

PMlibとは

- アプリケーションの計算性能をモニターするクラスライブラリ
 - 注目する区間の性能統計情報を簡単に測定・レポートする。
 - 富岳版では電力監視・制御インタフェースを追加
 - Power APIを容易に利用でき、HPCに適したレポート機能を利用できる
 - C++言語、Fortran言語に対応。
 - Linux系OSをもつコンピュータの上で使える。
 - 富岳
 - 富士通FX700、FX100
 - Intel Xeonサーバ(SkyL/IvyB/SandyB)・PC
 - Apple Macbook (ただしHWPC機能は利用不可)
 - オープンソース。
 - 最新版パッケージ (富岳電力制御機能対応)
 - <https://github.com/mikami3heart/PMlib>
 - 公式リポジトリ (電力未対応)
 - <http://avr-aics-riken.github.io/PMlib/>
 - 必要な前提ソフトウェア環境
 - Linux OS, C++, C, Fortranコンパイラ
 - (オプションに応じて) MPIライブラリ、PAPIライブラリ、OTFライブラリ

PMlibの利用方法

1. PMlibライブラリをインストールする
2. アプリケーション中の測定区間を決める
 - ソースプログラム中の注目箇所にPMlib APIを追加
3. アプリケーションをコンパイルしてPMlibとリンクする
4. アプリケーションを実行する
 - 実行時に性能統計情報がレポートされる
 - レポート出力情報は環境変数で制御可能なので、再コンパイルの必要はない

- 測定区間は少数の属性を持つ
 - ラベル：任意の名称
 - 測定する計算量の種類： 演算量、データ移動量など
- 測定する計算量をどう選択・算出するか？
 - PMlibがHWPC値を自動測定する方法
 - ユーザが明示的に申告する方法

PMlib基本API

- PMlib基本APIの一覧

引数 [内]は省略可能

関数名 (C++)	関数名 (C)	関数名 (Fortran)	機能	呼び出し位置と回数	引数
initialize	C_pm_initialize	f_pm_initialize	PMlib全体の初期化	冒頭・一回	[(1)測定区間数]
start	C_pm_start	f_pm_start	測定の開始	任意・任意 (startとstopでペア)	(1)ラベル
stop	C_pm_stop	f_pm_stop	測定の停止	任意・任意 (startとstopでペア)	(1)ラベル、[(2)計算量、(3)任意の係数]
report	C_pm_report	f_pm_report	統計レポートを出力	測定終了時・一回	(1)出力ファイルポインタ、[(2)ホスト名、(3)任意のコメント、(4)区間の表示順序指定]

- 電力制御用の追加関数（呼び出さなくとも電力レポートは出力可能）

関数名 (C++)	関数名 (C)	関数名 (Fortran)	機能	呼び出し位置と回数	引数
getPowerKnob	C_pm_getpowerknob	f_pm_getpowerknob	パワーノブ情報を取得する	任意・制約なし	(1)ノブの種類、(2)ノブの値
setPowerKnob	C_pm_setpowerknob	f_pm_setpowerknob	パワーノブの値を設定する	任意・制約なし	(1)ノブの種類、(2)ノブの値

- この他に多数のAPI有り。各APIとその引数仕様についての詳しい説明は、パッケージに含まれるdoc/以下のファイルを参照してください

- doc/Readme.md Doxygenファイル生成方法の説明
- doc/html/index.html Doxygenが生成するWeb資料

PMlibを利用するプログラムの構成例

- Fortran 元ソース例

```
program main

call mykernel()      ! 注目箇所

end
```

- C++ 元ソース例

```
int main(int argc, char *argv[])
{

mykernel();          //注目箇所

return 0;
}
```

- PMlib組み込み後

```
program main
call f_pm_initialize (1)
call f_pm_start ("label")
call mykernel (msize,n,a,b,c)
call f_pm_stop ("label")
call f_pm_report ("")
end
```

PMlib初期化

測定区間

レポート出力

- PMlib組み込み後

```
#include <PerfMonitor.h>
using namespace pm_lib;
PerfMonitor PM;
int main(int argc, char *argv[])
{
PM.initialize();
PM.start("label");
mykernel();
PM.stop ("label");
PM.report(stdout);
return 0;
}
```

