

CONCURRENCY & PARALLEL PROGRAMMING

Multithreaded Programs

*Auteurs: Tom Peerdeman &
René Aparicio Saez*

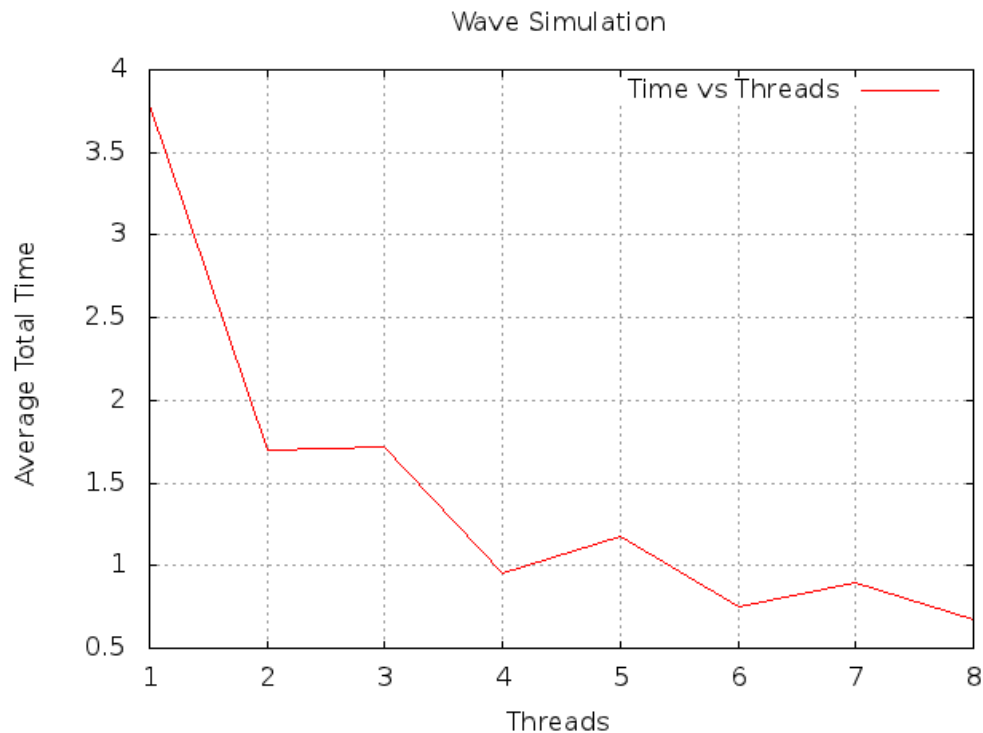
Datum: 09-11-2012

1 Assignment 1.1 - Wave simulation

1.1 Table with results

Tests on DAS4 are run for $i = 1.000.000$ and $t = 1.000$. The amount of threads used to generate the waves is increased to measure the difference in speed for the program. Each amount of threads is run 12 times. The highest value and the lowest value are disregarded. The remaining data is used to plot a graph.

i = 1,000,000				t = 1,000			
1 thread	2 threads	3 threads	4 threads	5 threads	6 threads	7 threads	8 threads
3.61657	1.68174	1.67706	0.967125	0.955026	0.707557	0.907312	0.681091
3.68809	1.69923	1.62901	0.952905	1.11722	0.783331	0.888922	0.677461
3.68735	1.68316	1.71819	0.946174	1.10978	0.802315	0.919069	0.652223
3.75564	1.71218	1.66693	0.92198	1.19389	0.722481	0.87717	0.736193
3.82117	1.70358	1.69521	0.951281	1.18303	0.791841	0.895334	0.656148
3.85017	1.69776	1.74229	1.28526	1.18654	0.69405	0.900644	0.661786
3.74723	1.70488	1.73762	0.972624	1.21719	0.794964	0.91415	0.666394
3.80248	1.71702	1.6256	0.9513	1.18918	0.746098	0.888725	0.951125
3.79577	1.69529	1.85265	0.949003	1.16967	0.789424	0.894571	0.66652
3.86725	1.70905	1.45095	0.958044	1.17816	0.716474	0.899676	0.710156
3.91954	1.70727	1.81612	0.964905	1.19274	0.733274	0.882675	0.640491
3.87399	1.70378	1.82773	0.960786	1.20634	0.682611	0.8860616	0.669534
Average of the remaining 10:							
3.788914	1.701978	1.713576	0.9564147	1.173655	0.7479494	0.89525656	0.6777506



Apparantly the performance is better if an even number of threads is used.

2 Assignment 1.2 - Sieve of Erasthotenes

On DAS4 (and on the uva computers) an error occurs. There are too many threads alive to create a new thread. This is understandable, because a new thread is created for every prime number found by the program. Which results in quite a high amount of threads. The program freezes and had to be terminated manually.