



Smoothlife - an adaptation of the game of life with continuous space

B. Seelbinder, R. Gerst

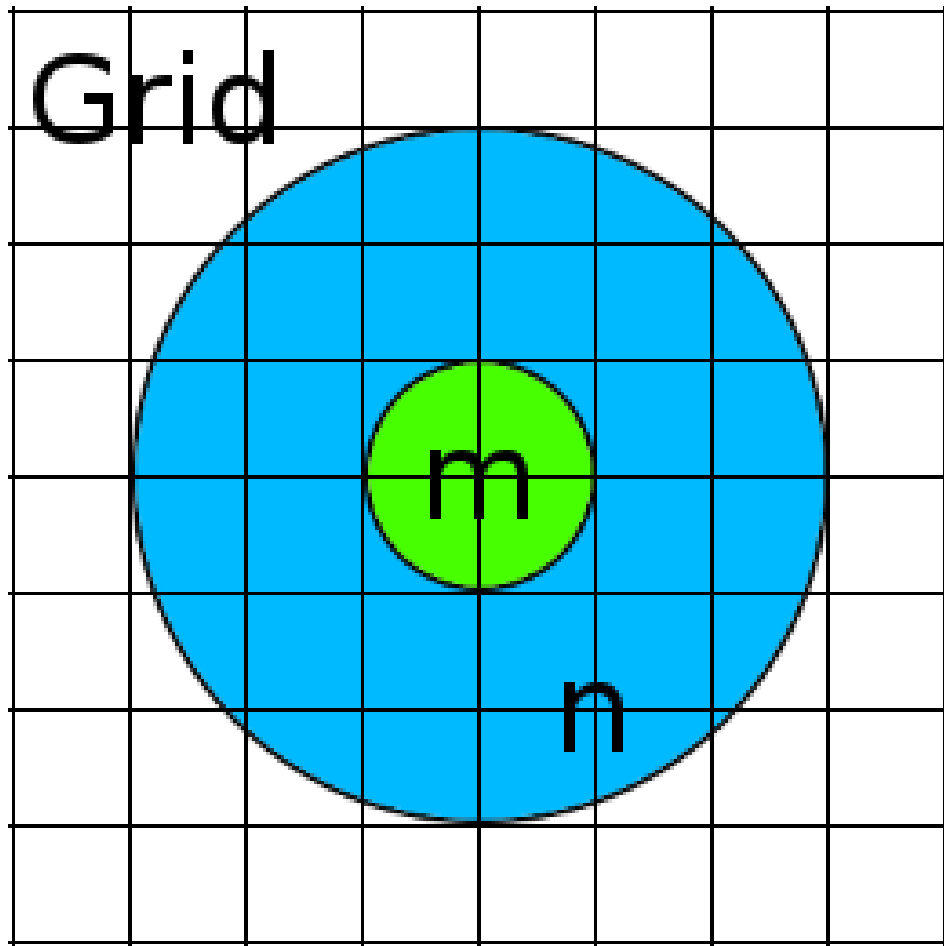
Paper: S. Rafler, Generalization of Conway's "Game of Life" to a continuous domain – SmoothLife, 2011

