Universität Hamburg, Fachbereich Informatik
Arbeitsbereich Wissenschaftliches Rechnen
Prof. Dr. T. Ludwig, Anna Fuchs, Jannek Squar
Georg von Bismarck, Niclas Schroeter

Übungsblatt 8 zur Vorlesung
Hochleistungsrechnen
im WiSe 2021/2022
Abgabe: 04.12.2021, 23:59

Parallelisierung mit MPI (Jacobi: 240 Punkte)

Parallelisieren Sie das Jacobi-Verfahren in dem sequentiellen partdiff-Programm gemäß dem besprochenen Parallelisierungsschema.
Beachten Sie dabei folgende Anforderungen:

- Abbruch
  - Es gibt hier zwei Fälle, die auf Korrektheit der Parallelisierung zu prüfen sind:
    1. Abbruch nach fester Iterationszahl (beide Störfunktionen)
    2. Abbruch nach Genauigkeit (beide Störfunktionen)
  - Dabei soll nach gleicher Iterationszahl das Ergebnis (Matrix und Fehlerwert) identisch bleiben. Außerdem soll bei Abbruch nach Genauigkeit im parallelen Programm nach derselben Iterationszahl wie im sequentiellen abgebrochen werden.
  - Überprüfen Sie, dass die Ergebnisse mit 24 Prozessen auf zwei Knoten identisch zum sequentiellen (Original als Referenz nehmen) Fall sind.
- Code
  - Zu keinem Zeitpunkt darf ein Prozess die gesamte Matrix im Speicher halten. DIe Matrix soll auf alle Prozesse gleichmäßig verteilt werden.
  - Das Programm muss weiterhin mit einem Prozess funktionieren (kontrolliert Abbrechen zählt nicht als funktionieren).
  - Das Programm muss mit beliebigen Prozesszahlen funktionieren.
  - Erstellen Sie eine eigene Funktion für die MPI-Parallelisierung des Jacobi-Verfahrens.
  - GS muss dabei weiterhin (sequentiell) funktionieren.
  - **Hinweis:** Sie können die in den Materialien bereitgestellte `DisplayMatrix`-Funktion als Grundlage für die parallele Ausgabe der Matrix benutzen.
- Laufzeit
  - Das Programm darf nicht langsamer als die sequentielle Variante sein.
- Kommunikation
  - Sie dürfen die Funktionen `MPI_Send` und `MPI_Isend` **nicht** verwenden. Nutzen Sie stattdessen ggf. die Funktionen `MPI_Ssend` und `MPI_Issend`.
  - Jeder nichtblockierende Kommunikationsaufruf (meist beginnend mit `MPI_I...`) muss mit einem passenden `MPI_Wait` oder einem erfolgreichen `MPI_Test` abgeschlossen werden. Anderenfalls ist der Aufruf falsch.

Hybride Parallelisierung (60 Bonuspunkte)

Erweitern Sie Ihre MPI-Version des Jacobi-Verfahrens zusätzlich um OpenMP.

Leistungsanalyse

Ermitteln Sie die Leistungsdaten Ihres Hybrid-Programms und vergleichen Sie die Laufzeiten für folgende Konfigurationen in einem beschrifteten Diagramm:

- 3 Knoten $\times$ 12 Prozesse
- 3 Knoten $\times$ 24 Prozesse
- 3 Knoten $\times$ 1 Prozess $\times$ 12 Threads
- 3 Knoten $\times$ 1 Prozess $\times$ 24 Threads
- 3 Knoten $\times$ 2 Prozesse $\times$ 6 Threads
- 3 Knoten $\times$ 2 Prozesse $\times$ 12 Threads
- 3 Knoten $\times$ 12 Prozesse $\times$ 2 Threads

Verwenden Sie hierzu 512 Interlines. Der kürzeste Lauf sollte mindestens 10 Sekunden rechnen; wählen Sie geeignete Parameter aus!

Schreiben Sie eine halbe Seite Interpretation zu diesen Ergebnissen. Beachten Sie die bisherigen Vorgaben zu Messungen (3x, gleicher Knoten, Tabelle, etc.).
Hinweis: Es ist empfehlenswert die Störfunktion $f(x,y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$ zu verwenden, da der erhöhte Rechenaufwand das Skalierungsverhalten verbessert.

Abgabe des Programms

Abzugeben ist ein gemäß den bekannten Richtlinien erstelltes und benanntes Archiv. Das enthaltene und gewohnt benannte Verzeichnis soll folgenden Inhalt haben:

- Alle Quellen, aus denen Ihr Programm besteht; **gut** dokumentiert! (Kommentare bei geänderten Code-Teilen!)
  - Erwartet werden die Dateien `Makefile`, `askparams.c`, `partdiff.c` und `partdiff.h`.
  - **Optional:** Eine Ausarbeitung `leistungsanalyse.pdf` mit den ermittelten Laufzeiten und der Leistungsanalyse.
- Ein `Makefile`
  - **Optional:** Ein Target `partdiff-par-hybrid` für die Binärdatei `partdiff-par-hybrid`, welche die Hybrid-Parallelisierung umsetzt.
- **Keine** Binärdateien!

Senden Sie das Archiv an `hr-abgabe@wr.informatik.uni-hamburg.de`.