



LProf

Version 1.0

MAQAO Tutorial series

1 Introduction

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

1) Data collection in two available modes:

- Sampling (default)
- Instrumentation

The sampling mode is based on the hardware counters and allows to profile large-scale parallel applications (2000+ cores) with a very low overhead. you can also drive your own analysis by giving your own hardware counters to sample.

2) Data display

You can easily see time-consuming functions and loops, see the time spent of your application between different categories (MPI, OpenMP...), detect load balancing issues.

2 Running MAQAO LProf

2.1 Sequential run command

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

Name of your application (or path if not present in the current directory)

If necessary, arguments of your application

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
```

Options		
name	comments	values
xp=	Specify your experiment directory	string
g=	Change the number of collected samples	small medium (default) large
ug=	Control the measurement via a signal (ctrl+z) or via a countdown	on off (default) time (in sec)
ldi=	To scan debug info into all the libraries or into specific library(ies) in giving a list.	on off (default) list,of,libraries
bkd=	To bypass the control kernel detection	on off (default)
p=	Choose your maqao profile (only available with sandy bridge).	memory compute branch dtlb
hwc=	Specify your own hardware counters	hwc_name@threshold_freq,... my_profile.mperf

3 Display

The two common display modes are the text and the HTML mode.

3.1 Text mode

3.1.1 Functions hotspots

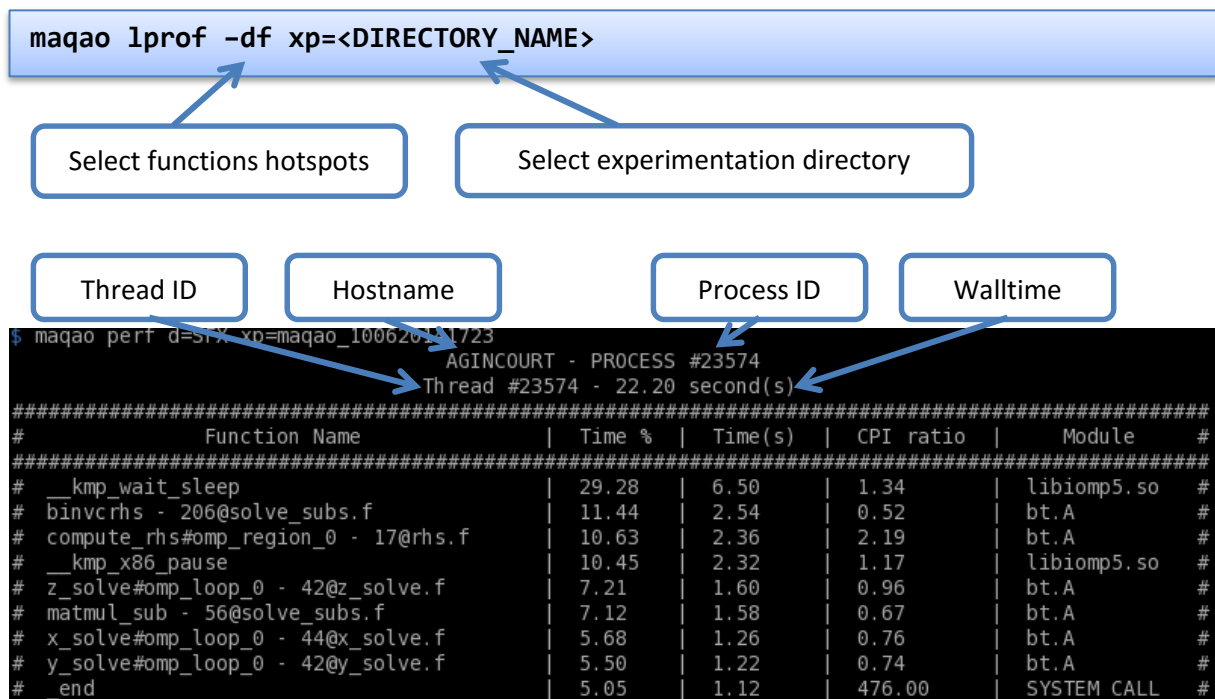


Figure 1 – SFX output

3.1.2 Loops hotspots

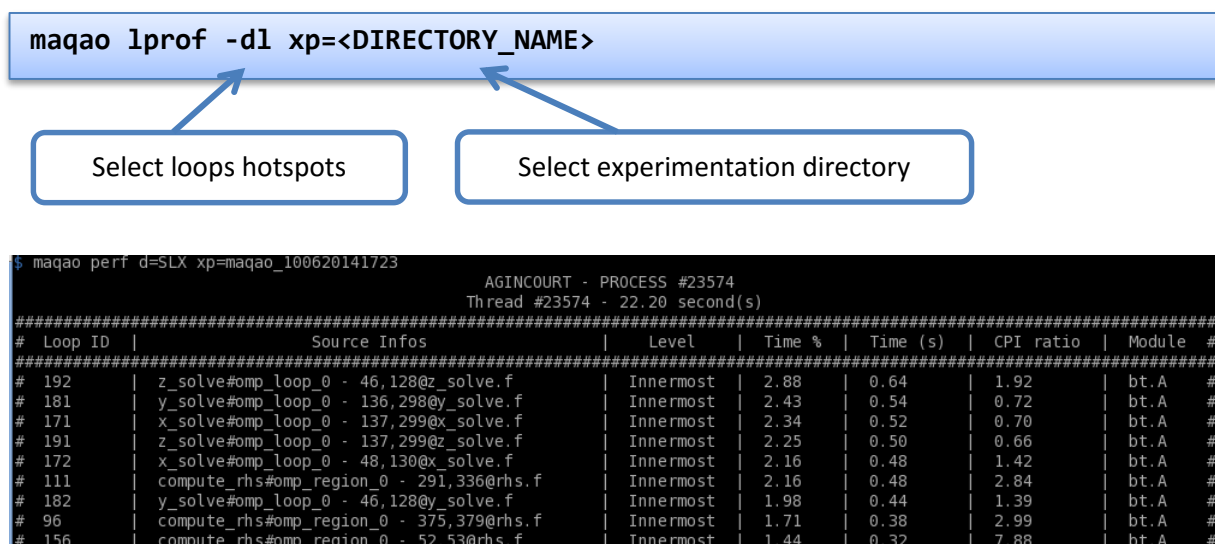


Figure 2 – SLX output

3.2 HTML

3.2.1 Generation of the HTML directory

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

Select html format (output-format)

Now you can find an index.html file into your <DIRECTORY_NAME>/html.

3.2.1.1 Interpret the results

Open<DIRECTORY_NAME>/html/index.html with your favorite web browser. You will see the MAQAO LProf home page (Figure 3).

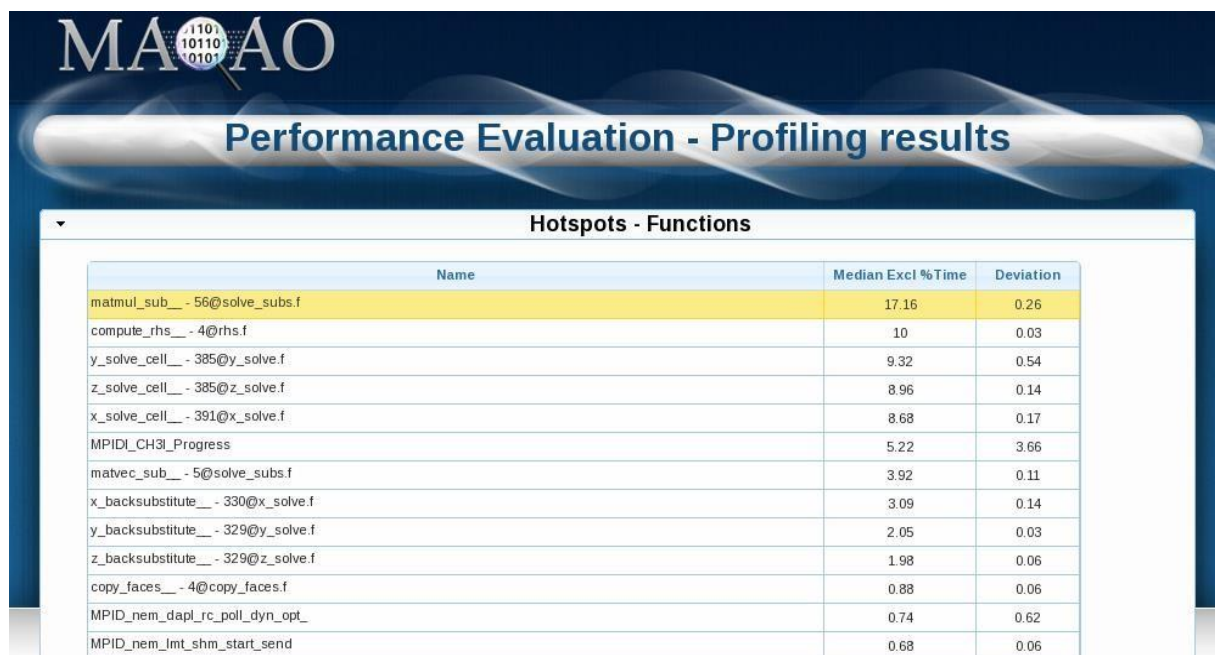


Figure 3 – MAQAO LProf HTML

Name	Format : Function name – source_line@filename
Median Excl % Time	Median exclusive time in percent
Deviation	Measure the variability of this function on each thread/process

If you double click on one function name, the load balancing graph will appear (Figure 4).

