# UWLCM - Eulerian-Lagrangian cloud model for heterogeneous computing clusters[1]

Piotr Dziekan and Piotr Zmijewski

Institute of Geophysics, Faculty of Physics, University of Warsaw, Poland

[1] Dziekan, P., and Zmijewski P. *Geosci. Model Dev. Discuss.* in review, 2022.

UNIVERSITY OF WARSAW

## Abstract

Understanding of clouds in the atmosphere of Earth is important for weather and climate predictions. Detailed numerical modeling is one of tools used to study clouds. To model clouds it is necessary to model air flow (dynamics) and water droplets (microphysics). In University of Warsaw Lagrangian Cloud Model (UWLCM) microphysics are modeled using a novel Lagrangian particle-based method and dynamics are modeled in an Eulerian manner. Lagrangian calculations can be done by GPUs, while Eulerian calculations are parallelly done by CPUs. We present how UWLCM has been adapted to distributed memory systems and show scaling tests.

## Mixed Eulerian-Lagrangian model

- Finite differences Eulerian dynamics (large eddy simulation).
- Aerosols and droplets represented by computational particles called super-droplets (SD), each representing large number of real particles.

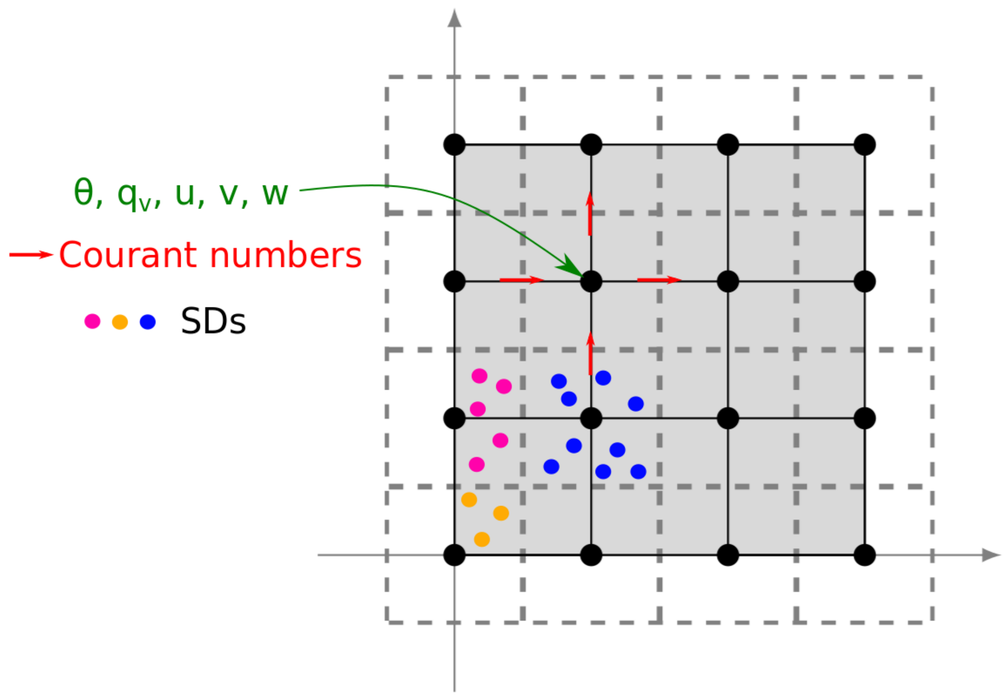


Figure 1: Schematic representation of the Eulerian-Lagrangian modeling approach. Shaded region is the modeled domain. Eulerian scalars are located at grid centers (black dots). Eulerian vectors are located at grid edges (red arrows). Super-droplets are colored to show the Eulerian cell they belong to.

## Domain decomposition

- MPI + OpenMP + intra-node CUDA memory copy.
- MPI decomposition along x axis.
- CPU and GPU data are co-located in space, CPU-CPU and GPU-GPU MPI communications only.
- Multiple GPUs per MPI task, further decomposition along x axis.
- CPU threads decomposition along y axis.

