LProf

Version 1.1
April 2018

1 Introduction

MAQAO Lightweight Profiler (LProf) is the MAQAO module which allows to easily profile an application to detect hot functions and loops in two steps:

1) Data collection using sampling

LProf uses hardware counters to profile large-scale parallel applications (2000+ cores) with a very low overhead.
It is also possible to provide a custom list of hardware counters to sample.

2) Data display

It allows to easily identify time-consuming functions and loops, observe the amount of time spent by the application between different categories (I/O, Runtime, etc…) and detect load balancing issues.

2 Running MAQAO LProf

2.1 Sequential Run Command

```
maqao lprof -- <application> [arg1 arg2 ...]
```

application's name (or path if not located in the current directory)

application's arguments, if any

2.2 Parallel Run Command

```
mpirun -n <NB_PROCESSES> maqao lprof -- <application> [arg1 arg2 ...]
```

MPI launcher command

number of processes

2.3 Options

To list all options:

```
maqao lprof --help
```

Common Options		
Name	**Short Description**	**Values**
xp=	Specify the experiment directory	Directory's name (a string)
g=	Change granularity (*i.e.* the number of collected samples)	small | medium (default) | large
ug=	Control (*i.e.* pause/resume) measurement via a signal (Ctrl+Z) or via a countdown	on | off (signal, default = off) or Time (countdown, in seconds)
ldi=	Scan debug information into all or specified (provided list) library(ies) to get loops details	on | off (all, default = off) or List of libraries ('lib1, lib2, …')
kdo=	Bypass kernel detection (*i.e.* version check)	on | off (default)
hwc=	Provide custom list of hardware counters to sample	hwc1_name@threshold_freq, … or hwc1_code@threshold_freq, …

3 Display

The two common display modes are text (default) and HTML.

3.1 Text Output

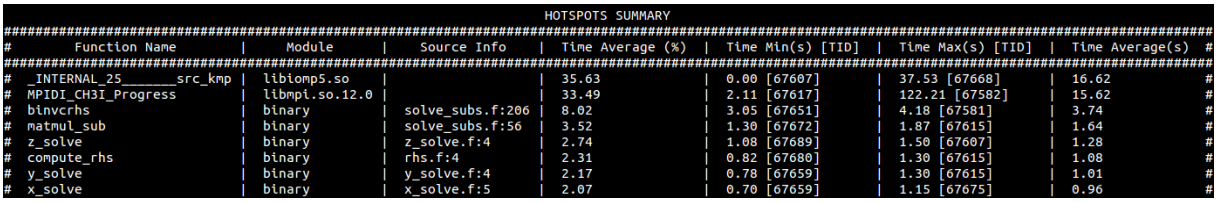
3.1.1 Functions Hotspots

To display summary view:

```
maqao lprof -df xp=<EXPERIMENT_DIRECTORY>
```

hotspots' level: functions

experiment directory (name or path) containing data to display



HOTSPOTS SUMMARY

Function Name	Module	Source Info	Time Average (%)	Time Min(s) [TID]	Time Max(s) [TID]	Time Average(s)
_INTERNAL_25_____src_kmp	libiomp5.so		35.63	0.00 [67607]	37.53 [67668]	16.62
MPIDI_CH3I_Progress	libmpi.so.12.0		33.49	2.11 [67617]	122.21 [67582]	15.62
binvcrhs	binary	solve_subs.f:206	8.02	3.05 [67651]	4.18 [67581]	3.74
matmul_sub	binary	solve_subs.f:56	3.52	1.30 [67672]	1.87 [67615]	1.64
z_solve	binary	z_solve.f:4	2.74	1.08 [67689]	1.50 [67607]	1.28
compute_rhs	binary	rhs.f:4	2.31	0.82 [67680]	1.30 [67615]	1.08
y_solve	binary	y_solve.f:4	2.17	0.78 [67659]	1.30 [67615]	1.01
x_solve	binary	x_solve.f:5	2.07	0.70 [67659]	1.15 [67675]	0.96

Figure 1 - LProf Output: Summary View (Functions)

To display thread view:

