EVITA
EuroHPC Virtual Training Academy

EVITA – EuroHPC Virtual Training Academy

Project Number: 101196394

EVITA CQF courses

Definition of EVITA CQF and its Integration into the European HPC Ecosystem

Disclaimer

This material should be regarded as a 'living tool' open for improvement and its content may be subject to modifications without notice. It has not yet undergone formal review by the JU and is shared for informational purposes only.

Funded by the European Union

EuroHPC Joint Undertaking

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the EuroHPC Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

Document Information

Document Name	EVITA CQF courses
Authors	Xavier Martorell (BSC), Victoria Döller (TUW), Claudia Blaas-Schenner (TUW), Kevin Lüdemann (GWDG)
Final Version Date	01/06/2026

Table of Contents

1. Introduction

In this report we provide the current list of courses that have been incorporated in the EVITA CQF. This framework provides a structured approach to organize master-level High-Performance Computing (HPC) courses, focusing on the competences provided and the qualification itineraries of interest for trainees, industry, and academia.

In this list, we include most of the courses already defined in the EUMaster4HPC project [1][2].

Attention: Not all courses are open for proposals in the first call. The unavailable courses are: PP.SMP3, PP.SMP4, PP.DMP1, PP.HS1, PP.HS3, PP.HS4, PP.MPI1, PP.MPI2, PP.MPI3, PP.PJC1, MAT1, MAT2, MAT3, MAT4, SDS.SSADM3, SDS.SSADM4, NDSSD.QC1, NDSSD.AI1, NDSSD.BDA1, NDSSD.BDA2, NDSSD.BDA3, NDSSD.BDA4, SE.SEP1, SE.SEP2, SE.SEP3, SE.SEP4, SE.SEP5, SE.SEP6. If a course is not available for proposals this is indicated in the title (not available for proposals).

2. List of courses – Parallel Programming

Area PP.SMP Shared-memory programming

PP.SMP1 EVITA course: Basic shared-memory parallel programming

PP.SMP1-SD1.2.2.1.1 EVITA module: **Shared-memory programming architecture**

Learning outcome: Describe shared-memory multiprocessor: CPU/cores/threads, memory hierarchy, basic I/O, storage, filesystems, directories, files, special files

Skills: SD1.2.2.1.1 Shared Memory architecture

Estimated commitment: 4 hours

Prerequisites: Sequential programming

PP.SMP1-SD1.2.2.1.8 EVITA module: **Security issues related to shared-memory environments**

Learning outcome: Describe the security issues related to shared-memory environments

Skills: SD1.2.2.1.8 Security considerations (new Skill needed)

Estimated commitment: 1 hour

Prerequisites: Sequential programming, shared-memory programming environment

PP.SMP1-SD1.2.1 EVITA module: **Structure of parallel algorithms**

Learning outcome: Describe structured parallelism (fork-join, worksharing), unstructured parallelism (tasking), and vectorization

Learning outcome: Describe the use of POSIX threads and/or OpenMP for shared-memory programming

Skills: SD1.2.1 Parallel Algorithms, SD1.2.2.1.7 Sample implementations of POSIX Threads and OpenMP

Estimated commitment: 4 hours

Prerequisites: Sequential programming, Shared-memory programming environment

PP.SMP1-SD1.2.2.1.2 EVITA module: **Synchronization constructs**

Learning outcome: Describe the various syncronization constructs (spin-lock, mutex, semaphones)

Skills: SD1.2.2.1.2 Synchronization

Estimated commitment: 2 hours

Prerequisites: Shared-memory programming environment, Structure of parallel algorithms

PP.SMP1-SD1.2.2.2 EVITA module: **Shared-memory applications using Posix Threads**

Learning outcome: Use POSIX threads to program sample applications

Skills: SD1.2.2.2 Parallelism with threading

Estimated commitment: 4 hours

Prerequisites: Shared-memory programming environment, Structure of parallel algorithms, Synchronization constructs, Use of synchronization constructs

PP.SMP1-SD1.2.2.4 EVITA module: **Shared-memory applications using OpenMP**

Learning outcome: Use OpenMP to program sample applications

Skills: SD1.2.2.4 Parallelism with compiler options - OpenMP

Estimated commitment: 4 hours

Prerequisites: Shared-memory programming environment, Structure of parallel algorithms, Synchronization constructs, Use of synchronization constructs

PP.SMP2 EVITA course: Advanced shared-memory parallel programming

PP.SMP2-SD1.2.2.1.6 EVITA module: **Compilers and code generation**

Learning outcome: Apply compiler optimizations to particular applications/benchmarks

Learning outcome: Combine automatic parallelization by the compiler with manual parallelization

Learning outcome: Combine vectorization and parallelization

Learning outcome: Debugging parallel applications in shared-memory environments

Skills: SD1.2.2.1.6

Estimated commitment: 4 hours

Prerequisites: Shared-memory programming environment, Synchronization constructs

PP.SMP2-SD1.2.2.5.1 EVITA module: **Parallelism with language extensions** - SYCL

Learning outcome: Use language extensions (e.g. SYCL) to parallelize applications

Learning outcome: Program applications with SYCL

Skills: SD1.2.2.5.1

Estimated commitment: 4 hours

Prerequisites: Shared-memory programming environment, Structure of parallel algorithms, Synchronization constructs, Compilers and code generation

PP.SMP2-SD1.2.2.5.2 EVITA module: **Parallelism with language extensions** - OpenACC

Learning outcome: Use language extensions (e.g. OpenACC) to parallelize applications

Learning outcome: Program applications with OpenACC

Skills: SD1.2.2.5.2

Estimated commitment: 4 hours

Prerequisites: Shared-memory programming environment, Structure of parallel algorithms, Synchronization constructs, Compilers and code generation

PP.SMP3 EVITA course: Introduction into Shared Memory Programming Using OpenMP **(not available for proposals)**

PP.SMP3-SD1.2.2.4 EVITA module: **Shared Memory Programming Using OpenMP**

Learning outcome: Understand the shared memory architecture - Unified Memory Access (UMA) and Non-Uniform Memory Access (NUMA), and hybrid distributed shared memory architecture

Learning outcome: Implementation of OpenMP programming model including parallel regions, environment routines, data sharing

Learning outcome: Efficiently handle OpenMP constructs such as work-sharing, synchronization constructs, Single Instruction Multiple Data (SIMD) directives

Learning outcome: Apply OpenMP programming knowledge to parallelize examples from science and engineering

Skills: SD1.2.2.4

Estimated commitment: 4 hours

Prerequisites: experience with C/C++ and/or FORTRAN

PP.SMP4 EVITA course: Accelerator-based parallel programming **(not available for proposals)**

PP.SMP4-SD1.2.8 EVITA module: **GPU Programming Using OpenACC**

Learning outcome: Understand GPU Architecture

Learning outcome: Implement the OpenACC Programming Model

Learning outcome: Handle Memory Management

Learning outcome: Apply OpenACC Programming Knowledge to accelerate examples from science and engineering

Skills: SD1.2.8

Estimated commitment: 4 hours

Prerequisites: experience with C/C++ and/or FORTRAN, OpenMP programming model

Area PP.DMP Distributed-memory programming

PP.DMP1 EVITA course: Distributed-memory parallel programming **(not available for proposals)**

PP.DMP1-SD1.2.3.1 EVITA module: **Distributed system architecture**

Learning outcome: Describe the basic architecture of the cluster, grid and cloud environments, including host nodes, processors, memory, storage, devices, synchronization and communication alternatives, compiler toolchains and debugging techniques

Learning outcome: Describe security issues related to distributed systems

Skills: SD1.2.3.1 Distribute memory

Estimated commitment: 4 hours

Prerequisites: Shared-memory programming environment, Synchronization constructs

PP.DMP1-SD1.2.3.2 EVITA module: **Sample Programming with message passing (e.g. MPI)**

Learning outcome: Use library-based communication programming (e.g. MPI) to parallelize applications

Learning outcome: Use appropriate communication primitives (point-to-point, collective communication, one-sided)

Learning outcome: Program sample distributed-memory applications

Skills: SD1.2.3.2

Estimated commitment: 4 hours

Prerequisites: Distributed system environments

PP.DMP1-SD1.2.3.3 EVITA module: **Resource management in distributed systems**

Learning outcome: Use proper resource allocation and management in MPI applications

Learning outcome: Use workload scheduling tools (e.g., SLURM)

Skills: SD1.2.3.3

Estimated commitment: 4 hours

Prerequisites: Distributed system environments, Programming with message passing (e.g. MPI)

PP.DMP2 EVITA course: Advanced distributed-memory parallel programming

PP.DMP2-SD8.3 EVITA module: **Work-flow applications concepts**

Learning outcome: Explain the design of applications based on work-flows

Learning outcome: Use work-flows to implement parallel applications

Skills: SD8.3 Workflow management systems principles

Estimated commitment: 4 hours

Prerequisites: Distributed system environments

PP.DMP2-SD8.4 EVITA module: **Pipeline applications design principles**

Learning outcome: Explain the design of applications based on pipelines

Learning outcome: Use pipelines to implement parallel applications

Skills: SD8.4

Estimated commitment: 4 hours

Prerequisites: Distributed system environments

PP.DMP2-SD8.5 EVITA module: **Microservices applications**

Learning outcome: Explain the design of applications based on microservices

Learning outcome: Use microservices to implement parallel applications

Skills: SD8.5

Estimated commitment: 4 hours

Prerequisites: Distributed system environments

Area PP.HS Heterogenous systems

PP.HS1 EVITA course: Heterogeneous system environments **(not available for proposals)**

PP.HS1-K1.5.3 EVITA module: **Heterogeneous systems** concepts and comparisons

Learning outcome: Describe the structure of heterogeneous systems

Learning outcome: Describe security issues related to heterogeneous systems

Skills: K1.5.3

Estimated commitment: 4 hours

Prerequisites: Shared-memory programming environment, Synchronization constructs...

PP.HS1-SD1.2.7.1 EVITA module: **Toolchains for heterogeneous systems**

Learning outcome: Use toolchains supporting CPU and accelerators (e.g., OpenMP target, OpenACC, CUDA, OpenCL, SYCL)

Learning outcome: Program small applications with heterogeneous programming models (e.g., OpenMP target, OpenACC, CUDA, OpenCL, SYCL)

Skills: SD1.2.7.1

Estimated commitment: 4 hours

Prerequisites: Heterogeneous systems

PP.HS1-K1.5.4 EVITA module: **Memory management in heterogeneous systems**

Learning outcome: Use heterogeneous memories (global, shared) in heterogeneous systems

Learning outcome: Evaluate and improve data transfers between host and accelerators

Skills: K1.5.4

Estimated commitment: 4 hours

Prerequisites: Heterogeneous Systems, Toolchains for Heterogeneous Systems

PP.HS2 EVITA course: Heterogeneous programming

PP.HS2-SD1.2.7.2 EVITA module: **Porting applications to Heterogeneous environments**

Learning outcome: Use manual code and data offloading for application acceleration (e.g., CUDA, OpenCL)

Learning outcome: Use guided code and data offloading for application acceleration (e.g., OpenMP target, OpenACC)

Learning outcome: Use language extensions for code and data offloading for application acceleration (e.g., SYCL)

Skills: SD1.2.7.2

Estimated commitment: 4 hours

Prerequisites: Heterogeneous systems, Toolchains for heterogeneous systems

PP.HS2-SD1.2.7.3 EVITA module: **Porting applications to FPGA/CGRA environments**

Learning outcome: Describe the architecture of a system including FPGAs/CGRAs

Learning outcome: Program applications using high-level synthesis

Skills: SD1.2.7.3

Estimated commitment: 4 hours

Prerequisites: Heterogeneous systems, Toolchains for heterogeneous systems

PP.HS2-SD1.2.7.4 EVITA module: **Resource management/Scheduling in heterogeneous systems**

Learning outcome: Use a queueing system to distribute resources among applications

Learning outcome: Use different scheduling policies

Skills: SD1.2.7.4

Estimated commitment: 4 hours

Prerequisites: Heterogeneous systems, Toolchains for heterogeneous systems

PP.HS3 EVITA course: How to use Neuromorphic Computing **(not available for proposals)**

PP.HS3-K1.5.2 EVITA module: **Neuromorphic Computing - Introduction to SpiNNaker**

Learning outcome: Learn what neuromorphic computing is, and what sort of problems it can solve

Learning outcome: Get to know different hardware systems used in neuromorphic computing

Learning outcome: Gain access to the novel neuromorphic computing platform SpiNNaker-2, and learn how to perform basic neuromorphic simulations with it

Skills: K1.5.2

Estimated commitment: 4 hours

Prerequisites: Basic computer architecture, Distributed system architecture

PP.HS4 EVITA course: How to understand and develop Neuromorphic Architectures **(not available for proposals)**

PP.HS4-K1.5.5 EVITA module: **Neuromorphics architectures**

Learning outcome: Understand the differences between neuromorphic architectures and comon architecture such as x86 or GPUs.

Learning outcome: Design a neuromorphic circuit.

