

1. Übung *Sequenzieller Löser* zur Vorlesung High Performance Computing im WS 16/17

Zu editierende Dateien:

- main.c
- sweeps/gameoflife.c
- (Optional) sweeps/mandelbrot.c
- simdata.h
- field.h
- filling.h
- vtk.h

Benötigte Dateien:

- sweeps/add.c

Aufgabe 1: Implementieren des Sweeps

Navigieren Sie auf der Konsole zum Labor template Ordner in ihrem home Verzeichnis: `cd lab1`

Erstellen Sie einen Ordner build und führen sie dort cmake aus:

```
mkdir build
cd build
cmake ..
```

Mit dem Aufruf von ‘make’ wird der Quelltext in eine ausführbare Datei übersetzt. Dabei wird das Programm `bin/hpc.omp` erzeugt. Um das Programm ausführen zu können, müssen Sie noch folgende Schritte umsetzen:

- Öffnen Sie die Datei `main.c` und füllen Sie die fehlenden Schritte der `main`-Funktion aus. Verwenden Sie hierzu ausschließlich fertige Funktionen aus der Vorlage.
- Wechseln Sie nun in das Verzeichnis `sweeps`. Implementieren Sie den Kernel für Conway’s Game of Life. Nutzen Sie hierzu die Makros `ALIVE` für eine lebende Zelle und `DEAD` für eine tote Zelle. Beachten Sie außerdem die notwendigen Getter/Setter aus der Datei `field.h`.
- (Optional) Implementieren Sie den Kernel für die Berechnung der Mandelbrot-Menge. Dieser berechnet für jede Zelle im Gebiet die Anzahl der Iterationen und gibt diese zurück. Falls die Berechnung endlos dauern würde, brechen Sie die Schleife nach `MAX_ITERATIONS` ab und geben Sie den Wert 0 zurück.

Zum Starten des Programms, verwenden Sie folgende Syntax:

```
./hpc.omp <sweep> <x blocks> <y blocks> <x block size> <y block size> <number  
of timesteps> <prefix>
```

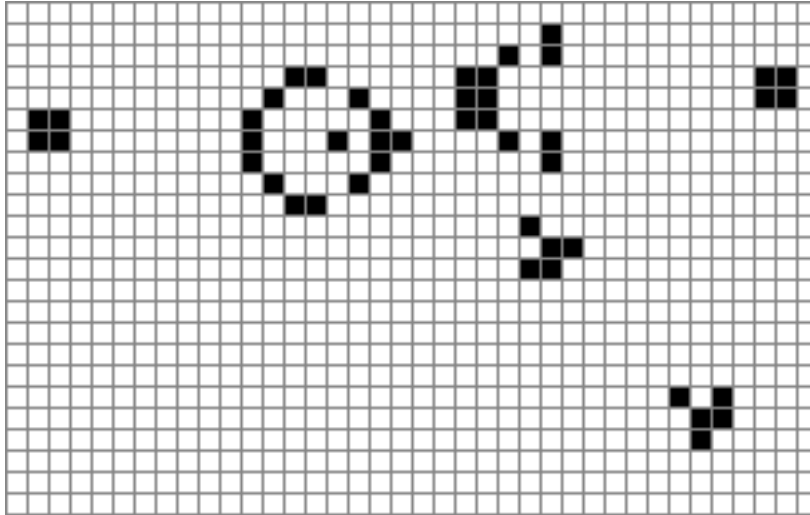
Setzen Sie die Gebietsgröße auf 256^2 , die Anzahl der Zeitschritte auf 10 und den Präfix für die Ausgabe auf `gol`. Wenn alles funktioniert hat, sollten Sie anschließend im Verzeichnis 10 Dateien nach dem Schema `gol-*.vtk` sehen.

Aufgabe 2: Visualisierung der Simulationsdaten

Öffnen Sie die VTK-Dateien mit Paraview. Bestätigen Sie nach dem Öffnen das Laden der Datei mit **Apply**.

- a) Ändern Sie das Farbschema und navigieren Sie durch die Frames.
- b) Validieren Sie ihr Simulationsergebnis, indem Sie eine "Gosper glider gun", verwenden.

<http://www.conwaylife.com/wiki/Gun>



- c) Validieren Sie ihr Simulationsergebnis, indem Sie eine geeignete Füllung (siehe `filling.h`), verwenden.
- d) (Optional) Falls Sie in Aufgabe 1 den sweep für die Berechnung der Mandelbrot-Menge implementiert haben, addieren Sie in Paraview für jeden Pixel den Wert 1 dazu und visualisieren Sie ihr Simulationsergebnis in Logarithmischer Darstellung.

Aufgabe 3: Performanz-Analyse

Führen Sie eine Laufzeitanalyse des Programms für verschiedene Gebietsgrößen durch. Erstellen Sie sich hierzu ein Laborlogbuch, welches Sie ab dieser Übung zu den Folgeterminen ebenfalls wieder zur Verfügung haben.

- a) Führen Sie den Löser jeweils 5 mal für die drei verschiedenen Gebietsgrößen 1024^2 , 2048^2 und 4096^2 aus. Achten Sie darauf, dass Sie mindestens 10, jedoch nicht mehr als 50 Zeitschritte rechnen. Deaktivieren das schreiben der Daten für die Messungen. Nutzen Sie das sbatch system, damit ihre Messungen auf einem Clusterknoten unabhängig durchgeführt werden können.
- b) Berechnen Sie, wie viel Speicherplatz im Arbeitsspeicher und auf der Festplatte für jeden der Gebietsgrößentests benötigt wird. Vergleichen Sie ihre berechneten Werte mit den Messungen.
- c) Tragen Sie die gemessenen Zeiten, sowie die Durchschnittswerte von jedem Test in Ihr Laborlogbuch ein.