The general idea

- **We have:**

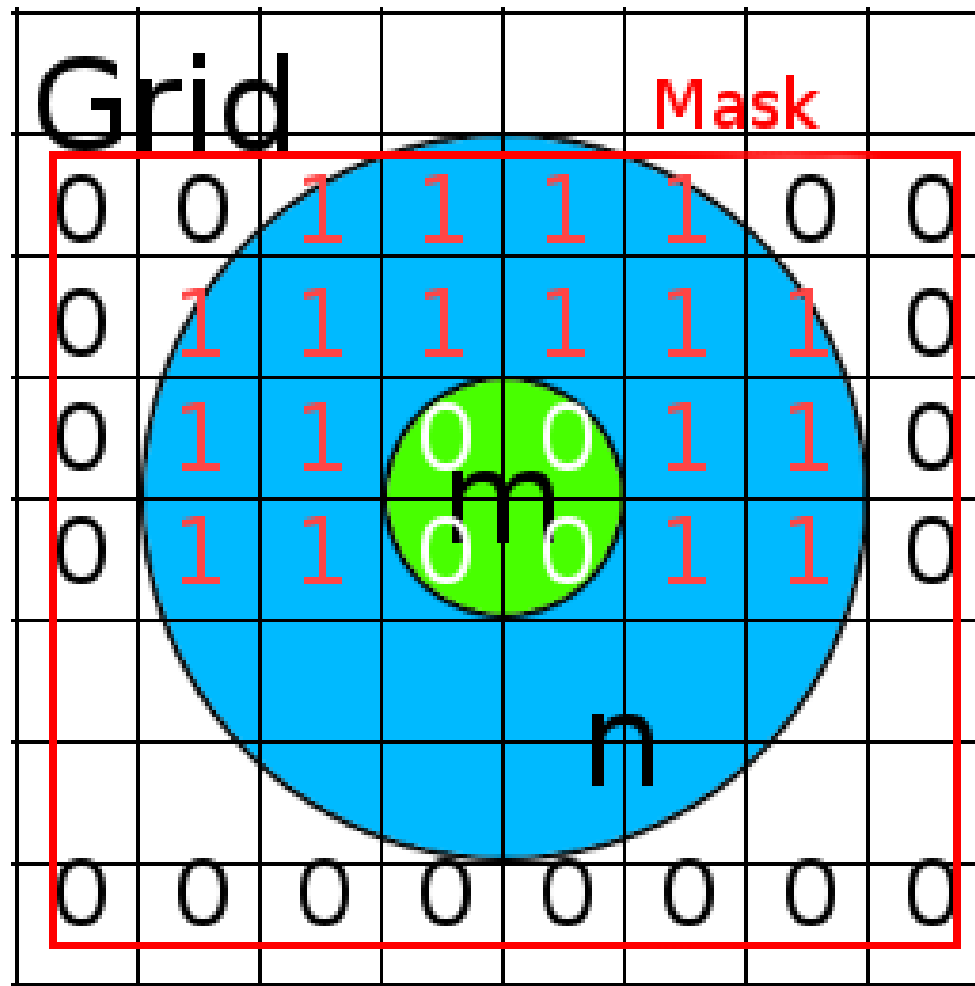
- A continuous space (as discrete grid)
- Border wrapping (infinite space)
- Cells (outer + inner filling)
- Infinitesimal small time steps (euler integration)

visualized



- $\text{grid}[x,y] \in [0,1]$
- M: Inner filling $\in [0,1]$
- N: Outer filling $\in [0,1]$

