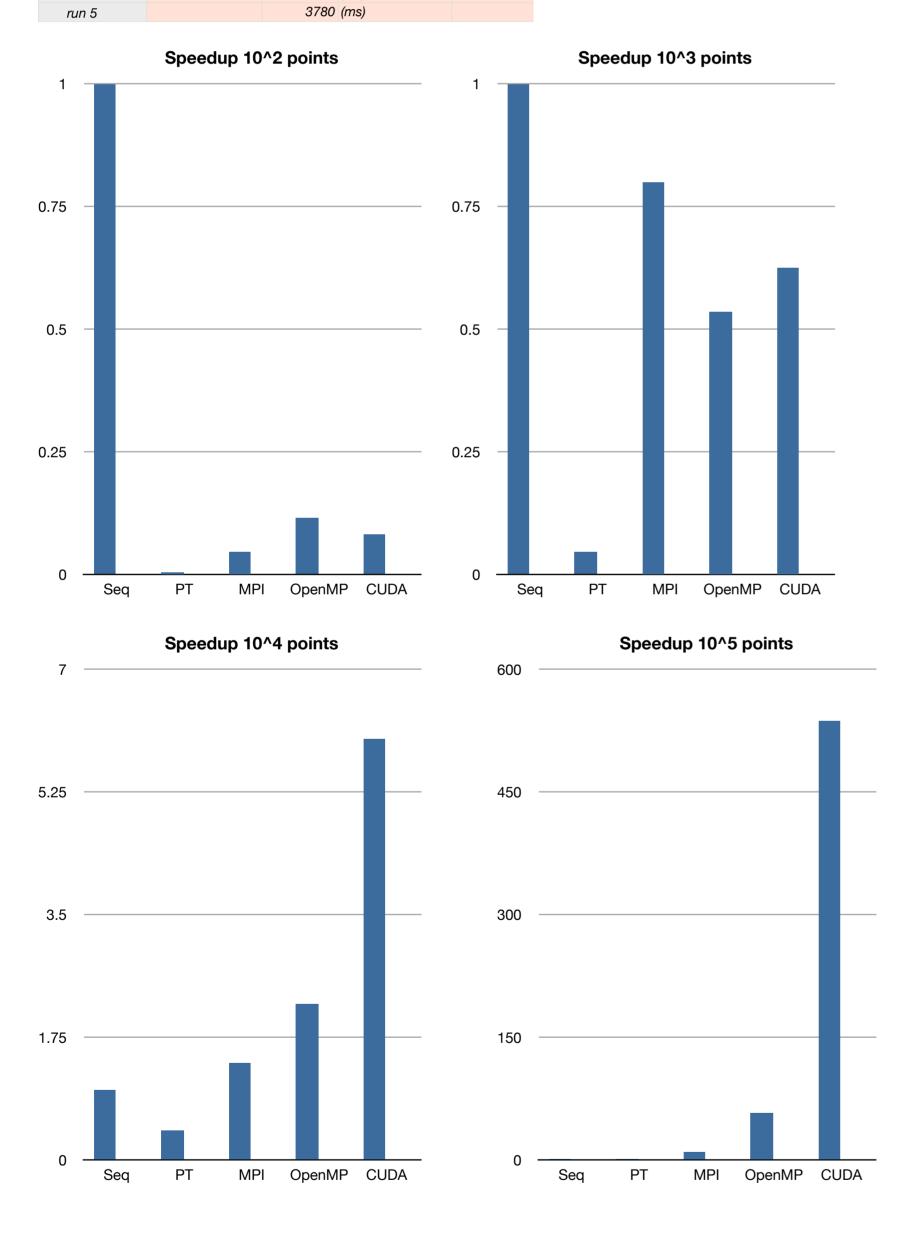
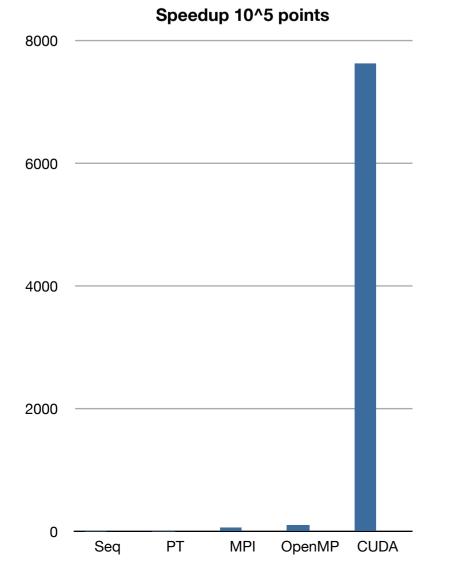
		Sequential on DAS				PTheads of	n DAS			
@ 100 000 step										
# points		Time Tim	ne Normalized	Spd	Info	Time	Time Normalized			
100		0.0272598	2.73E-09	1	6 threads	5.6740	9 5.67E-07	5E-03		
1000		0.264866	2.65E-09	1	6 threads	5.7501	4 5.75E-08	5E-02		
10000		2.92516	2.93E-09	1	6 threads	6.9434	9 6.94E-09	4E-01		
100000		67.3559	4.78E-08	1	6 threads	75.089	8 5.33E-08	9E-01		
1000000		344.238	2.83E-07	1	6 threads	148.45	7 1.22E-07	2E+00		
		MPI on DAS								
					OpenMP on DAS					
# points	Info	Time Tim	ne Normalized S	Speedup	Info	Time	Time Normalized	Speedup		
100	8 procs, 1 node	0.588229	5.88E-08		t=2, csize=10, static	0.23630	2 2.36E-08	1E-01		
1000	8 procs, 1 node	0.33138	3.31E-09		t=2, csize=10, guided	4.4944	5 4.94E-09			
10000	1 proc, 8 nodes	2.11306	2.11E-09		t=4, csize=10, guided					
100000	8 procs, 8 nodes	48.4409	4.84E-09		t=8, csize=10, guided					
1000000	1 proc, 8 nodes	447.843	4.48E-09	6E+01	t=8, csize=10, static	271.9	4 2.72E-09	1E+02		
		_								
	CUDA on DAS									
# points			ne Normalized S							
100 - avg	512 threads/block	0.3332	3.33E-08	8E-02						
run 1		572 (ms)								
run 2		230 (ms)								
run 3		320 (ms)								
run 4		213 (ms								
run 5	512 threads/block	331 (ms) 0.4236	4.24E-09	6E-01						
1000	512 threads/block	0.4236 144 (ms		0E-01						
run 1 run 2		126 (ms)								
