Praktikum Parallele Numerik

Fabian Miltenberger, Sébastien Thill, Thore Mehr

Betreuer: Markus Hoffmann, Thomas Becker

Zusammenfassung Im Rahmen dieses Praktikums haben wir viel gelernt.

1 Projekt 1

In diesem Projekt lag der Schwerpunkt auf dem Kennenlernen der Bibliothek *OpenMP* sowie deren Handhabung. Weiter ging es um den *IntelThreadChecker*, ein Programm zum Analysieren von Programmcode auf potentielle Fehler in der Parallelisierung. Zu guter letzt haben wir uns mit der FEM-Methode beschäftigt, dabei im Speziellen mit dem Gauß-Seidel-Verfahren zum Lösen linearer Gleichungen wie sie bei der Differenzenmethode vorkommen. Zu guter letzt betrachteten wir einige andere Verfahren zum Lösen solcher Probleme und haben das CG-Verfahren implementiert.

1.1 Aufgabe 1

```
Hello World, this is Thread0
Hello World, this is Thread5
Hello World, this is Thread4
Hello World, this is Thread7
Hello World, this is Thread1
Hello World, this is Thread3
Hello World, this is Thread6
Hello World, this is Thread2
```

Listing 1.1. Beispielhafte Ausgabe des Programms bei Ausführung mit 8 Fäden.

Wie in der Auflistung 1.1 zu sehen, folgt die Reihenfolge der ausgeführten Fäden keinem bestimmten Muster. Die Reihenfolgen zwischen verschiedenen Ausführungen sind in der Regel verschieden.

1.2 Aufgabe 2

1.3 Aufgabe 3

a) Im Folgenden einige bekannte Größen der Parallelisierung, wie sie in der Vorlesung Rechnerstrukturen gelehrt wurden.

Vorab sei $T(n)$ die Ausführungszeit auf n Prozessoren, $P(n)$ die Anzahl der auszuführenden Einheitsoperationen und $I(n)$ der Parallelindex.

Der Speedup/die Beschleunigung $S(n)$:

$$S(n) = \frac{T(1)}{T(n)} \tag{1}$$

Die Effizienz $E(n)$:

$$E(n) = \frac{S(n)}{n} = \frac{T(1)}{n \cdot T(n)} \tag{2}$$

Der Mehraufwand $R(n)$:

$$R(n) = \frac{P(n)}{P(1)} \tag{3}$$

Die Auslastung $U(n)$:

$$U(n) = \frac{U(n)}{n} = R(n) \cdot E(n) = \frac{P(n)}{n \cdot T(n)} \tag{4}$$

b) **Race Condition**

Wettlaufsituationen Dabei hängt Ergebnis von konkreter Ausführungsreihenfolge ab (daher Wettlauf) Entsteht, wenn verschiedene Fäden auf gleiche Variable zugreifen, und mindestens ein Faden deren Wert manipuliert Korrektheit der Ergebnisse hängt von Ausführungsreihenfolge ab

Dead lock

Zyklus im Allokationsgraphen

c) Die Vor- und Nachteile verschiedener Architekturen bezüglich verschiedener Aspekte:

	Anwenderfreundlichkeit	Energieeffizienz
GPU	Gut	Mittel
CPU	Gut	Gering
FPGA	Gering	Sehr gut
MIC	Gut	Gut

Tabelle 1. Verschiedene Beschleuniger im Vergleich

1.4 Aufgabe 4

a) Das Problem, das gelöst werden soll... Wie wir es gelöst haben... Zerfall der dünnen Matrix in vier bedingten Beiträgen einer Summe

b) Eine naive Parallelisierung des implementierten Gauß-Seidel-Verfahrens ist nicht möglich, da innerhalb der Berechnung von u^i – eine naive Parallelisierung würde versuchen diese Berechnung zu parallelisieren – Abhängigkeiten zwischen den einzelnen Einträgen bestehen. Für die Berechnung von $u_{x,y}^{i+1}$ werden diese Abhängigkeiten in Abbildung 1 dargestellt.

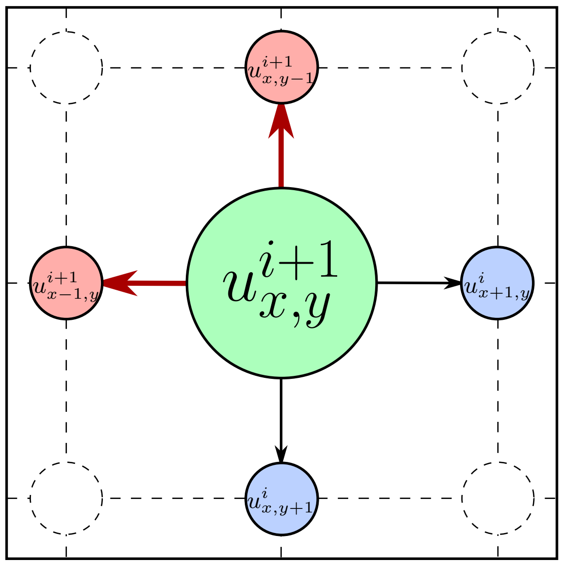

Abbildung 1. Abhängigkeiten zu Nachbarknoten, um Knoten $u_{x,y}^{i+1}$ zu berechnen. Es fällt auf, dass zwei Werte der gleichen Iterierten $i+1$ benötigt werden (die roten Knoten). Durch diese Abhängigkeit können nicht alle Einträge einer Iterierten gleichzeitig, das heißt parallel, berechnet werden.

