

## Exercise 1: MPI-Parallelization of a Jacobi Solver

Group No. 12

Aman Bhardwaj (12333472)

Florian Frech (12308544)

### Task 1: Questions

#### 1. Describe the advantages / disadvantages of a two-dimensional decomposition

(compared to a one dimensional decomposition)

Advantages of a 2D decomposition:

- It is easier to handle complex geometries which are irregular or intricate in shape as it decomposes the domain into smaller subdomains that can be handled and distributed evenly across individual processes.
- It provides clear representation of discontinuous properties and interfaces, thus making it easier to apply various numerical methods and boundary conditions leading to more accurate simulations.
- In a 2-D decomposition, each region can independently control and generate its own mesh, that provides more efficient mesh generation.
- The neighbouring grid points close to each other are assigned the same MPI process which helps in better cache utilization and more data locality.
- Conservative methods of physical quantities can be employed because they can be applied to individual regions independently.

Disadvantages of a 2D decomposition:

- Due to less connectivity of information between the neighbouring processes, it can become more challenging to implement and handle communication between neighbouring processes, which in turn lead to more complexity and potential errors.
- In some cases when the grid size is large in 2-D decomposition, it can result in increased communication volume as the information is to be exchanged in both dimensions.
- Storing stencil coefficients for each control volume within each MPI process can lead to increased memory usage, when the grid size is large and reduce scalability.

## 2. Discuss if the decomposition of the domain changes the order of computations performed during a single jacobi iteration

(i.e., if you expect a numerically identical result after each iteration, or not)

In a Domain decomposition, the order of computations within a single Jacobi iteration does not change and remains consistent as each MPI process independently updates the values of grid points within the assigned sub-domain which are based on the values of previous iteration from the neighbouring grid points.

Also, because the computations proceed sequentially within each segment and subdomain, each grid point is updated one after the other in the decompositions leading to numerically identical result.

## 3. An extension to the ghost layer would be to use a wider layer (of more than one ghost cell). This allows to perform multiple independent iterations before a communication of the ghost layer has to take happen.

Comment in which situation (w.r.t. the available bandwidth or latency between MPI-processes) multiple independent iterations are potentially advantageous.

Using a wider layer to perform multiple independent iterations before setting up a communication of the ghost layers can be advantageous in many ways in the cases when the bandwidth between MPI processes is high, domain is large and the latency is low, so this decreases the time spent on communication compared to that of computation.

It will result in better utilization of the resources and improved performance of the Jacobi solver. Also, all processes can proceed in a similar pace and reduce communication delays ensuring more balanced progress across the entire domain.

However, also the amount of required memory increases proportionally which cannot be ignored in all setups.

In general, using multiple ghost layers and performing multiple independent iterations is beneficial, if the latency of the process communication is high.

## 4. How big is the sum of all L2 caches for 2 nodes of the IUE-cluster

The IUE-cluster is based on a Dell Power Edge R740 platform and consists of 10 regular compute nodes, 2 fat compute nodes and 1 login/storage node.

The regular compute nodes use two Intel Xeon Gold 6248 processors at 2.5 GHz with 20 cores and 40 available threads.

The cache size of each node is  $L2_{single} = 27.5 \text{ MB}$ . (<https://ark.intel.com/content/www/de/de/ark/products/192446/intel-xeon-gold-6248-processor-27-5m-cache-2-50-ghz.html>)

However, this is the sum of the total cache. For this kind of processor the Intel Xeon Gold 6000 series uses 1 MB L2 cache per processor. (<https://www.techpowerup.com/cpu-specs/xeon-gold-6342.c2440>).

Thus the sum of all L2 caches for two nodes is:

$$L2_{total} = 2 \cdot \frac{CPUs}{Node} \cdot L_{single} \frac{MB}{CPU} = 4 \cdot 20 \text{ MB} = 80 \text{ MB}$$