Profiling d'applications parallèles

Principes, méthodes et outils

Anne Cadiou

Laboratoire de Mécanique des Fluides et d'Acoustique

Séminaires et ateliers pour le calcul scientifique
PMCS2I - LMFA
Vendredi 3 mai 2019

CC BY SA

Définition

Wikidépia

Analyser l'exécution d'une application parallèle, afin d'en connaître son comportement, d'évaluer les parties à optimiser

Objectifs :

- Connaître la consommation en temps et mémoire d'un code
- Optimiser le temps de calcul passé dans les différentes parties du code
- Évaluer la scalabilité (en temps effectif)
- Réduire la consommation mémoire
- Évaluer la performance des I/O
- Évaluer le comportement des communications

Sert à comprendre le comportement d'un code sur une architecture donnée, dans un environnement donné, afin d'exploiter au mieux toutes ses caractéristiques

Questions associées

- Mon code utilise-t'il efficacement les ressources CPU ?
- Quelles sont les caractéristiques parallèles de mon programme ?
- Combien de mémoire effective utilise mon programme ?
- La gestion des IO est-elle efficace ?

Caractéristiques liées au caractère parallèle

Facteurs associés au comportement

séquentiel

- Temps de calcul (CPU time, flops)
- Utilisation des caches et de la bande passante mémoire
- Lecture et écriture sur disque (par un processus)

parallèle

- Équilibrage de charges
- Coût des communications (MPI), de l'ordonnanceur (OMP)
- Coût des synchronisations, des opérations de réductions
- I/O parallèles
- Influence du mapping (MPI), de la gestion du cache (OMP)

Différentes approches d'analyse de performance

Échantillonnage

- l'exécution du programme est suspendue périodiquement pour effectuer des mesures
- inférence statistique du comportement du programme
- fonctionne sans modification particulière du programme

Instrumentation

- des capteurs sont insérés dans le code pour obtenir des informations détaillées
- biais important sur les petites fonctions
- nécessite la modification du code source

Profiling et trace

Profiling

- obtention de temps d'exécution (statistiques, génération de résumés)
- souvent basé sur de l'échantillonnage

Tracing

- obtention de cartographies, de chronologies (connexions, génération d'un journal de bord)
- souvent basé sur de l'instrumentation

Différents types de sorties

- Profils plats
  - temps CPU passé dans chaque fonction
  - nombre de fois où une fonction est appelée
  - ⇒ utile pour connaître les fonctions les plus coûteuses
- Graphes d'appels
  - nombre de fois où une fonction est appelée par d'autres
  - nombre de fois où une fonction en appelle d'autres
  - ⇒ utile pour identifier la relation entre les fonctions
  - ⇒ suggère la suppression de certains appels
- Sources annotées
  - indique le nombre de fois où une ligne est exécutée

Protocoles de mesure

PAPI — http://icl.cs.utk.edu/papi

- interface multiplateforme de compteurs hardware (opérations et cycles, accès mémoire et caches)
- utilisée par TAU, SCALASCA, ompP, mpiP

Score-P — http://www.score-p.org

- bibliothèque de profiling et trace pour les applications HPC
- utilisée par TAU, SCALASCA, Vampir, Periscope

Extrae — http://tools.bsc.es/extrae

- bibliothèque d'instrumentation
- utilisée par Paraver

Outils de base

Pour débuter

- gprof

  `https://sourceware.org/binutils/docs/gprof`
- Valgrind (Callgrind, Cachegrind, Massif)

  `http://valgrind.org/`

Outils basiques à l'intérieur d'un noeud

- VTune (commercial)
- ompP

  `http://www.ompp-tool.com`
- mpiP

  `http://mpip.sourceforge.net`
- ...

