Universität Marburg
Fachbereich Mathematik
Prof. Dr. K. Böhmer

SS 2000

Übungen zur Vorlesung: Numerik I

Blatt 1

Abgabe: Montag den 17.4.2000, 11:00 Uhr,

Programmieraufgaben eine Woche später

1. Die **Hypercube**-Systeme haben $p = 2^n$ Prozessoren. Zu jeder Ecke $x \in \{0,1\}^n$ des $n-$dimensionalen Einheitswürfels gehört ein Prozessor P_x. Die Prozessoren P_x und P_y sind durch einen Datenweg verbunden, wenn die Ecken x und y Nachbarecken des Würfels sind. Jeder der 2^n Prozessoren hat genau n Datenwege. Man braucht maximal n Schritte, um ein Datenpaket zwischen zwei **beliebigen** Prozessoren auszutauschen. Wie berechnet man eine kürzeste Route zwischen zwei beliebigen Prozessoren eines Hypercubes? $\boxed{\text{4S}}$

2. Sei $v = (v_1, \ldots, v_n)^T \in \mathbb{R}^n$ ein Vektor mit der Dimension $n = 2^k$. Entwerfen Sie einen möglichst schnellen Algorithmus zur Berechnung der Summe der Vektorkomponenten $\boxed{\text{4M}}$

$$\sum_{i=1}^{n} v_i$$

auf einem Rechner mit Hypercube-Architektur (siehe Aufgabe 1.) mit $p = n/2 = 2^{k-1}$ Prozessoren. Der Komunikationsaufwand zwischen den Prozessoren sei vernachlässigbar.

3. Seien $x, y \in \mathbb{R}^n$ zwei Vektoren und $\alpha \in \mathbb{R}$ ein Skalar. Entwerfen Sie einen *parallelen* Algorithmus zur Berechnung einer **saxpy**-Operation (mnemotechnisch als skalar alpha x plus y zu merken) $\boxed{\text{4M}}$

$$z := \alpha x + y = (\alpha x_1 + y_1, \ldots, \alpha x_n + y_n)$$

auf einem Vektorrechner mit einer Pipeline der Länge 2.