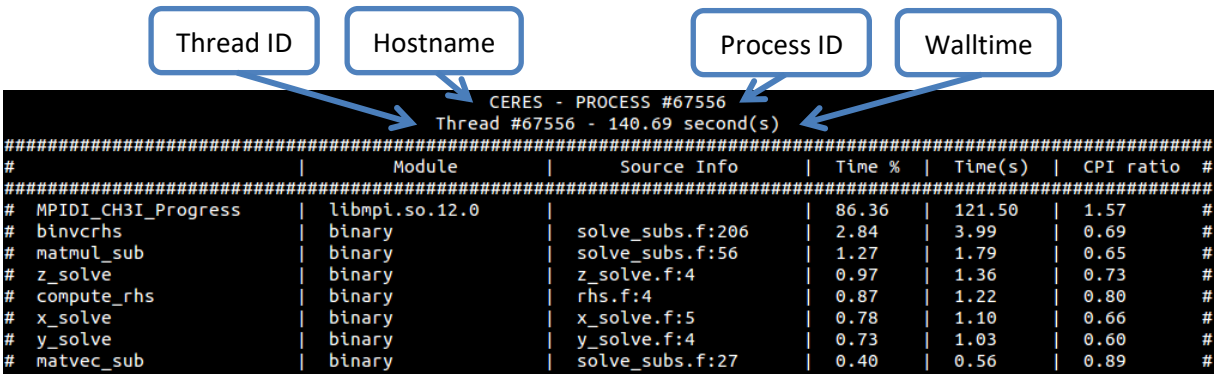
```
maqao lprof -df xp=<EXPERIMENT_DIRECTORY> -dt
```

Thread ID

Hostname

Process ID

Walltime



CERES - PROCESS #67556
Thread #67556 - 140.69 second(s)

	Module	Source Info	Time %	Time(s)	CPI ratio
MPIDI_CH3I_Progress	libmpi.so.12.0		86.36	121.50	1.57
binvcrhs	binary	solve_subs.f:206	2.84	3.99	0.69
matmul_sub	binary	solve_subs.f:56	1.27	1.79	0.65
z_solve	binary	z_solve.f:4	0.97	1.36	0.73
compute_rhs	binary	rhs.f:4	0.87	1.22	0.80
x_solve	binary	x_solve.f:5	0.78	1.10	0.66
y_solve	binary	y_solve.f:4	0.73	1.03	0.60
matvec_sub	binary	solve_subs.f:27	0.40	0.56	0.89

Figure 2 - LProf Output: Thread View (Functions)

3.1.2 Loops Hotspots

To display summary view:

```
maqao lprof -dl xp=<EXPERIMENT_DIRECTORY>
```

hotspots' level: loops

experiment directory (name or path) containing data to display

```
#################################################################################################
#  Loop ID  |  Module  |       Function Name       |        Source Info        |    Level     |
#################################################################################################
#  221      |  binary  |  matmul_sub               |  solve_subs.f:71-175      |  Single      |
#  230      |  binary  |  z_solve                  |  z_solve.f:146-308        |  Innermost   |
#  227      |  binary  |  z_solve                  |  z_solve.f:55-137         |  Innermost   |
#  204      |  binary  |  y_solve                  |  y_solve.f:145-307        |  Innermost   |
#  197      |  binary  |  x_solve                  |  x_solve.f:146-308        |  Innermost   |
#  201      |  binary  |  y_solve                  |  y_solve.f:55-137         |  Innermost   |
#  194      |  binary  |  x_solve                  |  x_solve.f:57-139         |  Innermost   |
#  226      |  binary  |  z_solve                  |  z_solve.f:415-423        |  Innermost   |
#  122      |  binary  |  compute_rhs              |  rhs.f:304-349            |  Innermost   |
#  193      |  binary  |  x_solve                  |  x_solve.f:395-399        |  Innermost   |
#  148      |  binary  |  compute_rhs              |  rhs.f:194-238            |  Innermost   |
```

Figure 3 – LProf Output: Summary View (Loops)

The above figure is truncated. In the actual output, four more columns are available on the right (same as functions mode):

Time Average (%), Time Min (s), Time Max (s) and Time Average (s).

To display thread view:

```
maqao lprof -dl xp=<EXPERIMENT_DIRECTORY> -dt
```

Thread ID

Hostname

Process ID

Walltime

```
                                CERES - PROCESS #67556
                            Thread #67556 - 140.69 second(s)
#################################################################################################################################
#  Loop ID  |  Module  |       Function Name       |        Source Info        |    Level     |  Time %  |  Time (s)  |  CPI ratio  #
#################################################################################################################################
#  221      |  binary  |  matmul_sub               |  solve_subs.f:71-175      |  Single      |  0.54    |  0.76      |  0.65       #
#  197      |  binary  |  x_solve                  |  x_solve.f:146-308        |  Innermost   |  0.37    |  0.52      |  0.59       #
#  227      |  binary  |  z_solve                  |  z_solve.f:55-137         |  Innermost   |  0.36    |  0.50      |  1.10       #
#  230      |  binary  |  z_solve                  |  z_solve.f:146-308        |  Innermost   |  0.35    |  0.50      |  0.49       #
#  204      |  binary  |  y_solve                  |  y_solve.f:145-307        |  Innermost   |  0.32    |  0.45      |  0.46       #
#  201      |  binary  |  y_solve                  |  y_solve.f:55-137         |  Innermost   |  0.27    |  0.37      |  0.91       #
#  194      |  binary  |  x_solve                  |  x_solve.f:57-139         |  Innermost   |  0.23    |  0.32      |  0.79       #
#  226      |  binary  |  z_solve                  |  z_solve.f:415-423        |  Innermost   |  0.18    |  0.26      |  0.82       #
#  122      |  binary  |  compute_rhs              |  rhs.f:304-349            |  Innermost   |  0.16    |  0.23      |  1.01       #
```

