

Übungsblatt 5

27.05.2016 / B. Leder

Wissenschaftliches Rechnen III / CP III

Aufgabe 5.1: *Konjugiere Gradienten auf der GPU*

Ziel ist eine Implementierung des Algorithmus 1 aus der Vorlesung auf der GPU.

Anforderungen / Hinweise:

- Der Algorithmus soll für beliebige Gittergrößen $N \geq 16$ ($N \times N$ innere Punkte) funktionieren (Beschränkt durch die Speichergröße).
- Um große Latenzen zu vermeiden, soll das Kopieren zwischen Host und Device auf einen Vektor am Anfang, einen Vektor am Ende beschränkt sein. Während einer Iteration sollen nur maximal M `double` Zahlen übertragen werden, wobei M unabhängig von N und viel kleiner als das maximal mögliche N sein soll.
- Je nach Kernel können verschiedene execution configuration benutzt werden. Zum Beispiel eine 1-D für die Reduktion und eine 2-D für alles andere.

Die Implementierung kann ausgehend von Ihren Programmen aus den vorherigen Übungsblättern erfolgen oder Sie starten von dem Paket [uebung5.zip](#). Dies enthält ein `README.md`, welches erklärt, wie die Dateien darin kompiliert werden. Sie können sich auch aus den Dateien Teile kopieren, um Ihre eigenen Programme zu ergänzen.

Messen Sie den Speedup für 10 Iteration als Funktion von N für eine sinnvolle/optimale execution configuration.

10 Punkte

Aufgabe 5.2: *Amdahlsches Gesetz*

Definitionen:

- $T_s(N)$, Zeit die das beste serielle Programm für ein Problem der Größe N benötigt
- $T_p(N)$, Zeit die ein paralleles Programm auf p Prozessoren für ein Problem der Größe N benötigt
- $S_p(N) = T_s(N)/T_p(N)$, ist der Speedup

Nehmen Sie an, dass jede Operation eine Zeiteinheit benötigt und das ein Teil $0 < f < 1$ der Operationen sequentiell ausgeführt werden müssen, also nicht parallelisiert werden können.

Beweisen Sie das Amdahlsches Gesetz

$$S_p(N) \leq 1/f$$

5 Punkte