

---

## Leistungsevaluation des parallelen PDE-Lösers (200 Punkte)

Auf diesem Übungsblatt sollen Sie anhand von Messungen die Qualität Ihrer Parallelisierung vom letzten Übungsblatt bestimmen. 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 Materialiensseite der Vorlesung zur Verfügung gestellt. Diese können bei Bedarf auch mit `SLURM_generator.sh` neu generiert werden. Desweiteren kann das Tool `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`<sup>1</sup> genutzt werden.

### Definition

Wenn im Folgenden von einer **Konfiguration** (K, P, N) die Rede ist, so ist damit immer gemeint, dass das Programm auf K Knoten mit insgesamt P gleichmäßig auf den Knoten verteilten Prozessen und N Interlines laufen soll.

### 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, 1.128), (4, 16, 1.600), (4, 24, 1.960), (8, 64, 3.200)

Lassen Sie Ihr Programm in folgenden Konfigurationen laufen:

- 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?

### 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)

---

<sup>1</sup><http://www.gnuplot.info/>

wobei  $N = 1.920$ . Diesmal mit folgenden Programmparametern:

- 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  $f(x, y) = 2 \cdot \pi^2 \cdot \sin(\pi \cdot x) \cdot \sin(\pi \cdot y)$

## 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)

wobei  $N = 200$ . Programmparameter sollen dabei sein:

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

## 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
  - Welche Effekte sind zu beobachten? Versuchen Sie diese zu interpretieren!
  - Alle Grafiken, Tabellen etc. müssen beschriftet und verständlich sein! Zeichnen Sie außerdem auch die Standardabweichung ein.
- Die Tabellen mit den gemessenen Werten als Plaintext (`case_name.dat`)

Sollten Sie im Laufe der Analyse Ihr Programm noch korrigieren oder verbessern, können Sie die neue Version gerne in der auf dem letzten Aufgabenblatt beschriebenen Form erneut abgeben.

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 angemessenen Anfangszeitpunkt für die Bearbeitung dieses Übungsblattes. Ausreden für eine verspätete Abgabe wegen einer Überbelegung des Clusters werden wir nicht akzeptieren.**