

Distributed-memory parallelization on GPUs and accelerators

Table of contents

Organization	1
How this site is organized	1
Getting started	2
Assessment	2
Contact	2

This website contains material for the class **High-performance computing: Distributed-memory parallelization on GPUs and accelerators** run by the [simulation group](#) at the [Department of Microsystems Engineering](#) of the [University of Freiburg](#) in the summer term 2026. The course is taught by Lars Pastewka and Yizhen Wang.

In this course you will write your own fluid-dynamics code: a two-dimensional [lattice Boltzmann](#) solver, written in C++ on top of the [Kokkos](#) hardware abstraction layer and parallelized for distributed-memory machines with MPI. The course thesis is *parallelization for heterogeneous compute architectures* using Kokkos (or another abstraction layer of your choice). The goal is to understand how heterogeneous compute environments — multicore CPUs, GPUs and clusters thereof — work and how to program them.

Organization

- **Lecture:** Wednesday 10:00–12:00, SR 00 017/19, Building 101
- **Tutorial:** Wednesday 14:00–16:00, R 00 019, Building 74
- **Communication:** We use a [Matrix room](#) for announcements and questions. Please join at the beginning of the term.
- **Online material:** All course material is published at <https://pastewka.github.io/Accelerators/>.

How this site is organized

- **Learning material** — the lecture notes that accompany the weekly lectures, from parallel hardware to the lattice Boltzmann method and MPI.
- **Project** — six milestones that pace the development of your own code, from setting up the build environment to a fully distributed solver. Starting point is the skeleton repository at <https://github.com/imtek-simulation/cmake-skeleton/>.
- **Notes** — practical how-tos on tooling: CMake, debugging, benchmarking, scaling studies, running on the bwUniCluster, and more.

- **Lecture slides** — the slide decks shown in the lectures.

Getting started

In the first week, you should:

1. Set up your build environment as described in [milestone 1](#).
2. Create your own repository from the [skeleton template](#) and make sure it compiles.
3. Apply for access to the [bwUniCluster](#) — accounts take time to be approved, so do this early.
4. Join the [Matrix room](#).

Assessment

The course is graded via a colloquium in the final week of the term, in which you present the results of your project. Figures in your reports and presentations must follow our [figure rules](#).

Contact

Please direct feedback and issues with the material posted here to [Lars Pastewka](#) or [Yizhen Wang](#).