

Übungen zu „Parallele und verteilte Algorithmen“, Sommer 2002

Nr. 4, Abgabe: 14. Mai in der Vorlesung

Die Abgabe ist in Gruppen bis zu zwei Personen erlaubt. Programme sind schriftlich (etwa als Ausdruck) **und** per email an pinf3@mathematik.uni-marburg.de abzugeben.

Odd-Even Transposition Sort

Das „Odd-Even Transposition Sort“-Verfahren sortiert eine Folge von Zahlen (a_0, \dots, a_{n-1}) (n sei o.B.d.A. gerade) in $n/2$ Iterationen auf einem eindimensionalen Gitter von Prozessoren. Dabei wird dem i -ten Prozessor das Element a_i zugewiesen. Der Algorithmus besteht aus zwei sich wiederholenden Phasen:

1. Beim Ungerade/Gerade-Tausch vergleicht jeder k -te Prozessor mit k *ungerade* sein Element mit demjenigen seines rechten Nachbarn (sofern vorhanden) und vertauscht die Elemente, falls dies nötig ist.
2. Beim Gerade/Ungerade-Tausch vergleicht jeder k -te Prozessor mit k *gerade* sein Element mit demjenigen seines rechten Nachbarn und vertauscht die Elemente, falls dies nötig ist.

10. Korrektheit und Analyse

8 P.

- (a) Zeigen Sie die Korrektheit des „Odd-Even Transposition Sort“-Verfahrens.
Hinweis: 0-1-Prinzip
- (b) Analysieren Sie die Kosten dieses Sortierverfahrens.

11. Programm

12 P.

- (a) Modifizieren Sie das Verfahren so, dass jedem Prozessor eine Teilliste von Zahlen zugeordnet wird, um die sehr feine Granularität zu erhöhen.
- (b) Implementieren Sie Ihr modifiziertes Verfahren in C und PVM.
- (c) Bestimmen Sie durch Zeitmessungen die im Vergleich zu sequentielltem Bubblesort erzielbaren Beschleunigungen auf 2, 4 und 8 Prozessoren beim Sortieren einer Liste mit 1000 ganzen Zahlen.

Auf der Vorlesungsseite finden Sie ein sequentielles Bubblesort-Programm und eine Beispielliste.