

Praktikum Parallele Programmierung 2012

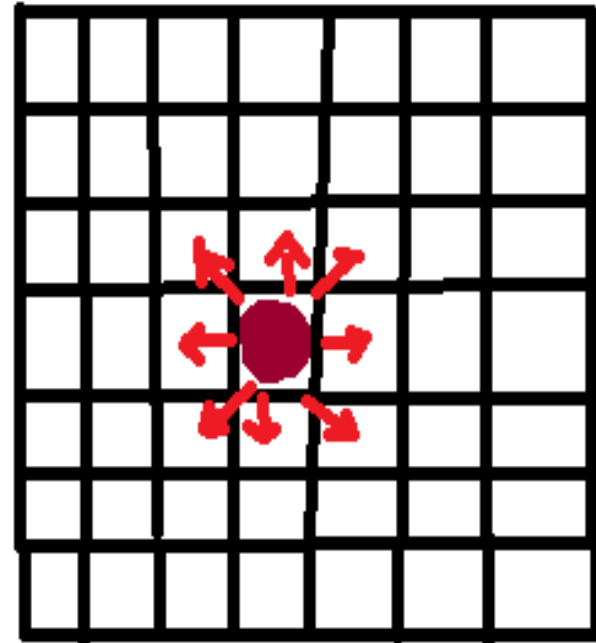
Parallele Implementierung einer
Populationssimulation
(von Nils Petersen)

Problemstellung

- Simulation einer Organismenpopulation
 - Verbreitung rezessiver Gene (Mendel Vererbung)
 - Mit und ohne erhöhte Fitness

Implementierung

- Simulation in 2D Gitter Welt
- Mehrere Organismen je (x,y) – Punkt
- Organismen sortiert in Listen
- Organismen finden Partner und bekommen Nachwuchs an ihrem Aufenthaltsort



Implementierung - Details

- Welt
 - Größe
 - Kapazität je Feld
 - Information über Populationsgröße in einem Punkt
- Organismus
 - Position
 - Alter
 - Gene / Allele inkl. Geschlecht
 - Gene / Allele des Partner für Nachwuchs
- ..viele weitere Parameter (z.B. Lebenserwartung, #Nachkommen)

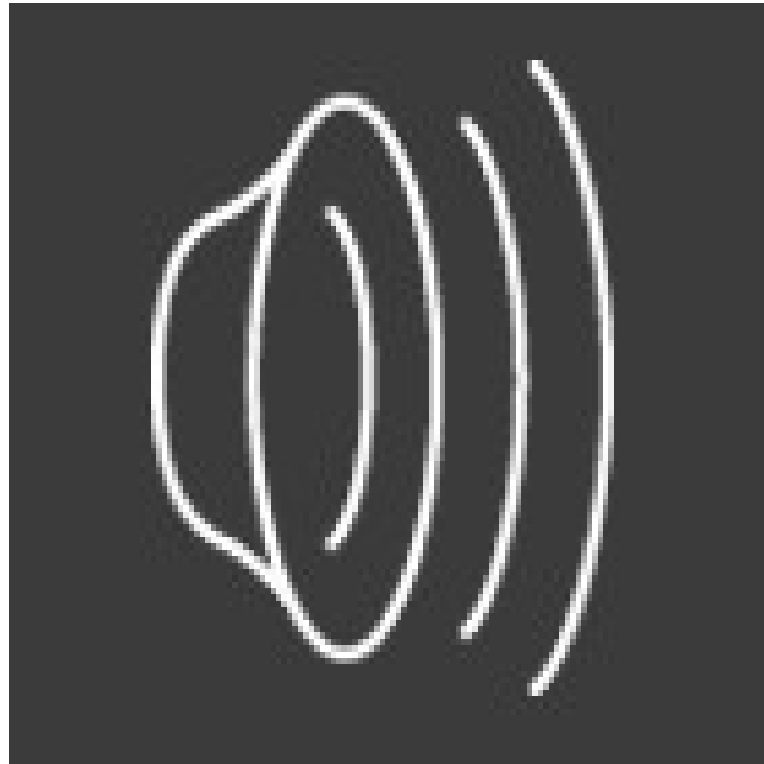
Iterativer Ablauf

- Überleben
 - Altersabhängiges sterben
 - Von Populationsdichte abhängig
 - Sigmoid-Funktionen
- Bewegung im Gitter
- Partnerwahl
- Geburt des Nachwuchs

