  <vector_length vlen="@vlen"/>
  <spawn id="@id">
    <process id="@id">
      <host name="@name"/>
      <frequency unit="MHz">@frequency</frequency>
      <cntfrq unit="Hz">@cntfrq</cntfrq>
      <thread id="@id">
        <cmg id="@cmg"/>
        <core id="@core"/>
      </thread>
    </process>
  </spawn>
</environment>
```

要素の説明

要素名	説明
environment	<environment> </environment> プロファイルデータの計測環境情報であることを示します。
measured_time	<measured_time unit="date"> @date </measured_time> プロファイルデータの計測日時を示します。

要素名	説明
	@date にはプロファイリングデータの計測日時が YYYY-MM-DDThh:mm:ss 形式で入ります。
type_of_program	<type_of_program program="@program" /> プログラム実行形態を示します。 @program には以下のいずれかが入ります。 “SERIAL”：逐次 “Thread(AUTO)”：自動並列のみ “Thread(OpenMP)”：OpenMPのみ “Thread(OpenMP+AUTO)”：OpenMP,自動並列併用 “MPI”：MPIのみ “MPI+Thread(AUTO)”：MPI,自動並列 “MPI+Thread(OpenMP)”：MPI,OpenMP “MPI+Thread(OpenMP+AUTO)”：MPI,OpenMP,自動並列併用
coordinate	<coordinate x="@x" y="@y" z="@z" /> MPIプログラム実行時の論理形状を示します。 @x 、@y 、@z にはそれぞれx軸、y軸、z軸の値が入ります。 **注意** データ計測対象がMPIプログラムの場合のみ出力します。 また、動的に生成されたプロセスの論理形状は出力しません。
vector_length	<vector_length vlen="@vlen"/> SVEベクトル長(ビット数)を示します。 @vlen にはSVEベクトル長(ビット数)が入ります。
spawn	<spawn id="@id"> </spawn> spawn番号を示します。本要素は動的プロセスが存在する場合、複数回出力することがあります。 @id には、spawn番号が入ります。動的プロセスではない場合、@id には0が入ります。
process	<process id="@id"> </process> プロセス番号を示します。本要素はプロセスが複数存在する場合、複数回出力します。 @id にはプロセス番号が入ります。
host	<host name="@name"/> ホスト名を示します。 @nameにはホスト名が入ります。
frequency	<frequency unit="MHz"> @frequency </frequency> CPU周波数を示します。単位は、MHz(unit="MHz")です。 @frequency にはCPU周波数が入ります。 **注意** 以下のいずれかの条件に該当する場合、CPU周波数は“-”を出力します。 ・ “3.1.4 プロファイルデータの計測”で-Inocpupaオプションが有効な場合 ・ “3.1.4 プロファイルデータの計測”で-Hmethod=fastオプションを指定し、かつ“3.1.1 計測区間指定ルーチンの追加”で指定した計測区間の計測が1度も実施されなかった場合

要素名	説明
	・“3.1.4 プロファイルデータの計測”で-Hmode=userオプションを指定した場合
cntfrq	<cntfrq unit="Hz"> @cntfrq </cntfrq> タイマークロック周波数を示します。単位は、Hz(unit="Hz")です。 @cntfrqにはタイマークロック周波数が入ります。 **注意** “3.1.4 プロファイルデータの計測”で-Inocpupaオプションが有効な場合、タイマークロック周波数は“0”を出力します。
thread	<thread id="@id"> </thread> スレッド番号を示します。本要素はスレッドが複数存在する場合、複数回出力します。 @id にはスレッド番号が入ります。
cmg	<cmg id="@cmg"/> CMG番号を示します。 @cmg にはCMG番号が入ります。
core	<core id="@core"/> コア番号を示します。 @coreにはコア番号が入ります。

3.2.3.2.3 性能情報 <information>

詳細プロファイラの性能情報を出力します。性能情報では、プロセス毎やスレッド毎に、時間統計情報、CPU性能解析情報、MPI通信コスト情報を出力します。

フォーマット

```
<information item="advanced">
  <region name="@name" id="@number">
    <spawn id="@id">
      <process id="@id">
        <time>
          <elapsed unit="s">@elapsed</elapsed>
          <user unit="s">@user</user>
          <system unit="s">@system</system>
        </time>
        <thread id="@id">
          <call_count>@call_count</call_count>
          <time>
            <user unit="s">@user</user>
            <system unit="s">@system</system>
          </time>
          <cpupa>
            .....
          </cpupa>
        </thread>
        <mpi>
          .....
        </mpi>
      </process>
    </spawn>
  </region>
</information>
```

要素の説明

要素名	説明
information	<information item="@item"> </information> 詳細プロファイラ性能情報であることを示します。 @itemには性能情報の種別が入ります。“advanced”固定です。
region	<region name="@name" id="@number"> 計測区間名を示します。 @nameには構成するグループ名、@numberには詳細番号が入ります。
spawn	<spawn id="@id"> </spawn> spawn番号を示します。本要素は動的に生成されたプロセスが存在する場合、複数回出力することがあります。 @id には、spawn番号が入ります。動的に生成されたではない場合、@id には0が入ります。
process	<process id="@id"> </process> プロセス番号を示します。本要素はプロセスが複数存在する場合、複数回出力します。 @id にはプロセス番号が入ります。
time (process直下)	<time> </time> プロセスの時間統計情報であることを示します。
elapsed (process,time直下)	<elapsed unit="s" > @elapsed </elapsed> プロセスごとの経過時間を示します。単位は秒(unit="s")です。 @elapsed にはプロセスごとの経過時間が入ります。
user (process,time直下)	<user unit="s"> @user </user> プロセスごとのユーザーCPU時間を示します。単位は秒(unit="s")です。 @user にはプロセスごとのユーザーCPU時間が入ります。 プログラム実行形態が“SERIAL”または“MPI”以外の場合またはプロファイルデータの計測時に-Inocputimeオプションを指定した場合、この項目は出力されません。
system (process,time直下)	<system unit="s"> @system </system> プロセスごとのシステムCPU時間を示します。単位は秒(unit="s")です。 @system にはプロセスごとのシステムCPU時間が入ります。 プログラム実行形態が“SERIAL”または“MPI”以外の場合またはプロファイルデータの計測時に-Inocputimeオプションを指定した場合、この項目は出力されません。
thread	<thread id="@id"> </thread> スレッド番号を示します。本要素はスレッドが複数存在する場合、複数回出力します。 @id にはスレッド番号が入ります。
call_count	<call_count> @call_count </call_count> 呼び出し回数を示します。 @call_count には呼び出し回数が入ります。
time (thread直下)	<time> </time> スレッドの時間統計情報であることを示します。
user (thread,time直下)	<user unit="s"> @user </user> スレッドごとのユーザーCPU時間を示します。単位は秒(unit="s")です。 @user にはスレッドごとのユーザーCPU時間が入ります。

要素名	説明
	プログラム実行形態が“SERIAL”または“MPI”以外の場合またはプロファイルデータの計測時に-Inocputimeオプションを指定した場合、この項目は出力されません。
system (thread,time直下)	<system unit="s"> @system </system> スレッドごとのシステムCPU時間を示します。単位は秒(unit="s")です。 @system にはスレッドごとのシステムCPU時間が入ります。 プログラム実行形態が“SERIAL”または“MPI”以外の場合またはプロファイルデータの計測時に-Inocputimeオプションを指定した場合、この項目は出力されません。
cpupa	<cpupa> </cpupa> CPU性能解析情報であることを示します。詳細は“3.2.3.2.4 CPU性能解析情報 <cpupa>”を参照してください。
mpi	<mpi> </mpi> MPI通信コスト情報であることを示します。詳細は“3.2.3.2.5 MPI通信コスト情報 <mpi>”を参照してください。

3.2.3.2.4 CPU性能解析情報 <cpupa>

CPU性能解析情報には、プログラム実行時のCPU動作状況を出力します。

フォーマット

```
<cpupa>
  <event name="@name"> @event </event>
</cpupa>
```

要素の説明

要素名	説明
cpupa	<cpupa> </cpupa> CPU性能解析情報であることを示します。
event	<event name="@name"> @event </event> CNTVCT（Counter-timer Virtual Count）、PMCCNTR（Performance Monitors Cycle Counter）、およびPMUイベントごとの計測値を出力します。本要素はCNTVCTとPMCCNTRの2つに加えて、計測したPMUイベントの数だけ出力します。 @name にはPMUイベント番号が入ります。 @event にはPMUイベント番号に対応する計測結果が入ります。 CNTVCTとPMCCNTRについてはArm社のドキュメントやホームページを参照してください。PMUイベントについては https://github.com/fujitsu/A64FX/tree/master/doc/ 内の“A64FX PMU Events”を参照してください。

3.2.3.2.5 MPI通信コスト情報 <mpi>

MPI通信コスト情報には、MPIルーチンの実行回数、メッセージ長、実行時間、および待ち時間を出力します。

フォーマット（MPI_startルーチン、MPI_startallルーチン以外の場合）

```
<mpi>
  <function name="@name">
    <time>
      <elapsed unit="s">@elapsed</elapsed>
      <wait unit="s">@wait</wait>
    </time>
    <call_count>@call_count</call_count>
    <total_message_length>@total_message_length</total_message_length>
    <message_length_histogram>
      <call_count min_length="@min_length" max_length="@max_length">@call_count</call_count>
```

```
    </message_length_histogram>
  </function>
</mpi>
```

フォーマット(MPI_startルーチンまたはMPI_startallルーチンの場合)

```
<mpi>
  <function name="@name">
    <time>
      <elapsed unit="s">@elapsed</elapsed>
      <wait unit="s">@wait</wait>
    </time>
    <request>
      <function name="@name">
        <call_count>@call_count</call_count>
        <total_message_length>@total_message_length</total_message_length>
        <message_length_histogram>
          <call_count min_length="@min_length" max_length="@max_length">@call_count</call_count>
        </message_length_histogram>
      </function>
    </request>
  </function>
</mpi>
```

要素の説明

要素名	説明
mpi	<mpi> </mpi> MPI通信コスト情報であることを示します。
function	<function name="@name"> </function> MPIライブラリ名を示します。 @name にはMPIライブラリ名が入ります。
time	<time> </time> MPIライブラリの時間統計情報であることを示します。
elapsed	<elapsed unit="s" > @elapsed </elapsed> MPIライブラリごとの経過時間を示します。単位は秒(unit="s")です。 @elapsed にはMPIライブラリごとの経過時間が入ります。
wait	<wait unit="s" > @wait </wait> MPIライブラリごとの待ち時間を示します。単位は秒(unit="s")です。 @wait にはMPIライブラリごとの待ち時間が入ります。
request	<request> </request> MPI_startやMPI_startallライブラリの場合に詳細な情報を示します。
call_count(function直下)	<call_count> @call_count </call_count> MPIライブラリごとの呼び出し回数を示します。 @call_count にはMPIライブラリごとの呼び出し回数が入ります。
total_message_length	<total_message_length> @total_message_length </total_message_length> MPIライブラリごとの合計メッセージ長を示します。 @total_message_length には、MPIライブラリごとの合計メッセージ長が入ります。
message_length_histogram	<message_length_histogram> </message_length_histogram>

要素名	説明
	メッセージ長のヒストグラムであることを示します。
call_count （message_length_histogram直下）	<call_count min_length="@min_length" max_length="@max_length"> @call_count </call_count> メッセージ長の範囲ごとの呼び出し回数を示します。 @min_length にはメッセージ長の下限が入ります。 @max_length にはメッセージ長の上限が入ります。 @call_count には呼び出し回数が入ります。

第4章 CPU性能解析レポート

この章では、CPU性能解析レポートについて説明します。

CPU性能解析レポートは、複数の実行により計測した多数のCPU性能解析情報を集約し、表と付随するグラフを用いてわかりやすく可視化します。CPU性能解析情報の計測には詳細プロファイラを使用します。CPU性能解析レポートは印刷時にA3サイズ1枚に収まるよう設計されています。CPU性能解析レポートは、表示する情報の種類や粒度によって以下の4つの段階(単体レポート、簡易レポート、標準レポート、詳細レポート)を用意しています。CPU性能解析レポートのファイルはMicrosoft Excelのファイル形式(.xlsm)です。

単体レポート

作成に必要な計測回数がもっとも少ないCPU性能解析レポートです。実行時間、演算性能、メモリースループット、命令数についてのおおまかな情報を出力します。単体レポートを使用する場合、詳細プロファイラによる計測を1回実施してください。

簡易レポート

作成に必要な計測回数が少ないCPU性能解析レポートです。手軽にCPU性能解析レポートを使用したい場合、簡易レポートを推奨します。情報量は標準レポートより少なくなりますが、レポート作成のための計測回数は標準レポートよりも削減できます。簡易レポートを使用する場合、詳細プロファイラによる計測を5回実施してください。

標準レポート

標準的なCPU性能解析レポートです。通常の使用では標準レポートを推奨します。標準レポートを使用する場合、詳細プロファイラによる計測を11回実施してください。

詳細レポート

もっとも詳細なCPU性能解析レポートです。標準レポートでは情報量に不足がある場合、詳細レポートを推奨します。レポート作成のための計測回数はもっとも多くなりますが、CPU性能解析レポートの全情報を表示します。詳細レポートを表示する場合、詳細プロファイラによる計測を17回実施してください。

以下に、CPU性能解析レポートで参照可能な情報の一覧を示します。表の“○”はすべての情報が出力されること、“△”は一部の情報が出力されること、“-”は情報が出力されないことを示します。

表4.1 CPU性能解析レポート 表一覧

表タイトル	レポート種別				表の概要
	単体	簡易	標準	詳細	
Information	○	○	○	○	計測環境の情報およびユーザーの指定内容を表示します。
Statistics	△	○	○	○	メモリースループット、命令数、演算数などCPUの動作状況に関係する情報を表示します。 **[単体]** メモリースループット、命令数、演算数などCPUの動作状況に関係する情報を表示します。 **[簡易・標準・詳細]** 単体レポートの内容に加えて浮動小数点演算におけるActive elementの比率を表示します。
Cycle Accounting	-	△	○	○	プログラムの実行時間の内訳を表示します。 **[簡易]** プログラムの実行時間を9種類に分類して表示します。 **[標準・詳細]** プログラムの実行時間を20種類に分類して表示します。
Busy	-	△	△	○	プログラムのメモリー・キャッシュおよび演算パイプラインのビジー率に関する情報を表示します。 **[簡易]** 1次キャッシュ、2次キャッシュ、メモリー、浮動小数点演算パイプラインなどのビジー率、およびSFI(Store Fetch Interlock)の発生率を表示します。 **[標準]**

表タイトル	レポート種別				表の概要
	単体	簡易	標準	詳細	
					簡易レポートの内容に加えて整数演算パイプライン、アドレス計算用演算パイプライン、およびPredicate演算用演算パイプラインのビジー率を表示します。 **[詳細]** 標準レポートの内容に加えてL1パイプラインにおけるActive elementの比率を表示します。
Cache	-	△	○	○	キャッシュミスに関する情報を表示します。 **[簡易]** 簡易レポートの場合、1次データキャッシュおよび2次キャッシュのキャッシュミス数、およびロード・ストア命令数との比率を表示します。 **[標準・詳細]** 標準レポートまたは詳細レポートの場合、簡易レポートの内容に加えてキャッシュミス数の内訳を追加します。
Instruction	-	△	△	○	命令ミックスに関する情報を表示します。 **[簡易]** 命令ミックスに関する情報を9種類に分類して表示します。 **[標準]** 命令ミックスに関する情報を25種類に分類して表示します。 **[詳細]** 命令ミックスに関する情報を28種類に分類して表示します。
FLOPS	-	△	○	○	浮動小数点演算に関する情報を表示します。 **[簡易]** Active elementの比率を加味した浮動小数点演算性能を表示します。 **[標準・詳細]** 簡易レポートの内容に加えて各精度浮動小数点演算数を表示します。
Extra	-	-	-	○	ギャザー命令の内訳、および命令ミックスに含まれない命令に関する情報を表示します。
Hardware Prefetch Rate (%) (/Hardware Prefetch)	-	-	-	○	ハードウェアプリフェッチの内訳を表示します。
Data Transfer CMGs	-	-	-	○	ユーザーが指定したCMGに対する、全CMG、メモリー、Tofu、およびPCI間のスループット情報を表示します。
Power Consumption (W)	-	-	○	○	コア、L2キャッシュ、およびメモリーの消費電力を表示します。

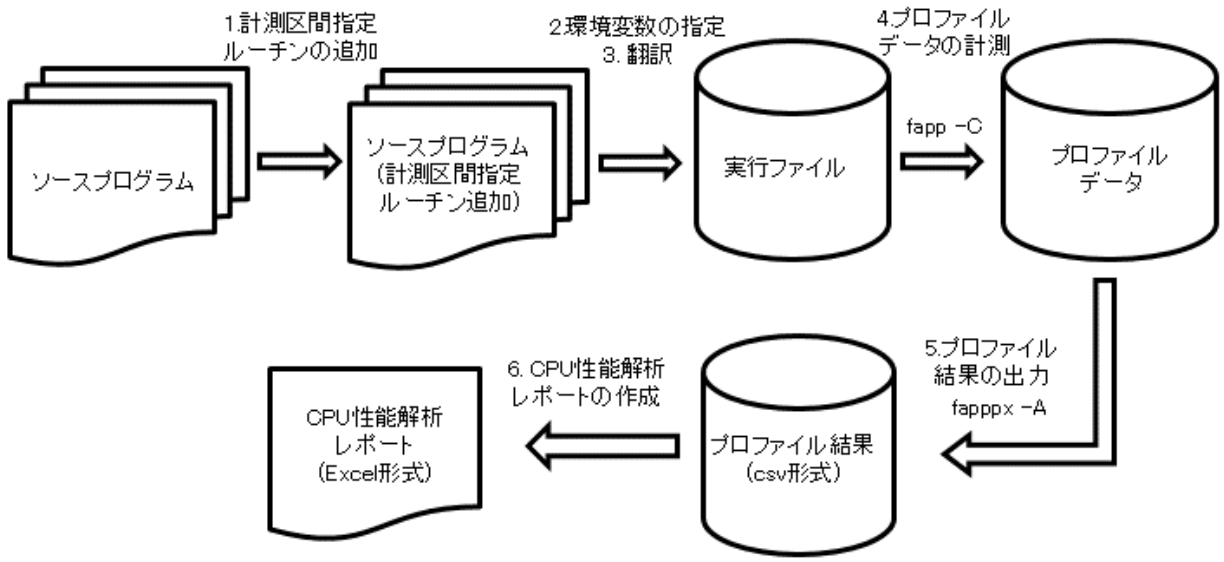
ポイント

各レポートで使用する入力ファイルは共通です。そのため、例えば標準レポートを作成後に詳細レポートに変更したい場合、追加の計測は差分の6回分だけ必要となります。

4.1 CPU性能解析レポートの使用手順

CPU性能解析レポートの使用手順を以下に記します。

図4.1 CPU性能解析レポートの使用手順



各操作の詳細を以下に示します。

4.1.1 計測範囲指定ルーチンの追加

CPU性能解析レポートで参照したい計測区間(“計測開始位置”および“計測終了位置”)の指定に必要な計測区間指定ルーチンをソースコードに追加します。詳細については“3.1.1 計測区間指定ルーチンの追加”を参照してください。

4.1.2 環境変数の指定

プロファイラ使用時に必要な環境変数を指定します。詳細については“2.1.2 環境変数の指定”を参照してください。

4.1.3 翻訳

プログラムを翻訳します。詳細については、“2.1.3 翻訳”を参照してください。

4.1.4 プロファイルデータの計測

詳細プロファイラのfappコマンドを使用して計測します。CPU性能解析レポート用にプロファイルデータを計測する場合、fappコマンドに-Heventオプションを指定する必要があります。fappコマンドについては“3.1.4 プロファイルデータの計測”を参照してください。

以下に、レポートごとの実行例を示します。

単体レポート

単体レポートを作成する場合、-Hevent=pa1を指定した1回の計測を実施してください。

```
# 単体レポート作成時の計測例
fapp -C -d ./rep1 -Hevent=pa1 ./a.out
```

簡易レポート

簡易レポートを作成する場合、-Hevent=pa1から-Hevent=pa5を指定した5回の計測を実施してください。

```
# 簡易レポート作成時の計測例
fapp -C -d ./rep1 -Hevent=pa1 ./a.out
fapp -C -d ./rep2 -Hevent=pa2 ./a.out
fapp -C -d ./rep3 -Hevent=pa3 ./a.out
```

```
fapp -C -d ./rep4 -Hevent=pa4 ./a.out
fapp -C -d ./rep5 -Hevent=pa5 ./a.out
```

標準レポート

標準レポートを作成する場合、-Hevent=pa1から-Hevent=pa11を指定した11回の計測を実施してください。

```
# 標準レポート作成時の計測例
fapp -C -d ./rep1 -Hevent=pa1 ./a.out
fapp -C -d ./rep2 -Hevent=pa2 ./a.out
fapp -C -d ./rep3 -Hevent=pa3 ./a.out
fapp -C -d ./rep4 -Hevent=pa4 ./a.out
fapp -C -d ./rep5 -Hevent=pa5 ./a.out
fapp -C -d ./rep6 -Hevent=pa6 ./a.out
fapp -C -d ./rep7 -Hevent=pa7 ./a.out
fapp -C -d ./rep8 -Hevent=pa8 ./a.out
fapp -C -d ./rep9 -Hevent=pa9 ./a.out
fapp -C -d ./rep10 -Hevent=pa10 ./a.out
fapp -C -d ./rep11 -Hevent=pa11 ./a.out
```

詳細レポート

詳細レポートを作成する場合、-Hevent=pa1から-Hevent=pa17を指定した17回の計測を実施してください。

```
# 詳細レポート作成時の計測例
fapp -C -d ./rep1 -Hevent=pa1 ./a.out
fapp -C -d ./rep2 -Hevent=pa2 ./a.out
fapp -C -d ./rep3 -Hevent=pa3 ./a.out
fapp -C -d ./rep4 -Hevent=pa4 ./a.out
fapp -C -d ./rep5 -Hevent=pa5 ./a.out
fapp -C -d ./rep6 -Hevent=pa6 ./a.out
fapp -C -d ./rep7 -Hevent=pa7 ./a.out
fapp -C -d ./rep8 -Hevent=pa8 ./a.out
fapp -C -d ./rep9 -Hevent=pa9 ./a.out
fapp -C -d ./rep10 -Hevent=pa10 ./a.out
fapp -C -d ./rep11 -Hevent=pa11 ./a.out
fapp -C -d ./rep12 -Hevent=pa12 ./a.out
fapp -C -d ./rep13 -Hevent=pa13 ./a.out
fapp -C -d ./rep14 -Hevent=pa14 ./a.out
fapp -C -d ./rep15 -Hevent=pa15 ./a.out
fapp -C -d ./rep16 -Hevent=pa16 ./a.out
fapp -C -d ./rep17 -Hevent=pa17 ./a.out
```

- CPU性能解析レポート用にプロファイルデータを計測する場合、fappコマンドに-Inocpupaオプションを指定しないでください。-Inocpupaオプションを指定した場合、-Heventオプションが無効となります。
- 1回の計測で指定できる-Heventオプションは1つだけです。複数の-Heventオプションを指定した場合、最後に指定した-Heventオプションが有効になります。
- すべての計測でプログラムの動作が同一である必要があります。例えば、入力データはすべての計測時に同じになるようにしてください。
- 計測の順番は順不同です。例えば、-Hevent=pa2を指定して計測した後に-Hevent=pa1を指定して計測しても問題ありません。
- 計測した-Heventオプションによって、“4.1.5プロファイル結果の出力”で出力するCSV形式ファイルの名前が決まっています。そのため、計測したプロファイルデータを格納するディレクトリ名は、指定した引数が判断できるよう命名することを推奨します。

4.1.5 プロファイル結果の出力

詳細プロファイラのfapppxコマンドまたはfappコマンドを使用して、CPU性能解析レポートに使用するCSV形式ファイルを出力します。CPU性能解析レポートの入力となるCSV形式ファイルを出力する場合、-tcsvオプションを指定してください。オプションの詳細については“3.1.5プロファイル結果の出力”を参照してください。以下に、レポートごとの実行例を示します。

単体レポート

単体レポートを作成する場合、-Hevent=pa1のプロファイルデータを使用してpa1.csvの1個のCSV形式ファイルを出力してください。

```
# "rep1"ディレクトリに格納したプロファイルデータを、CPU性能解析レポート用にCSV形式で出力する例
fapppx -A -d ./rep1 -Icpupa,nompi -tcsv -o pa1.csv
```

簡易レポート

簡易レポートを作成する場合、-Hevent=pa1から-Hevent=pa5のプロファイルデータを使用してpa1.csv～pa5.csvの5個のCSV形式ファイルを出力してください。

```
# "rep1"～"rep5"ディレクトリに格納したプロファイルデータを、CPU性能解析レポート用にCSV形式で出力する例
fapppx -A -d ./rep1 -Icpupa,nompi -tcsv -o pa1.csv
fapppx -A -d ./rep2 -Icpupa,nompi -tcsv -o pa2.csv
fapppx -A -d ./rep3 -Icpupa,nompi -tcsv -o pa3.csv
fapppx -A -d ./rep4 -Icpupa,nompi -tcsv -o pa4.csv
fapppx -A -d ./rep5 -Icpupa,nompi -tcsv -o pa5.csv
```

標準レポート

標準レポートを作成する場合、-Hevent=pa1から-Hevent=pa11のプロファイルデータを使用してpa1.csv～pa11.csvの11個のCSV形式ファイルを出力してください。