PMlibヘッダー

PMlib初期化

測定区間

レポート出力

PMlibの出力情報

- 1、基本レポート
 - 各測定区間の全プロセス平均性能統計値
 - 時間：各区間の平均時間、呼び出し回数、累積経過時間
 - 計算量：呼び出し1回あたりの量、合計量、速度
 - 区間を登録順または経過時間順にソート出力
 - ジョブ全体での総合性能
 - 代表プロセスのHWPCイベント統計量
 - （富岳のみ）マスターノードの消費電力内訳、アプリ全体での合計消費電力*
- 2、詳細レポート
 - 各MPIプロセス毎のプロファイルを出力
 - 各MPIプロセス毎のHWPCイベント統計量
 - 指定プロセス番号のOpenMPスレッド毎のHWPCイベント統計量
- 出力の内容は環境変数で指定可能

基本レポート例

PMLib Basic Report -----

Performance Statistics Report from PMLib version 8.0.0

Linked PMLib supports: MPI, OpenMP, HWPC, PowerAPI, no-OTF on this system

Host name : h31-6108c

Date : 2021/06/16 : 20:03:26

Parallel Mode: Hybrid (4 processes x 12 threads)

The following cotroll variables are provided to PMLib as environment variable.

HWPC_CHOOSER=FLOPS

POWER_CHOOSER=NODE

PMLIB_REPORT=BASIC

Active PMLib elapse time (from initialize to report/print) = 1.324e-01 [sec]

Exclusive sections and inclusive sections are reported below.

Inclusive sections, marked with (*), are not added in the statistics total.

Section Label	number of calls	averaged process execution time [sec]				hardware counted floating point ops.			
		total	[%]	total/call	sdv	f.p.ops	sdv	performance	
stream_check(*) :	1	1.319e-01	99.59	1.319e-01	3.93e-03	8.000e+08	0.00e+00	6.06 Gflops	(*)
sub3_add :	10	3.213e-02	24.26	3.213e-03	1.13e-04	2.000e+08	0.00e+00	6.23 Gflops	
sub4_triad :	10	3.211e-02	24.24	3.211e-03	1.28e-04	4.000e+08	0.00e+00	12.46 Gflops	
sub2_scale :	10	2.471e-02	18.66	2.471e-03	1.41e-04	2.000e+08	0.00e+00	8.09 Gflops	
sub1_copy :	10	2.459e-02	18.56	2.459e-03	8.49e-05	1.450e+03	0.00e+00	0.06 Mflops	
All sections combined		1.135e-01	-Exclusive HWPC sections-			8.000e+08		7.05 Gflops	
Total of all processes		1.135e-01	-Exclusive HWPC sections-			3.200e+09		28.19 Gflops	

基本レポート例（続き）

PMLib hardware performance counter (HWPC) report of the averaged process -----

Report for option HWPC_CHOOSER=FLOPS is generated.

Section		SP_OPS	DP_OPS	Total_FP	[Flops]	[%Peak]
stream_check(*)	:	0.000e+00	8.000e+08	8.000e+08	6.069e+09	7.902e-01
sub3_add	:	0.000e+00	2.000e+08	2.000e+08	6.225e+09	8.106e-01
sub4_triad	:	0.000e+00	4.000e+08	4.000e+08	1.246e+10	1.622e+00
sub2_scale	:	0.000e+00	2.000e+08	2.000e+08	8.094e+09	1.054e+00
sub1_copy	:	0.000e+00	1.450e+03	1.450e+03	5.898e+04	7.679e-06

PMLib Power Consumption report per node basis -----

Report of the master node is generated for POWER_CHOOSER=NODE option.

Section	Estimated power inside node [W]					Energy[Wh]
	total	CMG+L2	MEMORY	TF+A+U	P.meter	
stream_check(*)	: 157.96	108.07	44.15	9.58	152.32	5.58e-03
sub3_add	: 163.69	109.73	47.93	9.88	165.10	1.47e-03
sub4_triad	: 155.84	109.04	46.04	9.88	164.52	1.47e-03
sub2_scale	: 161.07	106.25	46.93	9.42	151.70	1.04e-03
sub1_copy	: 160.74	110.89	46.96	9.94	152.08	1.04e-03

new feature to produce
power consumption report

The aggregate power consumption of 4 processes on 1 nodes = 1.99e+01 [J] == 5.54e-03 [Wh]

電力レポートの細かさ指定 (環境変数)

Report for POWER_CHOOSER=**NUMA** option.

