

Meetrapport: multi-threading snelheid

Stoeltie, Ferdi – 1665045

Agterberg, Ole – 1651981

26-03-2020

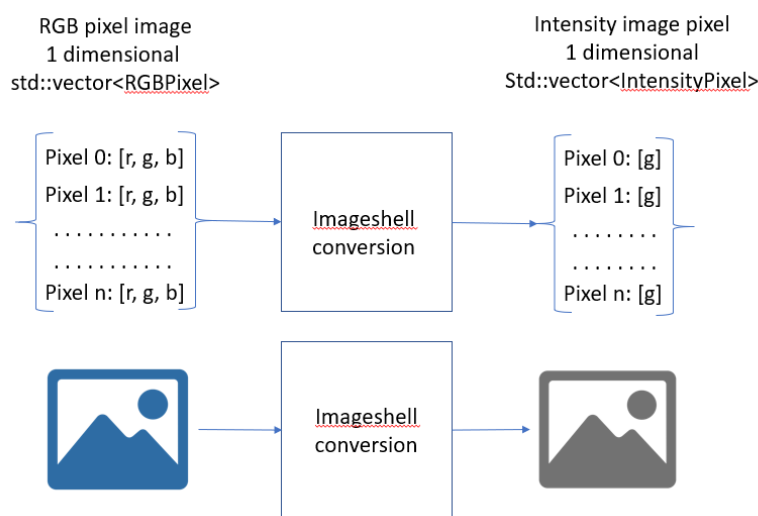


1. Doel

Dit meetrapport heeft als doel om de snelheid van de gekozen conversie methode te vergelijken wanneer dit parallel gedaan wordt door meerdere threads of alleen door de main thread. De resultaten van het meetrapport: “conversie snelheid”, zullen worden aangehaald om een vergelijking te trekken met de conversie snelheid d.m.v. multi-threading.

2. Hypothese

De verwachting is dat door het uitvoeren van de conversie methode volgens een multi-threaded aanpak, een behoorlijke snelheidswinst kan worden behaald. Wel is het van belang, dat het systeem meerdere beschikbare cores heeft om de conversie daadwerkelijk parallel te kunnen uitvoeren.

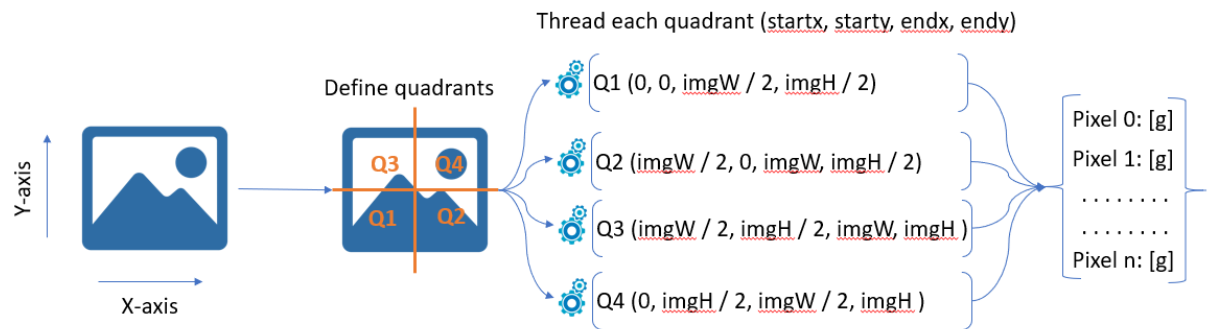


3. Werkwijze

De data wordt nog steeds opgeslagen in een (1-dimensionale) `std::vector` met een vaste grootte.

De grootte **N** word gedefinieerd door de hoogte **h** en breedte **b** te vermenigvuldigen ($N = h \cdot b$).

Er worden vier threads aangemaakt en elke thread neemt een kwart ($\frac{N}{4}$) van de vector om de conversie uit te voeren. Voordat de conversie begint, zal de tijd gemeten worden en ook direct na de conversie.



Optimalisatie

Er zal getest worden zonder optimalisatie evenals met optimalisatie: O2 (*optimized for speed*).

Systeem

Alle testen zijn uitgevoerd onder - zover als mogelijk - dezelfde omstandigheden.

