

Assignment 4

Erkin Kirdan - Manuel Rothenberg - Gabrielle Poerwawinata

1. Superuseful application - GNU gprof

Why are the Kernels not showing in the report?

- -O3 inlines the functions

Compiler options:

- -p : for compiling and linking for function profiling with gprof
- -g: for telling the compiler to generate full debugging information in the object file

1. Superuseful application - GNU gprof

Call graph (explanation follows)

granularity: each sample hit covers 2 byte(s) for 0.03% of 30.22 seconds

index	% time	self	children	called	name
					<spontaneous>
[1]	89.3	0.00	26.99		main [1]
		21.29	0.00	7972/7972	kernel2(double*, double*, int) [2]
		2.39	3.00	2028/2028	kernel1(double*, double*, int) [3]
		0.31	0.00	1/1	Stopwatch::stop() [9]
		0.00	0.00	1/1	Stopwatch::Stopwatch() [28]
		0.00	0.00	1/1	Stopwatch::start() [25]

		21.29	0.00	7972/7972	main [1]
[2]	70.5	21.29	0.00	7972	kernel2(double*, double*, int) [2]

		2.39	3.00	2028/2028	main [1]
[3]	17.8	2.39	3.00	2028	kernel1(double*, double*, int) [3]
		3.00	0.00	2028000000/2028000000	function(double) [4]

2. Quicksort - Intel VTune Amplifier XE

- Basic Hotspots: software-based sampling with a default resolution of 10 ms
- Advanced Hotspots: hardware-based sampling with a default resolution of 1 ms.
- For the quicksort application, sampling rate is already much larger than the execution time, so it's not needed to make an advanced hotspots profiling.
- The optimal final clause does depend on the number of threads since number of threads directly determine the spawning resolution. For example, for larger number of threads, final clause should be much lower.
- The optimal final clause does not depend on the array length.