

Zeef van Eratosthenes

Hussin Almoustafa

Ontwerp :

Er wordt gebruikt gemaakt van de zeef van Eratosthenes die de primes getallen teruggeeft van elke meegegeven lijst.

In het implementatie wordt gebruikt gemaakt van OpenMP en MPI voor een parallellisatie van de loops.

Hierbij is er ook gebruikt gemaakt van een schedule dynamic loop die verdeelt de iteraties in brokken van grootte chunk-size.

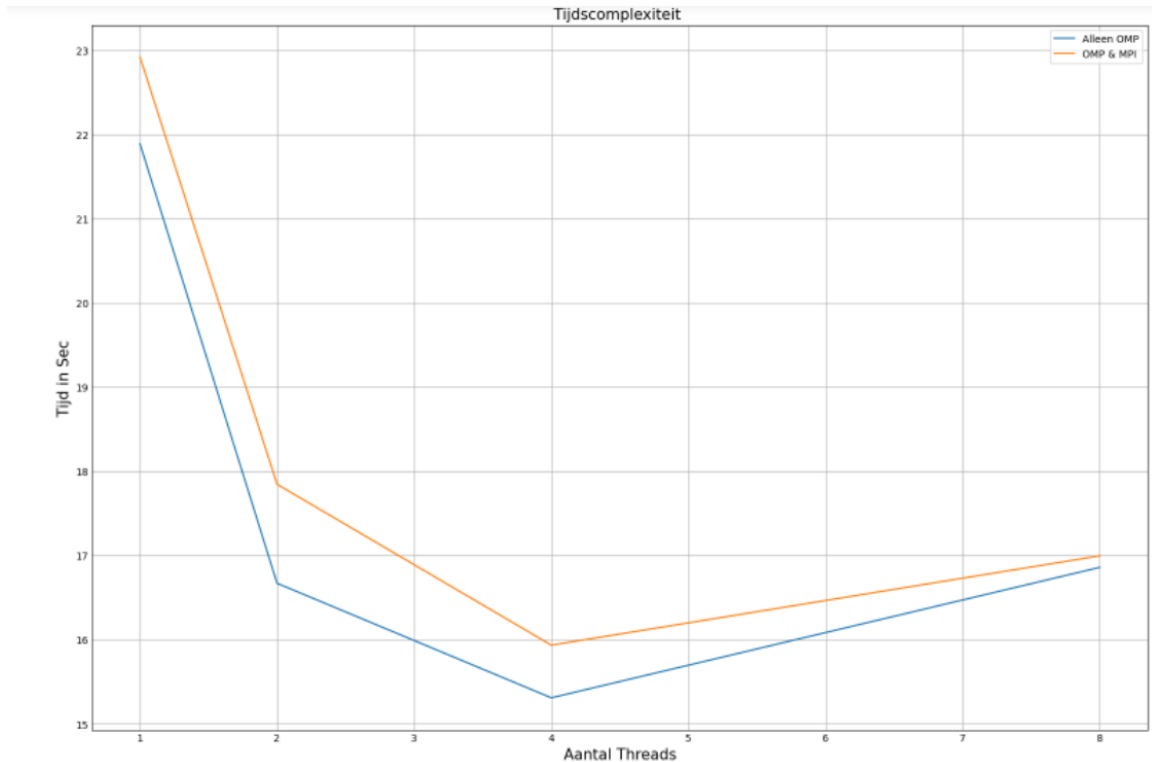
Bij MPI er is maar 1 procesor gebruikt . Er wordt ook bekijken naar runtime met verschillende threads (1, 2 ,4 en 8) zowel met alleen OMP als MPI&OMP.

De limit van de lijst die wordt aan gewerkt is (1000000 elementen).

code :

<https://github.com/hassoonsy2/High-Performance-Programming/blob/main/Zeef%20van%20Eratosthenes/sieve.cpp>

Tijdscomplexiteit & plot :



	A	B	C	D	E
1	Aantal Thr	Tijd in sec	Tijd in sec (MPI & OMP)		
2	1	22.918	21.889		
3	2	17.845	16.670		
4	4	15.935	15.310		
5	8	16.996	16.859		
6					
7					

Coloum A : Aantal Threads

Coloum B : run time van alleen OMP (limietwaarde van 1000000000 getal)

Coloum C : Runtime van MPI & OMP(limietwaarde van 1000000000 getal)

Conclusie

De combinatie van OMP en MPI maakt de Runtime iets sneller. Het kan zijn aan mijn implementatie ligt . Wat mij betreft is het maar een klein verschil in milliseconden.

Reflectie

Het maken van deze opdracht was niet zo makkelijk. Ook samen met andere twee opdrachten. Ik kon niet namelijk een tijd analyse afmaken. Ik snap het concept wel alleen het uitvoeren ging niet heel soepel. Mijn implementatie is ook niet het best.