

Übung 4

Computational Physics III

Matthias Plock (552335)

Paul Ledwon (561764)

29. Mai 2018

Inhaltsverzeichnis

1 Aufgabe 1	1
1.1 (Aufgabenteil i)	
Skizzierung der Reduktion	1
1.2 (Aufgabenteil ii, iii, iv)	
Messung der Laufzeiten, Global Memory Load Efficiency, Global Load Throughput und Achieved Occupancy	2
1.3 (Aufgabenteil v, vi)	
Optimaler Wert fuer die Anzahl der Elemente q, die vor Reduktion summiert werden . . .	3

1 Aufgabe 1

1.1 (Aufgabenteil i)

Skizzierung der Reduktion

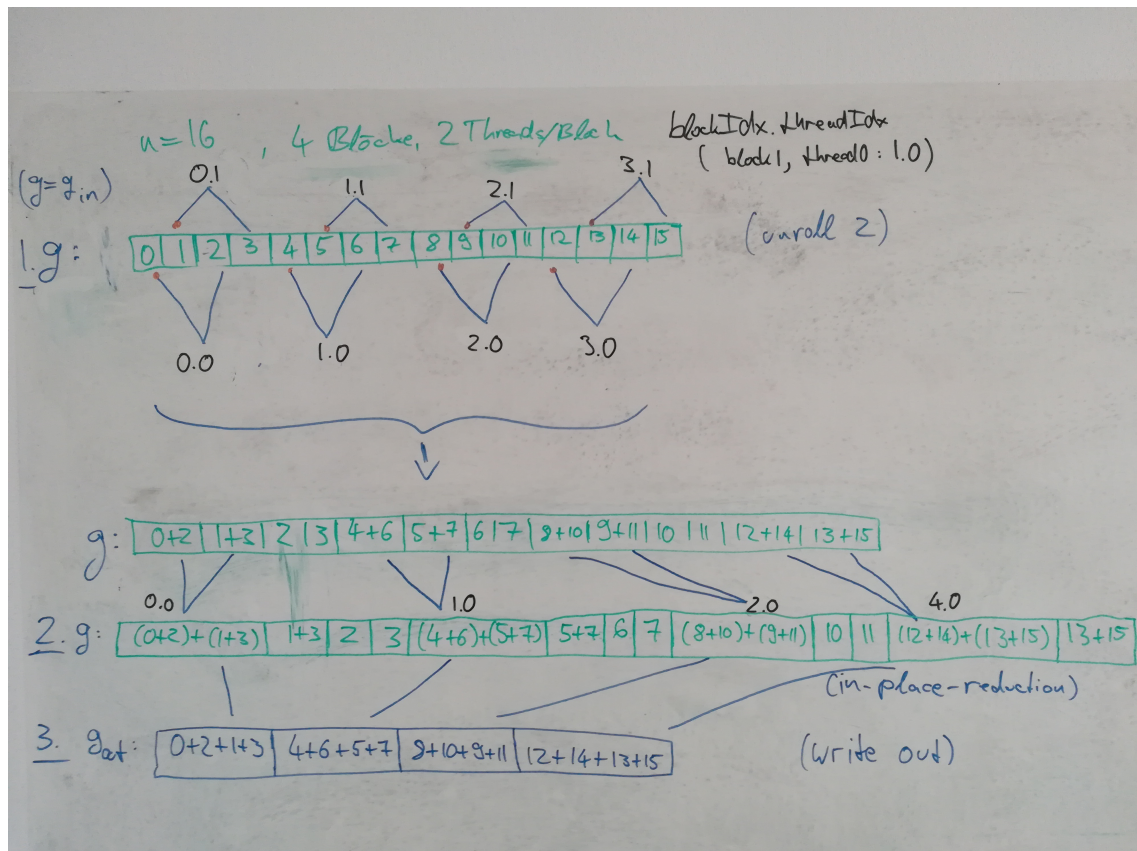
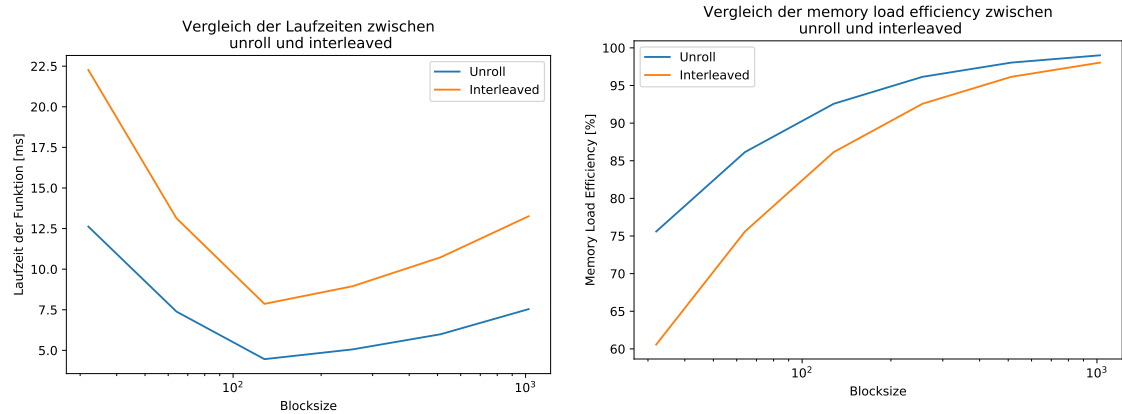