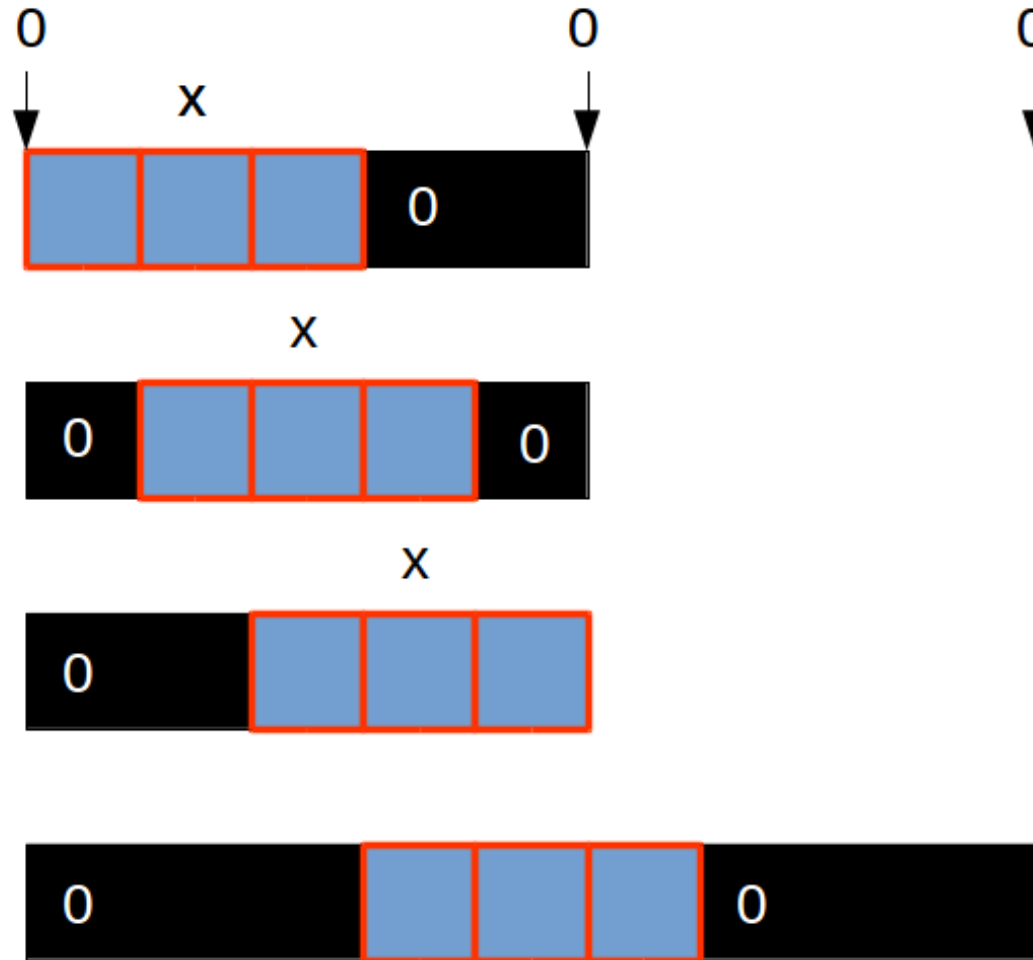
Calc filling using masks



Vectorization

- Filling can be vectorized
- Filling is calculated using masks
- **Problem:**
 - We can only align the mask OR the grid to memory boundaries
 - Vectorization over borders is difficult (but possible)
- **Solution:**
 - Offset-masks

