

LEHRSTUHL FÜR RECHNERARCHITEKTUR UND PARALLELE SYSTEME

**Praktikum Rechnerarchitektur**

Parallele Berechnung der Mandelbrotmenge

Wintersemester 2018/19

Maximilian Frühauf

Tobias Klasuen

Florian Lercher

Niels Mündler

---

# 1 Einleitung

## 1.1 Didaktische Ziele

Die Leistung und Geschwindigkeit individueller Rechenkerns stagniert seit einigen Jahren. Moderne Computer erlangen einen Großteil ihrer erhöhten Rechenleistung seit einiger Zeit nur noch durch Parallelisierung. Da dies stets einen geringen Overhead verursacht werden gut durchdachte Konzepte zur Ausnutzung mehrerer Kerne immer wichtiger.

Dieses Projekt soll Nutzern ermöglichen intuitiv ein Gefühl dafür zu gewinnen, welche wichtige Rolle hierbei die Aufteilung der Last zwischen den Rechenknoten spielt.

## 1.2 Verwendung der Mandelbrotmenge

Die Mandelbrotmenge ist eine Teilmenge der komplexen Zahlen. Um sie zu berechnen wendet man folgende Formel wiederholt auf jede komplexe Zahl  $c$  an:

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0 \quad (1)$$

In der Mandelbrotmenge befinden sich alle  $c$ , für die der Betrag von  $z_n$  für beliebig große  $n$  endlich bleibt. Wenn der Betrag von  $z$  nach einer Iteration größer als 2 ist, so strebt  $z$  gegen unendlich, das zugehörige  $c$  liegt also nicht in der Menge. Sobald  $|z_n| > 2$  kann die Berechnung daher abgebrochen werden.

Um nun für eine beliebige Zahl zu bestimmen, ob diese in der Mandelbrotmenge liegt, müssen theoretisch unendlich viele Rechenschritte durchgeführt werden. Zur computergestützten Bestimmung werden die Rechenschritte nach einer bestimmten Iteration abgebrochen die Zahl als in der Menge liegend betrachtet.

Es handelt sich also um eine Berechnung, die sehr zeitaufwändig ist, wobei die benötigte Zeit durch Erhöhen der Iterationszahl beliebig erhöht werden kann. Zusätzlich ist die Berechnung für jede einzelne komplexe Zahl unabhängig von jeder anderen Zahl.

Diese Eigenschaften ermöglichen es, zweierlei Dinge zu kontrollieren:

- Die Dauer der Berechnung
- Die Aufteilung der Berechnung auf unterschiedliche Rechenkerne

Somit kann gesichert werden, dass eine wahrnehmbare Zeit (100-200 ms) zur Berechnung benötigt wird. Zudem können die Berechnungen frei auf verfügbare Rechenknoten verteilt werden, sodass für verschiedenste Aufteilungen eine Rechenzeit visualisiert werden kann.

## 1.3 Darstellung der Mandelbrotmenge

Komplexe Zahlen lassen sich auch grafisch darstellen, indem man sie in ein Koordinatensystem einträgt. Dabei entspricht die x-Koordinate dem Realteil und die y-Koordinate dem Imaginärteil der Zahl. Für das Projekt wird ein Ausschnitt des Bildschirms als



Abbildung 1: Die Mandelbrotmenge, visualisiert in einem Ausschnitt des komplexen Zahlenraumes.

zweidimensionale Darstellung des komplexen Raumes betrachtet und für jeden darin liegenden Punkt die Zugehörigkeit zur Mandelbrotmenge bestimmt. Dabei wird der Raum jedoch diskretisiert, indem jedem Pixel des Bildschirms die komplexen Koordinaten  $c$  der linken oberen Ecke zugeordnet werden.

Die grafische Darstellung der Mandelbrotmenge wird durch Einfärbung des zu  $c$  gehörigen Pixels erhalten. Die Zahl der benötigten Iterationen bis zum Abbruch der Berechnung bestimmt dabei die Farbe, sodass alle Pixel innerhalb der Menge und alle Pixel außerhalb jeweils gleichfarbig sind.

Das entstehende Fraktal ist aufgrund seiner Form auch als “Apfelmännchen” bekannt (siehe 1.3). Die Menge ist zusammenhängend, jedoch bilden sich an ihren Rändern viele kleine und sehr komplexe Formen, die visuell ansprechend sind. Es eignet sich daher gut, um optisch Interesse am Projekt zu wecken.

## 1.4 MPI

Die Kommunikation der verschiedenen Prozesse des Backends wird mittels MPI realisiert. Das Message Passing Interface<sup>1</sup> ist eine weit verbreitete Spezifikation, die Kommunikation zwischen unabhängigen Rechenkernen regelt. Dadurch existieren viele gut funktionierende Umsetzungen in einer Vielzahl von Programmiersprachen. Durch MPI wird echte Parallelisierung mit geringem Overhead ermöglicht. Es können so die einzelnen Workerprozesse auf jeweils einem eigenen unabhängigen Rechenkernen laufen. Die Gestaltung von MPI erlaubt dabei alle möglichen Aufteilungen, von Kernen auf einem Prozessor bis hin zu unabhängigen Clustern, die lediglich eine SSH-Verbindung besitzen.

---

<sup>1</sup><https://www.mpi-forum.org/>

---

## 2 Problemstellung und Spezifikation

- Fachliche Spezifikation in Anhang
- NFRs in Einleitung schreiben

### 3 Dokumentation der Implementierung

- Sollte größter Teil werden
- 1. High level overview?
  2. Wie läuft das auf meinem System?
  3. Code Dokumentation / Entwicklerdokumentation

## 4 Ergebnisse / Evaluation

- Skalierbarkeitsgraph
- Wie gut ist SIMD / OpenMP / MPI / Mischformen?
- Frontend Overhead messen

## 5 Zusammenfassung

- Zusammenfassung
- Ausblick