Parallelization of Grid-oriented problems

Yaidel Reyes López

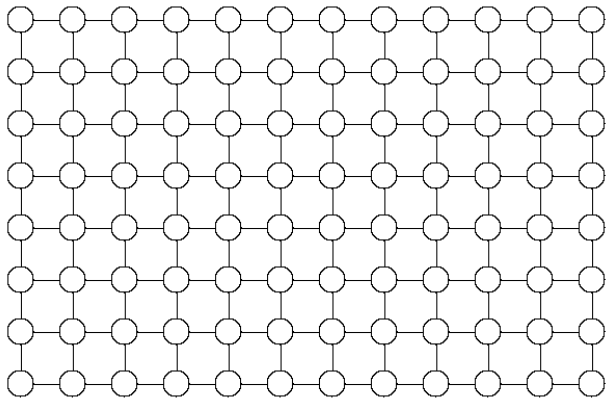
KULeuven

November 6, 2013

Adapted from slides by Prof. Dirk Roose

See also: BSP book, Section 4.8,p 210

Grid-oriented problems



- data set defined on a grid, local computations with small 'stencils', e.g. PDEs (finite differences), image processing, ...
- grid point: generic name for data associated with grid point, pixel, cell, fine element, ...
- grid, data set and associated work: partitioned in subdomains. The subdomains are assigned (mapped) to processors

Grid-oriented problems

Extra tasks (compared to sequential problem):

- partitioning and mapping to ensure work load balance and communication minimisation
- communication between neighbouring subdomains

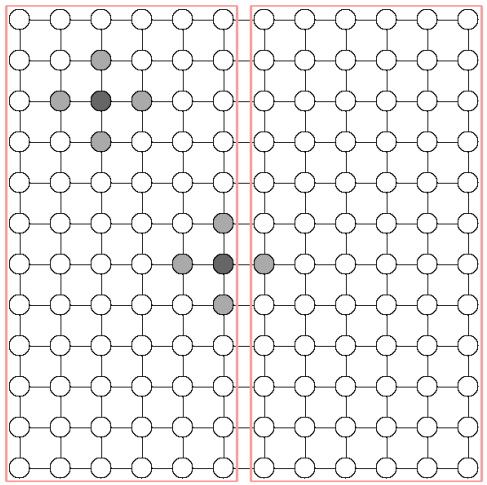
Model problems

- PDEs:
  - explicit time integration
  - relaxation methods (Jacobi, Gauss-Seidel, SOR, ...) on a structured (regular) 2D grid
- cellular automata:
  - game of life
- image processing:
  - convolution on a 2D pixel matrix

