Journal of Parallel and Distributed Computing 201 (2025) 105081

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

journal homepage: www.elsevier.com/locate/jpdc

The European master for HPC curriculum

Pascal Bouvry [a], Mats Brorsson [a], Ramon Canal [b], Aryan Eftekhari [c], Siegfried Höfinger [d], Didier Smets [e], Harald Köstler [f], Tomáš Kozubek [g], Ezhilmathi Krishnasamy [a], Josep Llosa [b], Alexandra Lukas-Rother [f], Xavier Martorell [b], Dirk Pleiter [h,*], Ana Proykova [i], Maria-Ribera Sancho [b], Olaf Schenk [c], Cristina Silvano [j]

[a] Parallel Computing & Optimisation Group and FSTM-DCS/SnT, University of Luxembourg, Esch-sur-Alzette, 4364, Luxembourg
[b] Universitat Politècnica de Catalunya, Barcelona, 08034, Spain
[c] Institute of Computing, Faculty of Informatics, Università della Svizzera italiana, Lugano, 6962, Switzerland
[d] Technische Universität Wien, Wien, 1040, Austria
[e] Laboratoire Jacques-Louis Lions, Sorbonne Université, Paris, 75005, France
[f] Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, 91058, Germany
[g] IT4Innovations, VSB-Technical University of Ostrava, Ostrava, 70800, Czech Republic
[h] Division of Computational Science and Technology, KTH Royal Institute of Technology, Stockholm, 10044, Sweden
[i] HPC laboratory, Faculty of Physics, University of Sofia, Sofia, 1164, Bulgaria
[j] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico Milano, Milano, 20133, Italy

A R T I C L E I N F O

Keywords:
Computing education
High-performance computing
Master in HPC
Model curricula

A B S T R A C T

The use of High-Performance Computing (HPC) is crucial for addressing various grand challenges. While significant investments are made in digital infrastructures that comprise HPC resources, its realisation, operation, and, in particular, its use critically depends on suitably trained experts. In this paper, we present the results of an effort to design and implement a pan-European reference curriculum for a master's degree in HPC.

1. Introduction

The use of High-Performance Computing (HPC) is critical to address many of the grand scientific, engineering, and societal challenges. The scientific challenges requiring HPC range from solving fundamental open questions of physics at the largest and smallest scales to challenges in life sciences like the decoding of the human brain. One of the most prominent challenges where HPC plays a crucial role is climate adaptation. Model simulations using HPC systems have played a key role in predicting climate change. Destination Earth is currently developing large-scale climate and weather digital twins, which will push the use of HPC to the next level [20]. The importance of HPC is also reflected in the growing volume of the market. Analysts from Hyperion Research predict the annual growth of the HPC market of 8.2 % over the next five years [6].

The role of HPC has also been recognized at the political level. For example, in Europe, a new organisation has been created called Joint Undertaking EuroHPC,[1] which establishes an exascale HPC infrastructure and supports related research. The strategy is to coordinate efforts and pool resources in order to facilitate a globally leading role of European stakeholders. Similar efforts are ongoing in other regions of the world, most notably in China, India, Japan, and the US.

The success of these efforts critically depends on the availability of appropriately trained researchers, engineers, and other experts. Ensuring the availability of such experts is challenging due to the breadth of required skills coming from various science and engineering domains. HPC is a very interdisciplinary area, which needs to be reflected in the

* Corresponding author.
E-mail address: pleiter@kth.se (D. Pleiter).

[1] https://eurohpc-ju.europa.eu/.

https://doi.org/10.1016/j.jpdc.2025.105081
Received 1 June 2024; Received in revised form 22 December 2024; Accepted 1 April 2025
Available online 3 April 2025

education. This ranges from various domains of computational science and engineering, from computer science to computer and electrical engineering.

To address the challenge of the future workforce, EuroHPC has provided funding for the EUMaster4HPC project.[2] The project consortium includes dozens of universities and educational institutions throughout Europe, HPC centres, and private companies that are active in the area of HPC. A primary objective of the project is the design of a common European master reference curriculum in HPC. It created an opportunity to formulate a holistic curriculum that draws on the knowledge and experience of a wide range of stakeholders, expands existing curricula with unique characteristics, and promotes new teaching and training methods. Examples are the training of transversal skills or the implementation of challenges. The curriculum is designed in a modular way to facilitate its uptake by a large number of universities throughout Europe. This makes it easier to implement it within existing programs. Furthermore, the project implements a pilot program involving about 150 students, who benefit from a mobility program to receive the most advanced education and training from multiple organisations.

With this paper, we make the following contributions:

1. Documentation of a holistic curriculum for a master's in HPC education, with a focus on its unique features.
2. Discussion of the methodology that led to the formulation of this curriculum.
3. Evaluation of early experiences with the implementation of the curriculum and the training of EUMaster4HPC students.

This paper is organised as follows: In Section 2, we discuss a selection of related work. The next Section 3, documents the methodology that resulted in the curriculum, which is documented in Section 4. In Section 5, we report on different aspects of the initial implementation of the curriculum. In Sections 6 and 7, we provide a first evaluation of the use of the curriculum based on a case study and a survey. We end with a summary and conclusions in Section 8.

2. Related work

The importance of quality education in HPC for basic sciences and engineering has long been recognised (see, for example, the early report by Long et al. [7]). An interesting newer development are the offerings to students who do not major in computer science, but instead are introduced to core concepts, thereby stressing the new objective of computational thinking [21].

EUMaster4HPC has extended the scope further by also considering HPC system administrators as a possible future role, which is neither covered by education for system administration nor the traditional HPC education. Such an extension has earlier been proposed in [24].

One particular challenge for education in the area of HPC is the fast evolution of the computational methods as well as the architectures and technologies. One notable example is the introduction of heterogeneous computing platforms. Qasem et al. suggested addressing the growing need for handling such platforms by defining modules that introduce fundamental concepts of heterogeneous (or accelerated) computing into undergraduate education in computer science and computational engineering [13]. Another strategy is the stronger involvement of supercomputing centres. A practical example is given by Chen et al. [3] reporting on efforts at China's National University of Defense Technology (NUDT) that has hosted the Tianhe supercomputers.

Over the years, numerous studies have been published introducing new methods for teaching HPC. One example is the organisation of hackathons as part of university education, of which the Brazilian Parallel Programming Marathons [8] is an example. The authors consider hackathons as a good way of helping students with a paradigm shift from traditionally sequential programming techniques to parallel and distributed computing. It also relates to our efforts to involve students from different universities in obtaining practical experiences on HPC systems (see Section 5.4.3), which others have proposed as an educational tool (see, e.g., [5]). Another example of papers describing the successful adoption of new ways of teaching HPC is the use of Jupyter notebooks [9]. Building small-scale, low-cost computing clusters based on simple devices, e.g. edge devices like Raspberry Pis, is not new (see, e.g., [4]) but still has a limited uptake. The integration of such a methodology within courses is discussed in [19]. More recently, Xu Zhiguang combined the approach of building clusters on-site using low-cost devices with the use of high-end computing resources in the cloud leveraging Google collabs [23].

Various works document the development of specific courses. One nice recent example is Varbanescu et al. [18] who developed a full performance engineering courses and documented the setup. Such courses fill an important gap as there is a need for methodology education in the area of performance analysis and engineering [1]. EUMaster4HPC addresses this by foreseeing a specialisation as Performance Analyst and Advisor (see Section 4.2.2).

A broader scope initiative in curriculum development for Parallel and Distributed Computing has been launched with the guidelines published by the NSF/IEEE-TCPP Curriculum Working Group [11]. An analysis of current trends in HPC education and corresponding challenges for future directions has been reported by the Working Group on Innovation and Technology in Computer Science Education [14]. Competency-based learning was identified as a needed transformation from the more prevalent knowledge-based learning in HPC education [2]. A good and continuous source of publications and presentations related to teaching methods and new courses in the area of HPC education are several workshop series:

- The EduPar workshops take usually place as part of the IEEE International Parallel & Distributed Processing Symposia (IPDPS) [10],
- EduHPC is a series of workshops in the context of the International Conferences for High Performance Computing (SC).[3]
- Edu-HiPC workshops have been organised for a few years on an annual basis in India.[4]

In the context of EuroHPC, a related effort is to define a training baseline that is not limited to university education at the master's level. This effort is being carried out by the CASTIEL2 and EuroCC2 projects.[5] The training baseline for HPC, High-Performance Data Analytics (HPDA), and Artificial Intelligence (AI) is defined around six different career path profiles. One of the main design goals of the training baseline was that each of the building blocks remain modular, allowing for further adaptations in future releases. The target sectors comprise "Research/Academic++", "SMEs" and "Big Industry" and corresponding career path profiles are "HPC application developer", "System Administrator", "HPC application user", "Data scientist / Analyst", "Executive / Technical lead" and "System designer". Each career path profile is assigned a specific list of training units, providing a stepwise acquisition of skills needed in the respective target profession.

