
Aufgabenblatt 6

1. Aufgabe (1 + 6 + 3 Punkte)

In dieser Aufgabe soll eine Lösung für die zweidimensionale diskrete Laplace-Gleichung ¹ mit Hilfe des iterativen Jacobi-Verfahrens ² angenähert werden. Das Jacobi-Verfahren benötigt zur näherungsweisen Lösung des Gleichungssystems zwei (quadratische) Matrizen `u_new` und `u_old` der Dimension N . Die Lösung kann mit folgendem Algorithmus bestimmt werden:

Eingabe : Matrix `u_old` mit (beliebigen) Initialwerten. Alle Randwerte der Matrix sollen pseudozufällig gesetzt sein und sind konstant.

```
while nicht konvergiert do  
  for alle inneren Elemente do  
    | u_new[i][j] = (u_old[i+1][j] + u_old[i-1][j] + u_old[i][j+1] + u_old[i][j-1])/4  
  end  
  kopiere u_new in u_old  
  bestimme Abweichung  
end
```

Um zu überprüfen, ob die Lösung konvergiert, kann die Abweichung folgendermaßen bestimmt werden:

```
for alle inneren Elemente do  
  | diff += (u_new[i][j] - u_old[i][j])2  
end  
Abweichung =  $\sqrt{\text{diff}}$ 
```

¹<https://de.wikipedia.org/wiki/Laplace-Gleichung>

²<https://de.wikipedia.org/wiki/Jacobi-Verfahren>

- a) Implementieren Sie zunächst eine sequentielle Variante, die mit Matrizen variabler Größe getestet werden kann.
- b) Parallelisieren Sie ihre Lösung aus Teilaufgabe a) mit MPI. Partitionieren Sie dazu die Matrizen derart, dass die parallelen Prozesse an unterschiedlichen Teilen arbeiten. Verwenden Sie an den Partitionsrändern sogenannte Geisterzellen (ghost cells) und achten Sie darauf, dass keine Deadlocks auftreten können. Testen Sie ihre Lösung mit allen 16 Praktikumsrechnern und analysieren Sie die Skalierbarkeit im Verhältnis zur sequentiellen Variante. Versuchen Sie ihre Lösung zu optimieren, indem Sie nicht-blockierende Kommunikation einsetzen.
- c) Erweitern Sie ihre Lösung aus Teilaufgabe b) um C++11-Threads. Dazu sollen auf jedem Rechenknoten Threads für die parallele Ausführung eingesetzt werden. Die Kommunikation innerhalb eines Knotens soll über geteilten Speicher erfolgen, die Kommunikation zwischen den Knoten weiterhin mit MPI. Pro Rechenknoten sollen bis zu 8 Threads gestartet werden können. Beachten Sie, dass Multithreading mit `MPI_Init_thread` explizit aktiviert werden muss. Wie verändert sich die Laufzeit im Vergleich zu Teilaufgabe b)?

Hinweise

Die Abnahme für das Blatt soll bis Dienstag, 31. Mai 2016 erfolgen.

Linksammlung

- <http://www.mpi-forum.org/>
- <http://www.open-mpi.org/doc/v1.8/>