Übungsblatt 9 zur Vorlesung Parallele Programmierung

Abgabe: 22.01.2022, 23:59

Jun.-Prof. Dr. Michael Kuhn (michael.kuhn@ovgu.de)
Michael Blesel (michael.blesel@ovgu.de)
Parallel Computing and I/O • Institut für Intelligente Kooperierende Systeme
Fakultät für Informatik • Otto-von-Guericke-Universität Magdeburg
https://parcio.ovgu.de

---

1. Parallelisierung mit MPI (600 Punkte)

Nun soll auch das Gauß-Seidel-Verfahren mittels Nachrichtenaustausch mit MPI parallelisiert werden. Dies soll natürlich so geschehen, dass auch ein Aufruf des Programms mit dem Jacobi-Verfahren noch dieselben Resultate wie im seriellen Fall liefert.

Überprüfen Sie zunächst die Korrektheit der Parallelisierung des Gauß-Seidel-Verfahrens für die beiden Fälle Abbruch nach Iterationszahl und Abbruch nach Genauigkeit.

Beim Abbruch nach Iterationszahl muss das Ergebnis identisch zum seriellen Programm sein. Beim Abbruch nach Genauigkeit muss die parallele Variante nicht unbedingt bei derselben Iteration wie die serielle abbrechen. Das Ergebnis muss bei gleicher Iterationszahl jedoch nach wie vor identisch zur seriellen Variante und unabhängig von der Prozessanzahl sein. Wichtig: Wenn dies nicht der Fall ist, ist Ihr Programm falsch!

Desweiteren stellen Sie bitte sicher, dass auch nach diesem Arbeitsschritt das Jacobi-Verfahren einwandfrei funktioniert.

Vorgaben und Hinweise

- Das Programm darf nicht langsamer als die serielle Variante sein.
- Zu keinem Zeitpunkt darf ein Prozess die gesamte Matrix im Speicher halten.
- Das Programm muss weiterhin mit einem Prozess funktionieren.
- Das Programm muss mit beliebigen Prozesszahlen funktionieren.
- Erstellen Sie eine eigene Funktion für die MPI-Version des Gauß-Seidel-Verfahrens.

Hinweis: Die Bearbeitungszeit ist schwer zu schätzen, da sie sehr stark von Ihren Vorkenntnissen und dem Glück, mit dem Sie auf Anhieb eine einigermaßen fehlerfreie MPI-Implementierung hinbekommen, abhängt. Bei komplexen Fehlern kann sich der Aufwand aber leicht stark erhöhen, fangen Sie deshalb frühzeitig an.

2. Leistungsanalyse (120 Bonuspunkte)

Ermitteln Sie die Leistungsdaten Ihres Programms und visualisieren Sie die Laufzeiten für folgende Konfigurationen in einem Diagramm. Wenn im Folgenden von einer Konfiguration (K, P, I) die Rede ist, so ist damit immer gemeint, dass das Programm auf K Knoten mit insgesamt P gleichmäßig auf den Knoten verteilten Prozessen und I Interlines laufen soll.

(1, 1, 836), (1, 2, 1.182), (1, 3, 1.448), (1, 6, 2.048), (1, 12, 2.896), (1, 24, 4.096),
(2, 48, 5.793), (4, 96, 8.192), (8, 192, 11.585)

Vergleichen Sie außerdem ihr Programm mit der ursprünglichen seriellen Variante. Der kürzeste Lauf sollte mindestens 30 Sekunden rechnen; wählen Sie geeignete Parameter!

Wiederholen Sie dabei jede Messung mindestens drei Mal, um aussagekräftige Mittelwerte bilden zu können. Dafür können Sie beispielsweise das Werkzeug `hyperfine` benutzen, das Sie mit `module load hyperfine` laden können. Es ist außerdem empfehlenswert die Störfunktion $f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$ zu verwenden, da der erhöhte Rechenaufwand das Skalierungsverhalten verbessert.

Für `K` Knoten, `P` Prozesse, `N` Iterationen und `I` Interlines können Sie folgendes Script benutzen:

```
#!/bin/bash

#SBATCH --nodes=K
#SBATCH --ntasks=P
#SBATCH --exclusive
#SBATCH --partition=vl-parcio

srun --mpi=pmi2 ./partdiff 1 1 I 2 2 N
```

Die Messungen sollen dabei mit Hilfe von SLURM auf den Rechenknoten durchgeführt werden. Geben Sie die für die Messungen verwendete Hardwarekonfiguration (Prozessor, Anzahl der Kerne, Größe des Arbeitsspeichers etc.) an.

Visualisieren Sie alle Ergebnisse in hinreichend beschrifteten Diagrammen. Schreiben Sie ca. eine viertel Seite Interpretation zu diesen Ergebnissen.

Abgabe

Als Abgabe erwarten wir ein gemäß den Vorgaben benanntes komprimiertes Archiv (`MustermannMusterfrau.tar.gz`), das ein gemäß den Vorgaben benanntes Verzeichnis (`Mustermann-Musterfrau`) mit folgendem Inhalt enthält:

- Eine Datei `gruppe.txt` mit den Gruppenmitgliedern (eines je Zeile) im folgenden Format:

  ```
  Erika Musterfrau <erika.musterfrau@example.com>

  Max Mustermann <max.mustermann@example.com>
  ```

- Der überarbeitete Code des `partdiff`-Programms im Verzeichnis `pde` (Aufgabe 1)

- Optional: Eine Ausarbeitung `leistungsanalyse.pdf` mit den ermittelten Laufzeiten und der Leistungsanalyse (Aufgabe 2)

Laden Sie das Archiv unter einer der folgenden URLs hoch:

1. Dienstag 7–9 Uhr (Julian Benda): `https://cloud.ovgu.de/s/aFtrkZ7cm43gifk`
2. Mittwoch 15–17 Uhr (Michael Blesel): `https://cloud.ovgu.de/s/3krS2Ep6pGyg8sn`
3. Mittwoch 15–17 Uhr (Johannes Wünsche): `https://cloud.ovgu.de/s/JsKdK5xBYbLjE22`