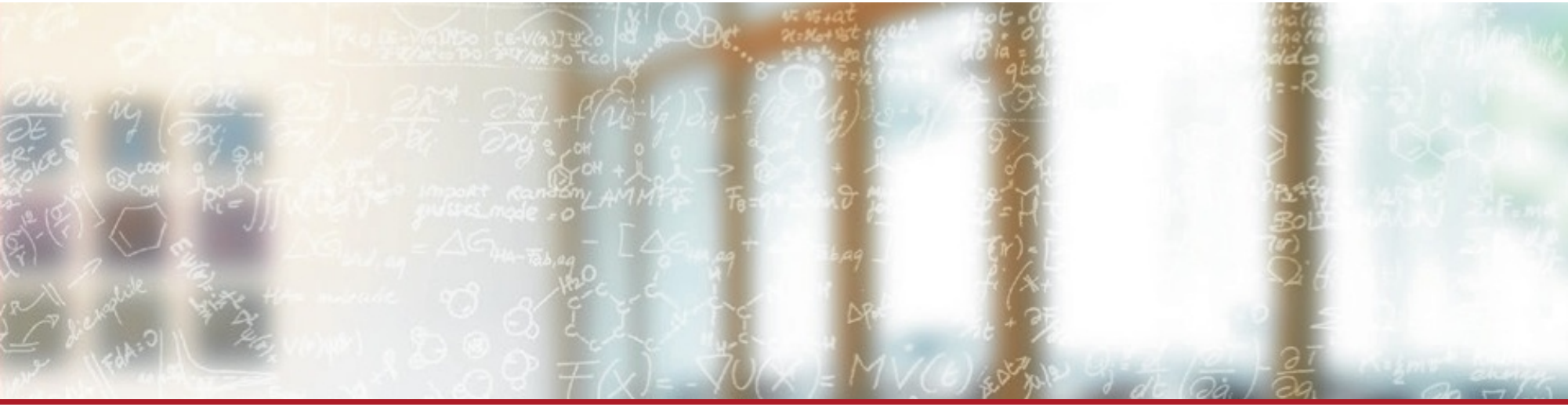
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Summer School 2018 – High Performance and Parallel GPU Computing

