Universität Hamburg, Fachbereich Informatik
Arbeitsbereich Wissenschaftliches Rechnen
Prof. Dr. T. Ludwig, Dr. Jannek Squar

Übungsblatt 05 zur Vorlesung
Hochleistungsrechnen
im WiSe 2026/2027
Abgabe: 14.11.2026, 23:59

---

1. Datenaufteilung (60 Punkte)

In vielen Fällen muss die Datenaufteilung explizit und manuell vorgenommen werden. Überlegen Sie sich eine geeignete effiziente Aufteilung der `partdiff` Matrix auf die Threads.
Hinweis: Sämtliche Indizierung (z.B. Thread-ID, Prozess-Rang, Zeilen-/Spalten-Index) soll immer bei 0 beginnen!

1. Entwickeln Sie Formeln in Abhängigkeit von $i \in \mathbb{N}_0$ Interlines und $T \in \mathbb{N}$ Threads um folgende Größen für einen Thread t zu bestimmen:
   - Anzahl der zu berechnenden Zeilen
   - Index Startzeile (inklusive)
   - Index Endzeile (inklusive)

   Beachten Sie, wie sich die Matrixgröße aus der Zahl der Interlines bestimmt. Achten Sie vor allem auf die Randzeilen-/spalten: Bedenken Sie, welche Bereiche nur gelesen und welche berechnet und geschrieben werden. Der Berechnungsaufwand der einzelnen Threads soll möglichst gleichmäßig verteilt sein. Bedenken Sie außerdem den Fall, dass es unter Umständen auch mehr Threads als Zeilen geben kann. Implementieren Sie die Berechnung der Start- und Endzeilen für jeden Thread anhand der entwickelten Formeln in `datenaufteilung.c`.

2. Gegeben seien drei Konfigurationen:

$$i = 0, \quad T = 5 \tag{1}$$

$$i = 0, \quad T = 10 \tag{2}$$

$$i = 1, \quad T = 5 \tag{3}$$

   Geben Sie die Implementierung in `datenaufteilung.c` ab. Geben Sie außerdem den Output des Programms im Terminal im PDF ab.

2. Parallelisierung mit POSIX-Threads (240 Punkte)

Parallelisieren Sie das **Jacobi-Verfahren** des **sequentiellen** Programms zur Lösung der Poisson-Gleichung mittels POSIX-Threads.

Folgende Bedingungen müssen Sie hierbei erfüllen:

- Korrektheit
  - Die parallele Variante muss dasselbe Ergebnis liefern wie die sequentielle: Gleiche Matrix und gleiche Norm des Fehlers. Sowohl der Abbruch nach Iterationszahl als auch der nach Genauigkeit müssen korrekt funktionieren und dieselben Ausgaben wie die sequentiellen Gegenstücke liefern!
- Code
  - Die Threads müssen **außerhalb** der `while`-Schleife erzeugt werden. Achten Sie dabei auf die nötige Synchronisation und den Geltungsbereich der Variablen.
  - Die Anzahl der Threads muss über den ersten Parameter gesetzt werden können.
  - Erinnerung: Wie immer gilt, dass Sie **keine** globalen Variablen einführen oder **`static`** verwenden dürfen.
  - Testen Sie Ihr Programm mit mehreren gleichen Läufen hintereinander, um so ggf. zeitbedingtes fehlerhaftes Verhalten aufzuspüren.
  - Gauß-Seidel muss weiterhin sequentiell funktionieren.
  - Ihr Programm soll bei Verwendung von `-Wall` und `-Wpedantic` ohne Fehler oder Warnungen kompilieren.
- Laufzeit
  - Das parallelisierte Programm sollte bei Ausführung mit 12 Threads, 512 Interlines und der komplexen Störfunktion einen Speedup von ungefähr 8 erreichen. 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.

Generell gelten für **alle** Messungen in HLR die verabredeten Bedingungen bzgl der Durchführung und Dokumentation

3. Leistungsanalyse (60 Punkte)

Ermitteln Sie die Leistungsdaten Ihres POSIX-Thread-Programms und visualisieren Sie die Laufzeiten für jeweils 1–12 Threads in einem beschrifteten Diagramm. Vergleichen Sie außerdem ihr Programm mit der ursprünglichen sequentiellen Variante und der parallelisierten Variante mit einem Thread. Verwenden Sie hierzu 512 Interlines. Der kürzeste Lauf sollte mindestens 30 Sekunden rechnen; wählen Sie geeignete Parameter aus!

Erläutern Sie die Ergebnisse Ihrer Messungen und ziehen Sie entsprechende Schlussfolgerungen. Die Messungen sollen mit Hilfe von SLURM auf den *west*-Rechenknoten durchgeführt werden; führen Sie die Messungen **nicht** auf dem Login-Knoten aus.

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

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

- Ein PDF mit der Datenaufteilung, den ermittelten Laufzeiten und der Leistungsanalyse
- Das vollständige, modifizierte Programm `datenaufteilung`
- Das vollständige, modifizierte Programm `partdiff`

Achten Sie darauf, dass Ihre Programme ohne Warnungen oder Fehler bauen und Valgrind keine Probleme anzeigt. Packen Sie ein komprimiertes Archiv (`.tar.gz`) aus dem sauberen Verzeichnis (**ohne Binärdateien oder versteckte Dateien**). Benennen Sie das Archiv nach den Nachnamen der Gruppenmitglieder (z. B. `MustermannMusterfrau.tar.gz`).
Ein Mitglied Ihrer Gruppe legt das Archiv dann in

```
/home/USERNAME/HR-Abgaben-2627/Blatt-05
```

ab und schickt nach erfolgter Abgabe eine E-Mail an `jannek.squar@uni-hamburg.de`, in der Sie einfach den absoluten Pfad zu Ihrem Archiv angeben:

```
/home/USERNAME/HR-Abgaben-2627/Blatt-05/MustermannMusterfrau.tar.gz.
```

Setzen Sie Ihre Gruppenmitglieder auf CC.