Performance Monitor Library

Ver. 2.2

Advanced Visualization Research Team
Advanced Institute for Computational Science
RIKEN

7-1-26, Minatojima-minami-machi, Chuo-ku, Kobe, Hyogo, 650-0047, Japan

http://aics.riken.jp/

October 2014

Release

Version 2.2	2014-10-30
Version 1.9.3	2013-06-27
Version 1.9.2	2013-06-26
Version 1.9.1	2013-06-25
Version 1.9	2013-06-14
Version 1.8	2013-06-13
Version 1.7	2013-05-08
Version 1.6	2013-04-13
Version 1.5	2012-12-05
Version 1.4	2012-09-06
Version 1.3	2012-07-23
Version 1.2	2012-07-11
Version 1.1	2012-05-02
Version 1.0	2012-04-28

COPYRIGHT

目次

第 1 章

計算性能モニタクラス

本章では，実行時の計算性能を測定するモジュール機能について説明する．

1.1　計算性能測定機能

計算性能測定機能は，プログラム実行時にプログラムコード内の測定対象部分の実行時間と計算量を測定し，その集計・統計情報を出力する機能を提供する．

1.1.1　測定対象と測定方法

測定する対象は時間と計算量である。時間および計算量についての定義は以降に説明される。各測定箇所はラベル名により識別され，次のプロパティを持つ．

ラベル　測定する箇所につけるラベル文字列。
測定対象タイプ　「計算時間」、「通信時間」、「自動決定」
排他測定フラグ　「排他測定」または「非排他測定」

■測定対象タイプ　時間は各測定箇所の処理を行うのに必要な経過時間を意味する。測定結果の表示分類のため、処理の種類が主として演算に対応する場合は「計算時間」、主として通信に対応する場合は「通信時間」としてタイプを設定し、処理のボリュームである計算量（演算量あるいは通信量）とともに集計される。

計算量を測定する方法には、ユーザーが計算量を明示的に自己申告する方法と、PMlib 内部で後述する HWPC(hardware performance counter) を用いて自動的に取得測定する方法とがある。

ユーザーが計算量を明示的に自己申告する場合は、測定対象タイプとして「計算時間」または「通信時間」のいずれかを指定して申告する。このとき申告される量は「計算時間」を指定した場合は，浮動小数点演算量であると解釈され，統計情報出力時に計算速度 (FLOPS 値) が出力される．「通信時間」を指定した場合は，バイト単位の通信量であると解釈され，統計情報出力時に通信速度 (Byte/s 単位) が出力される．ユーザープログラムで実際に申告するには、測定を終了する際に呼ばれる stop メソッドへの引数としてその値を与える。

計算量を PMlib 内部で自動測定する場合は測定対象タイプを「自動決定」と指定する。測定対象タイプが「自動決定」の場合様々なタイプの統計値を計算量として取得・表示する事ができ、環境変数 HWPC_CHOOSER の値によって選ぶ事になる。

環境変数 HWPC_CHOOSER が選択可能な値は以下である。

- FLOPS 浮動小数点演算
- VECTOR ベクトル命令
- BANDWIDTH バンド幅（メモリ・キャッシュ）
- CACHE キャッシュ階層のヒット・ミス

■排他測定と非排他測定　測定箇所は，排他測定フラグの真偽により，「排他測定」と「非排他測定」とに分類される．排他測定は，「(ほぼ) 完成されたプログラムの実行時の状況を把握するために，プログラムコードを排他的な領域に分割し，各領域の実行時間を測定する」という使用方法を想定している．排他測定では，統計情報出力時に，「全排他測定箇所の合計実行時間」に対する「個々の排他測定箇所の実行時間」の割合も出力する．

一方，非排他測定は，プログラムのデバッグあるいはチューニング時に，一時的に興味のある対象領域の実行時間を把握するのに使用することを想定している．そのため，非排他測定区間は，排他測定区間や他の非排他測定区間と重ねることも可能で，自由に設置できる．

■並列実行時の制限　並列実行時には，それぞれの排他測定区間は，並列計算ノードによらず同一回数だけ実行されるものと仮定している．計算ノード毎に実行回数が異なっていた場合には，後述する print メソッド実行時には，該当区間の統計情報は出力されない．

一方，非排他測定区間の設置には制限はなく，並列計算ノード毎に実行回数が異なっていても問題ない．非排他測定

区間の測定結果の出力には printDetail メソッドを使用する.