A notable difference to the EUMaster4HPC curriculum is the significantly reduced workload for a particular career path profile. Career path profiles are considered more as additional training within the realm of lifelong learning [15] to offer interested candidates a non-traditional

[2] https://eumaster4hpc.uni.lu/.

[3] For the 2024 edition, see https://tcpp.cs.gsu.edu/curriculum/?q=eduhpc24.

[4] For the 2024 edition, see https://tcpp.cs.gsu.edu/curriculum/?q=eduHiPC24.

[5] https://www.eurocc-access.eu/.

route to specialization. Preliminary discussions had identified micro-credentials [16] as a viable way of certification of the successful completion of a particular career path profile — albeit at the lower end of approximately 3-6 ECTS. Given the explicit emphasis on industry, the training baseline could offer interesting extensions to the EUMaster4HPC curriculum.

3. Methodology

The process of developing the curriculum has been a multistage effort. It started with an analysis of the need for experts trained in HPC-related topics in the private and academic sectors. This resulted in a set of skill profiles. Initial results of this effort have been published earlier [12]. For the design of the curriculum, a balance had to be maintained between fixed characteristics and variable elements. The fixed characteristics include a duration of 4 semesters. To obtain a master's degree, students are expected to take courses that translate into 30 ECTS per semester and a total of 120 ECTS to obtain a master's degree. To keep the curriculum flexible, the teaching content has been defined in terms of small modules with 1 to 3 ECTS per module and an effort of approximately 10 to 30 teaching hours. Although the set of modules, presented in the next Section, is based on current practices, it has been augmented and adjusted to take into account new trends and upcoming needs into account.

To assess the upcoming private sector requirements, a dialogue has been conducted with relevant companies through questionnaires, structured interviews, and discussions during conferences and other meetings related to HPC. The interaction was guided by an analysis of job advertisements. As these reflect only current needs, the dialogue focused on future trends. The result of this effort has been the identification of six different skill profiles:

- Computational scientist, HPC application support specialist,
- Application software developer,
- HPC system software developer, HPC software engineer,
- HPC hardware developer,
- HPC system administrator, and
- HPC architect.

A similar approach has also been taken for the academic sector. The results have been the following four profiles:

- Applications
- Parallel programming and tools support
- DevOps (system support and development)
- System architect

The assessment of the skill requirements in both sectors also confirmed the need to offer education and training in transversal and professional skills. These skills are often not yet addressed in existing curricula for HPC studies. We considered a wide range of skills, including those related to research techniques, topics related to entrepreneurship, and skills to address legal topics. For developing this part of the curriculum, the following steps have been performed:

- Analysis of existing curricula and initiatives across the organisations involved in the EUMaster4HPC project.
- Selection of relevant transversal skills and respective learning outcomes.
- Exploration of the educational approaches to achieve the expected learning outcomes.

For an overview of this part of the curriculum, see Section 4.3.

For each of the modules, learning outcomes have been defined, taking Bloom's taxonomy into account. The main strength of our perspective is the adoption of a structural methodology to offer a comprehensive approach to curriculum design. This methodology enables university programs to consider student requirements, evaluate the feasibility of teaching materials and pedagogical approaches, and develop assessments with concrete learning objectives. Bloom's taxonomy provides a framework for understanding the learner's needs, which is useful at the graduate level, as students may be accepted for reasons other than ability.

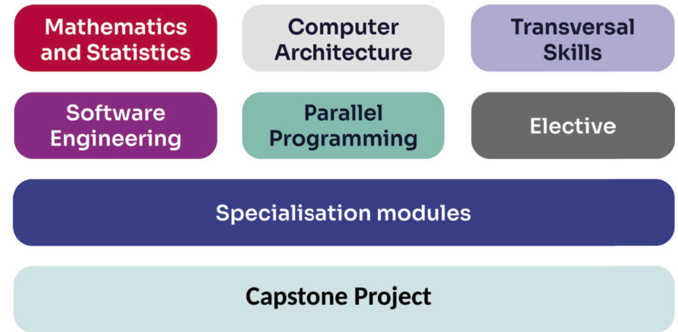

Fig. 1. Proposed fundamental modules and curriculum concept in EUMaster4HPC.

A new type of module, balancing modules, has been introduced to offer more flexibility in terms of study paths. These modules support the intake of students with bachelor's degrees other than computer science. An overview of the proposed balancing modules is provided in Section 4.4.

4. The curriculum

In this section, the curriculum is introduced. A visual overview is provided in Fig. 1 It shows the different domains of the fundamentals, which are introduced in Sections 4.1 and 4.3. The specialisations will be discussed in Section 4.2. Finally, the balancing modules will be presented in Section 4.4.

4.1. Fundamentals

We have identified five domains or specific areas for the fundamentals:

- Mathematics and Statistics
- Software Engineering
- Parallel Programming
- Computer Architecture
- Transversal Skills

The approach that we followed was to define small modules that can be combined to provide a number of courses, see Fig. 2. Transversal skills address issues like diversity, inclusion, and gender bias as a cross-cutting topic for all modules. Modules are set to have 3 ECTS so that by combining them it is easy to build regular courses of 6 ECTS, or larger ones, up to 9 ECTS. Each university imparting the master can decide to combine the modules as they prefer, provided that the final content is comparable. Some of the modules in each domain are "basic", in the sense that they overlap with contents potentially done in a previous bachelor's degree. Those modules are only assigned to students with different backgrounds. For example, students coming from a non-technical bachelor can be assigned modules such as "Basic Computer Architecture", "Basic Parallel Programming", or "Operating Systems".

Table 1 shows the weight of each area in the curriculum, as well as the amount of knowledge implemented as Balancing Modules, and the amount contributing to the core of Fundamentals.

The following subsections show the organization of the modules for each domain and their learning outcomes:

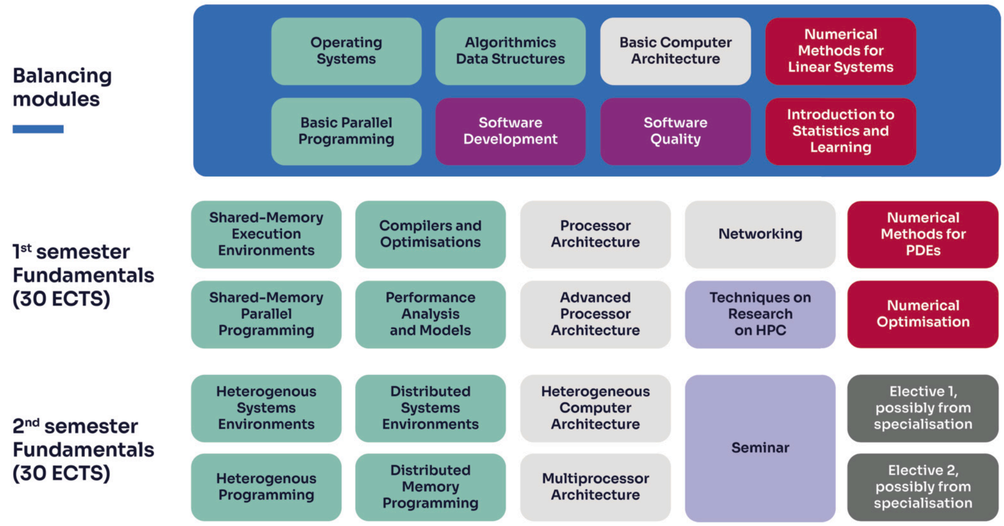

Fig. 2. Proposed fundamentals in EUMaster4HPC (Year One).

Table 1
Proposed ECTS for fundamentals and balancing modules in year one.

Area	Balancing ECTS	Fundamentals ECTS	Total
Mathematics and Statistics	6	6	12
Software Engineering	6	–	6
Parallel Programming	9	24	33
Computer Architecture	3	15	18
Transversal Skills	–	9	9
Total	24	54	78

4.1.1. *Mathematics and statistics*

This domain gets the students to a common level, by allowing those with no or low exposure to Algebra or Statistics to take the Balancing Modules of Numerical Methods for Linear Systems and Introduction to Statistics and Learning.

Once all students have been brought to a common level, the master program includes Numerical Methods for Partial Differential Equations, including the application of numerical analysis, considering stability, consistency, and rate of convergence, and the methods of finite elements, Galerkin, Fourier and spectral methods. On the practical side, students develop and compare the performance of different numerical methods.

