Image Processing

A case study for a domain decomposed MPI code

epcc

THE UNIVERSITY OF EDINBURGH

Reusing this material

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_US

This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

Acknowledge EPCC as follows: "© EPCC, The University of Edinburgh, www.epcc.ed.ac.uk"

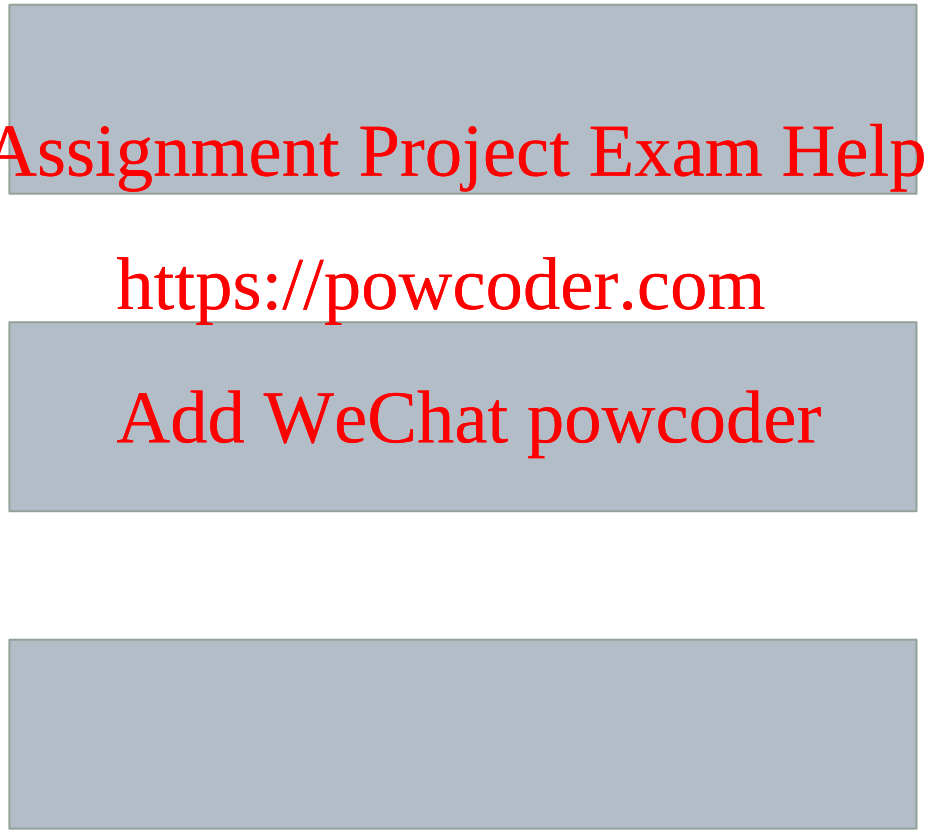
Note that this presentation contains images owned by others. Please seek their permission before reusing these images.

Domain Decomposition 1

- Starting with a big array:

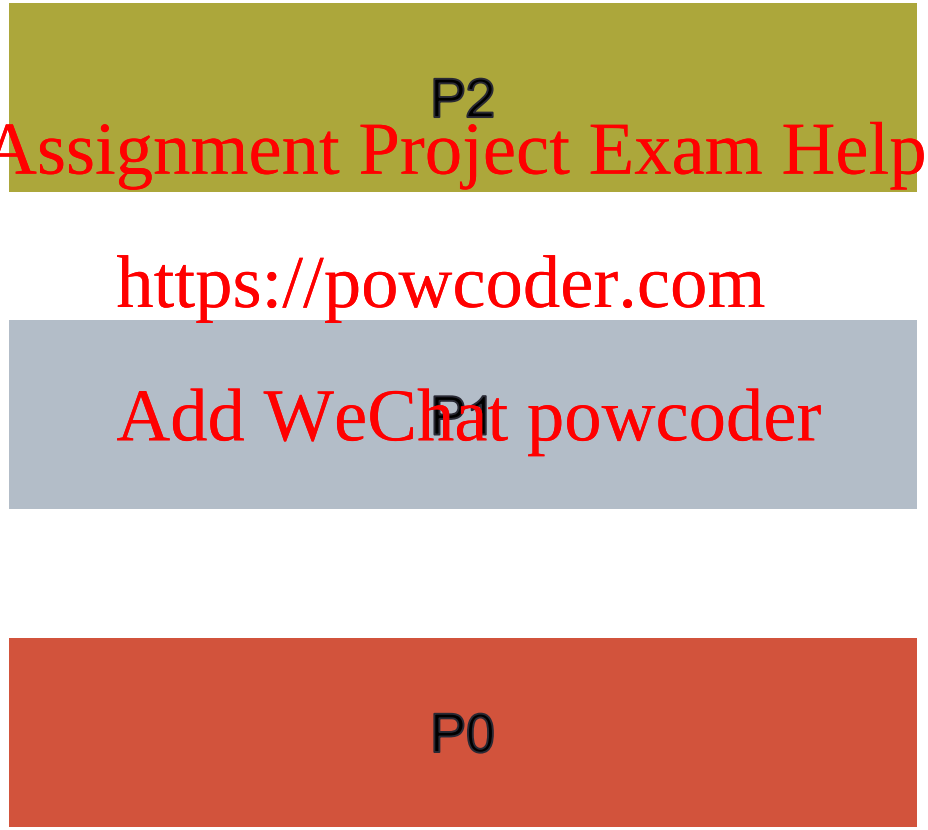
Domain Decomposition 2

- Split it into pieces:



