16. November 2011

Prof. Dr. R. Loogen, M. Dieterle, M. Grebe
Fachbereich Mathematik und Informatik
Hans-Meerwein-Straße
D-35032 Marburg

Philipps Universität Marburg

Übungen zu „Parallele und Verteilte Algorithmen“, Winter 2011/12

Nr. 5, Abgabe der Aufgaben: 23. November 2011 vor der Vorlesung

---

Aufgaben

5.1 Odd-Even-Transposition Sort

6 Punkte

Das Verfahren *Odd-Even-Transposition Sort* ist eine parallele Variante von Bubble-Sort. Dem i. Prozessor wird das Element a_i zugewiesen. Der Algorithmus besteht aus zwei sich wiederholenden Phasen:

- Beim Ungerade/Gerade-Austausch vergleicht jeder k. Prozessor (k ungerade) sein Element mit demjenigen seines rechten Nachbarn (sofern vorhanden) und vertauscht die Elemente falls dies nötig ist.
- Beim Gerade/Ungerade-Austausch vergleicht jeder k. Prozessor (k gerade) sein Element mit dem Element seines rechten Nachbarn und vertauscht diese falls nötig.

(a) Programmieren Sie Odd-Even-Transposition Sort mit Scala-Actors. / 4

(b) Bestimmen Sie die parallelen Kosten des Verfahrens (ausgehend vom PRAM-Modell, ignorieren Sie Kommunikationskosten). Wie viele parallele Schritte benötigt der Algorithmus? / 1

(c) Wie kann man auf einfache Weise mit weniger Prozessoren auskommen, ohne die Grundstruktur des Algorithmus zu ändern? / 1

Bitte wenden!

5.2 Bitonisches Sortieren

6 Punkte

Eine Variante des in der Vorlesung besprochenen *Odd-Even-Merge Sort* ist das *bitonische Sortieren*. Eine Folge von ganzen Zahlen $(a_i)_{i\in\{0\ldots n-1\}}$ heißt *bitonisch*, falls sie durch eine zyklische Verschiebung in eine aufsteigenden Teilfolge gefolgt von einer absteigenden Teilfolge transformiert werden kann, d.h.:

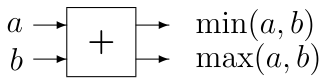
$$\begin{array}{rll} 1. & \exists i_0 \in \{0\ldots n-1\}: & a_0 \le a_1 \le \ldots \le a_{i_0} \ge a_{i_0+1} \ge \ldots \ge a_{n-1} \\ \text{oder } 2. & \exists k \in \{0\ldots n-1\}: & \text{die Folge } (a_{(i+k) \bmod n})_{i\in\{0\ldots n-1\}} \\ & & \text{(zyklische Verschiebung um } k\text{) erfüllt Bedingung 1} \end{array}$$

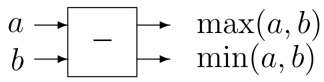
Eine zweielementige Folge ist stets bitonisch.

Es gilt der folgende **Satz**:

Seien $n = 2^l$ mit $l > 0$ und $(a_i)_{i\in\{0\ldots n-1\}}$ bitonische Folge.
Dann sind die Folgen $b_i = \min(a_i, a_{\frac{n}{2}+i})$ und $c_i = \max(a_i, a_{\frac{n}{2}+i})$ *ebenfalls bitonisch* (wobei $(i \in \{0\ldots \frac{n}{2}-1\})$). Weiter gilt: $b_j \le c_k \forall j,k \in \{0\ldots \frac{n}{2}-1\}$

(a) Skizzieren Sie ein Sortiernetzwerk, das mit aufsteigend und absteigend sortierenden *Komparatorbausteinen* aus einer beliebigen Folge zunächst rekursiv eine bitonische Folge herstellt und diese danach (nach Aussage des Satzes) sortiert. / 4





(b) Wie viele Schritte und elementare Bausteine benötigt das bitonische Sortierverfahren? Vergleichen Sie das Verfahren mit dem Sortiernetzwerk der Vorlesung. / 2