Simulation und Visualisierung

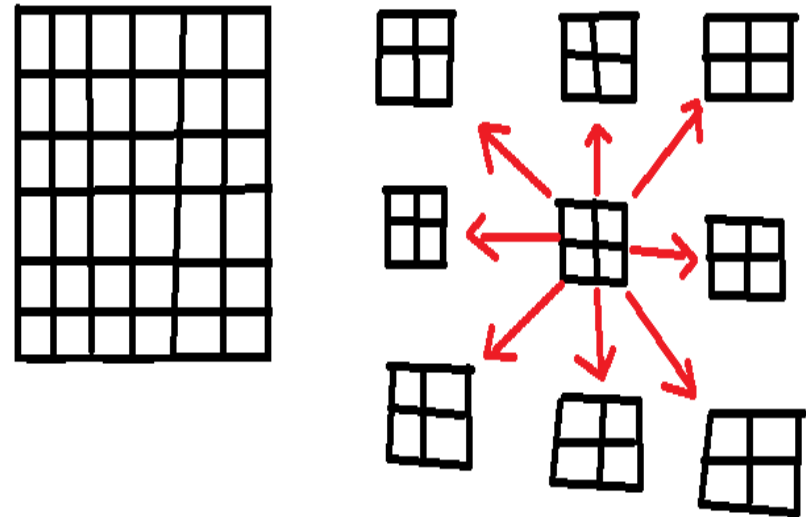
- Initiale Bevölkerung
 - Rezessive Gene mit geringerem Vorkommen
 - Neutral
 - Mit Fitnessvorteil
 - Höhere Lebenserwartung
 - Mehr Nachkommen
- Visualisierung

Film

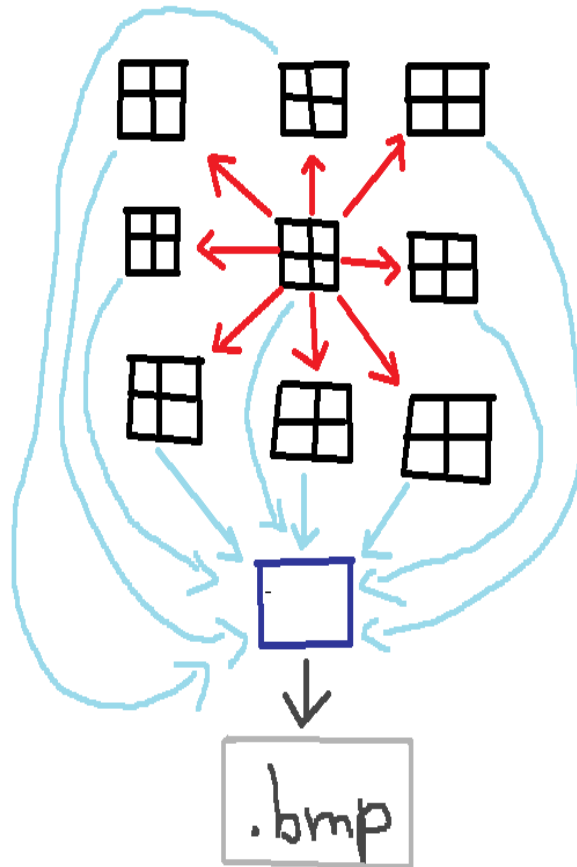


Parallelisierung

- Aufteilen der Karte auf $N*N$ Prozesse
 - Verschicken der Organismen an 8 Nachbarprozesse
 - Zunächst Anzahl
 - MPI_INT
 - Dann Organismen
 - MPI_Type_create_struct(..)
 - MPI_Isend, MPI_Irecv, MPI_Wait