Offset mask impression



Offset masks

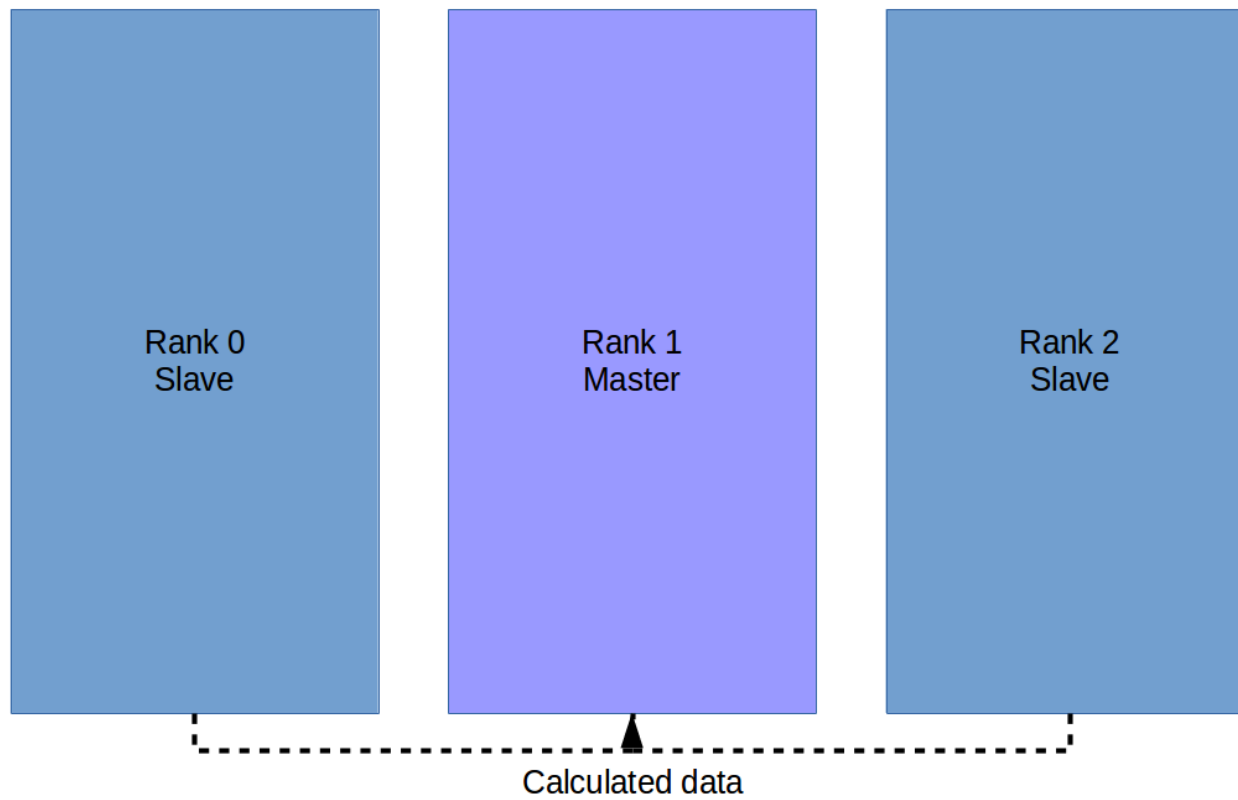
- Number depends on cache-line size **ONLY**
- **# = CACHELINE/DATA_TYPE** many (here: 16)
- **Pro: allows complete vectorization**
- **Con: increases number of calculations**

- **Speedup: 3.44 - 4 (thread dependent)**

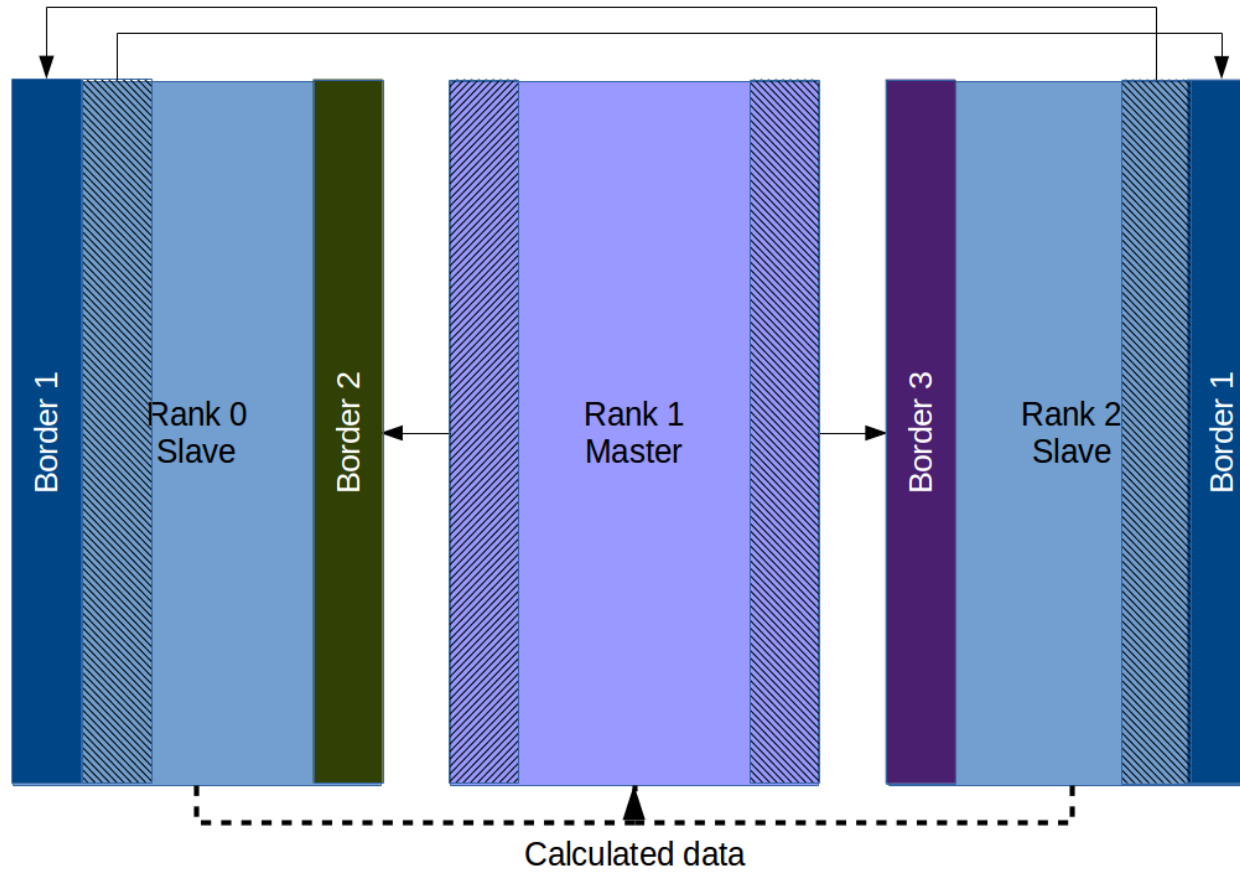
Parallelization & minor cache optimization