Later in the domain, Numerical Optimization includes the knowledge of mathematical aspects of optimization, smooth and non-smooth optimization techniques, stochastic gradient and Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithms. On the practical side, students develop sample applications using those optimization methods.

4.1.2. *Software engineering*

Software engineering techniques and tools are already common for students coming from CS-related backgrounds. For this reason, we consider these topics to belong to Balancing Modules. Two modules are defined: Software Development Techniques and Tools including the description and use of the software life cycle models, standard procedures (like ISO/IEC/IEEE 24765), requirements, use-cases, portability and performance portability; Software Quality, Validation and Testing includes the consideration and implementation of software sustainability, technical depth assessment, software quality criteria, testing methods, and their practical application.

4.1.3. *Parallel programming*

The domain of Parallel Programming includes 3 modules from the set of Balancing topics that are usually overlapping with CS-related bachelors: Operating Systems, Basic Parallel Programming, and Algorithms and Data Structures.

On top of those, the master curriculum provides 4 additional areas of knowledge:

Parallel programming tools The area includes Compilers and Optimizations, and Performance Analysis and Models. In this area the students get to know, the abilities to explain and use the concepts related to the compilation process, programming languages, code optimizations, exploitation of the memory hierarchy, main techniques for obtaining parallelism (instruction-level parallelism, vectorization, and multi-threading), performance analysis metrics and laws (Amdahl's and Gustafson's law), performance analysis models, and tools. On the more practical side, use profiling-based optimizations, and detect overhead and bottlenecks on parallel applications.

Shared-memory environments and programming The area covers the description and use of single-node shared-memory computers, applying parallelization techniques using well-known, standard, and state-of-the-art parallel programming models (currently proposed are MPI, OpenMP, SYCL, etc.). Use of automatic parallelization, and combine parallelization and vectorization. On the practical side, apply parallel constructs, tasking, and vectorization to particular benchmarks and code structures.

Heterogeneous environments and programming The area includes the description and use of single-node shared-memory systems with accelerators (currently GPUs, TPUs, DPUs and FPGAs), applying offloading techniques and managing data transfers with standard and state-of-the-art heterogeneous programming tools (currently CUDA, OpenCL, OpenACC, OpenMP target, SYCL, etc.), improve the use of the accelerators for particular algorithms. On the practical side, be able to analyse the performance of heterogeneous parallel applications and detect drawbacks.

Distributed-memory environments and programming The area covers the description and use of cluster, grid and cloud environments, compilation tools, organisation of applications based on workflows and pipelines, microservices, and parallelization of applications with standard, state-of-the-art distributed programming tools (currently MPI, etc.). On the practical side, use the techniques to exploit interoperability across different programming models (currently MPI+OpenMP, MPI+CUDA, etc.).

4.1.4. *Computer architecture*

The domain of Computer Architecture includes one module from the set of Balancing topics that usually overlaps with CS-related bachelors: Basic Computer Architecture.

On top of those, the master curriculum provides 4 additional areas of knowledge:

Processor architecture The area covers basic and advanced processor architecture. In this area, the students get to know the different types of processor architectures, the fundamental mechanisms for static ILP processors, software-level instruction scheduling techniques (list scheduling, unrolling, software pipelining), the memory hierarchy and its components, cache design and cache strategies and protection and security issues related to the processor architecture. On the more advanced side, students learn and get to use the fundamental mechanisms for dynamic ILP processors (out-of-order execution, branch prediction, exception handling, instruction and data prefetch, etc.) and consider protection and security issues in the processor design. On the practical side, analyse the performance and assess the impact on power and energy of single-threaded programs on in-order and out-of-order processors, and apply simple optimizations to code fragments.

Heterogeneous computer architecture The area covers accelerators (e.g., GPUs, FPGAs), their architecture and micro-architecture, hardware techniques for thread scheduling and synchronization, accelerators memory hierarchy, FPGA building blocks and FPGA interconnects. On the practical side, explore accelerator performance capabilities for synthetic benchmarks, program sample applications with FPGA programming workflows and explore the performance of selected computational motifs on accelerators (e.g., structured grids, FFT, dense linear algebra).

Multiprocessor architecture The area covers the implementation of different models of thread-level and data-level parallelism, such as core multi-threading, many-core processors, vector units, multi-chip modules, the impact of multiprocessing on the processor ISAs, implementation of hardware-level synchronization mechanisms, the operation of the memory hierarchy on multiprocessors: snooping and directory-based coherence protocols, coherency, and consistency. Design at the architectural level of hardware for shared-memory, basic components of modern parallel systems, including multiple processors, cache hierarchies and networks. And quantify performance metrics for parallel systems and evaluate parallel computer organizations. On the practical side, develop small benchmarks evaluating the performance of hardware synchronization mechanisms, the overhead of thread and process management and evaluate the mechanisms implementing data coherence and consistency.

Networking The area covers network topologies, routing algorithms, flow control, congestion control mechanisms, error detection/correction mechanisms, ISO/OSI model, network protocols (TCP/IP, RDMA, etc.) and network performance metrics (e.g., latency, bandwidth, and quality of service). On the practical side, develop benchmarks implementing communications with commonly used protocols (TCP/IP, RDMA, etc.), and analyse the performance of a network considering latency, bandwidth and quality of service.

4.1.5. Transversal skills

The Transversal Skills are distributed across the different modules of the curriculum. In this paper, we collected all the information, and it is presented in Section 4.3.

4.2. Specialisations

Specialisations are typically offered in the 3rd semester of the Master's degree. Within the EUMaster4HPC project, we have identified the following four specialisations as the most significant through the analysis of the job market around the private and academic sectors:

- Numerical and Data Specialist for Science Domains
- Performance Analyst and Advisor
- System Development and Support
- System Architect

Table 2
Proposed ECTS for specialisations and modules in year two.

Specialisation	Modules	ECTS
Numerical and Data Specialist for Science Domains	20	66
Performance Analyst and Advisor	15	51
System Development and Support	17	57
System Architect	16	54

For each specialisation, several teaching modules have been identified, including all relevant and actual topics in the form of learning outcomes that will allow students to gather additional knowledge and expertise within the selected specialisation. Each module has its prerequisites, the contents the students will learn, and various suggested practical topics. The introduced specialisations complement the fundamentals presented above and together are the basis for building the general curriculum of the master program.

The number of modules currently identified and respective ECTS offered by each specialisation are shown in Table 2. Each specialisation should contain a minimum of 30 ECTS of modules. Practically, they have more ECTS to allow more freedom to the implementing institutions. The number of ECTS to pass specialisation is 30. Each specialisation contains regular modules for 3 ECTS, a project module for a maximum of 9 ECTS, and a list of modules shared with other specialisations. All modules are grouped into thematic areas (see Fig. 3), which provide the students with the learning outcomes identified for each specialisation and further described in the following subsections.

The availability of courses in different implementations at different institutions may result in slight changes in the modules and ECTS.

4.2.1. Numerical and data specialist for science domains

This specialisation focuses on graduate experts in scientific computing, mathematical modelling, the development of numerical algorithms for supercomputers, parallel programming, and knowledge of some application domains, such as engineering, physics, and biomedical. Furthermore, AI modelling with HPC capability and high-performance data analytics that are available for targeting research and applications is trending in AI / ML. They will be able to interact in interdisciplinary teams with experts and scientists in the application domain on the development of efficient parallel implementations, performance analysis, large data analysis, and visualisation. The following will discuss more details about the learning outcomes of this specialisation.

Fig. 4 and Table 3 show an overview of the areas targeted and the expected ECTS within this specialisation. A detailed description of each area follows.

Scientific computing The topic will focus on efficient numerical algorithms and analysis for differential equations, especially for partial differential equations. In particular, this area contains modules such as Computational Numerical Linear Algebra, Parallel Numerical Methods, and Implementation of Partial Differential Equation (PDE) Solvers. They include algorithms for linear algebra: orthogonalisation, least-squares methods, solving large linear systems, and studying their stability and communication requirements. For example, how good is the stability of the algorithm with respect to the level of parallelism? Furthermore, different domain decomposition methods will be studied in the context of parallel numerical methods, such as multigrid and Schwarz methods. Finally, we introduce different well-known numerical methods for solving PDE that will be studied in detail, such as finite difference, finite volume and finite element methods. The study group is expected to implement any of those methods from scratch, and also to have hands-on experience with handling (or using) libraries used within those numerical methods. Importantly, all of those methods and techniques will always be discussed in the context of parallel architecture machines, including GPU-accelerated systems.

