

Guten Tag sehr geehrte Prüfungskommission,

mein Name lautet Thomas Mittermair und ich habe für heute eine Präsentation zum *Parallelen Programmierparadigma* vorbereitet, welche ich Ihnen nun vorstellen möchte.

## Begriffserklärung

Zunächst finde ich es für angebracht, den Titel der Präsentation, nämlich „*Paralleles Programmierparadigma*“, näher zu erläutern. Das Adjektiv „*parallel*“ ist dabei von essentieller Bedeutung und wird erst bei Gegenüberstellung mit dessen Gegenteil, dem Adjektiv „*sequentiell*“, klar. Sequentielle Vorgänge laufen aufeinander folgend und hintereinander ab, während parallele Aktivitäten gleichzeitig und nebeneinander verlaufen. Grafisch betrachtet erkennt man links einen sequentiellen Vorgang, während rechts drei parallele Abläufe dargestellt sind.

Gehen wir nun zum Terminus „*Paradigma*“ über. Dies ist ganz allgemein eine grundlegende wissenschaftliche Denkweise. Ein Programmierparadigma ist folglich eine Art bzw. ein Ansatz der Programmierung, da es wie auch im Alltag oft verschiedene Herangehensweisen an ein Problem gibt, die allesamt zum Ziel führen können. Diese Art der Programmierung befasst sich mit der Parallelverarbeitung, also dem gleichzeitigen Bearbeiten von mehreren Aufgaben.

## Motivation

Genug der Begriffe, kommen wir nun zur Motivation.

Das 21. Jahrhundert steht im Zeichen der Beschleunigung. Immer mehr Menschen führen ein immer schnelleres Leben, dies trifft ohne Abstriche auch auf das digitale Dasein zu. Die Benutzer von informatischen Systemen verlangen nach immer schnelleren Reaktionszeiten und sind keineswegs gewillt zu warten.

Das Problem hierbei liegt darin, dass Rechner bestimmten physikalischen Grenzen unterworfen sind, die eine Weiterentwicklung ab einem gewissen Punkt unmöglich machen. In diesem Zusammenhang ist beispielsweise die Lichtgeschwindigkeit als natürliche Grenze der Signalübertragungsgeschwindigkeit zu nennen. Und genau an dieser Stelle kommt das Parallele Programmierparadigma ins Spiel. Dies ist ein Ansatz, der zu einer Beschleunigung führen kann, ohne dabei vom Fortschritt der Computer abhängig zu sein.

Dieser Ansatz der Programmierung nimmt sich ein Beispiel an der Natur, wie auch durch die Abbildung angedeutet werden soll. Dort findet stets eine simultane Weiterentwicklung statt, denn jeder Mensch, jede Pflanze, also grundsätzlich jedes Lebewesen, entwickelt sich gleichzeitig, und nicht nacheinander, weiter.

Da mich dieses Thema persönlich schon seit der 3. Klasse beschäftigt und fasziniert, habe ich mich im Rahmen des 5. Jahres an dieser Schule mit dieser Materie näher befasst, und dann darüber auch eine Facharbeit geschrieben. Die Idee dahinter und die damit einhergehende erhebliche Leistungserhöhung möchte ich Ihnen nun näherbringen.

## Grundlegende Idee

Die grundlegende Idee besteht darin, große Probleme, welche als Ganzes betrachtet von Programmen kaum bewältigt werden können, in kleinere Einheiten aufzuteilen und mehrere Komponenten gleichzeitig daran arbeiten zu lassen. Diese parallel arbeitenden Komponenten werden im Folgenden als „*Abarbeitungseinheiten*“ bezeichnet. Als Resultat erhofft man sich, so schnell wie möglich eine Lösung für das gegebene Problem zu erzielen.

Den Sachverhalt, dass mehrere Komponenten, die zusammen an einem Problem arbeiten, selbiges oft effizienter lösen können als Einzelgänger, soll auch die obige Grafik unterstreichen. Gemeinsam können die kleinen Fische den Hai vertreiben, während sie alleine wohl gejagt und gefressen werden würden.

## Analogie: Mähen

Um Ihnen das Konzept hinter der parallelen Programmierung näherzubringen, habe ich mir eine Analogie ausgedacht, anhand welcher das Konzept erläutert werden kann.

Stellen wir uns vor, ein Bauer besitzt ein Grundstück, auf welchem Unkraut gewachsen ist. Er will nun dieses Feld so schnell wie möglich mit Hilfe von Rasenmähern von dem Gras befreien.

### Analogie: Mähen – Sequentielle Lösung

Der erste Ansatz des Bauers ist die sogenannte sequentielle Lösung, er verwendet also keine Parallelität, sondern kauft nur einen Rasenmäher und fährt mit diesem das gesamte Feld alleine ab. Betrachten wir das Ganze zunächst grafisch und messen die Zeit, die er in diesem Schema hierfür benötigt.

Man erkennt, dass der Vorgang insgesamt 12 Sekunden gedauert hat.

