PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen
Dipl.Inf. J.Berthold

D-35032 Marburg
Hans Meerwein Straße

27.Juni 2005

Übungen zu „Parallele Programmierung“, SS 2005

Nr. 11(letztes Blatt), Abgabe 07.Juli 2005 vor der Vorlesung

Hinweise:
Dieses Übungsblatt ist am **07.Juli 2005** in der Vorlesung abzugeben.
Am 4. und 5. Juli findet kein Tutorium statt.

Aufgaben

11.1 **Hyper-Quicksort** 6 Punkte

Das Verfahren *Hyper-Quicksort* sortiert ein Array mit Hilfe eines rekursiv-parallelen Verfahrens auf einem Hypercube-Netzwerk. Als Ausgangszustand seien die Elemente des Arrays bereits gleichmäßig auf alle Prozessoren verteilt. Es wird der folgende rekursive Algorithmus angewandt:

- Alle Prozessoren sortieren ihre lokalen Daten.
- Ein Prozessor wählt ein (für alle gemeinsames) Pivot-Element, welches mit einem Hypercube-Broadcast an alle anderen Prozessoren weitergegeben wird.
- Alle Prozessoren teilen ihre Teilliste gemäß dem Pivot in zwei Teile auf.
- Der “unpassende” Teil des lokalen Arrays wird an den Hypercube-Nachbarn *in der höchsten Dimension* geschickt und dort dem lokalen Array hinzugefügt. Prozessoren, in deren Adresse das höchstwertige bit gesetzt ist, empfangen die größeren Elemente.
- Der Hypercube wird entlang der höchsten Dimension in zwei Teil-Hypercubes geringerer Dimension geteilt.
- Das Verfahren wird auf den Teilhypercubes rekursiv so lange wiederholt, bis die Dimension 1 ist.

Die Prozessoren des Hypercubes, in aufsteigender Reihenfolge ihrer IDs betrachtet, enthalten nun das sortierte Array.

(a) Implementieren Sie das Verfahren in C + MPI, so dass die Listenlänge als Parameter vorgegeben werden kann.

(b) Bekanntlich hängt der Aufwand eines Quick-Sort-Verfahrens von der Datenaufteilung (bzw. Wahl des Pivots) ab.
Welchen Aufwand hat das Verfahren bei einer “guten” Aufteilung der Daten ?
Beschreiben Sie einen *worst-case* für die Datenaufteilung und den resultierenden Aufwand des Verfahrens für diesen Fall.

Bitte wenden!

11.2 Algorithmus von Cole – Skelett-Merging

6 Punkte

Gegeben seien sortierte endliche Folgen ganzer Zahlen X, Y, X' und Y' mit

$$X \propto X' \text{ und } Y \propto Y'$$

Zeigen Sie, dass unter diesen Voraussetzungen gilt:

(a) $X\&Y \propto X'$ und $X\&Y \propto Y'$

(b) Es gilt nicht, dass: $X\&Y \propto X'\&Y'$.

(c) Zwischen je k aufeinanderfolgenden Elementen von $X\&Y$ liegen höchstens $2k+2$ Elemente von $X'\&Y'$.

(d) $\frac{X\&Y}{4} \propto \frac{X'\&Y'}{4}$

11.3 Matrix-Transposition im Shuffle-Exchange-Netzwerk

3 Punkte

In einem Shuffle-Exchange-Netzwerk mit $n^2 = 2^{2q}$ Knoten sei eine $n \times n$-Matrix elementweise gespeichert, d.h. jeder Knoten $P_{i \cdot n+j}$ $(0 \leq i, j < n)$ enthält ein Element a_{ij} der Matrix.

Beschreiben Sie einen Algorithmus zur Transposition der gespeicherten Matrix.

Geben Sie Aufwand und Kosten Ihres Verfahrens an.