7. Übungsblatt

Ausgabe: 6. Januar 2020 · Besprechung: 13. Januar / 20. Januar

1 Vektorisierung, Aggregation (Besprechung am 13.01.)

Als Beispiel für den Einsatz von SIMD-Instruktionen wollen wir in dieser Aufgabe die Anfrage

```
SELECT  COUNT(*)
  FROM  R
 WHERE  R.a > 42
```

durch vektorisierten Code implementieren.

Gehen Sie dazu von einem Array aus, das mit Zufallszahlen zwischen 0 und 99 gefüllt ist.

Als Startpunkt finden Sie auf der Kurswebseite ein Gerüst für ein C-Programm (Datei `count.c`). Ihre Aufgabe ist es, die aktuelle skalare Implementierung durch eine SIMD-Implementation. Wie verändert sich dadurch die Performance des Algorithmus?

Hinweis: Zur Implementation verwenden Sie am einfachsten die *intrinsics* Ihres C-Compilers. Eine Liste der verfügbaren *intrinsics* finden Sie z. B. im *Intel 64 and IA-32 Architectures Software Developer's Manual*.

2 Vektorisierung, Dekompression (Besprechung am 20.01.)

In der Vorlesung wurde gezeigt, wie SIMD-Instruktionen genutzt werden, um 32-zu-n-Bit-Kompression zu erzeugen, d. h., statt voller 32 Bits werden nur n Bits zur Speicherung aller Zahlen eines Arrays verwendet.

Schreiben Sie entsprechende Routinen zur Kompression und Dekompression für $n = 9$. Verwenden Sie dazu einmal eine skalare Implementation und einmal eine SIMD-optimierte Version; vergleichen Sie die Laufzeiten beider Varianten. Auch hier finden Sie eine Vorlage auf der Kurswebseite (Datei `compression.c`).

Hinweis: Bei der reinen Dekompression wird Ihr System möglicherweise begrenzt durch die Bandbreite Ihres Hauptspeichers. Um den reinen Durchsatz Ihrer Dekompressionsroutine zu messen, können Sie die Dekompression z. B. direkt mit einer Aggregation (etwa der Bildung einer Summe) kombinieren und somit das tatsächliche Schreiben des Ergebnisses vermeiden.