### Analogie: Mähen – Parallele Lösung

Dem Bauer ist dies zu langsam, er sucht deshalb nach schnelleren Lösungen und stößt auf die Parallelverarbeitung. Aus diesem Grund kauft er sich insgesamt 3 Rasenmäher, fragt 2 Freunde und gemeinsam mähen alle 3 Personen gleichzeitig jeweils einen Teil des Grundstückes. Schauen wir uns nun auch dies schematisch an und stoppen wiederum die Zeit, welche die drei Freunde in dieser Simulation hierfür benötigen.

Es wird ersichtlich, dass das Mähen in diesem Fall nur noch 4 Sekunden benötigt hat. Dies entspricht einer 3-fachen Beschleunigung, was erheblich ist.

### Analogie: Mähen – Fazit

Aus diesem bildhaften Beispiel lässt sich bereits ein Fazit über das gesamte Parallele Programmierparadigma ableiten. Der Bauer kann durch den Einsatz von Parallelität eine erhebliche Zeiteinsparung und Beschleunigung verbuchen, jedoch nicht ohne Mehrkosten, denn er benötigt zusätzliche Menschen und Rasenmäher.

Parallelisierung kann also zu deutlichen Zeitgewinnen führen, wobei allerdings Mehrkosten für zusätzliche Ressourcen einhergehen.

## Eigene Forschung

Im Rahmen meiner Facharbeit konnte ich auch selbst ein wenig „Forschung“ betreiben und das parallele Programmierparadigma auch praktisch anwenden.

Das Problem, welches es zu lösen galt, war die Ermittlung der Primzahlen zwischen 1 und 100.000, was eine aufwendige Berechnung darstellt. Durch Parallelisierung ist eine Optimierung dieses Vorgangs möglich. Und kurz zur Erinnerung: Alle ganzen Zahlen, die größer als 1 und nur durch 1 und sich selbst teilbar sind, werden als Primzahlen bezeichnet.

Ich habe dann diese aufwendigen Berechnungen mit einer Abarbeitungseinheit durchgeführt, also in der Analogie einen Rasenmäher verwendet. Danach habe ich die Anzahl der Abarbeitungseinheiten, also die Anzahl der Rasenmäher, immer weiter erhöht und jeweils immer gemessen, wie lange die Berechnungen gedauert haben.

Aus diesen Messwerten konnte ich dann jeweils die Beschleunigung berechnen und sie auch grafisch in einem Diagramm darstellen, auf welches ich nun zu sprechen komme.

## Forschungs-Ergebnisse

Meine Forschungen haben zu dieser Grafik geführt.

Man erkennt, dass mit 2 Abarbeitungseinheiten eine 1,3-fache Beschleunigung erzielt wurde, mit 4 eine 2,2-fache mit 16 Abarbeitungseinheiten war eine 3,7-fache Beschleunigung zu verzeichnen. Das Diagramm zeigt folglich eine zunächst rätselhafte Eigenschaft auf: Die Verwendung von  $x$  Abarbeitungseinheiten führt **nicht** zu einer  $x$ -fachen Beschleunigung! Mehr als eine 3,8-fache Beschleunigung konnte nicht erreicht werden, auch nicht mit mehr als 1.000 Abarbeitungseinheiten! Ab einem gewissen Punkt kommt es zu einer Stagnation, wie auch aus der Abbildung klar wird.

Der Grund dafür liegt in der Natur der Parallelverarbeitung. Der Einsatz von mehr Abarbeitungseinheiten bringt einen organisatorischen Mehraufwand mit sich. Dieser Aufwand kann irgendwann sogar den Nutzen, der mit einer weiteren Abarbeitungseinheit einhergeht, übersteigen. In diesem Fall kommt es also zu einem Rückgang der Beschleunigung! Ansätze davon sind auch im Diagramm erkennbar.

## Zusammenfassung

Abschließend möchte ich noch ein kurzes Resümee ziehen.

Die Analogien und Forschungs-Ergebnisse haben gezeigt, dass sich durch das parallele Rechnen starke Performancesteigerungen erzielen lassen. Allerdings sollte man nicht nur auf den Zeitgewinn achten, sondern diese Verbesserung auch in ein Verhältnis mit dem damit einhergehenden Mehraufwand setzen. Im vorhergehenden Diagramm wurde sogar gezeigt, dass beim Einsatz von parallelen Abarbeitungseinheiten sogar das Sprichwort „*Weniger ist mehr*“ gelten kann, was man auf den ersten Blick nicht annehmen würde.

In diesem Sinne möchte ich meine Präsentation abschließen. Ich hoffe, ich konnte Ihnen dieses Konzept ein wenig näherbringen und bei einigen vielleicht das Interesse wecken. Ist dies der Fall, so verweise ich in diesem Zusammenhang auf meine Facharbeit.

Vielen Dank für Ihre Aufmerksamkeit, nun können gerne Fragen gestellt werden.