

Übungen zu „Parallele und Verteilte Algorithmen“, Winter 09/10

Prof. Dr. R. Loogen, Dipl.-Inform. Th. Horstmeyer · Fachbereich Mathematik und Informatik · Marburg

Nr. 4, Abgabe: Dienstag, 17. November 2009 vor der Vorlesung

Hinweise: Die Übungsblätter erscheinen dienstags, sind am darauffolgenden Dienstag abzugeben und werden bei Rückgabe im Tutorium besprochen. Auf jedem Blatt sind 12 Punkte erreichbar.

Bei Programmieraufgaben sind die Programme bitte *zusätzlich zur Papierversion* per Mail an horstmey @ mathematik abzugeben.

Die Abgabe ist in Gruppen bis zu zwei Personen erlaubt.

Aufgaben

4.1 Untere Schranken für paralleles Sortieren

3 Punkte

Begründen Sie die Korrektheit der im folgenden angegebenen unteren Schranken für das Sortieren von n Elementen auf verschiedenen Netzwerken mit jeweils n Knoten. Vor und nach dem Sortiervorgang sollen die zu sortierenden Elemente gleichmäßig verteilt sein, d.h. ein Element pro Prozessor.

- (a) $\Omega(n)$ auf einem eindimensionalen Gitter
- (b) $\Omega(\sqrt{n})$ auf einem zweidimensionalen Toroid
- (c) $\Omega(\log n)$ auf einem Shuffle-Exchange-Netzwerk

4.2 Sortiernetze

9 Punkte

Überlegen Sie sich, wie ein Odd-Even-Merger als paralleles Java-Programm modelliert werden kann.

- (a) Beschreiben Sie ihre Modellierung. Gehen Sie dabei insbesondere darauf ein, welche Aufgaben ein einzelner Thread übernimmt und wie die Werte zwischen den einzelnen Einheiten kommuniziert werden. / 3
- (b) Implementieren Sie einen (n,n) -Merger in Java gemäß ihrem Modell. / 4
- (c) Schreiben Sie zusätzlich eine Methode, die einen `int`-Array der Länge $2n = 2^k$ als Parameter erwartet, der an den Positionen 0 bis $n - 1$ bzw. n bis $2n - 1$ sortierte Folgen enthält. Diese sollen durch ihren Merger in einen komplett sortierten Array gemergt werden. / 2