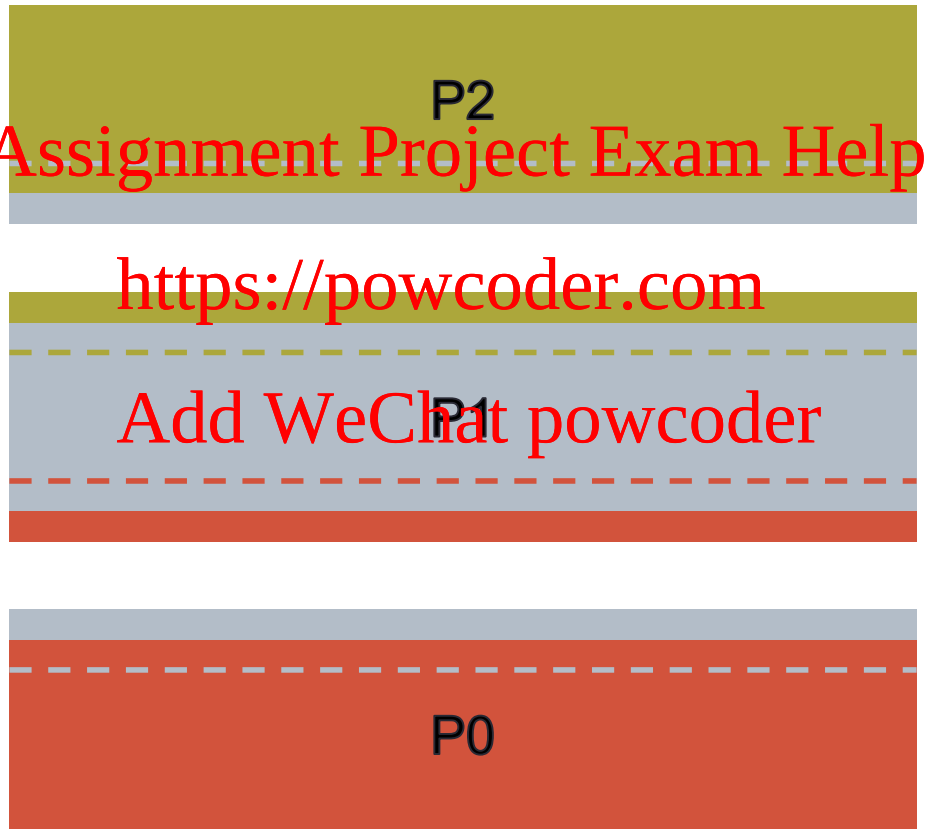
Domain Decomposition 3

- Assign pieces to processors:



Domain Decomposition 4

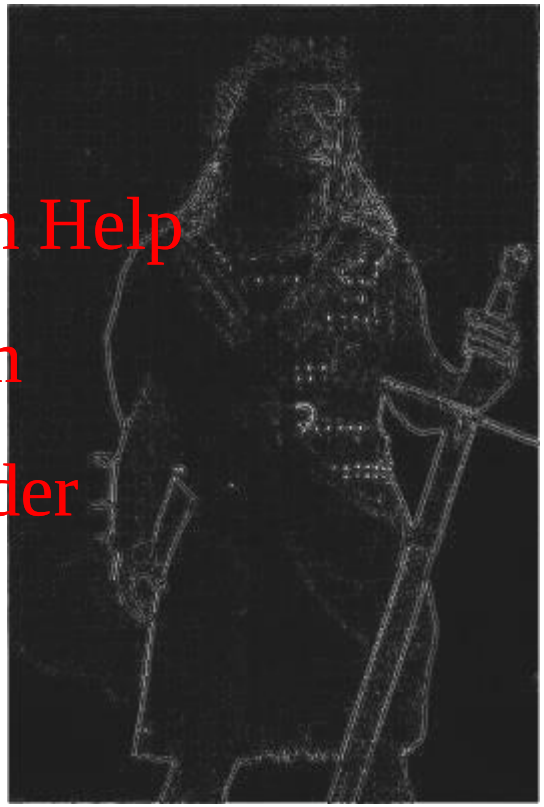
- Use Halos to deal with interactions



Edge detection / image reconstruction

single pass

hundreds of iterations



Edge detection

- Compare pixel to its four nearest neighbours
  - pixel values are from 0 (black) to 255 (white)

$$edge_{i,j} = image_{i+1,j} + image_{1,j+1} + image_{i-1,j} + image_{i,j-1} - 4\,image_{i,j}$$

