Traveling Salesman

Benjamin Ryker

December 3, 2018

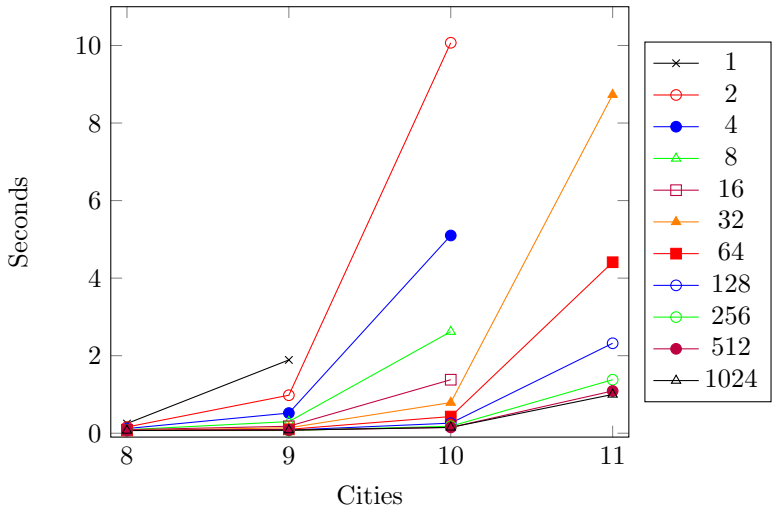
1 Results

This is the data I gathered on the runtime of the parallel implementation with 8-11 cities and 1-1024 threads. Fields marked as "-" represent tests that exceeded the maximum runtime of the CUDA Watchdog Timer (approx. 8-12 seconds).

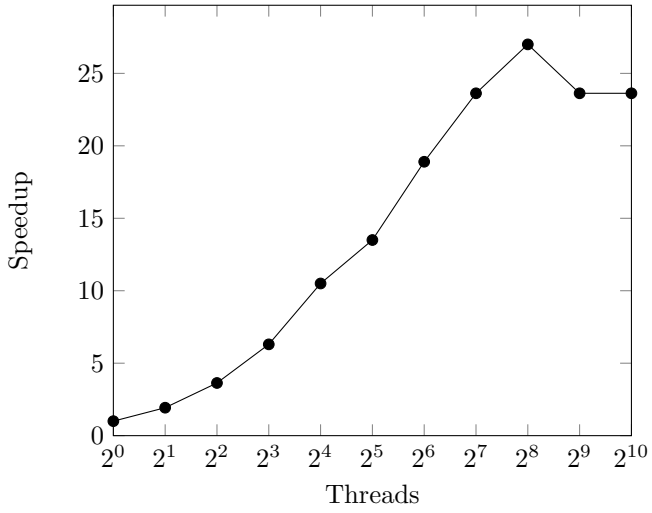
Threads	8	9	10	11	Speedup	Efficiency
1	00.25	01.89	-	-	1.00	1.00
2	00.16	00.98	10.07	-	1.93	0.96
4	00.12	00.52	05.10	-	3.63	0.91
8	00.09	00.30	02.62	-	6.30	0.79
16	00.09	00.18	01.38	-	10.50	0.66
32	00.07	00.14	00.79	08.73	13.50	0.42
64	00.07	00.10	00.43	04.41	18.90	0.30
128	00.07	00.08	00.26	02.32	23.63	0.18
256	00.07	00.07	00.18	01.38	27.00	0.11
512	00.07	00.08	00.15	01.09	23.63	0.05
1024	00.07	00.08	00.15	01.00	23.63	0.02

Since not all tests could be performed on a single thread and because 8 cities took less than a second on one thread, speedup and efficiency were only calculated for 9 cities.

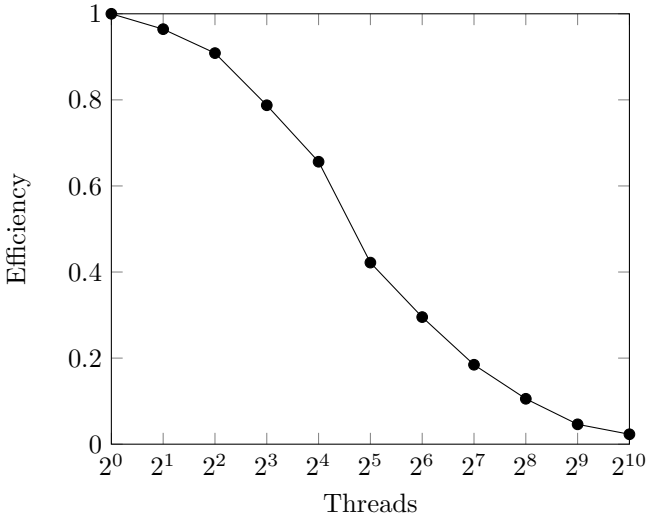
1.1 Time vs Number of Cities



1.2 Speedup



1.3 Efficiency



2 Conclusion

This algorithm consistently increases in performance as additional threads are added, but the timing method I used was not able to capture the exact time of very short runs. Also since 13! is greater than INT_MAX the algorithm cannot run on more than 13 cities. Although, at high numbers of cities the efficiency appeared to stay high, on low numbers of cities it dropped quickly