1.1.2　API 説明

計算性能測定機能は，個々の測定箇所毎に時間測定を担当する PerfWatch クラスと，全 PerfWatch を管理して統計情報の計算と出力を担当する PerfMonitor クラスにより提供される.

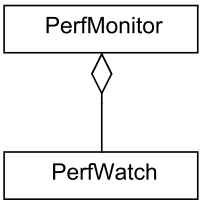

図 1.1　計算性能測定機能 クラス図

ユーザーがプログラムコードから利用するのは PerfMonitor クラスである。PerfWatch クラスは全て PerfMonitor クラスをとおして生成され操作されるので，プログラムコードから直接 PerfWatch クラスの API が呼ばれることはない.

PerfMonitor クラス公開メソッド

■初期化

引数はオプションで測定区間数 nWatch を指定する。測定区間数があらかじめわかっている場合はその値を引数として指定することが望ましい。測定区間数 (nWatch) + 1 個の測定時計 (PerfWatch クラス) をインスタンスして，全計算時間用測定時計を準備させる.

区間数が不明な場合は引数なしで呼び出すことが可能。その場合、PMlib はデフォルトの区間数を準備し、後に必要に応じて動的に区間数を増加させる。

```
void initialize(unsigned nWatch)
```

■測定区間にプロパティを設定

第 1 、第 2 引数は必須。第 3 引数はオプション。第 1 引数のラベル文字列 label で識別される測定区間に，第 2 引数の測定対象タイプ type と, 第 3 引数の排他測定フラグ (true=排他測定/false=非排他測定) を設定する。

```
void setProperties(string& label, Type type, bool exclusive=true)
```

なお測定対象タイプ type の指定には以下の定数を一つ選んで使用する。

通信　PerfMonitor::COMM
計算　PerfMonitor::CALC
自動決定　PerfMonitor::AUTO

■並列処理のタイプを設定

第 1 引数 parallel_mode で並列処理のタイプを設定"Serial", "OpenMP", "FlatMPI", "Hybrid" 第 2 引数 num_threads はスレッド数、第 3 引数 npes は MPI プロセス数

```
void setParallelMode(string& parallel_mode, int num_threads, int npes)
```

■測定スタート
ラベル label で識別される測定区間の始まりを指定する．第 1 引数はラベル label。

```
void start(string label)
```

■測定ストップ
ラベル label で識別される測定区間の終わりを指定する．第 1 引数はラベル label。第 2 引数は計算量 flopPerTask で、「タスク」あたりの演算量 (Flop) または通信量 (バイト) 第 3 引数は実行「タスク」数 iterationCount 第 2、第 3 引数は省略可。

```
void stop(string label, double flopPerTask=0.0, unsigned iterationCount=1)
```

■測定結果の集約
測定結果情報をノード 0 に集約し，統計情報を計算する．また，初期化時にスタートさせた全計算時間用測定をストップさせる．

```
void gather()
```

■測定結果の出力
排他測定区間についての統計情報を出力する．ノード 0 以外では, 呼び出されてもなにもしない．

```
void print(FILE* fp)
```

■詳細な測定結果の出力
非排他測定区間も含めて，詳細な統計情報を出力する．ノード 0 以外では, 呼び出されてもなにもしない．

```
void printDetail(FILE* fp)
```