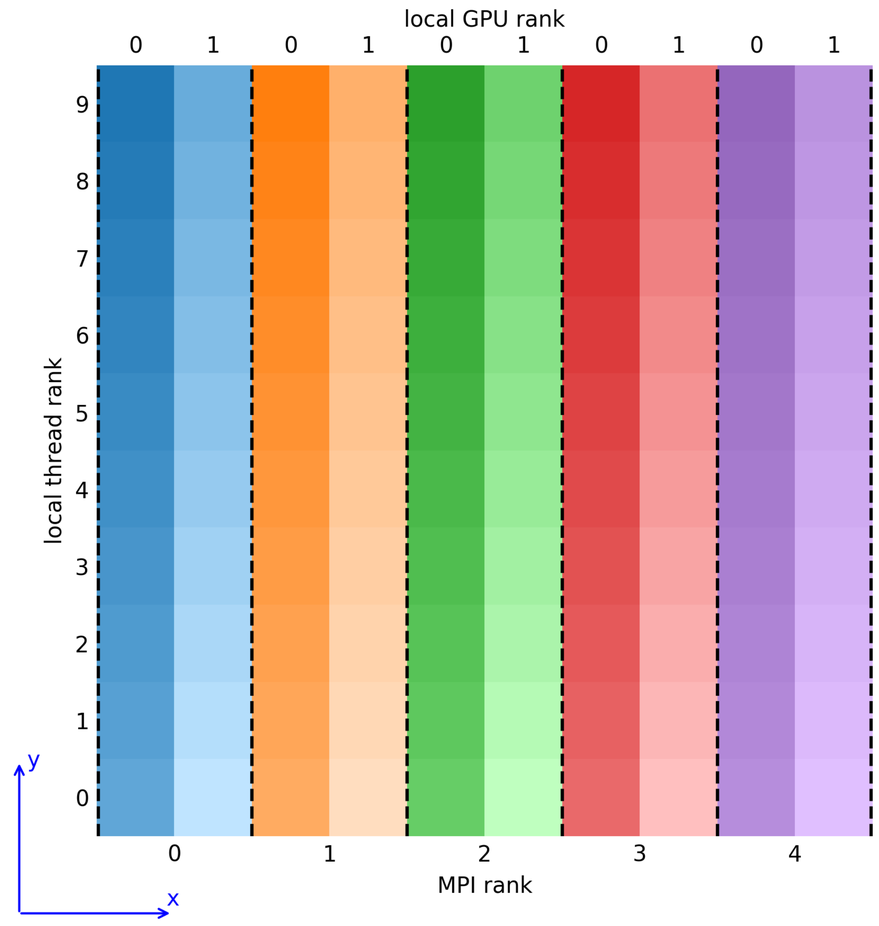


Figure 2: Top-down view of model domain showing Eulerian grid cells. Coloring depicts domain decomposition between MPI processes and shading depicts decomposition between threads and GPUs. Dashed black lines are boundaries that require MPI communication. Example with 5 MPI processes, each controlling 10 threads and 2 GPUs.

## Scaling scenarios

CPU workload is proportional to the number of grid cells $n_x * n_y * n_z$. GPU workload depends on the total number of computational particles $N_{\mathrm{SD}} * n_x * n_y * n_z$. GPU workload is limited by the GPU memory. Four cases are considered, in which more nodes are used for different reasons:

- *strong scaling* - Shorter simulation time.
- *SD scaling* - More detailed modeling of microphysics by increasing $N_{\mathrm{SD}}$. Weak scaling of GPU workload and strong scaling of CPU workload.
- *2D grid scaling* - Increasing number of cells in horizontal directions. Constant number of SD per cell. Weak scaling.
- *3D grid scaling* - Increasing number of cells in all directions. Constant number of SD per cell. Weak scaling.

