MATHEMATISCHES INSTITUT
DER UNIVERSITÄT ZU KÖLN

Dr. Stephanie Friedhoff, Dr. Martin Lanser

Einführung in HPC

Sommersemester 2016

Übung 6

Hinweis: Schreiben Sie bitte jede Aufgabe auf ein neues Blatt und auf **jedes Blatt Ihren Namen**. Auf die erste Seite Ihrer Übung schreiben Sie bitte zusätzlich zu Ihrem Namen Ihre Matrikelnummer.

Aufgabe 1 (2 + 2 + 1 Punkte).
Gegeben sei ein Parallelrechner bestehend aus $p = N^2$ Prozessoren, die in einem 2d-Torus der Größe $N \times N$ verbunden sind. Weiterhin seien $A, B \in \mathbb{R}^{n \times n}$ mit $n = mN$ Matrizen, die in Blöcke der Dimension $m \times m$ zerlegt seien:

$$A = \begin{pmatrix} A_{0,0} & \cdots & A_{0,N-1} \\ \vdots & & \vdots \\ A_{N-1,0} & \cdots & A_{N-1,N-1} \end{pmatrix}, \qquad B = \begin{pmatrix} B_{0,0} & \cdots & B_{0,N-1} \\ \vdots & & \vdots \\ B_{N-1,0} & \cdots & B_{N-1,N-1} \end{pmatrix}.$$

Wir nehmen an, dass die Matrizen A und B auf die p Prozessoren verteilt sind, sodass auf Prozessor $P_{i,j}$, $i, j = 0, \ldots, N-1$ die Blöcke

$$A_{i,(i+j) \mod N} \quad \text{und} \quad B_{(i+j) \mod N, j}$$

vorhanden sind.

(a) Skizzieren Sie die Verteilung der Daten für $N = 4$.

(b) Entwerfen Sie einen parallelen Algorithmus für die Matrizenmultiplikation $C = A \cdot B$, der nur Kommunikation zwischen (im Torus) **benachbarten Prozessoren** verwendet. Sie dürfen dabei nicht-blockierende Kommunikation annehmen.

Hinweis: Prozessor $P_{i,j}$ sollte den Block $C_{i,j}$ der Ergebnismatrix C berechnen. Welche Blöcke von A und B werden für diesen Block benötigt und auf welchen Prozessoren sind diese vorhanden?

(c) Wie sind die Matrizen A, B und C auf die Prozessoren verteilt nachdem Sie Ihren Algorithmus aus Teil (b) angewendet haben?

Aufgabe 2 (2 + 2 Punkte).
Es sei $\Omega = (0, 1) \times (0, 1)$ das offene Einheitsquadrat im zweidimensionalen Raum. Wir nehmen an, dass wir das Einheitsquadrat durch ein regelmäßiges Punktegitter mit einer Schrittweite von $h := 1/(N + 1)$ diskretisieren, welches wir durch

$$\Omega_h := \{(x_i, y_j) \ : \ x_i = ih, \ y_j = jh, \ i, j = 0, 1, \ldots, N + 1\}$$

definieren. Betrachten Sie die partielle Differentialgleichung

$$-\Delta u(x,y) + \frac{\partial^2 u(x,y)}{\partial x \partial y} = f(x,y), \quad (x,y) \in \Omega.$$

(a) Leiten Sie durch Taylor-Entwicklung und Vernachlässigung quadratischer Fehlerterme für die inneren Gitterpunkte den folgenden 9-Punkte Differenzenstern her:

$$\begin{matrix} -\frac{1}{4} & -1 & \frac{1}{4} \\ -1 & 4 & -1 \\ \frac{1}{4} & -1 & -\frac{1}{4} \end{matrix} .$$

Bemerkung: Differenzensterne sind Verallgemeinerungen der Differenzenquotienten für höhere Dimensionen. Obiger 9-Punkte-Stern angewendet auf die diskretisierte Funktion an einem inneren Gitterpunkt $(ih, jh),\ i,j = 1,\dots,N$ bedeutet

$$\frac{1}{4}u_{i-1,j-1} - u_{i,j-1} - \frac{1}{4}u_{i+1,j-1} - u_{i-1,j} + 4u_{i,j} - u_{i+1,j} \\ - \frac{1}{4}u_{i-1,j+1} - u_{i,j+1} + \frac{1}{4}u_{i+1,j+1} = h^2 f_{i,j}$$

(b) Geben Sie den zu der durch den Differenzenstern aus Teil (a) definierten Matrix A gehörenden Graphen $G(A)$ an. Ignorieren Sie dabei die Randpunkte.

Programmieraufgabe 5 (2 + 2 + 2 + 6 Punkte).
(**Information:** Für diese Aufgabe haben Sie zwei Wochen Zeit, d. h. der Abgabetermin für diese Aufgabe ist der 20. Juni 2016 bis 12:00 Uhr. Teil (a) - (c) werden als Theorieaufgabe gewertet.)

