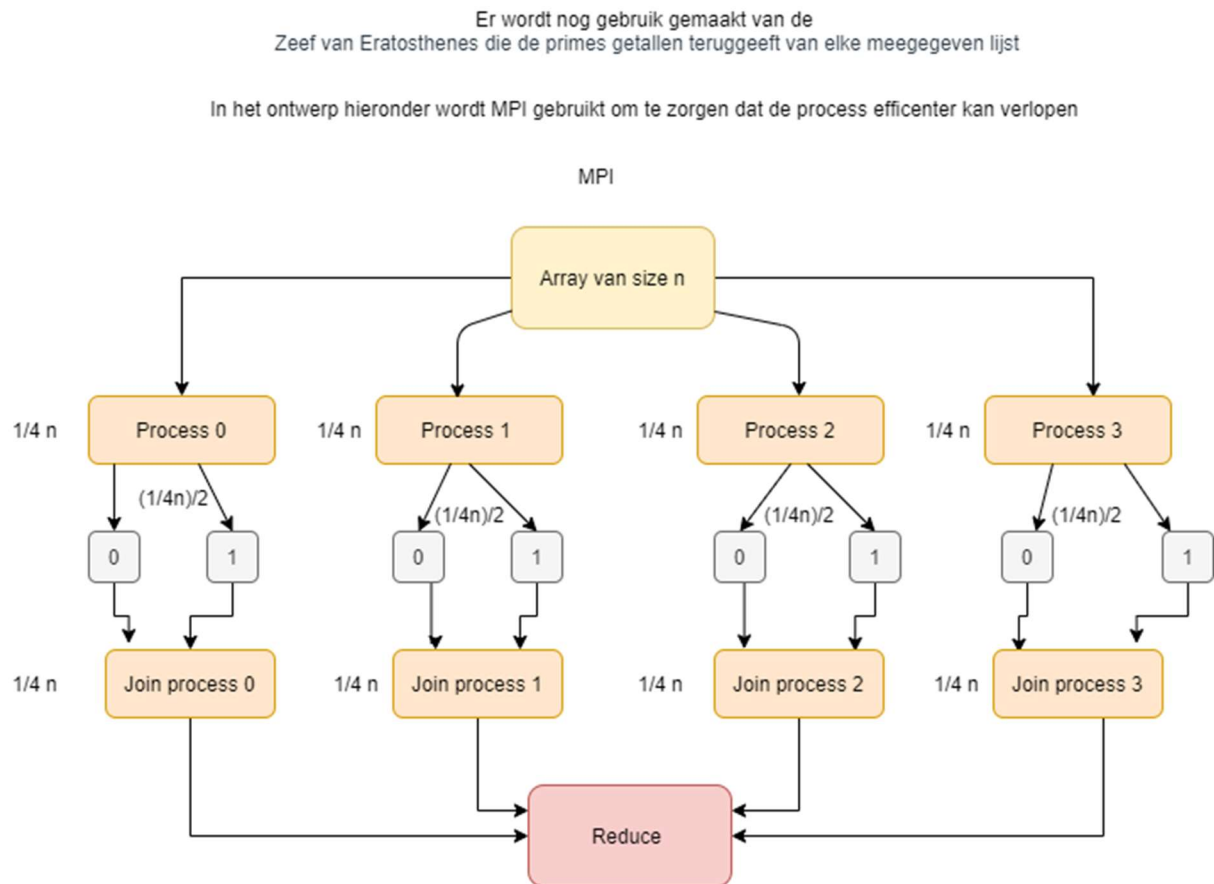


Analyse

1. Conceptueel ontwerp



Figuur 1: Ontwerp van het MPI en MP proces

2. Analyse performance winsten

MPI

N	1 node	2 nodes	4 nodes	8 nodes	Total
100	0.00059	0.00071	0.00314	0.01332	0.0177
1000	0.04024	0.01792	0.03030	0.01622	0.1046
10000	2.77844	1.87459	0.81184	0.71122	466.82
100000	152.981	85.4827	55.0889	53.0787	2.089.48

Figuur 2: Runtime van MPI code bij verschillende N

MPI en MP

N	1 process	2 process	4 process	8 process	Total
100	0.39490	0.43826	0.420763	0.307280	1.5612
1000	0.42969	0.49975	0.338518	0.329681	1.5976
10000	3.12201	3.56804	2.010084	0.842786	671.85
100000	5.69314	52.1859	24.22185	37.19366	1.452.59

