Einführung in die Parallelverarbeitung
Prof. Dr. Claudia Fohry
M.Sc. Jonas Posner

Universität Kassel
Sommersemester 2019
Abgabe am 08.07.2019 um 10 Uhr

Aufgabenblatt CUDA

`buddyKGVSum.cu`:

Die Definitionen *kgV*, *minKgV* und *Buddy* werden von dem Aufgabenblatt APGAS übernommen. Es ist zusätzlich eine Buddyfunktion $f(x,y)$ für zwei Zahlen $x,y \in \mathbf{N}_1$ definiert als:

$$f(x,y) = \begin{cases} 1 & \text{falls } x \text{ und } y \text{ Buddies sind (wenn } kgv(x,y) \geq minKgv) \\ 0 & \text{sonst.} \end{cases}$$

Gegeben sind zwei $n \times n$-Arrays `a` und `b`. Beide enthalten Zufallszahlen aus dem Intervall $[1,m)$. Wie in den Vorlesungsbeispielen `matmul.cu` und `tiled_matmul.cu` stellen wir die $n \times n$ Matrizen als eindimensionale Arrays der Größe n^2 dar.

Implementieren Sie die Methode

```
buddyKGVSum(const int * a,
    const int * b,
    const int n,
    const int m,
    const int minKgv,
    const int verbose,
    int * c)
```

Hierbei bezeichnen

- `a` und `b` die Eingabematrizen,
- `n` die Größe der Matrizen,
- `m` die Obergrenze der zufällig generierten Zahlen,
- `minKgv` Schwellwert für zwei Buddies,
- `verbose` das verbose-level (siehe Abschlussbemerkungen) und
- `c` die Ergebnismatrix.

Das Programm wird wie folgt aufgerufen:

```
./buddyKGVSum n m minKgv seed verbosity
```

Jeder Eintrag in der Ergebnismatrix c berechnet sich nach folgender Formel:

$$c_{i,j} = \sum_{g=1}^{n} \sum_{h=1}^{n} f(a_{i,g}, b_{j,h})$$

Beispiel:

```
./buddyKGVSum 2 10 10 123 3
 5 6
 9 4

 4 8
 2 1

 4 1
 2 1
 Execution time: 0.01
```

Innerhalb der Methode `buddyKGVSum` müssen alle notwendigen Schritte (Anlegen des benötigten Grafikkartenspeichers, Kopieren der Daten, Aufruf des CUDA-Kernels, ...) implementiert werden.

Es darf **nicht** vorausgesetzt werden, dass n eine Zweierpotenz ist. Darüber hinaus gehende Optimierungen des Basisalgorithmus sind erlaubt, aber nicht gefordert. Alle in der Vorlesung behandelten Effizienzaspekte sollten beachtet werden. Außerdem sollen auch mögliche CUDA Errors behandelt werden. Die Verwendung mehrerer Kernel-Aufrufe ist erlaubt, jedoch nicht gefordert. Sie dürfen Teile der Berechnung auf dem Host ausführen, jedoch soll der Anteil der hostseitigen Berechnungen gering gehalten werden (ähnlich zum Vorlesungsbeispiel `scalProd.cu`). Sie dürfen Ihr Programm auf die technischen Daten der in den NV-Rechnern verbauten Grafikkarten (nVidia GTX 480) hin optimieren. Ihr Programm muss in der Lage sein, korrekte Ergebnisse für $0 < n \leq 2000$ zu berechnen.

Sie finden in Moodle die Vorgabedateien `buddyKGVSum.cu`, `buddyKGVSum.cuh` und `macros.h`, in welchen unter anderem die Arrays initialisiert werden. Nutzen Sie diese Vorgabedateien, um ihr Programm zu erstellen.

Abschlussbemerkungen:

Die Implementierung der Methode `main` in `buddyKGVSum.cu` verwendet einen optionalen, fünften Parameter. Dieser Parameter gibt das verbose-level an. Wenn kein Wert oder `0` übergeben wird, wird nur die Laufzeit ausgegeben. Mit einem Wert von `1` wird die Ergebnismatrix `c` ausgegeben. Mit einem Wert von `2` werden die generierten Eingabematrizen `a` und `b` ausgegeben. Mit einem Wert von `3` werden sowohl die Eingabematrizen als auch die Ergebnismatrix ausgegeben. Zum debuggen dürfen Sie sich weitere verbose-level definieren.

Eventuelle Fragen zu den Aufgaben stellen Sie bitte im Diskussionsforum auf der Moodle-Seite der Veranstaltung.

Wettbewerb: Falls es einen klaren Gewinner gibt, so verbessert die Gruppe mit dem schnellsten Programm ihre Abschlussnote um 0.3. Allerdings kann eine Gruppe nur maximal einen Wettbewerbsbonus in die Gesamtveranstaltung (inkl. OpenMP, APGAS) einbringen. Am Wettbewerb nehmen nur Gruppen teil, deren abgegebenes Programm exakt der Spezifikation entspricht und für alle getesteten Eingaben korrekt ist. Die Messungen werden auf den NV-Rechnern durchgeführt.

Abgabe: Die Lösungen von der APGAS und CUDA Aufgabe müssen zusammen bis spätestens 08.07.2019 um 10 Uhr per Email an `jonas.posner@uni-kassel.de` mit dem Betreff "IntroPV-Abgabe APGAS+CUDA Gruppe X" gesendet werden. Später abgegebene Lösungen werden als nicht abgegeben gewertet. Bitte geben Sie zu jedem Programm die vollständigen Kommandozeilen zur Kompilierung und Ausführung des Programms mit ab. Das abgegebene Progam **muss** sich auf den NV-Rechnern kompilieren lassen.