Programming parallel computers: Exercise 3

Suhas Thejaswi Muniyappa (465959)

May 3, 2015

1 IS1

Image segmentation. The bottleneck is to reduce the number of divisions in the innermost loop. For a matrix of 250x250 the inner most loop executes almost 1 million iterations. so reduction in the number of instructions increases the performance of the program.

2 SO1

Sorting using multiple cores. Merge sort, the bottle neck is merging rather than the actual sorting procedure.

	ny	nx	1-thread	2-threads	4-threads	8-threads
is	1	1	0.000	0.000	0.000	0.000
is	1	10	0.000	0.000	0.000	0.000
is	1	100	0.000	0.000	0.000	0.000
is	1	200	0.000	0.000	0.000	0.001
is	1	400	0.001	0.001	0.001	0.002
is	10	1	0.000	0.000	0.000	0.000
is	10	10	0.000	0.000	0.000	0.000
is	10	100	0.004	0.002	0.001	0.002
is	10	200	0.014	0.008	0.005	0.005
is	10	400	0.032	0.023	0.019	0.017
is	100	1	0.000	0.000	0.000	0.000
is	100	10	0.002	0.001	0.001	0.001
is	100	100	0.145	0.075	0.047	0.036
is	100	200	0.572	0.290	0.156	0.141
is	100	400	2.281	1.149	0.617	0.558
is	200	1	0.000	0.000	0.000	0.000
is	200	10	0.015	0.004	0.002	0.002
is	200	100	0.588	0.292	0.156	0.140
is	200	200	2.282	1.150	0.613	0.551
is	200	400	9.067	4.572	2.436	2.185
is	400	1	0.001	0.000	0.000	0.000
is	400	10	0.026	0.021	0.007	0.007
is	400	100	2.313	1.166	0.628	0.554
is	400	200	9.156	4.612	2.486	2.190
is	400	400	36.453	18.366	9.722	8.678

Table 1: IS1 performance statistics

	ny	nx	Instructions	cache misses	CPU utilised
is	100	100	1,020,496,700	26,364	7.362
is	200	200	15,861,456,407	66,600	7.758
is	300	300	79,806,524,635	147,037	7.847
is	400	400	251,549,640,245	5,426,488	7.898