Skills: K1.5.5

Estimated commitment: 4 hours

Prerequisites: Basic computer architecture, Distributed system architecture

Area PP.OMP OpenMP

Several of those previous EVITA modules on parallel programming could be further refined using more specific modules and learning outcomes in order to provide additional details on the teaching content and methodology. For example, the EVITA module on "Parallelism with compiler directives" could be based on the following modules when their implementation is based on the OpenMP programming model:

PP.OMP1 EVITA course: Programming shared-memory systems with OpenMP

PP.OMP1-SD1.2.2.4.2 EVITA module: **Introduction to the OpenMP programming model**

Learning outcome: Describe the OpenMP context, and how it is useful and productive

Learning outcome: Describe the OpenMP execution and memory models

Skills: SD1.2.2.4.2

Estimated commitment: 1 hours

Prerequisites: Shared-memory programming environment

PP.OMP1-SD1.2.2.4.3 EVITA module: **The OpenMP fork-join model**

Learning outcome: Use the OpenMP parallel construct

Learning outcome: Use the OpenMP data environment (private, firstprivate, shared)

Learning outcome: Use the OpenMP synchronization constructs (critical, atomic, masked)

Skills: SD1.2.2.4.3

Estimated commitment: 3 hours

Prerequisites: Shared-memory programming environment, Structure of parallel algorithms, Synchronization constructs, Compilers and code generation

PP.OMP1-SD1.2.2.4.4 EVITA module: **The OpenMP worksharing constructs**

Learning outcome: Use the OpenMP worksharing for construct

Learning outcome: Use the OpenMP worksharing sections construct

Learning outcome: Use the OpenMP worksharing single construct

Skills: SD1.2.2.4.4

Estimated commitment: 2 hours

Prerequisites: The OpenMP fork-join model

PP.OMP1-SD1.2.2.4.5 EVITA module: **The OpenMP tasking model and dataflow programming**

Learning outcome: Describe the OpenMP tasking execution model

Learning outcome: Use OpenMP tasking data environment features

Learning outcome: Use different OpenMP tasking scheduling clauses

Learning outcome: Use OpenMP tasking with dependences to program dataflow applications

Skills: SD1.2.2.4.5

Estimated commitment: 2 hours

Prerequisites: The OpenMP fork-join model

PP.OMP1-SD1.2.2.4.6 EVITA module: **OpenMP interoperability with other programming models**

Learning outcome: Describe the interop construct

Learning outcome: Use MPI+OpenMP interoperability to implement parallel applications

Skills: SD1.2.2.4.6

Estimated commitment: 2 hours

Prerequisites: The OpenMP fork-join model, The OpenMP worksharing constructs, The OpenMP tasking model and dataflow programming, Programming with message-passing.

PP.OMP2 EVITA course: Programming heterogeneous systems with OpenMP

PP.OMP2-SD1.2.2.4.7 EVITA module: **The OpenMP heterogeneous environments support**

Learning outcome: Describe the OpenMP heterogeneous execution environment

Learning outcome: Use the OpenMP target construct to support heterogeneous applications

Skills: SD1.2.2.4.7

Estimated commitment: 4 hours

Prerequisites: The OpenMP tasking model and dataflow programming, Heterogeneous systems, Memory management in heterogeneous systems

Area PP.MPI MPI

PP.MPI1 EVITA course: Programming distributed-memory systems with MPI - Basics **(not available for proposals)**

PP.MPI1-SD1.2.3.1 EVITA module: **Introduction to the Message Passing Programming (MPI) Model**

Learning outcome: Describe the Message Passing Programming (MPI) model and explain its main advantages and shortcomings

Learning outcome: Describe the basic constructs of MPI (point-to-point, collective,

blocking, nonblocking communication)

Skills: SD1.2.3.1

Estimated commitment: 1 hour

Prerequisites: Distributed system environments, Resource management in distributed systems, Basic programming skills in either C/C++ or Fortran or Python

PP.MPI1-SD1.2.3.2 EVITA module: **MPI process model and language bindings**

Learning outcome: Use the MPI process model

Learning outcome: Compile and execute MPI programs

Skills: SD1.2.3.2

Estimated commitment: 1 hour

Prerequisites: Introduction to the Message Passing Programming (MPI) Model

PP.MPI1-SD1.2.3.3 EVITA module: **Messages and point-to-point communication in MPI**

Learning outcome: Use blocking point-to-point communication in MPI

Skills: SD1.2.3.3

Estimated commitment: 2 hours

Prerequisites: MPI process model and language bindings

PP.MPI1-SD1.2.3.4 EVITA module: **Nonblocking communication in MPI**

Learning outcome: Use non-blocking point-to-point communication in MPI

Learning outcome: Detect and correct common errors with non-blocking point-to-point communication in existing MPI programs

Skills: SD1.2.3.4

Estimated commitment: 2 hours

Prerequisites: Messages and point-to-point communication in MPI

PP.MPI1-SD1.2.3.5 EVITA module: **Collective communication in MPI**

Learning outcome: Use collective communication in MPI

Learning outcome: Detect missed optimization opportunities in existing MPI programs and update these codes with appropriate collective MPI routines

Skills: SD1.2.3.5

Estimated commitment: 2 hours

Prerequisites: MPI process model and language bindings

PP.MPI1-SD1.2.3.6 EVITA module: **Intermediate and advanced topics in MPI**

Learning outcome: Recognize intermediate and advanced topics in MPI (one-sided and shared memory one-sided communication, derived datatypes, groups, communicators, virtual topologies, MPI I/O, MPI+X)

Learning outcome: Describe why the basic MPI features are not enough and what is missing to write real-world MPI applications

Learning outcome: Explain the implicit synchronisation between MPI processes

Skills: SD1.2.3.6

Estimated commitment: 1 hour

Prerequisites: Nonblocking communication in MPI, Collective communication in MPI

PP.MPI1-SD1.2.3.7 EVITA module: **Fortran and MPI**

Learning outcome: Explain the special requirements of using MPI in Fortran

Learning outcome: Adapt Fortran MPI programs to use the newest Fortran support methods in MPI

Skills: SD1.2.3.7

Estimated commitment: 1 hour

Prerequisites: Nonblocking communication in MPI

PP.MPI2 EVITA course: Programming distributed-memory systems with MPI - Intermediate **(not available for proposals)**

PP.MPI2-SD1.2.3.1 EVITA module: **Recap MPI basics, overview of intermediate and advanced topics in MPI**

Learning outcome: Describe the Message Passing Programming (MPI) model and explain its main advantages and shortcoming

Learning outcome: Describe the essential concepts of MPI

Learning outcome: Explain the implicit synchronisation between MPI processes

Skills: SD1.2.3.1

Estimated commitment: 1 hour

Prerequisites: EVITA course: Programming distributed-memory systems with MPI – Basics

PP.MPI2-SD1.2.3.8 EVITA module: **Groups & communicators in MPI**

Learning outcome: Use groups and communicators in MPI, build new groups and communicators in MPI.

Skills: SD1.2.3.8

Estimated commitment: 2 hours

Prerequisites: EVITA course: Programming distributed-memory systems with MPI – Basics

PP.MPI2-SD1.2.3.9 EVITA module: **Virtual topologies in MPI**

Learning outcome: Understand virtual topologies in MPI

Learning outcome: Use virtual cartesian topologies in MPI

Skills: SD1.2.3.9

Estimated commitment: 2 hours

Prerequisites: Groups & communicators in MPI

PP.MPI2-SD1.2.3.10 EVITA module: **Derived datatypes in MPI**

Learning outcome: Use derived datatypes in MPI

Skills: SD1.2.3.10

Estimated commitment: 2 hours

Prerequisites: EVITA course: Programming distributed-memory systems with MPI – Basics

PP.MPI2-SD1.2.3.11 EVITA module: **Halo communication in MPI**

Learning outcome: Code and optimize halo communication in MPI

Skills: SD1.2.3.11

Estimated commitment: 3 hours

Prerequisites: Derived datatypes in MPI, Virtual topologies in MPI

PP.MPI3 EVITA course: Programming distributed-memory systems with MPI - Advanced **(not available for proposals)**

PP.MPI3-SD1.2.3.6 EVITA module: **Recap MPI basics and intermediate, overview of advanced topics in MPI**

Learning outcome: Describe the Message Passing Programming (MPI) model and explain its main advantages and shortcomings

Learning outcome: Describe the essential concepts of MPI

Learning outcome: Explain the implicit synchronisation between MPI processes

Skills: SD1.2.3.6

Estimated commitment: 1 hour

Prerequisites: EVITA course: Programming distributed-memory systems with MPI – Intermediate

PP.MPI3-SD1.2.3.12 EVITA module: **One-sided communication in MPI**

Learning outcome: Use one-sided communication in MPI

Skills: SD1.2.3.12

Estimated commitment: 2 hours

Prerequisites: EVITA course: Programming distributed-memory systems with MPI – Basics

PP.MPI3-SD1.2.3.13 EVITA module: **Shared memory one-sided communication in MPI**

Learning outcome: Use shared memory one-sided communication in MPI

Skills: SD1.2.3.13

Estimated commitment: 2 hours

Prerequisites: One-sided communication in MPI

PP.MPI3-SD1.2.3.14 EVITA module: **Introduction to hybrid programming - MPI+X**

Learning outcome: Describe how to combine MPI with other programming models, especially how to combine MPI with OpenMP

Skills: SD1.2.3.14

Estimated commitment: 2 hours

Prerequisites: EVITA course: Programming distributed-memory systems with MPI – Intermediate

PP.MPI3-SD1.2.3.15 EVITA module: **MPI I/O**

Learning outcome: Explain MPI I/O and its shortcomings

Learning outcome: Use MPI I/O

Skills: SD1.2.3.15, K1.3.1.4.3

Estimated commitment: 2 hours

Prerequisites: EVITA course: Programming distributed-memory systems with MPI – Intermediate

PP.MPI3-SD1.2.3.16 EVITA module: **What's next - new features of the latest MPI standard (currently MPI 5.0)**

Learning outcome: Describe the newest features of MPI (features that entered the MPI standard with MPI 4.0 und MPI 5.0)

Skills: SD1.2.3.16

Estimated commitment: 1 hour

Prerequisites: Programming distributed-memory systems with MPI - Intermediate

Area PP.PJC Python, Julia,and Co.

PP.PJC1 EVITA course: Python for HPC **(not available for proposals)**

PP.PJC1-SD1.2.9.1 EVITA module: **Getting started with Python for HPC**

Learning outcome: Understand the potential of Python for HPC as well as weaknesses and performance bottlenecks.

Learning outcome: Know some libraries that can help improve performance for scientific computing such as numpy, scipy, numexpr and numba.

Skills: SD1.2.9.1

Estimated commitment: 2 hours

Prerequisites: Basic Python syntax, package management, important libraries

Based on concept from: ASC, Geert Jan Bex

PP.PJC1-SD1.2.9.2 EVITA module: **Python Environments on HPC**

Learning outcome: Create and manage Python virtual environments on HPC systems to isolate dependencies and ensure reproducible execution of Python applications.

Skills: SD1.2.9.2

Estimated commitment: 1 hour

Prerequisites: Basic Python syntax, package management

Based on concept from: ASC, Geert Jan Bex

PP.PJC1-SD1.2.9.3 EVITA module: **Debugging, Benchmarking & Profiling**

Learning outcome: Apply built-in Python profiling methods and benchmarking functionalities to assess application performance, e.g., timeit, cProfile, line_profiler

Learning outcome: Know about third party profiling tools

Skills: SD1.2.9.3, PE2 – Measuring System Performance, PE2.3 – Profiling Tools

Estimated commitment: 3 hours

Prerequisites: Getting started with Python for HPC

Based on concept from: ASC, Geert Jan Bex

PP.PJC1-SD1.2.9.4 EVITA module: **Single-node parallelization & optimization**

Learning outcome: Use native code integrations to extend and accelerate Python HPC applications.

Learning outcome: Apply Cython to improve performance of Python HPC applications.

Learning outcome: Use Dask for single-node, shared-memory parallelism in Python HPC applications.

Skills: SD1.2.9.4, SD1.3 – Efficient Algorithms and Data Structures, PE4.3 Tuning via Reprogramming

Estimated commitment: 4 hours

Prerequisites: Getting started with Python for HPC; Debugging, Benchmarking & Profiling

Based on concept from: ASC, Geert Jan Bex

PP.PJC1-SD1.2.9.5 EVITA module: **Multi-node parallelization**

Learning outcome: Apply mpi4py to parallelise a Python program across multiple nodes on an HPC system

Learning outcome: Use Dask for distributed, multi-node parallelism in Python HPC applications.

Skills: SD1.2.9.5

Estimated commitment: 4 hours

Prerequisites: Previous Python modules.

Based on concept from: ASC, Geert Jan Bex

PP.PJC1-SD1.2.9.6 EVITA module: **Python on GPUs**

Learning outcome: Apply GPU-accelerated Python frameworks (e.g. Numba, RAPIDS, CuPy) to accelerate HPC workloads.

Skills: SD1.2.9.6

Estimated commitment: 4 hours

Prerequisites: previous Python modules.

Based on concept from: ASC, Geert Jan Bex

Area PP.BPC Basic Parallelism Concepts

PP.BPC1 EVITA course: Fundamentals of parallelism

PP.BPC1-K3.1 EVITA module: **Levels of parallelisation**

Learning outcome: Describe parallelization techniques at the intra-node level

Learning outcome: Understand Intra- and Inter-Node, as well as Multi Level approaches

Skills: K3.1

Estimated commitment: 2 hours

Prerequisites: Distributed systems architecture

PP.BPC1-K3.2 EVITA module: **Parallelization overhead**

Learning outcome: Comprehend that data communication is necessary for programs that are parallelized for distributed memory computers (if data communication is not necessary the program is called trivially or embarrassingly parallel)

Learning outcome: Comprehend that synchronization plays an important role with shared memory parallelization

Skills: K3.2

Estimated commitment: 2 hours

Prerequisites: Levels of parallelization

PP.BPC1-K3.3 EVITA module: **Domain decomposition**

Learning outcome: Describe typical decomposition strategies to split a domain into subdomains to make it suited for parallel processing

Learning outcome: Comprehend that in a domain decomposition a region is decomposed, e.g. box is split into smaller ones and a mesh is decomposed into smaller parts, in order to assign these subdomains to processes

Skills: K3.3

Estimated commitment: 4 hours

Prerequisites:

PP.BPC1-K3.4 EVITA module: **Autoparallelization**

Learning outcome: Describe the auto parallelization capabilities of current compilers (e.g. to automatically parallelize suitable loops), which are applicable at the intra-node level

Skills: K3.4

Estimated commitment: 2 hours

Prerequisites:

PP.BPC1-SD1.2.4 EVITA module: **Load balancing**

Learning outcome: Apply simple scheduling algorithms like task farming to achieve an appropriate distribution of the workloads across the multiple computing resources of the HPC system

Learning outcome: Apply domain decomposition strategies

Skills: SD1.2.4

Estimated commitment: 3 hours

Prerequisites:

PP.BPC2 EVITA course: Fundamental of performance estimation

PP.BPC2-K2.1 EVITA module: **Performance frontiers**

Learning outcome: Comprehend that FLOPS (Floating Point Operations per Second) as a key measurement is used in two distinct ways.