same data-dependency pattern
same parallelisation strategy

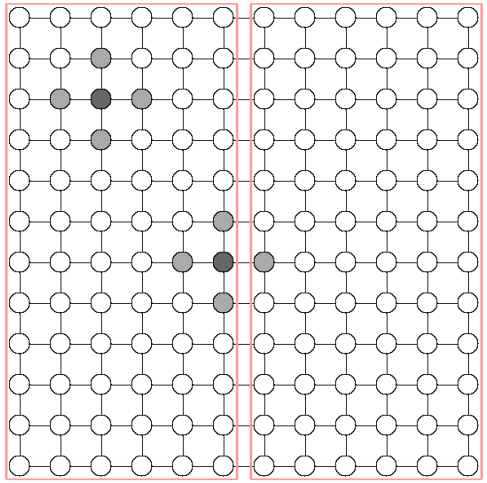
Computational molecules

5 point stencil

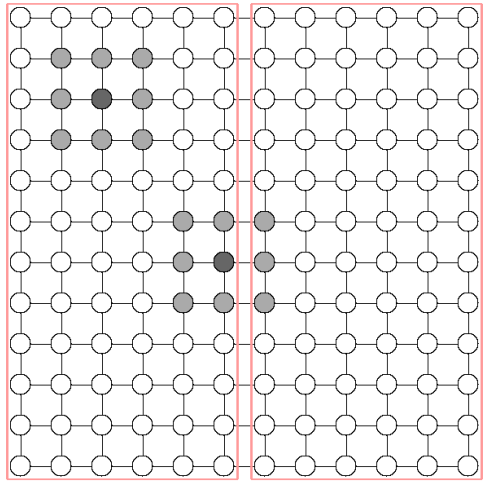


Computational molecules

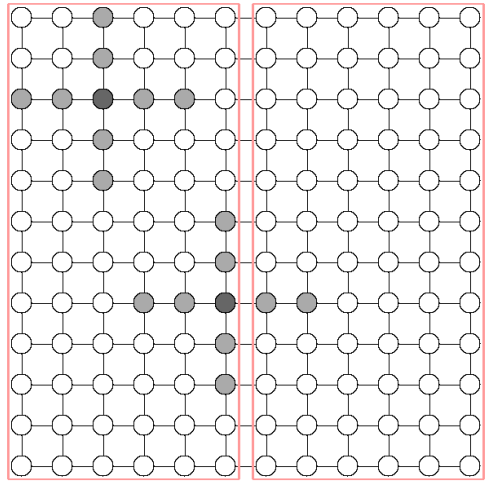
5 point stencil



9 point stencil

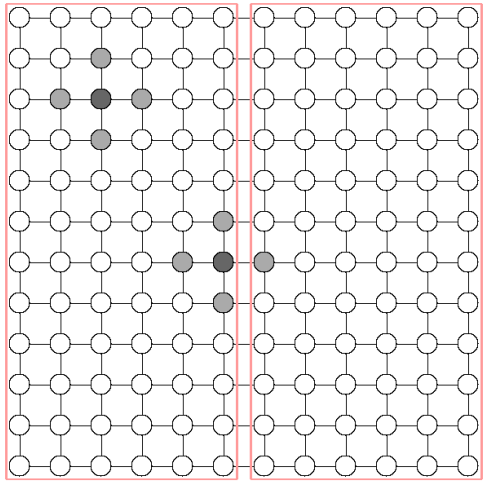


9 point stencil

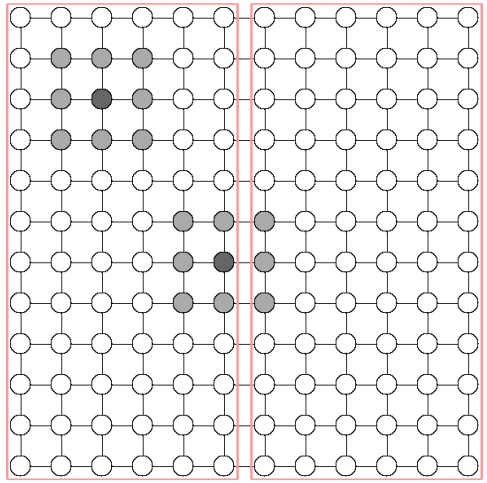