Voor elke test zal de functie `executePreProcessingStep1` meerdere malen aangeroepen worden, het aantal samples. Bij elke afbeelding is het aantal samples 100, behalve bij de eekhoorn daar is het aantal 10. De tijd die nodig was om deze test uit te voeren wordt bijgehouden. Vervolgens wordt de test vijf keer herhaald voor de student klassen en vijf keer herhaald voor de default klassen. Hierna zal het verschil in tijd in percentage berekend worden door de volgende formule:

$$Snelheids\ verbetering = \left(\frac{Default_{time}}{Student_{time}} - 1 \right) * 100\%$$

Het programma kan op vier verschillende manieren gecompileerd worden;

- O2S -> O2 (optimized for speed), single-threaded;
- O2M -> O2 (optimized for speed), multi-threaded;
- DM -> Default (not optimized), multi-threaded;
- DS -> Default (not optimized), single threaded.

Variable Category	O2	Multi- threaded
O2S	✓	✗
O2M	✓	✓
DM	✗	✓
DS	✗	✗

4. Resultaten

De resultaten van de metingen zijn in de volgende tabellen terug te vinden. De resultaten zijn in nanoseconden genoteerd; 1 seconde = $1e+9$.

Grootte N gebruikte afbeeldingen:

Afbeelding	N Pixels (h*b)
Child-1	57375
Female-1	50310
Male-4	174592
Male-5	778680
Animal-1	2359296

Tabel met de gemaakte metingen. Dikgedrukte cijfers zijn de gemiddelde van de testen

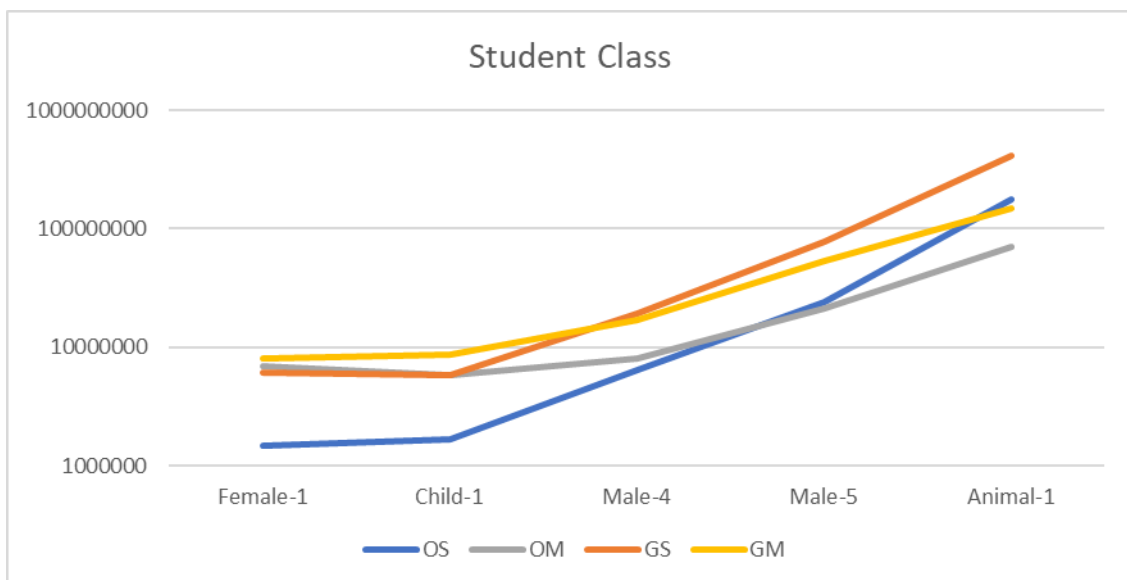
Student Class				
	OS	DS	OM	DM
Child-1	180864700	757285600	923948600	628110700
	162445400	640912900	598335200	1781656400
	161765500	517842800	569627700	700502000
	166638000	512369900	452688100	635744400
	168876000	461254900	357682100	537848800
	168117920	577933220	580456340	856772460
Female-1				
	167878700	838774000	516363600	1382683400
	149529500	665389600	1473162700	873731400
	137392500	586412900	693738200	552068300
	135789700	508044700	436786800	621217700
	137529700	490337700	381477200	622416900
	145624020	617791780	700305700	810423540
Male-4				
	554010500	2110608100	1332333900	2454656800
	525276400	2690071400	830082400	1484870600
	521435600	1668088900	648301600	1589449400
	652585400	1555005400	626913700	1532039500
	964585500	1572208800	552861400	1515979400
	643578680	1919196520	798098600	1715399140
Male-5				
	3210855600	7843665100	2407670200	5268675400
	2655215000	7286609500	3162210900	4508212100
	2072083100	7370217900	1775859300	5681585000
	2053688000	8442621900	1694748200	5773553800
	2046781100	8443485600	1687172600	5715325900
	2407724560	7877320000	2145532240	5389470440
Animal-1				
	1895250800	4759066600	1381439900	1612830200
	1732424600	3879787000	607916400	1406701700
	1775494200	4881481400	519001100	1428850700
	1735468700	3569592900	502822900	1451189400
	1734618300	3836421500	497771800	1492250800
	1774651320	4185269880	701790420	1478364560