Learning outcome: Comprehend that Moore's law from 1965, revised in 1975, states (in simple terms) that the complexity of integrated circuits and thus the computing power of CPUs for HPC systems, respectively, doubles approximately every two years

Skills: K2.1

Estimated commitment: 4 hours

Prerequisites:

PP.BPC2-K2.2 EVITA module: **Bounds for a parallel program**

Learning outcome: Explain how performance bounds of the various components of the HPC system (e.g. CPU, caches, memory) can limit the overall performance of a parallel program

Learning outcome: Explain how performance bounds of the various components of the HPC system (e.g. network, I/O) can limit the overall performance of a parallel program

Skills: K2.2

Estimated commitment: 2 hours

Prerequisites:

PP.BPC2-K2.3 EVITA module: **Performance characteristics**

Learning outcome: Understand Performance characteristics of I/O, CPU Usage, Memory, Communication

Skills: K2.3

Estimated commitment: 1 hour

Prerequisites:

3. List of courses – Computer Architecture

Area CA.CAC Computer Architecture Concepts

CA.CAC1 EVITA course: Computer architecture concepts

CA.CAC1-K1.2 EVITA module: **Basic computer architecture**

Learning outcome: Describe the main components of the computer architecture (processor, memory, IO subsystem, storage, networking) and their relationships

Learning outcome: Describe the taxonomy of instruction sets (ISA) and the characteristics of the different paradigms and alternatives (e.g., RISC vs. CISC, scalar vs. vector...)

Learning outcome: Describe the evolution of computer architectures (RISC, CISC, multi-/many-core, heterogeneous computing, in-memory computing)

Learning outcome: Translate simple high-level code to assembly code

Learning outcome: Evaluate the theoretical performance of simple code fragments.

Skills: K1.2

Estimated commitment: 4 hours

Prerequisites: Sequential programming

CA.CAC1-K1.2.3 EVITA module: **Memory hierarchy**

Learning outcome: Describe the structure and components of the memory hierarchy, and how they operate

Learning outcome: Explain the concepts of cache hit and cache miss

Learning outcome: Explain the different types of cache misses in single processor architectures: compulsory, capacity and conflict misses

Learning outcome: Describe how cache memories are used during application execution

Skills: K1.2.3

Estimated commitment: 4 hours

Prerequisites: Basic computer architecture

Area CA.PA Processor Architecture

CA.PA1 EVITA course: processor architecture

CA.PA1-K1.2.1.1 EVITA module: **In-order processor architecture**

Learning outcome: Describe the techniques to develop in-order scalar processors: pipelining, superscalar, VLIW, vector SIMD extensions, and their principles of operation

Learning outcome: Explain the fundamental mechanisms to achieve static Instruction-Level Parallelism (ILP)

Learning outcome: Describe the software-level instruction scheduling techniques (list scheduling, unrolling, software pipelining)

Learning outcome: Explain the memory hierarchy and its relationship with the processor: instruction and data caches, TLB, virtual to physical address translation, and main memory

Learning outcome: Design simple cache memories that fit within a processor design

Learning outcome: Describe the protection and security mechanisms needed in the processor architecture

Learning outcome: Analyze the execution of single-threaded programs on in-order processors, identifying performance bottlenecks and proposing improvements.

Learning outcome: Apply simple optimizations to code fragments to improve their performance and power consumption

Skills: K1.2.1.1

Estimated commitment: 4 hours

Prerequisites: Basic computer architecture

CA.PA1-K1.2.1.2 EVITA module: **Out-of-order processor architecture**

Learning outcome: Describe the fundamental mechanisms for dynamic Instruction-Level Parallelism (ILP) processors: out-of-order execution, branch prediction, exception handling, instruction and data prefetch, etc.)

Learning outcome: Explain the design and functionality of microprocessors with parallel computational pipelines and dynamic scheduling of instructions

Learning outcome: Describe the Scoreboard and Tomasulo architectures

Learning outcome: Explain the dynamic branch prediction techniques

Learning outcome: Describe the environment on which the processor sits, the uncore

Learning outcome: Explain the technology aspects of computer architecture development

Learning outcome: Describe the Roofline model

Learning outcome: Describe how theoretical design concepts are applied to actual processors (AMD, Intel, Arm, RISC-V)

Skills: K1.2.1.2

Estimated commitment: 4 hours

Prerequisites: In-order processor architecture

CA.PA1-K1.2.4 EVITA module: **Heterogeneous computer architecture - accelerators**

Learning outcome: Describe the computer architecture techniques used in accelerator architectures (GPU, FPGA)

Learning outcome: Ability to explain key features of modern accelerators (GPU, FPGA)

Learning outcome: Explain the accelerators architecture, ISA and microarchitecture

Learning outcome: Explain the specific accelerator's memory hierarchy

Learning outcome: Describe the inclusion of GPUs on a CPU system (integrated, discrete)

Skills: K1.2.4

Estimated commitment: 4 hours

Prerequisites: In-order processor architecture, Out-of-order processor architecture

CA.PA1-K1.2.5 EVITA module: **Multiprocessor architecture**

Learning outcome: Describe the implementation of different models of thread-level and data-level parallelism, such as core multithreading, many-core processors, vector units and multi-chip modules

Learning outcome: Explain the impact of multiprocessing on the processor ISAs

Learning outcome: Describe the implementation of hardware-level synchronization mechanisms

Learning outcome: Describe the operation of the memory hierarchy on multiprocessors: snooping and directory-based coherence protocols

Learning outcome: Explain the concepts of coherency and consistency

Learning outcome: Design at the architectural level hardware for shared-memory, and

evaluate its characteristics

Learning outcome: Design at the architectural level the basic components of modern parallel systems, including multiple processors, cache hierarchies and networks on chip

Learning outcome: Estimate performance metrics for parallel systems and evaluate parallel computer organizations

Learning outcome: Consider protection and security issues in the multiprocessor design

Skills: K1.2.5

Estimated commitment: 4 hours

Prerequisites: In-order processor architecture, Out-of-order processor architecture

Area CA.IOS Input/output subsystems

CA.IOS1 EVITA course: Input/output subsystem

CA.IOS1-K1.2.5 EVITA module: **Basic input/output**

Learning outcome: Describe the basic I/O subsystem components and their interrelations: buses, devices, protocols

Learning outcome: Explain the interaction of the processor with the I/O subsystem: device drivers, access to the I/O control and data, interrupts and polling

Learning outcome: Describe the main techniques and mechanisms to ensure error detection and correction

Skills: K1.2.5

Estimated commitment: 4 hours

Prerequisites: Basic computer architecture, Memory hierarchy

CA.IOS1-K1.3.1.1 EVITA module: **Storage systems**

Learning outcome: Describe the structure and operation of hard disks (HDD)

Learning outcome: Describe the structure and operation of solid state disks (SSD)

Learning outcome: Explain the support for reliability on HDD and SSD devices

Skills: K1.3.1.1

Estimated commitment: 4 hours

Prerequisites: Basic computer architecture, Memory hierarchy

CA.IOS1-K1.2.2 EVITA module: **Networking**

Learning outcome: Describe the network topologies

Learning outcome: Describe the routing algorithms

Learning outcome: Describe flow control

Learning outcome: Describe congestion control mechanisms

Learning outcome: Describe error detection/correction mechanisms

Learning outcome: Describe the ISO/OSI model

Learning outcome: Describe network protocols (TCP/IP, RDMA, etc.)

Learning outcome: Describe the network performance metrics (e.g., latency, bandwidth, and quality of service)

Learning outcome: Consider the protection and security issues related to communications

Skills: K1.2.2

Estimated commitment: 4 hours

Prerequisites: Basic computer architecture, Memory hierarchy

CA.IOS2 EVITA course: Parallel I/O with NetCDF, HDF5 and others

CA.IOS2-K1.3.1.4.1 EVITA module: **Parallel I/O with NetCDF**

Learning outcome: Describe the Classic NetCDF Model characteristics and limitations

Learning outcome: Describe the relationships among elements of the CF-NetCDF conventions and NetCDF entities

Skills: K1.3.1.4.1, K1.3.1.4.2

Estimated commitment: 4 hours

Prerequisites: Basic Input/Output, Storage Systems

CA.IOS2-K1.3.1.5.3 EVITA module: **Parallel I/O with PNetCDF**

Learning outcome: Interpret the PNetCDF files structure.

Learning outcome: Store and retrieve data in PNetCDF

Skills: K1.3.1.5.3

Estimated commitment: 3 hours

Prerequisites: Parallel I/O with NetCDF

CA.IOS2-K1.3.1.4.3 EVITA module: **Parallel I/O with HDF5 serial access to multiple files**

Learning outcome: Understand HDF5 data structure

Learning outcome: Describe HDF5 file access in serial mode

Skills: K1.3.1.4.3

Estimated commitment: 4 hours

Prerequisites: Parallel I/O with NetCDF, Parallel I/O with PNetCDF

CA.IOS2-K1.3.1.4.4 EVITA module: **Parallel I/O with HDF5 parallel access to single file**

Learning outcome: Understand parallel implementation of HDF5

Learning outcome: Use striping to write with multple writer to one file simulataneously

Skills: K1.3.1.4.4

Estimated commitment: 4 hours

Prerequisites: Parallel I/O with NetCDF, Parallel I/O with PNetCDF

Area CA.SCH Scheduling

CA.SCH1 EVITA course: Slurm for Users

CA.SCH1-USE2.1 EVITA module: **Job Scheduling**

Learning outcome: Run parallel programs in an HPC environment.

Learning outcome: Select the appropriate software environment.

Learning outcome: Use a workload manager like SLURM or TORQUE to allocate HPC resources (e.g. CPUs) and to submit a batch job.

Skills: USE2.1

Estimated commitment: 4 hours

Prerequisites: Distributed System Architecture

CA.SCH1-USE2.2 EVITA module: **Job scripts**

Learning outcome: Write robust job scripts, e.g. to simplify job submissions by the help of automated job chaining.

Learning outcome: Select the appropriate software environment.

Skills: USE2.2

Estimated commitment: 4 hours

Prerequisites: Job scheduling

CA.SCH2 EVITA course: Scheduling fundamentals and different scheduler

CA.SCH2-K4.1 EVITA module: **Basic principles of job scheduling**

Learning outcome: Comprehend the exclusive and shared usage model in HPC

Learning outcome: Comprehend the generic concepts and architecture of resource manager, scheduler, job and job script

Skills: K4.1

Estimated commitment: 4 hours

Prerequisites: Distributed System Architecture

CA.SCH2-K4.2 EVITA module: **Slurm the scheduler**

Learning outcome: Explain the architecture of SLURM, i.e., the role of slurmd, srun and the injection of environment variables

Learning outcome: Explain the function of the tools: sacct, sbatch, salloc, srun, scancel, squeue, sinfo

Skills: K4.2

Estimated commitment: 4 horus

Prerequisites: Basic principles of job scheduling

CA.SCH2-K4.4 EVITA module: **HTCondor the scheduler**

Learning outcome: Explain the architecture of HTCondor

Learning outcome: Describe the benefits of HTCondor compared to different Scheduler

Skills: K4.4

Estimated commitment: 4 hours

Prerequisites: Basic principles of job scheduling

CA.SCH2-K4.3 EVITA module: **Scheduling strategies**

Learning outcome: Undertand Backfilling

Learning outcome: Undertand Shortest Job First

Skills: K4.3

Estimated commitment: 1 hours

Prerequisites: Basic principles of job scheduling

CA.SCH2-K4.5 EVITA module: **Fairshare and priorities in Scheduling**

Learning outcome: Understand the concept of fair share

Learning outcome: Implement algorithm to combine scheduling strategies with fair share values

Skills: K4.5

Estimated commitment: 1 hour

Prerequisites: Basic principles of job scheduling

CA.SCH2-K4.6 EVITA module: **Two layer scheduling - running a scheduler through a scheduler**

Learning outcome: Understand the stregths and weaknesses of different schedulers

Learning outcome: Describe how to implement a scheduler in a scheduler on a level of service deployment

Skills: K4.6

Estimated commitment: 4 hours

Prerequisites: Basic principles of job scheduling

CA.SCH2-K4.7 EVITA module: **K8s the scheduler**

Learning outcome: Understand the concept and strategies of k8s

Learning outcome: Implement a simple workflow

Skills: K4.7

Estimated commitment: 4 hours

Prerequisites: Basic principles of job scheduling

CA.SCH2-K4.8 EVITA module: **Scaling of the scheduling problem - high number of short jobs**

Learning outcome: Understand the difficulties of scheduling different sized jobs

Learning outcome: Describe an effective way of managing batches of smaller jobs

Skills: K4.8

Estimated commitment: 2 hours

Prerequisites: Basic principles of job scheduling

4. List of courses – Mathematics and Statistics

Area MAT Mathematics and Statistics

MAT1 EVITA course: Numerical methods for linear systems **(not available for proposals)**

MAT1-SD9.1 EVITA module: **Direct methods for solving linear systems**

Learning outcome: Describe the principles of Gaussian elimination and LU factorization for solving linear systems.