```
# "rep1"～"rep11"ディレクトリに格納したプロファイルデータを、CPU性能解析レポート用にCSV形式で出力する例
fapppx -A -d ./rep1 -Icpupa,nompi -tcsv -o pa1.csv
fapppx -A -d ./rep2 -Icpupa,nompi -tcsv -o pa2.csv
fapppx -A -d ./rep3 -Icpupa,nompi -tcsv -o pa3.csv
fapppx -A -d ./rep4 -Icpupa,nompi -tcsv -o pa4.csv
fapppx -A -d ./rep5 -Icpupa,nompi -tcsv -o pa5.csv
fapppx -A -d ./rep6 -Icpupa,nompi -tcsv -o pa6.csv
fapppx -A -d ./rep7 -Icpupa,nompi -tcsv -o pa7.csv
fapppx -A -d ./rep8 -Icpupa,nompi -tcsv -o pa8.csv
fapppx -A -d ./rep9 -Icpupa,nompi -tcsv -o pa9.csv
fapppx -A -d ./rep10 -Icpupa,nompi -tcsv -o pa10.csv
fapppx -A -d ./rep11 -Icpupa,nompi -tcsv -o pa11.csv
```

詳細レポート

詳細レポートを作成する場合、-Hevent=pa1から-Hevent=pa17のプロファイルデータを使用してpa1.csv～pa17.csvの17個のCSV形式ファイルを出力してください。