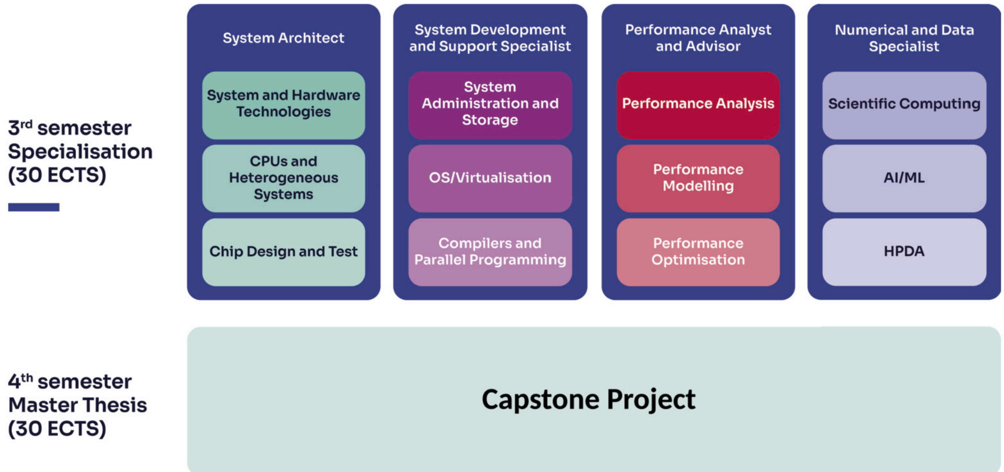

Fig. 3. Proposed specialisations in EUMaster4HPC (Year Two).

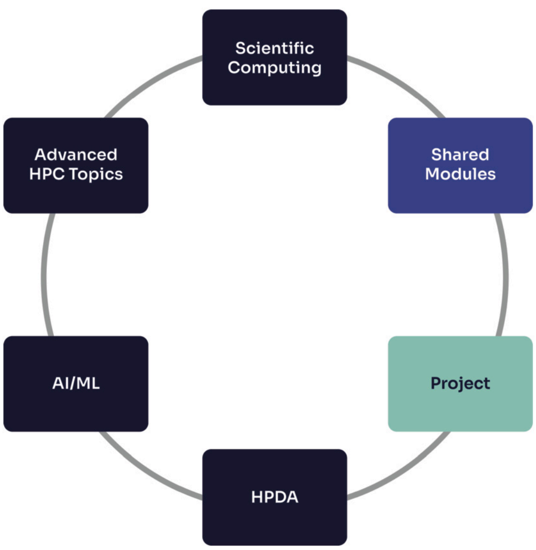

Fig. 4. Proposed modules (study areas) within Numerical and Data Specialists for Science Domains.

Table 3
Proposed modules (study areas) within Numerical and Data Specialist for Science Domains.

Area	Modules	ECTS
Scientific Computing	3	9
Advanced HPC Topics	3	9
AI/ML	3	9
HPDA	3	9
Project	**1**	**9**
Shared Modules	7	21
In Total	**20**	**66**

Advanced HPC topics for science domains It introduces advanced topics such as quantum computing and domain-specific programming models. Regarding quantum computing, it covers quantum computer architecture (for example, qubits) and programming models (for example, Qiskit, Ocean, Cirq, etc.) and applying the quantum computing technique (using the real machine or simulator) to solve applications from science and engineering. Furthermore, within domain-specific programming models, especially focusing on embedded systems and their programming models. For example, programmability and running numerical or AI algorithms such as in FPGAs and IPU architectures. Finally, it introduces how to run a large science and engineering simulation (for example, medical engineering, fluid dynamics, solid mechanics and dynamics, life sciences, material sciences, and chemistry) using large HPC machines, in particular, by studying software packages for simulation, parallelisation strategies, methods, mathematical properties, and implementation approaches on HPC systems.

AI/ML It covers the essential topics of expertise in machine learning, artificial intelligence, and computational statistics. Machine learning will focus on topics such as model evaluation and selection, anomaly detection, conformal learning (prediction with guarantees of accuracy), causal inference (identification of causal relationships), and hybrid approaches such as integrating numerical methods with artificial neural networks for science and engineering problems. Their HPC implementation uses existing software frameworks, for example, TensorFlow and PyTorch, in parallel architecture, using CPU and GPUs.

High-performance data analytics It targets big data, high-dimensional data analysis, and data visualisation. Big data analysis includes file systems and databases for big data and machine learning for big data connected with its analysis using tools and programming models, such as Apache Spark and the Hadoop Map-Reduce ecosystem. With respect to high-dimensional data analysis, more explanation will be given to the geometry of high-dimensional data sets, nonlinear dimension reduction and manifold models, linear dimension reduction, principal component analysis and kernel methods, mathematical distance and dimension reduction, singular value decomposition, and principal component analysis. Finally, we introduce data handling and visualisation tools like ParaView and VisIt.

Project It encourages the study group to focus either on mathematical modelling (numerical methods, optimisation, and statistical modelling) or parallel implementation and analysis of mathematical algorithms on HPC machines, especially considering real-world applications such as aerodynamics, machine learning, material science, etc.

Shared modules This will help the study group integrate their knowledge with other beneficial specialisation topics, considering the area of scientific computing and, for example, covering advanced topics in large-scale computer architecture (from "System Architect"), operating systems, storage, Virtualisation & Containerisation (from "System Development and Support") and performance analysis tools for heterogeneous HPC machines (from "Performance Analyst and Advisor"). Essentially,

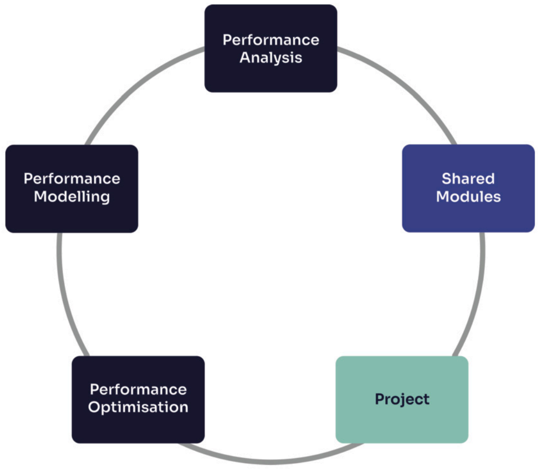

Fig. 5. Proposed modules (study areas) within Performance Analyst and Advisor.

Table 4
Proposed modules (study areas) within Performance Analyst and Advisor.

Area	Modules	ECTS
Performance Analysis	2	6
Performance Modelling	2	6
Performance Optimisation	2	6
Project	1	9
Shared Modules	8	24
In Total	**15**	**51**

it will give a further better understanding of the detailed study of computational science.

4.2.2. Performance analyst and advisor

This specialisation targets graduate experts in performance analysis and tuning of complex application codes for modern computer architectures and supercomputers. It follows a structured performance engineering approach consisting of performance analysis based on measurements, performance modelling to estimate performance limits and bottlenecks on a certain platform and performance optimisation using hand-crafted or automatically generated architecture-specific implementations.

Fig. 5 and Table 4 show an overview of the areas targeted and the expected ECTS within this specialisation. A detailed description of each area follows.

Performance analysis The content includes criteria for selecting the right tool for specific performance analysis tasks and introduces various performance tools such as profilers, tracers, debugging tools, and monitoring tools. Students will learn to profile parallel applications to identify performance bottlenecks, perform real-time debugging and tracing of multithreaded and distributed applications, and monitor system performance and resource utilization. The modules provide an in-depth exploration of profiling tools like Nvidia Insight, Intel VTune, and likwid, and debugging tools such as TotalView and DDT. Performance monitoring and tracing tools including Intel® Trace Analyzer, Scalasca, Score-P, Vampir, TAU, and Paraver, as well as monitoring tools like Ganglia, Nagios, and Prometheus are also covered. Students will create custom monitoring scripts and dashboards for HPC systems and develop optimisation strategies based on insights from performance tools, focusing on code restructuring, load balancing, and data management. Performance-driven decision-making for hardware upgrades or configurations is also addressed. Practical exercises involve hands-on experience with profiling tools to analyse compute and memory performance, using performance analysis tools to determine the performance of parallel applications, and detecting overhead and bottlenecks in sample parallel applications.

Performance modelling It covers the principles of performance models, comparing techniques for validating models against existing implementations, and calibrating models to ensure accuracy and reliability. It also includes performance modelling for software applications such as web applications, databases, and parallel programs, and involves analysing workload patterns, response time, and throughput to identify performance bottlenecks using models. Students will apply performance modelling at the node and system levels, including for distributed systems, cloud services, and HPC clusters. It also covers analysing system-level performance metrics and interactions, as well as scaling and load balancing in distributed systems. Performance models will be used to identify optimisation strategies, and students will apply model-driven optimisation techniques to improve system performance and perform cost-benefit analysis of optimisation decisions. Practical exercises include creating basic performance models, writing case studies of model validation, and predicting bottlenecks in parallel application samples.