Figuur 3: Runtime van MPI en MP code bij verschillende N

SEQUENTIAL

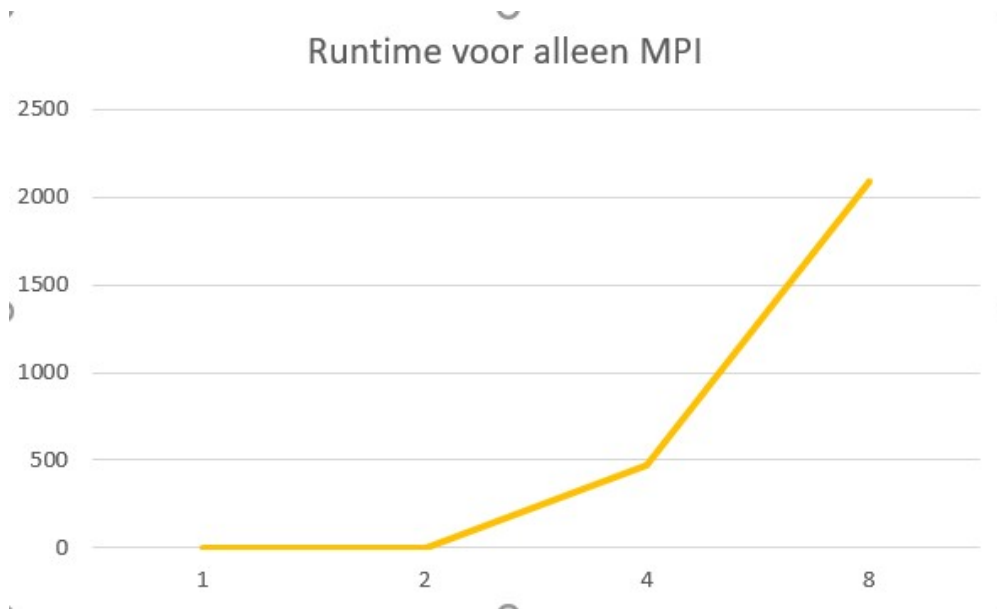
N	Sequential	Vectorized
100	0.0012	0.0000
1000	0.0173	0.0000
10000	1.2922	0.0012
100000	136.74	0.0036

Figuur 4: Runtime van de sequential zeef en vectorized zeef bij verschillende N

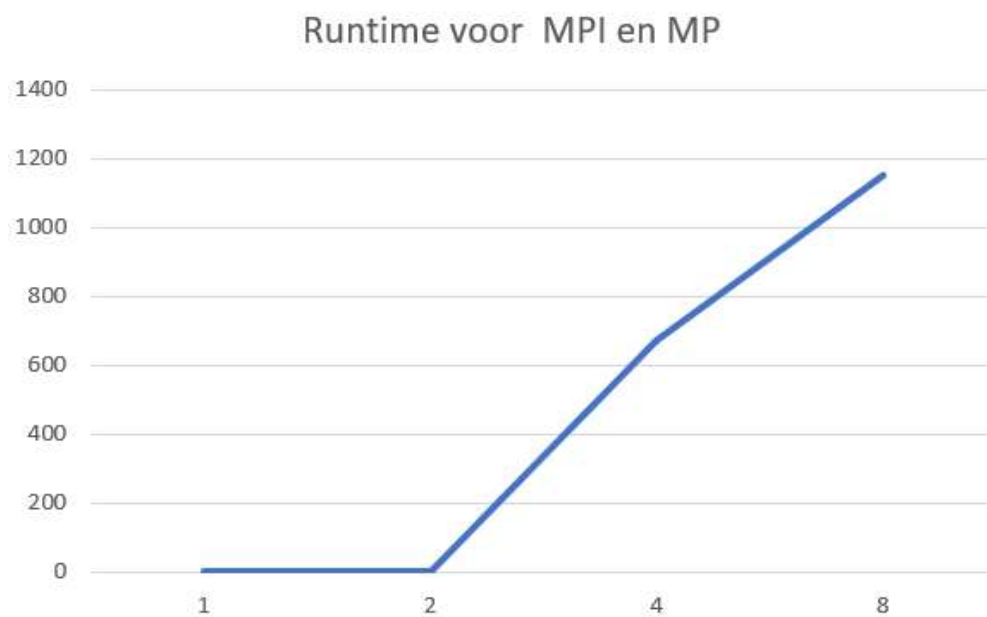
Uit alle vorige tabellen kunnen we zien dat het snelste implementatie de vectorized variant is. Het runnen van MPI en MP duurt hier het langste. Dit kan heel goed aan mijn code liggen of aan mijn laptop.

3. Plots

Hieronder zien we twee grafieken waar de runtime van alleen MPI en van MPI en MP geplot wordt



Figuur 5: Runtime van MPI code bij verschillende N



Figuur 6: Runtime van MPI en MP code bij verschillende N

4. Conclusie

Het toevoegen van de multiprocessing heeft volgens mij voegt geen verbetering aan het runnen van het proces. Wat mij betreft kan er beter gebruik gemaakt worden van mijn vectorized sieve ipv van mijn MPI en MP implementatie.

5. Reflectie

Het maken van deze opdracht was niet zo makkelijk. Ik snap het concept wel alleen het uitvoeren ging niet heel soepel. Mijn implementatie is ook niet het beste, geldt ook voor de resultaten die heb gehaald met de analyse. Het komt deels ook omdat ik helaas geen super laptop heb. Het was in iedere geval interessant om het algoritme sieve of Eratosthenes te leren kennen en nog zelf te maken. Ik heb waarschijnlijk ook veels te veel tijd ingestoken in het maken van de vectorized sieve, wat overigens best gelukt is.

Ik denk als ik meer tijd gehad heb, dan had ik waarschijnlijk deze opdracht beter kunnen maken qua code en analyse.