Table 2: IS1 performance statistics

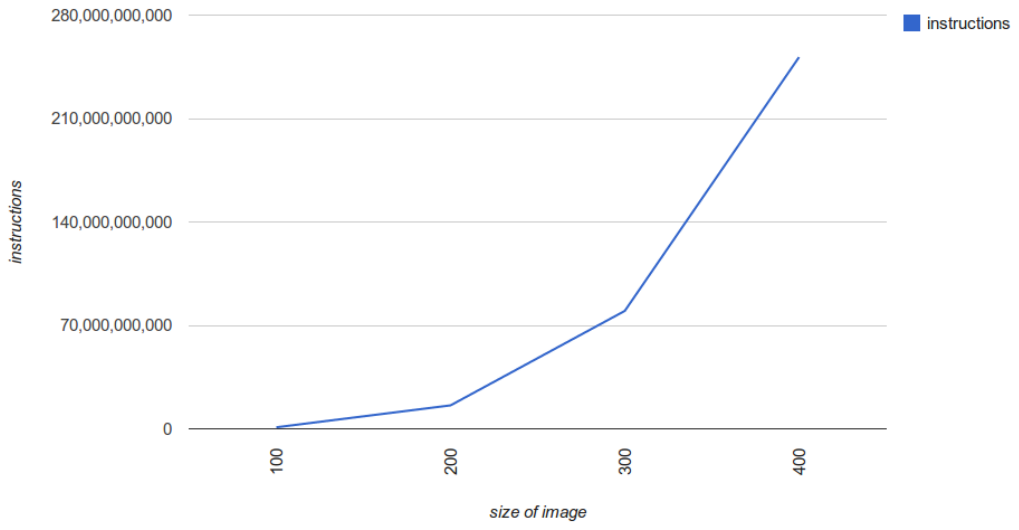

Figure 1: Total number of instructions for various image size

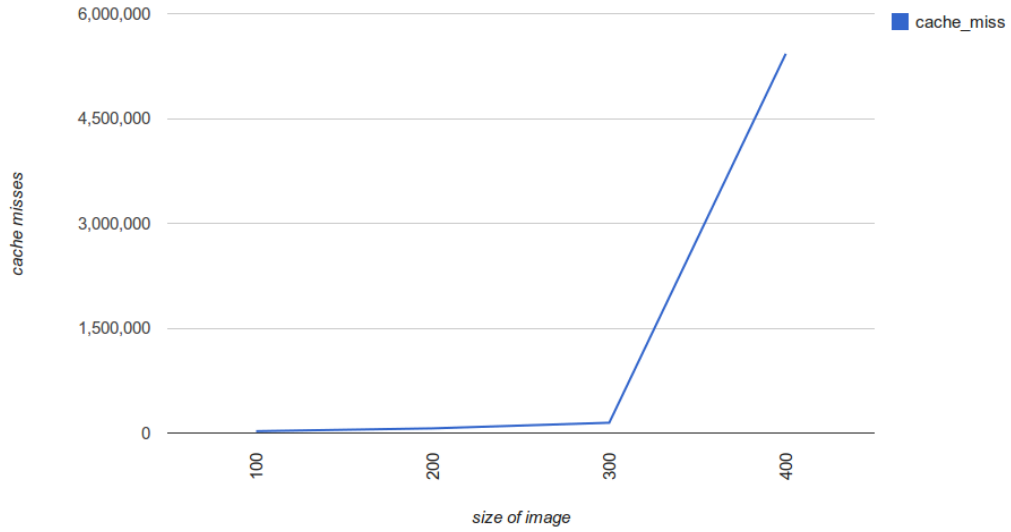

Figure 2: IS1 cache miss statistics for 8-threads

size	random	N	threads_1	threads_2	threads_4	threads_8	std::sort
so	0	100000000	7.748	4.495	3.098	2.735	7.964
so	1	100000000	2.334	1.625	1.502	1.713	2.322
so	2	100000000	1.764	1.47	1.491	1.68	1.694
so	3	100000000	1.594	1.3	1.239	1.393	1.758
so	4	100000000	1.18	1.108	1.16	1.333	1.274

Table 3: SO1 performance statistics

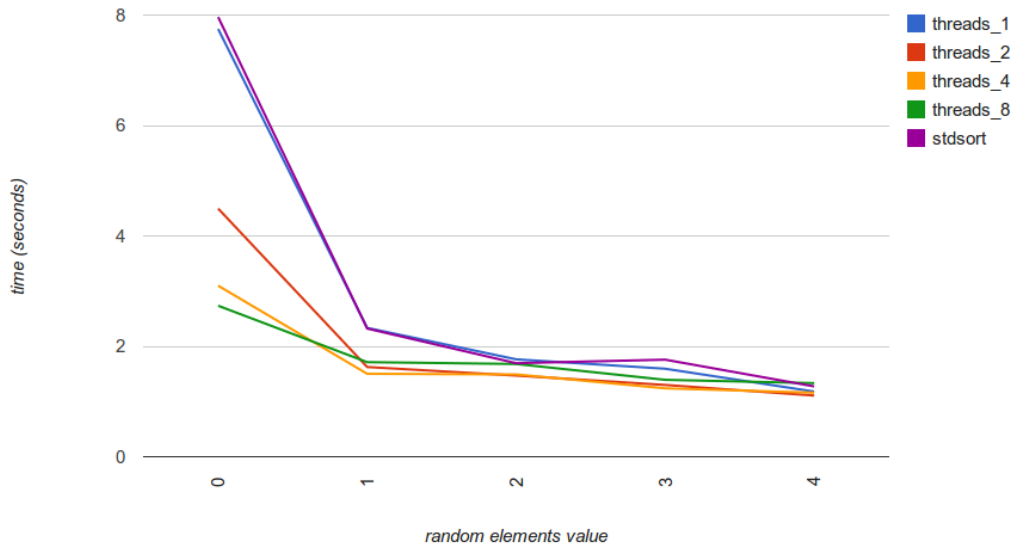

Figure 3: SO1 sorting performance comparision