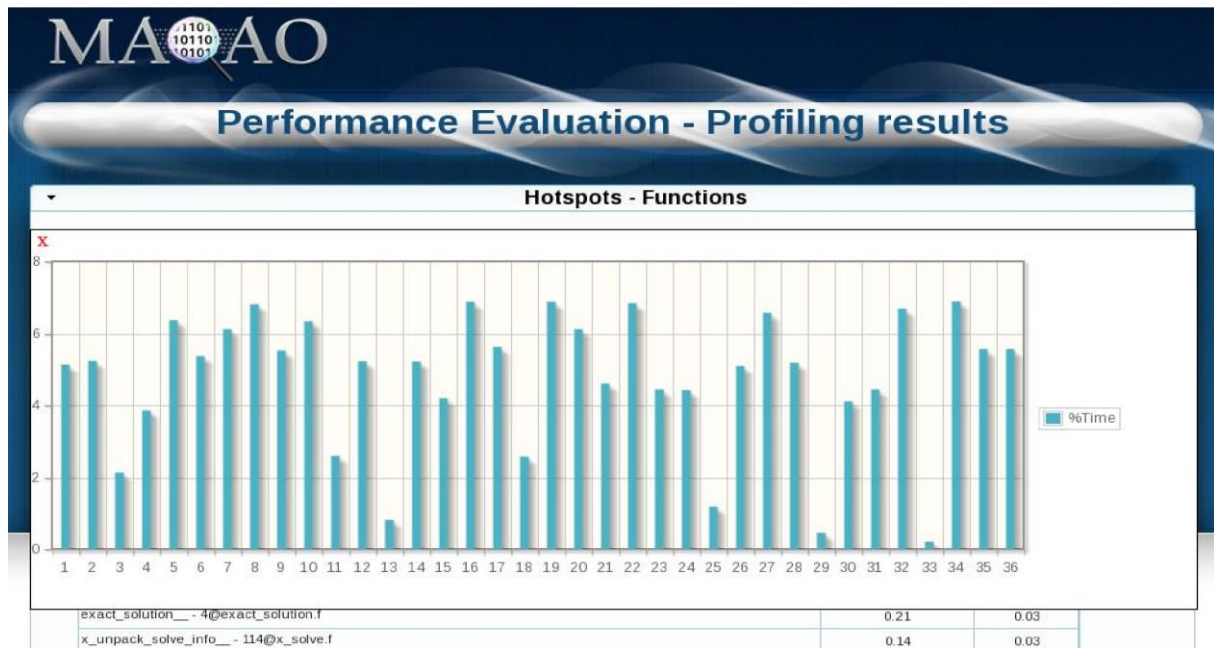


Figure 4 – Load balancing graph

The load balancing graph shows to you the time (in percent) spent in a function on each thread. If you want more details you can click on a bar to “zoom” into the desired thread (Figure 5). As you can see on the figure 5, you can expand to get some information such as the time spent in the loops of a function (you can see the loop hierarchy too).

cirrus5003 - Process #53572 - Thread #1			
Name	Excl %Time	Excl Time (s)	
matmul_sub__ - 56@solve_subs.f	16.92	16.48	
▶ compute_rhs__ - 4@rhs.f	9.92	9.66	
▼ y_solve_cell__ - 385@y_solve.f	9.08	8.84	
▼ loops	9.08		
▼ Loop 267 - y_solve.f@415	0		
▼ Loop 268 - y_solve.f@425	0		
○ Loop 272 - y_solve.f@426	0.25		
c Loop 270 - y_solve.f@524	6.57		
○ Loop 271 - y_solve.f@436	2.22		
○ Loop 269 - y_solve.f@716	0.04		
▼ x_solve_cell__ - 391@x_solve.f	9.01	8.78	
▼ loops	9.01		
▼ Loop 235 - x_solve.f@420	0		
▼ Loop 236 - x_solve.f@429	0		
○ Loop 237 - x_solve.f@709	0.06		
○ Loop 239 - x_solve.f@431	2.71		
c Loop 238 - x_solve.f@519	6.24		

Figure 5 – Thread information

Name	Format : Function name – source_line@filename
Excl % Time	Exclusive time spent in this function in percent
Excl Time (s)	Exclusive time spent in this function in seconds