
1 Qualitätsprüfung des parallelisierten PDE-Lösers (240 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 **3 Mal** durchgeführt werden.

Material

Die Job-Scripte für die Aufgaben werden auf der Materialiensseite der Vorlesung zur Verfügung gestellt. Diese können bei Bedarf auch mit `PBS_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 `gnuplot`¹ genutzt werden.

Tipp: Eine gute Einführung in die Verwendung von `gnuplot` finden Sie z. B. auf

<http://userpage.fu-berlin.de/~voelker/gnuplotkurs/gnuplotkurs.html>

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.

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 Prozessorzahl konstant bleibt. Ermitteln Sie zuerst den Speed-Up Ihres Programms indem Sie es in den folgenden Konfigurationen laufen lassen:

(1, 1, 100), (1, 2, 141), (1, 4, 200), (4, 8, 281), (4, 16, 400), (4, 24, 490), (5, 32, 566), (8, 64, 800)

Lassen Sie Ihr Programm in folgenden Konfigurationen laufen:

- Jacobi, Störfunktion $f(x, y) = 0$, 5.000 Iterationen
- Gauß-Seidel, Störfunktion $f(x, y) = 0$, 5.000 Iterationen

Wie erklären Sie sich die Wahl der Interlines in Bezug auf die Prozessorzahl?

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, 180, N), (10, 240, N)

wobei $N = 960$. Diesmal mit folgenden Programmparametern:

- Jacobi, Störfunktion $f(x, y) = 0$, 2000 Iterationen
- Gauß-Seidel, Störfunktion $f(x, y) = 0$, Genauigkeit $\leq 1.789 \times 10^{-4}$

¹<http://www.gnuplot.info/>

1.3 Kommunikation und Teilnutzung der Knoten

Lassen Sie Ihr Programm in folgenden Konfigurationen laufen:

(A)	(1, 12, N)	(2, 12, N)	(3, 12, N)	(4, 12, N)	(6, 12, N)	(8, 12, N)	(10, 12, N)
(B)	(1, 1, N)	(1, 2, N)	(1, 3, N)	(1, 4, N)	(1, 6, N)	(1, 8, N)	(1, 10, N) (1, 12, N)
(C)	(1, 1, N)	(2, 2, N)	(3, 3, N)	(4, 4, N)	(6, 6, N)	(8, 8, N)	(10, 10, N)

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

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

Abgabe

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

- PDF-Dokument `speedup.pdf` mit den Grafiken, Datentabellen und Interpretationen (Welche Effekte können Sie beobachten? Versuchen Sie diese zu interpretieren!)
- Die Tabellen mit den gemessenen Werten (Zeit und Flops) als Plaintext (`case.name.dat`)
- Das `gnuplot`-Script zum Generieren (*aller*) Ihrer Grafiken (`speedup.gnuplot`)

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 per E-Mail an `hr-abgabe@wr.informatik.uni-hamburg.de`.

Übersicht der durchzuführenden Messungen

Fall	Konfigurationen							
1.	(1, 1, 100)	(1, 2, 141)	(1, 4, 200)	(4, 8, 281)	(4, 16, 400)	(4, 24, 490)		
	(5, 32, 566)	(8, 64, 800)						
2.	(1, 12, N)	(2, 24, N)	(4, 48, N)	(8, 96, N)	(10, 120, N)	(10, 180, N)	(10, 240, N)	
3.(A)	(1, 12, N)	(2, 12, N)	(3, 12, N)	(4, 12, N)	(6, 12, N)	(8, 12, N)	(10, 12, N)	
3.(B)	(1, 1, N)	(1, 2, N)	(1, 3, N)	(1, 4, N)	(1, 6, N)	(1, 8, N)	(1, 10, N)	(1, 12, N)
3.(C)	(1, 1, N)	(2, 2, N)	(3, 3, N)	(4, 4, N)	(6, 6, N)	(8, 8, N)	(10, 10, N)	

Daraus ergibt sich zur Abschätzung der Laufzeit (Obergrenze):

Case	Ref. [s]	Erwartete Laufzeit [s]
Weak scaling	80	$9 \times 2 \text{ Konfigurationen} \times 3 \text{ Samples} \times 80 \text{ s} = 4320 \text{ s}$
Strong Scaling	480	$7 \times 2 \text{ Konfigurationen} \times 3 \text{ Samples} \times 480 \text{ s} = 20160 \text{ s}$
Kommunikation/Teilnutzung	60	$21 \times 2 \text{ Konfigurationen} \times 3 \text{ Samples} \times 60 \text{ s} = 7560 \text{ s}$

$\Rightarrow \text{Summe} = 32040 \text{ s} \approx 9 \text{ h}$

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.