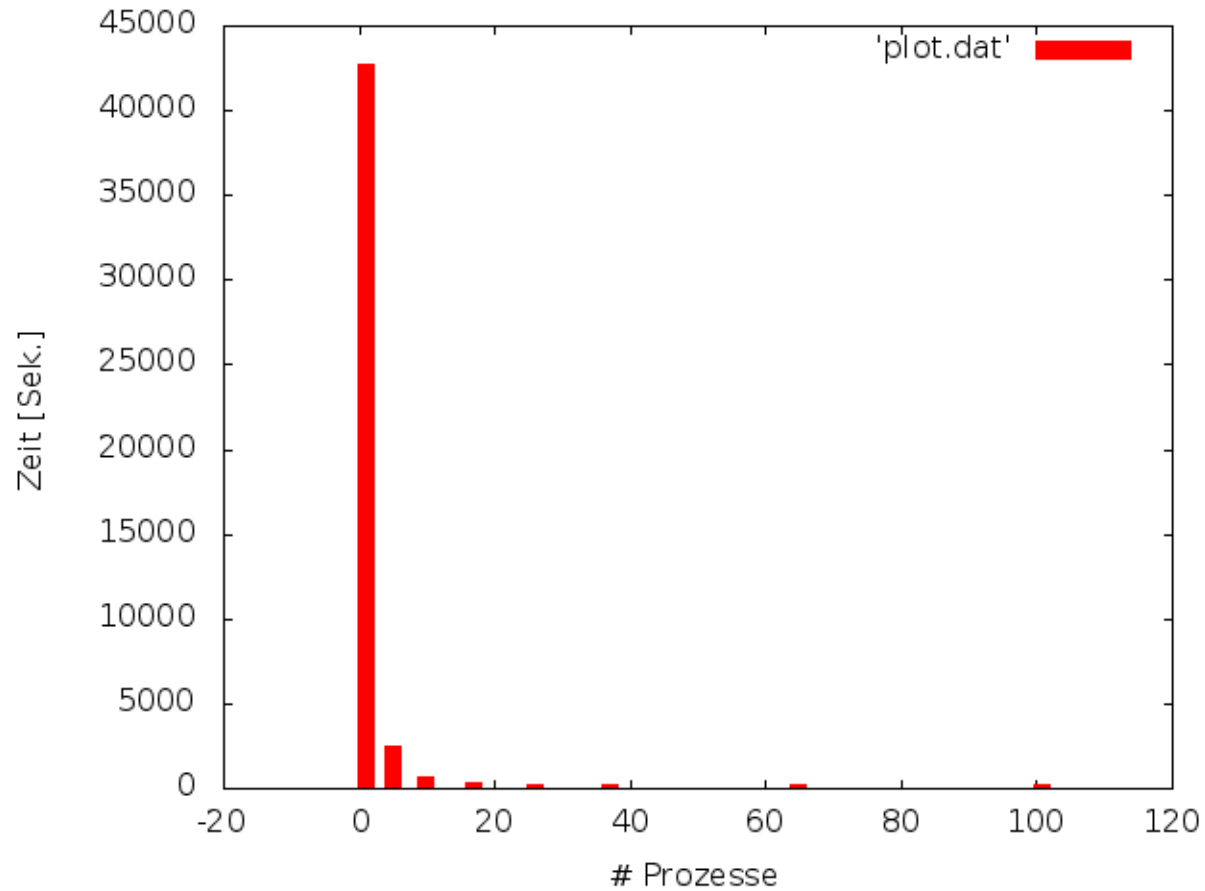
Visualisierung in einem Prozess



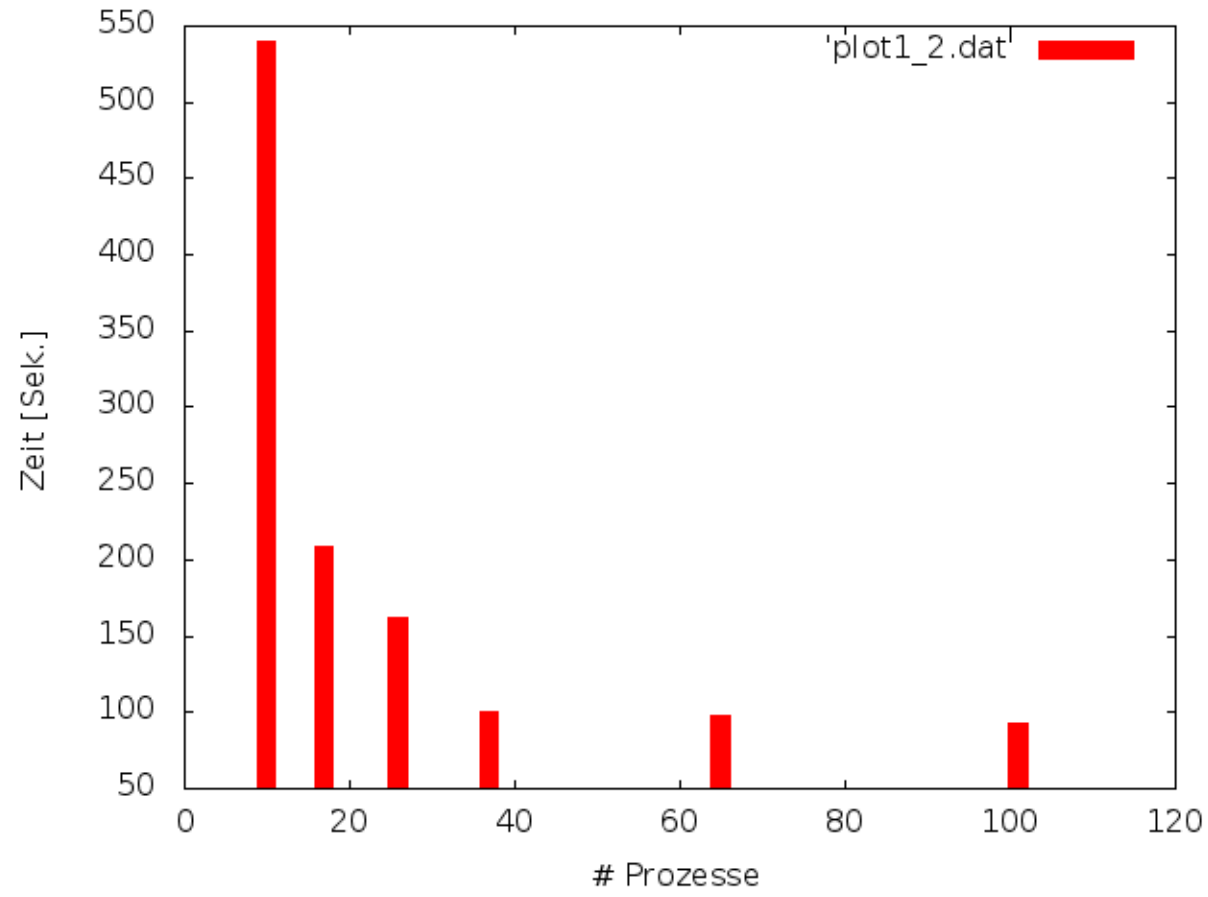
- + 1 Visualisierungsprozeß
- Sammeln der Populationsdaten (Populationsdichte und Anzahl der Phänotypen) und versenden an Visualisierungsprozess
 - In Struktur gespeichert (MPI_Type_create_struct())
 - MPI_Isend, MPI_Wait erst bei nächster Visualisierungsrunde
 - Visualisierungsprozess: MPI_Recv aller Daten, dann .bmp Datei erstellen