Performance optimisation For optimisation of a provided implementation on a specific platform, one first needs to be able to compare the architectures of modern supercomputers and understand single-core architecture and optimisation strategies. Memory hierarchy and data access optimisations, efficient shared memory parallelization, and an evaluation of different parallelization approaches for multi-core processors, including GPUs, are necessary. It also covers efficient distributed memory parallelization, advanced performance models, and the comparison of serial and parallel performance modelling. Additionally, students will evaluate energy-efficient implementation and execution of parallel programs. Practical exercises involve implementing basic numerical methods with high hardware efficiency on parallel computers and gaining insight into innovative programming techniques and alternative supercomputer architectures. This also includes advanced techniques like domain-specific languages and code generation.

Project The prerequisites of this (group) project are basic performance analysis and knowledge of performance models and computer architecture fundamentals. It usually involves profiling and optimizing codes on HPC systems or industry-related projects in the area of performance engineering.

Shared modules Performance engineering also requires insight into computer architecture and computational methods. Thus, this specialisation shares several optional subjects with the specialisations "System Architect" (Heterogeneous Systems and Accelerators, HPC System Design, Design Tools and Simulators, High-Level Digital Design), "System Development and Support" (Advanced Operating Systems, Compiler Design, Parallel Programming Models), and "Numerical and Data Specialist for Science Domains" (Domain Specific Programming).

4.2.3. System development and support

This specialisation targets graduate experts in the development of software for supercomputers, including the programming model, compiler, operating system, performance analysis tools, and middleware technologies. More specifically, we selected the following skills: the design and implementation of programming models, compilers, compiler optimisations, toolchain, software stacks (for CPU, GPU, FPGA, CRGA), performance analysis tools, operating systems, kernel development, parallel file systems, high-speed networking, synchronization, container technologies, virtualisation technologies, integration of HPC and cloud,

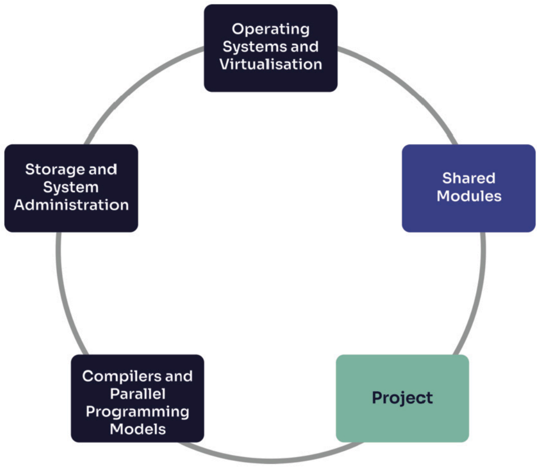

Fig. 6. Proposed modules (study areas) within System Development and Support.

Table 5
Proposed modules (study areas) within System Development and Support.

Area	Modules	ECTS
Operating Systems and Virtualisation	2	6
Storage and System Administration	2	6
Compilers and Parallel Programming Models	2	6
Project	1	9
Shared Modules	10	30
In Total	**17**	**57**

server administration, infrastructure setup and management, and security.

Fig. 6 and Table 5 show an overview of the areas targeted and the expected ECTS within this specialisation. A detailed description of each area follows.

Operating systems and virtualisation This area includes Advanced Operating Systems and Virtualisation and Containerization Support. They provide the students with the required design and implementation alternatives around scheduling policies, memory management, parallel I/O, device drivers, and support for virtualised environments, in order for them to design new approaches for supporting these and new features on upcoming systems. The practical side includes the development of OS drivers, and evaluating the support for virtualised environments.

Storage and system administration This area covers the aspects of System Administration on HPC Environments, including protection and security aspects, application schedulers and resource managers (currently Slurm...), installation of applications, development tools, development of plans for system maintenance, user management and physical integration of the components of a supercomputer. Also, the Storage, Parallel File Systems and Databases include the design, implementation and management of parallel file systems (currently GPFS, Lustre, Gluster, etc.), the use of advanced storage devices, and parallel high-performance databases. Practical includes installation of the OS itself, and configuration of services taking into account the protection and security requirements, as well as the design and implementation of parts of a parallel filesystem.

Compilers and parallel programming models This area covers the design and implementation of Parallel Programming Models and Compilers for HPC environments. It includes the design of parallel languages and models for programming HPC systems, design of high-level abstractions for expressing parallelism (currently MPI, OpenMP, Kokkos, Raja, etc.), task-based parallel programming, interoperability between models, compiler design for HPC and heterogeneous systems, and implementation of compiler optimisations. Practical covers the implementation of compiler passes for optimisations, code generation, and language and programming model extensions to improve expressiveness or performance portability.

Project In this module, students work in groups with the goal of providing a solution to a particular problem related to the specialisation areas.

Shared modules Additionally, this specialisation shares several optional subjects with the specialisations “System Architect” and “Performance Analyst and Advisor”. From “System Architect” the modules on “Design Tools and Simulators”, “Processor Design”, and “Multiprocessor Design”. From “Performance Analyst and Advisor”, the modules on “Tool-based Performance Engineering” and “Model-based Performance Engineering”.

4.2.4. *System architect*

This specialisation targets graduate experts in the design and development of supercomputers. Focus on processor, multiprocessor, supercomputer architectures, memory and I/O systems, networking, circuit design, verification and test, low-power techniques, and fabrication. The next subsections describe the learning outcomes of each identified study area.

Fig. 7 and Table 6 show an overview of the areas targeted and the expected ECTS within this specialisation. A detailed description of each area follows.

System and hardware technologies This study area includes contents related to HPC system design, hardware technologies and sustainable nanoelectronic design. In detail, HPC system design groups the content around HPC facilities (e.g. power delivery, cooling solutions, use of resources), and HPC system components (e.g. CPUs, accelerators, memories, networking, racks, blades). In the area of hardware technologies, we included the presentation of the underlying silicon and photonics technologies (MOS, CMOS, memories, photonics), Moore’s law and Dennard scaling, power wall and dark silicon, system reliability (soft errors, hard errors, solutions) and Security (Root of trust, encryption primitives). Finally, in the Sustainable nanoelectronic design, we include the principles of sustainability in integrated circuits, sources of power dissipation, low-power techniques, lifetime analysis and optimisation, CO2 footprint; and the societal and economic impact.

CPUs and heterogeneous systems This study area includes content related to processor design, multiprocessor design, high-level digital design, and heterogeneous systems and accelerators. Specifically, processor design includes CPU pipelining, memory hierarchy, caches, branch prediction, reservation stations, reorder buffer, load-store queue and hardware performance counters. Multiprocessor Design focuses on the logic needed for data coherency and consistency, the on-chip network and the memory hierarchy. Then, high-level digital design is devoted to the principles of digital systems based on programmable and configurable components, the system breakdown into hardware and software components and the design of processing subsystems (e.g., video, vector and matrix operations, sorting algorithms). Finally, heterogeneous systems and accelerators include single chip, multiple chip (chiplet) and

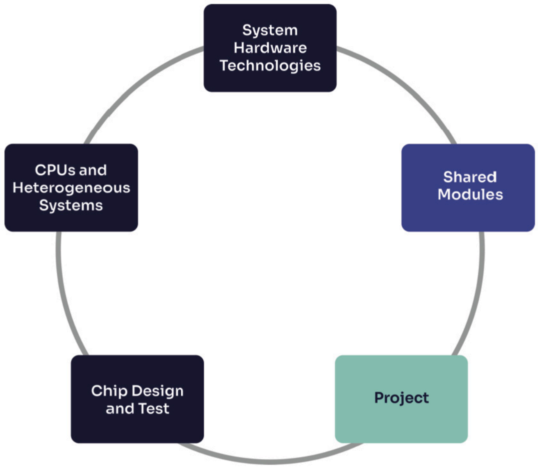

Fig. 7. Proposed modules (study areas) within System Architect.

Table 6
Proposed modules (study areas) within System Architect.

Area	Modules	ECTS
System and Hardware Technologies	3	9
CPUs and Heterogeneous Systems	4	12
Chip Design and Test	3	9
Project	1	9
Shared Modules	5	15
In Total	**16**	**54**

multiple board architectures and accelerators (e.g. Vector unit, GPU, FPGA, TPUs, and stencils).

