

## 1 Qualitätsprüfung des parallelisierten PDE Löser (150 Punkte)

Auf diesem Übungsblatt sollen sie anhand von Messungen die Qualität ihrer Parallelisierung vom letzten Übungsblatt bestimmen.

Um sinnvolle Messungen zu ermöglichen, bei denen nicht zufällig gleichzeitig messende Gruppen sich gegenseitig die Ergebnisse verfälschen, werden Sie ab sofort die Ressourcenverwaltung des PVS-Cluster nutzen. Diese sorgt dafür, dass anstehende Jobs in geregelter Weise nacheinander bzw. nebeneinander auf disjunkten Knotenmengen (oder Prozessormengen) ausgeführt werden. Es kommt hierbei das PBS-kompatible TORQUE zusammen mit dem Scheduler MAUI zum Einsatz.

Ihr bisher genutzter direkter Zugang auf die Rechenknoten `nodeXX` wird gesperrt und Sie müssen auch keinen MPD-Ring mehr aufbauen.

Alle weiteren notwendigen Informationen finden Sie im Wiki (Benutzer-Quickstart).

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

### 1.1 Speed-Up-Bestimmung

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

$(1, 1)$ ,  $(2, 2)$ ,  $(3, 3)$ ,  $(4, 4)$ ,  $(5, 5)$  und  $(6, 6)$

Erstellen sie eine Tabelle für die folgenden Programmparameter:

- Jacobi, 100 Interlines, Störfunktion  $f(x, y) = 0$ , 5000 Iterationen
- Jacobi, 100 Interlines, Störfunktion  $f(x, y) = 2\pi$ , 5000 Iterationen
- Gauß-Seidel, 100 Interlines, Störfunktion  $f(x, y) = 0$ , 5000 Iterationen
- Gauß-Seidel, 100 Interlines, Störfunktion  $f(x, y) = 2\pi$ , 5000 Iterationen

Erzeugen Sie aus ihrer Tabelle Grafiken mit Hilfe von `gnuplot`.

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

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

## 1.2 Weitere Analyse

Um mehr über das Verhalten Ihres Programms zu erfahren, lassen sie es in folgenden Konfigurationen laufen:

(1, 1) (1, 2), (2, 3), (2, 4), (3, 5), (3, 6), (4, 7) und (4, 8)

Erstellen Sie auch hier eine Tabelle, diesmal mit folgenden Programmparametern:

- Jacobi, 100 Interlines, Störfunktion  $f(x, y) = 0$ , Genauigkeit  $\leq 6,5 \cdot 10^{-5}$
- Jacobi, 100 Interlines, Störfunktion  $f(x, y) = 2\pi$ , Genauigkeit  $\leq 7,3 \cdot 10^{-6}$
- Gauß-Seidel, 100 Interlines, Störfunktion  $f(x, y) = 0$ , Genauigkeit  $\leq 6,3 \cdot 10^{-5}$
- Gauß-Seidel, 100 Interlines, Störfunktion  $f(x, y) = 2\pi$ , Genauigkeit  $\leq 1,4 \cdot 10^{-5}$

Erzeugen Sie aus der Tabelle wieder Grafiken mit **gnuplot**.

Welche Effekte können Sie beobachten? Versuchen Sie diese zu interpretieren.

## Abgabe

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

- die Tabellen mit den gemessenen Werten als Plaintext (**speedup.txt** und **analyse.txt**)
- das GNUplot-Skript zum generieren ihrer Grafiken (**speedup.gnuplot** und **analyse.gnuplot**)
- die erzeugten Grafiken (**speedupX.png** und **analyseX.png** mit  $X \in 1, \dots$ )
- Ihre Interpretation der Analyseresultate (**interpretation.txt**)  
Wer möchte kann das ganze auch gerne zusätzlich zu einem schönen PDF-Artikel zusammenfassen und beilegen.

Wer im Laufe der Analyse sein Programm noch korrigiert oder verbessert, kann die neue Version gerne in der auf dem letzten Aufgabenblatt beschriebenen Form extra abgeben (bitte in einer separaten E-Mail).

Senden Sie Ihre Abgabe per Mail an:

`stephan.krempel@pvs.informatik.uni-heidelberg.de`