Outils pour les IO

Darshan `https://www.mcs.anl.gov/research/projects/darshan/`

- Sans recompilation, avec la variable `LD_PRELOAD`
- Analyse les appels à Posix, MPI-IO, HDF5 et NetCDF
- Le code doit avoir un appel à la bibliohèque MPI (e.g. `MPI_Init` et `MPI_Finalize`)
- Compatible `C`, `C++`, `Fortran`

Profiling parallèle

- Intel Advisor, Trace Analyzer and Collector (commercial)
- `PerfSuite`

  `http://perfsuite.ncsa.illinois.edu/`
- `mpiP` (Lightweight, Scalable MPI Profiling)

  `http://mpip.sourceforge.net`
  - résultat sous format texte
  - à ajouter lors de l'édition de liens
- `IPM` (Integrated Performance Monitoring)

  `http://ipm-hpc.sourceforge.net`
  - résultat sous format texte
  - sans recompilation
- Paraver

  `https://tools.bsc.es/paraver/`
  - basé sur Extrae

- HPCToolkit

  `http://www.hpctoolkit.org`
  - basé sur de l'échantillonnage statistique
- MAQAO (Modular Assembly Quality Analyzer and Optimizer)

  `http://www.maqao.org/`
  - dédié à l'analyse intra-noeud
  - sans recompilation
- TAU (Tuning and Analysis Utilities)

  `http://www.cs.uoregon.edu/research/tau`
  - basé sur PAPI
  - séquentiel, parallèle (MPI et multithread)
  - graphique et ligne de commande
- SCALASCA (SCalable performance Analysis of LARgre SCale Applications)

  `http://www.scalasca.org/`
  - basé sur PAPI, exploite `Score-P`
  - séquentiel, parallèle (MPI et multithread)
  - graphique et ligne de commande

Outils de visualisation

- Vampir (commercial) https://vampir.eu/
- CUBE http://www.scalasca.org/
  - utilisé par SCALASCA, Score-P, PerfSuite, Marmot, ompP, etc.
- Paraver http://www.scalasca.org/
  - utilisé par Extrae
- PerfExplore PerfView http://perfsuite.ncsa.illinois.edu/
  - inclus dans PerfSuite
- ...

Méthode

Préparation

- Préparer l'environnement
- Obtenir un cas de référence
- Instrumenter le code le cas échéant

Mesure

- Collecter les données
- Aggréger les informations
- Se focaliser sur une partie (appel, mémoire, IO)

Analyse

- Calculer les métriques
- Identifier les problèmes affectant les performances

Optimisation

- Modifier le code pour résoudre ou diminuer les problèmes

Benchmark NPB-MZ-MPI

NAS Parallel Benchmark (MPI/OpenMP)
https://www.nas.nasa.gov/publications/npb.html

Cas BT-MZ

- Résolution des équations de Navier-Stokes compressibles 3D
- Grille régulière
- Avance de 200 pas de temps (solveur tri-diagonal par blocs)

- NPB : The NAS Parallel Benchmarks
- MZ : multi-zone
- BT : Block Tri-diagonal solver

Préparation du cas test

Obtention du benchmark

```
~$ wget https://www.nas.nasa.gov/assets/npb/NPB3.4-MZ.tar.gz

~$ tar xvfz NPB3.4-MZ.tar.gz

~$ cd NPB3.4-MZ
NPB3.4-MZ$ cd NPB3.4-MZ-MPI
NPB3.4-MZ/NPB3.4-MZ-MPI$ ls
bin  BT-MZ  common  config  LU-MZ  Makefile  README  README.install
     SP-MZ  sys  test_scripts
```

Sélection des options pour le Makefile en fonction du calculateur

```
cp config/NAS.samples/make.def.gcc_mpich config/make.def
make bt-mz CLASS=B NPROCS=4
```

Options de compilation

config/make.def

```
  (...)
#-----------------------------------------------------
# This is the fortran compiler used for fortran programs
#-----------------------------------------------------
FC = mpif90
# This links fortran programs; usually the same as ${FC}
FLINK = $(FC)

#-------------------------------------------------------
# These macros are passed to the linker
#-------------------------------------------------------
F_LIB =

#-------------------------------------------------------
# These macros are passed to the compiler
#-------------------------------------------------------
F_INC =

#-------------------------------------------------------
# Global *compile time* flags for Fortran programs
#-------------------------------------------------------
FFLAGS  = -O3 -fopenmp
  (...)
```