Eine Mögliche Lösung wäre, zur Berechnung von u^i über die einzelnen Diagonalen zu parallelisieren. Dies würde funktionieren, da diagonal keine Abhängigkeit zwischen den Einträgen besteht. Dennoch ist diese Lösung nicht praktikabel, da der Synchronisationsaufwand aufgrund der Anzahl der Diagonalen zu groß – und die Anzahl der parallelisierten Einträge nicht selten zu klein ist (abhängig von der Eingabegröße, aber mit größeren Eingaben wächst auch die Anzahl der Diagonalen und der damit verbundene Mehraufwand).

c) Ausführliche Beschreibung der Vorgehensweise...

Die Laufzeiten sind Tabelle 2 zu entnehmen.

$u^{i+1}_{0,0}$	$u^{i+1}_{1,0}$	$u^{i}_{2,0}$	$u^{i}_{3,0}$	$u^{i-1}_{4,0}$	$u^{i-1}_{5,0}$	$u^{i-2}_{6,0}$
$u^{i+1}_{0,1}$	$u^{i}_{1,1}$	$u^{i}_{2,1}$	$u^{i-1}_{3,1}$	$u^{i-1}_{4,1}$	$u^{i-2}_{5,1}$	$u^{i-2}_{6,1}$
$u^{i}_{0,2}$	$u^{i}_{1,2}$	$u^{i-1}_{2,2}$	$u^{i-1}_{3,2}$	$u^{i-2}_{4,2}$	$u^{i-2}_{5,2}$	$u^{i-3}_{6,2}$
$u^{i}_{0,3}$	$u^{i-1}_{1,3}$	$u^{i-1}_{3,3}$	$u^{i-2}_{3,3}$	$u^{i-2}_{5,3}$	$u^{i-3}_{5,3}$	$u^{i-3}_{7,3}$
$u^{i-1}_{0,4}$	$u^{i-1}_{1,4}$	$u^{i-2}_{2,4}$	$u^{i-2}_{3,4}$	$u^{i-3}_{4,4}$	$u^{i-3}_{5,4}$	$u^{i-4}_{6,4}$
$u^{i-1}_{0,5}$	$u^{i-2}_{1,5}$	$u^{i-2}_{2,5}$	$u^{i-3}_{3,5}$	$u^{i-3}_{4,5}$	$u^{i-4}_{5,5}$	$u^{i-4}_{6,5}$
$u^{i-2}_{0,6}$	$u^{i-2}_{1,6}$	$u^{i-3}_{2,6}$	$u^{i-3}_{3,6}$	$u^{i-4}_{4,6}$	$u^{i-4}_{5,6}$	$u^{i-5}_{6,6}$

Abbildung 2. Verdeutlichung der Vorgehensweise der Parallelisierung. Zuerst werden diejenigen Einträge von u parallel berechnet, die sich in grau Markierten Feldern befinden. Anschließend parallel die Einträge in den weißen Feldern. Es ist zu beachten, dass die Einträge $u_{x,y}$ für unterschiedliche Iterierte i berechnet werden.

Problemgröße			Seq.	2 Threads		4 Threads		…	32 Threads	
l	d	n	Laufzeit	Laufzeit	Speedup	Laufzeit	Speedup	…	Laufzeit	Speedup
4	15	225	0,005	0,007	0,357	0,008	0,625	…	1,361	0,0389
5	31	961	0,053	0,056	0,473	0,049	1,08	…	4,041	0,178
8	255	65.025	115	81	1,42	45,3	2,54	…	25,33	4,54
9	511	261.121	1340	943	1,42	504	2,67	…	177	7,57

Tabelle 2. Laufzeiten unserer Parallelisierung gegenüber der sequentiellen Version.

1.5 Aufgabe 5

a) Die Bedingungen lauten nach den Folien der FEM-Einführung:
Ω sei ein beschränktes Gebiet
Γ sei hinreichend glatt
$f : \Omega \to \mathbb{R}$ gegebene Funktion, wie es hier der Fall ist.

b) Gesucht ist f mit

$$u(x,y) = \sin(2M\pi x)\sin(2N\pi y) \tag{5}$$

Dies kann durch einfache Anwendung des *Laplace-Operators* Δ berechnet werden:

$$\begin{aligned} f(x,y) &= -\Delta u(x,y) \\ &= -\frac{\partial u}{\partial x^2} - \frac{\partial u}{\partial y^2} \\ &= -\frac{\partial}{\partial x}(2M\pi\cos(2M\pi x)\sin(2N\pi y)) - \frac{\partial}{\partial y}(2N\pi\sin(2M\pi x)\cos(2N\pi y)) \\ &= 4M^2\pi^2\sin(2M\pi x)\sin(2N\pi y) + 4N^2\pi^2\sin(2M\pi x)\sin(2N\pi y) \\ &= (M^2+N^2)4\pi^2\sin(2M\pi x)\sin(2N\pi y) \end{aligned} \tag{6}$$

c) Da in Gleichung 6 $M, N \in \mathbb{N}$ beliebig, wählen wir der Einfachheit halber $M = N = 1$ für die Lösung dieser Teilaufgabe. Damit ergibt sich

$$f(x,y) = 8\pi^2\sin(2\pi x)\sin(2\pi y) \tag{7}$$

Die analytische Lösung $u(x,y)$ ist bereits aus der Aufgabenstellung mit Gleichung 5 gegeben und wird zur Überprüfung der Ergebnisse verwendet.

1.6 Aufgabe 6

a)
b)
c)

2 Parojekt 2