## Scaling results

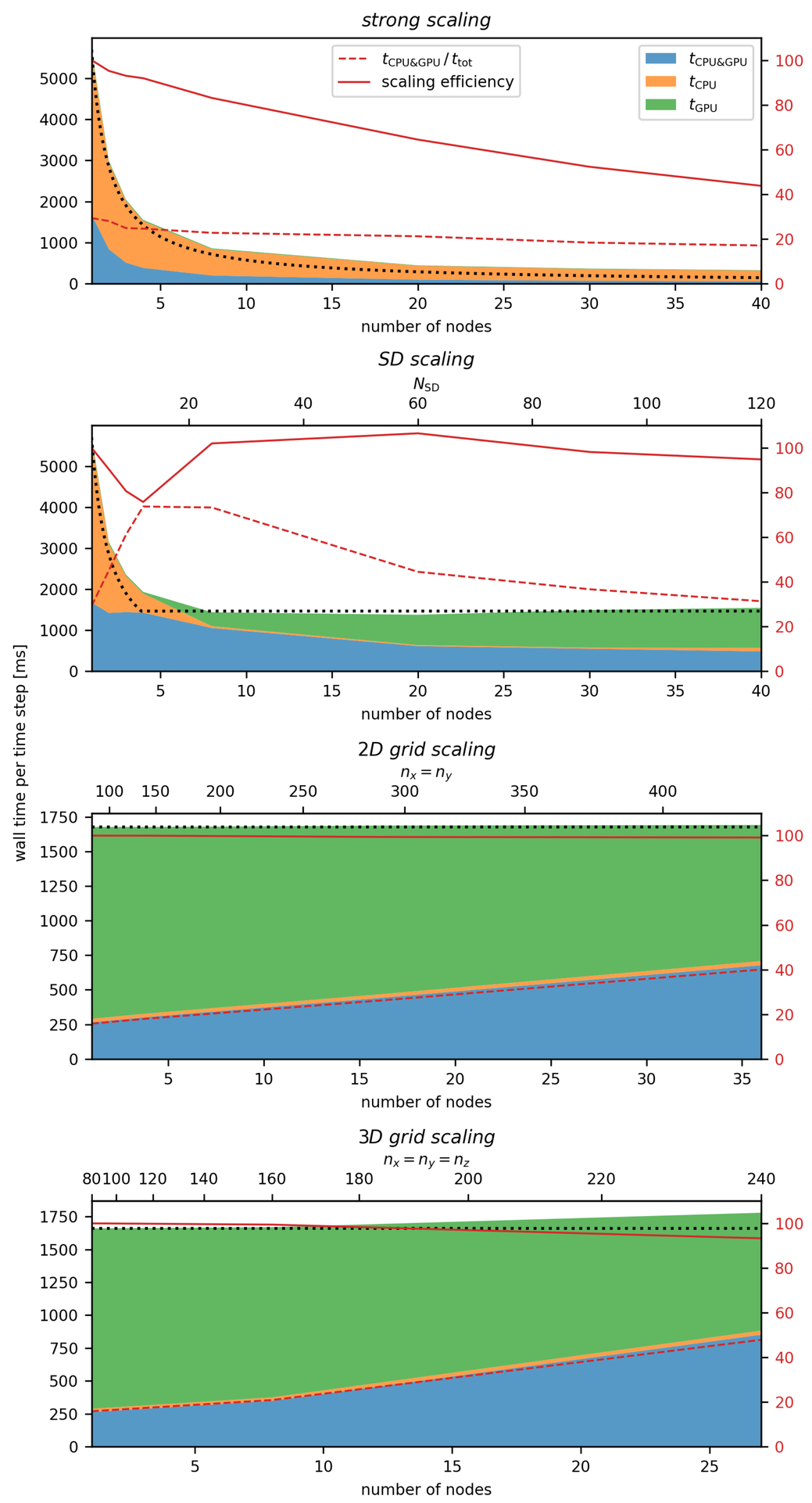


Figure 3: Stacked wall time of simultaneous CPU and GPU (blue), CPU-only (orange) and GPU-only (green) computations vs number of nodes. Solid red line is scaling efficiency. Dashed red line is the ratio of wall time of simultaneous CPU and GPU computations to the total wall time. Black dotted line is perfect scaling.

- For an optimal ratio of CPU and GPU workloads CPUs and GPUs work simultaneously ca. 80% of time.
- GPU wall time scales better than CPU wall time.
- Due to differences in scaling of GPU and CPU computations, there is an optimal number of nodes for which the time of simultaneous CPU and GPU computations is maximized.
- Scaling efficiency of the total wall time is ca. 100 % up to the optimal number of nodes.

www.igf.fuw.edu.pl pdziekan@fuw.edu.pl

work financed by NATIONAL SCIENCE CENTRE POLAND