Learning outcome: Apply direct methods to solve small to medium-sized systems and understand their computational and memory characteristics.

Skills: SD9.1

Estimated commitment: 1 hours

Prerequisites: Basic linear algebra; familiarity with matrices and vector operations.

MAT1-SD9.2 EVITA module: **Stationary iterative methods**

Learning outcome: Understand the relevance and benefits of iterative methods.

Learning outcome: Describe the Jacobi, Gauss-Seidel, and Successive Over-Relaxation (SOR) methods.

Learning outcome: Explain and analyze convergence of the methods.

Learning outcome: Apply Jacobi, Gauss-Seidel, and SOR to representative linear systems.

Learning outcome: Understand the parallelization capabilities of the different stationary iterative methods.

Skills: SD9.2

Estimated commitment: 3 hours

Prerequisites: direct methods for solving linear systems

MAT1-SD9.3 EVITA module: **Polynomial iterative methods and Krylov subspace methods**

Learning outcome: Understand polynomial iterative methods and how iterative refinements are generated.

Learning outcome: Explain the Conjugate Gradients method (CG), MinRes, and Generalized Minimum Residual method (GMRES).

Learning outcome: Apply Krylov methods to solve linear systems and analyze convergence.

Learning outcome: Understand the parallelization capabilities of the different methods.

Skills: SD9.3

Estimated commitment: 3 hours

Prerequisites: direct methods and stationary iterative methods for solving linear systems

MAT1-SD9.4 EVITA module: **Sparse matrix structures and preconditioning**

Learning outcome: Understand sparse matrix storage schemes and their impact on algebraic operations and performance.

Learning outcome: Describe the purpose and effect of preconditioning on convergence of

stationary and Krylov methods.

Learning outcome: Identify and apply basic preconditioners (e.g., Jacobi, ILU(0), incomplete factorizations).

Learning outcome: Understand the parallelization capabilities of the different preconditioners.

Skills: SD9.4

Estimated commitment: 2 hours

Prerequisites: direct methods, stationary iterative methods, and polynomial iterative methods for solving linear systems

MAT2 EVITA course: Introduction to statistics **(not available for proposals)**

MAT2-SD9.5 EVITA module: **Probability theory and data analysis foundations**

Learning outcome: Describe the foundations of probability theory, including random variables, distributions, expectation, variance, and dependence.

Learning outcome: Describe data analysis concepts used to summarize, explore, and interpret large computational datasets.

Skills: SD9.5

Estimated commitment: 3 hours

Prerequisites:

MAT2-SD9.6 EVITA module: **Statistical tests and experiment design**

Learning outcome: Describe point estimation and interval estimation, including confidence intervals and uncertainty quantification.

Learning outcome: Describe the notion, purpose, and interpretation of statistical tests, including hypotheses, test statistics, and p-values.

Learning outcome: Describe principles of experimental design, including factor selection, replication, randomization, and variance control.

Skills: SD9.6

Estimated commitment: 3 hours

Prerequisites: basic probability theory

MAT2-SD9.7 EVITA module: **Linear models and introductory statistical learning**

Learning outcome: Describe linear models: fit and interpret regression and time series models.

Learning outcome: Describe basic statistical learning and supervised classification and regression.

Skills: SD9.7

Estimated commitment: 2 hours

Prerequisites: basic probability theory

MAT2-SD9.8 EVITA module: **Non-parametric methods and principal component analysis (PCA)**

Learning outcome: Describe non-parametric decision rules like k-nearest neighbours, decision trees and Principal Component Analysis (PCA).

Skills: SD9.8

Estimated commitment: 2 hours

Prerequisites: basic linear algebra

MAT2-SD9.9 EVITA module: **Neural networks and Monte Carlo methods**

Learning outcome: Describe neural network models at an introductory level, including basic architectures and training concepts.

Learning outcome: Describe Monte Carlo methods and their use for stochastic simulation, integration, and uncertainty propagation in HPC contexts.

Skills: SD9.9

Estimated commitment: 3 hours

Prerequisites: basic probability theory, basic simulation concepts

MAT3 EVITA course: Numerical methods for PDEs **(not available for proposals)**

MAT3-SD9.10 EVITA module: **Finite difference methods for elliptic, parabolic, hyperbolic, and transport equations**

Learning outcome: Describe finite difference method and apply it to Laplace, Poisson, heat, wave, and transport-type equations, including stencil structures.

Learning outcome:Identify computational characteristics of finite-difference stencils.

Skills: SD9.10

Estimated commitment: 4 hours

Prerequisites: Linear algebra, Basics about PDEs

MAT3-SD9.11 EVITA module: **Numerical analysis of finite differences schemes**

Learning outcome: Define and analyze stability, consistency, and rate of convergence for PDE discretizations.

Skills: SD9.11

Estimated commitment: 2 hours

Prerequisites: Linear algebra, Basics about PDEs, previous modules

MAT3-SD9.12 EVITA module: **Introductory finite element method and Galerkin approach**

Learning outcome: Describe the method of finite elements and the Galerkin method

Skills: SD9.12

Estimated commitment: 2 hours

Prerequisites: Linear Algebra, previous modules

MAT3-SD9.13 EVITA module: **Fourier and spectral methods**

Learning outcome: Describe Fourier-based solution strategies for PDEs

Learning outcome: Explain the spectral method and identify scenarios where it is advantageous.

Skills: SD9.13

Estimated commitment: 2 hours

Prerequisites: previous modules

MAT4 EVITA course: Numerical optimization **(not available for proposals)**

MAT4-SD9.14 EVITA module: **Optimization Problem Classes and Optimality Theory**

Learning outcome: Describe major classes of optimization problems: continuous vs. discrete, constrained vs. unconstrained, convex vs. non-convex, smooth vs. nonsmooth, and deterministic vs. stochastic optimization.

Learning outcome: Describe first- and second-order optimality conditions for smooth optimization problems.

Learning outcome: Explain the Karush-Kuhn-Tucker (KKT) conditions and Lagrangian duality, and their role in algorithm design and problem decomposition.

Skills: SD9.14

Estimated commitment: 3 hours

Prerequisites: Calculus, linear algebra.

MAT4-SD9.15 EVITA module: **Convex and Nonsmooth Optimization Methods**

Learning outcome: Describe convex and nonsmooth optimization problems, including linear programming formulations.

Learning outcome: Explain projected gradient and proximal gradient methods for constrained and composite optimization problems.

Learning outcome: Describe acceleration techniques such as Nesterov acceleration and their effect on convergence rates and computational efficiency.

Skills: SD9.15

Estimated commitment: 3 hours

Prerequisites: previous modules.

MAT4-SD9.16 EVITA module: **Large-Scale and Stochastic Optimization Algorithms**

Learning outcome: Describe stochastic gradient descent (SGD), and applicability to large-scale problems.

Learning outcome: Describe quasi-Newton methods with emphasis on L-BFGS and explain their suitability for memory- and compute-constrained HPC settings.

Skills: SD9.16

Estimated commitment: 3 hours

Prerequisites: previous modules.

5. List of courses – Specialisation: System Development and Support

Area SDS.OSV Operating Systems and Virtualisation

SDS.OSV1 EVITA course: Virtualization and the Linux operating system

SDS.OSV1-ADM2.3 EVITA module: **Virtualization and Virtual machines**

Learning outcome: Understand image management

Learning outcome: Understand hardware passthrough

Skills: ADM2.3

Estimated commitment: 4 hours

SDS.OSV1-USE1.3 EVITA module: **UNIX File System Tree**

Learning outcome: Describe the organization of a hierarchical file system

Learning outcome: Describe the Filesystem Hierarchy Standard and the relevance of the directories: etc, home, opt, lib, bin, sbin, tmp, ...

Skills: USE1.3

Estimated commitment: 1 hours

SDS.OSV1-USE1.6 EVITA module: **CLI file editors**

Learning outcome: Discuss what a CLI file editor needs to be able to do and how it should behave

Learning outcome: Demonstrate the usage of an editor to open a file in write mode and save the file

Skills: USE1.6

Estimated commitment: 3 hours

SDS.OSV1-USE1.7 EVITA module: **Terminal multiplexers**

Learning outcome: Give examples for use cases for such a multiplexer

Learning outcome: Review editing files using a terminal multiplexer

Skills: USE1.7

Estimated commitment: 2 hours

SDS.OSV1-USE1.8 EVITA module: **Managing background processes and signaling**

Learning outcome: Understand interrupting or aborting of a program execution using CTRL-C and CTRL-Z

Learning outcome: Understand the behaviour of the kill command

Skills: USE1.8

Estimated commitment: 2 hours

SDS.OSV1-ADM7.1 EVITA module: **Different Linux derivatives and their use in HPC systems**

Learning outcome: Understand the origins of currently existing Linux derivatives for HPC Systems

Learning outcome: Describe how a derivative can be created

Skills: ADM7.1

Estimated commitment: 4 hours

SDS.OSV1-ADM7.2 EVITA module: **Managing services using SystemD**

Learning outcome: Understand Services in Linux and the chage from init.d to SystemD

Learning outcome: Understand the structure of servcice files

Skills: ADM7.2

Estimated commitment: 2 hours

SDS.OSV1-ADM7.3 EVITA module: **Writing your own SystemD service files**

Learning outcome: Read existing service files and display their structure in the boot process visually

Learning outcome: Write your own service file and place it into the execution tree

Skills: ADM7.3

Estimated commitment: 3 hours

SDS.OSV1-ADM7.4 EVITA module: **Compiling and loading hardware modules in the Linux kernel**

Learning outcome: Understand how hardware modules are handled under Linux

Learning outcome: Compile a hardware module and activate it

Skills: ADM7.4

Estimated commitment: 4 hours

SDS.OSV2 EVITA course: Device management and process handling on a CPU

SDS.OSV2-ADM7.5 EVITA module: **NUMA systems and huge pages**

Learning outcome: Understand what NUMA domains are

Learning outcome: Implement a usage demonstration of NUMA boumds

Skills: ADM7.5

Estimated commitment: 4 hours

SDS.OSV2-ADM7.6 EVITA module: **Linux device drivers (communications, storage, non-volatile memory)**

Learning outcome: Understand device drivers and how programs communication with them

Learning outcome: Pic a device and access it using POSIX interface

Skills: ADM7.6

Estimated commitment: 3 hours

SDS.OSV2-ADM7.7 EVITA module**: Memory locallity**

Learning outcome: Understand the concept of locallity

Learning outcome: Implement an example that exploits locallity and shows performance differences

Skills: ADM7.7

Estimated commitment: 3 hours

Area SDS.SSADM Storage and System Administration

SDS.SSADM1 EVITA course: Storage, Parallel File Systems and Databases

SDS.SSADM1-K1.3.1.2 EVITA module: **Storage system and mechanisms**

Learning outcome: Understand Object storage

Learning outcome: Understand Burst Buffer

Skills: K1.3.1.2

Estimated commitment: 2 hours

SDS.SSADM1-K1.3.1.3.1.1 EVITA module: **Parallel file systems and how they work**

Learning outcome: Comprehend that the parallel or cluster aspect is twofold: Firstly, the hardware is parallel itself (the file system is provided by several servers that operate in a coordinated way). Secondly, parallel I/O is enabled, i.e. more than one process can consistently write to the same file at the same time

Skills: K1.3.1.3.1.1

Estimated commitment: 2 hours

SDS.SSADM1-K1.3.1.3.2 EVITA module: **Network file systems**

Learning outcome: Comprehend that Network file systems are not designed for (very) high I/O loads

Skills: K1.3.1.3.2

Estimated commitment: 1 hour

SDS.SSADM1-K1.3.1.6 EVITA module: **The POSIX interface for storage systems**

Learning outcome: Understand the POSIX interface

Skills: K1.3.1.6

Estimated commitment: 2 hours

SDS.SSADM1-K1.3.3 EVITA module: **Datamanagement and how to organize files and folders**

Learning outcome: Understand Cache coherence

Learning outcome: Understand Information lifecycle management

Skills: K1.3.3

Estimated commitment: 2 hours

SDS.SSADM1-K1.3.4 EVITA module: **Data access patterns**

Learning outcome: Understand access patterns

Learning outcome: Understand how different access patterns influence data read and write operation2

Skills: K1.3.4

Estimated commitment: 2 hours

SDS.SSADM1-K1.3.2 EVITA module: **Data reduction techniques**

Learning outcome: Understand Data reduction technique

Skills: K1.3.2

Estimated commitment: 2 hours

SDS.SSADM2 EVITA course: System Administration on HPC Environments

SDS.SSADM2-ADM3.4.4.1.1 EVITA module: **Cluster management using Warewulf booting a default container**

Learning outcome: Remember the concept of stateless deployment

Learning outcome: Understand the concept of deploying servers using Warewulf

Skills: ADM3.4.4.1.1, SDM.3.4.2

Estimated commitment: 4 hours

SDS.SSADM2-ADM3.4.4.1.2 EVITA module: **Cluster management using Warewulf creating a custom container and adding software**

Learning outcome: Understand the container-built scripts used in Warewulf.

Learning outcome: Deploy a custom container script and update the software used in it

Learning outcome: Apply changes to the user system in the custom container script

Skills: ADM3.4.4.1.2

Estimated commitment: 4 hours

SDS.SSADM2-SD3.1.1.1 EVITA module: **Installation and set up of a slurm controller**

Learning outcome: Remember the concept of workload managment

Learning outcome: Understand the functioning principle of Slurm as a workload manager

Learning outcomes: Install Slurm as a cluster manager.

