

Laboratory B

Group 4 Predrag Bozhovikj, Erik Sievers

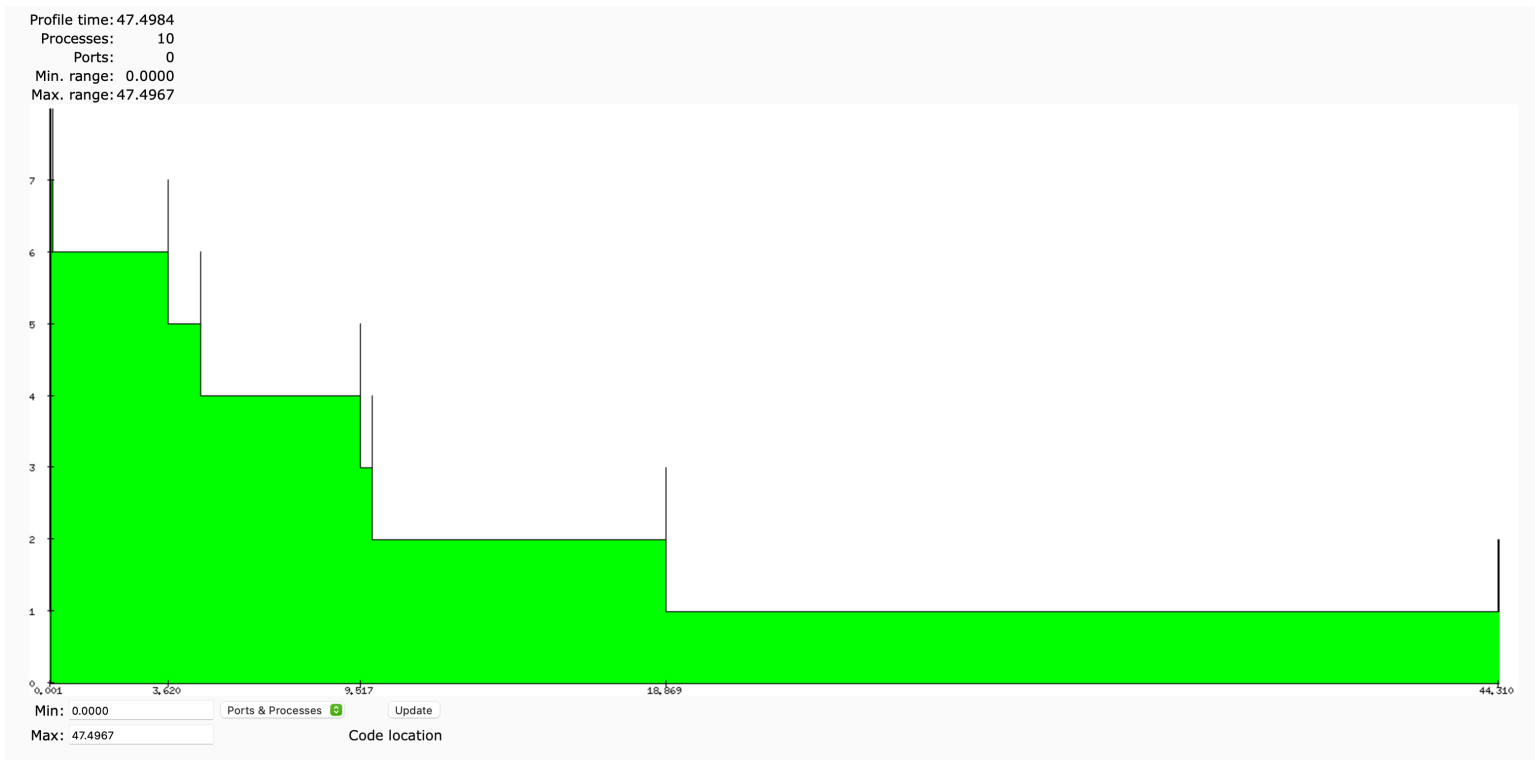
Part one: parallelising the multiple puzzles

We ran the benchmarks on two different machines: one with an i7 Dual-Core processor and one with an i7 Quad-Core processor.

For 100 benchmark runs, running the benchmark takes 77.02 seconds on the Dual-Core and 81.77 seconds on the quad core. When we parallelize the solving of the various puzzles, the speedup is fairly small: solving the puzzles in parallel takes 47.50s on the dual core and 55s on the quad core. The reason can be found when looking at how long each puzzle takes to solve:

1	i7 Dual-Core
2	====
3	{77023483,
4	[{wildcat,0.42735},
5	{diabolical,53.09819},
6	{vegard_hanssen,119.4723},
7	{challenge,12.75032},
8	{challenge1,512.22093},
9	{extreme,12.560649999999999},
10	{seventeen,59.704879999999996}]}
11	
12	i7 Quad-Core
13	====
14	{81769956,
15	[{wildcat,0.47587},
16	{diabolical,65.51498000000001},
17	{vegard_hanssen,139.65792000000002},
18	{challenge,11.45117},
19	{challenge1,539.63801},
20	{extreme,12.84231},
21	{seventeen,48.11901}]}

As we can see, the challenge1 puzzle is by far the slowest to solve: in fact, it takes longer to solve that puzzle than all other puzzles combined. Analysing using percept confirms this: most processes finish quickly, apart from one.



Contrast this with the sequential implementation that does all work on one thread.



We tried three different approaches to parallelising the solver: 1. Parallelising the refinement of rows using `spawn_link` 2. Parallel search using a worker pool. 3. Parallelising the calls to `refine` in `guesses` using a worker pool 4. Parallelising the calls to `refine` in `guesses` using `spawn_link` 5. Parallelising the calls to `refine_row` in `refine_rows`

We ran the benchmark with two different computers: one with an i7 Dual-Core processor and one with an i7 Quad-Core processor. Both computers had 16gb of RAM.

Task	Dual-Core	Quad-Core
Sequential	77s	81s
Parallel problems	48s	55s
Parallel search	59s	41s
Parallel guess	115s	92s
Parallel refine	~31m	~38m

The best parallelisation opportunity we found (and the only that improved the execution time) was the parallel search, which on the dual core processor resulted in a speed-up of 30% and on the quad-core 98%. The other forms of parallelism seem to be too finely granular to be worthwhile and actively worsen the performance.

The benchmark for the parallel refine is approximate. After the benchmark didn't finish for 10 minutes, it was cancelled. Running one execution instead of 100 took 19s and 23s respectively. Multiplying that by 100, we get a rough estimate of 31 minutes and 38 minutes, respectively.