

Algorithm Engineering - Graphikprozessoren SS 2010

Übungsblatt 1

http://algo2.iti.kit.edu/cuda_10.php

{luxen|osipov|schieferdecker}@kit.edu

Ausgabe: Dienstag, 13.04.2010

Abgabe: Montag, 19.04.2010 (**verlängert bis 25.04.2010**)

Korrektur (19.04.2010): Maximalwerte für n in Aufgabe 3 herabgesetzt

Aufgabe 1 (*Einführung*)

Am Institut steht ein Rechenknoten mit dedizierten GPGPU Karte (*general-purpose computing on graphics processing units*) zur Verfügung (i10pc123). Außerdem wird voraussichtlich im Laufe der Woche ein weiterer Rechenknoten mit einer GPGPU Karte aufgebaut (i10pc12x). Jeder Teilnehmer erhält Zugangsdaten, um sich auf unseren Poolrechnern (Raum 205) und den beiden Rechenknoten vor Ort bzw. über SSH anzumelden und zu arbeiten. Desweiteren steht jedem Teilnehmer ein *svn-Repository* unter zur Verfügung (Loginnamen, Adressen und Passwörter werden am Einführungstermin bekannt gegeben).

Hinweis: Wir beschränken uns auf die Unterstützung von linuxbasierten Betriebssystemen.

- a) Melden Sie sich auf einem Institutsrechner an und ändern Sie Ihr Passwort (mit `passwd`).
- b) Laden Sie die Vorlage `report.zip` aus Ihrem Repository. Verwenden Sie diese im Folgenden, um Protokolle der Praktikumstermine mit $\text{\LaTeX} 2_{\epsilon}$ zu erstellen. **Stellen Sie fertige Protokolle (PDF) für jeden Termin bis Mitternacht vor dem nächsten Termin ins Repository.** Unterlagen zum Umgang mit `svn` finden Sie z.B. unter <http://subversion.tigris.org>. Unter KDE kann auch die Anwendung `kdesvn` verwendet werden.
- c) Unter den folgenden Adressen finden Sie Unterlagen zur Verwendung von CUDA. Machen Sie sich mit ihnen vertraut soweit es für die Lösung der Aufgaben notwendig ist:
 - http://developer.nvidia.com/object/cuda_3_0_downloads.html/
 - <http://llpanorama.wordpress.com/cuda-tutorial/>
- d) (*Zusatz-freiwillig*) Installieren Sie auf Ihrem Heimrechner das CUDA 3.0 Toolkit (zu finden auf http://developer.nvidia.com/object/cuda_3_0_downloads.html). Falls Sie eine CUDA fähige Graphikkarte besitzen, vergewissern Sie sich auch, dass ein CUDA-aktivierter Graphikkartentreiber installiert ist. Falls Sie keine CUDA fähige Graphikkarte besitzen, können Sie im Folgenden den Emulationsmodus verwenden.

Aufgabe 2 (*"Hello World"*)

Als Einstieg in die Programmierung mit CUDA sollen zunächst zwei kleine Anwendungen geschrieben werden, die in der Literatur des Öfteren als "Hello World" Anwendung in CUDA bezeichnet werden.

