

## Übung: Parallele Algorithmen mit OpenCL

*Sommersemester 2013*

### Blatt 7

Laden Sie sich zur Bearbeitung der Aufgaben die aktuelle Version des OpenCL Wrappers unter folgender URL [http://www-lehre.inf.uos.de/~pa/Uebungen/Blatt7/PA\\_Blatt7.zip](http://www-lehre.inf.uos.de/~pa/Uebungen/Blatt7/PA_Blatt7.zip) herunter. Für die Aufgaben steht bereits eine Templateklasse zu Verfügung, die Ihnen die OpenCL Initialisierung und das Erstellen der benötigten Basisobjekte abnimmt. Außerdem ist es jetzt möglich die Auswahl von Device und Platform stärker zu beeinflussen. Sollten Sie bis jetzt Probleme mit der automatischen Auswahl von Device und Platform gehabt haben, können Sie sich die Klasse `PlatformDeviceFilterUsage` ansehen.

### Aufgabe 7.1 REDUCTION: Maximum (55 Punkte)

In dieser Aufgabe sollen Sie einen REDUCTION Algorithmus implementieren, der als Operator die Maximumsfunktion verwendet. Die Ausgangsdaten sollen aus einer positiven Folge von  $N$  Integerwerten bestehen. Die Aufgabe gliedert sich in drei Teilaufgaben, die jeweils für sich ausführbar sein sollen. Legen Sie dementsprechend für jede Aufgabe eine neue Java-Klasse an, wobei Sie die Kernel in das gleiche Programm schreiben können:

1. Implementieren Sie zuerst den Algorithmus so, wie auf Folie 19 der Vorlesungsfolien beschrieben. Machen Sie sich deutlich, dass der Kernel mehr als einmal ausgeführt werden muss, um das Ergebnis zu berechnen. Beachten Sie bitte, dass auch eine ungerade Anzahl von Werten korrekt verarbeitet wird.
2. Als zweite Variante soll die Reduktion mit einem einzigen Kernel-Aufruf durchgeführt werden. Sie können Folie 25 hierbei als Hilfestellung nehmen. Verwenden Sie jedoch nicht, wie auf der Folie 25,  $N$  Threads sondern  $N/2$  Threads. Berücksichtigen Sie außerdem den Fall, dass eine ungerade Anzahl zu verarbeitender Werte vorliegt. Beachten Sie, dass die Größe einer Workgroup durch die jeweilige Hardware beschränkt ist. Bei der GTS 450 (FIRMI) liegt die maximale Größe bei 1024 Threads. Auf der GTS 450 könnten Sie also maximal 2048 Werte verarbeiten. Wenn Sie wissen

möchten, welches die maximale Workgroupsize Ihrer Hardware ist, können Sie sich mit Hilfe der Funktion `printDeviceInfo` der Klasse `CLUtil` diese ausgeben lassen. Wählen Sie den Datensatz entsprechend einer möglichen Workgroupsize.

3. Die dritte Variante soll nicht mehr auf eine Workgroup beschränkt bleiben. Dieser Ansatz ist genau dann notwendig, wenn es sich um eine Anzahl von Werten handelt, die nicht mehr mit einer einzigen Workgroup verarbeitet werden kann/sollte. Machen Sie sich klar, dass, wie in Variante 1, wieder mehrere Kernel-Aufrufe nötig sind, um die Reduktion vollständig durchzuführen. Auch hierbei ist der Fall einer ungeraden Anzahl von Werten zu beachten.

Geben Sie für alle drei Varianten die Anzahl der benötigten Kernel-Aufrufe in Abhängigkeit der Daten und der gewählten Workgroupsize an. Außerdem soll die verwendete Zahlenfolge sowie der maximale Wert auf der Konsole ausgegeben werden.

**Hinweis:** Erfahrungsgemäß ist es zu Anfang einfacher, den Algorithmus für Daten mit einer Zweierpotenz Anzahl von Elementen zu implementieren. Wenn dies erfolgreich geschehen ist, können Sie sich den allgemeinen Fall anschauen.

### **Aufgabe 7.2 Matrixmultiplikation mit Hilfe von REDUCTION (30 Punkte)**

Als Beispiel für die Nichteindeutigkeit der Parallelisierung von Problemen, soll die Matrixmultiplikation von Blatt 4 noch einmal mit Hilfe des REDUCTION Algorithmus und dem Operator `+` gelöst werden. Bevor Sie die Reduktion anwenden können, benötigen Sie zuerst alle skalaren Multiplikationen, um anschließend das Skalarprodukt mit Hilfe von REDUCTION zu berechnen. Schreiben Sie sich hierzu einen Kernel, der alle nötigen Multiplikationen zur Bildung der jeweiligen Skalarprodukte berechnet und passend für die Reduktion in einen Buffer schreibt. Machen Sie sich klar, dass es sich hierbei um keine vollständige Reduktion über die gesamten Daten handelt, sondern diese Reduktion lediglich abschnittsweise pro Skalarprodukt geschehen muss. Falls Sie Variante 3 aus Aufgabe 7.1 bereits implementiert haben, können Sie diese hier direkt verwenden. Sie müssen jedoch darauf achten, den Algorithmus rechtzeitig abubrechen. Geben Sie das Ergebnis auf der Konsole aus.

### **Aufgabe 7.3 Fragen: Parallelität (15 Punkte)**

Beantworten Sie ihrem Tutor Fragen zum Inhalt der Vorlesung.