Laufzeiten

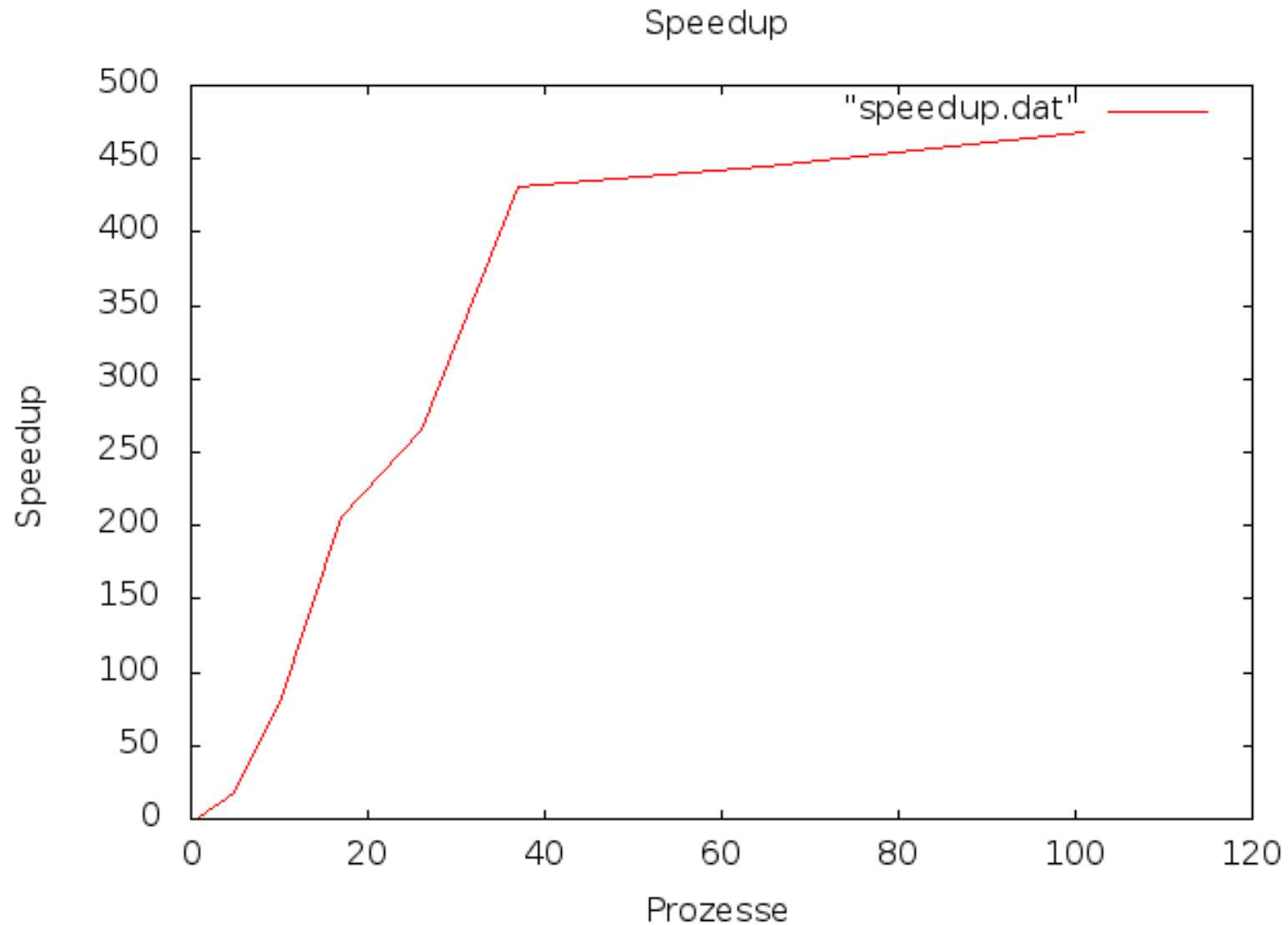
(Weltgröße 240 x 240 Zellen)