```
# "rep1"～"rep17"ディレクトリに格納したプロファイルデータを、CPU性能解析レポート用にCSV形式で出力する例
fapppx -A -d ./rep1 -Icpupa,nompi -tcsv -o pa1.csv
fapppx -A -d ./rep2 -Icpupa,nompi -tcsv -o pa2.csv
fapppx -A -d ./rep3 -Icpupa,nompi -tcsv -o pa3.csv
fapppx -A -d ./rep4 -Icpupa,nompi -tcsv -o pa4.csv
fapppx -A -d ./rep5 -Icpupa,nompi -tcsv -o pa5.csv
fapppx -A -d ./rep6 -Icpupa,nompi -tcsv -o pa6.csv
fapppx -A -d ./rep7 -Icpupa,nompi -tcsv -o pa7.csv
fapppx -A -d ./rep8 -Icpupa,nompi -tcsv -o pa8.csv
fapppx -A -d ./rep9 -Icpupa,nompi -tcsv -o pa9.csv
fapppx -A -d ./rep10 -Icpupa,nompi -tcsv -o pa10.csv
fapppx -A -d ./rep11 -Icpupa,nompi -tcsv -o pa11.csv
fapppx -A -d ./rep12 -Icpupa,nompi -tcsv -o pa12.csv
fapppx -A -d ./rep13 -Icpupa,nompi -tcsv -o pa13.csv
fapppx -A -d ./rep14 -Icpupa,nompi -tcsv -o pa14.csv
fapppx -A -d ./rep15 -Icpupa,nompi -tcsv -o pa15.csv
fapppx -A -d ./rep16 -Icpupa,nompi -tcsv -o pa16.csv
fapppx -A -d ./rep17 -Icpupa,nompi -tcsv -o pa17.csv
```

注意

- CPU性能解析レポート用にCSV形式ファイルを出力する場合、fapppxコマンドまたはfappコマンドに-Inocpupaオプションを指定しないでください。-Inocpupaオプションを指定した場合、CPU性能解析レポートに必要な情報が出力されません。
- CPU性能解析レポートに使用するCSV形式ファイル名は固定です。fappコマンドによる計測時に-Hevent=*event*オプションの*event*に指定した値に.csvを追加してファイル名としてください。なお、別のファイル名で出力後、手動でファイル名を変更しても問題ありません。
- 出力の順番は順不同です。例えば、pa2.csvを出力した後にpa1.csvを出力しても問題ありません。

4.1.6 CPU性能解析レポートの作成

“4.1.5 プロファイル結果の出力”で出力したCSV形式ファイルをCPU性能解析レポートに取り込みます。以下の手順で実施します。CPU性能解析レポートが出力するメッセージについては“4.1.6.1 CPU性能解析レポートのエラーメッセージおよび警告メッセージ”をご確認ください。

1. “4.1.5 プロファイル結果の出力”で出力したCSV形式ファイルおよびCPU性能解析レポートファイル(cpu_pa_report.xlsm)を同一ディレクトリ内に格納してください。CPU性能解析レポートファイルはログインノードの以下に格納されています。

   ```
   /製品インストールパス/misc/cpupa/cpu_pa_report.xlsm
   ```

   “*製品インストールパス*”については、システム管理者にお問い合わせください。

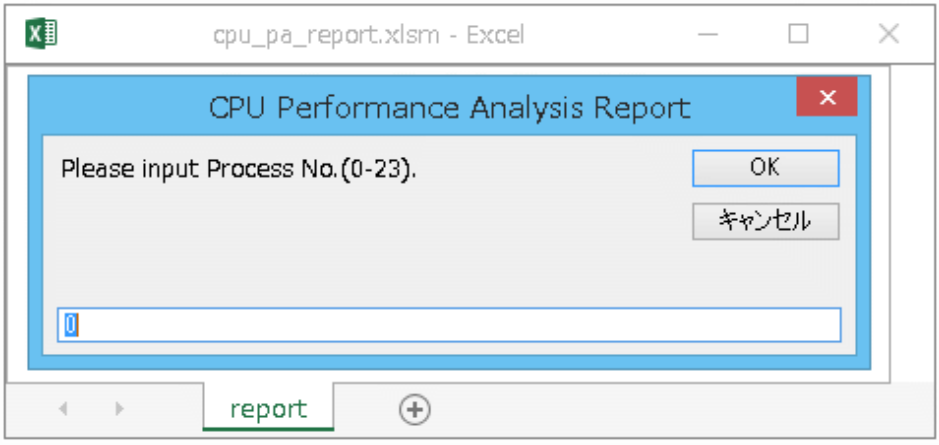
2. 1.で準備したディレクトリをMicrosoft Excelが実行できる環境にコピーします。
3. CPU性能解析レポートファイル(cpu_pa_report.xlsm)を起動します。

   注意

   CPU性能解析レポートはMicrosoft Excelのマクロ機能を使用しています。セキュリティ設定によってマクロが無効化された場合、手動でマクロを有効にしてください。マクロを有効にする方法についてはご利用されているMicrosoft Excelのヘルプなどをご参照ください。

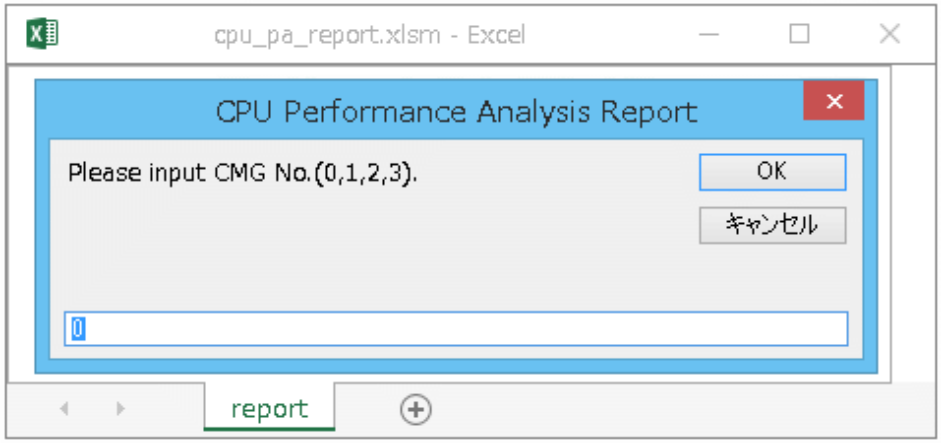
4. CPU性能解析レポートに出力する内容を選択するためのダイアログが表示されます。必要な情報を入力してください。
   a. CSV形式ファイル内に複数プロセスのデータが存在する場合、プロセス番号入力画面を表示します。参照したいプロセス番号を入力してください。pa1.csvのデータ内に入力したプロセス番号が存在しない場合、警告メッセージを表示してプロセス番号入力画面に戻ります。プロセスが1つしかない場合、プロセス番号入力画面を表示しません。

   図4.2 プロセス番号入力画面



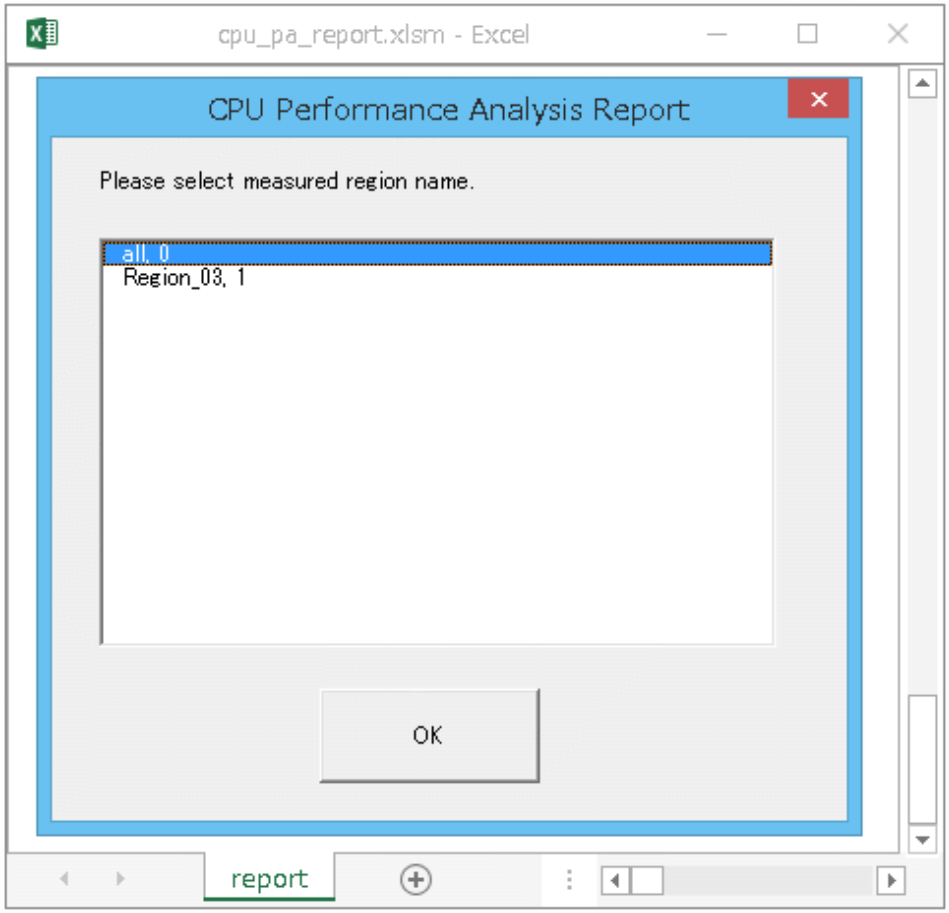
b. 指定したプロセスが複数のCMGで実行された場合、CMG番号入力画面を表示します。参照したいCMG番号を入力してください。pa1.csvのデータ内に入力したCMG番号が存在しない場合、警告メッセージを表示してCMG番号入力画面に戻ります。CMGが1つしかない場合、CMG番号入力画面を表示しません。

図4.3 CMG番号入力画面



c. 指定したプロセス内に計測区間指定ルーチンで指定した計測区間が複数存在する場合、下図の計測区間選択画面を表示します。参照したい計測区間名を選択してください。計測区間が1つしかない場合、計測区間選択画面を表示しません。

図4.4 計測区間選択画面



5. CSV形式ファイルの読み込みを開始します。読み込み対象となるファイルは、CPU性能解析レポートファイルと同一ディレクトリ内にあるpa1.csv～pa17.csvという名前のファイルのみです。正常終了した場合、作成したレポートごとに以下のメッセージを表示します。

メッセージ	メッセージ内容
CPU Performance Analysis Report (Single Report) created.	CPU性能解析レポート(単体レポート)を作成しました。
CPU Performance Analysis Report (Brief Report) created.	CPU性能解析レポート(簡易レポート)を作成しました。
CPU Performance Analysis Report (Standard Report) created.	CPU性能解析レポート(標準レポート)を作成しました。
CPU Performance Analysis Report (Detail Report) created.	CPU性能解析レポート(詳細レポート)を作成しました。

ポイント

作成するCPU性能解析レポートの種別はディレクトリ内のCSV形式ファイルによって決定します。優先順位は以下のとおりです。

1. pa1.csv～pa17.csvファイルがすべて存在する場合、詳細レポートを作成します。
2. pa1.csv～pa11.csvファイルがすべて存在する場合、標準レポートを作成します。
3. pa1.csv～pa5.csvファイルがすべて存在する場合、簡易レポートを作成します。
4. pa1.csvファイルが存在する場合、単体レポートを作成します。

注意

CPU性能解析レポートに読み込むCSV形式ファイルは、作成したいレポートにあわせて過不足がないようにしてください。入力ファイルに過不足があった場合、CPU性能解析レポートは作成されますが出力結果は保証しません。

注意

作成したCPU性能解析レポートファイルは自動で保存されません。必要に応じてファイルを保存してください。ただし、CSV形式ファイル読み込み済みのCPU性能解析レポートファイルは新規のCSV形式ファイル読み込み処理を行いません。そのため、CPU性能解析レポートファイルを流用したい場合、上書き保存ではなく別名保存をしてください。

4.1.6.1 CPU性能解析レポートのエラーメッセージおよび警告メッセージ

CPU性能解析レポートが出力するエラーメッセージおよび警告メッセージを以下に示します。エラーメッセージが出力された場合、Excelファイルを終了します。警告メッセージが出力された場合、処理を継続します。*paXX.csv* には該当する入力ファイル名が入ります。

表4.2 エラーメッセージ

エラーメッセージ	エラー内容
pa1.csv : Cannot open the file.	pa1.csv : ファイルが開けません。
paXX.csv : The version of the CPU Performance Analysis Report file does not match that of the fapppx command.	*paXX.csv* : CPU性能解析レポートファイルのバージョンがfapppxコマンドと一致しません。
paXX.csv : The file name does not match the argument in the -Hevent option.	*paXX.csv* : ファイル名が-Heventオプションの指定と一致しません。
paXX.csv : Data measured on a static process is not found.	*paXX.csv* : 静的プロセスのデータが見つかりません。
paXX.csv : The file format is not supported.	*paXX.csv* : そのファイルフォーマットはサポートしていません。
paXX.csv : No data found for specified process number.	*paXX.csv* : 指定されたプロセス番号のデータが見つかりません。
paXX.csv : No data found for specified CMG number.	*paXX.csv* : 指定されたCMG番号のデータが見つかりません。

エラーメッセージ	エラー内容
paXX.csv : No data found for specified region name.	*paXX.csv* : 指定された計測区間名のデータが見つかりません。
paXX.csv : The environment seems to not bind process to core.	*paXX.csv* :CPUバインドがされていない、またはCPUバインドが適切ではないデータが含まれています。
An unknown error has occurred.	原因不明のエラーが発生しました。
Missing input files.	入力ファイルが不足しています。

表4.3 警告メッセージ

警告メッセージ	警告内容
Data measured on a dynamically generated process is ignored.	動的に生成されたプロセスのデータが含まれていますが、無視します。
No data found for specified process number.	指定されたプロセス番号のデータが見つかりません。(プロセス番号入力画面時)
No data found for specified CMG number.	指定されたCMG番号のデータが見つかりません。(CMG番号入力画面時)
paXX.csv : The measured time difference with pa1 exceeds 5%. (XX=*XX...*)	*pa*XX.*csv* : pa1との計測区間の実行時間の差が5%を超えました。 **注意** 本メッセージに限り、メッセージ先頭のpaXX.csvは固定です。計測時間の差が検知されたファイルの番号は(XX=*XX...*)にカンマ区切りで番号のみ出力されます。例えば、pa2.csvとpa5.csvの結果に差異が見つかった場合、(XX=2,5)と出力されます。 本メッセージが出力された場合、出力結果の精度が低くなる可能性があります。該当するCSVファイルを再計測することを推奨します。
Because of the short measured time, there may be large error in the results. The average measured time of the region is less than 150 microseconds.	計測区間の実行時間が短いため、結果に含まれる測定誤差が多いかもしれません。計測区間の1回あたりの実行時間が平均で150マイクロ秒未満です。

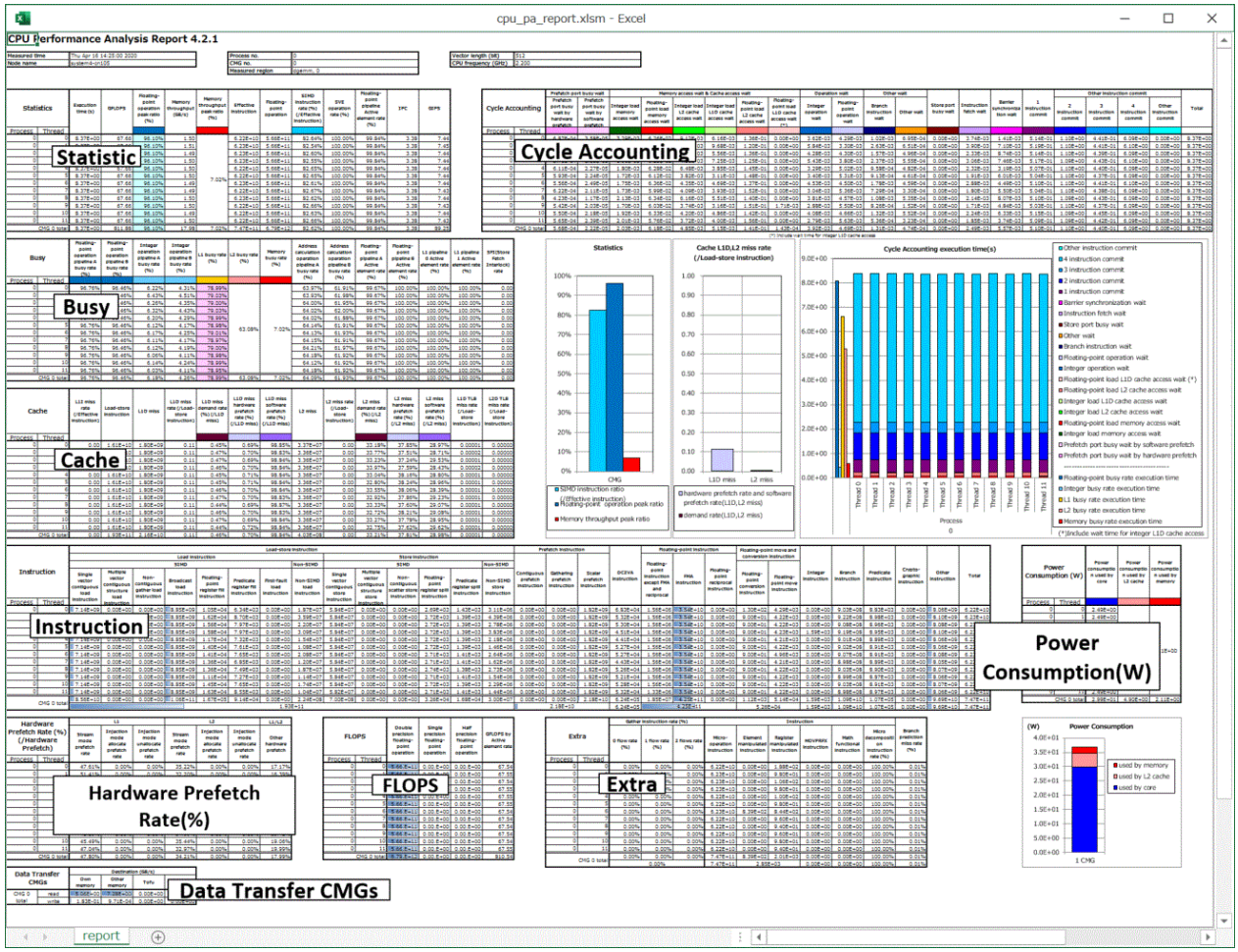
4.2 CPU性能解析レポート出力結果

CPU性能解析レポートの出力結果について説明します。

4.2.1 CPU性能解析レポート出力結果の概要

CPU性能解析レポートは以下の構成となっています。各表で参照できる情報の一覧については“第4章 CPU性能解析レポート”を参照してください。

図4.5 CPU性能解析レポートの構成



それぞれの表は以下の構成となっています。

図4.6 表の構成

表タイトル	Cycle Accounting		Operation wait		Other wait		項目名
			Integer operation wait	Floating-point operation wait	Branch instruction wait	Other wait	
	Process	Thread					項目色
プロセス番号 ＋ スレッド番号	0	0	5.27E-06	4.14E-07	4.92E-06	1.86E-05	算出値
	0	1	2.47E-06	3.34E-07	1.85E-06	1.36E-05	
	0	2	3.43E-06	3.91E-07	1.81E-06	1.48E-05	
	0	3	3.35E-06	3.56E-07	2.12E-06	1.49E-05	
	0	4	2.51E-06	4.36E-07	2.15E-06	1.37E-05	
	0	5	3.30E-06	3.46E-07	2.08E-06	1.49E-05	
	0	6	3.20E-06	4.47E-07	2.20E-06	1.47E-05	
	0	7	3.50E-06	3.39E-07	2.12E-06	1.55E-05	
	0	8	3.22E-06	4.13E-07	2.08E-06	1.49E-05	
	0	9	3.36E-06	3.27E-07	2.05E-06	1.49E-05	
	0	10	3.25E-06	4.37E-07	1.63E-06	1.53E-05	
	0	11	3.15E-06	3.78E-07	2.52E-06	1.43E-05	
CMG番号		CMG 0 total	3.33E-06	3.85E-07	2.29E-06	1.50E-05	代表値

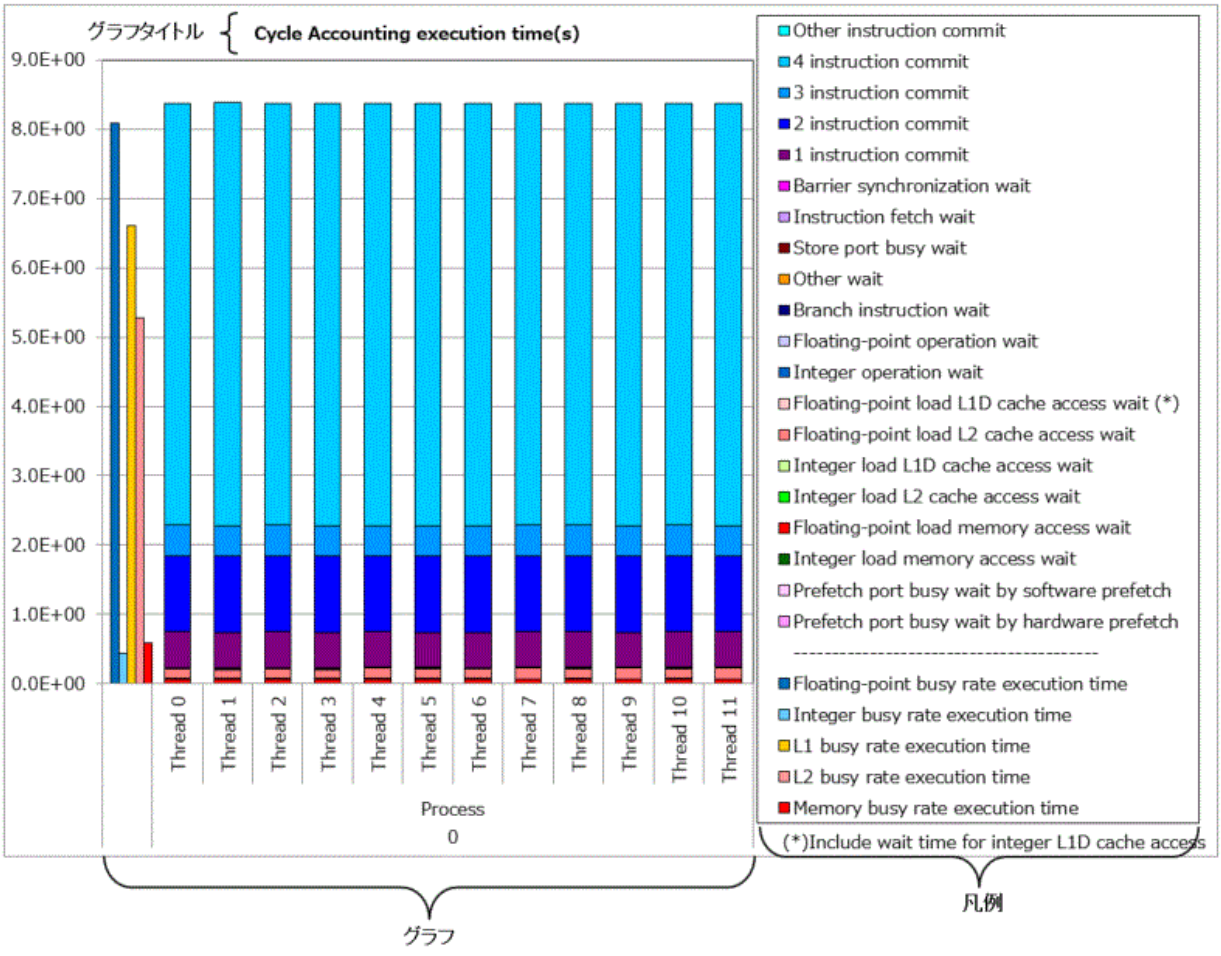
表内の項目の説明を以下に記します。

表4.4 各項目の説明

名称	説明
表タイトル	表のタイトルです。“4.2.2 CPU性能解析レポート出力結果の詳細”配下に同名の見出しがありますので、詳細な情報については該当する項目を参照してください。
プロセス番号+スレッド番号	プロセス番号、スレッド番号を示します。
CMG番号	CMG番号を示します。
項目名	項目の分類および項目名を表示します。
項目色	表に対応するグラフにおける色を示します。
算出値	各プロセス、各スレッドに対応する項目の値を小数第2位まで表示します。
代表値	算出値の平均値または合計値を示します。平均値、合計値のどちらを示すかは表によって異なります。代表値が存在しない表もあります。

一部の表については、グラフも用意しています。

図4.7 グラフの構成



グラフ内の項目の説明を以下に記します。

表4.5 各項目の説明

名称	説明
グラフタイトル	グラフのタイトルです。“対応する表タイトル”+“グラフの概要”という命名規則です。
グラフ	表の内容をグラフで表示します。グラフの種別は項目によって異なります。

名称	説明
凡例	グラフの凡例です。グラフの凡例色は表の“項目色”と一致します。

4.2.2 CPU性能解析レポート出力結果の詳細

CPU性能解析レポートファイルに存在する表およびグラフの詳細を以下に示します。

4.2.2.1 Information

Informationでは、実測結果を元に計測環境の情報およびユーザーの指定内容を表示します。

図4.8 Informationのレイアウト

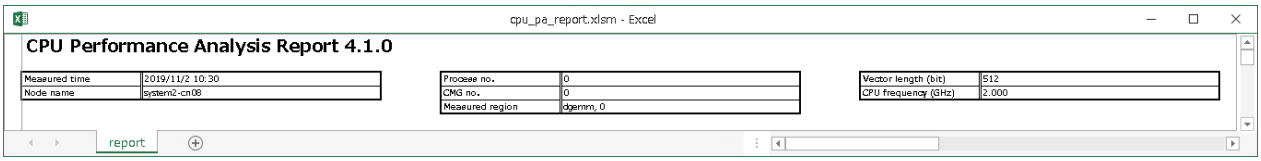


表4.6 Informationの出力項目

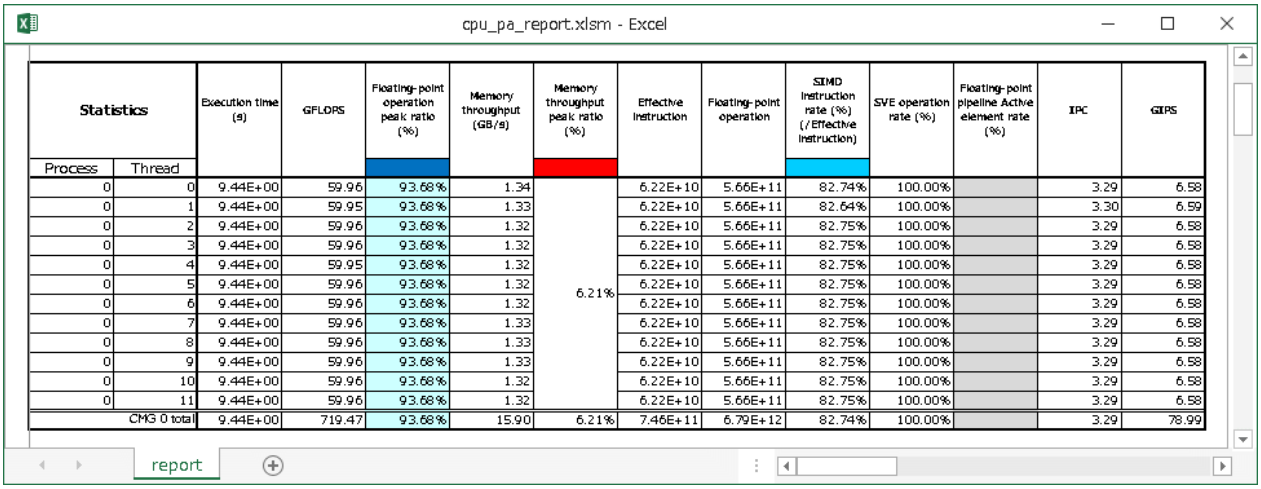
項目	説明
Measured time	計測日時
Node name	プロセスを実行したノード名
Process no.	指定したプロセス番号
CMG no.	指定したCMG番号
Measured region	指定した計測区間名
Vector length (bit)	ベクトル長 (bit)
CPU frequency(GHz)	実測結果から算出のCPUのクロック周波数 (GHz) **注意** “4.1.4 プロファイルデータの計測”で-Hmode=user オプションまたは-Hmethod=normalオプションを指定した場合、CPU frequency(GHz)の値は保証しません。 実測結果を元に算出しているため、プロファイルデータ計測環境情報のCPU周波数とは異なる可能性があります。

4.2.2.2 Statistics

Statisticsでは、メモリースループット、命令数、演算数などCPUの動作状況に関係する情報を表示します。簡易レポート、標準レポート、および詳細レポートの場合、浮動小数点演算におけるActive elementの比率も表示します。セルの背景色が水色に変化している場合、その項目の演算性能が良好であることを示します。Statisticsには対応する表とグラフがあります。“表4.7 Statistics(単体レポート)の出力項目”または“表4.8 Statistics(簡易レポート、標準レポート、および詳細レポート)の出力項目”で青字記載されている項目について棒グラフで表示します。

4.2.2.2.1 Statistics (単体レポート)

図4.9 Statistics(単体レポート)のレイアウト



Statistics		Execution time (s)	GFLOPS	Floating-point operation peak ratio (%)	Memory throughput (GB/s)	Memory throughput peak ratio (%)	Effective instruction	Floating-point operation	SIMD instruction rate (%) (/Effective instruction)	SVE operation rate (%)	Floating-point pipeline Active element rate (%)	IPC	GIPS
Process	Thread												
0	0	9.44E+00	59.96	93.68%	1.34		6.22E+10	5.66E+11	82.74%	100.00%		3.29	6.58
0	1	9.44E+00	59.95	93.68%	1.33		6.22E+10	5.66E+11	82.64%	100.00%		3.30	6.59
0	2	9.44E+00	59.96	93.68%	1.32		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	3	9.44E+00	59.96	93.68%	1.32		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	4	9.44E+00	59.95	93.68%	1.32		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	5	9.44E+00	59.96	93.68%	1.32	6.21%	6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	6	9.44E+00	59.96	93.68%	1.32		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	7	9.44E+00	59.96	93.68%	1.33		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	8	9.44E+00	59.96	93.68%	1.33		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	9	9.44E+00	59.96	93.68%	1.33		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	10	9.44E+00	59.96	93.68%	1.32		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
0	11	9.44E+00	59.96	93.68%	1.32		6.22E+10	5.66E+11	82.75%	100.00%		3.29	6.58
	CMG 0 total	9.44E+00	719.47	93.68%	15.90	6.21%	7.46E+11	6.79E+12	82.74%	100.00%		3.29	78.99

図4.10 Statistics(単体レポート)のグラフ

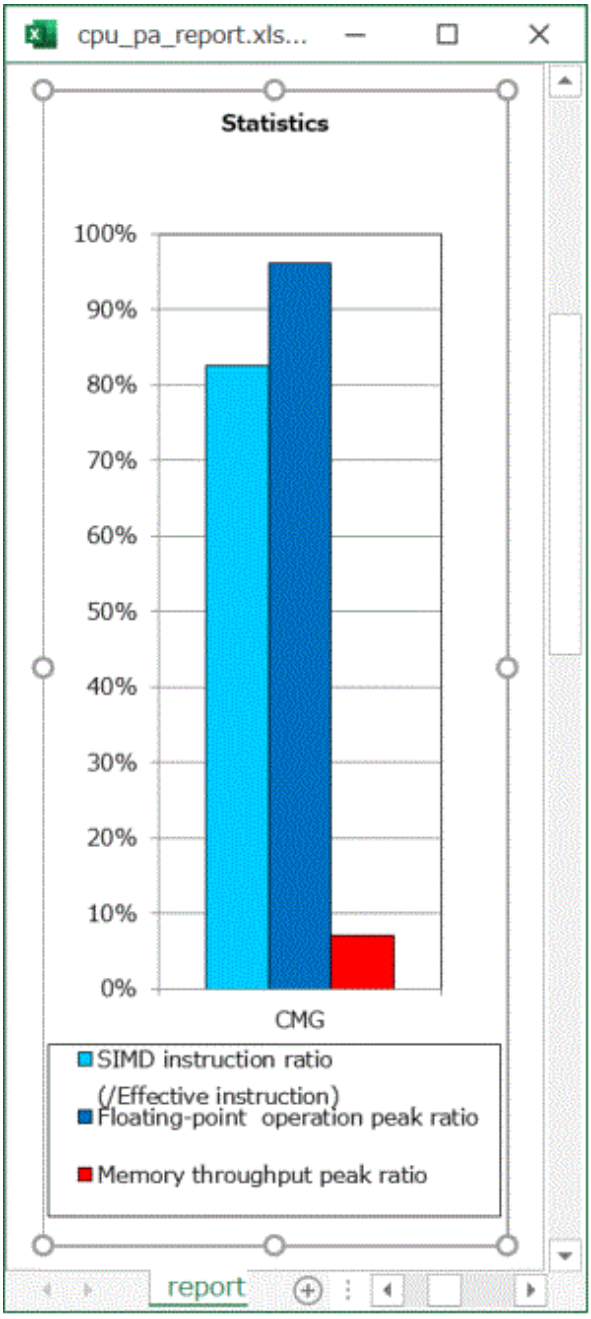


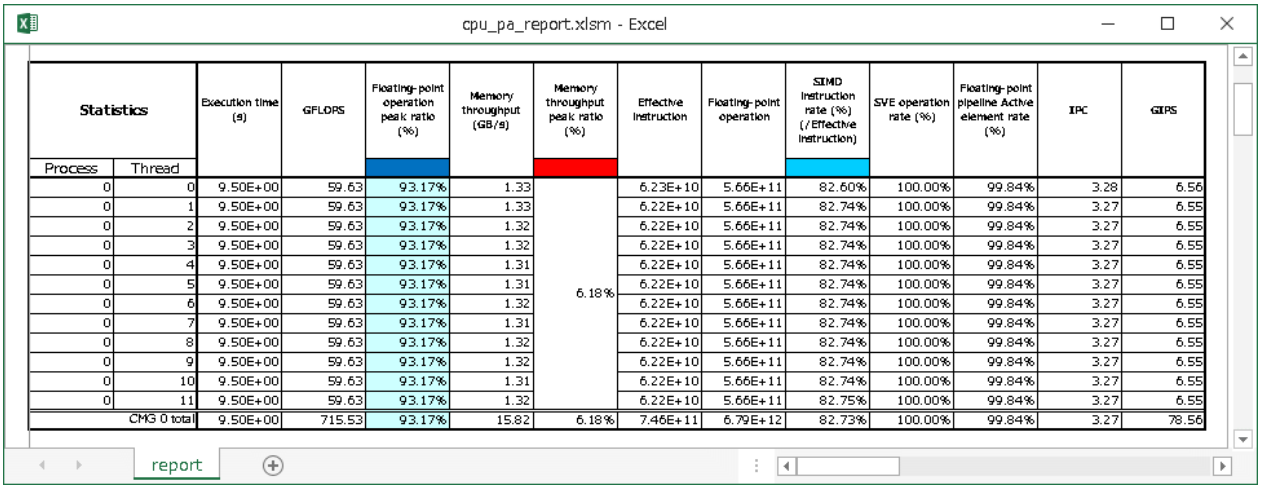
表4.7 Statistics(単体レポート)の出力項目

項目	説明
Execution time(s)	計測対象区間の命令実行に要した時間(秒)
GFLOPS	1秒あたりに実行された浮動小数点演算数 注意 GFLOPS値はすべてActive elementとして算出しています。そのため、Inactive elementの多いプログラムでは本来のGFLOPS値より高い値を出力します。
Floating-point operation peak ratio(%)	浮動小数点演算性能の理論値に対する実測値の比率(%)

項目	説明
	注意 浮動小数点演算性能の理論値は倍精度演算として算出しています。そのため、単精度や半精度の演算の場合、実際の割合の2倍や4倍の値を出力します。 “4.1.4 プロファイルデータの計測”で-Hmode=user オプションまたは-Hmethod=normalオプションを指定した場合、Floating-point operation peak ratio(%)の値は保証しません。
Memory throughput(GB/s)	メモリースループット(GB/s)
Memory throughput peak ratio(%)	メモリースループットの理論値に対する実測値の比率(%) **注意** • メモリースループットの理論値は、1つのCMGを対象としています。自CMGのメモリーだけでなく他CMGのメモリーも同時にアクセスすると、メモリースループットの実測値は理論値を超える場合があります。このときには、メモリースループットピーク比は、100%を超える可能性があります。 • メモリースループットの理論値は、CMG内の全てのコアから同時にメモリーにアクセスすることを前提としています。しかし、実際には、各コアが異なるタイミングでメモリーアクセスを行うことがあります。このような場合、全コアのメモリーアクセス量を計測区間内で集計すると、異なるタイミングのメモリーアクセスを同時アクセスとして集計することになります。その結果、メモリーアクセス量と実行時間から求めたメモリースループットの実測値が理論値を超える場合があります。このときには、メモリースループットピーク比は、100%を超える可能性があります。
Effective instruction	実行された命令の総数 **注意** 実行された命令の総数には、MOVPRFX命令を含みません。
Floating-point operation	実行された浮動小数点演算の総数
SIMD Instruction rate(%) (/Effective Instruction)	実行された命令の総数に対するSIMD命令数の比率(%)
SVE operation rate(%)	実行された浮動小数点演算の総数に対するSVE演算数の比率(%)
IPC	1サイクルあたりに実行された命令数
GIPS	1秒あたりに実行された命令数

4.2.2.2.2 Statistics (簡易レポート、標準レポート、および詳細レポート)

図4.11 Statistics(簡易レポート、標準レポート、および詳細レポート)のレイアウト



Statistics		Execution time (s)	GFLOPS	Floating-point operation peak ratio (%)	Memory throughput (GB/s)	Memory throughput peak ratio (%)	Effective instruction	Floating-point operation	SIMD instruction rate (%) (/Effective instruction)	SVE operation rate (%)	Floating-point pipeline Active element rate (%)	IPC	GIPS
Process	Thread												
0	0	9.50E+00	59.63	93.17%	1.33		6.23E+10	5.66E+11	82.60%	100.00%	99.84%	3.28	6.56
0	1	9.50E+00	59.63	93.17%	1.33		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	2	9.50E+00	59.63	93.17%	1.32		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	3	9.50E+00	59.63	93.17%	1.32		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	4	9.50E+00	59.63	93.17%	1.31		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	5	9.50E+00	59.63	93.17%	1.31	6.18%	6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	6	9.50E+00	59.63	93.17%	1.32		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	7	9.50E+00	59.63	93.17%	1.31		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	8	9.50E+00	59.63	93.17%	1.32		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	9	9.50E+00	59.63	93.17%	1.32		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	10	9.50E+00	59.63	93.17%	1.31		6.22E+10	5.66E+11	82.74%	100.00%	99.84%	3.27	6.55
0	11	9.50E+00	59.63	93.17%	1.32		6.22E+10	5.66E+11	82.75%	100.00%	99.84%	3.27	6.55
	CMG 0 total	9.50E+00	715.53	93.17%	15.82	6.18%	7.46E+11	6.79E+12	82.73%	100.00%	99.84%	3.27	78.56

図4.12 Statistics(簡易レポート、標準レポート、および詳細レポート)のグラフ

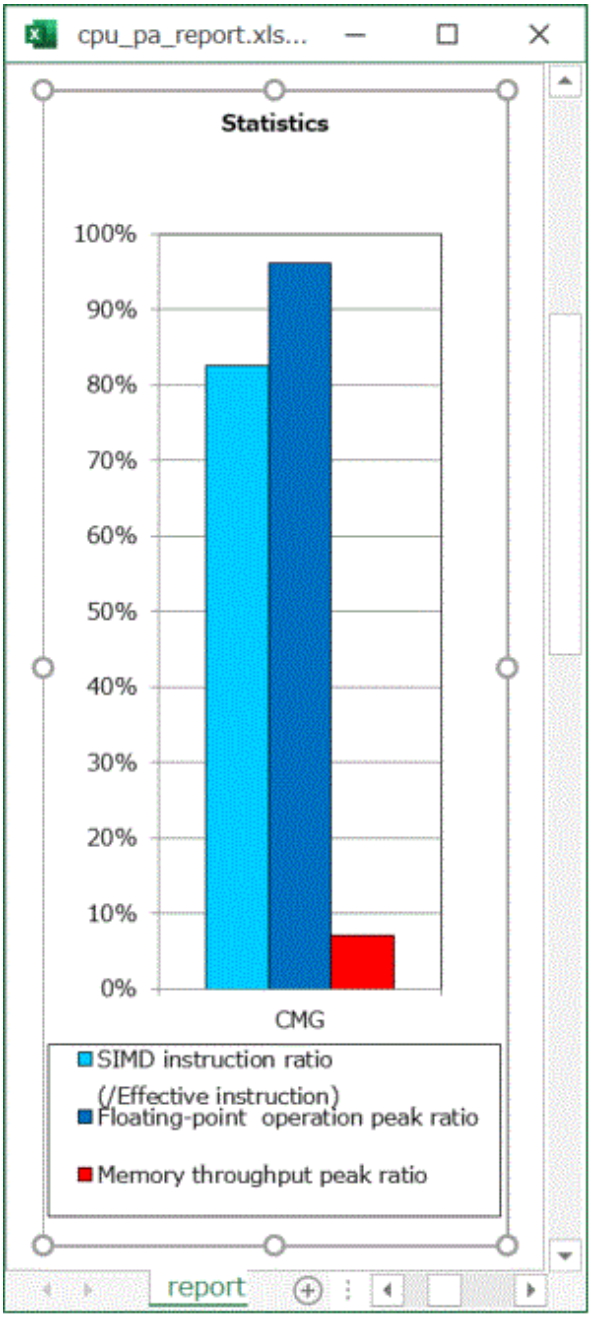


表4.8 Statistics(簡易レポート、標準レポート、および詳細レポート)の出力項目

項目	説明
Execution time(s)	計測対象区間の命令実行に要した時間(秒)
GFLOPS	1秒あたりに実行された浮動小数点演算数 **注意** GFLOPS値はすべてActive elementとして算出しています。そのため、Inactive elementの多いプログラムでは本来のGFLOPS値より高い値を出力します。
Floating-point operation peak ratio(%)	浮動小数点演算性能の理論値に対する実測値の比率(%)

項目	説明
	注意 浮動小数点演算性能の理論値は倍精度演算として算出しています。そのため、単精度や半精度の演算の場合、実際の割合の2倍や4倍の値を出力します。 “4.1.4 プロファイルデータの計測”で-Hmode=user オプションまたは-Hmethod=normalオプションを指定した場合、Floating-point operation peak ratio(%)の値は保証しません。
Memory throughput(GB/s)	メモリースループット(GB/s)
Memory throughput peak ratio(%)	メモリースループットの理論値に対する実測値の比率(%) **注意** ・ メモリースループットの理論値は、1つのCMGを対象としています。自CMGのメモリーだけでなく他CMGのメモリーも同時にアクセスすると、メモリースループットの実測値は理論値を超える場合があります。このときには、メモリースループットピーク比は、100%を超える可能性があります。 ・ メモリースループットの理論値は、CMG内の全てのコアから同時にメモリーにアクセスすることを前提としています。しかし、実際には、各コアが異なるタイミングでメモリーアクセスを行うことがあります。このような場合、全コアのメモリーアクセス量を計測区間内で集計すると、異なるタイミングのメモリーアクセスを同時アクセスとして集計することになります。その結果、メモリーアクセス量と実行時間から求めたメモリースループットの実測値が理論値を超える場合があります。このときには、メモリースループットピーク比は、100%を超える可能性があります。
Effective instruction	実行された命令の総数 **注意** 実行された命令の総数には、MOVPRFX命令を含みません。
Floating-point operation	実行された浮動小数点演算の総数
SIMD Instruction rate(%) (/Effective Instruction)	実行された命令の総数に対するSIMD命令数の比率(%)
SVE operation rate(%)	実行された浮動小数点演算の総数に対するSVE演算数の比率(%)
Floating-point pipeline Active element rate(%)	浮動小数点演算におけるActive elementの比率(%) **注意** ・ SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が100%に近い場合のみ、限定的に利用することを推奨します。 ・ store命令の比率が高いと、Active elementの比率の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。 ・ 1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。
IPC	1サイクルあたりに実行された命令数

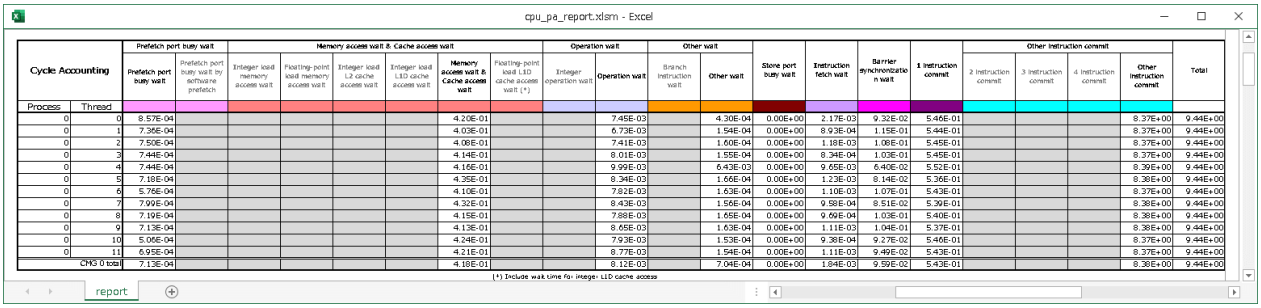
項目	説明
GIPS	1秒あたりに実行された命令数

4.2.2.3 Cycle Accounting

プログラムの実行時間の内訳を表示します(単位:秒)。簡易レポートでは9種類、標準レポートおよび詳細レポートでは20種類に分類して表示します。Cycle Accountingには対応する表とグラフがあります。“表4.9 Cycle Accounting(簡易レポート)の出力項目”または“表4.10 Cycle Accounting(標準レポートおよび詳細レポート)の出力項目”の出力項目(Totalを除く)を積み上げ棒グラフで表示します。また、比較対象として“4.2.2.4 Busy”の出力項目“L1 busy rate(%) ”、“L2 busy rate(%)”、“Memory busy rate(%)”、“Floating-point operation pipeline busy rate(%)”、“Integer operation pipeline busy rate(%)”の値を実行時間に換算して棒グラフで表示します。

4.2.2.3.1 Cycle Accounting(簡易レポート)

図4.13 Cycle Accounting(簡易レポート)のレイアウト



cpu_pa_report.xlsm - Excel

Cycle Accounting		Prefetch port busy wait		Memory access wait & Cache access wait					Operation wait		Other wait		Store port busy wait	Instruction fetch wait	Barrier synchronization wait	1 instruction commit	Other instruction commit				Total
		Prefetch port busy wait	Prefetch port busy wait by software prefetch	Integer load memory access wait	Floating-point load memory access wait	Integer load L2 cache access wait	Integer load L1D cache access wait	Memory access wait & Cache access wait	Floating-point load L1D cache access wait (*)	Integer operation wait	Operation wait	Branch instruction wait	Other wait					2 instruction commit	3 instruction commit	4 instruction commit	Other instruction commit	
Process	Thread																					
0	0	8.57E-04						4.20E-01			7.45E-03		4.30E-04	0.00E+00	2.17E-03	9.32E-02	5.46E-01				8.37E+00	9.44E+00
0	1	7.36E-04						4.03E-01			6.73E-03		1.54E-04	0.00E+00	8.93E-04	1.15E-01	5.44E-01				8.37E+00	9.44E+00
0	2	7.50E-04						4.08E-01			7.41E-03		1.60E-04	0.00E+00	1.18E-03	1.08E-01	5.45E-01				8.37E+00	9.44E+00
0	3	7.44E-04						4.14E-01			8.01E-03		1.55E-04	0.00E+00	8.34E-04	1.03E-01	5.45E-01				8.37E+00	9.44E+00
0	4	7.44E-04						4.16E-01			9.99E-03		6.43E-03	0.00E+00	9.65E-03	6.40E-02	5.52E-01				8.39E+00	9.44E+00
0	5	7.18E-04						4.35E-01			8.34E-03		1.66E-04	0.00E+00	1.23E-03	8.14E-02	5.36E-01				8.38E+00	9.44E+00
0	6	5.76E-04						4.10E-01			7.82E-03		1.63E-04	0.00E+00	1.10E-03	1.07E-01	5.43E-01				8.37E+00	9.44E+00
0	7	7.99E-04						4.32E-01			8.43E-03		1.56E-04	0.00E+00	9.88E-04	8.51E-02	5.39E-01				8.38E+00	9.44E+00
0	8	7.19E-04						4.15E-01			7.88E-03		1.65E-04	0.00E+00	9.69E-04	1.03E-01	5.40E-01				8.38E+00	9.44E+00
0	9	7.13E-04						4.13E-01			8.65E-03		1.63E-04	0.00E+00	1.11E-03	1.04E-01	5.37E-01				8.38E+00	9.44E+00
0	10	5.06E-04						4.24E-01			7.93E-03		1.53E-04	0.00E+00	9.38E-04	9.27E-02	5.46E-01				8.37E+00	9.44E+00
0	11	6.95E-04						4.21E-01			8.77E-03		1.54E-04	0.00E+00	1.11E-03	9.49E-02	5.43E-01				8.37E+00	9.44E+00
CHS 0 total		7.13E-04						4.18E-01			8.12E-03		7.04E-04	0.00E+00	1.84E-03	9.59E-02	5.43E-01				8.38E+00	9.44E+00

(*) Include wait time for integer L1D cache access

report

図4.14 Cycle Accounting(簡易レポート)のグラフ

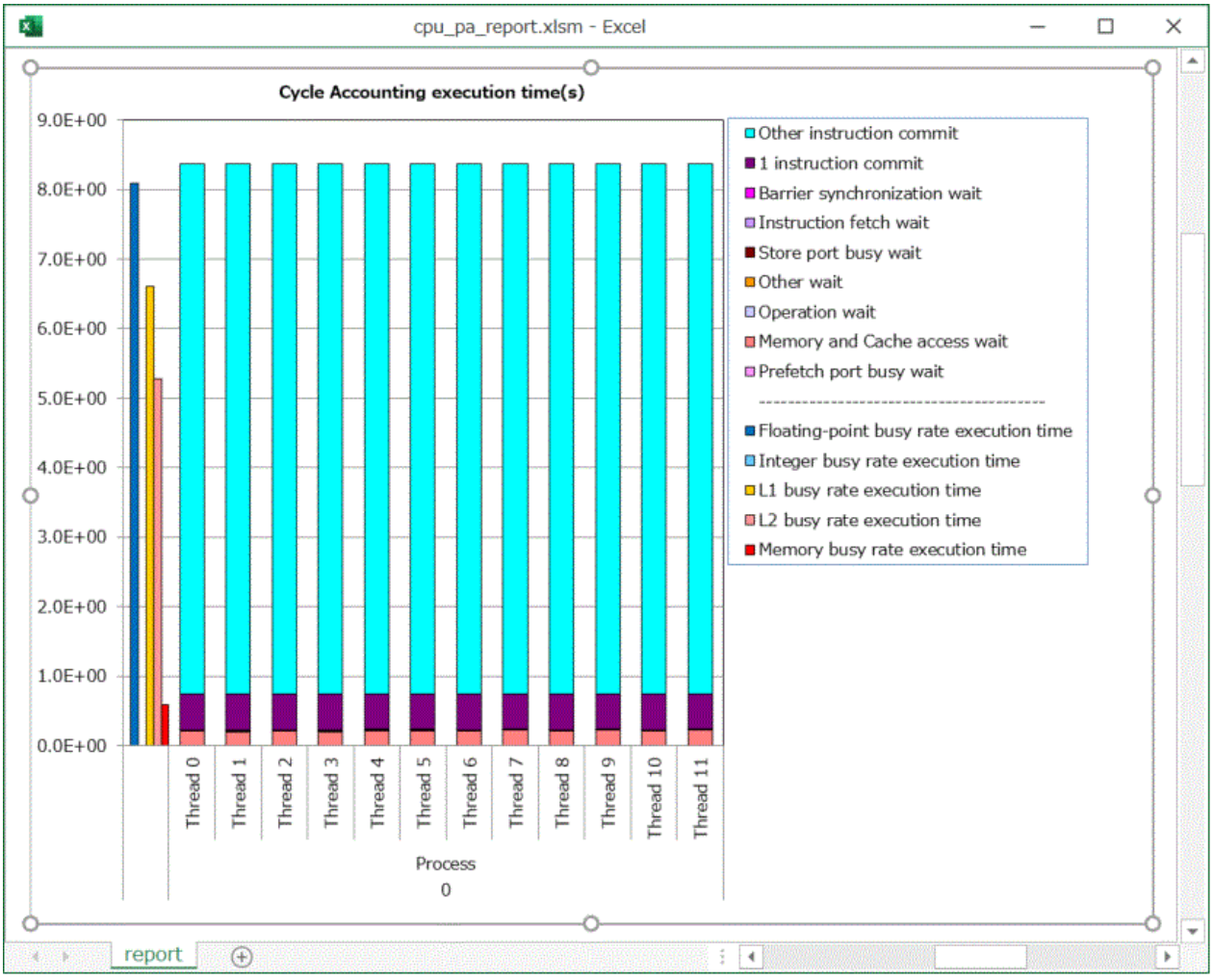


表4.9 Cycle Accounting(簡易レポート)の出力項目

項目	説明
Prefetch port busy wait	プリフェッチポートがビジーであるために完了命令数が0であった時間
Memory access wait & Cache access wait	メモリーアクセス待ち、およびキャッシュアクセス待ちのために完了命令数が0であった時間
Operation wait	実行中の命令の中で一番古い命令が、演算実行中のために完了命令数が0であった時間
Other wait	その他の要因によって完了命令数が0であった時間
Store port busy wait	ストアポートがフルであるために完了命令数が0であった時間
Instruction fetch wait	命令の読み込みを待っているために完了命令数が0であった時間
Barrier synchronization wait	WFE命令もしくはWFI命令により命令制御部が停止しているため完了命令数が0であった時間
1 instruction commit	1サイクルに1命令実行した時間
Other instruction commit	1サイクルに2～8命令実行した時間
Total	全体の時間

4.2.2.3.2 Cycle Accounting(標準レポートおよび詳細レポート)

図4.15 Cycle Accounting(標準レポートおよび詳細レポート)のレイアウト

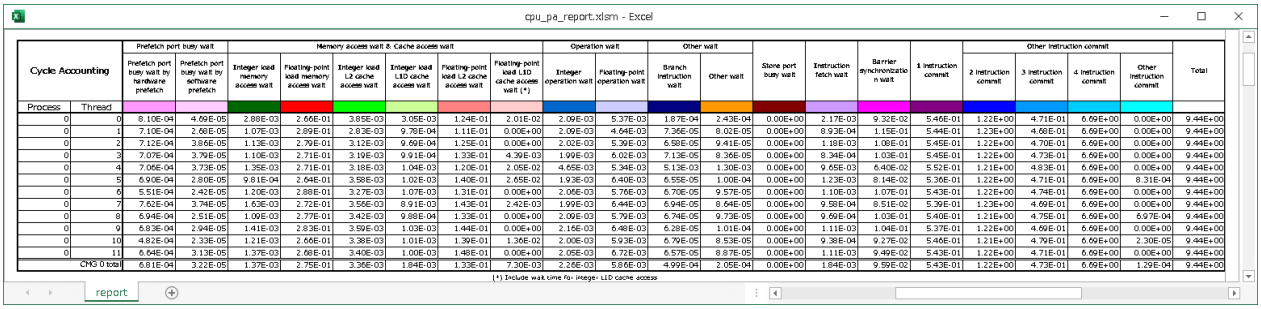


図4.16 Cycle Accounting(標準レポートおよび詳細レポート)のグラフ

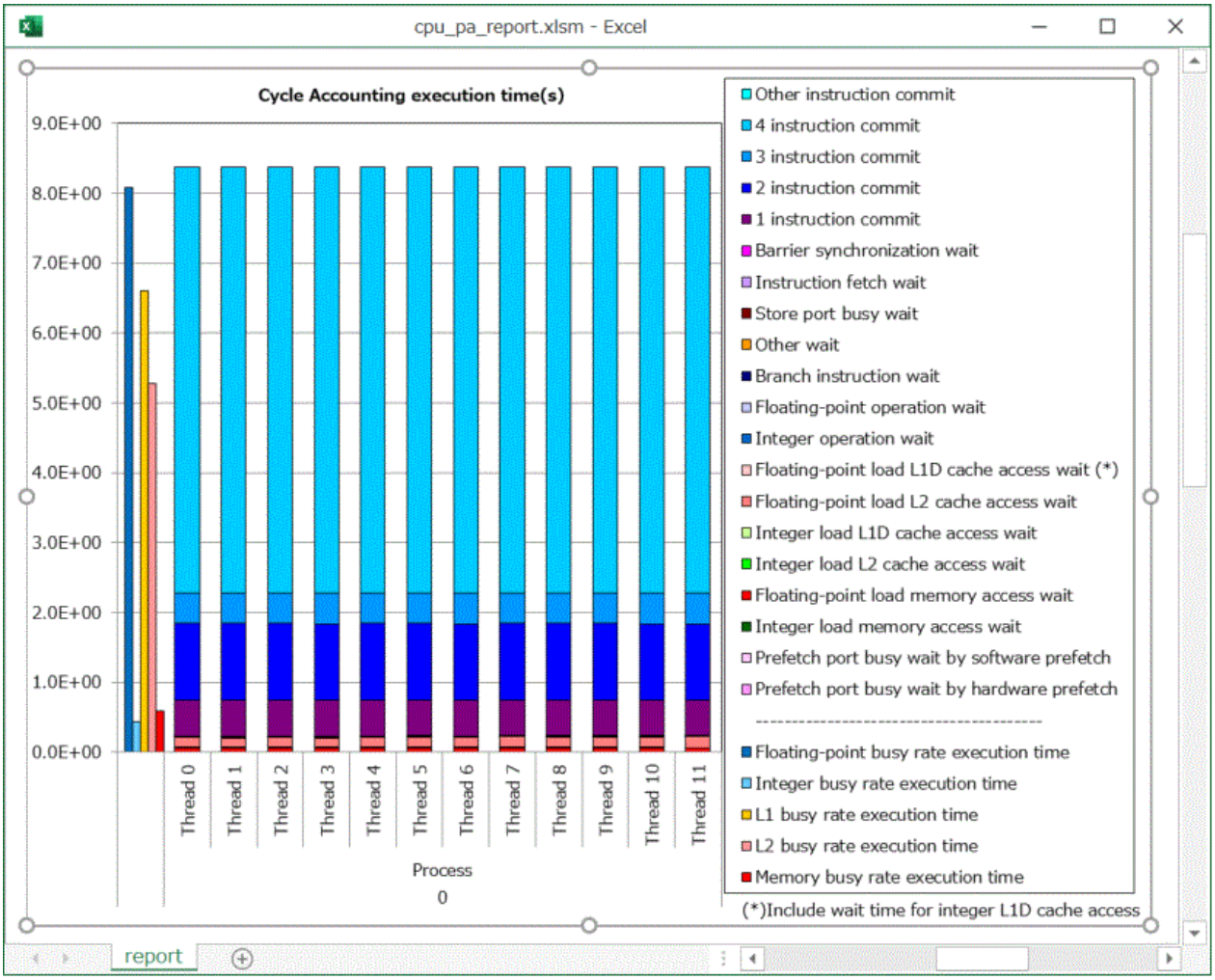


表4.10 Cycle Accounting(標準レポートおよび詳細レポート)の出力項目

項目	説明
Prefetch port busy wait by hardware prefetch	ハードウェアプリフェッチによるプリフェッチポートのビジー待ちのために完了命令数が0であった時間
Prefetch port busy wait by software prefetch	ソフトウェアプリフェッチによるプリフェッチポートのビジー待ちのために完了命令数が0であった時間
Integer load memory access wait	実行中の命令の中で一番古い命令が整数ロードのメモリーアクセスによるデータ待ちのために完了命令数が0であった時間
Floating-point load memory access wait	実行中の命令の中で一番古い命令が浮動小数点ロードのメモリーアクセスによるデータ待ちのために完了命令数が0であった時間
Integer load L2 cache access wait	実行中の命令の中で一番古い命令が整数ロードの2次キャッシュアクセスによるデータ待ちのために完了命令数が0であった時間
Integer load L1D cache access wait	実行中の命令の中で一番古い命令が整数ロードの1次データキャッシュアクセスによるデータ待ちのために完了命令数が0であった時間
Floating-point load L2 cache access wait	実行中の命令の中で一番古い命令が浮動小数点ロードの2次キャッシュアクセスによるデータ待ちのために完了命令数が0であった時間
Floating-point load L1D cache access wait	実行中の命令の中で一番古い命令が浮動小数点ロードの1次データキャッシュアクセスによるデータ待ちのために完了命令数が0であった時間

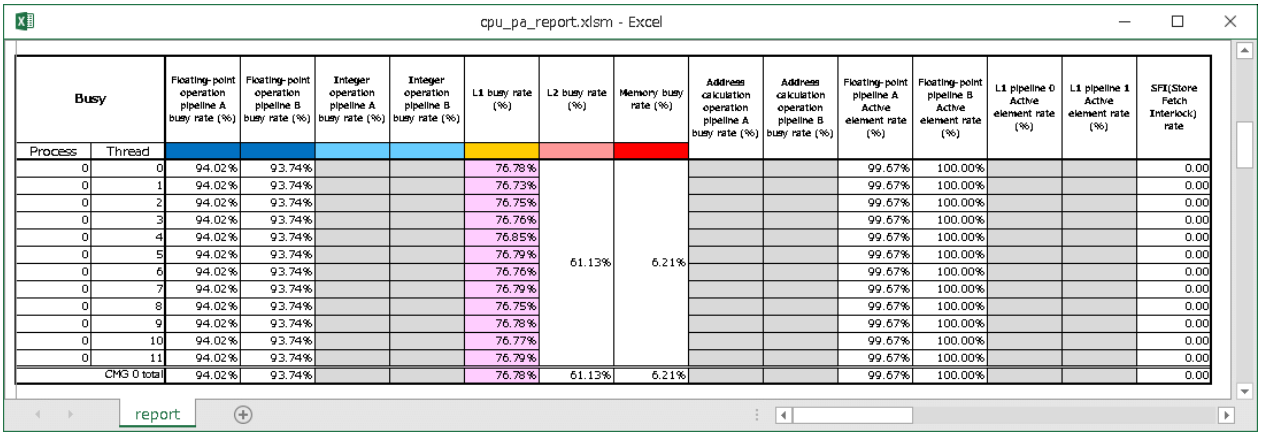
項目	説明
	注意 Floating-point load L1D cache access waitは汎用レジスタのL1Dキャッシュアクセス待ちを含む場合があります。
Integer operation wait	実行中の命令の中で一番古い命令が整数演算実行中のために完了命令数が0であった時間
Floating-point operation wait	実行中の命令の中で一番古い命令が浮動小数点演算実行中のために完了命令数が0であった時間
Branch instruction wait	実行中の命令の中で一番古い命令が分岐実行中のために完了命令数が0であった時間
Other wait	その他の要因によって完了命令数が0であった時間
Store port busy wait	ストアポートがフルであるために完了命令数が0であった時間
Instruction fetch wait	命令の読み込み待ちのために完了命令数が0であった時間
Barrier synchronization wait	WFE命令もしくはWFI命令により命令制御部が停止しているために完了命令数が0であった時間
1 instruction commit	1サイクルに1命令実行した時間
2 instruction commit	1サイクルに2命令実行した時間
3 instruction commit	1サイクルに3命令実行した時間
4 instruction commit	1サイクルに4命令実行した時間
Other instruction commit	1サイクルに5～8命令実行した時間
Total	全体の時間

4.2.2.4 Busy

Busyでは、プログラムのメモリー・キャッシュおよび演算パイプラインのビジー率に関する情報を表示します。簡易レポートの場合、1次キャッシュ、2次キャッシュ、メモリー、浮動小数点演算パイプラインなどのビジー率、およびSFI(Store Fetch Interlock)の発生率を表示します。標準レポートの場合、簡易レポートの内容に加えて整数演算パイプライン、アドレス計算用演算パイプライン、およびPredicate演算用演算パイプラインのビジー率を表示します。詳細レポートの場合、標準レポートの内容に加えてL1パイプラインにおけるActive elementの比率を表示します。セルの背景色がピンク色に変化している場合、その項目が性能ボトルネックになっている可能性があることを示します。

4.2.2.4.1 Busy(簡易レポート)

図4.17 Busy(簡易レポート)のレイアウト



Busy		Floating-point operation pipeline A busy rate (%)	Floating-point operation pipeline B busy rate (%)	Integer operation pipeline A busy rate (%)	Integer operation pipeline B busy rate (%)	L1 busy rate (%)	L2 busy rate (%)	Memory busy rate (%)	Address calculation operation pipeline A busy rate (%)	Address calculation operation pipeline B busy rate (%)	Floating-point pipeline A Active element rate (%)	Floating-point pipeline B Active element rate (%)	L1 pipeline 0 Active element rate (%)	L1 pipeline 1 Active element rate (%)	SFI(Store Fetch Interlock) rate
Process	Thread														
0	0	94.02%	93.74%			76.78%					99.67%	100.00%			0.00
0	1	94.02%	93.74%			76.73%					99.67%	100.00%			0.00
0	2	94.02%	93.74%			76.75%					99.67%	100.00%			0.00
0	3	94.02%	93.74%			76.76%					99.67%	100.00%			0.00
0	4	94.02%	93.74%			76.85%					99.67%	100.00%			0.00
0	5	94.02%	93.74%			76.79%	61.13%	6.21%			99.67%	100.00%			0.00
0	6	94.02%	93.74%			76.76%					99.67%	100.00%			0.00
0	7	94.02%	93.74%			76.79%					99.67%	100.00%			0.00
0	8	94.02%	93.74%			76.75%					99.67%	100.00%			0.00
0	9	94.02%	93.74%			76.78%					99.67%	100.00%			0.00
0	10	94.02%	93.74%			76.77%					99.67%	100.00%			0.00
0	11	94.02%	93.74%			76.79%					99.67%	100.00%			0.00
CMG 0 total		94.02%	93.74%			76.78%	61.13%	6.21%			99.67%	100.00%			0.00

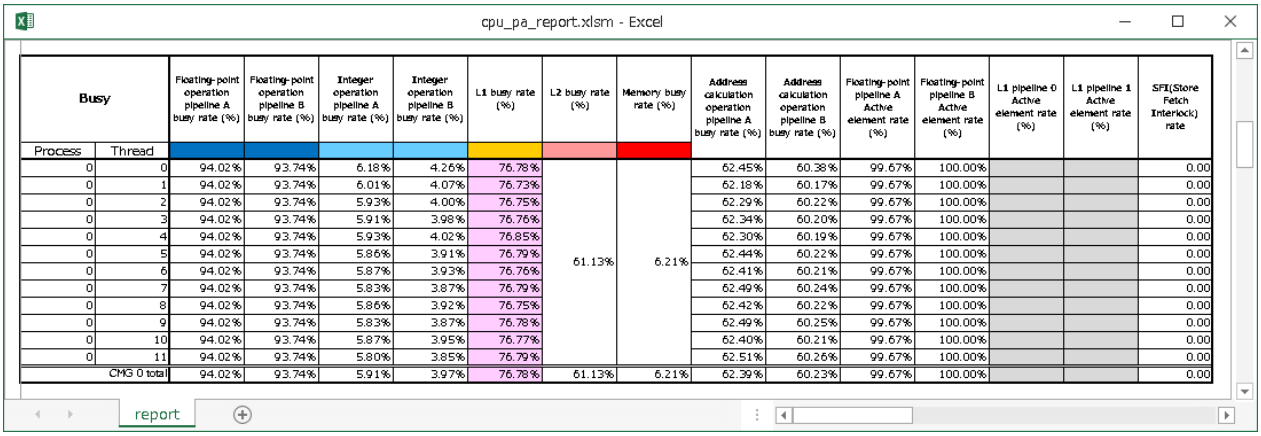
表4.11 Busy(簡易レポート)の出力項目

項目	説明
Floating-point operation pipeline A busy rate(%)	浮動小数点演算パイプラインAのビジー率(%)
Floating-point operation pipeline B busy rate(%)	浮動小数点演算パイプラインBのビジー率(%)
L1 busy rate(%)	1次キャッシュのビジー率(%)
L2 busy rate(%)	2次キャッシュのビジー率(%)
Memory busy rate(%)	メモリーのビジー率(%) **注意** ・メモリースループットの理論値は、1つのCMGを対象としています。自CMGのメモリーだけでなく他CMGのメモリーも同時にアクセスすると、メモリースループットの実測値は理論値を超える場合があります。このときには、メモリーのビジー率（メモリースループットピーク比と同じ）は、100%を超える可能性があります。 ・メモリースループットの理論値は、CMG内の全てのコアから同時にメモリーにアクセスすることを前提としています。しかし、実際には、各コアが異なるタイミングでメモリーアクセスを行うことがあります。このような場合、全コアのメモリーアクセス量を計測区間内で集計すると、異なるタイミングのメモリーアクセスを同時アクセスとして集計することになります。その結果、メモリーアクセス量と実行時間から求めたメモリースループットの実測値が理論値を超える場合があります。このときには、メモリーのビジー率（メモリースループットピーク比と同じ）は、100%を超える可能性があります。
Floating-point pipeline A Active element rate(%)	浮動小数点演算パイプラインAにおけるActive elementの比率(%) **注意** ・SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が高い場合のみ、限定的に利用することを推奨します。 ・store命令の比率が高いと、Active elementの比率の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。 ・1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。
Floating-point pipeline B Active element rate(%)	浮動小数点演算パイプラインBにおけるActive elementの比率(%) **注意** ・SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が高い場合のみ、限定的に利用することを推奨します。 ・store命令の比率が高いと、Active elementの比率の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。

項目	説明
	• 1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。
SFI(Store Fetch Interlock) rate	SFI(Store Fetch Interlock)による待ち時間の比

4.2.2.4.2 Busy(標準レポート)

図4.18 Busy(標準レポート)のレイアウト



Busy		Floating-point operation pipeline A busy rate (%)	Floating-point operation pipeline B busy rate (%)	Integer operation pipeline A busy rate (%)	Integer operation pipeline B busy rate (%)	L1 busy rate (%)	L2 busy rate (%)	Memory busy rate (%)	Address calculation operation pipeline A busy rate (%)	Address calculation operation pipeline B busy rate (%)	Floating-point pipeline A Active element rate (%)	Floating-point pipeline B Active element rate (%)	L1 pipeline 0 Active element rate (%)	L1 pipeline 1 Active element rate (%)	SFI(Store Fetch Interlock) rate
Process	Thread														
0	0	94.02%	93.74%	6.18%	4.26%	76.78%			62.45%	60.38%	99.67%	100.00%			0.00
0	1	94.02%	93.74%	6.01%	4.07%	76.73%			62.18%	60.17%	99.67%	100.00%			0.00
0	2	94.02%	93.74%	5.93%	4.00%	76.75%			62.29%	60.22%	99.67%	100.00%			0.00
0	3	94.02%	93.74%	5.91%	3.98%	76.76%			62.34%	60.20%	99.67%	100.00%			0.00
0	4	94.02%	93.74%	5.93%	4.02%	76.85%			62.30%	60.19%	99.67%	100.00%			0.00
0	5	94.02%	93.74%	5.86%	3.91%	76.79%	61.13%	6.21%	62.44%	60.22%	99.67%	100.00%			0.00
0	6	94.02%	93.74%	5.87%	3.93%	76.76%			62.41%	60.21%	99.67%	100.00%			0.00
0	7	94.02%	93.74%	5.83%	3.87%	76.79%			62.49%	60.24%	99.67%	100.00%			0.00
0	8	94.02%	93.74%	5.86%	3.92%	76.75%			62.42%	60.22%	99.67%	100.00%			0.00
0	9	94.02%	93.74%	5.83%	3.87%	76.78%			62.49%	60.25%	99.67%	100.00%			0.00
0	10	94.02%	93.74%	5.87%	3.95%	76.77%			62.40%	60.21%	99.67%	100.00%			0.00
0	11	94.02%	93.74%	5.80%	3.85%	76.79%			62.51%	60.26%	99.67%	100.00%			0.00
	CMG 0 total	94.02%	93.74%	5.91%	3.97%	76.78%	61.13%	6.21%	62.39%	60.23%	99.67%	100.00%			0.00

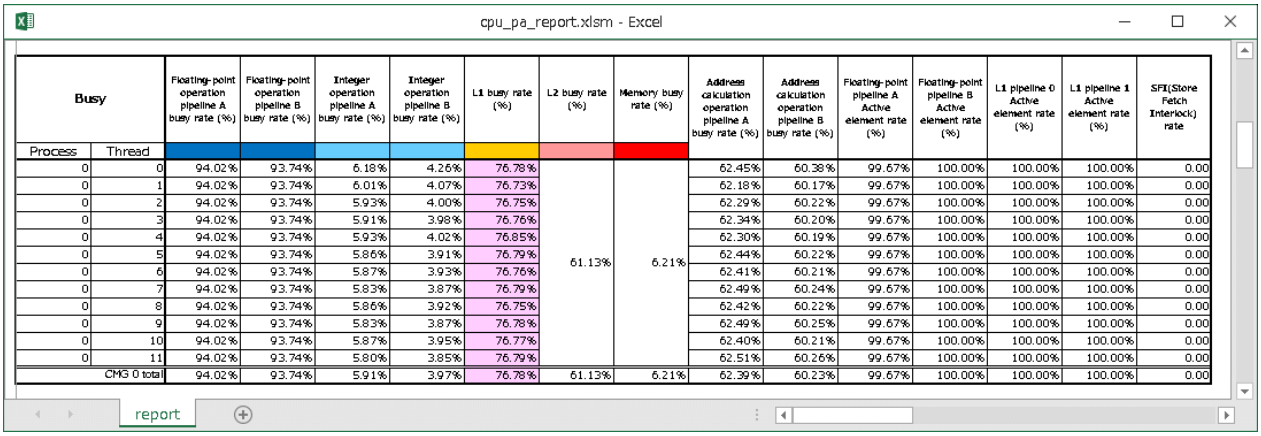
表4.12 Busy(標準レポート)の出力項目

項目	説明
Floating-point operation pipeline A busy rate(%)	浮動小数点演算パイプラインAのビジー率(%)