1.1.3　PerfMonitor クラスの使用方法

PerfMonitor クラスを使用するにはヘッダファイル PerfMonitor.h をインクルードする．また，並列化時には MPI_Init(), MPI_Finalize() はユーザプログラムの管理で行うこと．

■初期化・区間の登録

1. PerfMonitor インスタンス化
2. initialize メソッドによる測定区間数の登録
3. setRankInfo メソッドによる並列ランク情報の設定
4. setProperties メソッドによる各測定区間へのプロパティの設定

初期化の例

```
int my_rank; // 自身のランク番号
…
// (1) PerfMonitor インスタンス化
PerfMonitor PM
…
// (2) 測定区間を初期登録
PM.initialize();
```

```
 …
// (3) 並列情報設定
PM.setRankInfo(my_rank);
…
// (4) 各測定区間にプロパティを設定
PM.setProperties("SEC1", PerfMonitor::CALC);        //計算, 排他
PM.setProperties("SEC2", PerfMonitor::COMM);        //通信, 排他
PM.setProperties("SEC3", PerfMonitor::AUTO);        //自動決定, 排他
…
```

■測定区間の指定　測定対象タイプ type を自動決定 (AUTO) と指定する場合は、測定対象区間を同一ラベル label を指定して start メソッドと stop メソッドで挟むだけでよい。

```
PM.setProperties(label1, PerfMonitor::AUTO);
PM.start(label1);
/* 測定対象区間 */
PM.stop(label1);
```

測定対象タイプ type を計算 (CALC) または通信 (COMM) と指定する場合は、stop メソッドに「タスク」あたりの計算量 flopPerTask と「タスク」の数 iterationCount （反復数など）も指定することが望ましい。

以下の例では，測定対象区間内で浮動小数点演算量 flop のブロック（タスク）を nStep 回実行したことを stop メソッドで明示的に申告している。

```
PM.setProperties(label1, PerfMonitor::CALC);
PM.start(label1);
for (int i = 0; i < nStep; i++) {
  /* このブロックの浮動小数計算量が flop */
}
PM.stop(label1, flop, nStep);
```

測定対象タイプ type を計算 (CALC) または通信 (COMM) と指定して、stop メソッドに計算量を指定しない場合は、環境変数 HWPC_CHOOSER の指定と矛盾が無い測定対象タイプに限って測定・出力が可能である。

■測定結果の集計・出力

測定結果出力メソッドを呼び出す前に，全ノードで集計メソッド gather を呼び出す必要がある．出力メソッド print および printDetail は，ノード 0 から呼ばれた時のみ指定されたファイルへ測定結果情報を出力する．

```
// 測定結果の集計 (gathre メソッドは全ノードで呼ぶ)
PM.gather();
// コンソールへの測定結果の出力 (ノード 0 が出力。ノード 0 以外は何もしない)
PM.print(stdout);
PM.printDetail(stdout);

if (p.ID == 0) {
  FILE* fp = fopen("timing.txt", "w");
  // ファイルへの測定結果の出力 (ノード 0 のみからの呼び出しも OK)
  PM.print(fp);
  PM.printDetail(fp);
  fclose(fp);
}
```

1.1.4　統計情報出力内容

■print メソッド出力内容

print メソッドは時間および計算量の測定結果を表示する。プログラムが MPI 並列化されている場合は全プロセスの平均値を表示する。各プロセスがスレッド並列化されている場合は当プロセスに帰属するスレッドの計算量が合算されて