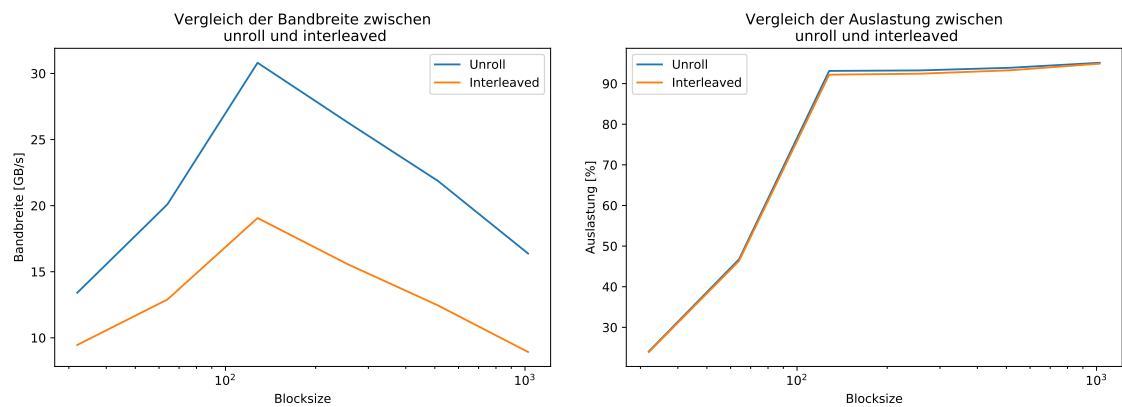
Abbildung 1: Skizzierung der Reduktion von einem Array mit 16 Elementen

1.2 (Aufgabenteil ii, iii, iv)

Messung der Laufzeiten, Global Memory Load Efficiency, Global Load Throughput und Achieved Occupancy



(a) Laufzeiten in Abhängigkeit der gewählten Blocksize. (b) Memory Load Efficiency in Abhängigkeit der gewählten Blocksize.



(c) Bandbreitennutzung in Abhängigkeit der gewählten Blocksize. (d) Achieved Occupancy in Abhängigkeit der gewählten Blocksize.

Abbildung 2: Unterschiedliche Parameter in Abhängigkeit der Blocksize. Man sieht, dass die Laufzeit und die Bandbreitennutzung bei einer Blocksize von 128 (d.h. 4 warps pro Block) am optimalsten ist. Ab diesem Wert ist auch die Auslastung im Bereich von ueber 90%.