Floating-point operation pipeline B busy rate(%)	浮動小数点演算パイプラインBのビジー率(%)
Integer operation pipeline A busy rate(%)	整数演算パイプラインAのビジー率(%)
Integer operation pipeline B busy rate(%)	整数演算パイプラインBのビジー率(%)
L1 busy rate(%)	1次キャッシュのビジー率(%)
L2 busy rate(%)	2次キャッシュのビジー率(%)
Memory busy rate(%)	メモリーのビジー率(%) **注意** • メモリースループットの理論値は、1つのCMGを対象としています。自CMGのメモリーだけでなく他CMGのメモリーも同時にアクセスすると、メモリースループットの実測値は理論値を超える場合があります。このときには、メモリーのビジー率(メモリースループットピーク比と同じ)は、100%を超える可能性があります。 • メモリースループットの理論値は、CMG内の全てのコアから同時にメモリーにアクセスすることを前提としています。しかし、実際には、各コアが異なるタイミングでメモリーアクセスを行うことがあります。このような場合、全コアのメモリーアクセス量を計測区間内で集計すると、異なるタイミングのメモリーアクセスを同時アクセスとして集計することになります。その結果、メモリーアクセス量と実行時間から求めたメモリースルー

項目	説明
	プットの実測値が理論値を超える場合があります。このときには、メモリーのビジー率（メモリースループットピーク比と同じ）は、100%を超える可能性があります。
Address calculation operation pipeline A busy rate(%)	アドレス計算用演算パイプラインAのビジー率(%)
Address calculation operation pipeline B busy rate(%)	アドレス計算用演算パイプラインBのビジー率(%)
Floating-point pipeline A Active element rate(%)	浮動小数点演算パイプラインAにおけるActive elementの比率(%) **注意** ・SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が高い場合のみ、限定的に利用することを推奨します。 ・store命令の比率が高いと、Active elementの比率の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。 ・1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。
Floating-point pipeline B Active element rate(%)	浮動小数点演算パイプラインBにおけるActive elementの比率(%) **注意** ・SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が高い場合のみ、限定的に利用することを推奨します。 ・store命令の比率が高いと、Active elementの比率の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。 ・1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。
SFI(Store Fetch Interlock) rate	SFI(Store Fetch Interlock)による待ち時間の比

4.2.2.4.3 Busy(詳細レポート)

図4.19 Busy(詳細レポート)のレイアウト



Busy		Floating-point operation pipeline A busy rate (%)	Floating-point operation pipeline B busy rate (%)	Integer operation pipeline A busy rate (%)	Integer operation pipeline B busy rate (%)	L1 busy rate (%)	L2 busy rate (%)	Memory busy rate (%)	Address calculation operation pipeline A busy rate (%)	Address calculation operation pipeline B busy rate (%)	Floating-point pipeline A Active element rate (%)	Floating-point pipeline B Active element rate (%)	L1 pipeline 0 Active element rate (%)	L1 pipeline 1 Active element rate (%)	SFI(Store Fetch Interlock) rate
Process	Thread														
0	0	94.02%	93.74%	6.18%	4.26%	76.78%	61.13%	6.21%	62.45%	60.38%	99.67%	100.00%	100.00%	100.00%	0.00
0	1	94.02%	93.74%	6.01%	4.07%	76.73%			62.18%	60.17%	99.67%	100.00%	100.00%	100.00%	0.00
0	2	94.02%	93.74%	5.93%	4.00%	76.75%			62.29%	60.22%	99.67%	100.00%	100.00%	100.00%	0.00
0	3	94.02%	93.74%	5.91%	3.98%	76.76%			62.34%	60.20%	99.67%	100.00%	100.00%	100.00%	0.00
0	4	94.02%	93.74%	5.93%	4.02%	76.85%			62.30%	60.19%	99.67%	100.00%	100.00%	100.00%	0.00
0	5	94.02%	93.74%	5.86%	3.91%	76.79%			62.44%	60.22%	99.67%	100.00%	100.00%	100.00%	0.00
0	6	94.02%	93.74%	5.87%	3.93%	76.76%			62.41%	60.21%	99.67%	100.00%	100.00%	100.00%	0.00
0	7	94.02%	93.74%	5.83%	3.87%	76.79%			62.49%	60.24%	99.67%	100.00%	100.00%	100.00%	0.00
0	8	94.02%	93.74%	5.86%	3.92%	76.75%			62.42%	60.22%	99.67%	100.00%	100.00%	100.00%	0.00
0	9	94.02%	93.74%	5.83%	3.87%	76.78%			62.49%	60.25%	99.67%	100.00%	100.00%	100.00%	0.00
0	10	94.02%	93.74%	5.87%	3.95%	76.77%			62.40%	60.21%	99.67%	100.00%	100.00%	100.00%	0.00
0	11	94.02%	93.74%	5.80%	3.85%	76.79%			62.51%	60.26%	99.67%	100.00%	100.00%	100.00%	0.00
	CMG 0 total	94.02%	93.74%	5.91%	3.97%	76.78%	61.13%	6.21%	62.39%	60.23%	99.67%	100.00%	100.00%	100.00%	0.00

表4.13 Busy(詳細レポート)の出力項目

項目	説明
Floating-point operation pipeline A busy rate(%)	浮動小数点演算パイプラインAのビジー率(%)
Floating-point operation pipeline B busy rate(%)	浮動小数点演算パイプラインBのビジー率(%)
Integer operation pipeline A busy rate(%)	整数演算パイプラインAのビジー率(%)
Integer operation pipeline B busy rate(%)	整数演算パイプラインBのビジー率(%)
L1 busy rate(%)	1次キャッシュのビジー率(%)
L2 busy rate(%)	2次キャッシュのビジー率(%)
Memory busy rate(%)	メモリーのビジー率(%) **注意** ・メモリースループットの理論値は、1つのCMGを対象としています。自CMGのメモリーだけでなく他CMGのメモリーも同時にアクセスすると、メモリースループットの実測値は理論値を超える場合があります。このときには、メモリーのビジー率(メモリースループットピーク比と同じ)は、100%を超える可能性があります。 ・メモリースループットの理論値は、CMG内の全てのコアから同時にメモリーにアクセスすることを前提としています。しかし、実際には、各コアが異なるタイミングでメモリーアクセスを行うことがあります。このような場合、全コアのメモリーアクセス量を計測区間内で集計すると、異なるタイミングのメモリーアクセスを同時アクセスとして集計することになります。その結果、メモリーアクセス量と実行時間から求めたメモリースループットの実測値が理論値を超える場合があります。このときには、メモリーのビジー率(メモリースループットピーク比と同じ)は、100%を超える可能性があります。
Address calculation operation pipeline A busy rate(%)	アドレス計算用演算パイプラインAのビジー率(%)

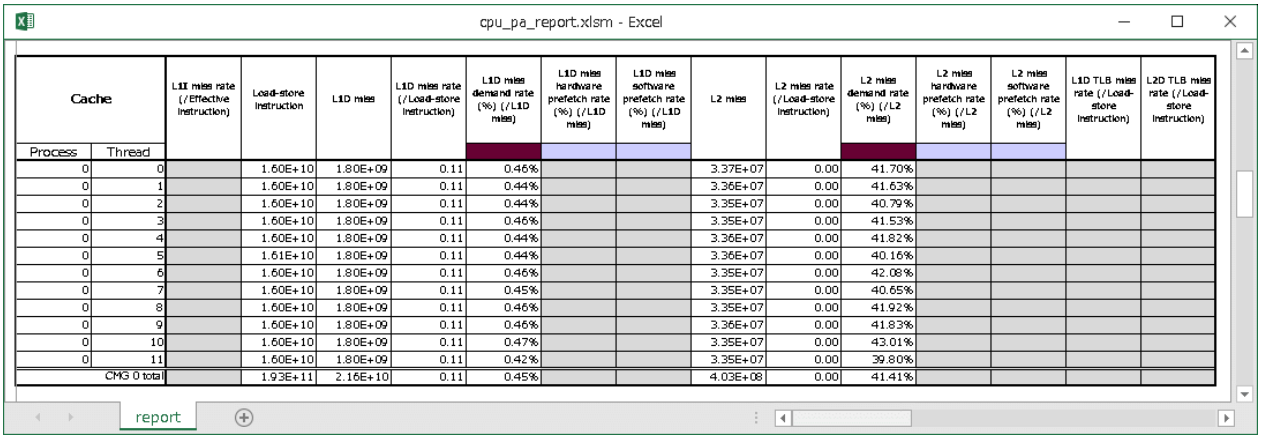
項目	説明
Address calculation operation pipeline B busy rate(%)	アドレス計算用演算パイプラインBのビジー率(%)
Floating-point pipeline A Active element rate(%)	浮動小数点演算パイプラインAにおけるActive elementの比率(%) **注意** ・SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が高い場合のみ、限定的に利用することを推奨します。 ・store命令の比率が高いと、Active elementの比率の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。 ・1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。
Floating-point pipeline B Active element rate(%)	浮動小数点演算パイプラインBにおけるActive elementの比率(%) **注意** ・SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が高い場合のみ、限定的に利用することを推奨します。 ・store命令の比率が高いと、Active elementの比率の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。 ・1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。
L1 pipeline 0 Active element rate(%)	L1パイプライン0におけるActive elementの比率(%)
L1 pipeline 1 Active element rate(%)	L1パイプライン1におけるActive elementの比率(%)
SFI(Store Fetch Interlock) rate	SFI(Store Fetch Interlock)による待ち時間の比

4.2.2.5 Cache

Cacheでは、キャッシュミスに関する情報を表示します。簡易レポートの場合、1次データキャッシュおよび2次キャッシュのキャッシュミス数、およびロード・ストア命令数との比率を表示します。標準レポートまたは詳細レポートの場合、簡易レポートの内容に加えて複数の命令数との比較を追加します。セルの背景色がピンク色に変化している場合、その項目が性能ボトルネックになっている可能性があることを示します。Cacheには対応する表とグラフがあります。“表4.14 Cache(簡易レポート)の出力項目”または“表4.15 Cache(標準レポートおよび詳細レポート)の出力項目”で青字記載されている項目を積み上げ棒グラフで表示します。

4.2.2.5.1 Cache(簡易レポート)

図4.20 Cache(簡易レポート)のレイアウト



Cache		L1I miss rate (/Effective Instruction)	Load-store Instruction	L1D miss	L1D miss rate (/Load-store Instruction)	L1D miss demand rate (%) (/L1D miss)	L1D miss hardware prefetch rate (%) (/L1D miss)	L1D miss software prefetch rate (%) (/L1D miss)	L2 miss	L2 miss rate (/Load-store Instruction)	L2 miss demand rate (%) (/L2 miss)	L2 miss hardware prefetch rate (%) (/L2 miss)	L2 miss software prefetch rate (%) (/L2 miss)	L1D TLB miss rate (/Load-store Instruction)	L2D TLB miss rate (/Load-store Instruction)
Process	Thread														
0	0		1.60E+10	1.80E+09	0.11	0.46%			3.37E+07	0.00	41.70%				
0	1		1.60E+10	1.80E+09	0.11	0.44%			3.36E+07	0.00	41.63%				
0	2		1.60E+10	1.80E+09	0.11	0.44%			3.35E+07	0.00	40.79%				
0	3		1.60E+10	1.80E+09	0.11	0.46%			3.35E+07	0.00	41.53%				
0	4		1.60E+10	1.80E+09	0.11	0.44%			3.36E+07	0.00	41.82%				
0	5		1.61E+10	1.80E+09	0.11	0.44%			3.36E+07	0.00	40.16%				
0	6		1.60E+10	1.80E+09	0.11	0.46%			3.35E+07	0.00	42.08%				
0	7		1.60E+10	1.80E+09	0.11	0.45%			3.35E+07	0.00	40.65%				
0	8		1.60E+10	1.80E+09	0.11	0.46%			3.35E+07	0.00	41.92%				
0	9		1.60E+10	1.80E+09	0.11	0.46%			3.36E+07	0.00	41.83%				
0	10		1.60E+10	1.80E+09	0.11	0.47%			3.35E+07	0.00	43.01%				
0	11		1.60E+10	1.80E+09	0.11	0.42%			3.35E+07	0.00	39.80%				
CMG 0 total			1.93E+11	2.16E+10	0.11	0.45%			4.03E+08	0.00	41.41%				

図4.21 Cache(簡易レポート)のグラフ

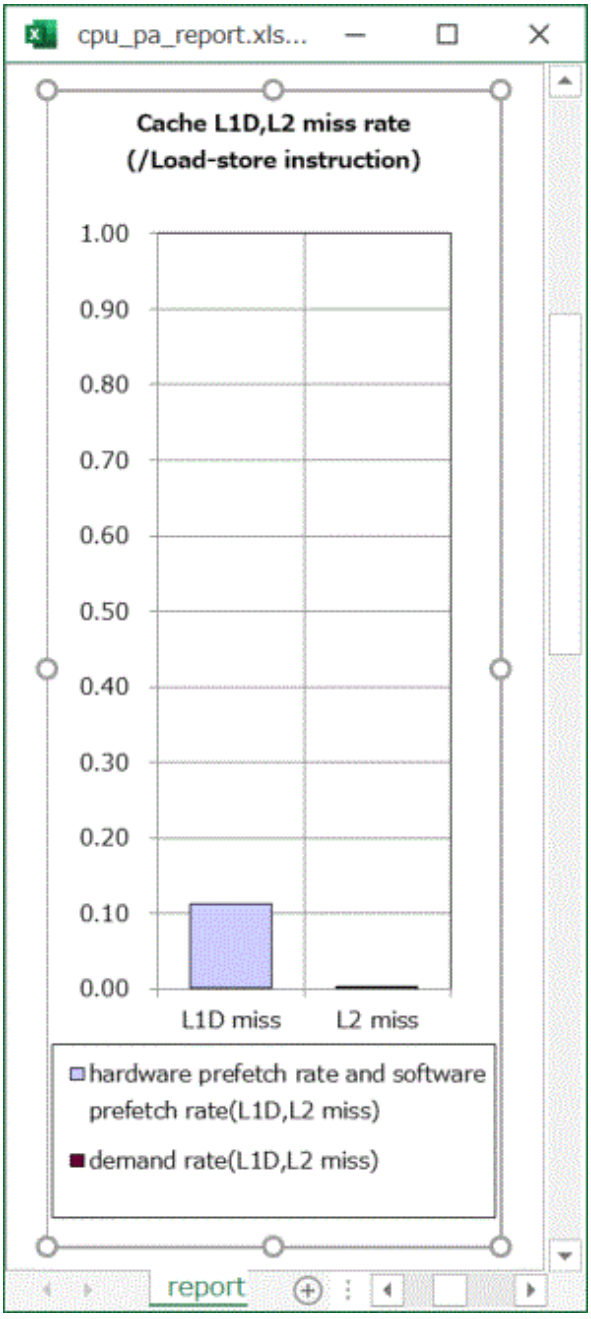


表4.14 Cache(簡易レポート)の出力項目

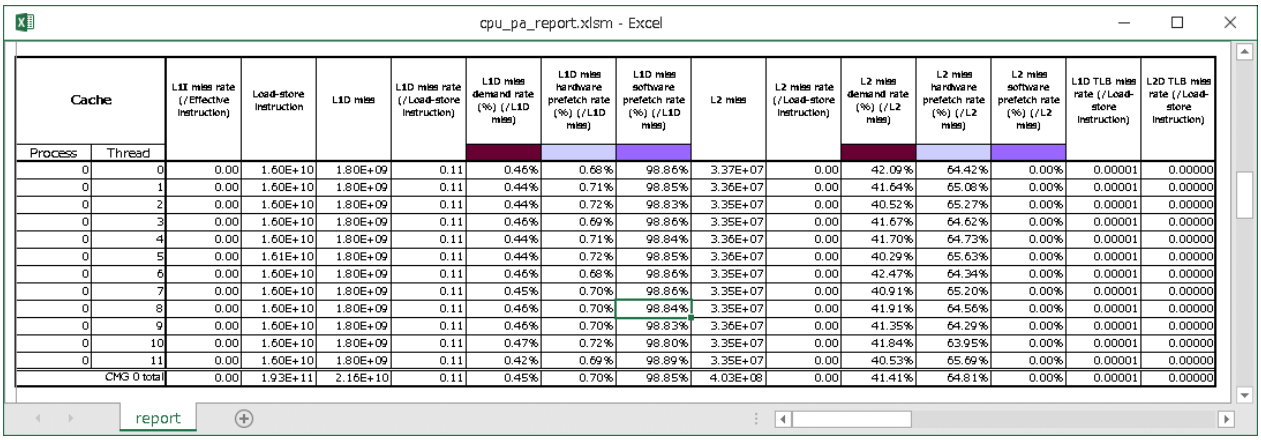
項目	説明
Load-store instruction	ロード・ストア命令の数
L1D miss	1次データキャッシュミスの数
L1D miss rate (/Load-store instruction)	ロード・ストア命令数に対する、1次データキャッシュミスの比
L1D miss demand rate(%) (/L1D miss)	1次データキャッシュミスのうち、デマンドアクセスによる1次データキャッシュミスの比率(%)
L2 miss	2次キャッシュミスの数
L2 miss rate (/Load-store instruction)	ロード・ストア命令数に対する、2次キャッシュミスの比
L2 miss demand rate(%) (/L2 miss)	2次キャッシュミスのうち、デマンドアクセスによる2次キャッシュミスの比率(%)

ポイント

積み上げ棒グラフの合計値は“L1D miss rate(%)(/Load-store instruction)”または“L2 miss rate(%)(/Load-store instruction)”と一致します。

4.2.2.5.2 Cache(標準レポートおよび詳細レポート)

図4.22 Cache(標準レポートおよび詳細レポート)のレイアウト



Cache		L1I miss rate (/Effective instruction)	Load-store instruction	L1D miss	L1D miss rate (/Load-store instruction)	L1D miss demand rate (%) (/L1D miss)	L1D miss hardware prefetch rate (%) (/L1D miss)	L1D miss software prefetch rate (%) (/L1D miss)	L2 miss	L2 miss rate (/Load-store instruction)	L2 miss demand rate (%) (/L2 miss)	L2 miss hardware prefetch rate (%) (/L2 miss)	L2 miss software prefetch rate (%) (/L2 miss)	L1D TLB miss rate (/Load-store instruction)	L2D TLB miss rate (/Load-store instruction)
Process	Thread														
0	0	0.00	1.60E+10	1.80E+09	0.11	0.46%	0.68%	98.86%	3.37E+07	0.00	42.09%	64.42%	0.00%	0.00001	0.00000
0	1	0.00	1.60E+10	1.80E+09	0.11	0.44%	0.71%	98.85%	3.36E+07	0.00	41.64%	65.08%	0.00%	0.00001	0.00000
0	2	0.00	1.60E+10	1.80E+09	0.11	0.44%	0.72%	98.83%	3.35E+07	0.00	40.52%	65.27%	0.00%	0.00001	0.00000
0	3	0.00	1.60E+10	1.80E+09	0.11	0.46%	0.69%	98.86%	3.35E+07	0.00	41.67%	64.62%	0.00%	0.00001	0.00000
0	4	0.00	1.60E+10	1.80E+09	0.11	0.44%	0.71%	98.84%	3.36E+07	0.00	41.70%	64.73%	0.00%	0.00001	0.00000
0	5	0.00	1.61E+10	1.80E+09	0.11	0.44%	0.72%	98.85%	3.36E+07	0.00	40.29%	65.63%	0.00%	0.00001	0.00000
0	6	0.00	1.60E+10	1.80E+09	0.11	0.46%	0.68%	98.86%	3.35E+07	0.00	42.47%	64.34%	0.00%	0.00001	0.00000
0	7	0.00	1.60E+10	1.80E+09	0.11	0.45%	0.70%	98.86%	3.35E+07	0.00	40.91%	65.20%	0.00%	0.00001	0.00000
0	8	0.00	1.60E+10	1.80E+09	0.11	0.46%	0.70%	98.84%	3.35E+07	0.00	41.91%	64.56%	0.00%	0.00001	0.00000
0	9	0.00	1.60E+10	1.80E+09	0.11	0.46%	0.70%	98.83%	3.36E+07	0.00	41.35%	64.29%	0.00%	0.00001	0.00000
0	10	0.00	1.60E+10	1.80E+09	0.11	0.47%	0.72%	98.80%	3.35E+07	0.00	41.84%	63.95%	0.00%	0.00001	0.00000
0	11	0.00	1.60E+10	1.80E+09	0.11	0.42%	0.69%	98.89%	3.35E+07	0.00	40.53%	65.69%	0.00%	0.00001	0.00000
CMG 0 total		0.00	1.93E+11	2.16E+10	0.11	0.45%	0.70%	98.85%	4.03E+08	0.00	41.41%	64.81%	0.00%	0.00001	0.00000

図4.23 Cache(標準レポートおよび詳細レポート)のグラフ

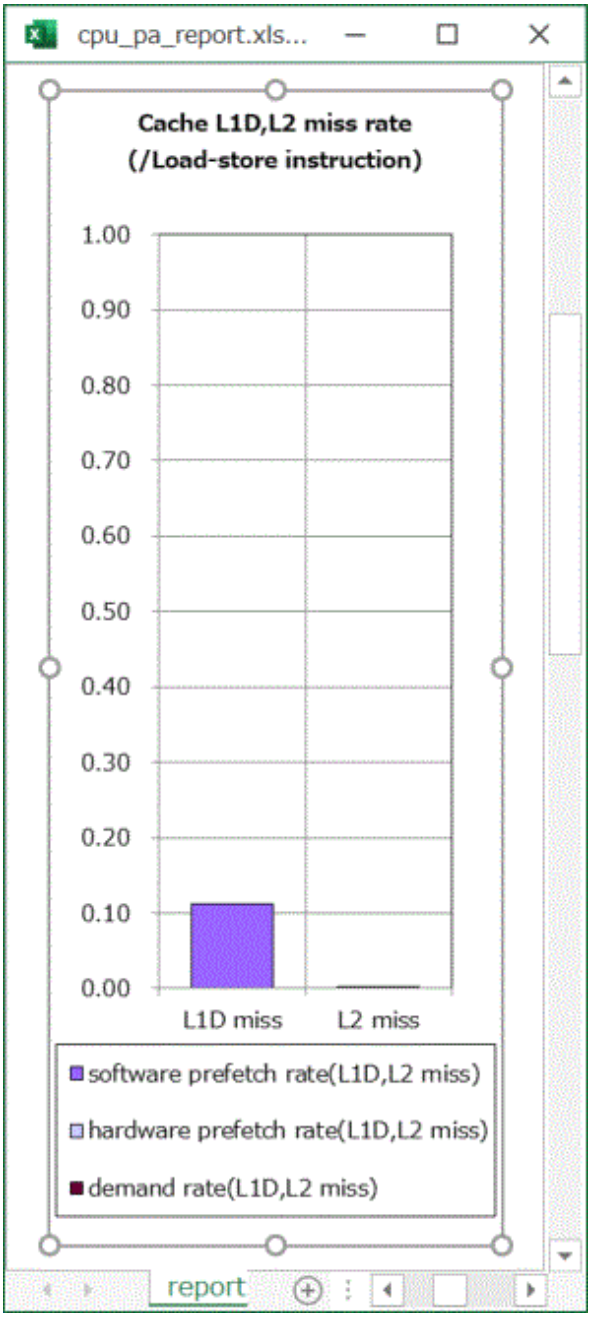


表4.15 Cache(標準レポートおよび詳細レポート)の出力項目

項目	説明
L1I miss rate (/Effective instruction)	実行された命令の総数に対する1次命令キャッシュミスの比
Load-store instruction	ロード・ストア命令の数
L1D miss	1次データキャッシュミスの数
L1D miss rate (/Load-store instruction)	ロード・ストア命令数に対する、1次データキャッシュミスの比
L1D miss demand rate(%) (/L1D miss)	1次データキャッシュミスのうち、デマンドアクセスによる1次データキャッシュミスの比率(%)
L1D miss hardware prefetch rate(%) (/L1D miss)	1次データキャッシュミスのうち、ハードウェアプリフェッチによる1次データキャッシュミスの比率(%)

項目	説明
L1D miss software prefetch rate(%) (/L1D miss)	1次データキャッシュミスのうち、ソフトウェアプリフェッチによる1次データキャッシュミスの比率(%)
L2 miss	2次キャッシュミスの数
L2 miss rate (/Load-store instruction)	ロード・ストア命令数に対する、2次キャッシュミスの比
L2 miss demand rate(%) (/L2 miss)	2次キャッシュミスのうち、デマンドアクセスによる2次キャッシュミスの比率(%)
L2 miss hardware prefetch rate(%) (/L2 miss)	2次キャッシュミスのうち、ハードウェアプリフェッチによる2次キャッシュミスの比率(%)
L2 miss software prefetch rate(%) (/L2 miss)	2次キャッシュミスのうち、ソフトウェアプリフェッチによる2次キャッシュミスの比率(%)
L1D TLB miss rate (/Load-store instruction)	ロード・ストア命令数に対する、L1D TLBミスの比
L2D TLB miss rate (/Load-store instruction)	ロード・ストア命令数に対する、L2D TLBミスの比

ポイント

積み上げ棒グラフの合計値は“L1D miss rate(%)(/Load-store instruction)”または“L2 miss rate(%)(/Load-store instruction)”と一致します。

キャッシュミスの比率については、計測時の誤差やブレの影響により、値が範囲を超える可能性があります。キャッシュミスの比率が100%を超えた場合は100%に、0%を下回った場合には0%とみなしてください。

4.2.2.6 Instruction

Instructionでは、命令ミックスに関する情報を表示します。簡易レポートでは10種類、標準レポートでは25種類、詳細レポートでは28種類に分類して表示します。

4.2.2.6.1 Instruction(簡易レポート)

図4.24 Instruction(簡易レポート)のレイアウト

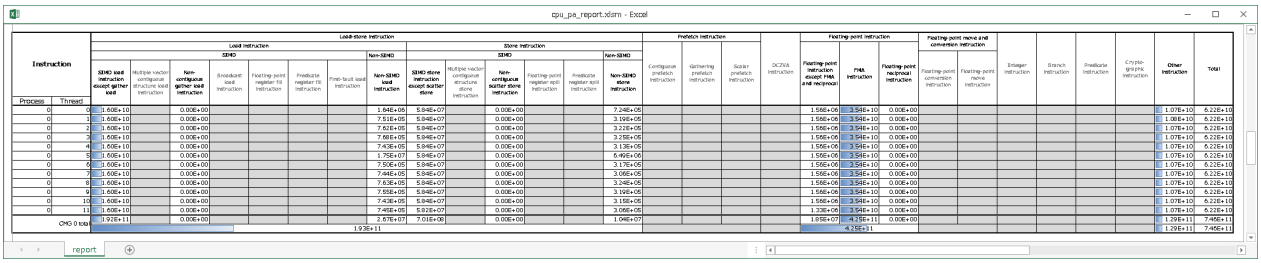


表4.16 Instruction(簡易レポート)の出力項目

項目	説明
SIMD load instruction except gather load	ギャザーロード命令を除くSIMDロード命令の数
Non-contiguous gather load instruction	非連続ギャザーロード命令の数
Non-SIMD load instruction	非SIMDロード命令の数
SIMD store instruction except scatter store	スキャッターストア命令を除くSIMDストア命令の数
Non-contiguous scatter store instruction	非連続スキャッターストア命令の数

項目	説明
Non-SIMD store instruction	非SIMDストア命令の数
Floating-point instruction except FMA and reciprocal	浮動小数点積和演算命令および浮動小数点逆数演算命令を除く浮動小数点演算命令の数
FMA instruction	浮動小数点積和演算命令の数
Floating-point reciprocal instruction	浮動小数点逆数演算命令の数
Other instruction	その他の命令の数
Total	全命令数 **注意** 全命令数は、MOVPRFX命令を含みません。

4.2.2.6.2 Instruction(標準レポート)

図4.25 Instruction(標準レポート)のレイアウト

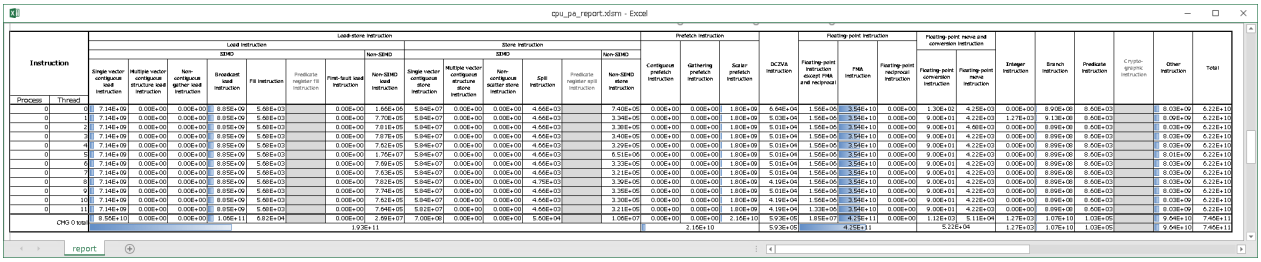


表4.17 Instruction(標準レポート)の出力項目

項目	説明
Single vector contiguous load instruction	SIMD連続ロード命令の数
Multiple vector contiguous structure load instruction	SIMD連続ストラクチャーロード命令の数
Non-contiguous gather load instruction	非連続ギャザーロード命令の数
Broadcast load instruction	ブロードキャストロード命令の数
Fill instruction	fill命令の数
First-fault load instruction	First-faultロード命令の数
Non-SIMD load instruction	非SIMDロード命令の数
Single vector contiguous store instruction	SIMD連続ストア命令の数
Multiple vector contiguous structure store instruction	SIMD連続ストラクチャーストア命令の数
Non-contiguous scatter store instruction	非連続スキャッターストア命令の数
Spill instruction	spill命令の数
Non-SIMD store instruction	非SIMDストア命令の数
Contiguous prefetch instruction	連続プリフェッチ命令の数

項目	説明
Gathering prefetch instruction	ギャザープリフェッチ命令の数
Scalar prefetch instruction	通常プリフェッチ命令の数
DCZVA instruction	DCZVA命令の数
Floating-point instruction except FMA and reciprocal	浮動小数点積和演算命令および浮動小数点逆数演算命令を除く浮動小数点演算命令の数
FMA instruction	浮動小数点積和演算命令の数
Floating-point reciprocal instruction	浮動小数点逆数演算命令の数
Floating-point conversion instruction	浮動小数点精度変換命令の数
Floating-point move instruction	浮動小数点移動命令の数
Integer instruction	整数演算命令の数
Branch instruction	分岐命令の数
Predicate instruction	Predicate演算命令の数
Other instruction	その他の命令の数
Total	全命令数 **注意** 全命令数は、MOVPRFX命令を含みません。

4.2.2.6.3 Instruction(詳細レポート)

図4.26 Instruction(詳細レポート)のレイアウト

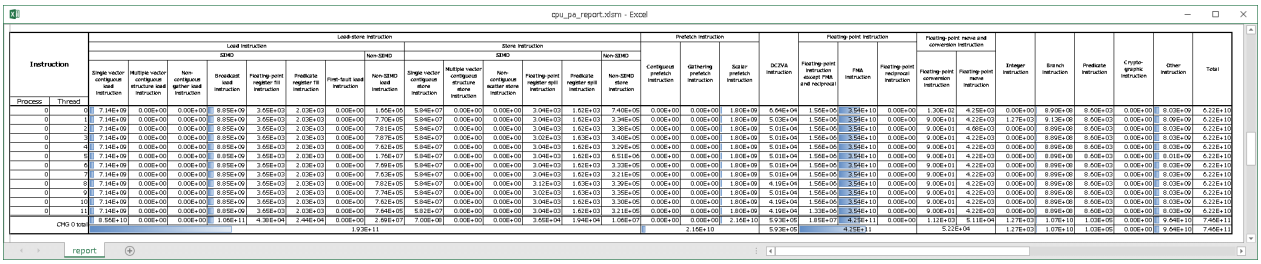


表4.18 Instruction(詳細レポート)の出力項目

項目	説明
Single vector contiguous load instruction	SIMD連続ロード命令の数
Multiple vector contiguous structure load instruction	SIMD連続ストラクチャーロード命令の数
Non-contiguous gather load instruction	非連続ギャザーロード命令の数
Broadcast load instruction	ブロードキャストロード命令の数
Floating-point register fill instruction	浮動小数点レジスターへのfill命令数
Predicate register fill instruction	Predicateレジスターへのfill命令数
First-fault load instruction	First-faultロード命令の数
Non-SIMD load instruction	非SIMDロード命令の数

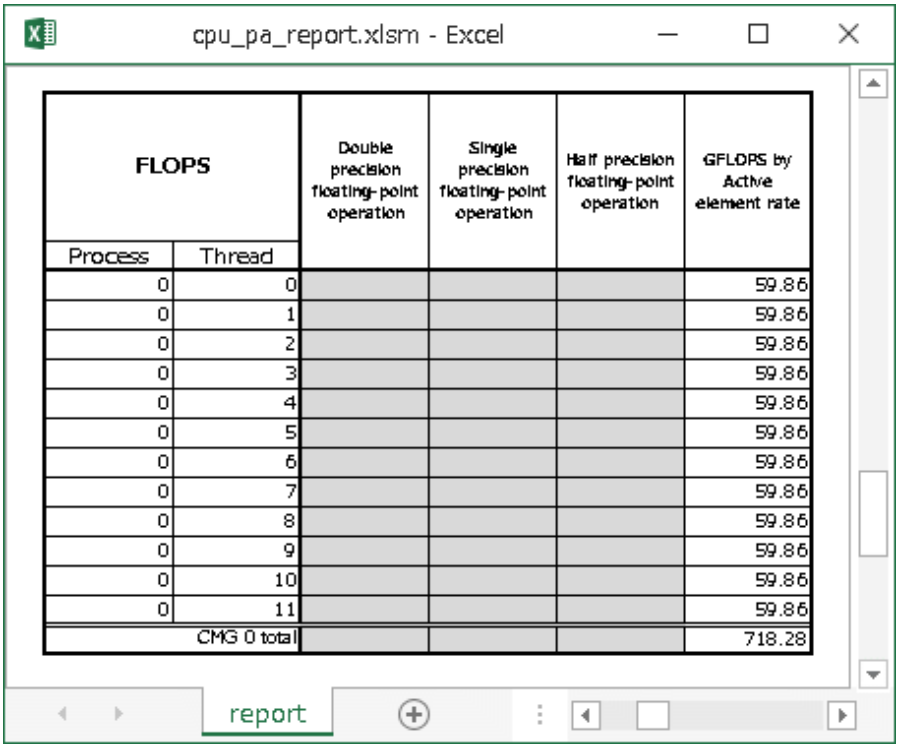
項目	説明
Single vector contiguous store instruction	SIMD連続ストア命令の数
Multiple vector contiguous structure store instruction	SIMD連続ストラクチャーストア命令の数
Non-contiguous scatter store instruction	非連続スキャッターストア命令の数
Floating-point register spill instruction	浮動小数点レジスターからのspill命令数
Predicate register spill instruction	Predicateレジスターからのspill命令数
Non-SIMD store instruction	非SIMDストア命令の数
Contiguous prefetch instruction	連続プリフェッチ命令の数
Gathering prefetch instruction	ギャザープリフェッチ命令の数
Scalar prefetch instruction	通常プリフェッチ命令の数
DCZVA instruction	DCZVA命令の数
Floating-point instruction except FMA and reciprocal	浮動小数点積和演算命令および浮動小数点逆数演算命令を除く浮動小数点演算命令の数
FMA instruction	浮動小数点積和演算命令の数
Floating-point reciprocal instruction	浮動小数点逆数演算命令の数
Floating-point conversion instruction	浮動小数点精度変換命令の数
Floating-point move instruction	浮動小数点移動命令の数
Integer instruction	整数演算命令の数
Branch instruction	分岐命令の数
Predicate instruction	Predicate演算命令の数
Crypto-graphic instruction	暗号命令の数
Other instruction	その他の命令の数
Total	全命令数 **注意** 全命令数は、MOVPRFX命令を含みません。

4.2.2.7 FLOPS

簡易レポートの場合、Active elementの比率を考慮した浮動小数点演算性能を表示します。標準レポートまたは詳細レポートの場合、簡易レポートの内容に加えて各精度浮動小数点演算数を表示します。

4.2.2.7.1 FLOPS(簡易レポート)

図4.27 FLOPSのレイアウト(簡易レポート)



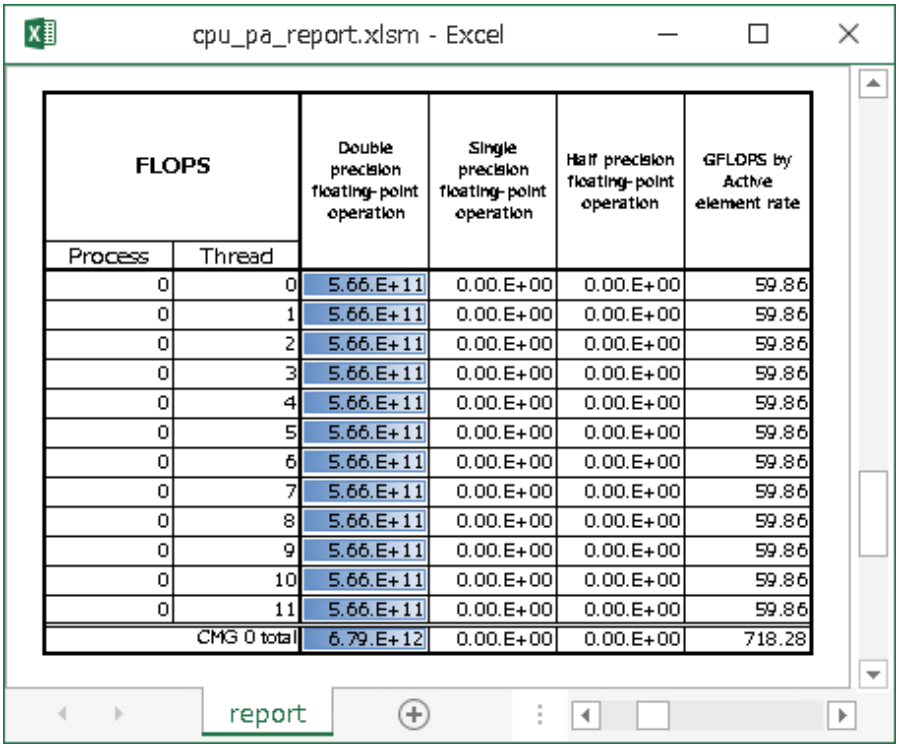
FLOPS		Double precision floating-point operation	Single precision floating-point operation	Half precision floating-point operation	GFLOPS by Active element rate
Process	Thread				
0	0				59.86
0	1				59.86
0	2				59.86
0	3				59.86
0	4				59.86
0	5				59.86
0	6				59.86
0	7				59.86
0	8				59.86
0	9				59.86
0	10				59.86
0	11				59.86
CMG 0 total					718.28

表4.19 FLOPSの出力項目(簡易レポート)

項目	説明
GFLOPS by Active element rate	Active elementの比率を考慮した浮動小数点演算性能 **注意** ・通常のGFLOPS値はすべてActive elementとして算出しています。そのため、Inactive elementの多いプログラムでは本来のGFLOPS値より高い値を出力します。Predicateを用いた演算を多用するプログラムでは、Active elementの比率を考慮した、1秒あたりに実行された浮動小数点演算数を使用することでより精度の高い値を算出することが可能になります。ただし、store命令の比率が高いと、Active elementの比率を考慮した、1秒あたりに実行された浮動小数点演算数の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。 ・SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が高い場合のみ、限定的に利用することを推奨します。 ・1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。

4.2.2.7.2 FLOPS(標準レポートおよび詳細レポート)

図4.28 FLOPSのレイアウト(標準レポートおよび詳細レポート)



FLOPS		Double precision floating-point operation	Single precision floating-point operation	Half precision floating-point operation	GFLOPS by Active element rate
Process	Thread				
0	0	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	1	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	2	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	3	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	4	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	5	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	6	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	7	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	8	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	9	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	10	5.66.E+11	0.00.E+00	0.00.E+00	59.86
0	11	5.66.E+11	0.00.E+00	0.00.E+00	59.86
CMG 0 total		6.79.E+12	0.00.E+00	0.00.E+00	718.28

表4.20 FLOPSの出力項目(標準レポートおよび詳細レポート)

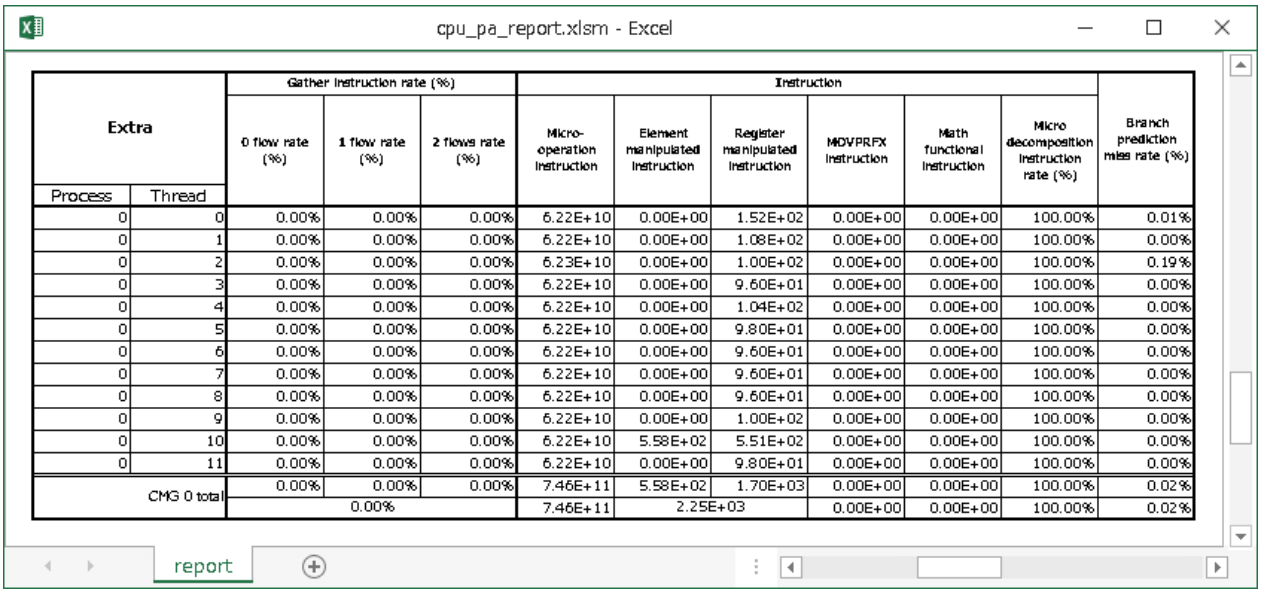
項目	説明
Double precision floating-point operation	倍精度浮動小数点演算数
Single precision floating-point operation	単精度浮動小数点演算数
Half precision floating-point operation	半精度浮動小数点演算数
GFLOPS by Active element rate	Active elementの比率を加味した浮動小数点演算性能 **注意** ・ 通常のGFLOPS値はすべてActive elementとして算出しています。そのため、Inactive elementの多いプログラムでは本来のGFLOPS値より高い値を出力します。Predicateを用いた演算を多用するプログラムでは、Active elementの比率を加味した、1秒あたりに実行された浮動小数点演算数を使用することでより精度の高い値を算出することが可能になります。ただし、store命令の比率が高いと、Active elementの比率を加味した、1秒あたりに実行された浮動小数点演算数の精度が悪化する場合があります。store命令の比率が低い場合のみ、限定的に利用することを推奨します。 ・ SVE命令以外ではActive elementの比率を100%として換算するため、SVE命令以外が多いケースでは、本来のActive elementの比率より高い値を出力します。SVE命令の比率が高い場合のみ、限定的に利用することを推奨します。

項目	説明
	・ 1byteデータ型の演算においては、Active elementの比率を100%として換算するため、1byteデータ型の演算が多い場合では、本来のActive elementの比率より高い値を出力します。1byteデータ型の演算の比率が低い場合のみ、限定的に利用することを推奨します。

4.2.2.8 Extra

Extra では、ギャザー命令の内訳、および命令ミックスに含まれない命令に関する情報を表示します。

図4.29 Extraのレイアウト



Extra		Gather instruction rate (%)			Instruction						
Process	Thread	0 flow rate (%)	1 flow rate (%)	2 flows rate (%)	Micro-operation instruction	Element manipulated instruction	Register manipulated instruction	MOVPRFX instruction	Math functional instruction	Micro decomposition instruction rate (%)	Branch prediction miss rate (%)
0	0	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	1.52E+02	0.00E+00	0.00E+00	100.00%	0.01%
0	1	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	1.08E+02	0.00E+00	0.00E+00	100.00%	0.00%
0	2	0.00%	0.00%	0.00%	6.23E+10	0.00E+00	1.00E+02	0.00E+00	0.00E+00	100.00%	0.19%
0	3	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	9.60E+01	0.00E+00	0.00E+00	100.00%	0.00%
0	4	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	1.04E+02	0.00E+00	0.00E+00	100.00%	0.00%
0	5	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	9.80E+01	0.00E+00	0.00E+00	100.00%	0.00%
0	6	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	9.60E+01	0.00E+00	0.00E+00	100.00%	0.00%
0	7	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	9.60E+01	0.00E+00	0.00E+00	100.00%	0.00%
0	8	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	9.60E+01	0.00E+00	0.00E+00	100.00%	0.00%
0	9	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	1.00E+02	0.00E+00	0.00E+00	100.00%	0.00%
0	10	0.00%	0.00%	0.00%	6.22E+10	5.58E+02	5.51E+02	0.00E+00	0.00E+00	100.00%	0.00%
0	11	0.00%	0.00%	0.00%	6.22E+10	0.00E+00	9.80E+01	0.00E+00	0.00E+00	100.00%	0.00%
CMG 0 total		0.00%	0.00%	0.00%	7.46E+11	5.58E+02	1.70E+03	0.00E+00	0.00E+00	100.00%	0.02%
		0.00%			7.46E+11	2.25E+03		0.00E+00	0.00E+00	100.00%	0.02%

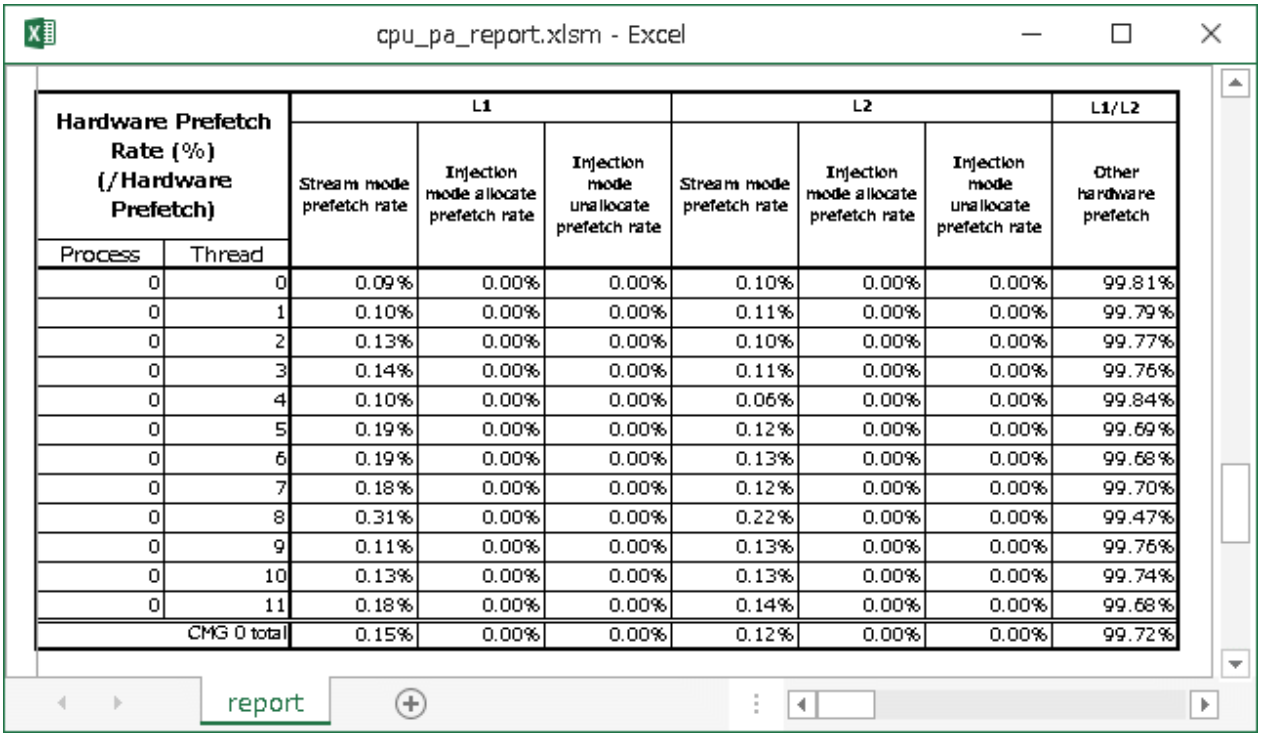
表4.21 Extraの出力項目

項目	説明
0 flow rate(%)	ギャザー命令のうちInactive elementであるために0フロー実行となった命令の比率(%)
1 flow rate(%)	ギャザー命令のうち1フロー実行となった命令の比率(%)
2 flows rate(%)	ギャザー命令のうち2フロー実行となった命令の比率(%)
Micro-operation instruction	マイクロ命令の数
Element manipulated instruction	要素間操作命令の数
Register manipulated instruction	レジスター間操作命令の数
MOVPRFX instruction	MOVPRFX命令の数
Math functional instruction	数学関数補助演算命令の数
Micro decomposition instruction rate(%)	マイクロ命令分解率(%)
Branch prediction miss rate(%)	分岐予測ミス率(%)

4.2.2.9 Hardware Prefetch Rate (%) (/Hardware Prefetch)

Hardware Prefetch Rate (%) (/Hardware Prefetch)では、ハードウェアプリフェッチの内訳を表示します。ハードウェアプリフェッチには“Stream detect mode”や“Prefetch injection mode”などのモードがあり、各モードがどの程度動作しているかを表示します。

図4.30 Hardware Prefetch Rate(%) (/Hardware Prefetch)のレイアウト



Hardware Prefetch Rate (%) (/Hardware Prefetch)		L1			L2			L1/L2
Process	Thread	Stream mode prefetch rate	Injection mode allocate prefetch rate	Injection mode unallocate prefetch rate	Stream mode prefetch rate	Injection mode allocate prefetch rate	Injection mode unallocate prefetch rate	Other hardware prefetch
0	0	0.09%	0.00%	0.00%	0.10%	0.00%	0.00%	99.81%
0	1	0.10%	0.00%	0.00%	0.11%	0.00%	0.00%	99.79%
0	2	0.13%	0.00%	0.00%	0.10%	0.00%	0.00%	99.77%
0	3	0.14%	0.00%	0.00%	0.11%	0.00%	0.00%	99.76%
0	4	0.10%	0.00%	0.00%	0.06%	0.00%	0.00%	99.84%
0	5	0.19%	0.00%	0.00%	0.12%	0.00%	0.00%	99.69%
0	6	0.19%	0.00%	0.00%	0.13%	0.00%	0.00%	99.68%
0	7	0.18%	0.00%	0.00%	0.12%	0.00%	0.00%	99.70%
0	8	0.31%	0.00%	0.00%	0.22%	0.00%	0.00%	99.47%
0	9	0.11%	0.00%	0.00%	0.13%	0.00%	0.00%	99.76%
0	10	0.13%	0.00%	0.00%	0.13%	0.00%	0.00%	99.74%
0	11	0.18%	0.00%	0.00%	0.14%	0.00%	0.00%	99.68%
CMG 0 total		0.15%	0.00%	0.00%	0.12%	0.00%	0.00%	99.72%

表4.22 Hardware Prefetch Rate(%) (/Hardware Prefetch)の出力項目

	項目	説明
L1キャッシュ	Stream mode prefetch rate	ハードウェアプリフェッチのうち、ストリームモードプリフェッチの比率
	Injection mode allocate prefetch rate	ハードウェアプリフェッチのうち、プリフェッチインジェクションモードがALLOCATEモードであるプリフェッチの比率
	Injection mode unallocate prefetch rate	ハードウェアプリフェッチのうち、プリフェッチインジェクションモードがUNALLOCATEモードであるプリフェッチの比率
L2キャッシュ	Stream mode prefetch rate	ハードウェアプリフェッチのうち、ストリームモードプリフェッチの比率
	Injection mode allocate prefetch rate	ハードウェアプリフェッチのうち、プリフェッチインジェクションモードがALLOCATEモードであるプリフェッチの比率
	Injection mode unallocate prefetch rate	ハードウェアプリフェッチのうち、プリフェッチインジェクションモードがUNALLOCATEモードであるプリフェッチの比率
L1/L2キャッシュ共通	Other hardware prefetch	ハードウェアプリフェッチのうち、その他のプリフェッチの比率

4.2.2.10 Data Transfer CMGs

Data Transfer CMGsでは、ユーザーが指定したCMGに対する自メモリー、他メモリー、Tofu、およびPCI間のスループット情報を表示します。

図4.31 Data Transfer CMGsのレイアウト

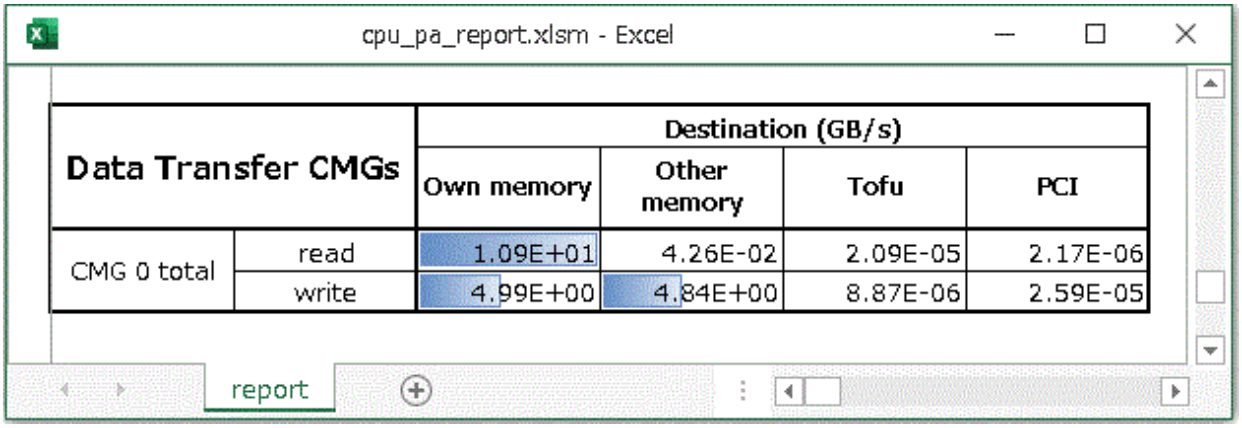


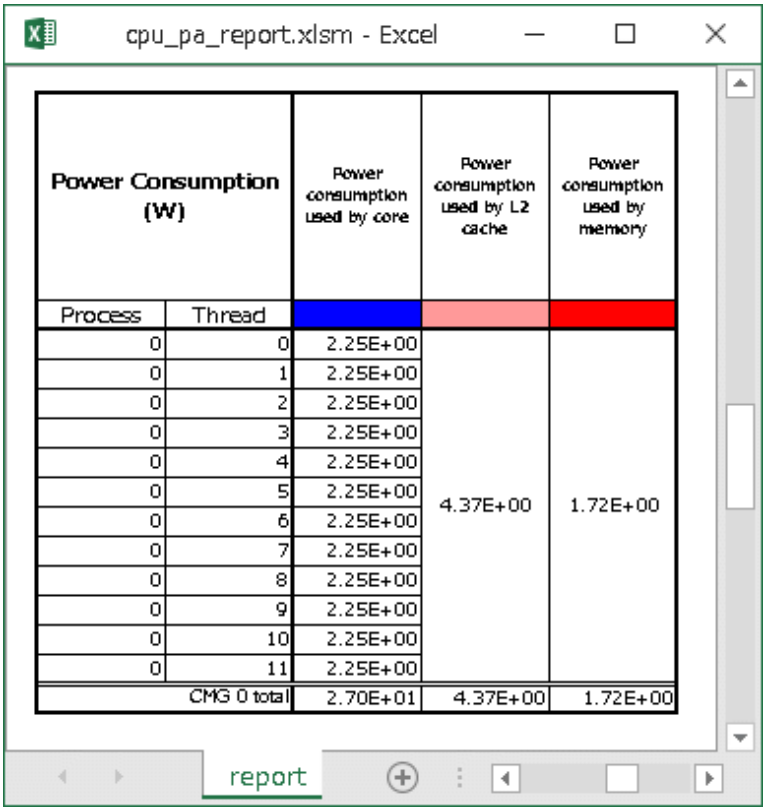
表4.23 Data Transfer CMGsの出力項目

	項目	説明
read	Own memory	自メモリーから対象CMGへのデータ転送のスループット性能(GB/s)
	Other memory	他メモリーから対象CMGへのデータ転送のスループット性能(GB/s)
	Tofu	Tofuから対象CMGへのデータ転送のスループット性能(GB/s)
	PCI	PCIから対象CMGへのデータ転送のスループット性能(GB/s)
write	Own memory	対象CMGから自メモリーへのデータ転送のスループット性能(GB/s)
	Other memory	対象CMGから他メモリーへのデータ転送のスループット性能(GB/s)
	Tofu	対象CMGからTofuへのデータ転送のスループット性能(GB/s)
	PCI	対象CMGからPCIへのデータ転送のスループット性能(GB/s)

4.2.2.11 Power Consumption (W)

Power Consumption (W)では、コア、L2キャッシュ、およびPMUカウンターから得られた値を換算して求めた消費電力をW(ワット)単位で表示します。Power Consumption (W) には対応する表とグラフがあります。“表4.24 Power Consumption (W)の出力項目”の出力項目を積み上げ棒グラフで表示します。表示対象のCMGが消費した電力を表示します。

図4.32 Power Consumption (W)のレイアウト



Power Consumption (W)		Power consumption used by core	Power consumption used by L2 cache	Power consumption used by memory
Process	Thread			
0	0	2.25E+00		
0	1	2.25E+00		
0	2	2.25E+00		
0	3	2.25E+00		
0	4	2.25E+00		
0	5	2.25E+00	4.37E+00	1.72E+00
0	6	2.25E+00		
0	7	2.25E+00		
0	8	2.25E+00		
0	9	2.25E+00		
0	10	2.25E+00		
0	11	2.25E+00		
	CMG 0 total	2.70E+01	4.37E+00	1.72E+00

図4.33 Power Consumption (W)のグラフ

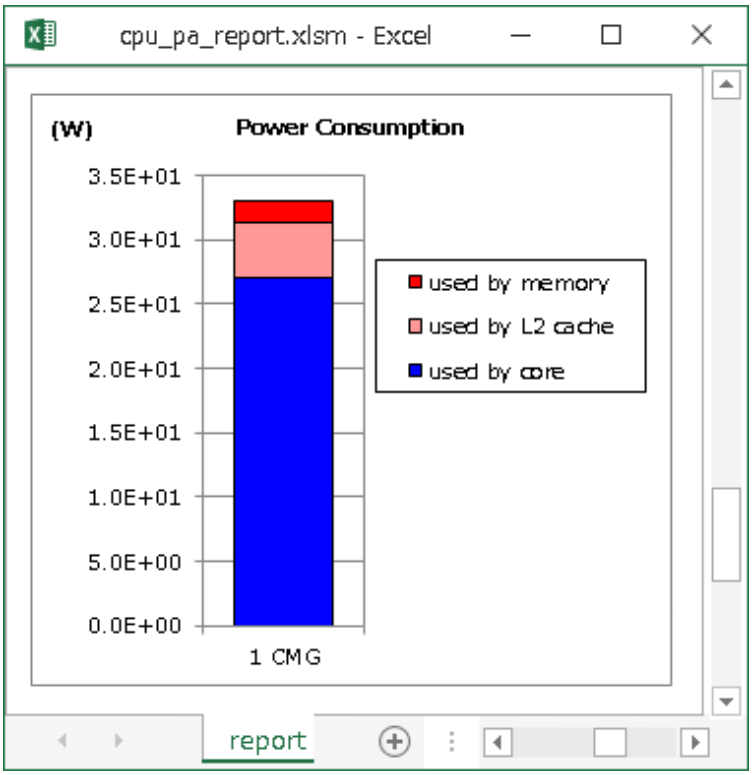


表4.24 Power Consumption (W)の出力項目

項目	説明
Power consumption used by core	コアで消費される電力

項目	説明
Power consumption used by L2 cache	L2キャッシュで消費される電力
Power consumption used by memory	メモリーで消費される電力

第5章 注意事項

プロファイラを使用する際の注意事項を以下に示します。

5.1 プロファイラ共通の注意事項

5.1.1 COARRAY使用時の注意

翻訳時オプション -Ncoarrayまたは--linkcoarrayが有効な場合、以下に注意してください。

- 基本プロファイラ、詳細プロファイラ共通
  - ランク番号またはプロセス番号に+1をした値が像番号に相当します。
  - COARRAY機能が利用しているMPIライブラリのコストが計上されることがあります。
  - プロファイルデータ計測環境情報のType of programの表示が“MPI”と表示されません。
  - プロファイルデータ計測環境情報のVirtual coordinateが表示されません。
- 基本プロファイラ固有
  - Process番号がすべて“Process 0”として出力されます。
- 詳細プロファイラ固有
  - MPI通信コスト情報が空になる場合があります。
  - ランク番号またはプロセス番号に+1をした値が像番号に相当します。

5.1.2 翻訳時オプションによる影響

プログラムの翻訳時に以下の翻訳時オプションを指定した場合、プロファイラが意図しない挙動となる場合があります。本文中に登場する翻訳時オプションについては“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。

翻訳時オプション -f{pie|PIE}

位置独立実行可能プログラム(PIE)をプロファイラで計測することはできません。

5.1.3 ノード共有ジョブ

ジョブ種別がノード共有ジョブの場合、プロファイルデータの計測に失敗する、または正しい情報が出力されない場合があります。MPI実行の場合、プロファイルデータの計測に失敗しプログラムが終了します。ノード共有ジョブの詳細については“ジョブ運用ソフトウェア”のマニュアルを参照してください。

5.1.4 スレッド並列の情報計測対象

プロファイラは富士通製コンパイラと連携して動作します。そのため、スレッド並列の計測は、富士通製コンパイラによる自動並列とOpenMPによる並列処理のみが対象になります。pthreadによる並列は計測されません。

5.1.5 mpiexecコマンド

- mpiexecコマンドに"-x LD_LIBRARY_PATH=value"を指定してプログラムを実行している場合は以下の点に注意してください。

  基本プロファイラ固有

  fipp走行時にはvalueに"/*製品インストールパス*/lib64/fipp"が優先されるようパスを設定してください。

例

