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

Übungsblatt 04 zur Vorlesung
Hochleistungsrechnen
im WiSe 2026/2027
Abgabe: 07.11.2026, 23:59

---

1. Parallelisierung mit OpenMP (150 Punkte)

Parallelisieren Sie das **Jacobi-Verfahren** des **sequentiellen** Programms zur Lösung der Poisson-Gleichung mittels OpenMP. Beachten Sie hierbei die nachfolgenden Punkte:

- 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 Anzahl der Threads muss über den ersten Parameter gesetzt werden können.
  - Sie müssen die Klausel `default(none)` benutzen
  - Testen Sie Ihr Programm mit mehreren gleichen Läufen hintereinander, um so ggf. zeitbedingtes fehlerhaftes Verhalten aufzuspüren.
  - Gauß-Seidel muss weiterhin sequentiell funktionieren.
- Laufzeit
  - Das parallelisierte Programm sollte bei Ausführung mit 12 Threads und 512 Interlines einen Speedup von ungefähr 10 erreichen. Belegen Sie dies durch entsprechende Messungen. 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.
  - Wie immer gilt für alle Messungen in HLR, wenn nicht anders verabredet:
    * Wiederholen Sie jede Messung mindestens **drei** Mal, um aussagekräftige Mittelwerte und Standardabweichungen bilden zu können
    * Geben Sie die Messwerte in einer Tabelle ab
    * Die Messungen müssen auch den sequentiellen und das parallelisierte Programm mit einem Thread enthalten.
    * Messen Sie nur auf den West-Rechenknoten
- Scheduler
  - Testen Sie in Ihrer Implementation verschiedene Scheduler-Optionen von OpenMP, `static`, `dynamic` und `guided`, und vergleichen Sie deren Auswirkungen auf die Laufzeit. Belegen Sie Ihre Ergebnisse durch entsprechende Messungen.

Hinweise

Mit der Option `-fopenmp` erzeugt `gcc` OpenMP-Code. Ein Beispielprogramm ist unter `/home/hr/openmp/hello` verfügbar und lässt sich direkt aufrufen. Es arbeitet standardmäßig mit so vielen Threads, wie logische Cores vorhanden sind (24 auf den West-Rechenknoten). Den Quellcode dazu finden Sie unter `/home/hr/openmp/hello.c`.

Tutorials zur Programmierung mit OpenMP finden Sie unter folgender URL:

`https://hpc-tutorials.llnl.gov/openmp/`

Mit dem Tool `htop` auf dem Ausführungsort können Sie direkt die Auslastung der einzelnen Cores betrachten.

2. Leistungsanalyse (120 Punkte)

Messung 1

Ermitteln Sie die Leistungsdaten Ihres OpenMP-Programms und vergleichen Sie die Laufzeiten für jeweils 1–12 Threads in einem beschrifteten Diagramm.

- Vergleichen Sie Ihre Variante auch unbedingt mit dem sequentiellen ursprünglichen Programm
- Verwenden Sie hierzu 512 Interlines
- Der **kürzeste** Lauf sollte mindestens 30 Sekunden rechnen; wählen Sie geeignete Parameter aus

Messung 2

Ermitteln Sie weiterhin, wie Ihr OpenMP-Programm in Abhängigkeit von der Matrixgröße (Interlines) skaliert.

- Verwenden Sie hierzu 12 Threads und 11 Messungen zwischen 1 und 1024 Interlines (wobei Interlines = 2^i für $0 \leq i \leq 10$).
- Der längste Lauf sollte 20 bis 40 Minuten rechnen; starten Sie mit 1024 Interlines und wählen Sie geeignete Parameter aus. Für die folgenden Läufe können Sie die Interlines-Zahl dann entsprechend der gegebenen Formel verringern.

Visualisieren Sie alle Ergebnisse in hinreichend beschrifteten Diagrammen. Interpretieren Sie Ihre Ergebnisse.

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:

- Vollständiger und kommentierter Quellcode inkl. Makefile
- PDF mit den ermittelten Laufzeiten, den Leistungsanalysen und Erklärungen

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-04
```

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-04/MustermannMusterfrau.tar.gz.
```

Setzen Sie Ihre Gruppenmitglieder auf CC.