Compilation du benchmark - sans profiling

```
NPB3.4-MZ/NPB3.4-MZ-MPI$ make bt-mz CLASS=B NPROCS=4
   ========================================
   =      NAS PARALLEL BENCHMARKS 3.4      =
   =      MPI+OpenMP Multi-Zone Versions   =
   =      MPI/Fortran                      =
   ========================================

cd BT-MZ; make CLASS=B VERSION=
make[1]: Entering directory 'BT-MZ'
  (...)
../sys/setparams bt-mz B
mpif90 -c  -O3 -fopenmp bt_data.f
mpif90 -c  -O3 -fopenmp mpinpb.f
mpif90 -c  -O3 -fopenmp bt.f
  (...)
mpif90 -O3 -fopenmp -o ../bin/bt-mz.B.x bt.o bt_data.o initialize.o
     exact_solution.o exact_rhs.o set_constants.o adi.o  rhs.o
     zone_setup.o x_solve.o y_solve.o  exch_qbc.o solve_subs.o
     z_solve.o add.o error.o verify.o setup_mpi.o mpinpb.o error_cond
     .o ../common/print_results.o ../common/timers.o
make[1]: Leaving directory 'BT-MZ'
```

Exécutable dans `bin/bt-mz.B.x`

```
ls -lh bin
total 164K
-rwxr-xr-x 1 acadiou acadiou 162K avril 22 20:46 bt-mz.B.x
```

Exécution du benchmark

Environnement et exécution

```
export OMP_NUM_THREADS=2
mpirun -np 2 ../bin/bt-mz.B.x
```

Sortie

```
 NAS Parallel Benchmarks (NPB3.4-MZ MPI+OpenMP) - BT-MZ Benchmark

 Number of zones:   8 x   8
 Total mesh size:   304 x   208 x  17
 Iterations: 200    dt:   0.000300
 Number of active processes:      2

 Use the default load factors
 Total number of threads:      4  (  2.0 threads/process)

 Calculated speedup =      4.00

 Time step    1
 Time step   20
  (...)
 Time step  200
 Verification being performed for class B
 accuracy setting for epsilon =  0.1000000000000E-07
  (...)
 Verification Successful
```

```
 BT-MZ Benchmark Completed.
 Class           =                           B
 Size            =            304x  208x 17
 Iterations      =                        200
 Time in seconds =                      89.60
 Total processes =                          2
 Total threads   =                          4
 Mop/s total     =                    6709.97
 Mop/s/thread    =                    1677.49
 Operation type  =             floating point

 Compile options:
    FC           = mpif90
    FLINK        = $(FC)
    F_LIB        = (none)
    F_INC        = (none)
    FFLAGS       = -O3 -fopenmp
    FLINKFLAGS   = $(FFLAGS)
    RAND         = (none)
  (...)
 npb@nas.nasa.gov
```

Instrument

Recompilation

```
make clean
```

Chargement de l'environnement

```
export PATH="/softs/scorep/bin:$PATH"
export LD_LIBRARY_PATH="/softs/scorep/lib:$LD_LIBRARY_PATH"
export PATH="/softs/scalasca-2.5/bin:$PATH"
export LD_LIBRARY_PATH="/softs/scalasca-2.5/lib:$LD_LIBRARY_PATH"
```

Modification du Makefile

```
  (...)
#-------------------------------------------------------
# This is the fortran compiler used for fortran programs
#-------------------------------------------------------
FC = $(PREP) mpif90
# This links fortran programs; usually the same as ${FC}
FLINK = $(FC)
```

```
make bt-mz PREP=scorep CLASS=B NPROCS=4
```

```
ls -lh bin
total 7,9M
-rwxr-xr-x 1 acadiou acadiou 7,9M avril 22 20:47 bt-mz.B.x
```

Ré-exécution du benchmark

```
export OMP_NUM_THREADS=2
mpirun -np 2 ../bin/bt-mz.B.x
```