Section	Estimated power inside node [W]										TF+A+U	P. meter	Energy [Wh]
	total	CMGOL2	CMG1L2	CMG2L2	CMG3L2	MEM0	MEM1	MEM2	MEM3				
stream_check (*) :	158.71	28.84	28.75	28.84	28.69	9.40	9.39	9.36	9.32	10.17	154.23	7.03e-03	
sub3_add :	149.38	25.91	25.88	25.88	25.93	11.58	11.57	11.71	7.30	9.45	136.53	1.22e-03	
sub4_triad :	155.91	26.54	26.64	26.67	26.97	12.56	12.55	12.50	8.65	10.07	159.93	1.42e-03	
sub2_scale :	168.65	29.52	29.53	29.53	29.11	12.40	12.35	12.23	8.04	10.33	117.35	8.02e-04	
sub1_copy :	144.72	24.90	24.87	24.87	25.05	10.59	11.04	11.16	8.14	8.99	150.26	1.03e-03	

The aggregate power consumption of 4 processes on 1 nodes = 2.36e+01 [J] == 6.55e-03 [Wh]

Report for POWER_CHOOSER=**PARTS** option.

Section	Estimated power inside node [W]																			P. meter	Energy [Wh]
	total	CMG0	CMG1	CMG2	CMG3	L2CMG0	L2CMG1	L2CMG2	L2CMG3	Acore0	Acore1	TofuD	UnCMG	MEM0	MEM1	MEM2	MEM3	PCI	TofuOpt		
stream_check(*)	158.08	23.37	23.33	23.38	23.46	3.61	3.59	3.60	3.60	0.72	0.58	0.31	5.28	11.17	11.11	11.12	11.12	0.00	2.64	153.27	5.58e-03
sub4_triad	157.56	22.75	23.24	22.78	22.78	3.90	3.70	3.81	3.82	0.74	0.59	0.32	5.38	12.09	10.97	12.06	12.10	0.00	2.69	141.19	1.26e-03
sub3_add	158.25	22.93	22.81	22.95	22.98	3.67	3.75	3.79	3.80	0.73	0.59	0.32	5.39	11.99	11.24	11.71	11.73	0.00	2.61	140.17	1.25e-03
sub2_scale	160.70	23.62	23.47	23.62	23.64	3.68	3.76	3.67	3.72	0.74	0.61	0.33	5.54	11.76	12.41	11.70	11.83	0.00	2.87	121.25	8.30e-04
sub1_copy	166.39	25.25	25.30	25.27	25.36	4.01	4.03	3.83	3.81	0.76	0.61	0.33	5.33	11.53	11.81	11.03	11.04	0.00	2.67	182.37	1.25e-03

total

CMG0

CMG1

CMG2

CMG3

L2CMG0

L2CMG1

L2CMG2

L2CMG3

Acore0

Acore1

TofuD

UnCMG

MEM0

MEM1

MEM2

MEM3

PCI

TofuOpt

| P.meter

| Energy [Wh]

HWPCレポートの種類

HWPC_CHOOSER=FLOPS environment variable is provided.

Header	ID :	SP_OPS	DP_OPS	Total_FP	[Flops]	[%Peak]
Rank	0 :	0.000e+00	1.201e+10	1.201e+10	8.847e+08	1.152e-01
Rank	1 :	0.000e+00	1.201e+10	1.201e+10	8.847e+08	1.152e-01

HWPC_CHOOSER=BANDWIDTH environment variable is provided.

Header	ID :	CMG_bus_RD	CMG_bus_WR	RD [Bytes]	WD [Bytes]	Mem [B/s]	[Bytes]
Rank	0 :	1.880e+08	2.279e+05	4.813e+10	5.834e+07	3.551e+09	4.819e+10
Rank	1 :	1.879e+08	2.079e+05	4.809e+10	5.321e+07	3.547e+09	4.814e+10

HWPC_CHOOSER=VECTOR environment variable is provided.

