

Reflectie

Leerling: Sietse Neve

Leerlingnummer: 1810364

Github link: <https://github.com/DrZisnotavailable/HPP.git>

In dit vak heb ik geleerd hoe je programma's sneller maakt door middel van parallelisatie. Met OpenMP kun je makkelijk loops versnellen op één core met meerdere threads. MPI gebruik je als je meerdere processen of zelfs meerdere machines hebt.

Door de opdrachten ben ik gaan begrijpen hoe je werk kunt verdelen over processen (bijvoorbeeld bij de Julia Set en circuit satisfiability) en waarom het belangrijk is om goed te meten wat het oplevert qua tijd. Vooral bij de combinatie van OpenMP en MPI zie je dat een goede structuur echt uitmaakt.

Ik heb gezien dat het niet genoeg is om gewoon “meer threads” of “meer processen” te gebruiken — soms krijg je dan juist meer overhead. Daarom is het belangrijk dat je werk slim verdeeld is, en dat je synchronisatie en timing goed regelt.

Wat betreft complexiteit: de meeste algoritmes die we gebruikten, zoals brute-force checken van circuits of het renderen van fractals, hebben lineaire tot kwadratische tijdscomplexiteit ($O(n)$ of $O(n^2)$). Door parallelisatie verlaag je niet de Big-O zelf, maar verdeel je het werk — dus je krijgt **$O(n/p)$** waar p het aantal threads of processen is. Daardoor wordt het wel sneller, maar alleen als je overhead beperkt blijft.

Al met al snap ik nu beter hoe je software schaalbaar en efficiënter maakt, en hoe je dat test. Het was nuttig, ook al was het niet altijd even motiverend om mee bezig te zijn.