run 3	In this overview and graph the results of the CUDA									
run 4		381 (ms			experiments are compared with the results from					
run 5		842 (ms			•	-	I, 2.1 and 3.1.ln			
10000	512 threads/block	0.4874	4.87E-10	6E+00						
run 1	354 (ms) the winner from each previous experiment is									
run 2		170 (ms)			selected to a	void an unn	ecessarily big ta	able.		
run 3		350 (ms)								
run 4		293 (ms			We can also s	see the time	does not increa	ase that		
run 5		1270 (ms)								
100000	512 threads/block	0.8902	8.90E-11	5E+02			points. Probab			
run 1		1560 (ms	:)				sm in the graphi			
run 2		486 (ms)			the time need	ded to trans	fer the data is re	esponsible		
run 3		1260 (ms	:)		for the most t	time.				
run 4		634 (ms)	:)							
run 5		511 (ms								
1000000	512 threads/block	3.712	3.71E-11	8E+03	3					
run 1		4000 (ms)								
run 2		3440 (ms)								
run 3		3450 (ms)								
run 4		3890 (ms	:)							





In the first and second graph (10^2 and 10^3 number of points) we can clearly see the overhead the various multithread systems cause are to high to gain a speedup. At this number of points all the added complexity is just slowing the calculations down.

However, from 10⁴ points the multithread implementations start to catch up and even outpace the sequential implementations. CUDA is faster than every other implementation.

The more points we add the higher the speedup gets.

