

## 5. Übungsblatt

Ausgabe: 15. Januar 2019 · Besprechung: 21. Januar 2019

### GPU-Programmierung

Bei diesem Aufgabenblatt geht es darum, einfache Datenbankoperationen auf einer GPU zu implementieren. Gehen Sie davon aus, dass Daten des TPC-H-Benchmarks spaltenorientiert wie folgt repräsentiert sind:

```
uint32_t  orderkey [N];  
uint32_t  partkey  [N];  
...  
char      comment [45*N];  
uint32_t  shipdate [N];
```

Auf diesen Daten soll nun eine Anfrage der folgenden Art ausgeführt werden:

```
SELECT  SUM (quantity * extendedprice)  
FROM    lineitem  
WHERE   suppkey < Z
```

wobei Z ein Parameter ist, um die Selektivität der Anfrage zu variieren.

### Aufgabe

Implementieren Sie die obige SQL-Anfrage (hart-codiert) mit Hilfe von CUDA oder OpenCL.

- Wie wirkt sich der Parameter Z auf die Laufzeit Ihres Programms aus? Vergleichen Sie das Verhalten mit dem Verhalten einer naiven C-Implementation (auf der CPU).
- Wie verhält sich die Performanz Ihres Programms im Verhältnis zu der einer CPU-basierten Lösung?

### Hinweise

- Ich selbst habe für diesen Zweck Daten einfach wie folgt generiert:

```
for (unsigned int i = 0; i < nitems; i++)
{
    (*quantity)[i]      = (random () % 10);
    (*extendedprice)[i] = (random () % 1000);
    (*suppkey)[i]       = (random () % 1000);

    if ((*suppkey)[i] < Z)
        sum += (*quantity)[i] * (*extendedprice)[i];
}
```

- Auf der Kurswebseite habe ich zur Inspiration das `coalescing`-Beispiel aus der Vorlesung zum Download zur Verfügung gestellt. Der Code wurde entwickelt unter macOS und muss u. U. für andere Systeme noch angepasst werden.