```
fipp -C -d ./tmp mpiexec -x LD_LIBRARY_PATH=/製品インストールパス/lib64/fipp:/製品インストールパス/lib64:./tmplib ./
a.out
```

詳細プロファイラ固有

fapp走行時にはvalueに"/*製品インストールパス*/lib64/fapp"が優先されるようパスを設定してください。

例

```
fapp -C -d ./tmp mpiexec -x LD_LIBRARY_PATH=/製品インストールパス/lib64/fapp:/製品インストールパス/lib64:./tmplib ./
a.out
```

- mpiexecコマンドに実行定義ファイル指定方式の{ -app | --app }オプションを指定して計測を行った場合、動作を保証しません。

  mpiexecの実行定義ファイル指定方式の{ -app | --app }オプションについては、“MPI使用手引書”を参照してください。

5.1.6 MPIプロファイリングインターフェース利用による影響

プロファイラではMPI関数をフックすることにより情報計測しているため、プロファイラとMPIプロファイリングインターフェースを併用することはできません。MPIプロファイリングインターフェースを利用してMPI関数をフックするプログラムに対してプロファイラで計測を行った場合、以下のように情報が出力される場合があります。

- 基本プロファイラ、詳細プロファイラ共通
  - プロファイルデータ計測環境情報のType of programが“MPI”と表示されません。
  - プロファイルデータ計測環境情報のVirtual coordinateが表示されません。
- 基本プロファイラ固有
  - Process番号がすべて“Process 0”として出力されます。
- 詳細プロファイラ固有
  - MPI通信コスト情報が空になる場合があります。

5.1.7 MPIプログラムの言語間結合

MPIプログラムを言語間結合する場合は、リンク時に以下のオプションを指定してください。言語間結合については、“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。

使用するリンクコマンド	指定が必要なオプション
mpifccpxコマンド(ネイティブコンパイラではmpifccコマンド) mpiFCCpxコマンド(ネイティブコンパイラではmpiFCCコマンド)	-lfjprofmpifオプション
mpifrtpxコマンド(ネイティブコンパイラではmpifrtコマンド)	-lfjprofmpiオプション

5.1.8 終了ステータス

プロファイラで計測する際、返却される終了ステータスは、計測対象プログラムのものではなく、プロファイラの終了ステータスとなる場合があります。

5.1.9 LD_PRELOAD

環境変数LD_PRELOADを使用しないでください。使用した場合は、fippコマンドおよびfappコマンドは正しく動作しません。

5.2 基本プロファイラの注意事項

5.2.1 翻訳時オプションによる影響

プログラムの翻訳時に以下の翻訳時オプションを指定した場合、基本プロファイラで正しく計測ができない場合があります。本文中に登場する翻訳時オプションについては“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。

最適化オプション

C言語またはC++言語のプログラムの場合、基本プロファイラでプログラムのループ情報を計測するためには翻訳時に最適化オプション(-O1以上)を指定してください。

翻訳時オプション -g

翻訳時オプション -g が有効な場合、基本プロファイラは使用しないでください。-gを使用すると、デバッグ情報が増加するため、実行時間や使用メモリ量が増加する場合があります。

翻訳時オプション -Nlineまたは-ffj-line

基本プロファイラ使用時には、翻訳時オプション-Nline(デフォルトオプション) または -ffj-line を使用してください。

翻訳時オプション -Nnolineまたは-ffj-no-line

翻訳時オプション-Nnolineまたは-ffj-no-lineが有効な場合、fippコマンドの-Puserfuncオプションによるコストを正しく計測できません。このようなプログラムの場合、-Pnouserfuncオプションを有効にすることでコストを正しく計測できます。また、翻訳時オプション -Nnoline または -ffj-no-lineが有効な場合、MPIライブラリのコストを正しく計測できません。

翻訳時オプション -Kltoまたは-flto

翻訳時オプション -Kltoまたは-fltoが有効な場合、基本プロファイラが出力する以下の情報について正しく表示されない場合があります。

ループコスト分布情報

C言語およびC++言語では、そのオブジェクトに含まれる関数のループコスト分布情報は出力されません。

コールグラフ情報

表示される手続名に、リンク時最適化が内部的に生成した手続名が表示される場合があります。

ソースコード情報

各行のコストが正しく表示されない場合があります。

翻訳時オプション -fno-debug-info-for-profiling

C++言語のプログラムをclangモードで翻訳するときに翻訳時オプション -fno-debug-info-for-profiling が有効な場合、コンパイラがプロファイラ向けに生成する情報において、異なる実体に対して同じ名前が割り当たることがあります。この場合、基本プロファイラの結果において、同じ名前を持つ異なる実体を区別することはできません。

5.2.2 SIGVTALRMシグナルの捕捉および発行禁止

基本プロファイラはSIGVTALRMシグナルを捕捉することでプロファイルデータを計測しています。プログラム内でSIGVTALRMシグナルを捕捉または発行すると、プロファイルデータを正しく計測することができません。

5.2.3 プロファイルデータ計測時のサンプリング間隔

プロファイルデータ計測時のサンプリング間隔は、その値より小さい近似のOSのタイマー割込み間隔の影響を受けるため、指定した時間にならない場合があります。具体的には以下の影響があります。

OSのタイマー割込み間隔より小さいサンプリング間隔を指定した場合

OSのタイマー割込み間隔値に切り上げます。

OSのタイマー割込み間隔より大きいサンプリング間隔を指定した場合

OSのタイマー割込み間隔の倍数のうち、指定した値より小さい近似の値に丸めます。

タイマー割込みの間隔はOSのノイズ対策により異なりますが、およそ11～14ミリ秒です。サンプリング間隔とタイマー割込み間隔の対比について、以下に例を示します。

例

タイマー割込み間隔が14ミリ秒の場合

- オプション-iの指定値が10 : サンプリング間隔 14ミリ秒
- オプション-iの指定値が25 : サンプリング間隔 14ミリ秒
- オプション-iの指定値が100 : サンプリング間隔 98(=14*7)ミリ秒

5.2.4 プロファイラの作業域

プロファイルデータの計測時に基本プロファイラの作業域が不足した場合、プログラムの終了時に以下のメッセージを出力します。