当プロセスの値となる。

「Report of Timing Statistics」に続いて以下の出力

Total execution time
: ノード 0 における，initialize メソッド呼び出しから gather メソッド呼び出しまでの時間

Parallel Mode
: 並列実行モードと，そのプロセス，スレッド数

Total time of measured sections
: 全排他測定区間における「積算実行時間のノード間平均」の合計

「Statistics per MPI process [Node Average]」に続いて，各排他測定区間毎に以下の出力

call
: 測定回数 (「start メソッド〜stop メソッド」ペアの呼び出し回数)

accumulated time
: 積算実行時間 (avr[sec] はノード間平均値，avr[%] は「Total time of measured sections」値に対する平均値の割合，sdv[sec] は標準偏差，avr/call[sec] は 1 測定あたりの平均値)

flop | messages[Bytes]
: 積算浮動小数演算量または通信量 (avr はノード間平均値，speed は積算実行時間のノード間平均値より求めた計算速度または通信速度)

なお，排他測定区間における測定回数がノード毎に異なっていた場合には，その排他測定区間に関する統計情報の計算は行われず，「NA」と出力される．

print メソッドタイプ出力例

```
#PMlib_report_section Basic Report type -------------

Report of Timing Statistics PMlib version 2.2
Operator  : user
Host name : vsp20
Date      : 2014/10/29 : 16:53:47

Parallel Mode                    :   Hybrid (4 processes x 8 threads)

Total execution time            = 5.703156e+00 [sec]
Total time of measured sections = 4.556153e+00 [sec]

Statistics per MPI process [Node Average]. Only the exclusive sections are listed.
Label               |  call  |        accumulated time[sec]           | [flop_counts or message_bytes ]
                    |        |   avr     avr[%]    sdv     avr/call   |   avr       sdv      speed
--------------------+--------+----------------------------------------+---------------------------
Poisson_SOR2_SMA    :  26200   1.484e+00  32.56  6.90e-02  5.662e-05    2.318e+10  0.00e+00  15.63 Gflops
Poisson_Src_Norm    :  13300   8.590e-01  18.85  1.45e-02  6.459e-05    4.112e+10  0.00e+00  47.87 Gflops
File_Output         :     11   8.061e-01  17.69  1.29e-02  7.328e-02    1.311e+07  0.00e+00  16.26 Mflops
Sync_Pressure       :  26200   6.942e-01  15.24  9.75e-02  2.650e-05    1.717e+09  0.00e+00   2.47 GB/sec
A_R_Poisson_Src     :  13300   2.289e-01   5.02  2.48e-02  1.721e-05    8.512e+05  0.00e+00   3.72 MB/sec
Poisson_BC          :  26200   1.468e-01   3.22  5.50e-03  5.601e-06    0.000e+00  0.00e+00   0.00 Mflops
Pseudo_Velocity     :    100   7.701e-02   1.69  5.56e-04  7.701e-04    5.230e+09  0.00e+00  67.91 Gflops
Projection_Velocity :    262   7.458e-02   1.64  4.62e-04  2.847e-04    1.820e+09  0.00e+00  24.40 Gflops
Poisson_Setup_for_I :  13100   7.206e-02   1.58  2.60e-03  5.501e-06    0.000e+00  0.00e+00   0.00 Mflops
Projection_Velocity :    262   1.307e-02   0.29  5.44e-04  4.988e-05    0.000e+00  0.00e+00   0.00 Mflops
Sync_Velocity       :    100   1.284e-02   0.28  4.23e-05  1.284e-04    1.573e+08  0.00e+00  12.25 GB/sec
Copy_Array          :    200   1.201e-02   0.26  1.53e-03  6.006e-05    0.000e+00  0.00e+00   0.00 Mflops
Variation_Space     :    100   1.094e-02   0.24  1.29e-04  1.094e-04    4.260e+08  0.00e+00  38.94 Gflops
Sync_Pseudo_Velocit :    100   1.000e-02   0.22  2.02e-03  1.000e-04    3.932e+07  0.00e+00   3.93 GB/sec
Divergence_of_Pvec  :    100   6.305e-03   0.14  2.78e-04  6.305e-05    2.425e+08  0.00e+00  38.46 Gflops

--------------------+--------+----------------------------------------+---------------------------
Process Total       |          4.556e+00                                  7.225e+10            15.86 Gflops
```

```
Aggregate performance of the job                                    63.43 Gflops
```

