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

Übungsblatt 11 zur Vorlesung
Hochleistungsrechnen
im WiSe 2021/2022
Abgabe: 22.01.2022, 23:59

1 Leistungsevaluation des parallelen PDE-Lösers (200 Punkte)

Auf diesem Übungsblatt sollen Sie anhand von Messungen die Qualität Ihrer Parallelisierung vom letzten Übungsblatt bestimmen.

- Visualisieren Sie für die drei nachstehenden Fälle (1.1, 1.2 und 1.3) die Ergebnisse und interpretieren Sie diese.
- Welche Effekte können beobachtet werden? Interpretieren Sie diese.
- Alle Grafiken, Tabellen etc. müssen beschriftet und verständlich sein.
- Zeichnen Sie auch die Standardabweichung ein.

Durchführung

Um sinnvolle Messungen zu ermöglichen, wird wieder für alle Messungen das Batch-Queueing-System auf dem Cluster genutzt. Jede Messung soll **drei Mal** durchgeführt werden. (**Hinweis:** Die mitgelieferten Job-Scripte wiederholen die Messung bereits drei Mal.)

Die Job-Scripte für die Aufgaben werden auf der Materialienseite der Vorlesung zur Verfügung gestellt. Diese können bei Bedarf auch mit `SLURM_generator.sh` neu generiert werden. Des Weiteren kann das Skript `gather_results.sh` benutzt werden, um aus der Ausgabe der Job-Scripte Tabellen zu erstellen. Diese können zur Visualisierung der gewonnenen Daten genutzt werden. Zur Visualisierung soll ein geeignetes Werkzeug wie z.B. `gnuplot`[1] oder `pyplot`[2] genutzt werden.

Definition

Konfiguration (K, P, N) beschreibt einen Programmlauf auf K Knoten mit insgesamt P gleichmäßig auf den Knoten verteilten Prozessen und N Interlines.

1.1 Weak Scaling

Von Weak Scaling ist die Rede, wenn ein Algorithmus so skaliert, dass die Effizienz bei einer Anpassung der Problemgröße an die Prozesszahl konstant bleibt. Ermitteln Sie zuerst den Speedup Ihres Programms indem Sie es in den folgenden Konfigurationen laufen lassen:

(1, 1, 400), (1, 2, 564), (2, 4, 800), (4, 8, 1128), (4, 16, 1600), (4, 24, 1960), (8, 64, 3200)

[1] `http://www.gnuplot.info/`
[2] `https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html`

- Jacobi, 1.000 Iterationen, Störfunktion $f(x,y) = 2 \cdot \pi^2 \cdot \sin(\pi \cdot x) \cdot \sin(\pi \cdot y)$
- Gauß-Seidel, 1.000 Iterationen, Störfunktion $f(x,y) = 2 \cdot \pi^2 \cdot \sin(\pi \cdot x) \cdot \sin(\pi \cdot y)$

Frage: Wie erklären Sie sich die Wahl der Interlines in Bezug auf die Prozesszahl?

1.2 Strong Scaling

Von Strong Scaling ist die Rede, wenn ein Algorithmus so skaliert, dass auch bei gleicher Problemgröße mehr Kerne effizient genutzt werden können.
Lassen Sie Ihr Programm in folgenden Konfigurationen laufen:

(1, 12, N) (2, 24, N), (4, 48, N), (8, 96, N), (10, 120, N), (10, 240, N)

mit $N = 1920$. Nutzen Sie folgende Programmparameter:

- Jacobi, 500 Iterationen, Störfunktion $f(x,y) = 2 \cdot \pi^2 \cdot \sin(\pi \cdot x) \cdot \sin(\pi \cdot y)$
- Gauß-Seidel, 500 Iterationen, Störfunktion $f(x,y) = 2 \cdot \pi^2 \cdot \sin(\pi \cdot x) \cdot \sin(\pi \cdot y)$

1.3 Kommunikation und Teilnutzung der Knoten

Lassen Sie Ihr Programm in folgenden Konfigurationen laufen:

(1, 10, N) (2, 10, N) (3, 10, N) (4, 10, N) (6, 10, N) (8, 10, N) (10, 10, N)

mit $N = 200$. Nutzen Sie folgende Programmparameter:

- Jacobi, Genauigkeit $\leq 3.3504^{-5}$, $f(x,y) = 0$
- Gauss-Seidel, Genauigkeit $\leq 3.3504 \times 10^{-5}$, $f(x,y) = 0$

2 Visualisierung mit Vampir (180 Punkte)

Visualisieren Sie den Ablauf Ihrer Jacobi- und Gauß-Seidel-Parallelisierungen für folgende Kombinationen von Prozessen und Knoten (Prozesse, Knoten) und einer Anzahl von 20 Iterationen: (3, 2) und (5, 4).

- Erzeugen Sie die Spurdaten und visualisieren Sie sie mit Vampir (Hinweise zur Verwendung von Vampir finden Sie auf Übungsblatt 7)
- Interpretieren Sie die Visualisierung. Zu diesem Zweck erstellen Sie am besten Detailvergrößerungen der entsprechenden Zeitabschnitte in Vampir und von diesem Bild dann einen Screenshot (über `Window` → `Save Screenshot`). Die folgenden drei Zeitabschnitte müssen diskutiert werden:
  1. Die Startphase des Programms bis zum Ende der ersten Iteration in allen Prozessen.
  2. Die Phase der Synchronisation zwischen den Prozessen bei Ende einer Iteration. Bei Jacobi ist das der Zeitabschnitt, in dem alle Prozesse ihre Zeilen austauschen. Bei Gauß-Seidel nehmen wir denselben Zeitabschnitt, auch wenn ja dann aber die Prozesse in unterschiedlichen Iterationen sind.
  3. Die Endphase des Programms mit dem Einsammeln der Ergebnisse.
- Diskutieren und analysieren Sie hierbei insbesondere vorhandene Anomalien der Gesamtlaufzeiten.

Abgabe

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

- PDF-Dokument `speedup.pdf` mit den Grafiken, Tabellen und Interpretationen
- Die Tabellen mit den gemessenen Werten als Plaintext (*`case_name`*`.dat`)
- Ein PDF-Dokument `vampir_visualisierung.pdf` mit den Screenshots und einer Interpretation der Visualisierung
- Die Spurdaten (das gesamte `scorep`-Unterverzeichnis)

Senden Sie Ihre Abgabe an `hr-abgabe@wr.informatik.uni-hamburg.de`.

Bitte wählen Sie einen dem Zeitpunkt der Abgabe, der Anzahl der Gruppen und der Laufzeit angemessen Anfangszeitpunkt für die Bearbeitung dieses Übungsblattes, da die Auslastung des Clusters erfahrungsgemäß hoch sein wird. Begründungen für eine verspätete Abgabe wegen einer Überbelegung des Clusters werden wir nicht akzeptieren.