

Übungsblatt 5, für den 27.04.2018

Aufgaben:

1. Erstellen sie ein sequentielles C Programms welches
 - a. Einen Eingabe-Parameter N entgegen nimmt
 - b. Einen Array von Bytes mit N Elementen anlegt
 - c. Diesen Array mit Zufallszahlen 0/1 befüllt
 - d. Die Anzahl an 1 Einträgen ermittelt
2. Parallelisieren Sie Schritt 1d) mit OpenMP
3. Parallelisieren Sie Schritt 1d) mit OpenCL unter Verwendung des *Recursive Multi-stage Reduction* Verfahrens aus der Vorlesung
4. Testen Sie Ihre Versionen durch vergleichen der Ergebnisse für verschiedene Ns und random seeds.
5. Vergleichen Sie die Performance der Implementation 1d, 2 und 3 für verschiedene N, sodass 1, 2, 3 ... Stages notwendig sind, bis das GPU global memory limit erreicht ist;

Vorgehensweise (Hinweis):

1. Für Aufgabe 1:
 - Verwenden sie rand() aus <stdlib.h> für die Erzeugung der Zufallszahlen
 - Um Verschiedene Test-Cases zu erzeugen variieren Sie den seed (srand).
2. Für Aufgab 5:
 - Kompilieren sie Version 1 und 2 mit aktiven Compiler-Optimierungen.
 - Messen Sie für die OpenCL Version nur die Rechenzeit on-Device, nicht den Datentransfer;

Ziel:

1. Eigenständiges entwickeln eines Programms ohne Vorlage
2. Implementation einer Reduction mit OpenCL

Abgabe:

- Per email an herbert.jordan@uibk.ac.at, 1 Abgabe pro **Gruppe**
Betreff: “[PS703106] [UE05] GR_XX - NAME1 NAME2 NAME3”
keine Binaries (make clean)
Vor (!) Übungsbeginn