Thematic CERN School of Computing on Scientific Software for Heterogeneous Architectures

Sunday 5 October 2025 - Saturday 11 October 2025

Medils Institute, Split

Academic programme

The school will focus on the theme of **Scientific Software for Heterogeneous Architectures**. The complete programme will offer 22 hours of lectures and hands-on exercises, and a student presentations session.

Introduction lecture

Preparing for the HL-LHC computational challenge
HEP data processing and analysis workflows
Upgrades of the LHC accelerator and experiments
Evolution of hardware and computing infrastructure
Impact on HEP data processing software

Track 1: CPU Architecture and High Performance

4 hours of lectures and 2 hours of hands-on exercises

CPU Hardware Architecture and Evolution
Hardware evolution of the CPU
Memory hierarchy, caching, NUMA
Microarchitecture of modern CPUs

Performance Analysis on Modern CPUs
Performance analysis tools for Linux
CPU features for performance analysis
Top-down microarchitecture analysis

Low-level Performance Optimization Guidelines
Main sources of performance bottlenecks
Floating point arithmetics performance
Advanced low-level performance tuning

Data-Oriented Design
Principles of data-oriented design
Memory access and data-type profiling
Data structure performance optimization

Track 2: Parallel and Optimised Scientific Software

5 hours of lectures and 2 hours of hands-on exercises

Writing parallel software
Amdahl's and Gustafson's laws
Asynchronous execution
Finding concurrency, task vs. data parallelism

Using threading in C++ and Python, comparison with multi-process
Resource protection and thread safety
Locks, thread local storage, atomic operations

Writing efficient software
virtues of functional programming
practical usage in C++ and why it's efficient
how to help the compiler to produce faster code
doing more at compile time
Templating versus inheritance, pros and cons of virtual inheritance

Optimizing existing large codebase
Measuring performance, tools and key indicators
Improving memory handling
The nightmare of thread safety
Code modernization and low level optimizations
Data structures for efficient computation in modern C++

Practical vectorization
Measuring vectorization level
What to expect from vectorization
Preparing code for vectorization
Vectorizing techniques in C++: intrinsics, libraries, autovectorization

Track 3: Programming for Heterogeneous Architectures

4 hours of lectures and 4 hours of hands-on exercises

Scientific computing on heterogeneous architectures
Introduction to heterogeneous architectures and the performance challenge
From general to specialized: Hardware accelerators and applications
Type of workloads ideal for different accelerators
Trade-offs between multi-core and many-core architectures
Implications of heterogeneous hardware on the design and architecture of scientific software
Embarrassingly parallel scientific applications in HPC and CERN

Programming for GPUs
From SIMD to SPMD, a programming model transition
Thread and memory organization
Basic building blocks of a GPU program
Control flow, synchronization, atomics

Performant programming for GPUs
Data locality, coalesced memory accesses, tiled data processing
GPU streams, pipelined memory transfers

Under the hood: branchless, warps, masked execution
Debugging and profiling a GPU application

Design patterns and best practices
Good practices: single precision, floating point rounding, avoid register spilling, prefer single source
Other standards: SYCL, HIP, OpenCL
Middleware libraries and cross-architecture compatibility
Reusable parallel design patterns with real-life applications

Additional lectures

Student lightning talks session