Sortie

```
 NAS Parallel Benchmarks (NPB3.4-MZ MPI+OpenMP) - BT-MZ Benchmark

 Number of zones:   8 x   8
 Total mesh size:   304 x   208 x  17
 Iterations: 200    dt:   0.000300
 Number of active processes:      2

 Use the default load factors
 Total number of threads:      4  (  2.0 threads/process)

 Calculated speedup =      4.00

 Time step    1
 Time step   20
  (...)
 Time step  200
 Verification being performed for class B
 accuracy setting for epsilon =  0.1000000000000E-07
  (...)
 Verification Successful
```

Coût supplémentaire en temps (overhead) lié à l'intrumentation :

```
 BT-MZ Benchmark Completed.
  (...)
 Time in seconds =                    100.58
  (...)
 npb@nas.nasa.gov
```

Génération des fichiers de mesure

```
$ ls -1 scorep-bt-mz/
MANIFEST.md
profile.cubex
scorep.cfg
```

Rapport sous forme de texte

```
$ scorep-score scorep-bt-mz/profile.cubex

Estimated aggregate size of event trace:                   50MB
Estimated requirements for largest trace buffer (max_buf): 25MB
Estimated memory requirements (SCOREP_TOTAL_MEMORY):       29MB
(hint: When tracing set SCOREP_TOTAL_MEMORY=29MB to avoid intermediate flushes
     or reduce requirements using USR regions filters.)

flt     type max_buf[B]    visits time[s] time[%] time/visit[us]  region
         ALL 26,089,663 1,218,400  404.18   100.0         331.73  ALL
         OMP 26,047,552 1,217,152  401.63    99.4         329.98  OMP
         MPI     42,070     1,246    1.91     0.5        1529.89  MPI
```

Détail par fonction

```
$ scorep-score -r scorep-bt-mz/profile.cubex

 (...)
   OMP  2,238,336    51,456    0.02     0.0     0.35  !$omp parallel @exch_qbc.f
        :217
   OMP  2,238,336    51,456    0.02     0.0     0.33  !$omp parallel @exch_qbc.f
        :206
   OMP  2,238,336    51,456    0.02     0.0     0.38  !$omp parallel @exch_qbc.f
        :256
   OMP  2,238,336    51,456    0.02     0.0     0.35  !$omp parallel @exch_qbc.f
        :245
   OMP  1,124,736    25,856    0.07     0.0     2.82  !$omp parallel @rhs.f:29
   OMP  1,119,168    25,728    0.01     0.0     0.46  !$omp parallel @add.f:23
   OMP  1,119,168    25,728    0.03     0.0     1.12  !$omp parallel @y_solve.f
        :44
   OMP  1,119,168    25,728    0.03     0.0     1.13  !$omp parallel @x_solve.f
        :47
   OMP  1,119,168    25,728    0.03     0.0     1.21  !$omp parallel @z_solve.f
        :44
   OMP    668,928    51,456    0.04     0.0     0.87  !$omp implicit barrier
        @exch_qbc.f:215
 (...)
```

```
(...)
  MPI      17,889        402     0.01      0.0     18.03  MPI_Isend
  MPI      17,889        402     0.00      0.0      7.97  MPI_Irecv
  OMP      11,136        256     0.00      0.0      2.95  !$omp parallel @initialize
       .f:23
  OMP       8,320        640     0.00      0.0      0.20  !$omp atomic @error.f:104
  OMP       8,320        640     0.00      0.0      0.18  !$omp atomic @error.f:51
  OMP       5,568        128     0.00      0.0      1.35  !$omp parallel @error.f:87
  OMP       5,568        128     0.00      0.0      2.10  !$omp parallel @exact_rhs.
       f:22
  OMP       5,568        128     0.00      0.0      1.34  !$omp parallel @error.f:28
  MPI       5,226        402     1.61      0.4   4013.78  MPI_Waitall
(...)
  MPI         476         14     0.00      0.0    106.39  MPI_Bcast
  MPI         204          6     0.00      0.0    582.44  MPI_Reduce
  MPI         136          4     0.01      0.0   1256.30  MPI_Barrier
  MPI          68          2     0.00      0.0     86.88  MPI_Allgather
  MPI          52          4     0.00      0.0      0.75  MPI_Comm_rank
  MPI          52          4     0.00      0.0      3.03  MPI_Comm_size
  MPI          26          2     0.00      0.0    105.51  MPI_Comm_split
```