```
fipp: work memory overflowed. specify memsize or more to -m option and retry.
```

メッセージ中の *memsize* は、プロファイルデータの計測時に使用する基本プロファイラの作業域の推奨値です。このメッセージが出力された場合、fipp コマンドに-m *memsize* オプションを指定して基本プロファイラの作業域を大きくして再度プロファイルデータを計測してください。

5.2.5 -pallオプション

fipppxコマンドまたはfippコマンドに-pallオプションを指定した場合、全プロセスの情報を出力します。並列プロセスが大きい場合、処理に時間がかかります。fippコマンドは全プロセスのプロファイルデータを計測します。fipppxコマンドまたはfippコマンド実行時に時間がかかる場合、一度にすべてのプロセスの情報を出力するのではなく、保存したプロファイルデータを使用して複数回に分けてプロセスを限定して出力することを推奨します。

5.2.6 CPU動作状況

CPU動作状況の計測対象には基本プロファイラ自身も含まれます。

5.2.7 コスト情報

- 実体が異なっていても、同じ名前の関数は1つの関数のコストとして計上されます。翻訳時に最適化オプション(-O1以上)が指定された場合、最適化の影響により各命令の行情報がソースの行と異なってしまうことがあります。例えば、各命令の行情報がループの開始位置となり、ループの開始位置でコストが計上される場合がこれに該当します。
- 出力項目の情報集計レベルがApplicationの場合、出力されるコストはプログラム全体のコストになります。
- ループコスト分布情報および行コスト分布情報では集計に含まれないコストがあるため、プログラム全体のコストと個々のコストの合計が一致しない場合があります。
- 各タスクにおけるコストがプログラム全体に占める割合は小さいことが予測されます。プロファイル結果の出力時に手続き情報の出力件数(-lオプション)を指定しない場合、省略値として10件分の手続情報をコストの多い順に出力するため、タスク情報が出力されない可能性があります。
- ヘッダーファイルを含めたソースコードで作成されたプログラムでは、ヘッダーファイルに該当する部分のコスト情報は、ヘッダーファイルの行番号としてプロファイルデータを出力します。
- インライン展開が行われたプログラムでは、インライン展開された部分のコストはインライン展開呼出し元手続のコストとして計上します。
- 基本プロファイラでコストを計測した場合、以下のコスト情報が関数呼び出し行の次の行番号に表示されることがあります。
  - スレッド間同期待ちコスト
  - MPIコスト
  - -Puserfuncオプションを指定して計測したユーザー定義関数のコスト

  例

  コメントなど、処理のない行は無視します