Figure 4 - LProf Output: Thread View (Loops)

3.2 HTML Output

3.2.1 Generation of HTML results

```
maqao lprof xp=<EXPERIMENT_DIRECTORY> of=html
```

experiment directory (name or path) containing data to display

output format: HTML

This command generates an 'index.html' file into the <EXPERIMENT_PATH>/html/ directory. Open this file into a web browser to see the results.

3.2.2 Interpretation of the Results

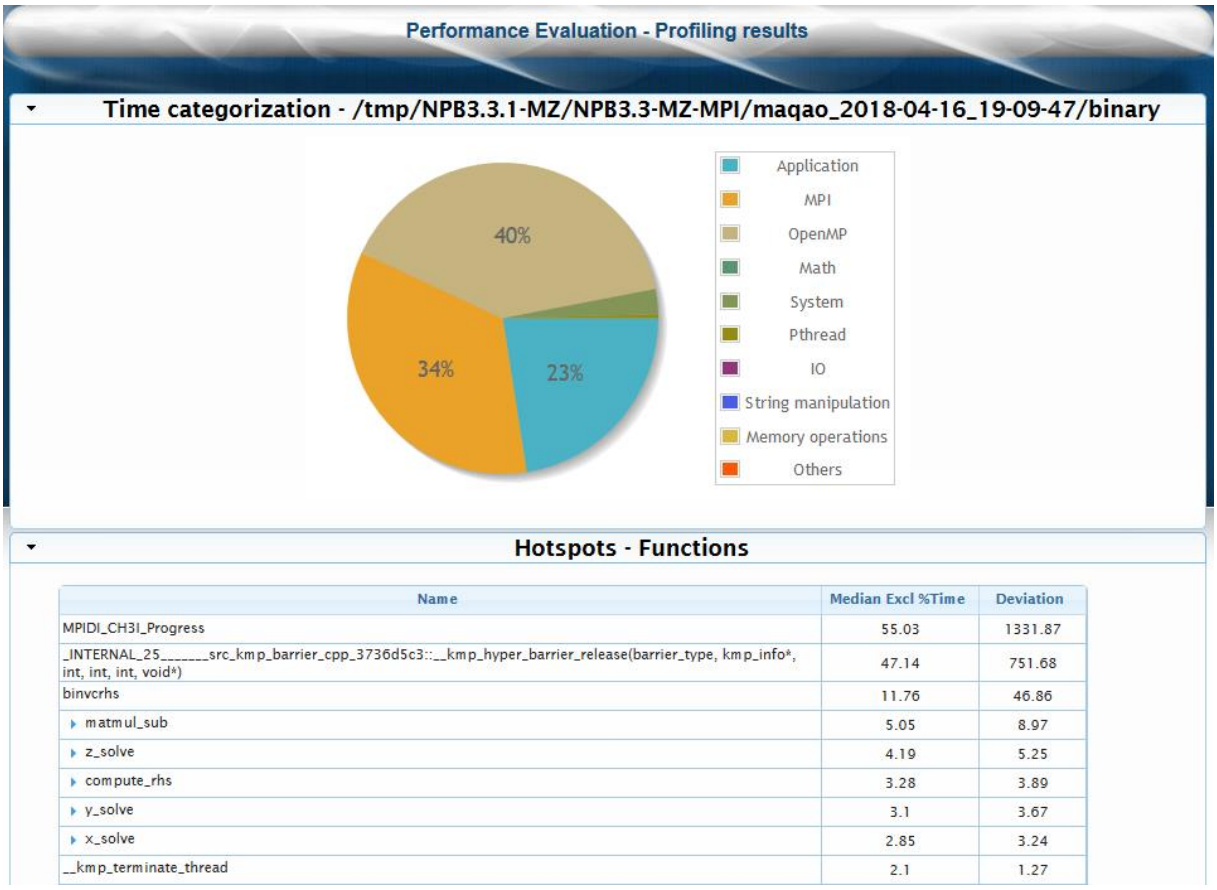


Figure 5 - LProf Output: HTML View

Name	Function's name
Median Excl % Time	Median exclusive time in percent
Deviation	Variability of function's time on each thread/process

The load balancing graph of a function across the threads/processes/nodes can be displayed by right-clicking on its name (figures 6 & 7).