■ ■ ■

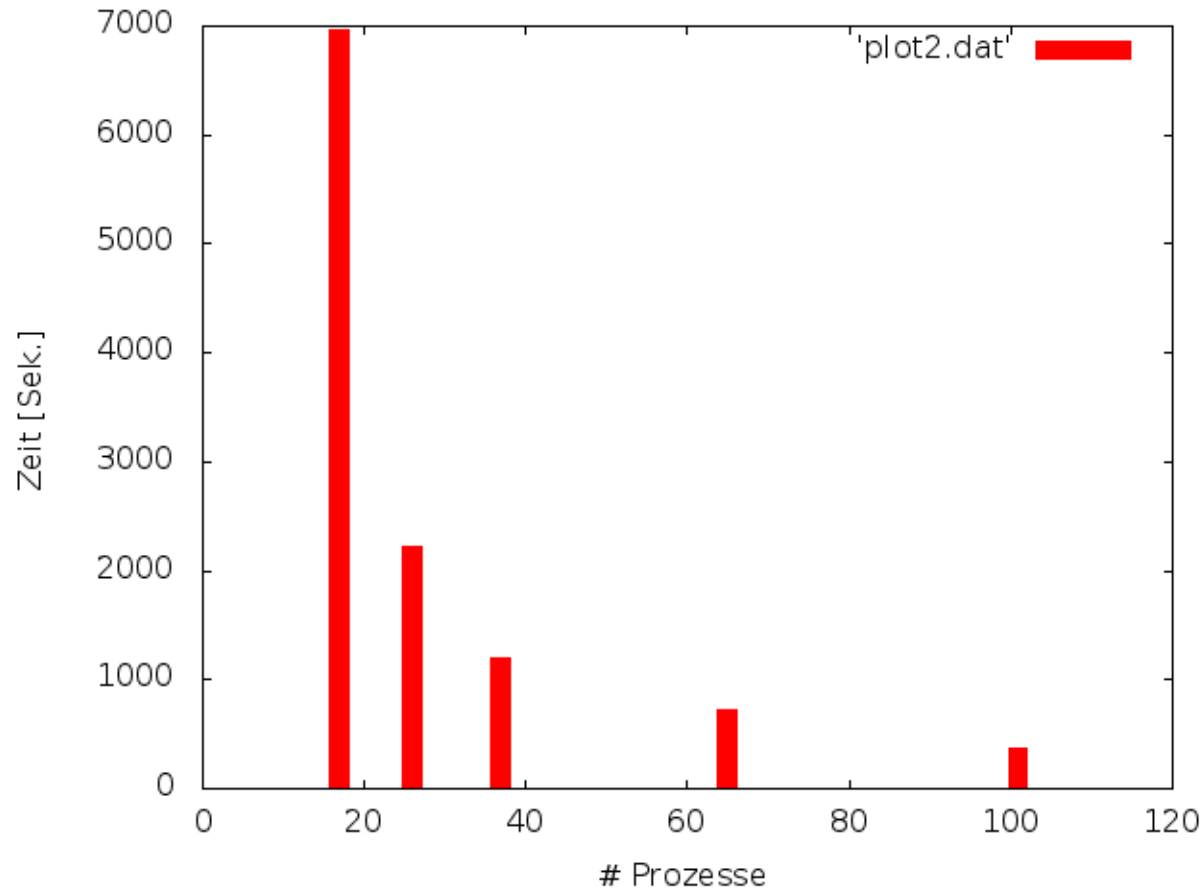


Speedup

