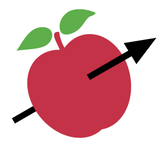
Supersonic Algorithms

Anton Rodenwald (18), Schillerschule Hannover



Ideenfindung und Forschungsfrage

- Sortieralgorithmen für Zahlen im Informatikunterricht kennengelernt
- Ich fragte mich, wie man Zahlen am schnellsten sortieren kann
- Sortieralgorithmen sind bereits bekannt, deswegen Fokus auf Implementation
- Forschungsfrage: Wie lassen sich Programme durch geschickte Implementation in ihrer Ausführung beschleunigen?
- Kein Vergleich von Sortieralgorithmen, fast immer Quicksort verwendet
- Fokus auf die Programmiersprachen Python, C++ und C
- In Interquellen wurde C++ um ein vielfaches schneller als Python beschrieben
- Es war auch meine Vermutung, dass C++ deutlich schneller Zahlen sortiert

Material, Vorgehen und Methode

- Ich nutze meinen PC und Laptop zur Implementation
- Ich verwende die Programme Visual Studio Code und Compiler oder Interpreter für meine Programmiersprachen
- Ich implementierte anschließend die Quicksort auf verschiedene Arten und erhielt so viele Variationen zum testen
- Die Tests führte ich auf meinem Desktop PC durch (CPU ist der Ryzen 7 2700, 16 GB Arbeitsspeicher), der immer gleich ausgelastet war, um Vergleichbarkeit zu gewährleisten
- Die Zeit stoppte ich mit den Zeitfunktionen der Sprachen

Beispiel der Zeitnahme in Python

```python
import time

class Timer:
    timers = {}
    @staticmethod
    def start(name):
        Timer.timers[name] = time.time_ns()

    @staticmethod
    def stop(name):
        diff = time.time_ns() - Timer.timers[name]
        diffseconds = diff / (10 ** 9)
        print(name, ":", round(diffseconds, 5), "Seconds")
```

Ergebnissdiskussion

Schwierigkeiten

- Meine größte Schwierigkeit war, dass ich noch nicht viel Erfahrung mit der Optimierung und dem wissenschaftlichen Arbeiten hatte
- Erstellung der Dokumentation mit LaTeXteils frustrierend
- Mit einigen Bibliotheken und Sprachen schwierigkeiten
- Mir war das testen mit 10 Millionen Zahlen in Go nicht möglich, da dort Rekursion nicht so gut funktioniert

Beantwortung der Forschungsfrage und Ausblick

- Die besten Möglichkeiten zur Optimierung in Python sind NumPy und Numba
- Cython schlechter, CTypes komplex
- In C++ bietet AVX2 interessante Möglichkeiten der Optimierung
- Python kann ähnlich schnell wie C++ sein kann
- Insgesamt profitiert Python vorallem von schnellen C Bibliotheken