Header	ID :	DP_SVE_op	DP_FIX_op	SP_SVE_op	SP_FIX_op	Total_FP	Vector_FP	[Vector %]
Rank	0 :	0.000e+00	1.201e+10	0.000e+00	0.000e+00	1.201e+10	0.000e+00	0.000e+00
Rank	1 :	0.000e+00	1.201e+10	0.000e+00	0.000e+00	1.201e+10	0.000e+00	0.000e+00

HWPC_CHOOSER=CACHE environment variable is provided.

Header	ID :	LOAD_INS	STORE_INS	L1_HIT	L1_TCM	L2_TCM	[L1\$ hit%]	[L2\$ hit%]	[L*\$ hit%]
Rank	0 :	1.201e+10	4.933e+06	1.185e+10	1.878e+08	1.877e+08	9.844e+01	4.536e-04	9.844e+01
Rank	1 :	1.201e+10	4.933e+06	1.185e+10	1.878e+08	1.877e+08	9.844e+01	4.657e-04	9.844e+01

HWPC_CHOOSER=CYCLE environment variable is provided.

Header	ID :	TOT_CYC	TOT_INS	FP_inst	FMA_inst	[FMA_ins%]	[Ins/cyc]
Rank	0 :	2.714e+10	3.307e+10	9.003e+09	3.003e+09	3.336e+01	1.015e-01
Rank	1 :	2.714e+10	3.307e+10	9.003e+09	3.003e+09	3.336e+01	1.015e-01

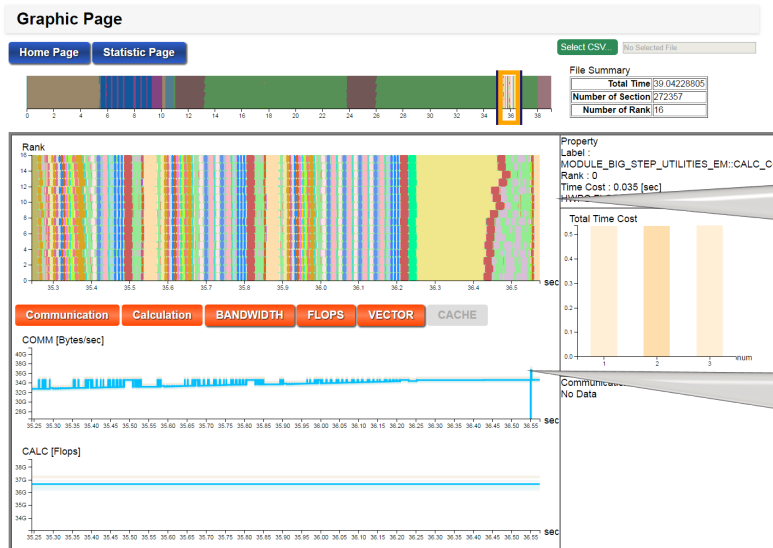
HWPC_CHOOSER=LOADSTORE environment variable is provided.

Header	ID :	LOAD_INS	STORE_INS	SVE_LOAD	SVE_STORE	SVE_SMV_LD	SVE_SMV_ST	GATHER_LD	SCATTER_ST	[Vector %]
Rank	0 :	1.201e+10	4.935e+06	1.200e+10	3.003e+06	0.000e+00	0.000e+00	0.000e+00	0.000e+00	9.997e+01
Rank	1 :	1.201e+10	4.933e+06	1.200e+10	3.003e+06	0.000e+00	0.000e+00	0.000e+00	0.000e+00	9.997e+01

出力のオプション

- 詳細プロファイルテキストレポート
 - プロセス毎・スレッド毎の計算量・システムの実効性能諸値
 - アプリケーションと一体で利用しやすい
- OTF(Open Trace Format)ファイル出力
 - 専用のポスト処理プログラムTRAiLを用いてWebブラウザで表示
 - 計算性能を時刻歴に可視化する機能

今のところHWPCデータにのみ対応
見た目はかわいらしくなるが、実用面
ではテキストの方が使いやすい



ランク毎処理時刻グラフ
横軸: 時刻
縦軸: ランク番号
実行されているラベルを
前面に表示

性能グラフ
横軸: 時刻
縦軸: 性能測定値
ランク毎に1つの曲線を表示