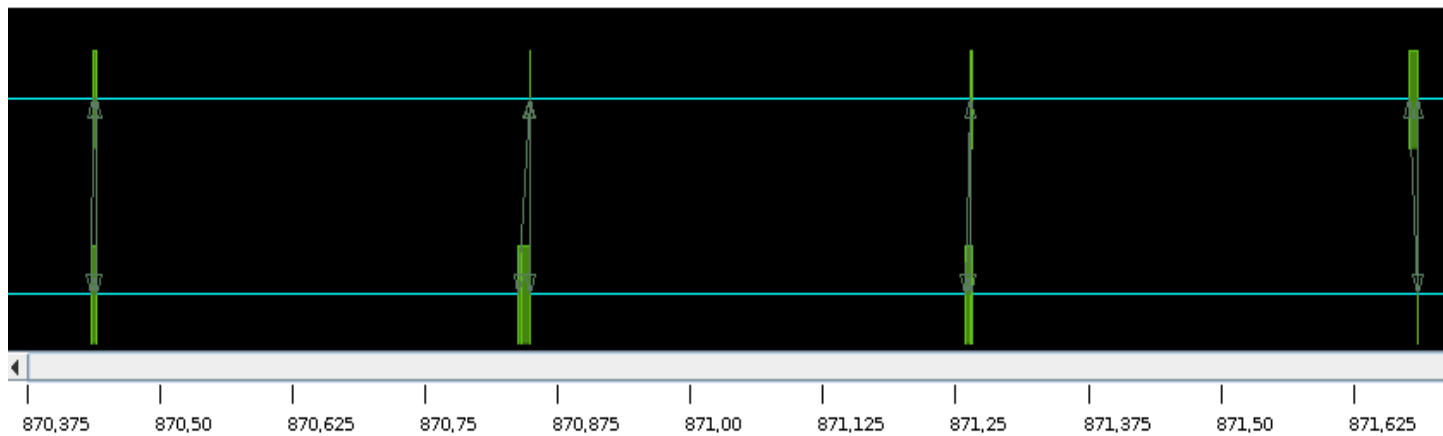
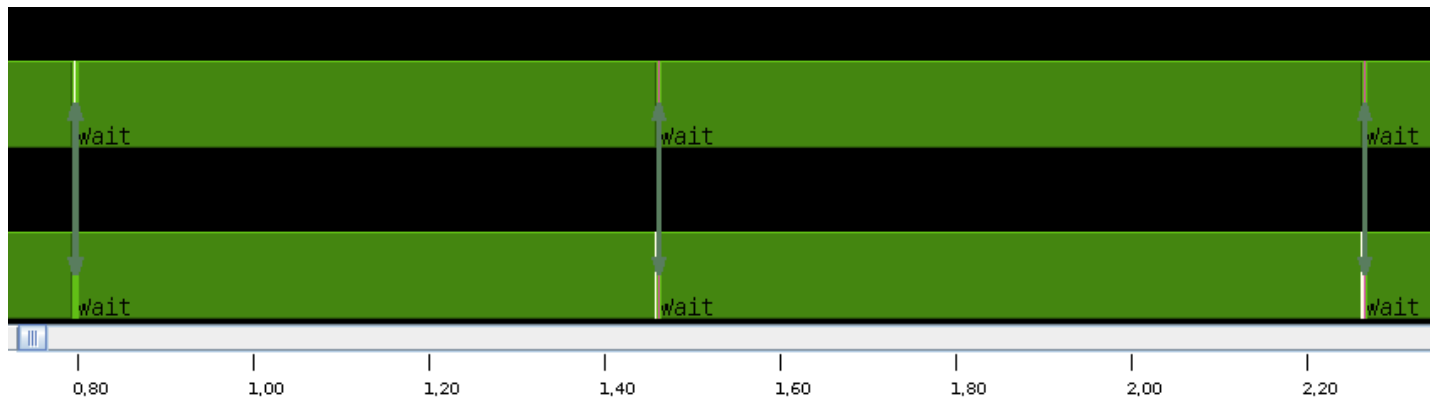