Computational molecules

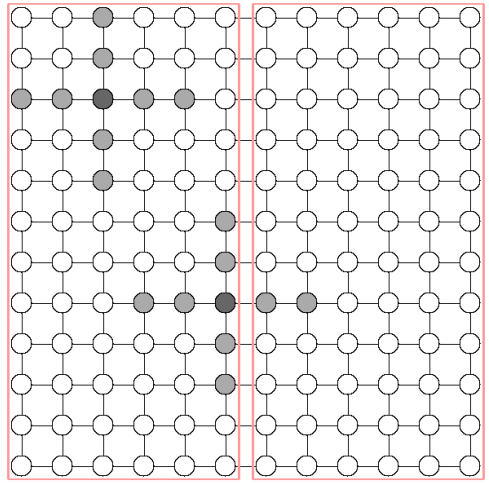
5 point stencil



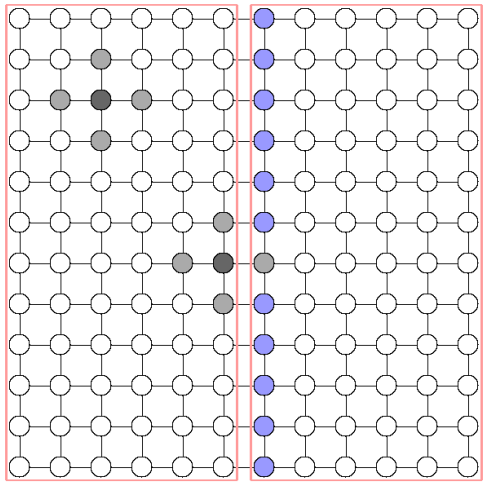
9 point stencil

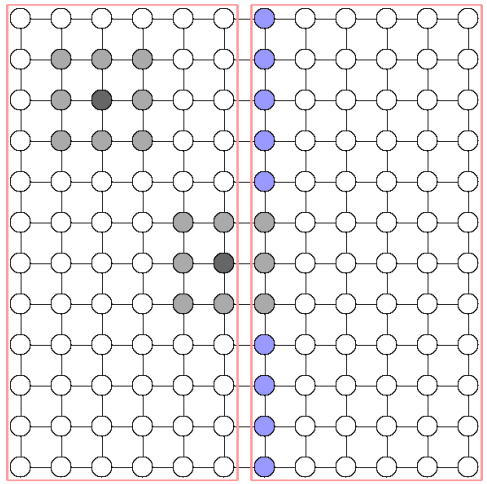


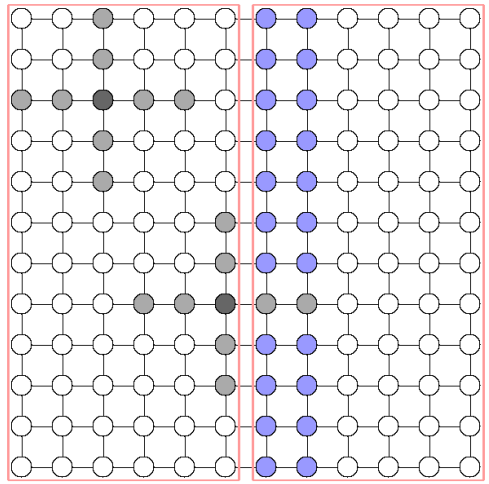
9 point stencil



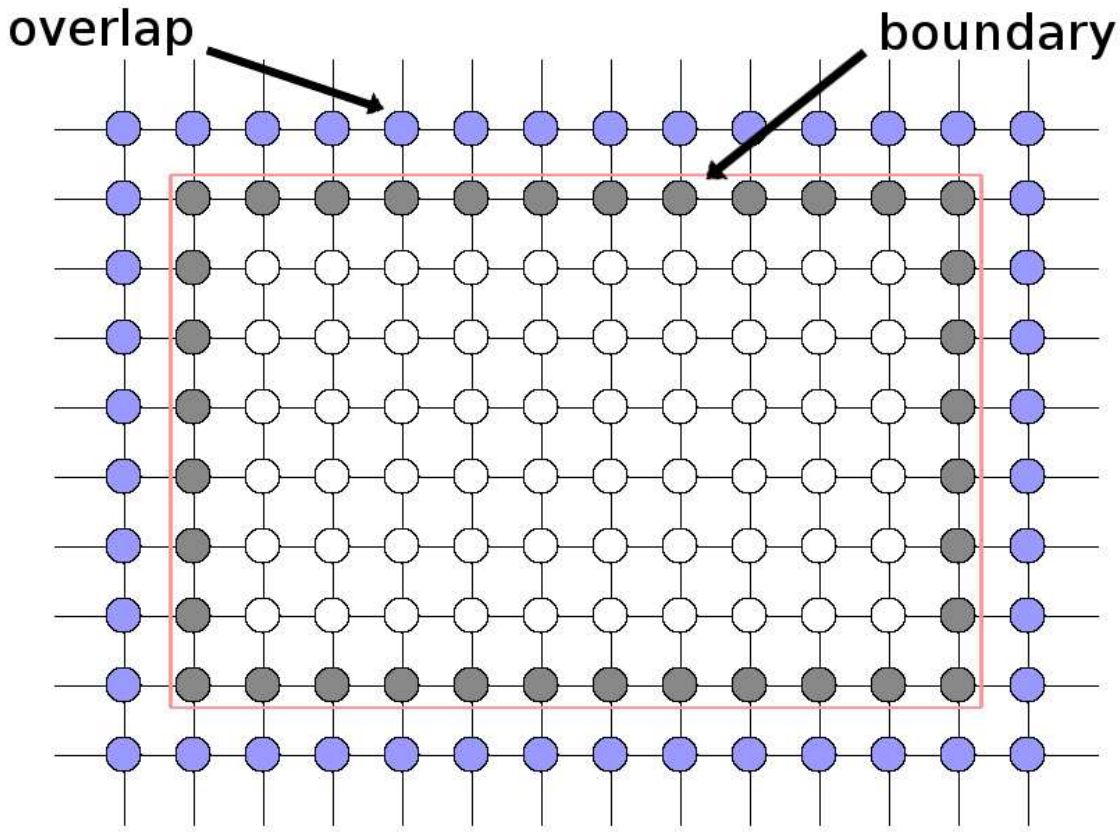
Overlap







Subdomains and overlap regions



Note: overlap region can have a width > 1.

