Übungen zur Vorlesung

Wissenschaftliches Rechnen – Paralleles Höchstleistungsrechnen

Prof. Dr. C. Engwer, S. Westerheide
http://wwwmath.uni-muenster.de/num/Vorlesungen/WissenschaftlichesRechnen_SS11/

Abgabe 4. 7. 2011. Abg. der Programmieraufgaben per Email an sebastian.westerheide@uni-muenster.de, schriftliche Abgabe Dienstags in der Vorlesung.

- Alle Programmierübungen müssen per Email und in ausgedruckter Form abgegeben werden.
- Achten sie darauf, ihr Programm ordentlich zu formatieren und gut zu kommentieren.

Auf diesem Aufgabenblatt sollen Sie sich mit der Isoeffizienz-Analyse für parallele Systeme auseinandersetzen. Machen Sie sich vor Bearbeitung der Übungsaufgaben klar, was eine Isoeffizienz-Funktion ist und wie sich die Gleichung für ihre Berechnung herleiten lässt.

Übung 1 Einfache Isoeffizienz-Analyse

Es sollen $n \in \mathbb{N}$ (Maschinen-)Zahlen auf einem Hypercube der Dimension $2^d, d \in \mathbb{N}$, addiert werden. Die Rechenzeit für eine Addition sei T_A $\big($der sequentielle Algorithmus benötigt also die Zeit $(n-1) \cdot T_A\big)$ und T_M sei die Zeit für den Nachrichtenaustausch. Jeder der P Prozessoren soll zuerst einen festen Anteil n/P addieren, dann wird im Baum aufaddiert. Die n Zahlen werden gleichmäßig auf die Prozessoren verteilt.

(a) Wie ist die parallele Ausführungszeit $T_P(n, P)$?

(b) Wir wollen die Addition isoeffizient skalieren. Von welcher Komplexität ist die Isoeffizienz-Funktion des Systems?

(c) Wie muss man demnach n bei Verdopplung der Prozessoranzahl P vergrößern, damit die Effizienz konstant bleibt?

Hinweis: Sie dürfen annehmen, dass $n - 1 \approx n$ und $n/P - 1 \approx n/P$.

4 Punkte

Übung 2 Isoeffizienz-Funktionen

Der Overhead hängt gewöhnlich nicht nur von der Anzahl der Prozesse P, sondern auch von der Problemgröße W ab. Falls eine Isoeffizienz-Funktion existiert, ist es daher trotzdem oft schwer oder unmöglich, sie als Funktion des Parameters P anzugeben. Betrachten wir zum Beispiel ein hypothetisches paralleles System mit Overhead

$$T_o(W, P) = P^{3/2} + P^{3/4} W^{3/4}.$$

Die Gleichung der Isoeffizienz-Funktion

$$K \cdot W_K(P) = T_o\big(W_K(P), P\big) = P^{3/2} + P^{3/4}\big(W_K(P)\big)^{3/4},$$

$K \in \mathbb{R}$ konstant, ist schwer zu lösen.

Manchmal ist es dennoch möglich zu untersuchen, wann die Effizienz bei steigender Prozessorzahl nicht schlechter wird. Dies ist genau dann der Fall, wenn T_o bei steigendem P und W höchstens wie W wächst. Besteht T_o wie oben aus mehreren Summanden, wird die Isoeffizienz-Funktion für jeden Teilsummanden berechnet. Der Summand von T_o, der das höchste Wachstum von W in Abhängigkeit von P hervorruft, bedingt die Asymptotik der Isoeffizienz-Funktion des gesamten parallelen Systems.

Welche Asymptotik besitzt demnach die Isoeffizienz-Funktion für ein paralleles System mit obiger Overhead-Funktion?

2 Punkte

Übung 3 Isoeffizienz-Analyse: Matrix-Vektor-Multiplikation im Vergleich

In der Vorlesung wurde gezeigt, dass für eine Matrix-Vektor-Multiplikation $\mathbf{y} = A\mathbf{x}$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, $A \in \mathbb{R}^{M \times N}$, $M, N \in \mathbb{N}$ bei einer blockweisen Aufteilung der Matrix über ein Prozessorfeld (ohne Rückverbindungen) mit *Cut-Through-Routing* eine Isoeffizienz-Funktion der Komplexitiät $\Theta\big(P(\operatorname{ld}\sqrt{P})^2\big)$ existiert.

Untersuchen Sie nun für $N = M$ den Fall, daß die Matrix A und der Vektor $\mathbf{x}$ zeilenweise auf P im Hypercube angeordneten Prozessoren verteilt sind, wie in Figure 1 angedeutet. Der Vektor $\mathbf{y}$ habe die gleiche Verteilung wie Vektor $\mathbf{x}$ und verwendet werde wieder *Cut-Through-Routing*. Da $\mathbf{x}$ auf jedem Prozessor zur Verfügung stehen muss, benötigen Sie für diesen Fall die Zeit

$$T_{All-to-All-Broadcast-HC} \approx t_s \operatorname{ld} P + t_b m(P-1)$$

für Nachrichten der Länge m.

(a) Wie ist nun die parallele Ausführungszeit bestehend aus Rechen- und Kommunikations-Zeit? (Bezeichnen Sie die Zeit für eine elementare Rechenoperation mit T_f; die Kommunikation besteht nur in dem *All-to-All-Broadcast-HC*)

(b) Von welcher Komplexität ist für diese Lösung die Isoeffizienz-Funktion? Welches ist der dominierende Term? (Sie können die Terme von $T_o = pT_p - W$ wie in Übung 2 getrennt betrachten, der algorithmisch komplexere setzt sich durch)

(c) Wenn mit beiden Lösungen (zeilen- und blockweise Verteilung für den Fall $M = N$) die gleiche Effizienz erreicht werden soll, für welchen Algorithmus ist dann bei steigendem P ein größeres N erforderlich (mit Begründung)?

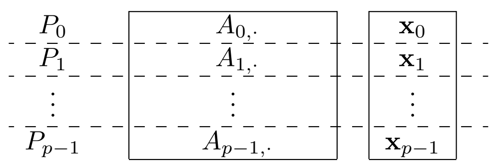

Figure 1: Zeilenweise Aufteilung einer Matrix A und eines Vektors $\mathbf{x}$ für das Matrix-Vektor-Produkt.

Hinweis: Sie dürfen annehmen, dass $P - 1 \approx P$ und $N - 1 \approx N$ ist (P und N groß).

4 Punkte