Laufzeiten größerer Welten

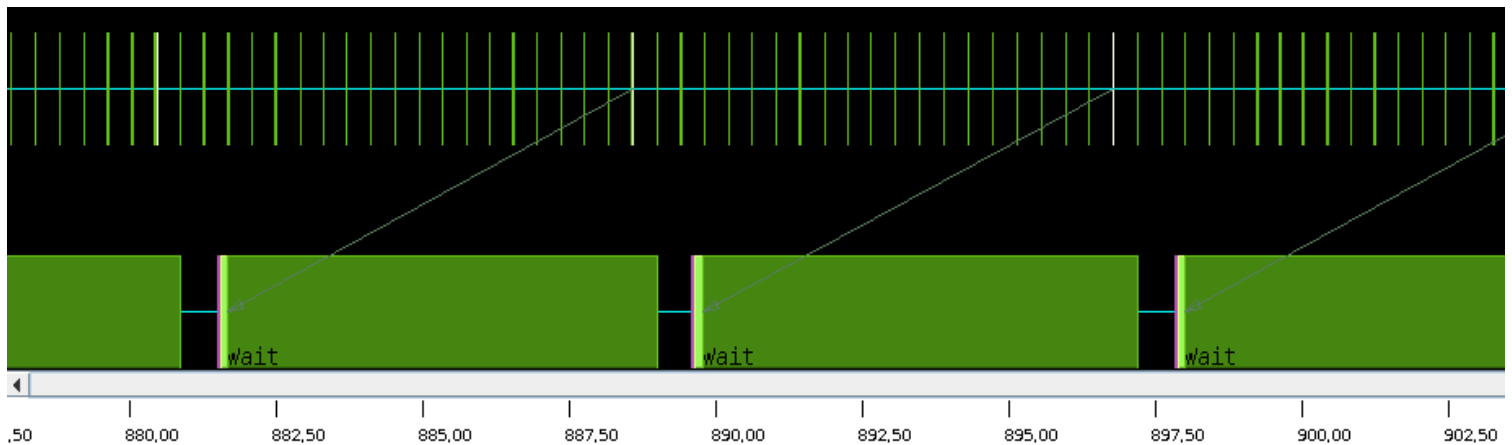
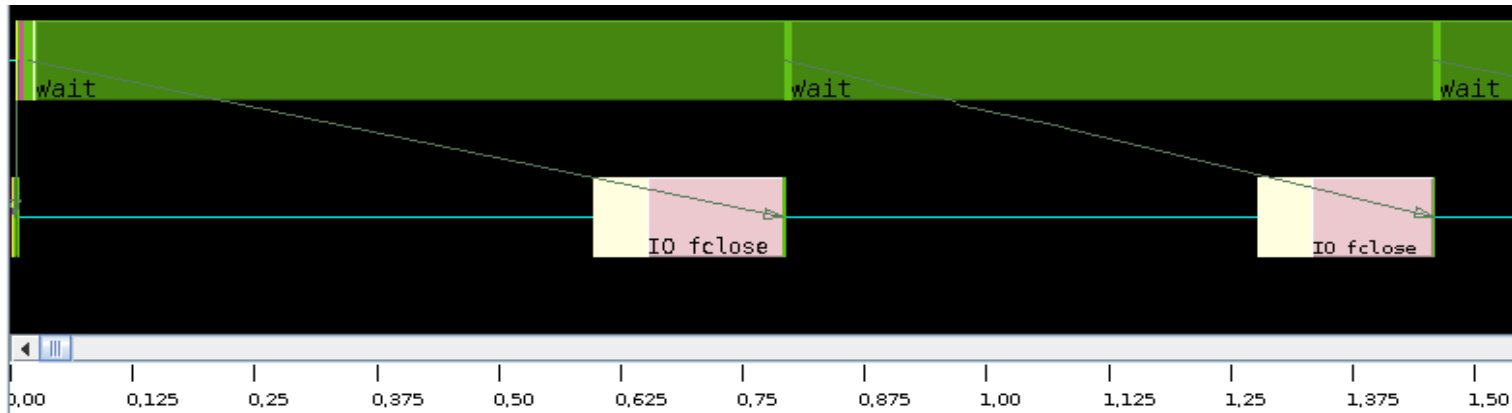
(Weltgröße 480 x 480 Zellen)



Lastausgleich (2 Simulationsknoten)



Lastausgleich (Simulations- und Visualisierungsknoten)



Zusammenfassung

- Parallelisierung der Populationssimulation in einer Gitterwelt
 - Gleichmäßige Aufteilung der Karte
 - Effizienz der Parallelisierung abhängig von der Kartengröße
- Verbesserungsmöglichkeiten
 - Effizientere Funktionen (z.B. Partnersuche)
 - Organismenzahl nicht vorher senden
 - Anpassen der Prozesszahl an Weltgröße
 - Nicht-quadratische Aufteilung der Welt
 - Mehr als ein Visualisierungsprozess

Vielen Dank für eure Aufmerksamkeit!!!