Prof. Andreas Goerdt
Andreas Berger

TU Chemnitz
Wintersemester 2002/03

Parallele Algorithmen
1. Übung

1. Aufgabe

Das Matrix-Vektor-Produkt einer $n \times n$-Matrix $A = (a)_{ij}$ mit einem Vektor $b = (b_1, \ldots, b_n)^t$ ist definiert als

$$A \cdot b = \begin{pmatrix} \sum_{j=1}^{n} a_{1j} \cdot b_j \\ \vdots \\ \sum_{j=1}^{n} a_{nj} \cdot b_j \end{pmatrix}.$$

Eine Variante, dieses Produkt auf einer PRAM mit n Prozessoren zu berechnen, besteht darin, dass jeder Prozessor P_i zunächst einen Vektor $(a_{1i} \cdot b_i, \ldots, a_{ni} \cdot b_i)^t$ bestimmt. Dieser Schritt ist offensichtlich in paralleler Zeit $\mathcal{O}(n)$ möglich. Schließlich muss die Summe dieser n Vektoren, die jeweils im lokalen Speicher liegen, berechnet werden.

Geben Sie einen Algorithmus mit Zeitbedarf $\mathcal{O}(n)$ für diese Operation an!

2. Aufgabe

Berechnungen können durch gerichtete, azyklische Graphen (kurz: DAG) dargestellt werden.

Wir betrachten das Problem der Matrix-Matrix-Multiplikation zweier quadratischer $n \times n$- Matrizen. Dabei verwenden wir ein Verfahren, welches das Ergebnis entsprechend der Definition für diese Operation mit einer Arbeit von $\Theta(n^3)$ bestimmt.

(a) Zeichnen Sie den DAG für $n = 2$!

(b) Wieviele Knoten und welche Tiefe haben jeweils die entsprechenden Graphen G_n für die Multiplikation von $n \times n$- Matrizen?
Welche Konsequenzen ergeben sich daraus für eine parallele Realisierung des Verfahrens?

3. Aufgabe

Für eine Berechnung sei ein DAG mit k Knoten und Tiefe d gegeben. Zeigen Sie:

(a) Für jedes Schedule S (also für jede Parallelisierung) mit p Prozessoren gilt

$$Zeit(S) \geq \frac{1}{2}\left(\left\lceil \frac{k}{p} \right\rceil + d\right).$$

(b) Für jede Prozessoranzahl p gibt es ein Schedule S_p mit

$$Zeit(S_p) \leq \left\lfloor \frac{k}{p} \right\rfloor + d.$$

4. Aufgabe

Wir wenden das Prinzip aus Aufgabe 3(b) auf PRAM-Berechnungen an. Für ein Problem sei ein paralleler Algorithmus bekannt, der Instanzen der Größe n auf $p = p(n)$ Prozessoren in paralleler Zeit $T_p(n)$ mit einer Arbeit von $W(n)$ berechnet.

(a) In welcher Zeit $T_{p'}(n)$ kann das selbe Problem auf einer Maschine mit p' Prozessoren gelöst werden?

(b) Wieviele Prozessoren sind notwendig, um n Zahlen in Zeit $\mathcal{O}(\log n)$ zu addieren?

5. Aufgabe

Gegeben seien n Bits $b_1, ..., b_n$, die zu Beginn zusammen mit der Zahl n im globalen Speicher einer PRAM liegen. Zu berechnen ist das logische AND $a = b_1 \wedge ... \wedge b_2$ aller Bits.

(a) Geben Sie einen Algorithmus an, der dieses Problem auf einer EREW-PRAM mit p Prozessoren in Zeit $\mathcal{O}(\frac{n}{p} + \log n)$ löst!

(b) Lässt sich der Zeitbedarf durch Verwendung des CRCW-Modells verringern?

6. Aufgabe

Ein weiteres Modell für parallele Berechnungen stellt das sogenannte Netzwerkmodell dar. Die Prozessoren haben dabei im Gegensatz zur PRAM keinen Zugriff auf einen gemeinsamen Speicher. Der Informationsaustausch findet ausschließlich durch Kommunikation von im Netzwerk benachbarten Prozessoren statt.

Wir betrachten das Problem der „Präfix-Summe“. Zu einer gegebenen Anzahl n von Zahlen a_1 bis a_n sollen die folgenden n Teilsummen s_1 bis s_n berechnet werden:

$$s_i = \sum_{k=1}^{i} a_k.$$

Die Prozessoren seien linear durchnummeriert. Prozessor i besitze am Anfang den Wert a_i und soll nach Beendigung der Operation die Präfixsumme s_i lokal verfügbar haben. Geben Sie jeweils einen Algorithmus und dessen Zeitbedarf, abhängig von n, für die parallele Berechnung in den folgenden Netzwerken an.

(a) ein lineares Feld aus n Prozessoren

(b) ein 2-dimensionales, quadratisches Gitter aus $n = m^2$ Prozessoren

(c) ein Hyperwürfel aus $n = 2^d$ Prozessoren

Notieren Sie zum Vergleich die Programme für eine PRAM! Geben Sie Zeit- und Arbeitsbedarf an!