Chip design and test This study area groups the content related to Physical Design and Test and the related tools. In detail, System-on-Chip Physical Design includes the timing and power constraints to a complex integrated circuit, the physical implementation of a complex integrated circuit, low power design techniques, design, analysis and evaluation of electronic systems in applications such as automation, energy distribution and generation, consumer electronics, biomedicine, etc. In the module dedicated to the testing of digital designs, we include fault models, fault coverage, observability, automatic test pattern generators, scan-based tests and self-tests. Finally, in the area of Design Tools and Simulators, students will be presented and use HDLs (e.g., SystemC, SystemVerilog) to describe systems, apply the VLSI design flow to hardware design and use physical design tools (partitioning, chip planning, placement, global routing, detailed routing).

Project In this module, we expect students to select a certain topic related to ongoing research projects, running compute-time projects on HPC systems, industry-related projects in the area of system architecture, or performing hands-on system development, modelling or optimisation.

Shared modules Additionally, this specialisation shares several optional subjects with the specialisations "System Development and Support" and "Performance Analyst and Advisor". From "System Development and Support", the modules on Advanced Operating Systems, Storage, Parallel File Systems and Databases and Compiler Design are highly interesting for this specialisation together with the Tool-Based Performance Analysis in the "Performance Analyst and Advisor" specialisation.

4.3. Transversal skills

The EUMaster4HPC curriculum is designed to include in addition to fundamental HPC skills also transversal skills. These transversal skills are essential for students to succeed in their careers and to prepare them effectively for the evolving job market. We adopt the following definition of transversal skills by the UNESCO International Bureau of Education: "Skills that are typically considered as not specifically related to a particular job, task, academic discipline or area of knowledge and that can be used in a wide variety of situations and work settings (for example, organisational skills)" [17]. A subset of these skills involves soft skills (sometimes also called professional skills): "Term used to indicate a set of intangible personal qualities, traits, attributes, habits and attitudes that can be used in many different types of jobs".

Nowadays, many top technology companies test applicants on their soft skills during the interview process. Taking into account this attitude, a set of advanced skills is proposed within the curriculum to train students to address issues such as diversity, inclusion, gender, and unconscious bias.

Based on the method described in Section 3, six groups of skills have been identified, which are listed below. For each of these groups, modules have been defined in terms of content topics:

- Research Techniques: scientific methods, critical thinking, curiosity, artificial intelligence in higher education.
- Scientific Writing: writing papers, publishing scientific results, scientific publications databases, technical presentations, and proposal writing.
- Work Environment: ethics and integrity, team management, human resources, relational skills (communications), priority management, gender, diversity.
- Entrepreneurship: business organization and management, business plan development, market viability, financial management, (public) procurement procedures, intellectual property, and identifying customer needs.
- Legal Aspects: licensing, open data, intellectual property, data protection and privacy (GDPR), EU regulations in information technology.
- Other: attention to detail and aesthetics, curiosity, willingness to raise questions and to rely on the expertise of other people, ambition to acquire more skills and willingness to continuously learn new things.

4.4. Balancing modules

Most of the time, high-performance computing is misinterpreted as related to pure computer science. However, this is not true; from the early stages to the present day, computers have been used for doing arithmetic computations using large volumes of data sets. For example, solving a large number of systems of equations, AI computations using large volumes of data sets, material science and biomedical simulation-based specific mathematical modelling and computer graphics.

All of those examples require broad knowledge of physics, mathematics, engineering, and computer science skills. Therefore, to become a highly qualified person in high-performance computing, one must have more than intermediate knowledge in physics, mathematics, and engineering.

Therefore, it is essential that we encourage students with backgrounds in physics, engineering, and mathematics to study master's in HPC. However, those who have bachelor's degrees in engineering, physics, and mathematics may lack knowledge of pure computer science.

To avoid these problems, we have introduced the balancing module. Although most of the bachelor's studies would consist of our proposed balancing model modules (courses), see Fig. 2 and Table 1. But in any case, if students lack it, they can take those courses from the balancing module and get ready to follow up on the master's in HPC from year one.

`Scenario 1`: If a student comes from a pure physics background, then the student should be taking courses in parallel programming, software engineering, and computer architecture, considering that they lack those courses or knowledge within their bachelor's studies.

`Scenario 2`: If a student comes from pure IT engineering, then the student must take courses in mathematics and statistics or any other courses within the balancing modules if they lack knowledge or courses within their bachelor's studies.

5. Implementation

5.1. Students study journey

We suggest that students study at (at least) two universities during their two years of master's in HPC. For most of them, they will spend their first year at their existing local (home) university, and the second year at another (host university). Due to the additional living costs implied by this policy, students of the EUMaster4HPC get a dedicated grant for their mobility.

5.2. Internships

Students of EUMaster4HPC are strongly encouraged to participate in internships. In some awarding universities, an internship during the second semester of the second year is already mandatory and part of the curriculum; it is also strongly linked with the master thesis and is supervised by both a local academic and a host company staff. In other words, students have some more flexibility; either they can continue and convert the internship into a master thesis, or they can do it within a shorter 2-4 weeks and do their master thesis within a university lab. Due to this difference in existing practices and the related differences in the number of offers available to students, the partners of EUMaster4HPC that have existing contacts with HPC industries and/or existing databases for internship offers to direct them to a task leader (currently, the CINI consortium) for pooling. The task leader is then responsible for the promotion of these offers.

5.3. Capstone project

The capstone project that concludes the student's study programme is a Master's thesis that adheres to the principles of accredited European universities (i.e. the Bologna process [22]). This project is intended to demonstrate the student's ability to engage in critical thinking, contribute knowledge, and demonstrate an advanced understanding of high-performance computing. It involves identifying and solving a specific problem using the knowledge and skills acquired during the EU-accredited coursework. With some guidance, the students showcase their ability to analyse issues and apply advanced computational tools and techniques learnt throughout the programme.

Students can and are encouraged to complete their master's thesis at their home university, a host (partnering double-degree) university, or an external partner from industry, supercomputing and research centres, or research laboratories. These external institutions, and the proposed thesis topic, must be recognised by the home and host university. Specific mechanisms must be established within the university network to facilitate the submission of project topics by external institutions and ensure the dissemination of this information to prospective students. The partnerships within the consortium for EUMaster4HPC serve as an example of how such mechanisms can be effectively implemented. These options provide students with three possible pathways to complete their Master's thesis:

i) **Home University Thesis**: Supervisor and examiner from the home university.
ii) **Host University Thesis**: Supervisor and examiner from the host university and supervisor from the home university.
iii) **External Thesis**: Supervisor and examiner from the host or host university and supervisor from external institution.

5.4. Academic activities

An effective study program should offer various activities and opportunities to improve student's learning and professional development in careers in HPC [5]. These activities are central in providing hands-on experience and real-world applications of theoretical knowledge, bridging the gap between academia and industry. In addition, these programs emphasise transferable skills such as presentation skills, working in groups, and applying theoretical knowledge to practical scenarios.

5.4.1. Summer schools

Summer schools aim to improve interpersonal skills and provide students with networking opportunities. They bring together students from related master's programs, professors, and stakeholders from HPC centres and companies to explore a range of HPC-related topics, from applied mathematics (numerical and data science) to computer science disciplines such as computer architecture and software. This diverse curriculum not only broadens the academic horizons of students but also strengthens their professional networks, promotes interdisciplinary learning, and facilitates future collaborations.

These summer schools are held during the university holidays to ensure that all students can attend. Examples of summer school events specifically tailored to enhance European Master's programs include:

- **EUMaster4HPC Summer School 2023 on HPC for Environmental Aspects**: Université Grenoble Alpes, Saint-Martin-d'Hères, France, July 16 to 28, 2023.
- **EUMaster4HPC Summer School 2024 on HPC in Data Science**: IT4Innovations National Supercomputing Center at VSB – Technical University of Ostrava and VSC Research Center at TU Wien, hosted by IT4Innovations, Ostrava, Czech Republic, August 18 to 24, 2024.[6]
- **CSCS-USI Summer University 2024 on Effective High Performance Computing and Data Analytics**: Swiss National Supercomputing Center (ETH Zurich / CSCS) and Università della Svizzera italiana (USI), Lugano, Switzerland, July 6 to 12, 2024.

5.4.2. Workshops

Workshops will be conducted at participating universities to foster a sense of community among students and highlight industrial and research needs in HPC and High-Performance Data Analytics. These events enhance the core curriculum, serving as essential platforms to facilitate student interactions and enrich their educational experiences. Students are required to attend at least three workshops in the middle of the first, second, and third semesters. During the workshops, students can present their interesting findings through their mini-coursework projects. In addition, they can meet and interact with HPC experts during the workshop.

5.4.3. Challenges

Student challenge events, which involve participants during their second semester, should typically occur after students have completed their foundational courses in parallel programming and mathematical modelling. These competitions are designed to test participants' skills in real-world scenarios. They focus on parallelisation and performance

[6] https://eumaster4hpc.uni.lu/summer-school-2024/.

analysis of numerical algorithms that should be implemented on parallel heterogeneous supercomputers. These challenges are hosted by leading supercomputing centres and are conducted virtually. For example, the 2024 EUMaster4HPC Student Challenge at EuroHPC Summit 2024 involved students working in teams to parallelise and optimise the "Conjugate Gradient iterative solver" on Luxembourg's National Supercomputer, MeluXina.