Hotspots - Functions

Name	Median Excl %Time	Deviation
MPIDI_CH3I_Progress	55.03	1331.87
_INTERNAL_25_____src_kmp_barrier_cpp_3736d5c3::__kmp_hyper_barrier_release(barrier_type, kmp_info*, int, int, int, void*)	47.14	751.68
binvcrhs	11.76	46.86
▸ matmul_sub	5.05	8.97
▸ z_solve	4.19	5.25
▸ compute_rhs	3.28	3.89
▸ y_solve	3.1	3.67
▸ x_solve	2.85	3.24
__kmp_terminate_thread	2.1	1.27

Load balancing view
Sorted Load balancing view
Node view

Figure 6 - LProf Output: HTML Right-Click Options

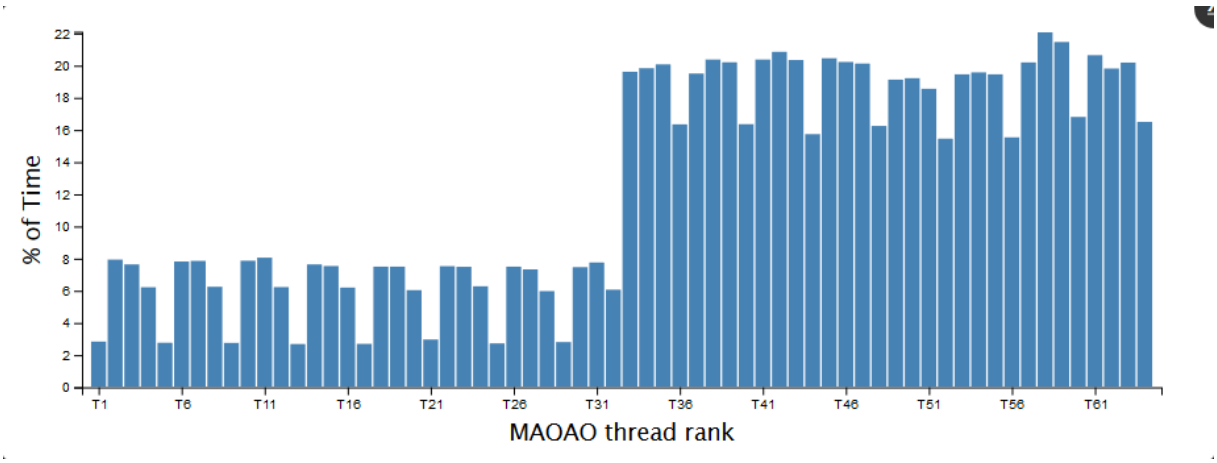


Figure 7 - LProf Output: HTML Load Balancing Graph

The load balancing graph shows the time (in percent) spent in a function on each thread. To get more details, it is possible to double-click on a bar to view the corresponding thread (figure 8).

ceres - Process #67574 - Thread #67612

Name	Excl %Time	Excl Time (s)
_INTERNAL_25_____src_kmp_barrier_cpp_3736d5c3::__kmp_hyper_barrier_gather(barrier_type, kmp_info*, int, int, void (*)(void*, void*), void*)	24.74	4.94
binvcrhs solve_subs.f:206	19.51	3.89
MPIDI_CH3I_Progress	12.36	2.47
▸ matmul_sub solve_subs.f:56	8.97	1.79
▸ z_solve z_solve.f:4	6.98	1.39
▸ compute_rhs rhs.f:4	5.78	1.15
▸ x_solve x_solve.f:5	5.68	1.13
▸ y_solve y_solve.f:4	5.49	1.10

Figure 8 - LProf Output: HTML Thread View

The thread view allows to expand functions to get some information such as the time spent in its loops (if any) as well as the loop hierarchy too (figure 9).

matmul_sub solve_subs.f:56	8.97	1.79
z_solve z_solve.f:4	6.98	1.39
loops	6.96	
Loop 223 - z_solve.f:53-423	0	
Loop 225 - z_solve.f:54-423	0	
Loop 228 - z_solve.f:54-423	0.12	
Loop 224 - z_solve.f:313-314	0.37	
Loop 227 - z_solve.f:55-137	2.49	
Loop 226 - z_solve.f:415-423	1.45	
Loop 229 - z_solve.f:351-373	0.15	
Loop 230 - z_solve.f:146-308	2.38	
compute_rhs rhs.f:4	5.78	1.15

Figure 9 - LProf Output: HTML Loops Details

Name	Function's name - source_file:line Loop's ID - source_file:first_line-last_line
Excl % Time	Exclusive time spent in the function (percent)
Excl Time (s)	Exclusive time spent in the function (seconds)