Skills: SD3.1.1.1

Estimated commitment: 4 hours

SDS.SSADM2-ADM3.4.3.3 EVITA module: **Cluster deployment and user management using Warewulf**

Learning outcome: Remember cluster management using Warewulf

Learning outcome: Understand user management in Warewulf

Learning outcome: Remember identity management such as LDAP

Skills: ADM3.4.3.3, ADM2.5

Estimated commitment: 3 hours

SDS.SSADM2-ADM2.2.1 EVITA module: **Using the Network File System (NFS) in a stateless cluster**

Learning outcome: Describe common usage of NFS

Learning outcome: Deploy NFS infrastructure on a server and a client

Learning outcome: Examine NFS deployments

Skills: ADM2.2.1

Estimated commitment: 1 hour

SDS.SSADM2-ADM4.1 EVITA module: **Software environment and software provisioning**

Learning outcome: Remember environment modules for the user perspective

Learning outcome: Understand how to manage software using SPACK

Learning outcome: Install Lmod and SPACK on a server

Learning outcome: Install software using the provided module files.

Skills: ADM4.1

Estimated commitment: 4 hours

SDS.SSADM2-ADM3.6 EVITA module: **Monitoring in High-Performance Computing Using the TIG Stack**

Learning outcome: Understand the need and use of Monitoring on a cluster

Learning outcome: Install a basic stack consisting of Telegraf, Influx, and Grafana (TIG)

Skills: ADM3.6

Estimated commitment: 4 hours

SDS.SSADM2-ADM3.8 EVITA module: **Firewall and their importance for a cluster**

Learning outcome: Understand the concept of a firewall

Learning outcome: Classify problems that can be solved by using a firewall

Learning outcome: Install a simple firewall on a cluster and set up some rules

Skills: ADM3.8

Estimated commitment: 2 hours

SDS.SSADM2-ADM6 EVITA module: **Documentation of system settings and setup procedures**

Learning outcome: Remember Dokumentation best practices

Learning outcome: Differentiate between user documentation and admin documentation

Learning outcome: Understand the need for documenting the steps taken to reproduce a

system configuration

Learning outcome: Document a process from the admin and user perspectives.

Skills: ADM6

Estimated commitment: 1 hour

SDS.SSADM2-ADM3.7 EVITA module: **Intelligent Platform Management (IPMI)**

Learning outcome: Understand the requirement for remote cluster management

Learning outcome: Review the abilities of the IPMI-equipped server

Skills: ADM3.7

Estimated commitment: 2 hours

SDS.SSADM2-ADM3.3 EVITA module: **Clustershell**

Learning outcome: Understand the difficulty of managing many computers remotely

Learning outcome: Use the clustershell to ecute commands on remote nodes and copy files between cluster manager and remote node

Skills: ADM3.3

Estimated commitment: 1 hour

SDS.SSADM2-SD1.4 EVITA module: **Deep dive into container for system administration**

Learning outcome: Understand when to use containers for HPC

Learning outcome: Be able to convert a local workflow on workstation/laptop into a container suitable for HPC systems

Learning outcome: Getting data in and out of containers, getting MPI and CUDA to work, etc.

Learning outcome: Use GitLab for building containers, using containers for testing, and storing containers

Skills: SD1.4

Estimated commitment: 4 hours

SDS.SSADM3 EVITA course: Energy efficient HPC **(not available for proposals)**

SDS.SSADM3-PE6.1 EVITA module: **Energy Measurement and Monitoring in HPC Systems**

Learning outcome: Operate power-monitoring systems available in data centres to measure and record energy consumption of HPC infrastructure components.

Skills: PE6.1

Estimated commitment: 1.5 hours

Prerequisites: K1.1 System Architechtures, K1.2 Hardware Architectures

SDS.SSADM3-PE6.2 EVITA module: **Power Management and Energy Optimization in HPC**

Learning outcome: Manage CPU and GPU power states in HPC systems to control and optimise energy consumption.

Learning outcome: Optimise the energy efficiency of parallel applications by analysing and improving their computational and communication behaviour on HPC systems.

Skills: PE6.2

Estimated commitment: 3 hours

Prerequisites: K1.1 System Architecture, K1.2 Hardware Architectures, USE2 Running of Parallel Programs, K2 Performance Modeling, PE2.3 Profiling tools, Energy Measurement and Monitoring in HPC Systems

SDS.SSADM4 EVITA course: Deep dive into container for system administration **(not available for proposals)**

SDS.SSADM4-SD1.4 EVITA module: **Deep Dive into container for system administration**

Learning outcome: Understand when to use containers for HPC

Learning outcome: Be able to convert a local workflow on workstation/laptop into a container suitable for HPC systems

Learning outcome: Getting data in and out of containers, getting MPI and CUDA to work, etc.

Learning outcome: Use GitLab for building containers, using containers for testing, and storing containers

Skills: SD1.4

Estimated commitment: 4 hours

SDS.SSADM5 EVITA course: Cluster infrastructure

SDS.SSADM5-ADM2.1 EVITA module: **Cluster infrastructure - Networking equipment**

Learning outcome: Understand Linux Network Configuration

Learning outcome: Understand Connecting Physical hardware

Skills: ADM2.1

Estimated commitment: 4 hours

SDS.SSADM5-ADM2.2 EVITA module: **Cluster infrastructure - Storage equipment and servers**

Learning outcome: Understand file systems and remembering: Local file systems, file systems over network, quota, snapshots

Learning outcome: Describe common usage of NFS in a cluster

Skills: ADM2.2

Estimated commitment: 4 hours

SDS.SSADM5-ADM2.4 EVITA module: **Cluster infrastructure - Power distribution and management**

Learning outcome: Understand Breakers and disconnects

Learning outcome: Understand Distributing Load

Skills: ADM2.4

Estimated commitment: 3 hours

SDS.SSADM5-ADM2.5.1 EVITA module: **Cluster infrastructure - Air cooled systems and racks**

Learning outcome: Understand Air cooling

Learning outcome: Describe air cooling implementation for full racks

Skills: ADM2.5.1

Estimated commitment: 2 hours

SDS.SSADM5-ADM2.5.2 EVITA module: **Cluster infrastructure - Direct liquid colled systems and racks**

Learning outcome: Understand water cooling

Learning outcome: Describe waiter and direct lquid cooling for full racks

Skills: ADM2.5.2

Estimated commitment: 3 hours

SDS.SSADM6 EVITA course: Cluster security

SDS.SSADM6-ADM1.7.1 EVITA module: **Cluster security - Best practices for admins**

Learning outcome: Remember best practices

Skills: ADM1.7.1

Estimated commitment: 2 hours

SDS.SSADM6-ADM1.7.2 EVITA module: **Cluster security - User management with Unix Permission model**

Learning outcome: Understand the Unix Permission Model

Learning outcome: Describe security implications

Skills: ADM1.7.2

Estimated commitment: 2 hours

SDS.SSADM6-ADM1.7.3 EVITA module: **Cluster security - Port security and using SSH and VPN**

Learning outcome: Understand Network segmentation and firewalls

Learning outcome: Understand externally open ports and their security implications

Skills: ADM1.7.3

Estimated commitment: 3 hours

SDS.SSADM6-ADM1.7.4 EVITA module: **Cluster security - 2FA methods for accounts and SSH**

Learning outcome: Understand the need for 2FA methods

Learning outcome: Investigate different 2FA implementation for SSH

Skills: ADM1.7.4

Estimated commitment: 2 hours

SDS.SSADM6-ADM1.7.52 EVITA module: **Cluster security - Fencing methods for sensitive data (GDPR)**

Learning outcome: Understand fencing of users and data

Learning outcome: Implement a simple fencing mechanis using POSIX

Skills: ADM1.7.52

Estimated commitment: 2 hours

SDS.SSADM7 EVITA course: Modelling costs

SDS.SSADM7-K5 EVITA module: **Modelling costs of running an HPC system**

Learning outcome: Describe the impact of a cluster nodes type (e.g. CPU type, main memory expansion, or GPU extensions) and of the storage media type (SSD, disk, or e.g. tape for long term archiving (LTA) purposes) on its costs

Learning outcome: Describe how to assess runtime costs for jobs

Skills: K5

Estimated commitment: 4 hours or more

SDS.SSADM8 EVITA course: Parallel filesystem types

SDS.SSADM8-K1.3.1.3.1.2 EVITA module: **Parallel File system: Lustre**

Learning outcome: Understand the benefits and weaknesses of Lustre

Learning outcome: Compare Lustre against other solutions.

Skills: K1.3.1.3.1.2

Estimated commitment: 2 hours

SDS.SSADM8-K1.3.1.3.1.3 EVITA module: **Parallel File system: Gluster**

Learning outcome: Understand the benefits and weaknesses of Gluster

Learning outcome: Compare Gluster against other solutions

Skills: K1.3.1.3.1.3

Estimated commitment: 2 hours

SDS.SSADM8-K1.3.1.3.1.4 EVITA module: **Parallel File system: Ceph**

Learning outcome: Understand the benefits and weaknesses of Ceph

Learning outcome: Compare Ceph against other solutions

Skills: K1.3.1.3.1.4

Estimated commitment: 2 hours

SDS.SSADM8-K1.3.1.3.1.5 EVITA module: **Parallel File system: GPFS**

Learning outcome: Understand the benefits and weaknesses of GPFS

Learning outcome: Compare GPFS against other solutions

Skills: K1.3.1.3.1.5

Estimated commitment: 2 hours

SDS.SSADM8-K1.3.1.3.1.6 EVITA module: **Parallel File system: Vast-Data**

Learning outcome: Understand the benefits and weaknesses of Vast-Data

Learning outcome: Compare Vast-Data against other solutions especially open source offers

Skills: K1.3.1.3.1.6

Estimated commitment: 2 hours

SDS.SSADM8-K1.3.1.3.1.7 EVITA module: **Parallel File system: Orange-FS**

Learning outcome: Understand the benefits and weaknesses of Orange-FS

Learning outcome: Compare Oragnge-FS against other solutions

Skills: K1.3.1.3.1.7

Estimated commitment: 2 hours

SDS.SSADM8-K1.3.1.3.1.8 EVITA module: **Parallel File system: Daos**

Learning outcome: Understand the benefits and weaknesses of Daos

Learning outcome: Compare Daos against other solutions

Skills: K1.3.1.3.1.8

Estimated commitment: 2 hours

SDS.SSADM8-K1.3.1.3.1.9 EVITA module: **Parallel File system: BeeGFS**

Learning outcome: Understand the benefits and weaknesses of BeegFS

Learning outcome: Compare BeeGFS against other solutions

Skills: K1.3.1.3.1.9

Estimated commitment: 2 hours

SDS.SSADM9 EVITA course: Networking

SDS.SSADM9-K1.2.2.2 EVITA module: **Networking: DHCP**

Learning outcome: Understand DHCP and how to use it

Learning outcome: Implement and set up a simple DHPC server

Skills: K1.2.2.2

Estimated commitment: 3 hours

SDS.SSADM9-K1.2.2.3 EVITA module: **Networking: DNS**

Learning outcome: Understand global DNS settings

Learning outcome: Investigate how to register a DNS entry

Skills: K1.2.2.3

Estimated commitment: 2 hours

SDS.SSADM9-K1.2.2.4 EVITA module: **Networking: IP Routing**

Learning outcome: Understand IP ranges and how to route between them

Learning outcome: Implement a simply IP route between two different reanges

Skills: K1.2.2.4

Estimated commitment: 3 hours

SDS.SSADM9-K1.2.2.5 EVITA module: **Networking: Subnetting and VLan**

Learning outcome: Understand the user of Switch level isolation

Learning outcome: Understand the concept of Vlan subnetting

Skills: K1.2.2.5

Estimated commitment: 4 hours

SDS.SSADM9-K1.2.2.6 EVITA module: **Networking: TCP/IP**

Learning outcome: Understand the concept of a TCP connection

Learning outcome: Understand the difference TCP and UDP

Skills: K1.2.2.6

Estimated commitment: 4 hours

Area SDS.CPPM Compilers and Parallel Programming Models

SDS.CPPM1 EVITA course: IO programming middleware

SDS.CPPM1-SD1.2.5.1.13 EVITA module: I**O programming middleware: NetCDF APIs**

Learning outcome: Examine the NetCDF remote data access

Learning outcome: Examine different NetCDF programming APIs (choose a few)

Skills: SD1.2.5.1.13, SD1.2.5.1

Estimated commitment: 4 hours

SDS.CPPM1-SD1.2.5.2 EVITA module: **IO Programming middleware: XIOS**

Learning outcome: Describe the XIOS-XML terminology and structuration

Learning outcome: Test XIOS on a specific architecture

Skills: SD1.2.5.2

Estimated commitment: 4 hours

SDS.CPPM1-SD1.2.5.3 EVITA module: **IO Programming middleware: HDF5**

Learning outcome: Describe the data structure of HDF5

Learning outcome: Implement file access using the HDF5 API in different languages

Skills: SD1.2.5.3

Estimated commitment: 4 hours

SDS.CPPM2 EVITA course: Compilers

SDS.CPPM2-SD1.5.1 EVITA module: **Compilers: GCC**

Learning outcome: Understand the working concept of the GCC Compiler

Learning outcome: Implement a simple program that user optimization feature of the

compiler

Skills: SD1.5.1

Estimated commitment: 4 hours

SDS.CPPM2-SD1.5.2 EVITA module: **Compilers: LLVM**

Learning outcome: Understand the working concept of the LLVM Compiler

Learning outcome: Implement a simple program that user optimization feature of the compiler

Skills: SD1.5.2

Estimated commitment: 4 hours

SDS.CPPM2-SD1.5.3 EVITA module: **Compilers: Intel-one API**

Learning outcome: Understand the working concept of the Intel-One API Compiler

Learning outcome: Implement a simple program that user optimization feature of the compiler