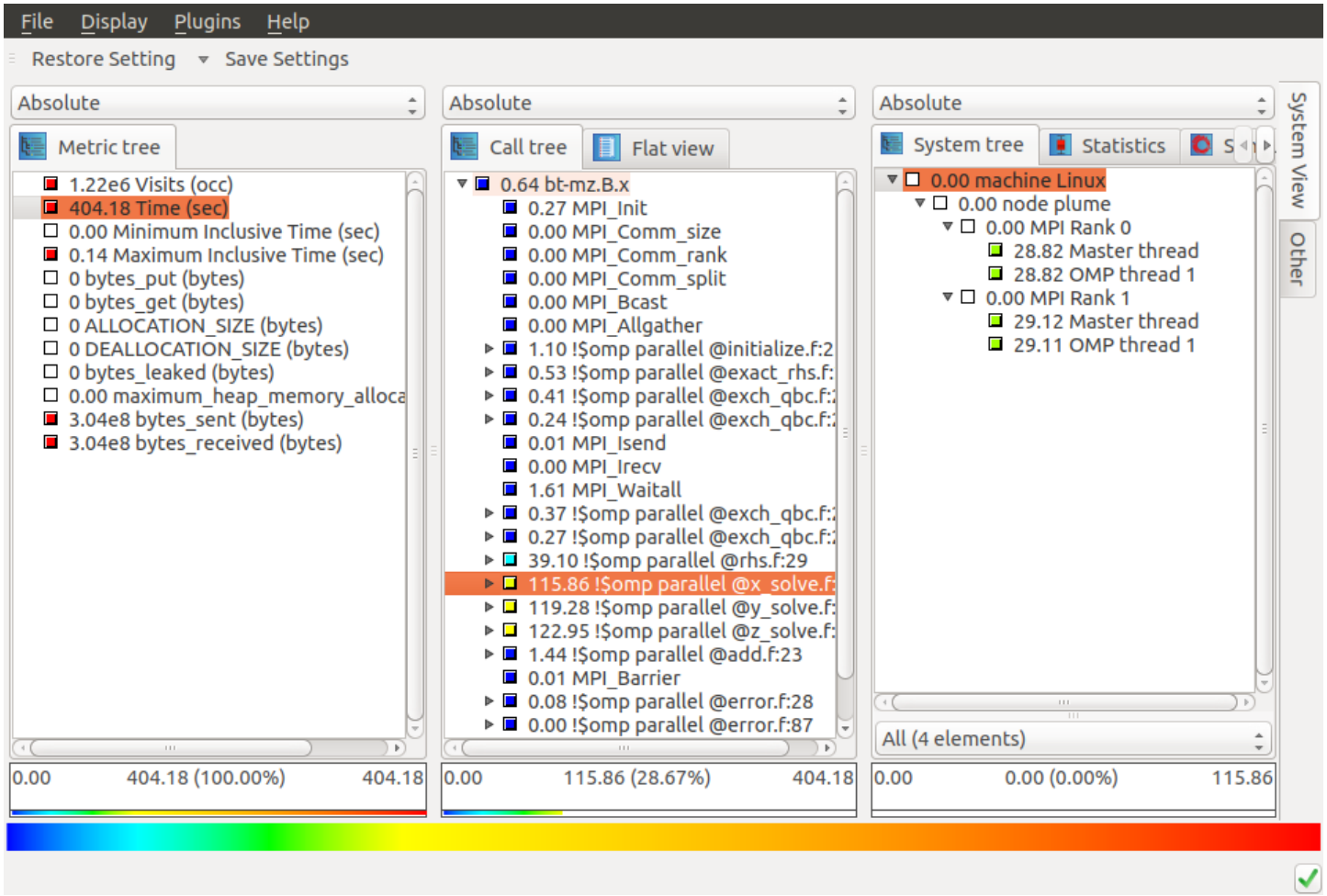
Profil plat extrait avec CUBE

```
$ cube_stat -t 20 -p scorep-bt-mz/profile.cubex

cube::Region                                NumberOfCalls ExclusiveTime InclusiveTime
!$omp do @z_solve.f:51                              25728    121.182131    121.182131
!$omp do @y_solve.f:51                              25728    117.609398    117.609398
!$omp do @x_solve.f:53                              25728    114.142487    114.142487
!$omp do @rhs.f:292                                 25856     10.884323     10.884323
!$omp do @rhs.f:182                                 25856     10.555631     10.555631
!$omp do @rhs.f:77                                  25856     10.358581     10.358581
!$omp do @rhs.f:34                                  25856      3.031725      3.031725
!$omp do @rhs.f:59                                  25856      2.408346      2.561018
!$omp implicit barrier @z_solve.f:427               25728      1.740977      1.740977
!$omp implicit barrier @x_solve.f:406               25728      1.689967      1.689967
!$omp implicit barrier @y_solve.f:405               25728      1.644672      1.644672
MPI_Waitall                                           402      1.613540      1.613540
!$omp do @add.f:23                                  25728      1.343585      1.425174
!$omp do @rhs.f:399                                 25856      1.040513      1.040513
!$omp do @initialize.f:51                             256      0.995974      0.995974
bt-mz.B.x                                               2      0.639516    404.180371
!$omp implicit barrier @rhs.f:410                   25856      0.573769      0.573769
!$omp do @exch_qbc.f:256                            51456      0.352579      0.392707
!$omp do @exch_qbc.f:245                            51456      0.306733      0.356091
MPI_Init                                                2      0.271824      0.271824
```

