

Aufgabe 2: FORTRAN Zugriffsoptimierung Übungsaufgaben

Dieses Übungsblatt umfasst im Wesentlichen die Aufgabe zur Optimierung von sequentiellen FORTRAN Programmen.

Dazu wird auf der Webseite ein Programm bereitgestellt, das mit den Kenntnissen aus der Vorlesung optimiert werden sollen.

Wie gehabt, sollten Probleme auftauchen, wendet Euch bitte an die Mailingliste:

`PPG-19@wr.informatik.uni-hamburg.de`

Aufgabe 2A: Zugriffsoptimierung (120 Punkte + 60 Bonuspunkte)

Das Testprogramm

`SoA_bad_Access.f90`

berechnet die Durchschnittswerte der Farbkanäle mehrerer Pixel Codes aber mit einer schlechten Performance.

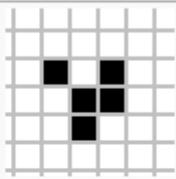
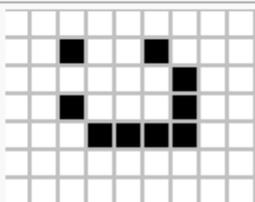
Daher soll als Standardaufgabe das Zugriffsmuster und die Implementierung des Speicher-Layouts optimiert werden, dafür werden je 60 Punkte vergeben. Für zusätzliche Optimierung können zusätzlich 60 Bonuspunkte erreicht werden.

Für die Abgabe soll zunächst der Ist-Zustand des Programms mittels dem Linux `time` Kommando mit drei Messungen als Basiswert festgehalten werden. Die Änderungen am Programm sollen mit einem kurzen Text in einer PDF Datei beschrieben werden. Hierzu gehört auch die erreichte Verbesserung der Laufzeit (jeweils Mittelwert und Standardabweichung aus den 3 Messungen) mit einer Tabelle oder als Grafik zu dokumentieren. Optional kann für die Messung der Cache-Misses auch die Software `perf` auf dem eigenen Rechner genutzt werden.

Aufgabe 2B: Game of Life - Vollständig (90 Punkte)

In der neuen Aufgabe soll das Konzept von Conway's "Game of Life" in einem Programm **vollständig** umgesetzt werden. Als Basis dient Euer Programm, das Ihr für das Übungsblatt 1 erstellt habt. Die Programmstruktur die in Übungsblatt 1 vorgegeben war soll beibehalten werden.

Als Muster sollen der “Glider” und das “Lightweight spaceship (LWSS)” verwendet werden, siehe Abbildung “Spaceships”.

Spaceships	
Glider	
Lightweight spaceship (LWSS)	

Quelle: http://en.wikipedia.org/wiki/Conway's_Game_of_Life

Da die Figuren nicht stationär sind, werden zyklische Randbedingung angenommen, d.h. wenn die Figur die Matrix z.B. am rechten Rand verlassen hat soll sie am linken Rand wieder in die Matrix eingefügt werden.

Es sollen die “Lebenszyklen” der verschiedene Muster mit 160 Iterationen durchlaufen und die Muster entsprechend dargestellt werden. Für die Ausgabe stellen wir das Programm Glider Movie auf unserer Webseite zur Verfügung, dessen Ausgaberroutine Unicode-Zeichen verwendet um eine gute lesbare Darstellung zu erreichen. Ihr dürft beliebige Teile von Glider Movie wiederverwenden, die **Subroutine printTwoDLogical** müsst Ihr verwenden.

Abgabe

Die auf dem Cluster lauffähigen FORTRAN Programme sollen bis zum Dienstag den 30.04.2019 geschickt werden an:

ppg-abgabe@wr.informatik.uni-hamburg.de

dabei ist zu beachten:

1. **Dokumentation der Programm-Optimierung**, sowie
2. **NUR den Quellcode inkl. dem Makefile** schicken, und
3. alles **als komprimiertes Archiv .tgz** schicken! D.h. es soll wirklich nur **ein einzelnes Archiv** geschickt werden!

Als Subject im Kopf der Mail bitte die Angabe: PPG-19 Blatt 2 und die Liste der Familiennamen der Personen in der Übungsgruppe.