Skills: SD1.5.3

Estimated commitment: 4 hours

SDS.CPPM2-SD1.5.4 EVITA module: **Compilers: CUDA-PTX**

Learning outcome: Understand the working concept of the CUDA-PTX Compiler

Learning outcome: Implement a simple program that user optimization feature of the compiler

Skills: SD1.5.4

Estimated commitment: 4 hours

SDS.CPPM2-SD1.5.5 EVITA module: **Compilers: Clang**

Learning outcome: Understand the working concept of the Clang Compiler

Learning outcome: Implement a simple program that user optimization feature of the compiler

Skills: SD1.5.5

Estimated commitment: 4 hours

SDS.CPPM2-SD1.5.6 EVITA module: **Compiler techniques for data level parallelism**

Learning outcome: Understand the compiers ability to directly perform data level parallelism

Learning outcome: Implement a simple working example

Skills: SD1.5.6

Estimated commitment: 4 hours

SDS.CPPM2-SD1.5.7 EVITA module: **Compiler techniques for thread level parallelism**

Learning outcome: Understand compiler techniques for additional thread level parallelism

Learning outcome: Implement a simple working example

Skills: SD1.5.7

Estimated commitment: 4 hours

6. List of courses – Specialisation: System Architect

The specialisation „System Architect" consists of the following courses and modules:

Area SA.SHT System and Hardware Technologies

SA.SHT1 EVITA course: System and Hardware Technologies

SA.SHT1-K1.4.1 EVITA module: **HPC System Components and Structure**

Learning outcome: Describe the HPC systems components: CPUs, sockets, accelerators, memories, storage, IO interconnect, buses, networking, racks, blades

Learning outcome: Describe the HPC facility components: Resources needed, power delivery, cooling solutions

Learning outcome: Describe the sustainability of HPC systems

Learning outcome: Analyse HPC facilities and their sustainability

Skills: K1.4.1

Estimated commitment: 4 hours

Prerequisites: Processor Architecture, Networking

SA.SHT1-K1.4.2 EVITA module: **HPC System Design**

Learning outcome: Plan the development of an HPC facility: components, building scheduling

Learning outcome: Prepare the sustainability plan for an HPC facility

Learning outcome: Design small HPC systems with different target objectives: data centers, research centers, accelerated systems...

Skills: K1.4.2

Estimated commitment: 4 hours

Prerequisites: HPC System Components and Structure

SA.SHT1-K1.4.3 EVITA module: **Hardware Technologies and their characteristics**

Learning outcome: Describe the characteristics of the different hardware technologies: MOS, CMOS, memories, photonic alternatives

Learning outcome: Assess Moore's Law and Dennard Scaling

Learning outcome: Describe the Power Wall and Dark Silicon

Learning outcome:Characterise fundamental blocks (e.g., ALU, register bank, memories...) at the transistor level

Skills: K1.4.3

Estimated commitment: 4 hours

Prerequisites:

SA.SHT1-K1.4.4 EVITA module: **Evaluation of the Hardware Technologies**

Learning outcome: Evaluate the reliability of a system based on a particular hardware technology: Soft errors, Hard errors, solutions

Learning outcome: Evaluate the security of a system: Root of trust, encryption primitives

Learning outcome: Evaluate particular system designs: performance, power consumption...

Skills: K1.4.4

Estimated commitment: 4 hours

Prerequisites: Hardware Technologies and their Characteristics

SA.SHT1-K1.4.5 EVITA module: **Sustainable Nanoelectronic Design**

Learning outcome: Summarise the principles of sustainability in integrated circuits (ICs)

Learning outcome: Explain the sources of power dissipation in ICs

Learning outcome: Compute the CO2 footprint of ICs

Learning outcome: Compute the societal and economic impacts of ICs

Skills: K1.4.5

Estimated commitment: 4 hours

Prerequisites: Hardware Technologies and their Characteristics

SA.SHT1-SD1.4.6 EVITA module: **Sustainable Nanoelectronic Development**

Learning outcome: Apply low-consumption techniques for ICs

Learning outcome: Apply lifetime analysis and optimization of ICs

Learning outcome: Design digital CMOS integrated circuits of medium complexity

Learning outcome: Design power control logic

Skills: SD1.4.6

Estimated commitment: 4 hours

Prerequisites: Evaluation of the Hardware Technologies, Sustainable Nanoelectronic Design

Area SA.CPUHS CPUs and Heterogenous Systems

SA.CPUHS1 EVITA course: CPUs and Heterogeneous Systems

SA.CPUHS1-K1.2.1.3 EVITA module: **Processor Design**

Learning outcome: Design an N-stage processor pipeline (e.g., fetch, decode, execute, memory access, commit)

Learning outcome: Design the memory hierarchy, instruction and data caches

Learning outcome: Design the building blocks of the processor datapath, considering the structural, control and data hazards: reservation stations, reorder buffer

Learning outcome: Design a branch predictor

Skills: K1.2.1.3

Estimated commitment: 4 hours

Prerequisites: Processor Architecture

SA.CPUHS1-K1.2.1.4 EVITA module: **Processor Development**

Learning outcome: Develop the HDL modules of a processor pipeline and datapath

Learning outcome: Develop a N-stage pipelined processor

Learning outcome: Implement the processor with an open ISA (e.g. RISC-V) on simulation

Learning outcome: Use Hardware Performance Counters and the Performance Monitoring Unit to obtain low-level details from existing hardware

Skills: K1.2.1.4

Estimated commitment: 4 hours

Prerequisites: Processor Architecture, Processor Design

SA.CPUHS1-K1.2.1.5 EVITA module: **Multiprocessor Design**

Learning outcome: Explain the core and microarchitecture support for MPs

Learning outcome: Describe the on-chip interconnect

Learning outcome: Summarise the design of multicore chips

Learning outcome: Evaluate multiprocessor designs

Skills: K1.2.1.5

Estimated commitment: 4 hours

Prerequisites: Processor Design

SA.CPUHS1-K1.2.1.6 EVITA module: **Multiprocessor Memory Hierarchy**

Learning outcome: Define the concepts of data coherency and consistency

Learning outcome: Explain the requirements for data coherency and consistency

Learning outcome: Design the memory hierarchy for multiprocessor systems

Learning outcome: Evaluate memory hierarchy designs

Skills: K1.2.1.6

Estimated commitment: 4 hours

Prerequisites: Multiprocessor Design

SA.CPUHS1-K1.2.1.7 EVITA module: **Multiprocessor Cluster Design**

Learning outcome: Design a multiprocessor cluster

Learning outcome: Compare cluster architecture alternatives

Learning outcome: Compare cluster interconnection network alternatives

Learning outcome: Evaluate CPU cluster designs

Skills: K1.2.1.7

Estimated commitment: 4 hours

Prerequisites: Multiprocessor Design, Multiprocessor Memory Hierarchy

SA.CPUHS1-K1.2.1.8 EVITA module: **High-Level Digital Design**

Learning outcome: Apply the high-level design principles of digital systems based on programmable and configurable components

Learning outcome: Breakdown a system into hardware and software components

Learning outcome: Summarise the concepts related to hardware/software communication buses and interfaces

Learning outcome: Apply the techniques for design optimisations

Learning outcome: Design custom processing subsystems (e.g., video, vector and matrix operations, sorting algorithms)

Learning outcome: Design digital communications and information processing systems

Learning outcome: Implement digital communications and information processing systems using High-Level Description Languages

Skills: K1.2.1.8

Estimated commitment: 4 hours

SA.CPUHS1-K1.2.1.9 EVITA module: **Design of Heterogeneous Systems and Accelerators**

Learning outcome: Organise the architecture of a heterogeneous system: single chip, multiple chip (chiplet) and multiple boards

Learning outcome: Design a vector unit architecture and programming

Learning outcome: Design a GPU architecture and programming

Learning outcome: Design an FPGA architecture and programming

Learning outcome: Design on-chip application-specific accelerators (e.g., stencil, AI)

Learning outcome: Design off-chip accelerators based on embedded processors and microcontrollers

Skills: K1.2.1.9

Estimated commitment: 4 hours

SA.CPUHS1-K1.2.1.10 EVITA module: **Development of Heterogeneous Systems and Accelerators**

Learning outcome: Develop algorithms and heterogeneous architectures for specific HPC applications

Learning outcome: Use acceleration on heterogeneous architectures for specific HPC applications

Learning outcome: Accelerate applications by using SIMD extensions (SSE, AVX, SVE, RISC-V V)

Learning outcome: Exploit GPUs massive parallelism for particular HPC applications (e.g., Deep Learning)

Skills: K1.2.1.10

Estimated commitment: 4 hours

Area SA.CDT Chip Design and Test

SA.CDT1 EVITA course: Chip Design and Test

SA.CDT1-K1.2.1.11.1 EVITA module: **System-on-chip Physical Design**

Learning outcome: Understand and apply timing and power constraints to a complex integrated circuit

Learning outcome: Perform the physical implementation of a complex integrated circuit

Learning outcome: Apply low power design techniques to integrated circuit design

Learning outcome: Develop techniques for the design, analysis and evaluation of electronic systems in applications such as automation, energy distribution and generation, consumer electronics, bio-medicine, etc.

Learning outcome: Analyse, design and evaluate microelectronic integrated circuits

Learning outcome: Use state-of-the-art computer aided design (CAD) tools for the design of integrated circuits

Learning outcome: Perform the physical design of a medium-sized microprocessor with realistic tools (e.g., placement, routing)

Learning outcome: Perform custom layout/routing and test (DRC, NCC)

Skills: K1.2.1.11.1

Estimated commitment: 4 hours

Prerequisites: Processor Design, Hardware Technologies and their Characteristics

SA.CDT1-K1.2.1.11.2 EVITA module: **Test of Digital Designs**

Learning outcome: Summarise the need of manufacturing test and the type of defects and faults that can be detected early on

Learning outcome: Assess the fault models

Learning outcome: Explain the fault coverage, and how observability of the faults can be improved

Learning outcome: Use automatic test pattern generators

Learning outcome: Use the techniques of scan-based tests and self-tests

Learning outcome: Develop a set of tests for simple designs

Learning outcome: Implement a system-level test for a simulated processor design

Skills: K1.2.1.11.2

Estimated commitment: 4 hours

Prerequisites: System-on-chip Digital Design

SA.CDT1-K1.2.1.11.3 EVITA module: **Design Tools and Simulators**

Learning outcome: Use hardware design tools

Learning outcome: Use HDLs (e.g., SystemC, SystemVerilog) to describe systems

Learning outcome: Apply the VLSI design flow to hardware design

Learning outcome: Use physical design tools (partitioning, chip planning, placement, global

routing, detailed routing)

Learning outcome: Develop small HDL designs to be executed on simulation or FPGAs

Learning outcome: Use optimisation techniques to improve HDL designs (data-flow graph analysis, resource allocation, loop unrolling)

Learning outcome:

Skills: K1.2.1.11.3

Estimated commitment: 4 hours

Prerequisites: Basic Parallel Programming, Processor Architecture

7. List of courses - Specialization: Performance Engineering

The specialisation „Performance Engineering" consists of the following courses and modules:

Area PE.PA Performance Analysis

PE.PA1 EVITA course: Performance Analysis

PE.PA1-PE2.4 EVITA module: **Tool-based Performance Analysis – single node**

Learning outcome: Describe the type of performance analysis tools available in single node systems: profilers, tracers, debugging, and monitoring tools

Learning outcome: Describe the type of tools oriented to performance analysis

Learning outcome: Specify the selection criteria for choosing the right tool for specific performance analysis tasks

Skills: PE2.4

Estimated commitment: 4 hours

Prerequisites: Parallel Programming, Computer Architecture, Basic Performance Analysis and Models

PE.PA1-PE2.5 EVITA module: **Practical Single-node Performance Analysis**

Learning outcome: Describe monitoring system performance and resource utilization

Learning outcome: Use profiling on parallel applications to identify performance bottlenecks

Learning outcome: Apply tracing and debugging techniques for multi-threaded applications

Skills: PE2.5

Estimated commitment: 4 hours

PE.PA1-PE4.2 EVITA module: **Tuning the Performance Analysis Environment**

Learning outcome: Create custom monitoring scripts and dashboards for HPC systems

Learning outcome: Understand strategies for optimisation based on insights gained from performance tools

Learning outcome: Analyse performance-driven decision-making for hardware upgrades or configurations

Skills: PE4.2

Estimated commitment: 2 hours

Prerequisites: Tool-based Performance Analysis - single node

PE.PA1-PE2.3.8 EVITA module: **Performance Analysis with Profiling Tools**

Learning outcome: Use profiling tools such as Nvidia Insight, Intel VTune or likwid

Learning outcome: Apply profiling tools to analyse compute and memory performance

Learning outcome: Detect overhead and bottlenecks on parallel applications

Skills: PE2.3.8

Estimated commitment: 2 hours

Prerequisites: Tuning Performance Analysis Environments

PE.PA1-PE2.6 EVITA module: **Performance Analysis with Parallel Debuggers**

Learning outcome: Use parallel debuggers such as TotalView or DDT

Skills: PE2.6

Estimated commitment: 2 hours

Prerequisites: Tuning Performance Analysis Environments

PE.PA1-PE2.7 EVITA module: **Performance Analysis with Tracing and Monitoring Tools**

Learning outcome: Use tracing tools such as Intel Trace Analyser, Scalasca, Score-P, Vampir, TAU and Paraver for understanding application behaviour

Learning outcome: Explore and apply monitoring tools like Ganglia, Nagios and Prometheus