Erklaerung blabla.

1.3 (Aufgabenteil v, vi)

Optimaler Wert fuer die Anzahl der Elemente q , die vor Reduktion summiert werden

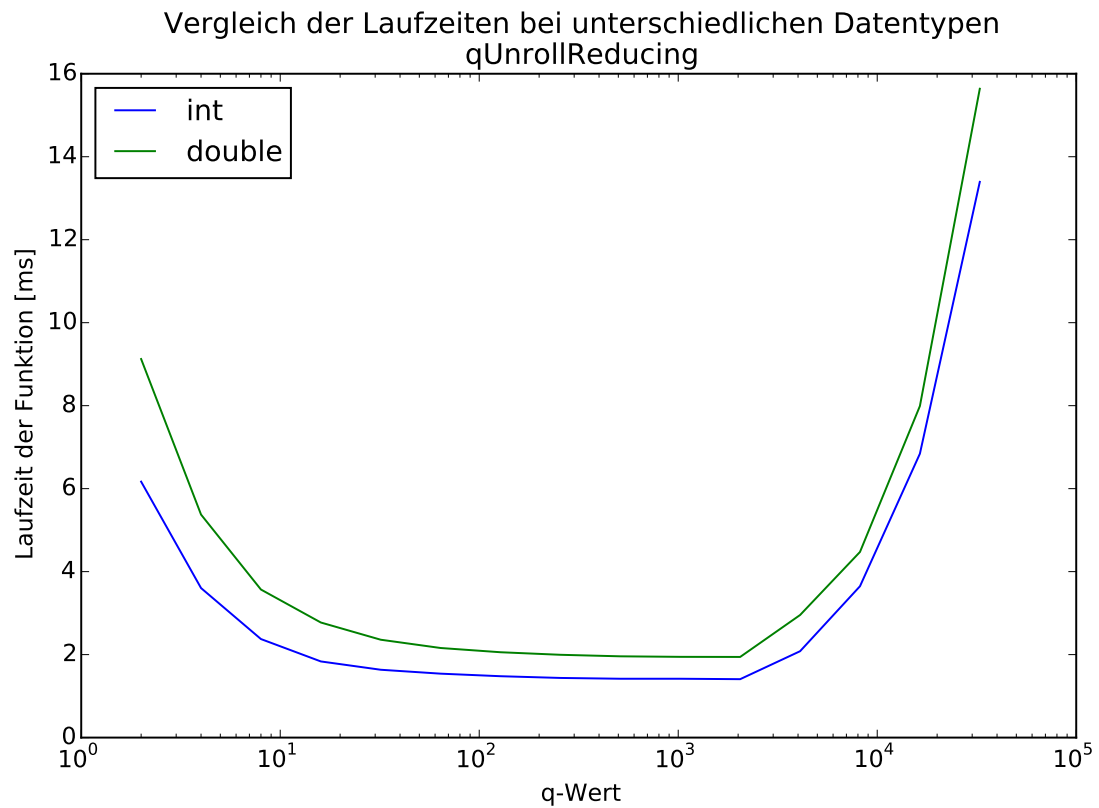


Abbildung 3: Laufzeiten der Methoden `qReduceUnrolling` und `qReduceUnrollingDouble` fuer verschiedene Werte von q . Vor der Reduktion werden q Werte pro Thread addiert. Das resultiert mitunter in einer kuerzeren Ausfuehrungszeit. Das Minimum bei den Datentypen `single` sowie `double` liegt bei $q = 2048$. Die Laufzeiten fuer den Datentyp `double` sind etwa 30% groesser als fuer den Datentyp `int`. Das koennte daran liegen, dass auf der Grafikkarte nicht so viele Register fuer den Datentyp `double` vorhanden sind.