

Aufgabe 6: MPI-Matrix Operationen und Parallelisierungsschema der Poissongleichung

In den folgenden Aufgaben sollen MPI-Matrix-Operationen mit `scatterv` geübt und ein Konzept für die MPI-Implementation der Poisson-Gleichung erarbeitet werden.

Aufgabe 6A: MPI-Matrix Operationen mit Scatterv und Gatherv (90 Punkte)

Die Aufgabe 6A basiert auf dem Setup für Aufgabe 5A.

In der Aufgabe ist eine Integer-Matrix der Größe 24×24 zu erstellen. Diese Integer-Matrix $M(24, 24)$ wird in absteigender Reihenfolge, d.h. von $M(1, 1) = 576$ bis $M(24, 24) = 1$, initialisiert.

Diesmal wird diese Matrix für die Prozessoranzahl 5 und 7 mittels dem Befehl `SCATTERV` in ungleich große Teilmatrizen aufgeteilt. Dabei dürfen die einzelnen Prozesse nur die benötigten Teilmatrizen im Speicher halten.

Pro Teilmatrix wird die Operation $N(i, j) = N(i, j) * (\text{ProzessorID} + 1)$ ausgeführt. Nach dieser Berechnung wird von jeder Teilmatrix die Summe über alle Elemente jeder Teilmatrix errechnet.

Anschließend werden mittels des MPI Befehls `GATHERV` die Teilmatrizen zu einer neuen Gesamtmatrix $M(24, 24)$ zusammengeführt. Auch für die Gesamtmatrix wird die Summe über alle Elemente errechnet und diese Gesamtsumme mit der Summe verglichen, welche als Gesamtsumme der Teilmatrizen ermittelt wird.

Aufgabe 6B: Konzept für die Parallelisierung der Poisson-Gleichung nach Jakobi (120 Punkte)

Ziel der kommenden Aufgaben ist es die Poissongleichung mittels Nachrichtenaustausch mit MPI zu parallelisieren. Daher soll zuerst anhand des Jakobi Verfahrens das Parallelisierungsschema beschrieben werden.

Der Verlauf soll derart abgebildet werden, dass die gesamte Matrixdaten auf die Prozesse aufgeteilt werden, d.h. jeder Prozess bearbeitet nur einen Teil der Daten. Zu beachten ist ferner, dass zur Berechnung der Werte einer Zeile man immer die Werte der darüber- und der darunterliegenden Zeile braucht. Wenn also die zu berechnende Zeile die erste oder letzte des Blockes eines Prozesses ist, so wird die benötigte Nachbarzelle von einem Nachbarprozess verwaltet; dieser muss sie dann zusenden. Gleichermaßen muss man, wenn man eine Randzeile neu berechnet hat, diese an den entsprechenden Nachbarprozess weitersenden. Die Problematik liegt darin, dies zum richtigen Zeitpunkt mit den richtigen Werten zu tun.

Zu beachten ist ferner, dass der erste und letzte Prozess keinen vorherigen bzw. nachfolgenden Nachbarprozess mehr hat. Um das Speicherverhalten zu optimieren, dürfen die Prozesse nur die benötigte Teilmatrix im Speicher halten. Somit können auch Probleme berechnet werden, welche nicht in den Hauptspeicher eines einzelnen Knoten passen.

Einzugehen ist auf Probleme und Besonderheiten die für die Beendigung des Programmes notwendig sind. Abzugeben ist eine Beschreibung als PDF Datei, in der angegeben wird in welcher Form die Parallelisierung des Jakobi-Verfahrens für beide Kriterien (Iteration und Genauigkeit) umgesetzt werden soll, mit Details zum Ablauf und den verwendeten MPI Befehlen.

Das PDF Dokument sollte folgende Punkte enthalten:

- Prosabeschreibung der Datenaufteilung der Matrix auf die einzelnen Prozesse
 - Welche Daten der Matrix werden von welchem Prozess verwaltet?
- Parallelisierungsschema für das Jakobi Verfahren
 - Beschreibt aus Sicht eines Prozesses, wann die Berechnung und wann die Kommunikation mit seinem Nachbarn erfolgt.
 - Welche Daten benötigt der Prozess von seinem Nachbarn und wann tauscht er die Daten aus?
- Diskussion der Abbruchproblematik
 - Zu unterscheiden ist das Abbruchkriterium nach Iteration und Genauigkeit.
 - Wie wird das Kriterium der Genauigkeit von allen Prozessen festgestellt, und wie erfolgt die Mitteilung das die Arbeit beenden wird?

Abgabe

Die auf dem Cluster lauffähigen FORTRAN Programme sollen als Quellcode und die Beschreibung als PDF mit der Angabe der Gruppe (Personen in der Gruppe) bis zum 30.5.2017 geschickt werden an:

ppg-abgabe@wr.informatik.uni-hamburg.de

Als Subject im Kopf der Mail bitte die Angabe: PPG-17 Blatt6 und die Liste der Familiennamen der Personen in der Übungsgruppe.