6. Case study: University of Luxembourg

Based on the recommendations from the EUMaster4HPC curriculum, various universities within the consortium have adopted different modules of this curriculum to enhance existing programs or developed new master's programs. Examples for the latter are the University of Luxembourg, Politecnico di Milano in Italy, and Sofia University in Bulgaria, which all have established a new master's program in High-Performance Computing (HPC) in alignment with the EUMaster4HPC curriculum. In this section, we report on selected challenges and solutions associated with launching the new master's program at the University of Luxembourg and outline the administrative and technical difficulties encountered during its implementation.

The master's program was established in the academic year of 2023 (September) and is open not only to students within the EUMaster4HPC network but also to other students. Notably, students enrolled in the EUMaster4HPC must complete one year of study at a partner university.

- For the academic year 2023-2024, the University of Luxembourg reported an enrolment of 27 students, comprising 13 local students and 14 EUMaster4HPC students.
- For the academic year 2024-2025, enrolment increased to 46 students, including 22 local students and 24 EUMaster4HPC students.
- Each semester, around 7-8 teaching staff were involved in delivering courses.

6.1. Administrative procedures

Similarly to many other universities in Europe, a new proposal for a master's program requires approval from the university's governing board, which comprises rectors and other board members. Several queries and concerns were raised, which are summarized as follows, along with proposed solutions:

- *What will happen to this new master's program after the EUMaster4HPC project concludes?*: The newly created master's program will be integrated into a stand-alone master's program or potentially merged with other Erasmus Mundus programs to ensure sustainability in the long term.
- *How will laboratory equipment be utilized for HPC exercises and simulations?*: Questions arose regarding the need for new hardware, such as advanced GPUs or multicore CPUs, and concerns about the cost-effectiveness of purchasing equipment that may become obsolete. As EUMaster4HPC is part of the EuroHPC Joint Undertaking (JU) initiative, students do have direct access to EuroHPC JU resources for their laboratory and simulation practicals. Being part of a network of universities opens the opportunity to also access facilities at other sites for educational purposes. For example, the Barcelona Supercomputing Center (BSC) offers extensive facilities, including courses and labs that require hardware specification-based labs. Thus, the proposed master's program can be considered to be cost-effective.
- *What about human resources for teaching activities related to new content?*: In Luxembourg, there is an ongoing development in HPC knowledge and skills, allowing the university to teach HPC-related courses through research scientists and postdoctoral researchers. For specialized courses, in the beginning, it hired external partners with expertise, such as in quantum computing, through external staff. However, at the time of writing this article, the University of Luxembourg has enough teaching staff to teach all the courses in HPC.

Table 7
Overview of MOOC Courses: course names, institutions, launch dates, and enrolment numbers.

MOOC Name	Institution	Launched	Enrolled
Cancer Genome Analysis Using SigProfiler Suite of Tools	Middle East Technical University	2023-Sep.	18
Introduction to Quantum Computing	Politecnico di Milano	2023-Sep.	17
Energy Aware Parallel Computing	AGH University of Krakow	2023-Sep.	19
Massive Parallel Programming on GPUs and Applications	Sorbonne University	2023-Sep.	19
Introduction to OpenACC, OpenMP Offloading, and HIP Programming Models	University of Luxembourg	2024-Dec.	4

6.2. Practical implementation

In many cases, it is common for courses from one specific master's program to overlap with those in another master's program, either as compulsory or optional courses. Prior to the establishment of the new master's program in Masters in High-Performance Computing (MHPC), the University of Luxembourg offered both a Master's in Data Science and a Master's in Computer Science. Consequently, the new master's program is designed to integrate a selection of courses from these existing programs. This section summarizes the methodologies employed to curate a comprehensive list of courses for the MHPC curriculum.

- Given that EUMaster4HPC proposed a modular curriculum, only a subset of modules needed to be implemented at the University of Luxembourg. Each course is designed to provide up to 3 ECTS credits, aligning with common practices at the University of Luxembourg. For example, FPGA programming and Introduction to Quantime Cpompiuting both have a 3 ECTS.[7]
- Given that the University of Luxembourg previously offered master's programs in Computer Science and Data Science, which included courses related to parallel programming, machine learning, and artificial intelligence, the alignment of these courses with the new master's program in High-Performance Computing (HPC) was straightforward. The university's focus on developing numerical and data-specialist competencies for scientific domains facilitated this integration.
- In instances where specific required courses were not found within the existing master's programs or other curricula, new courses and modules were developed to fulfil the recommendations from the EUMaster4HPC curriculum. Noteworthy additions include topics such as computer architectures, accelerator computing, and quantum computing.
  As part of the EUMaster4HPC initiative, a series of Massive Open Online Courses (MOOCs) are being developed. To date, five courses have been successfully created. They can either be used as supplementary modules or as stand-alone options tailored to meet the specific curricular needs of universities in fulfilling their master's program requirements. Since the launch of our MOOC initiative, there has been a consistent increase in participation rates. Table 7 provides information on the overall course being developed and the total number of participants who have participated so far. Suc-

[7] https://www.uni.lu/fstm-en/study-programs/master-in-high-performance-computing/programme/.

cessful completion of the courses provides opportunities to earn between 1 and 3 ECTS credits.

7. Students' feedback

The EUMaster4HPC has started educating students at different universities throughout Europe following curricula that are being adapted to the common European master reference curriculum in HPC, which has been documented in the previous sections of this paper. The first cohort started in autumn 2022 and finished in 2024. Two more cohorts started in 2023 and 2024. The students of these cohorts all benefited from a mobility program such that they could spend the first and second years at different universities and join other organisations for internships.

The students of these cohorts were invited to participate in a survey, to which 37 responses involving students from all cohorts were received. The selected results of this survey are documented in Table 8. We would like to highlight that this is feedback to a curriculum, which is still in an early stage.

The feedback received on questions related to the fundamentals and specialisations shows a strong appreciation of the curriculum. Both the figures as additionally collected comments indicate that at least some of the students struggled to have a full overview of the program. A clear majority of the students confirmed that the specialisations are aligned with their career goals.

A large majority of the students showed excitement about the mobility program. This figure is likely biased as these students applied for EUMaster4HPC knowing that mobility is mandatory.

By far most of the students rate the overall structure of the program as good or even excellent. This is consistent with the indicated willingness to recommend it to others.

In the final part of the survey, we asked the students about their future plans. 30-35% of them aim for a career in industry or aim for academic research. 24% of the students did not provide an answer, which probably means that they have no concrete plans, yet. The same fraction of students did not reply to the answer, how well the program prepares them for a career in HPC-related industry and academia, while more than 80% have the opinion that the program prepares them very or even extremely well for such a career.

8. Summary and conclusions

In this paper, we provided an overview of the pan-European reference curriculum for a master's in HPC that has been developed within the EUMaster4HPC project. The curriculum has been developed based on a collection and assessment of skills that various stakeholders consider relevant in the future. In the process of identifying skills, private companies were involved. A large collection of modules has been defined that provide flexibility for adaptation to existing programs and cope with the diverse skill profiles of HPC experts. The curriculum is being used to educate up to 150 students. A mobility program allows them to be educated and trained in various European universities as well as in HPC data centres and companies.

The method of designing and initial implementation of the curriculum has been a good way to involve private companies and HPC data centres in training future HPC experts. This can be expected to result in a higher visibility of the master in HPC educational programs, which increases both the career opportunities of the involved students as well as the attractiveness of this path of education. Ultimately, the targeted benefit is to increase the number of experts in the area of HPC with people who can drive further evolutions of HPC technologies and/or people who are able to leverage HPC for addressing the various grand challenges.

Table 8
EUMaster4HPC students' survey results.