Skills: PE2.7

Estimated commitment: 2 hours

Prerequisites: Tuning Performance Analysis Environments

PE.PA1-PE2.3.9 EVITA module: **Tool-based Performance Analysis – multi-node**

Learning outcome: Describe typical bottlenecks in distributed applications

Learning outcome: Apply profiling tools to the execution of distributed applications

Learning outcome: Apply debugging tools to distrbuted applications

Learning outcome: Apply tracing and monitoring tools to the execution of distributed applications

Skills: PE2.3.9

Estimated commitment: 4 hours

PE.PA2 EVITA course: Performance engineering tools

PE.PA2-PE2.3.1 EVITA module: **Performance engineering tools - ScoreP**

Learning outcome: Instrument applications including one or more parallel paradigms from the following: MPI, SHMEM, OpenMP, Pthreads

Learning outcome: Instrument applications including at least one specialized form of measurement from the following: Accelerator usage, I/O Performance, Hardware counters, Memory usage

Skills: PE2.3.1

Estimated commitment: 4 hours

PE.PA2-PE2.3.2 EVITA module: **Performance engineering tools - Scalasca**

Learning outcome: Automatically run Score-P instrumented measurements (including automatic analysis) of applications containing: MPI, OpenMP and/or Pthreads

Learning outcome: Identify the causing delay (imbalance) causing waiting time

Skills: PE2.3.2

Estimated commitment: 4 hours

PE.PA2-PE2.3.3 EVITA module: **Performance engineering tools - Vampir**

Learning outcome: Launch Vampir both stand-alone and connected to a VampirServer instance if available

Learning outcome: Use the summary timeline to quickly identify changed behavior over time and determine next steps for investigation

Skills: PE2.3.3

Estimated commitment: 4 hours

PE.PA2-PE2.3.4 EVITA module: **Performance engineering tools - Darshan**

Learning outcome: Understand Darshan and the librabry based data collection

Learning outcome: Instrument I/O behaviour

Skills: PE2.3.4

Estimated commitment: 4 hours

PE.PA2-PE2.3.5 EVITA module: **Performance engineering tools - PIKA**

Learning outcome: Detect pathological performance behavior: Requested memory > used memory and Requested compute resources (CPU cores, GPUs) > used compute resources

Learning outcome: Understand the resource utilization based on the application algorithm

Skills: PE2.3.5

Estimated commitment: 4 hours

PE.PA2-PE2.3.6 EVITA module: **Performance engineering tools - Lo2S**

Learning outcome: Sample applications including one or more parallel paradigms from the following: MPI (node-level), OpenMP, Pthreads

Learning outcome: Use Lo2s in application monitoring and system monitoring mode

Skills: PE2.3.6

Estimated commitment: 4 hours

PE.PA2-PE2.3.7 EVITA module: **Performance engineering tools - Nvidia Nsight Systems**

Learning outcome: Use the CLI to identify common optimization targets

Learning outcome: Generate traces for analysis in the GUI

Skills: PE2.3.7

Estimated commitment: 4 hours

PE.PA2-PE2.3.8 EVITA module: **Performance engineering tools - Nvidia Nsight Compute**

Learning outcome: Instrument a CUDA kernel and generate occupancy figures

Learning outcome: Analyse occupancy and optimize grid and block structure

Skills: PE2.3.8

Estimated commitment: 4 hours

PE.PA2-PE2.3.9 EVITA module: Performance engineering tools - Paraver, Dimemas and Extrae

Learning outcome: Trace parallel applications with Extrae

Learning outcome: Analyze parallel application bottlenecks with Paraver

Learning outcome: Predict application performance with varying computer characteristics with Dimemas

Skills: PE2.3.9

Estimated commitment: 4 hours

PE.PA3 EVITA course: Cost awareness

PE.PA3-PE1.1 EVITA module: **Cost awareness: Time to solution constraints**

Learning outcome: Remember the fundamental concepts and definitions related to time to solution constraints in performance engineering.

Learning outcome: Apply standard methods and tools to measure and evaluate time to solution in different scenarios

Skills: PE1.1

Estimated commitment: 2 hours

PE.PA3-PE1.2 EVITA module: Cost awareness: Total cost of Ownership

Learning outcome: Define the concept of Total Cost of Ownership and its relevance to performance engineering

Learning outcome: Calculate TCO for various engineering assets using structured financial models

Skills: PE1.2

Estimated commitment: 3 hours

Area PE.PM Performance Modelling

PE.PM1 EVITA course: Performance Modelling

PE.PM1-PE7.1 EVITA module: **Model-based Performance Analysis - single node**

Learning outcome: Understand the principles of analytical performance models

Learning outcome: Compare techniques for validating models against existing implementations

Learning outcome: Calibrate analytical performance models to ensure accuracy and reliability

Skills: PE7.1

Estimated commitment: 4 hours

PE.PM1-PE7.2 EVITA module: **Practical Model-Based Performance Analysis**

Learning outcome: Apply performance modeling to different types of applications: web applications, databases, parallel programs

Learning outcome: Apply modeling to workload patterns, response time and throughput

Learning outcome: Use performance modeling to identify performance bottlenecks

Learning outcome: Apply node-level performance modeling

Learning outcome: Create basic performance models

Learning outcome: Write case studies of model validation

Learning outcome: Predict bottlenecks on parallel algorithms and applications

Skills: PE7.2

Estimated commitment: 4 hours

PE.PM1-PE7.3 EVITA module: **Model-based Performance Analysis - multi-node**

Learning outcome: Apply system-level performance modeling

Learning outcome: Analyse system-level performance metrics and interactions

Learning outcome: Apply performance modeling to distributed systems, cloud sercices and HPC clusters

Learning outcome: Model and analyse scaling and load-balancing in distributed systems

Learning outcome: Leverage performance models to identify adequate optimization strategies

Learning outcome: Apply model-driven optimisation techniques to improve system performance

Learning outcome: Perform cost-benefit analysis of optimisation decisions

Skills: PE7.3

Estimated commitment: 4 hours

PE.PM2 EVITA course: Bechmarks and controlled experiments

PE.PM2-PE3.1 EVITA module: **Design and Documentation of Benchmarks**

Learning outcome: Develop comprehensive design documents that clearly outline system architecture and performance criteria

Learning outcome: Utilize diagrams and flowcharts to visually represent system designs and benchmarking processes

Skills: PE3.1

Estimated commitment: 2 hours

PE.PM2-PE3.2 EVITA module: **Controlled experiments**

Learning outcome: Comprehend that benchmarking is very essential in the HPC environment and can be applied to a variety of issues

Learning outcome: Comprehend that benchmarking is also a basis for dealing with questions emerging from tuning

Skills: PE3.2

Estimated commitment: 2 hours

PE.PM2-PE3.3 EVITA module: **Strong vs. Weak Scaling**

Learning outcome: Differentiate the two types of scaling.

Learning outcome: Interpret typical weak and strong scaling plots

Skills: PE3.3

Estimated commitment: 1 hour

Area PE.PO Performance Optimization

PE.PO1 EVITA course: Performance Engineering for Supercomputers

PE.PO1-PE5.1 EVITA module: **Single-core Performance Engineering**

Learning outcome: Understand single-core architecture and optimisation techniques

Learning outcome: Understand memory hierarchy and data access optimisations

Skills: PE5.1

Estimated commitment: 4 hours

Prerequisites:

PE.PO1-PE5.2 EVITA module: **Multicore and heterogeneous systems Performance Engineering**

Learning outcome: Perform efficient shared-memory parallelisation

Learning outcome: Evaluate parallelisation approaches for multi-core processors including GPUs

Learning outcome: Compare serial and parallel performance modeling

Learning outcome: Evaluate energy-efficient implementation and execution of parallel programs

Skills: PE5.2

Estimated commitment: 4 hours

PE.PO1-PE5.3 EVITA module: **Distributed systems Performance Engineering**

Learning outcome: Perform efficient distributed memory parallelisation

Learning outcome: Summarise advanced performance models

Learning outcome: Compare architectures of modern supercomputers

Learning outcome: Acquire an insight into innovative programming techniques and alternative supercomputer architectures

Skills: PE5.3

Estimated commitment: 4 hours

PE.PO1-PE5.4 EVITA module: **Practical Performance Engineering**

Learning outcome: Describe the performance analysis metrics

Learning outcome: Use the performance analysis metrics

Learning outcome: Implement basic numerical methods with high hardware efficiency on parallel computers

Skills: PE5.4

Estimated commitment: 4 hours

8. List of courses - Specialization: Numerical and Data Specialist for Science Domains

The specialisation „Numerical and Data Specialist" consists of the following courses and modules:

Area NDSSD.BDA Big Data Analytics

NDSSD.BDA1 EVITA course: Theoretical principles **(not available for proposals)**

NDSSD.BDA1-BDA1.1 EVITA module: **Theoretical principles: 6V**

Learning outcome: Define and understand the 6Vs of big data: Volume, Velocity, Variety, Veracity, Value, and Variability

Learning outcome: Discuss case studies that illustrate the real-world application of the 6Vs in various industries

Skills: BDA1.1

Estimated commitment: 2 hours

NDSSD.BDA1-BDA1.2 EVITA module: **AI and Data science**

Learning outcome: Understand the fundamental concepts behind AI and its role in data science

Learning outcome: Apply AI techniques to automate data processing, model complex

patterns, and predict future trends based on big data sets

Skills: BDA1.2

Estimated commitment: 2 hours

NDSSD.BDA1-BDA1.3 EVITA module: **Data Mining Techniques**

Learning outcome: Understand the fundamentals of data mining and its role in extracting valuable insights from large datasets

Learning outcome: Explore different data mining techniques such as classification, clustering, association rule mining, and anomaly detection

Skills: BDA1.3

Estimated commitment: 2 hours

NDSSD.BDA1-BDA1.4 EVITA module: **Algorithmic Foundations**

Learning outcome: Understand the role of algorithms in solving computational problems and optimizing data processing tasks

Learning outcome: Analyze the time and space complexity of algorithms using Big O notation to assess their efficiency and scalability

Skills: BDA1.4

Estimated commitment: 2 hours

NDSSD.BDA1-BDA1.5 EVITA module: **Ethical and Data Privacy Principles**

Learning outcome: Understand the ethical implications of big data analytics on individuals, communities, and society

Learning outcome: Examine privacy regulations such as GDPR, CCPA, and HIPAA and their implications for big data analytics

Skills: BDA1.5

Estimated commitment: 2 hours

NDSSD.BDA2 EVITA course: Big data tools **(not available for proposals)**

NDSSD.BDA2-BDA2.1 EVITA module: **Big data tools: Ophidia**

Learning outcome: Understand Ophidia's architecture and how it integrates with existing HPC environments to support scalable data analytics

Learning outcome: Implement data operations using Ophidia's functional interface, which includes aggregation, selection, and array manipulation

Skills: BDA2.1

Estimated commitment: 3 hours

NDSSD.BDA2-BDA2.2 EVITA module: **Big data tools: Jupyter Notebooks**

Learning outcome: Understand the architecture of Jupyter Notebooks and their integration within HPC environments to support scalable data analysis

Learning outcome: Develop and execute code efficiently in Notebooks using languages like

Python, R, and Julia

Skills: BDA2.2

Estimated commitment: 3 hours

NDSSD.BDA2-BDA2.3 EVITA module: **Big data tools: Cloud**

Learning outcome: Understand the role of cloud computing in big data analytics, identifying how cloud resources can be leveraged to process and analyze large datasets

Learning outcome: Implement data migration strategies to efficiently transfer large volumes of data to and from cloud environments

Skills: BDA2.3

Estimated commitment: 3 hours

NDSSD.BDA2-BDA2.4 EVITA module: **Big data tools: RayDP**

Learning outcome: Understand RayDP's architecture and its integration points with Apache Spark, identifying how it enhances functionality in distributed data processing

Learning outcome: Set up and configure RayDP in a HPC environment, optimizing it for specific data workflows

Skills: BDA2.4

Estimated commitment: 3 hours

NDSSD.BDA2-BDA2.5 EVITA module: **Big data tools: Spark-Horovod**

Learning outcome: Understand the integration of Spark with Horovod, recognizing how it facilitates distributed deep learning

Learning outcome: Set up and configure Spark-Horovod environments in HPC settings, ensuring optimal configuration for specific project requirements

Skills: BDA2.5

Estimated commitment: 3 hours

NDSSD.BDA2-BDA2.6 EVITA module: **Big data tools: Dask**

Learning outcome: Understands what Dask is and when to use it

Learning outcome: Distinguishes between Dask collections (arrays, dataframes, bags)

Skills: BDA2.6

Estimated commitment: 3 hours

NDSSD.BDA3 EVITA course: Storage for Big Data **(not available for proposals)**

NDSSD.BDA3-BDA3.1 EVITA module: **Parallel file systems for Big Data**

Learning outcome: Compare commonly used systems (e.g., Lustre, GPFS, BeeGFS) in terms of scalability and performance

Learning outcome: Apply best practices for managing I/O contention and metadata performance

Skills: BDA3.1

Estimated commitment: 4 hours

NDSSD.BDA3-BDA3.2 EVITA module: **Object Storage for Big Data**

Learning outcome: Explain the concept of object storage and how it differs from file/block storage

Learning outcome: Evaluate performance considerations such as consistency, latency, and throughput.

