

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 11

Laden Sie sich zur Bearbeitung der Aufgabe das bereitgestellte Framework unter folgendem Link http://www-lehre.inf.uos.de/~pa/Uebungen/Blatt11/PA_Blatt11.zip herunter.

Aufgabe 11.1 Parallel Prefix-Sum (SCAN): Local Memory (100 Punkte)

Implementieren Sie den aus den letzten Vorlesungen bekannten SCAN Algorithmus unter der Verwendung von *Local Memory*. Führen Sie zuerst den gesamten Algorithmus innerhalb einer Workgroup auf doppelt so vielen Daten wie Work-Items durch. Jedes Work-Item muss also zuerst zwei Zahlen in das *Local Memory* laden, bevor die Berechnung starten kann. Machen Sie sich klar, dass das *Local Memory* doppelt so viele Elemente speichern muss wie Work-Items existieren. Benutzen Sie zum Setzen der *Local Memory* Größe die Funktion `clSetKernelArg(kernel, index, (long)localWorksizeBytes)`. Beachten Sie, dass das letzte Argument eine *long* Variable sein muss, die den Speicher in Byte angibt. Alle Zwischenergebnisse sollen wieder im *Local Memory* hinterlegt werden, sodass am Ende der Berechnung jedes Work-Item genau zwei globale Schreibzugriffe ausführen muss. Als zweiten Schritt sollen Sie sich überlegen, den Algorithmus für größere Datenmengen, die nicht von einer Workgroup verarbeitet werden können/sollten, zu erweitern. Als Tipp hierfür dient das folgende Schema:

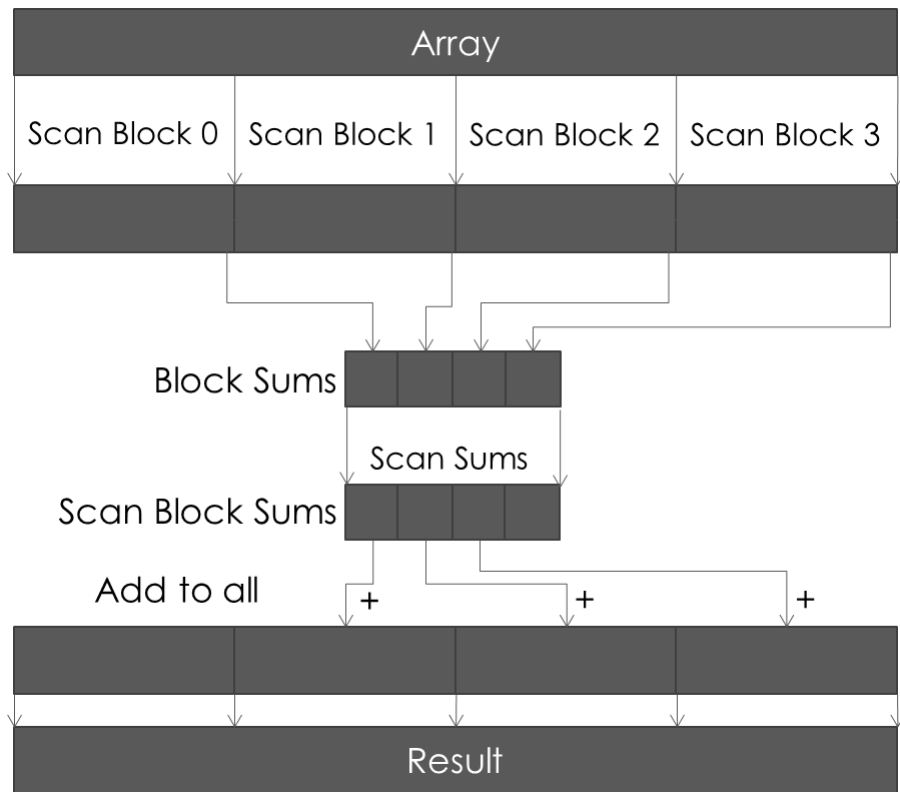


Figure 1: Scan für große Arrays

Zuletzt sollen Sie den Implementierten SCAN gegen eine zweite Version Ihres Algorithmus testen, die nur auf dem *global Memory* arbeitet. Messen Sie die Laufzeiten und vergleichen Sie.