- Pad 2D arrays with halos
  - in serial code, halo values set to white (i.e. 255)

Image reconstruction

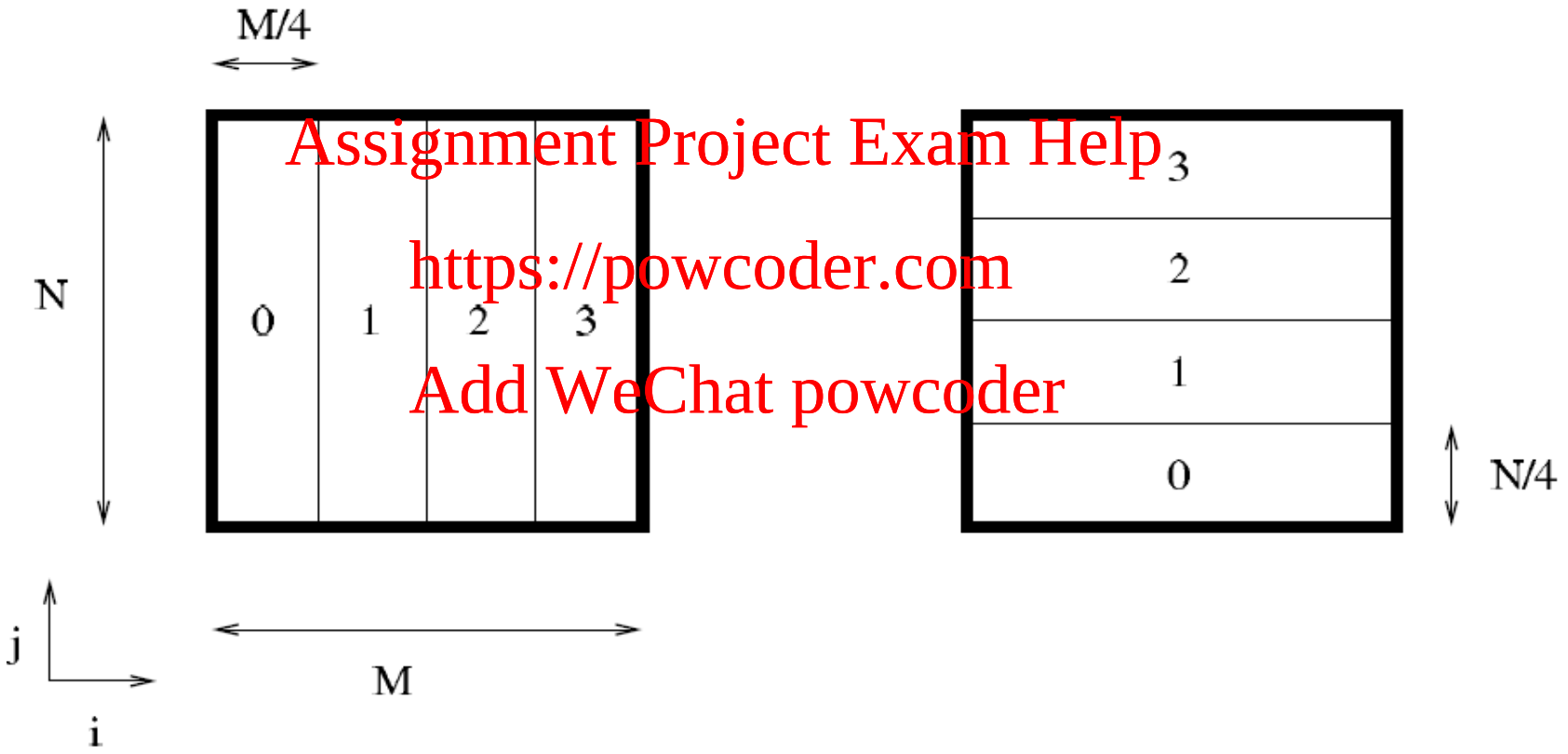
- Jacobi Solver to undo the simple edge detection algorithm (a five-point stencil)
  - simple example of discretised partial differential equation with nearest-neighbour interactions
  - actually solving $\nabla^2 image = edge$

$$new_{i,j} = \frac{1}{4}\left(old_{i+1,j} + old_{i,j+1} + old_{i-1,j} + old_{i,j-1} - edge_{i,j}\right)$$

- Repeat many times
  - in parallel, must update halo values from neighbours every iteration

Domain Decomposition

- Different choices in C and Fortran



The case study

- I provide you with:
  - More detailed printed instruction
  - Tar-ball (Choice of C or Fortran)
    - Input routine
    - Output routine
    - Couple of input files
- Tasks
  - Write a serial code (with halos for fixed boundary conditions)
    - ***check that the serial code works!!***
  - Distribute the work onto the processors; separate reconstructions
  - Get the halos exchanged; single reconstruction, identical to serial
  - Further suggestions on the instruction sheet