■printDetail **メソッド出力内容**　各測定区間，各ノード毎に，２種類の統計情報（タイプ A, タイプ B）を出力する

time[s]
: 積算実行時間

time[%]
: 積算実行時間の「Total time of measured sections」値に対する割合

t_wait[s]
: 待ち時間 (「積算実行時間のノード間最大値」と「積算実行時間」の差)

t[s]/call
: 1 測定あたりの実行時間

flop|msg
: 積算浮動小数演算量または通信量 (バイト単位)

speed
: 積算実行時間より求めた計算速度または通信速度

なお，非排他測定区間については，ラベル文字列をその前に「*」をつけて表示する．

printDetail **メソッドタイプ** A **出力例**

各測定区間のラベル行、Header 行に続いて、測定対象タイプ type で指定した測定値が各プロセス毎に出力される。その測定値の平均値が上記 print メソッドの出力値である。

```
#PMlib_report_section Detail Report A ------------

    Elapsed time variation over MPI ranks

Label  Allocate_Arrays
Header ID  :     call   time[s] time[%]  t_wait[s]  t[s]/call   flop|msg    speed  (speed unit)
Rank     0 :        4  4.712e-03    0.1  7.200e-05  1.178e-03  0.000e+00  0.000e+00 Flops
Rank     1 :        4  4.784e-03    0.1  0.000e+00  1.196e-03  0.000e+00  0.000e+00 Flops
Rank     2 :        4  4.769e-03    0.1  1.550e-05  1.192e-03  0.000e+00  0.000e+00 Flops
Rank     3 :        4  4.779e-03    0.1  5.007e-06  1.195e-03  0.000e+00  0.000e+00 Flops
Label  Restart_Process
Header ID  :     call   time[s] time[%]  t_wait[s]  t[s]/call   flop|msg    speed  (speed unit)
Rank     0 :        1  8.106e-06    0.0  0.000e+00  8.106e-06  0.000e+00  0.000e+00 Flops
Rank     1 :        1  8.106e-06    0.0  0.000e+00  8.106e-06  0.000e+00  0.000e+00 Flops
Rank     2 :        1  7.153e-06    0.0  9.537e-07  7.153e-06  0.000e+00  0.000e+00 Flops
Rank     3 :        1  8.106e-06    0.0  0.000e+00  8.106e-06  0.000e+00  0.000e+00 Flops
Label  Search_Vmax
Header ID  :     call   time[s] time[%]  t_wait[s]  t[s]/call   flop|msg    speed  (speed unit)
Rank     0 :      100  2.120e-03    0.0  3.576e-05  2.120e-05  5.898e+07  2.782e+10 Flops
Rank     1 :      100  2.156e-03    0.0  0.000e+00  2.156e-05  5.898e+07  2.736e+10 Flops
Rank     2 :      100  2.135e-03    0.0  2.098e-05  2.135e-05  5.898e+07  2.763e+10 Flops
Rank     3 :      100  2.078e-03    0.0  7.749e-05  2.078e-05  5.898e+07  2.838e+10 Flops
Label  A_R_Vmax
Header ID  :     call   time[s] time[%]  t_wait[s]  t[s]/call   flop|msg    speed  (speed unit)
Rank     0 :      100  1.371e-03    0.0  7.551e-04  1.371e-05  3.200e+03  2.333e+06 Bytes/sec
Rank     1 :      100  1.981e-03    0.0  1.452e-04  1.981e-05  3.200e+03  1.615e+06 Bytes/sec
Rank     2 :      100  1.819e-03    0.0  3.076e-04  1.819e-05  3.200e+03  1.759e+06 Bytes/sec
Rank     3 :      100  2.126e-03    0.0  0.000e+00  2.126e-05  3.200e+03  1.505e+06 Bytes/sec
```

printDetail メソッドタイプ B 出力例

測定対象タイプ type が自動決定 (AUTO) と指定され、かつ環境変数 HWPC_CHOOSER の値が指定されている場合は、HWPC_CHOOSER の値に応じて HWPC が自動計測した内容が出力される。各測定区間のラベル行、Header 行に続いて、各プロセス毎の HWPC 測定値および統計値が出力される。