Graphe d'appel avec CUBE

```
$ cube scorep-bt-mz/profile.cubex
```



Analyse avec SCALASCA

```
scalasca -analyze mpirun -np 2 ../bin/bt-mz.B.x

S=C=A=N: Scalasca 2.5 runtime summarization
S=C=A=N: ./scorep_bt-mz_2x2_sum experiment archive
S=C=A=N: Mon Apr 22 22:11:51 2019: Collect start
  (...)
 Iterations      =                    200
 Time in seconds =                 100.02
  (...)
S=C=A=N: Mon Apr 22 22:13:33 2019: Collect done (status=0) 102s
S=C=A=N: ./scorep_bt-mz_2x2_sum complete.
```

Génération des fichiers de sortie

```
$ ls -1 scorep_bt-mz_2x2_sum/
MANIFEST.md
profile.cubex
scorep.cfg
scorep.log
```

Examen des fichiers de sortie

```
scalasca -examine -s scorep_bt-mz_2x2_sum

INFO: Post-processing runtime summarization report (profile.cubex)
    ...
/softs/scorep/bin/scorep-score -r ./scorep_bt-mz_2x2_sum/profile.
    cubex > ./scorep_bt-mz_2x2_sum/scorep.score
INFO: Score report written to ./scorep_bt-mz_2x2_sum/scorep.score
```

Visualisation avec CUBE

```
$ cube scorep-bt-mz/summary.cubex
```

File Display Plugins Help

Restore Setting Save Settings

Absolute | Absolute | Absolute

Metric tree

- 404.57 Time (sec)
- 1.22e6 Visits (occ)
- 6.08e8 Bytes transferred (bytes)
- 0 MPI file operations (occ)
- 0.00 Computational imbalance (sec)
  - 0.89 Overload
  - 0.89 Underload
- 0.00 Minimum Inclusive Time (sec)
- 0.14 Maximum Inclusive Time (sec)
- 0 ALLOCATION_SIZE (bytes)
- 0 DEALLOCATION_SIZE (bytes)
- 0 bytes_leaked (bytes)
- 0.00 maximum_heap_memory_alloca

