

Wissenschaftliches Rechnen III / CP III

Übungsblatt 5

Tizia Kaplan (545978)
Benjamin Dummer (532716)
Antoine Hoffmann (426675)
Gruppe 10

03.06.2016

Online-Version: https://www.github.com/BeDummer/CP3_UE5

Aufgabe 5.1

Hier sollte nun die CG Methode auf der GPU implementiert werden. Wir haben $N=32$ gewählt um die Anzahl der inneren Punkte zu definieren. Für die Reduktion wurde eine 1-dimensionale Konfiguration (block2 bzw. grid2) genutzt, alle anderen Kernel haben eine 2-dimensionale Execution Configuration (block bzw. grid). In `linalg.cu` haben wir noch die Kernel `vec_add_gpu` und `prod_gpu` hinzugefügt. Nach einer 0. Iteration geht die eigentliche Iteration zum Finden des Lösungsvektors x . Leider konvergiert unsere Implementierung noch nicht. Das Residuum wird stetig größer, welches eigentlich bis zur Toleranz `tol` stetig kleiner werden sollte. Offensichtlich ist noch ein grober Fehler in der Iteration, den wir leider nicht debuggen konnten. Allerdings wird für größere N (128, 512) das Residuum nur langsam größer. Offensichtlich hat die Anzahl der inneren Datenelemente einen starken Einfluss auf die gesamte Implementierung. Zwar ist die Rechenzeit auf der GPU noch nicht belastbar, da die Iteration noch nicht funktioniert. Allerdings wollen wir trotzdem theoretisch den Speedup darstellen. Dazu haben wir das gegebene CPU `run-cg` mit einer Zeitmessung modifiziert und somit eine Zeit t_{cpu} messen können. Ebenfalls bei unserer Implementierung wurde die GPU Rechenzeit aufgenommen. Somit hat sich exemplarisch für $N = 32$ ein Speedup von $Speedup = \frac{t_{cpu}}{t_{gpu}} = 0.5894$ ergeben. Somit ist momentan noch keine schnellere Rechnung auf der GPU implementiert.

Aufgabe 5.2

Es gilt das *Amdahlsche Gesetz* $S_p(N) \leq \frac{1}{f}$ zu zeigen. Es wird für jede Rechenoperation die Zeit Δ_T angenommen.

$$T_s(N) = N\Delta_T \quad (1)$$

$$T_p(N) = fN\Delta_T + \frac{(1-f)N\Delta_T}{p} \quad (2)$$

$$S_p(N) = \frac{T_s(N)}{T_p(s)} \quad (3)$$

$$\begin{aligned}
 &= \frac{N\Delta_T}{fN\Delta_T + \frac{(1-f)N\Delta_T}{p}} \\
 &= \frac{1}{f + \frac{1-f}{p}} = \frac{p}{f(p-1) + \textcolor{red}{1}} \\
 \text{da } f < 1, &\leq \frac{p}{f(p-1) + \textcolor{red}{f}} = \frac{p}{fp} = \frac{1}{f} \quad \square
 \end{aligned}$$

Von der vorletzten zur letzten Zeile wird im Nenner $f - 1$ addiert. Da $f < 1$ gilt, wird der Nenner durch diesen Schritt kleiner und somit der gesamte Term größer.

Anhänge

- Datei: `run-cg.cu` (Hauptprogramm)
- Datei: `linalg.h` und `linalg.cu` (hinzugefügte Kernelfunktionen)
- Datei: `run-cg-old.cu` (altes Hauptprogramm mit Zeitmessung)