

Institut für Informatik  
Henning Wenke M.Sc.  
Sascha Kolodzey B.Sc.  
Nils Vollmer B.Sc.

Universität Osnabrück  
<http://www-lehre.inf.uos.de/~pa/>

## Übung: Parallele Algorithmen mit OpenCL

*Sommersemester 2013*

### Blatt 10

Laden Sie sich zur Bearbeitung der Aufgaben das bereitgestellte Framework unter folgendem Link [http://www-lehre.inf.uos.de/~pa/Uebungen/Blatt10/PA\\_Blatt10.zip](http://www-lehre.inf.uos.de/~pa/Uebungen/Blatt10/PA_Blatt10.zip) herunter.

#### **Aufgabe 10.1 RADIXSORT (40 Punkte)**

Implementieren Sie das in der Vorlesung vorgestellte Sortierverfahren RADIXSORT, wie auf der Folie *Paralleles Radix Sort (LSD)* beschrieben. Es sollen hierbei nur Zahlen zwischen 0 und 803 sortiert werden können. Messen Sie die Kernel-Laufzeiten mit der Methode `measureKernelCall()` innerhalb der Klasse `CLUtil`. Überlegen Sie sich Möglichkeiten, die Ausführungszeit des Algorithmus zu beschleunigen. Die schnellste Lösung wird ausgezeichnet!

**Hinweis:** Die Verwendung von *Local-Memory* ist nicht zulässig.

#### **Aufgabe 10.2 BUCKETSORT (30 Punkte)**

Implementieren Sie den aus Info-A bekannten BUCKERSORT Sortier-Algorithmus <http://www-lehre.inf.uos.de/~ainf/2012/Vorlesung/BucketSort.java> parallel. Das Eingabealphabet soll dabei auf 256 Zeichen beschränkt bleiben. Messen Sie die Kernel-Laufzeiten für Eingaben verschiedener Größen und vergleichen Sie diese mit der Implementierung aus der Info-A Vorlesung. Die schnellste Lösung wird ausgezeichnet!

**Hinweis:** Die Verwendung von *Local-Memory* ist nicht zulässig.

#### **Aufgabe 10.3 Tiefensortierung (20 Punkte)**

Sie haben die Aufgabe bekommen, Objekte innerhalb des kontinuierlichen Würfels  $K^3 = \{(x_1, x_2, x_3) \mid x_i \in [-a, a]\}$  gemäß der Tiefe zu sortieren. Sortiert werden soll auf der  $x_3$ -Achse zwischen  $-a$  (am kleinsten) bis  $a$  (am größten). Überlegen Sie sich theoretisch, wie Sie dieses Problem parallel lösen können.

**Aufgabe 10.3 BITONICSORT (10 Punkte)**

Erklären Sie Ihrem Tutor die Funktionsweise des *Bitonic-Merge-Sort* Algorithmus aus der Vorlesung.