```
67 void call_barrier()
68 {
69   int idx=0, cnt=10000;
70   for (idx=0; idx<cnt; idx++)
71     {
72       MPI_Barrier(MPI_COMM_WORLD);
73       /* comment */
74     }
75   return;
76 }
```

```
           MPI          %  Communication (s)     Line
-------------------------------------------------------
            33    94.2857             0.3430       --   Process      1
-------------------------------------------------------
            20    57.1429             0.2079       74   call_barrier
            12    34.2857             0.1247       18   main
             1     2.8571             0.0104       64   main
```

次の処理(例題では“}”)が同一行にある場合、行番号のズレは発生しません。

```
67 void call_barrier()
68 {
69   int idx=0, cnt=10000;
70   for (idx=0; idx<cnt; idx++) { MPI_Barrier(MPI_COMM_WORLD); } return; }
```

```
           MPI          %  Communication (s)     Line
-------------------------------------------------------
            33    91.6667             0.3362       --   Process      1
-------------------------------------------------------
            12    33.3333             0.1223       18   main
             1     2.7778             0.0102       64   main
            20    55.5556             0.2038       70   call_barrier
```

- 最適化の影響により、MPIコスト、スレッド間同期待ちコスト情報が出力できない場合があります。

5.2.8 ソースコード情報

基本プロファイラの手続コスト分布情報においてユーザー手続が上位5件に存在しない場合、ソースコード情報は出力されません。該当する場合、次のメッセージが出力されます。