- a) Ermitteln Sie zunächst mit Hilfe von CUDA-Funktionsaufrufen, welche GPGPU Karte in i10pc123 verbaut ist (`cudaGetDeviceCount`) und ermitteln Sie die sonstigen Eigenschaften dieser Karte (`cudaGetDeviceProperties`).
- b) Das zweite Einführungsprogramm soll Quadratzahlen berechnen und dabei die Parallelverarbeitung der Graphikkarte ausnutzen.
 - 1) Schreiben Sie ein Programm, das die Quadratzahlen von 1 bis 500 mit Hilfe der Graphikkarte berechnet.
Hinweis: Machen Sie sich mit der Übertragung von Daten zwischen Haupt- und Graphikkartenspeicher und der Ausführung eines *kernels* vertraut.
 - 2) Erweitern Sie Ihr Programm, so dass es die Quadratzahlen von 1 bis n , $n \leq 10\,000\,000$ berechnet. Ab welchem Wert für n müssen Sie aufpassen und warum?
Hinweis: Sie müssen die Verarbeitung in mehrere Blöcke aufteilen.
 - 3) Messen Sie die Laufzeit für $n \in \{10, 20, 100, 200, 1\,000, 2\,000, \dots, 10\,000\,000\}$. Zeit für das Übertragen von Daten zwischen Haupt- und Graphikkartenspeicher muss nicht berücksichtigt werden. Verwenden Sie zur Zeitmessung die auf der Homepage zur Verfügung gestellte Klasse `timer`.
 - 4) Vergleichen Sie die Laufzeiten zu einer Ausführung im Emulationsmodus (kompilieren mit `nvcc -deviceemu`). Erstellen und beschreiben Sie hierfür ein Diagramm mit `gnuplot`. Sie können das bereitgestellte Skript verwenden. Weitere Unterlagen zu `gnuplot` finden Sie unter <http://www.gnuplot.info/>.
- c) (*Zusatz-freiwillig*) Ermitteln Sie auch die Eckdaten der GPGPU Karte in unserem zweiten Rechenknoten (sobald dieser verfügbar wird) und in Ihrem Heimrechner. Wie ist der Geschwindigkeitsunterschied bei der Berechnung der Quadratzahlen zwischen den drei Rechnern. Treten beim Ausführen neue Probleme auf?

Aufgabe 3 (Multiplikation von Matrizen)

Im Folgenden sollen verschiedene Ansätze zur Multiplikation von Matrizen untersucht werden. Der Einfachheit halber gehen wir von quadratischen, dicht besetzten Matrizen der Größe $n \times n$ aus, $n \leq 18\,000$.

- a) Implementieren Sie einen Algorithmus zur Multiplikation zweier Matrizen auf der CPU. Beschreiben Sie die Funktionsweise Ihres Ansatzes.
- b) Implementieren Sie einen CUDA-Algorithmus zur Multiplikation zweier Matrizen. Beschreiben Sie wieder Ihren Ansatz.
- c) Untersuchen Sie jeweils die folgenden Laufzeiteigenschaften:
 - 1) Messen Sie die Laufzeit bei mehreren Durchläufen und festem Parameter n . Wie stark schwanken die Ergebnisse? Welche Probleme ergeben sich für $n > 18\,918$ auf i10pc123? Wie könnte man dieses Problem umgehen?
Hinweis: Als Faustregel sollte eine Messung mehrere Sekunden laufen.
 - 2) Messen Sie die Laufzeit für $n = \{3, 10, 30, 100, \dots, 18\,000\}$. Plotten Sie die Mittelwerte und Varianzen einer genügend großen Anzahl an Ergebnissen. Passen Sie hierfür das Skript aus Aufgabe 2 an.

Füllen Sie die Matrizen mit gleichverteilten Zufallszahlen aus $[0, 1\,000]$.

- d) (*Zusatz-freiwillig*) Implementieren Sie eine Variante von Aufgabe 3b, die für Matrizen mit $n > 18\,918$ funktioniert.
- e) (*Zusatz-freiwillig*) Vergleichen Sie wieder die Laufzeiten mit Ihrem Heimrechner oder unserem zweiten Rechenknoten (ggf. muss der maximale Wert von n kleiner gewählt werden).

Lernziele – Woche 1

Nach Bearbeitung des vorliegenden Übungsblattes sollten Sie mit folgenden Themen vertraut sein:

- Rudimentärer Umgang mit `svn`, `gnuplot` und $\text{\LaTeX} 2_{\epsilon}$
- Kompilieren einer Anwendung zur Verwendung auf GPGPU Karten
- Datenübertragung zwischen Haupt- und Graphikkartenspeicher
- Threadmodell in CUDA