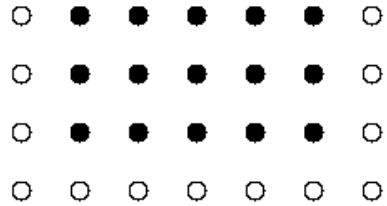
Skeleton of a typical program

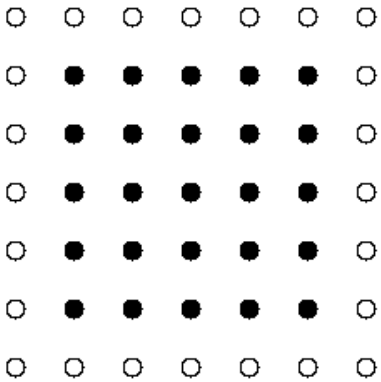
In every subdomain (processor):

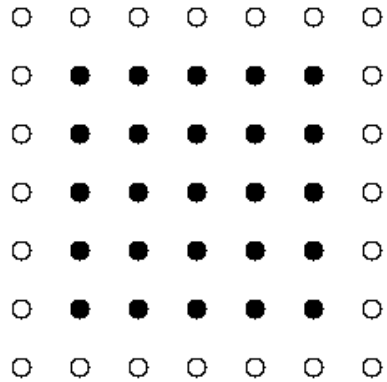
- exchange data in the overlap region
  communication with procs. holding neighbouring subdomains
- do calculations for all grid points in subdomain
- check for stopping criterion (e.g. convergence check)
  global communication (reduction)