```
Symbol information up to the 5th do not include information which relates to the source code.
```

5.2.9 コールグラフ情報

- 基本プロファイラのコールグラフ情報において、次のようなプログラムでは呼出し経路が解析できない場合があります。

  最適化によりフレームポインタレジスタが保証されていない場合

  呼出し経路のネストレベルが1の場合は、ネストレベルが“<???>”として出力されます。呼出し経路のネストレベルが2以上の場合は、コールグラフ情報が誤って出力されます。コールグラフ情報を正しく出力するためには、翻訳時に以下のオプションを指定してください。

  - FortranおよびC言語/C++言語(tradモード)の場合：-Knoomitfp オプション
  - C言語/C++言語(clangモード)の場合：-fno-omit-frame-pointer オプション

  詳細については、“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。

  呼出し経路のネストレベルが128以上となる場合

  ネストレベルが“<???>”として出力されます。

手続の入口または出口でサンプリングによる割込みが発生した場合

呼出し元を正しく解析できずに、コールグラフ情報が誤って出力されることがあります。行コスト分布情報に以下の行のコストが含まれている場合、この現象が起きていることがあります。

Fortranのプログラムの場合

- SUBROUTINE文
- FUNCTION文
- ENTRY文
- RETURN文
- END文

C言語/C++言語のプログラムの場合

- 関数の開始を示す括弧
- 関数の終了を示す括弧
- return文

- インライン展開が行われたプログラムでは、インライン展開された部分のコストはインライン展開呼出し元手続のコストとして計上します。
- ある手続において、以下の2つが等しくならないことがあります。
  - 手続コスト分布情報におけるApplicationレベルの当該手続のコストの値
  - 当該手続のコールグラフ情報におけるコストの値の合計

5.2.10 行番号0へのコスト

行番号 0 にコストが計上される場合があります。行番号0のコスト情報が存在する場合の出力内容は以下の通りです。

コスト情報の場合

行番号0にサンプリングしたコスト情報を出力する場合があります。コスト情報は複数出力する場合があります。コスト情報の手続名には、コストが計上された手続名が入ります。

ソースコード情報の場合

行番号0として計上されたコストの合計値をソースコードの先頭行(0行)に出力します。ソースコードには固定値“/* other costs */”が入ります。

5.2.11 -u オプション

- 翻訳時に手続名が加工された場合、手続名と生成手続名が一致せず正しく合算されない可能性があります。翻訳後の手続名については“Fortran使用手引書”、“C言語使用手引書”、および“C++言語使用手引書”を参照してください。
- -uオプションと-f *func_name*オプションを同時に指定する場合、以下の注意事項があります。
  - -f *func_name*オプションに生成手続名を指定した場合、生成手続が属する手続の情報を出力します。
  - -f *func_name*オプションに指定した手続または生成手続のコストが0、かつ、手続と生成手続のコストの合算値が1以上の場合、警告メッセージを出力して-f *func_name*オプションを無視します。
- “表2.6 スレッド並列プログラムの生成手続名”の「生成手続名」列に記載の「手続名」が、あるコストで出力されない場合があります。このときに、-uオプションを指定すると、該当コストの「手続名」列には何も出力されません。

5.2.12 -Minlinedオプション

実行ファイルに最適化によってインライン展開された関数が多く含まれる場合、基本プロファイラの処理に時間がかかる、または使用メモリが増加することがあります。

5.2.13 サンプリング数

sleep関数(sleep、usleep、nanosleep)や入出力文の処理を含むプログラムを基本プロファイラで計測する場合、コスト情報のサンプリング数を正しく出力しない場合があります。

5.2.14 サンプリングによるシグナル割込み

基本プロファイラのサンプリングはシグナル割込み(SIGVTALRM)で実装されています。そのため、シグナル割込みによって影響がある関数を使用している場合には、期待どおりの動作にならない場合があります。その場合には、シグナル割込みが発生することを前提にプログラムを修正してください。

5.3 詳細プロファイラの注意事項

5.3.1 MPIのスレッドサポート

詳細プロファイラでは、スレッドサポートのレベルがMPI_THREAD_SERIALIZEDまたはMPI_THREAD_MULTIPLEの場合、プロファイルデータを正しく計測することができません。

5.3.2 CPU性能解析情報

-Hmethod=fastオプションを指定した場合、計測を行っているプロセスがSleep状態でCPUに割り当てられていない間も情報を計測します。例えば、-Hmethod=normalオプションを指定した場合と比べ、CPU性能解析レポートの実行時間の値が大きくなる場合があります。

CPU性能解析情報の計測対象には詳細プロファイラ自身も含まれます。

5.3.3 -Hevent_rawオプション

-Hevent_rawオプションを-Hmethod=normalオプションと同時に指定した場合、同じイベント番号を複数指定することはできません。以下のエラーメッセージを出力し、処理が終了します。

```
RTINF2xxx : Internal error. PAPI return code = xxx.
```

5.3.4 MPIライブラリの経過時間情報

mpiFCCコマンドで作成したアプリケーションの場合、MPIライブラリの経過時間情報のMPIライブラリ名は、C++言語のメンバ関数名を出力します。子スレッドから呼び出されるMPI関数のMPIライブラリ経過時間情報は、詳細プロファイラで計測できません。

5.3.5 CPUバインド

プロファイルデータを計測する場合、スレッドとCPUが1対1の関係になるようにバインドする必要があります。

CPUバインドするための詳細設定については、スレッド並列プログラムを実行するときは、“Fortran使用手引書”、“C言語使用手引書”、または“C++言語使用手引書”を参照してください。
非スレッド並列かつMPIプログラムを実行(mpiexecで実行)するときは、VCOORDファイルを使ってCPUバインド可能です。VCOORDファイルの指定方法は“MPI使用手引書”の{ -vcoordfile | --vcoordfile }オプションを参照してください。VCOORDファイルではすべてのプロセスに対してCPU(コア)数を1(core=1)に指定してください。
非スレッド並列かつ非MPIプログラムを実行するときは、tasksetコマンドまたはnumactlコマンドでCPUバインド可能です。詳細は各コマンドのmanページを参照してください。

5.3.6 MPI通信コスト情報を計測できないルーチン

“mpi_f08.mod”または“mpi_f08_ext.mod”に定義された一部のルーチンは“3.2.2.3 MPI通信コスト情報”を計測できません。
“mpi_f08.mod”および“mpi_f08_ext.mod”については、“MPI使用手引書”を参照してください。

5.4 CPU性能解析レポートの注意事項

5.4.1 CPU性能解析レポートファイル

“4.1.6 CPU性能解析レポートの作成”が完了したCPU性能解析レポートファイル(cpu_pa_report.xlsm)をファイル保存した場合、新規の読み込み処理を行いません。別の計測結果を読み込む場合、データ読み込み後にファイル保存をしないか、または新規にCPU性能解析レポートファイルをコピーしてください。

5.4.2 動的生成したプロセス

CPU性能解析レポートは動的生成したプロセスには対応していません。

付録A　トラブルシューティング

プロファイラに関するトラブルシューティングについて説明します。

A.1　基本プロファイラ

A.1.1　プロファイルデータの計測を行うと通常実行に比べて実行時間が長くかかる

プロファイルデータの計測時、通常実行時に比べて実行時間が長くかかってしまう場合、以下のいずれかの原因の可能性があります。

- 基本プロファイラによるサンプリング時のオーバーヘッドが原因の可能性があります。fippコマンドの-iオプションでサンプリング間隔を大きくすることにより、オーバーヘッドの発生回数を減少させプログラムの実行時間を短縮することができます。fippコマンドについては、“2.1.4 プロファイルデータの計測”を参照してください。
- 基本プロファイラ使用時に計測に必要な情報をプログラムから読み込みますが、プログラムの行数、ファイル数、またはシンボル数が多い場合、読み込み処理に時間がかかる可能性があります。-Sregionオプションを使用するなど計測対象範囲が特定のオブジェクトに限定できる場合、計測対象外のオブジェクト翻訳時に翻訳時オプション-Nnolineまたは-ffj-no-lineを指定することで読み込み処理の時間を削減することができます。翻訳時オプション-Nnolineまたは-ffj-no-lineについては“Fortran使用手引書”、“C言語使用手引書”、または“C++使用手引書”を参照してください。ただし、翻訳時オプション-Nnolineまたは-ffj-no-lineを付けたオブジェクトは以下の情報が出力されません。
  - “2.2.2.4.1 手続コスト分布情報”の“手続の開始行番号”および“手続の終了行番号”
  - “2.2.2.4.2 ループコスト分布情報”の全情報
  - “2.2.2.4.3 行コスト分布情報”の全情報

A.1.2　プロファイルデータの計測を行うと通常実行に比べてメモリー使用量が増加する

プロファイルデータの計測時、通常実行時に比べてメモリー使用量が増加する場合、以下の可能性があります。

基本プロファイラによるプロファイルデータ計測時、実行ファイルの手続情報、ループ情報、および行情報のメモリー領域(以降、デバッグ情報と呼びます)をfippコマンドの-m *memsize*オプションで確保したメモリー領域とは別にプロセスごと確保します。そのため、メモリー使用量が増加しメモリー不足によりプロファイルデータの計測に失敗することがあります。この場合、実行ファイルのデバッグ情報量を削減してプロファイルデータを計測してください。デバッグ情報のメモリー使用量の目安は1手続あたり約300Byte、1ループあたり約150Byte、1行あたり約150Byteです。ただし、デバッグ情報量は手続名の長さ、実行スレッド数、基本プロファイラに指定したオプション等により変動します。また、オブジェクト翻訳時に翻訳時オプション-Nnolineまたは-ffj-no-lineを有効にすることでループ情報と行情報用のメモリーを確保しなくなりデバッグ情報量を削減することができます。翻訳時オプション-Nnolineまたは-ffj-no-lineについては“Fortran使用手引書”、“C言語使用手引書”、または“C++使用手引書”を参照してください。ただし、翻訳時オプション-Nnolineまたは-ffj-no-lineを付けたオブジェクトは以下の情報が出力されません。

- “2.2.2.4.1 手続コスト分布情報”の“手続の開始行番号”および“手続の終了行番号”
- “2.2.2.4.2 ループコスト分布情報”の全情報
- “2.2.2.4.3 行コスト分布情報”の全情報

A.1.3　ソースコードにない手続名(ライブラリ名など)が表示される

ユーザー手続名だけのプロファイルデータを計測するためには、fippコマンドで-Puserfuncオプションを指定してください。fippコマンドについては、“2.1.4 プロファイルデータの計測”を参照してください。

A.1.4　プロファイルデータのオープンに失敗する

プロファイルデータ計測対象のプログラムが正常終了していない可能性があります。プロファイルデータを再度計測してください。

A.1.5　__?unknownというシンボルが出力される

プロファイルデータの計測時に、コストがどの手続にも該当しないことがあります。このコストは、“__?unknown”というシンボル名で出力されます。計測対象がスレッド並列プログラムの場合、スレッド間同期待ちコストがこのケースに該当することがあります。プロファイルデータの

計測時にfippコマンドの-iオプションでサンプリング間隔を大きくすると、このシンボルが出力されなくなることがあります。また、fippコマンドの-Icallオプションと-Inocallオプションとではフレームポインタを辿る処理に違いがあり、プロファイルデータの計測時にfippコマンドの-Icallオプションでコールグラフ情報を計測すると、このシンボルが出力されなくなることがあります。fippコマンドについては、“2.1.4 プロファイルデータの計測”を参照してください。

A.2 詳細プロファイラ

A.2.1 プロファイルデータの計測を行うと通常実行に比べて実行時間が長くかかる

詳細プロファイラルーチンの測定区間数や呼出し回数を少なくすることにより、詳細プロファイラの実行時間を短縮することができます。

A.2.2 プロファイルデータのオープンに失敗する

プロファイルデータ計測対象のプログラムが正常終了していない可能性があります。プロファイルデータを再度計測してください。

A.3 CPU性能解析レポート

A.3.1 CSV形式ファイルの読み込みに失敗する(ファイルの行数制限越え)

CSV形式ファイルの行数がExcelで扱える最大行数(1048576行)を超える場合、以下のメッセージが出力され、CSV形式ファイルを読み込めません。CSV形式ファイルの行数を最大行数(1048576行)内に抑えてください。

```
The file format is not supported.
```

以下は、CSV形式ファイルの行数を削減する方法です。(併用可能)

1. fapp_start/fapp_stopの計測区間の数を削減する。
2. fapp_start/fapp_stopの引数levelを使用する。
   引数levelで計測区間をレベル分けし、-Lオプションを指定して不要なレベルの区間の出力を抑止します。
3. fapppxコマンドに-Inompiオプションを指定し、MPI情報の出力を抑止する。
4. fapppxコマンドに-p<n>,limit=<m>オプションを指定し、出力情報のサイズ増加を抑止する。

付録B　メッセージ一覧

ここでは、プロファイラが出力する代表的なメッセージについて説明します。メッセージは標準エラー出力に出力します。

B.1　メッセージ一覧(fippコマンド)

fipp: -C or -A option is not specified.

[メッセージの説明]

-Cオプション、または-Aオプションが指定されていません。

[システムの処理]

処理を終了します。

[利用者の処置]

-Cオプション、または-Aオプションを指定してください。

fipp: -d option is not specified.

[メッセージの説明]

-dオプションが指定されていません。

[システムの処理]

処理を終了します。

[利用者の処置]

-dオプションを指定してください。

fipp: The specified argument of -d option is not directory.

[メッセージの説明]

-dオプションの引数指定に誤りがあります。

[システムの処理]

処理を終了します。

[利用者の処置]

-dオプションの引数指定が正しいか確認してください。

fipp: The specified directory in -d option is permission denied.

[メッセージの説明]

-dオプションに指定したディレクトリに読み取り権、書き込み権、または実行権がありません。

[システムの処理]

処理を終了します。

[利用者の処置]

-dオプションに指定したディレクトリに読み取り権、書き込み権、および実行権を付与してください。

fipp: The executable program was not specified to an operand.

[メッセージの説明]

実行ファイルが指定されていません、または存在していません。

[システムの処理]

処理を終了します。

[利用者の処置]

存在する実行ファイルを指定してください。

---

fipp: The specified argument of -l *parm* option is invalid.

[メッセージの説明]

-lオプションに誤った引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

---

fipp: The specified argument of -l *parm* option is not integer.

[メッセージの説明]

-lオプションに数値以外の引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

---

fipp: The specified value of -l *parm* option is outside the range. The default value is applied. { limit = 0 }

[メッセージの説明]

-lオプションに範囲外の引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

-l 0オプションを有効にして処理を継続します。

[利用者の処置]

引数 *parm* を修正してください。

---

fipp: The specified argument of -H option is invalid.

[メッセージの説明]

-Hオプションの引数指定に誤りがあります。

[システムの処理]

処理を終了します。

[利用者の処置]

-Hオプションの引数を修正してください。

---

fipp: The specified argument of -P *parm* option is invalid.

[メッセージの説明]

-Pオプションに誤った引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

---

fipp: -Inocall option cannot be specified together with -Puserfunc option.

[メッセージの説明]

-Puserfuncオプションと-Inocallオプションが同時に指定されています。

[システムの処理]

処理を終了します。

[利用者の処置]

-Puserfuncオプション指定時は、-Icallオプションを指定してください。

---

fipp: The -Icall option is necessary for specified -Puserfunc option.

[メッセージの説明]

-Puserfuncオプションが指定されていますが、-Icallオプションが指定されていません。

[システムの処理]

処理を終了します。

[利用者の処置]

-Puserfuncオプション指定時は、-Icallオプションを指定してください。

---

fipp: The specified argument of -S *parm* option is invalid.

[メッセージの説明]

-Sオプションに誤った引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

fipp: The specified argument of -i *parm* option is not integer.

[メッセージの説明]

-iオプションに数値以外の引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

fipp: The specified value of -i *parm* option is outside the range. The default value is applied. { interval = 100 }

[メッセージの説明]

-iオプションに範囲外の引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

-i 100オプションを有効にして処理を継続します。

[利用者の処置]

引数 *parm* を修正してください。

fipp: The specified argument of -m *parm* option is not integer.

[メッセージの説明]

-mオプションに数値以外の引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

fipp: The specified value of -m *parm* option is outside the range. The default value is applied. { memsize = 3000 }

[メッセージの説明]

-mオプションに範囲外の引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

-m 3000オプションを有効にして処理を継続します。

[利用者の処置]

引数 *parm* を修正してください。

---

fipp: The specified value of -m *parm* option is within the range but large. Therefore, the working memory may not be allocated.

[メッセージの説明]

-mオプションに大きな引数 *parm* が指定されています。1ノードあたりのプロセス数と1プロセスあたりのスレッド数の積が大きい場合、作業用メモリーが確保できないことがあります。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を継続します。

[利用者の処置]

必要に応じて、1ノードあたりのプロセス数と1プロセスあたりのスレッド数の積、または、引数 *parm* を小さくしてください。

---

fipp: The specified argument of -L *parm* option is invalid.

[メッセージの説明]

-Lオプションに誤った引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

---

fipp: The specified argument of -W *parm* option is invalid.

[メッセージの説明]

-Wオプションに誤った引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

---

fipp: Cannot specify the "-Wspawn" option when executing non-MPI program.

[メッセージの説明]

非MPIプログラムに対して-Wspawnオプションを指定しています。

[システムの処理]

処理を終了します。

[利用者の処置]

-Wspawnオプションを指定しないでください。

---

fipp: The profiling data is not correctly generated.

[メッセージの説明]

実行ファイルのファイル形式に誤りがあります。

[システムの処理]

処理を終了します。

[利用者の処置]

実行ファイルのファイル形式を確認してください。

---

fipp: The files had already existed in the specified value of fjprof_spawn_dir_name key.

[メッセージの説明]

infoキー"fjprof_spawn_dir_name"に指定したディレクトリが存在しています。

[システムの処理]

処理を終了します。

[利用者の処置]

fjprof_spawn_dir_nameに指定した値が正しいか確認してください。

---

fipp: The specified value of fjprof_spawn_dir_name key is not directory.

[メッセージの説明]

infoキー"fjprof_spawn_dir_name"に指定したディレクトリと同名のファイルが存在しています。

[システムの処理]

処理を終了します。

[利用者の処置]

fjprof_spawn_dir_nameに指定した値が正しいか確認してください。

---

fipp: The specified value of fjprof_spawn_dir_name key is permission denied.

[メッセージの説明]

infoキー"fjprof_spawn_dir_name"に指定したディレクトリに読み取り権、書き込み権、または実行権がありません。

[システムの処理]

処理を終了します。

[利用者の処置]

fjprof_spawn_dir_name に指定したディレクトリに読み取り権、書き込み権、および実行権を付与してください。

---

fipp: obsolete option *parm1* changed to *parm2*.

[メッセージの説明]

parm1 は旧オプションです。*parm2* オプションに変更します。

[パラメーターの説明]

parm1 : 旧オプション名

parm2 : 新オプション名

[システムの処理]

parm2 オプションを有効にして処理を継続します。

[利用者の処置]

parm1 オプションを*parm2* オプションに修正してください。

---

fipp: *parm1* option was specified. *parm2* option is ignored.

[メッセージの説明]

parm1 オプションが指定されたため、*parm2* オプションを無効にしました。

[パラメーターの説明]

parm1 : 有効になるオプション名

parm2 : 無効になるオプション名

[システムの処理]

parm1 オプションのみ有効にして処理を継続します。

[利用者の処置]

parm1 オプション指定時は、*parm2* オプションを指定しないでください。

---

fipp: *parm1* option cannot be specified together with *parm2* option.

[メッセージの説明]

parm1 オプションと*parm2* オプションが同時に指定されています。

[パラメーターの説明]

parm1 : オプション名

parm2 : オプション名

[システムの処理]

処理を終了します。

[利用者の処置]

parm1 オプションか*parm2* オプションのどちらか片方を削除してください。

---

fipp: The specified argument of -M *parm* option is invalid.

[メッセージの説明]

-Mオプションに誤った引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

B.2 メッセージ一覧(fipppxコマンド)

fipppx: -A option is not specified.

[メッセージの説明]

-Aオプションが指定されていません。

[システムの処理]

処理を終了します。

[利用者の処置]

-Aオプションを指定してください。

fipppx: invalid argument of option -- *parm*

[メッセージの説明]

parm オプションの引数指定に誤りがあります。

[パラメーターの説明]

parm : 問題のあるオプション名、または引数

[システムの処理]

処理を終了します。

[利用者の処置]

parm の情報をもとにオプション、または引数を修正してください。

fipppx: No information on the specified region. : *func_name*

[メッセージの説明]

-fオプションに指定した手続名 *func_name* のコスト情報はありません。

[パラメーターの説明]

func_name : 手続名

[システムの処理]

本オプションを無視して処理を継続します。

[利用者の処置]

特にありません。

fipppx: *parm1* option cannot be specified together with *parm2* option.

[メッセージの説明]

parm1 オプションと*parm2* オプションが同時に指定されています。

[パラメーターの説明]

parm1 : オプション名

parm2 : オプション名

[システムの処理]

処理を終了します。

[利用者の処置]

parm1 オプションか*parm2* オプションのどちらか片方を削除してください。

B.3 メッセージ一覧(fappコマンド)

fapp: -C or -A option is not specified.

[メッセージの説明]

-Cオプション、または-Aオプションが指定されていません。

[システムの処理]

処理を終了します。

[利用者の処置]

-Cオプション、または-Aオプションを指定してください。

fapp: -d option is not specified.

[メッセージの説明]

-dオプションが指定されていません。

[システムの処理]

処理を終了します。

[利用者の処置]

-dオプションを指定してください。

fapp: The specified argument of -d option is not directory.

[メッセージの説明]

-dオプションの引数指定に誤りがあります。

[システムの処理]

処理を終了します。

[利用者の処置]

-dオプションの引数指定が正しいか確認してください。

fapp: The specified directory in -d option is permission denied.

[メッセージの説明]

-dオプションに指定したディレクトリに読み取り権、書き込み権、または実行権がありません。

[システムの処理]

処理を終了します。

[利用者の処置]

-dオプションに指定したディレクトリに読み取り権、書き込み権、および実行権を付与してください。

fapp: The specified argument of -I *parm* option is invalid.

[メッセージの説明]

-Iオプションに誤った引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

---

fapp: The specified argument of -H option is invalid.

[メッセージの説明]

-Hオプションの引数指定に誤りがあります。

[システムの処理]

処理を終了します。

[利用者の処置]

-Hオプションの引数を修正してください。

---

fapp: The specified *parm* argument of -H option is invalid.

[メッセージの説明]

-Hオプションの引数指定に誤りがあります。

[パラメーターの説明]

parm : “event=”、“event_raw=”、“mode=”、または“method=”

[システムの処理]

処理を終了します。

[利用者の処置]

-Hオプションの引数を修正してください。

---

fapp: The number of PMU event specified to -Hevent_raw option exceed the limit. (max = 8)

[メッセージの説明]

-Hevent_raw=オプションの引数に指定した数値の個数が多すぎます。

[システムの処理]

処理を終了します。

[利用者の処置]

-Hオプションの引数を修正してください。

---

fapp: The specified argument of -W *parm* option is invalid.

[メッセージの説明]

-Wオプションに誤った引数 *parm* が指定されています。

[パラメーターの説明]

parm : ユーザーが指定した引数

[システムの処理]

処理を終了します。

[利用者の処置]

引数 *parm* を修正してください。

fapp: The executable program was not specified to an operand.

[メッセージの説明]

実行ファイルが指定されていません、または存在していません。

[システムの処理]

処理を終了します。

[利用者の処置]

存在する実行ファイルを指定してください。

fapp: The files had already existed in the specified value of fjprof_spawn_dir_name key.

[メッセージの説明]

infoキー"fjprof_spawn_dir_name"に指定したディレクトリが存在しています。

[システムの処理]

処理を終了します。

[利用者の処置]

fjprof_spawn_dir_nameに指定した値が正しいか確認してください。

fapp: The specified value of fjprof_spawn_dir_name key is not directory.

[メッセージの説明]

infoキー"fjprof_spawn_dir_name"に指定したディレクトリと同名のファイルが存在しています。

[システムの処理]

処理を終了します。

[利用者の処置]

fjprof_spawn_dir_nameに指定した値が正しいか確認してください。

fapp: The specified value of fjprof_spawn_dir_name key is permission denied.

[メッセージの説明]

infoキー"fjprof_spawn_dir_name"に指定したディレクトリに読み取り権、書き込み権、または実行権がありません。

[システムの処理]

処理を終了します。

[利用者の処置]

fjprof_spawn_dir_name に指定したディレクトリに読み取り権、書き込み権、および実行権を付与してください。

fapp: *parm1* option was specified. *parm2* option is ignored.

[メッセージの説明]

parm1 オプションが指定されたため、*parm2* オプションを無効にしました。

[パラメーターの説明]

parm1 : 有効になるオプション名

parm2 : 無効になるオプション名

[システムの処理]

parm1 オプションのみ有効にして処理を継続します。

[利用者の処置]

parm1 オプション指定時は、*parm2* オプションを指定しないでください。

---

fapp: obsolete option *parm1* changed to *parm2*.

[メッセージの説明]

parm1 は旧オプションです。*parm2* オプションに変更します。

[パラメーターの説明]

parm1 : 旧オプション名

parm2 : 新オプション名

[システムの処理]

parm2 オプションを有効にして処理を継続します。

[利用者の処置]

parm1 オプションを*parm2* オプションに修正してください。

---

fapp: *parm1* option cannot be specified together with *parm2* option.

[メッセージの説明]

parm1 オプションと*parm2* オプションが同時に指定されています。

[パラメーターの説明]

parm1 : オプション名

parm2 : オプション名

[システムの処理]

処理を終了します。

[利用者の処置]

parm1 オプションか*parm2* オプションのどちらか片方を削除してください。

B.4 メッセージ一覧(fapppxコマンド)

---

fapppx: -A option is not specified.

[メッセージの説明]

-Aオプションが指定されていません。

[システムの処理]

処理を終了します。

[利用者の処置]

-Aオプションを指定してください。

---

fapppx: invalid argument of option -- *parm*

[メッセージの説明]

parm オプションの引数指定に誤りがあります。

[パラメーターの説明]

parm : 問題のあるオプション名、または引数

[システムの処理]

処理を終了します。

[利用者の処置]

parm の情報をもとにオプション、または引数を修正してください。

---

fapppx: The -Hpa option is obsolete and will be removed in a future release. The option is ignored.

[メッセージの説明]

-Hpaオプションは不要になりました。将来的には廃止予定です。

[システムの処理]

本オプションを無視して処理を継続します。

[利用者の処置]

特にありませんが、-Hpaオプションを指定しないことを推奨します。

---

fapppx: obsolete option *parm1* changed to *parm2*.

[メッセージの説明]

parm1 は旧オプションです。*parm2* オプションに変更します。

[パラメーターの説明]

parm1 : 旧オプション名

parm2 : 新オプション名

[システムの処理]

parm2 オプションを有効にして処理を継続します。

[利用者の処置]

parm1 オプションを *parm2* オプションに修正してください。