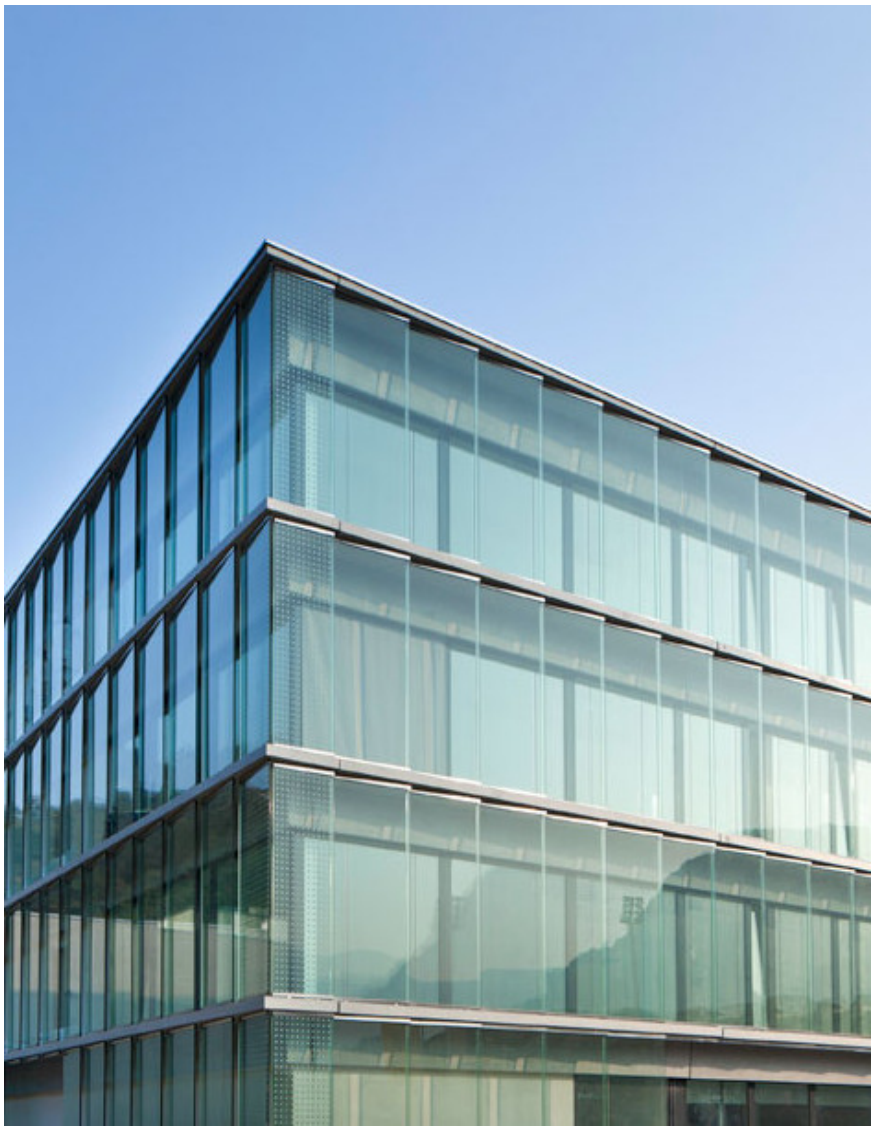
Maxime Martinasso, CSCS

July 24, 2018

Course Objectives

- Make the MiniApp runs on multiple nodes
- Add MPI function calls in the MiniApp code

MiniApp changes



- Initialize and finalize MPI
- Create a Cartesian topology
- Change linear algebra functions
- Exchange ghost cells
- Summary

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

MiniApp changes review

Initialize and finalize MPI

Setup the code to be MPI aware.

- File to edit: `main.cu`
- Initialize MPI and get the current rank and the number of ranks
- Finalize MPI

Create a Cartesian topology

Minapp uses a 2D grid, each rank will work on a sub part of the grid.
Make a 2D domain decomposition of the grid depending on the number of ranks.

- File to edit: `data.cu`
- Create the dimension of the decomposition depending on the number of ranks
- Create a non-periodic Cartesian topology for the grid of domains
- Identify coordinates of the current rank in the domain grid
- Identify neighbours of the current rank: east, west, north and south directions

Change linear algebra functions

Make the dot product and the computation of the norm over all ranks.

- File to edit: `linalg.cu`
- Add a collective operation to compute the dot product
- Add a collective operation to compute the norm

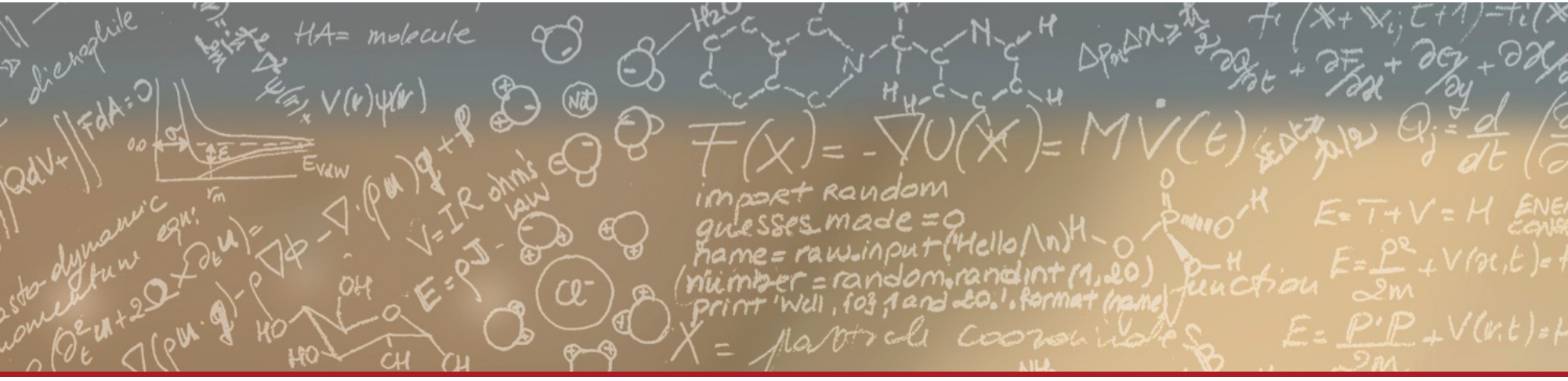
Exchange ghost cells

Use point to point communication to exchange ghost cells among neighbours. Use RDMA.

- File to edit: `operators.cu`
- Add point-to-point communication for all neighbours in all directions
- Use Non-blocking communication
- Try to overlap computation and communication
- Optional - try MPI 1-sided

Summary

- Initialize MPI: `main.cu`
- Domain decomposition: `data.cu`
- Parallel linear algebra: `linalg.cu`
- Exchange ghost cells: `operators.cu`



Thank you for your attention.