Fundamentals	
How do you find the content of the fundamental courses in the first year?	
Very engaging and comprehensive	46%
Good but needs some improvement	38%
Average and needs more depth	3%
No answer	14%
How do you feel the first-year curriculum prepares you well for the second-year specialisation?	
Strongly agree	24%
Agree	49%
Neutral	11%
Disagree	3%
No answer	14%
Specialisations	
How satisfied are you with the range of specialization options available?	
Very satisfied	32%
Satisfied	32%
Neutral	19%
Dissatisfied	3%
No answer	14%
How clear are the pathways to each specialisation?	
Very clear	14%
Clear	38%
Somewhat clear	16%
Not clear	14%
No answer	19%
How do you feel the specialisation courses align with your career goals?	
Yes, completely	51%
Somewhat	22%
Neutral	8%
No answer	19%
Exchange abroad	
How excited are you about studying abroad in the second year?	
Very excited	54%
Excited	24%
Neutral	3%
No answer	19%
How do you rate the support provided for the exchange abroad?	
Excellent	14%
Good	41%
Average	22%
Poor	5%
No answer	19%
Program structure	
How would you rate the overall structure of the programme?	
Excellent	35%
Good	41%
Average	3%
No answer	22%
Would you recommend this Master's programme to others?	
Yes, definitely	57%
Yes, with some reservations	22%
No answer	22%
Post-graduation impact	
How well do you think the programme prepares you for a career in HPC-related industries or academia?	
Extremely well	19%
Very well	54%
Somewhat well	3%
No answer	24%
Which career path are you most likely to pursue after completing the programme?	
Industry roles in HPC or technology	35%
Research roles in academia or labs	32%
Entrepreneurship in HPC applications	8%
No answer	24%

CRediT authorship contribution statement

Pascal Bouvry: Conceptualization. **Mats Brorsson:** Conceptualization. **Ramon Canal:** Writing – original draft. **Aryan Eftekhari:** Writing – original draft, Conceptualization. **Siegfried Höfinger:** Writing – original draft, Conceptualization. **Didier Smets:** Writing – original draft, Conceptualization. **Harald Köstler:** Writing – original draft, Conceptualization. **Tomáš Kozubek:** Writing – original draft, Conceptualization. **Ezhilmathi Krishnasamy:** Conceptualization. **Josep Llosa:** Writing – original draft. **Alexandra Lukas-Rother:** Writing – review & editing, Conceptualization. **Xavier Martorell:** Writing – original draft, Conceptualization. **Dirk Pleiter:** Writing – original draft, Conceptualization. **Ana Proykova:** Writing – original draft. **Maria-Ribera Sancho:** Writing – review & editing. **Olaf Schenk:** Writing – original draft, Conceptualization. **Cristina Silvano:** Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: All authors report financial support provided by EuroHPC JU. Dirk Pleiter reports a relationship with KTH Royal Institute of Technology. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Funding for this work has been received from the EuroHPC JU under Grant Agreement 101051997 (EUMaster4HPC). We would like to thank Régis Beck (University of Luxembourg) for his help with the survey.

References

[1] A. Basden, M. Weinzierl, T. Weinzierl, B.J.N. Wylie, A novel performance analysis workshop series concept, developed at Durham University under the umbrella of the ExCALIBUR programme, https://doi.org/10.5281/zenodo.5155503, 2021.

[2] J. Chen, S. Ghafoor, J. Impagliazzo, Producing competent HPC graduates, Commun. ACM 65 (2022) 56–65, https://doi.org/10.1145/3538878.

[3] J. Chen, J. Impagliazzo, L. Shen, High-performance computing and engineering educational development and practice, in: 2020 IEEE Frontiers in Education Conference (FIE), 2020, pp. 1–8.

[4] A. Georgi, S. Höhlig, R. Geyer, W.E. Nagel, Linux cluster in theory and practice: a novel approach in teaching cluster computing based on the intel atom platform, in: Proceedings of the International Conference on Computational Science, ICCS 2011, Proc. Comput. Sci. 4 (2011) 1917–1926, https://doi.org/10.1016/j.procs.2011.04.209.

[5] S.L. Harrell, H.A. Nam, V.G.V. Larrea, K. Keville, D. Kamalic, Student cluster competition: a multi-disciplinary undergraduate HPC educational tool, in: Proceedings of the Workshop on Education for High-Performance Computing, Association for Computing Machinery, New York, NY, USA, 2015.

[6] E.e.a. Joseph, Hyperion research ISC24 market update, https://hyperionresearch.com/hpc-market-update-briefing-during-isc24-virtual/, 2024.

[7] L.N. Long, P.J. Morris, K. Morooney, S. Kellogg, The teaching and learning of high performance computing, J. Eng. Educ. 87 (1998) 591–597, https://doi.org/10.1002/j.2168-9830.1998.tb00397.x.

[8] L. Marzulo, C. Bianchini, L. Santiago, V. Ferreira, B. Goldstein, F. França, Teaching high performance computing through parallel programming marathons, in: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2019, pp. 296–303.

[9] L. Oden, K. Nölp, P. Brauner, Integrating interactive performance analysis in Jupyter notebooks for parallel programming education, in: 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2024, pp. 369–376.

[10] S.K. Prasad, Message from the EduPar-24 workshop chairs, in: 2024 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), IEEE Computer Society, Los Alamitos, CA, USA, 2024, p. 334.

[11] S.K. Prasad, T. Estrada, S. Ghafoor, A. Gupta, K. Kant, C. Stunkel, A. Sussman, R. Vaidyanathan, C. Weems, K. Agrawal, M. Barnas, D.W. Brown, R. Bryant, D.P. Bunde, C. Busch, D. Deb, E. Freudenthal, J. Jaja, M. Parashar, C. Phillips, B. Robey, A. Rosenberg, E. Saule, C. Shen, NSF/IEEE-TCPP curriculum initiative on parallel and distributed computing - core topics for undergraduates, version II-beta, http://tcpp.cs.gsu.edu/curriculum/, 2020.

[12] A. Qasem, H. Anzt, E. Ayguade, K. Cahill, R. Canal, J. Chan, E. Fosler-Lussier, F. Göbel, A. Jain, M. Koch, M. Kuzak, J. Llosa, R. Machiraju, X. Martorell, P. Nayak, S. Oottikkal, M. Ostasz, D.K. Panda, D. Pleiter, R. Ramnath, M.R. Sancho, A. Sclocco, A. Shafi, H. Spreeuw, H. Subramoni, K. Tomko, Lightning talks of EduHPC 2022, in: 2022 IEEE/ACM International Workshop on Education for High Performance Computing (EduHPC), 2022, pp. 42–49.

[13] A. Qasem, D.P. Bunde, P. Schielke, A module-based introduction to heterogeneous computing in core courses, J. Parallel Distrib. Comput. 158 (2021) 56–66, https://doi.org/10.1016/j.jpdc.2021.07.011.

[14] R.K. Raj, C.J. Romanowski, J. Impagliazzo, S.G. Aly, B.A. Becker, J. Chen, S. Ghafoor, N. Giacaman, S.I. Gordon, C. Izu, S. Rahimi, M.P. Robson, N. Thota, High performance computing education: current challenges and future directions, in: Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education, Association for Computing Machinery, New York, NY, USA, 2020, pp. 51–74.

[15] The Council of the European Union, Council Recommendation on key competences for lifelong learning, Off. J. Eur. Union 189 (2018) 1–13, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018H0604(01).

[16] The Council of the European Union, Council Recommendation on a European approach to micro-credentials for lifelong learning and employability, Off. J. Eur. Union C243 (2022) 10–25, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32022H0627(02).

[17] UNESCO International Bureau of Education, Glossary of curriculum terminology, https://unesdoc.unesco.org/ark:/48223/pf0000223059.locale=en, 2013.

[18] A.L. Varbanescu, S.N. Swatman, A. Pathania, Performance engineering for graduate students: a view from Amsterdam, in: Proceedings of the SC 2023 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis, Association for Computing Machinery, New York, NY, USA, 2023, pp. 357–365.

[19] S. Vargas-Pérez, Designing an independent study to create HPC learning experiences for undergraduates, in: 2022 IEEE 29th International Conference on High Performance Computing, Data and Analytics Workshop (HiPCW), 2022, pp. 6–11.

[20] N. Wedi, P. Bauer, I. Sandu, J. Hoffmann, S. Sheridan, R. Cereceda, T. Quintino, D. Thiemert, T. Geenen, Destination Earth: high-performance computing for weather and climate, Comput. Sci. Eng. 24 (2022) 29–37, https://doi.org/10.1109/MCSE.2023.3260519.

[21] J.M. Wing, D. Stanzione, Progress in computational thinking, and expanding the HPC community, Commun. ACM 59 (2016) 10–11, https://doi.org/10.1145/2933410.

[22] B. Wächter, The Bologna process: developments and prospects, Eur. J. Educ. 39 (2004) 265–273, http://www.jstor.org/stable/1503854.

[23] Z. Xu, Teaching heterogeneous and parallel computing with Google Colab and raspberry Pi clusters, in: Proceedings of the SC 2023 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis, Association for Computing Machinery, New York, NY, USA, 2023, pp. 308–313.

[24] A. Younts, S.L. Harrell, Teaching HPC systems administrators, J. Comput. Sci. Educ. 11 (2020) 100–105, https://doi.org/10.22369/issn.2153-4136/11/1/16.