

PDS

OpenMP Convolutie

Randy Thiemann

1 november 2014

1 Aanpak

Ik heb voor de implementatie twee plaatsen gevonden om multi-threading toe te passen.

- In de binnenste for loops waar de hadamar operatie van de afbeelding en de kernel wordt uitgevoerd.
- In de buitenste for loops waar de convolutiefunctie wordt aangeroepen.

Voor het programma te testen op het VSC, heb ik beide methodes op mijn persoonlijke computer getest. Hier uit is gebleken dat de eerste methode veel minder effectief is dan de tweede methode, de eerste methode is 4 tot 5 keer trager afhankelijk van de scheduler, en dat als we deze combineren de code zelfs trager draait dan zonder enige OpenMP optimalisatie.

Ik heb dus besloten enkel de tweede methode toe te passen. Op mijn computer, met een 7216x5412 afbeelding en een 31x31 kernel, duurt het gemiddeld 44 seconden op zowel static, dynamic als guided scheduling.

2 Resultaten

Op het VSC heb ik met dezelfde afbeelding en kernel 10 runs gedaan met drie verschillende schedulers: static, dynamic en guided. Deze test werd herhaald met 1, 2, 4, 8, 16 en 20 cores. De gemiddelden en standaardafwijkingen zijn te vinden in onderstaande grafieken.

Hieruit blijkt dat tot 4 cores, op het vsc, guided de beste gemiddelde runtime geeft. Dan is voor 8 en 16 cores static het beste, eindigend met dynamic op 20 cores. De algemene trend is wel zoals verwacht.

In de standaardafwijkingen is over het algemeen de methode met het laagste gemiddelde ook deze met de laagste standaardafwijking.