- Using **OpenMP**
- Running over all **x & y** coordinates of the grid
 - **#pragma omp parallel for schedule(static)**
for (int x = x_start; x < x_start + w; ++x)
- Offset masks depend on **x** coordinate only
 - › **Off := (x - x_center) % CACHELINE_FLOATS**
 - › Only **1** mask for all **y** coordinates needed per core
- **Speedup: 3-5 (load dependent)**

MPI - Master and Slaves

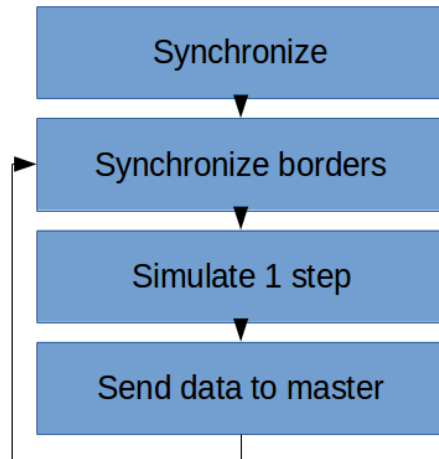


MPI - Borders

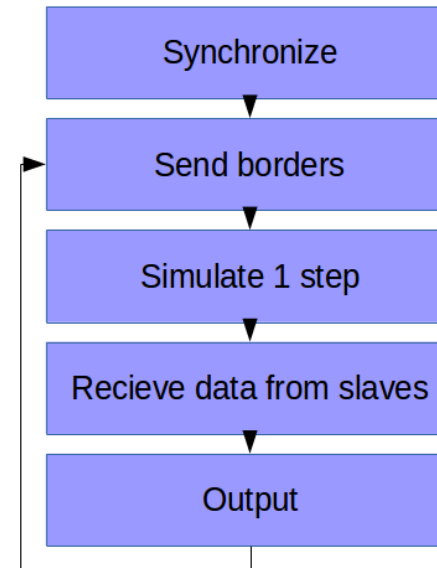


MPI - Pipeline

Slave simulator MPI



Master simulator MPI



MPI - Speedup

MPI Speedup 2xhost: 1.2 - 1.4

MPI 4x MIC: 3.3

MPI Speedup All: 1.15

total speedup: 34

• 1.39 calc/sec → 47.8 calc/sec

Test: 720x720, r=10 & r=21

DEMO