Gegeben sei eine regelmäßige Diskretisierung des Einheitsquadrats mit $(N+2) \times (N+2)$ Gitterpunkten und demnach $N \times N$ inneren Punkten (siehe auch Vorlesung und Aufgabe 2). Sei $n := N^2$. Wir betrachten die zugehörige Laplacematrix $A \in \mathbb{R}^{n \times n}$ für alle inneren Punkte und den 5-Punkte-Stern

$$\begin{matrix} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{matrix}$$

(a) Beschreiben Sie die Matrix-Vektor-Multiplikation $y = Ax$ komponentenweise unter Ausnutzung des 5-Punkte-Sterns, d. h. überlegen Sie sich eine Funktion f, die für einen minimalen, von Ihnen zu definierenden Teilvektor x_{I_i} von x die i-te Komponente y_i von y berechnet; $i = 1, ..., n$. Beachten Sie den Sonderfall von Punkten die direkt neben dem Rand liegen! Geben Sie einen kurzen Algorithmus an, der y aus x unter Ausnutzung von f berechnet, ohne dabei die Matrix A aufzustellen. Eine solche Matrix-Vektor-Multiplikation nennt sich auch „matrixfrei“.

(b) Ist die Reihenfolge (z. B. zeilenweise), in der die Komponenten y_i berechnet werden generell von Bedeutung für die Performance der matrixfreien Matrix-Vektor-Multiplikation? Falls ja, überlegen Sie sich eine möglichst effiziente Variante und erläutern Sie Ihre Überlegungen.

(c) Betrachten wir nun die **parallele** Matrix-Vektor-Multiplikation $y = Ax$. Gegeben sei dazu ein Parallelrechner mit $p = P^2$ Prozessoren und es sei $N = m * P^2$, $m \in \mathbb{N}$. Wir partitionieren unser Gitter einmal in Streifen mit je $m \times N$ Gitterpunkten und einmal in Quadrate mit je $mP \times mP$ Gitterpunkten (s. Abbildung 1). Jeder parallele Prozessor bearbeite genau eine der Partitionen, d. h. jeder Prozessor hält alle Zeilen der Matrix A und des Vektors x die zu Gitterpunkten aus der entsprechenden Partition gehören. Berechnen Sie Speed-up und Scale-up der parallelen matrixfreien Matrix-Vektor-Multiplikation für beide Varianten. Betrachten Sie dazu Rechen- und Kommunikationszeit wie üblich. Betrachten Sie nur den parallelen Fall mit Prozessorzahlen $p >= 9$. Kommentieren Sie Ihre Ergebnisse.

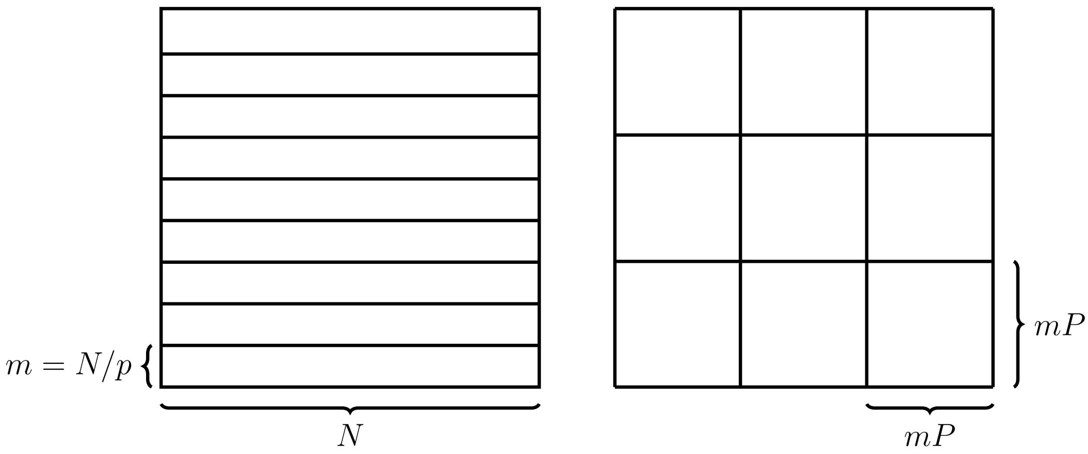

Abbildung 1: Partition des $N \times N$-Gitters in Streifen und Quadrate mit $p = P^2 = 9$.

(d) Implementieren Sie die parallele und matrixfreie Matrix-Vektor-Multiplikation der Laplace-Matrix für beide in Abbildung 1 vorgeschlagenen Partitionierungen. Testen Sie Ihr Programm für verschiedene p, n und z. B. mit $x = (1, ..., 1)^T$.

Hinweis: Für den Austausch der Werte in den Randpunkten der Partitionen ist es hilfreich für jeden Prozess lokal die Partition und dessen Rand der Breite 1 zu betrachten (sogenannte „ghost layer“). D. h. verwenden Sie lokal Streifen mit je $(m + 2) \times (N + 2)$ Gitterpunkten für die Partition in Streifen bzw. Quadrate mit je $(mP + 2) \times (mP + 2)$ Gitterpunkten.

Abgabedatum: 13. Juni 2016 bis 12:00 Uhr im entsprechenden Kasten in Raum 3.01 des Mathematischen Instituts oder am Ende der Vorlesung.