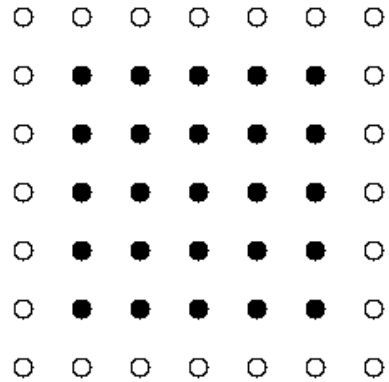
Exchange overlap regions

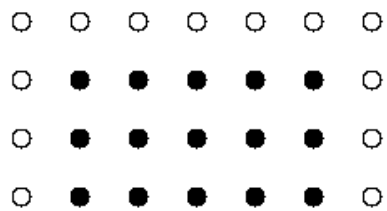
5 point stencil





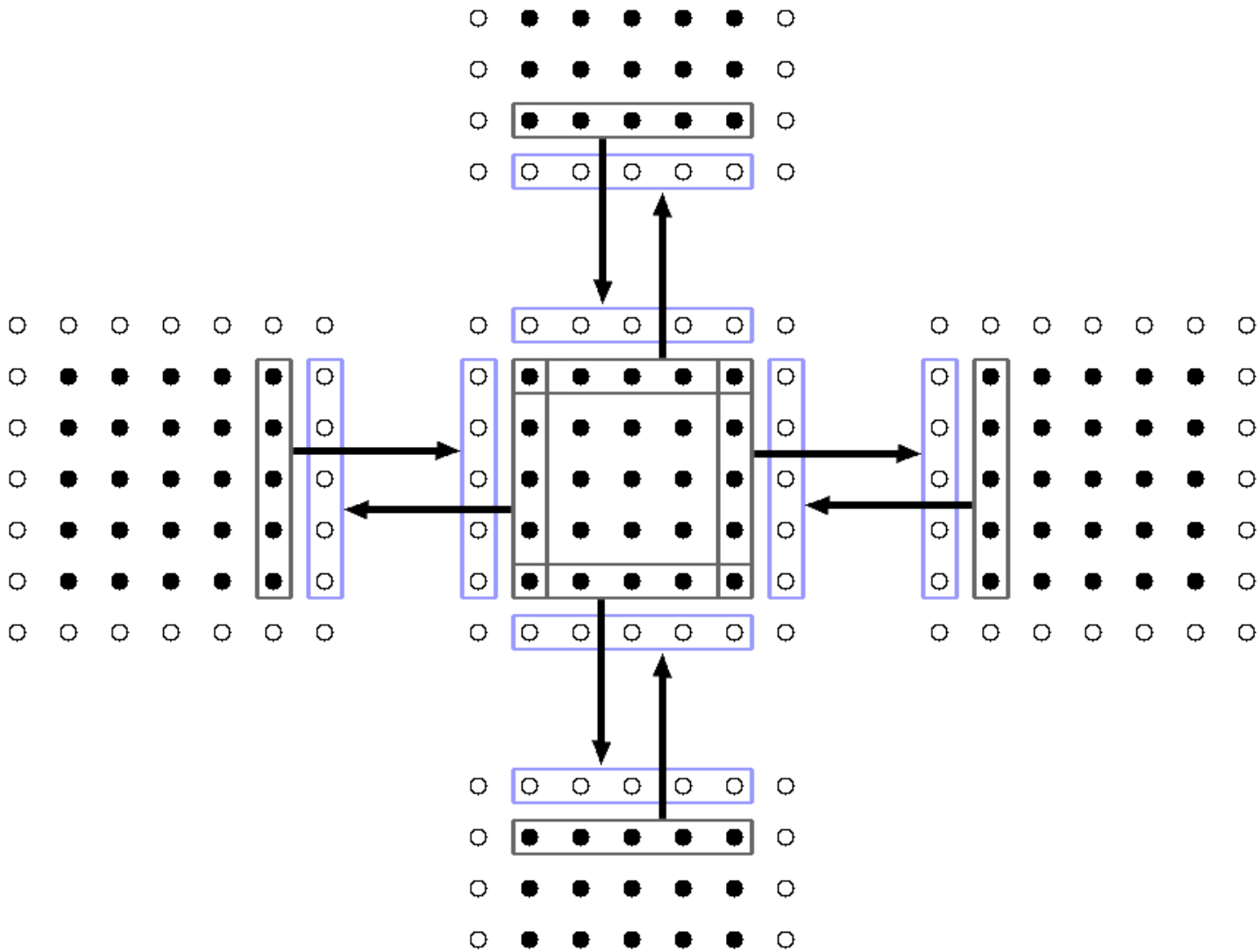






Exchange overlap regions

5 point stencil



Analysis of communication cost

- assume: p processors, M grid points
- no load imbalance: M/p grid points per processor

BSP const (assumes one *flop* per grid point):

$$(M/p)/r + bg + 2l$$

- communication volume $\sim$ perimeter of subdomain
  - b depends on the length of the boundary
- computation volume $\sim$ area of subdomain

Partitioning strategies

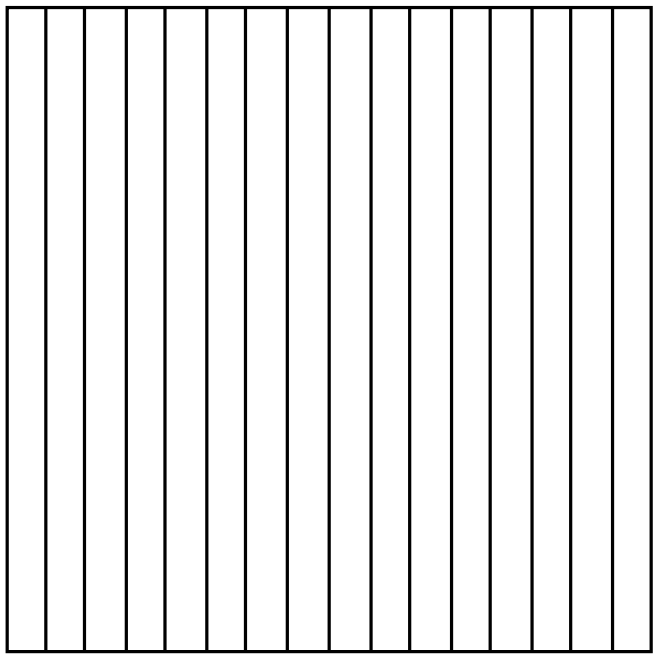
Desired partitioning: minimal perimeter for same area

Partitioning strategies

Desired partitioning: minimal perimeter for same area

Which of these is preferred?

strips



squares