5. Verwerking

De verwerkte resultaten zijn allemaal door het aantal samples gedeeld. Dus 100 bij child-1, female-1, male-4, male-5 en 10 bij animal-1.

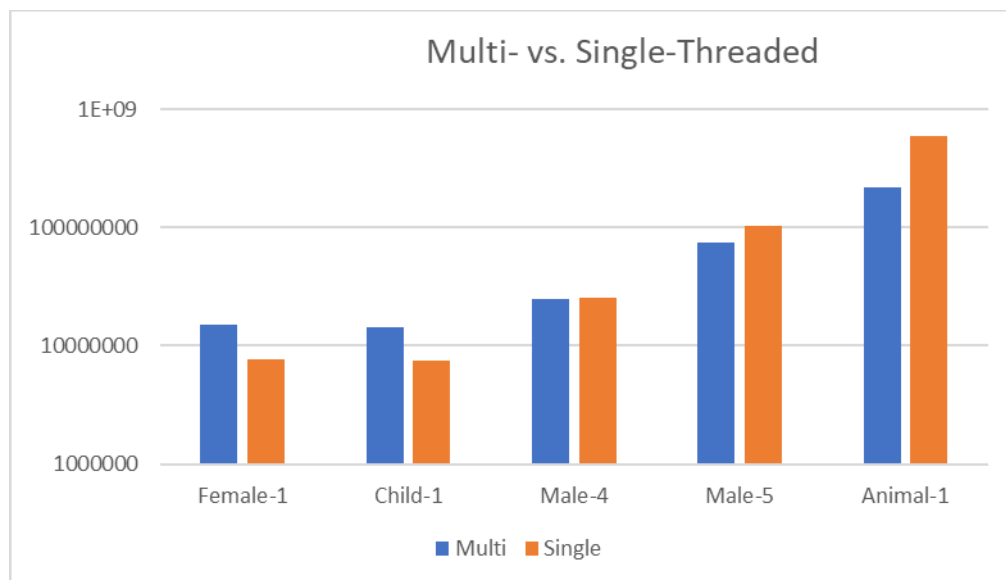
In de volgende tabel zijn de gemiddelde metingen van de testen te vinden. De bijbehorende grafiek laat zien dat zowel met als zonder optimalisatie de multi-threading sneller wordt. De grafiek is y-as logaritmisch.

	OS	GS	OM	GM
Female-1	1456240	6177918	7003057	8104235
Child-1	1681179	5779332	5804563	8567725
Male-4	6435787	19191965	7980986	17153991
Male-5	24077246	78773200	21455322	53894704
Animal-1	177465132	418526988	70179042	147836456



In deze tabel zijn de gemiddelde test resultaten weergegeven per afbeelding.

	Multi	Single	Verbetering
Female-1	15107292	7634158	-49%
Child-1	14372288	7460511	-48%
Male-4	25134977	25627752	2%
Male-5	75350027	102850446	36%
Animal-1	218015498	595992120	173%



Bij afbeeldingen met weinig pixels (~50k pixels). biedt de single-threaded aanpak betere prestaties. Bij grotere afbeeldingen (~170k pixels) biedt een multi-threaded aanpak betere prestaties. Ook bij deze grafiek is de y-as logaritmisch.

6. Conclusie

Uit de resultaten valt te concluderen dat het niet altijd beter is om de conversie multi-threaded uit te voeren. Bij afbeeldingen met een lage hoeveelheid pixels zal het vermoedelijk meer tijd kosten om de threads te alloceren, starten en de-alloceren dan dat het kost om de bewerking single-threaded uit te voeren. Ook is het van belang dat het systeem de vier threads parallel kan uitvoeren.