HWPC_CHOOSER=FLOPS を指定した場合

```
    Hardware performance counter (HWPC) statistics were collected with HWPC_CHOOSER=FLOPS
    PMlib detected the CPU architecture:
        GenuineIntel
        Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz
    The available Hardware Performance Counter (HWPC) events depend on this CPU architecture.
    HWPC event values of each MPI process are shown below. sum of threads.

Label   Allocate_Arrays
Header  ID :     SP_OPS     DP_OPS    [Flops]
Rank     0 :  6.593e+05  1.740e+02  1.237e+07
Rank     1 :  6.621e+05  1.590e+02  1.375e+07
Rank     2 :  6.637e+05  1.610e+02  1.228e+08
Rank     3 :  6.636e+05  2.100e+02  1.352e+08
```

HWPC_CHOOSER=BANDWIDTH を指定した場合

```
Label   Allocate_Arrays
Header  ID :     LD_INS     SR_INS     L1_HIT    HIT_LFB L2_DRD_REQ L2_DRD_HIT L2_PF_MISS L2_RFO_MIS   [HW B/s]
Rank     0 :  7.624e+08  1.329e+08  7.619e+08  2.080e+05  3.298e+05  2.614e+05  2.721e+05  3.233e+04  5.273e+08
Rank     1 :  8.604e+08  1.383e+08  8.599e+08  2.280e+05  3.259e+05  2.586e+05  2.426e+05  2.214e+04  4.921e+08
Rank     2 :  7.830e+08  1.216e+08  7.825e+08  2.244e+05  3.291e+05  2.614e+05  2.413e+05  2.132e+04  3.853e+08
Rank     3 :  9.371e+08  1.557e+08  9.366e+08  2.240e+05  3.293e+05  2.613e+05  2.384e+05  2.182e+04  4.371e+08
```

HWPC_CHOOSER=CACHE を指定した場合

```
Label   Allocate_Arrays
Header  ID :     L1_TCM     L2_TCM     L3_TCM    OFFCORE
Rank     0 :  5.115e+05  1.429e+05  1.341e+04  2.253e+05
Rank     1 :  4.726e+05  1.288e+05  1.264e+04  2.249e+05
Rank     2 :  4.670e+05  1.288e+05  1.157e+04  2.345e+05
Rank     3 :  4.701e+05  1.351e+05  2.545e+04  2.030e+05
```

Header ID の行で用いられるシンボルの内容

```
HWPC events legend:
    FP_OPS: floating point operations
    SP_OPS: single precision floating point operations
    DP_OPS: double precision floating point operations
    VEC_SP: single precision vector floating point operations
    VEC_DP: double precision vector floating point operations
    LD_INS: memory load instructions
    SR_INS: memory store instructions
    L1_HIT: level 1 cache hit
    L2_HIT: level 2 cache hit
    L3_HIT: level 3 cache hit
    HIT_LFB cache line fill buffer hit
    L1_TCM: level 1 cache miss
    L2_TCM: level 2 cache miss
    L3_TCM: level 3 cache miss by demand
    OFFCORE: demand and prefetch request cache miss
    TOT_CYC: total cycles
```

```
    TOT_INS: total instructions
    FP_INS: floating point instructions
Derived statistics:
    [GFlops]: floating point operations per nano seconds (10^-9)
    [Mem GB/s]: memory bandwidth in load+store GB/s
    [L1$ %]: Level 1 cache hit percentage
    [LL$ %]: Last Level cache hit percentage
```

要確認 シンボルはさらに整理が必要