0.00 | 0.89 (50.00%) | 1.79

Call tree | Flat view

- 0.00 MPI_Comm_size
- 0.00 MPI_Comm_rank
- 0.00 MPI_Comm_split
- 0.00 MPI_Bcast
- 0.00 MPI_Allgather
- 0.01 !$omp parallel @initialize.f:2
- 0.01 !$omp parallel @exact_rhs.f:
- 0.01 !$omp parallel @exch_qbc.f:
- 0.01 !$omp parallel @exch_qbc.f:
- 0.00 MPI_Isend
- 0.00 MPI_Irecv
- 0.00 MPI_Waitall
- 0.00 !$omp parallel @exch_qbc.f:
- 0.01 !$omp parallel @exch_qbc.f:
- 0.11 !$omp parallel @rhs.f:29
- 0.31 !$omp parallel @x_solve.f:47
- 0.13 !$omp parallel @y_solve.f:44
- 0.28 !$omp parallel @z_solve.f:44
- 0.01 !$omp parallel @add.f:23
- 0.00 MPI_Barrier
- 0.00 !$omp parallel @error.f:28
- 0.00 !$omp parallel @error.f:87
- 0.00 MPI_Reduce
- 0.00 MPI_Finalize

0.00 | 0.31 (35.20%) | 0.89

System tree | Statistics

- 0.00 machine Linux
  - 0.00 node plume
    - 0.00 MPI Rank 0
      - 0.00 Master thread
      - 0.00 OMP thread 1
    - 0.00 MPI Rank 1
      - 0.27 Master thread
      - 0.05 OMP thread 1

All (4 elements)

0.00 | 0.00 (0.00%) | 0.31

System View | Other

Ready

File Display Plugins Help

Restore Setting Save Settings

Absolute

Metric tree

- 0.00 Time (sec)
  - 0.00 Execution
    - 392.87 Computation
    - 0.00 MPI
      - 0.27 Management
      - 0.01 Synchronization
      - 0.00 Communication
        - 1.72 Point-to-point
        - 0.00 Collective
        - 0.00 One-sided
      - 0.00 File I/O
    - 7.03 OpenMP
  - 0.00 Overhead
  - 2.66 Idle threads
- 1.22e6 Visits (occ)
- 6.08e8 Bytes transferred (bytes)
- 0 MPI file operations (occ)
- 0.00 Computational imbalance (sec)
  - 0.89 Overload
  - 0.89 Underload
- 0.00 Minimum Inclusive Time (sec)
- 0.14 Maximum Inclusive Time (sec)
- 0 ALLOCATION_SIZE (bytes)
- 0 DEALLOCATION_SIZE (bytes)

0.00 1.72 (0.43%) 404.57

Absolute

Call tree Flat view

- 0.00 MPI_Comm_rank
- 0.00 MPI_Comm_split
- 0.00 MPI_Bcast
- 0.00 MPI_Allgather
- 0.00 !$omp parallel @initialize.f:2
- 0.00 !$omp parallel @exact_rhs.f:
- 0.00 !$omp parallel @exch_qbc.f:
- 0.00 !$omp parallel @exch_qbc.f:
- 0.01 MPI_Isend
- 0.00 MPI_Irecv
- 1.71 MPI_Waitall
- 0.00 !$omp parallel @exch_qbc.f:
- 0.00 !$omp parallel @exch_qbc.f:
- 0.00 !$omp parallel @rhs.f:29
- 0.00 !$omp parallel @x_solve.f:47
- 0.00 !$omp parallel @y_solve.f:44
- 0.00 !$omp parallel @z_solve.f:44
- 0.00 !$omp parallel @add.f:23
  - 0.00 !$omp do @add.f:23
- 0.00 MPI_Barrier
- 0.00 !$omp parallel @error.f:28
- 0.00 !$omp parallel @error.f:87
- 0.00 MPI_Reduce
- 0.00 MPI_Finalize

0.00 1.71 (99.38%) 1.72

Absolute

System tree Statistics

- 0.00 machine Linux
  - 0.00 node plume
    - 0.00 MPI Rank 0
      - 1.13 Master thread
      - 0.00 OMP thread 1
    - 0.00 MPI Rank 1
      - 0.58 Master thread
      - 0.00 OMP thread 1

All (4 elements)

0.00 1.13 (65.80%) 1.71

System View Other

Selected "Master thread"

Références

- `VI-HPS Tool guide`
- `NAS Benchmarks`
- Tutoriels `https://computing.llnl.gov/`
- G. Markomanolis *Studying the behavior of parallel applications and identifying bottlenecks by using performance analysis tools*, École Méthodologie et outils d'optimisation en développement logiciel, IN2P3, 2012