Skills: BDA3.2

Estimated commitment: 4 hours

NDSSD.BDA4 EVITA course: Workflows for Big Data **(not available for proposals)**

NDSSD.BDA4-BDA6.1 EVITA module: **Big Data analysis workflows**

Learning outcome: Define and outline the stages of a typical big data analysis workflow, from data collection to data interpretation

Learning outcome: Develop data ingestion strategies to effectively gather and store data from various sources, ensuring quality and accessibility

Skills: BDA6.1

Estimated commitment: 3 hours

NDSSD.BDA4-BDA6.2 EVITA module: **Data driven Workflows**

Learning outcome: Define and outline the stages of a typical big data analysis workflow, from data collection to data interpretation

Learning outcome: Implement data cleaning and preprocessing techniques to prepare raw data for analysis, enhancing data quality and usefulness

Skills: BDA6.2

Estimated commitment: 3 hours

NDSSD.BDA4-BDA6.3 EVITA module: **Integrated BDA with HPC Workflows**

Learning outcome: Identify the components of data-driven workflows, understanding how data is sourced, processed, and analyzed to inform decisions

Learning outcome: Design and implement workflows that integrate data collection, analysis, and interpretation seamlessly to support continuous improvement and real-time decision making

Skills: BDA6.3

Estimated commitment: 4 hours

Area NDSSD.SC Scientific Computing

NDSSD.SC1 EVITA course: Computational Numerical Linear Algebra

NDSSD.SC1-SD10.1 EVITA module: **Parallel Linear Algebra Algorithms**

Learning outcome: Describe the parallel algorithm "orthogonalization"

Learning outcome: Describe the parallel algorithm "least squares"

Learning outcome: Describe the parallel algorithm "solve linear system"

Skills: SD10.1

Estimated commitment: 4 hours

NDSSD.SC1-SD10.2 EVITA module: **Stability of Linear Algebra Algorithms**

Learning outcome: Describe the numerical stability of an algorithm

Learning outcome: Determine the stability of the algorithm "orthogonalization"

Learning outcome: Determine the stability of the algorithm "least squares"

Learning outcome: Determine the stability of the algorithm "solve linear system"

Learning outcome: Explain how to reach a compromise between stability and level of parallelism

Skills: SD10.2

Estimated commitment: 4 hours

Prerequisites: Parallel Linear Algebra Algorithms

NDSSD.SC1-SD10.3 EVITA module: **Communication requirements of Linear Algebra Algorithms**

Learning outcome: Compute the communication requirements of a parallel algorithm

Learning outcome: Determine the communication requirements of the algorithm "orthogonalization"

Learning outcome: Determine the communication requirements of the algorithm "least squares"

Learning outcome: Determine the communication requirements of the algorithm "solve linear system"

Skills: SD10.3

Estimated commitment: 4 hours

NDSSD.SC1-SD10.4 EVITA module: **Linear Algebra Algorithms for Large Data Analysis**

Learning outcome: Describe the relationship between matrices and tensors

Learning outcome: Implement the transformation of algorithms using matrices to tensors

Skills: SD10.4

Estimated commitment: 2 hours

NDSSD.SC1-SD10.5 EVITA module: **Performance of Linear Algebra Algorithms**

Learning outcome: Analyse the performance of Linear Algebra Algorithms on shared memory multiprocessors

Learning outcome: Analyse the performance of Linear Algebra Algorithms on distributed

memory systems

Learning outcome: Analyse the performance of Linear Algebra Algorithms on hybrid systems (shared and distributed memory)

Skills: SD10.5

Estimated commitment: 2 hours

Area NDSSD.QC Quantum Computing

NDSSD.QC1 EVITA course: Quantum Computing **(not available for proposals)**

NDSSD.QC1-QC1.1 EVITA module: **Qubits and Quantum Computers**

Learning outcome: Describe the Qubit concept

Learning outcome: Describe the properties of Qubits

Learning outcome: Describe the measurement techniques for Qubits

Learning outcome: Describe the different types of Quantum Computers

Learning outcome: Compare the different types of Quantum Computers

Learning outcome: Summarise circuit models and mathematical definitions of quantum gates

Skills: QC1.1

Estimated commitment: 4 hours

NDSSD.QC1-QC6 EVITA module: **Quantum Computing and HPC**

Learning outcome: Apply programming models for Quantum Computing environments: Qiskit, Ocean, Cirq

Learning outcome: Analyse QC applications from material science and physics, optimization and financial problems

Learning outome: Implement AI/ML or optimization problems in QC using programming models for QC environments

Skills: QC6

Estimated commitment: 4 hours

Area NDSSD.DSHPC Domain-Specific High Performance Computing

NDSSD.DSHPC1 EVITA course: Domain-Specific Approaches

NDSSD.DSHPC1-SD11.1 EVITA module: **Overview of Domain-Specific Architectures**

Learning outcome: Describe Edge and Embedded architectures

Learning outcome: Describe domain-specific parallel architectures

Learning outcome: Describe the FPGA architecture

Learning outcome: Describe the Tensor Programming Unit (TPU) architecture

Learning outcome: Describe the Intelligence Processing Unit (IPU) architecture

Skills: SD11.1

Estimated commitment: 4 hours

Prerequisites: Basic Computer Architecture

NDSSD.DSHPC1-SD11.2 EVITA module: **Domain-Specific Programming**

Learning outcome: Describe the programming environments on the Edge platforms

Learning outcome: Describe the programming environments on Embedded systems

Learning outcome: Describe the application design flow on domain-specific environments

Learning outcome: Implement AI/ML or BLAS algorithms on domain-specific environments

Learning outcome: Evaluate the performance and power consumption

Skills: SD11.2

Estimated commitment: 4 hours

Prerequisites: Overview of Domain-Specific Architectures

NDSSD.DSHPC2 EVITA course: Numerical Simulations on HPC systems

NDSSD.DSHPC2-SD11.3 EVITA module: **Fluid-Dynamics Numerical Simulations**

Learning outcome: Explain physical modeling of fluid-dynamics applications

Learning outcome: Implement numerical methods

Learning outcome: Analyse the mathematical properties of numerical methods

Learning outcome: Use available fluid-dynamics software packages for simulation

Learning outcome: Develop fluid-dynamics applications

Learning outcome: Parallelise fluid-dynamics applications

Skills: SD11.3

Estimated commitment: 4 hours

Prerequisites: Parallel Linear Algebra Algorithms, Numerical Analysis of Finite Differences Schemes

NDSSD.DSHPC2-SD11.4 EVITA module: **Solid Mechanics Numerical Simulations**

Learning outcome: Explain physical modeling for solid mechanics

Learning outcome: Use available solid mechanics software packages for simulation

Learning outcome: Develop solid mechanics applications

Learning outcome: Parallelise solid mechanics applications

Skills: SD11.4

Estimated commitment: 4 hours

Prerequisites: Parallel Linear Algebra Algorithms, Numerical Analysis of Finite Differences Schemes

Area NDSSD.AI AI/ML

NDSSD.AI1 EVITA course: AI/ML **(not available for proposals)**

NDSSD.AI1-BDA5.6 EVITA module: **Artificial Intelligence and HPC**

Learning outcome: Design artificial neural networks (perceptrons, backpropagation, stochastic gradient descent, deep learning, recurrent nets)

Learning outcome: Combine different machine learning methods for solving actual problems in natural sciences

Learning outcome: Implement advance topics in model evaluation and selection, anomaly detection, conformal learning (prediction with guarantees of accuracy), and causal inference (identification of causal relationships)

Learning outcome: Perform test problem implementations focusing on CPU (multicore or HPC cluster) and GPU/GPUs (with inter and intra node settings)

Learning outcome: Implement selected problems from science and engineering using existing software frameworks; for example, Tensorflow, PyTorch, in parallel architectures using CPUs and GPUs.

Skills: BDA5.6

Estimated commitment: 4 hours

NDSSD.AI1-BDA5.4 EVITA module: **Machine Learning and HPC**

Learning outcome: Understand supervised learning, Bayes classifier, perceptron, support vector machines, K-nearest neighbor models, regression and logistic regression

Learning outcome: Summarise maximum likelihood estimation and Bayesian methods

Learning outcome: Undeerstand unsupervised learning: clustering algorithms, EM algorithm, mixture models, model selection and kernel methods

Learning outcome: Compare deep learning models such as fully connected neural networks, convolutional neural networks, and recurrent neural networks

Learning outcome: Perform test problem implementations focusing on CPU (multicore or HPC cluster) and GPU/GPUs (within inter and intra node settings)

Learning outcome: Implement selected problems from science and engineering using existing frameworks; for example, Tensorflow, PyTorch, in parallel architectures using CPUs and GPUs

Skills: BDA5.4

Estimated commitment: 4 hours

Prerequisites: Artificial Intelligence and HPC

NDSSD.AI1-BDA8 EVITA module: **Computational Statistics**

Learning outcome: Understand random number generation: inversion method, transformations and rejection sampling.

Learning outcome: Understand Gaussian processes and Karhunen-Loève expansion.

Learning outcome: Implement and analyse Monte Carlo methods, estimates, and errors; multi-level (or multifidelity) Monte Carlo methods.

Learning outcome: Compare variance reduction techniques.

Learning outcome: Understand sensitivity analysis of multiple parameters: one-at-a-time, elementary effect methods, and variance-based methods.

Learning outcome: Summarise surrogate models or emulators: polynomial chaos expansions and reduced-order models.

Learning outcome: Analyse uncertainty propagation through Monte Carlo methods and their applications to differential models.

Learning outcome: Apply statistical inverse problems: a Bayesian setting for parameter estimation.

Learning outcome: Analyse Monte Carlo Markov chain methods and their application to differential models.

Learning outcome: Understand data assimilation: Kalman filters and particle filters and their application to differential models.

Learning outcome: Implement selected test problems in parallel using CPU and GPU.

Skills: BDA8

Estimated commitment: 4 hours

9. List of courses - Software Engineering

Area SE.SEP Software Engineering Practices

SE.SEP1 EVITA course: Programming best practices **(not available for proposals)**

SE.SEP1-SD2.1 EVITA module: **Programming best practice: IDE**

Learning outcome: Configure and use integrated development environments (IDEs) like Eclipse, e.g. to seamlessly perform the typical development cycle with the steps edit, build (compile and link), and test

Skills: SD2.1

Estimated commitment: 2 hours

SE.SEP1-SD2.2 EVITA module: **Programming best practice: Debugging**

Learning outcome: Debug a program using simple techniques such as inserting debugging output statements into the source code, e.g. using printf – also against the background of potential problems with the ordering of the (stdout) output that may exist in parallel environments like MPI

Learning outcome: Use sophisticated debuggers such as GDB

Skills: SD2.2

Estimated commitment: 4 hours

SE.SEP1-SD2.3 EVITA module: **Programming best practice: Programming Idioms**

Learning outcome: Describe programming idioms for a specific programming language, e.g. Fortran, Python, C, C++

Learning outcome: Recognize where programming idioms are violated and to refactor the code to comply to a specific programming idiom

Skills: SD2.3

Estimated commitment: 3 hours

SE.SEP1-SD2.4 EVITA module: **Programming best practice: Logging**

Learning outcome: Describe logging in general like log levels etc. (e.g. ERROR, WARN, INFO, DEBUG, TRACE)

Learning outcome: Select appropriate information that should be logged e.g. timestamp, pid, thread, level, loggername) in order to be able to identify the problem

Skills: SD2.4

Estimated commitment: 3 hours

SE.SEP1-SD2.5 EVITA module: **Programming best practice: Exception Handling**

Learning outcome: Differentiate among the terms “mistake”, “fault”, “failure”, and “error”

Learning outcome: Describe exception handling concepts in general (e.g. Errors vs. Exceptions)

Skills: SD2.5

Estimated commitment: 3 hours

SE.SEP1-SD2.6 EVITA module: **Programming best practice: Coding Standards**

Learning outcome: Understand and apply coding standards

Learning outcome: Differentiate between differet styles

Skills: SD2.6

Estimated commitment: 2 hours

SE.SEP1-SD2.7 EVITA module: **Programming best practice: Testing**

Learning outcome: Understand testing procedures

Learning outcome: Implement a set of test for a simple arithemtic function

Skills: SD2.7

Estimated commitment: 2 hours

SE.SEP1-SD2.8 EVITA module: **Programming best practice: Portability**

Learning outcome: Understand portability in terms of linked libraries

Learning outcome: Understand code that can be used on different OS

Skills: SD2.8

Estimated commitment: 2 hours

SE.SEP1-SD2.9 EVITA module: **Programming best practice: Scalability**

Learning outcome: Understand problem that can be solved using scalable code

Learning outcome: Implement a task distribution worker that scales depending on the number of request to the service

Skills: SD2.9

Estimated commitment: 3 hours

SE.SEP2 EVITA course: Version controll and issue tracking **(not available for proposals)**

SE.SEP2-SD3.1 EVITA module: **Version control**

Learning outcome: Describe the basics of version control systems, e.g. what is version control

Learning outcome: Discuss the benefits of using version control for software development especially in a team

Skills: SD3.1

Estimated commitment: 2 hours

SE.SEP2-SD3.2 EVITA module: **Issue Tracking and Bug Tracking**

Learning outcome: Describe concepts of issue/bug tracking systems and their basic concepts like task, sub-task, new feature, story, release planning, sprint planning in order to structure and organize the development process (e.g. assigning tasks to developers, reporting bugs, writing user stories, managing the stages of an issue (to do, in progress, in review, done) etc.)

Learning outcome: Apply issue tracking in order to manage tasks, bug reports, and other issues occuring during development and enabling task assignment in the team

Skills: SD3.2

Estimated commitment: 4 hours

SE.SEP2-SD3.3 EVITA module: **Release management**

Learning outcome: Describe the basics of release management and what the benefits are of applying a release management process in the context of high-performance computing “fault”, “failure”, and “error”

Learning outcome: Discuss the differences among Major Release, Minor Release, Emergency Fix (and potentially other types of releases) and what should be contained in each of them

Skills: SD3.3

Estimated commitment: 3 hours

SE.SEP2-SD3.4 EVITA module: **Deployement Management**

Learning outcome: Discuss why different environments for testing, development, production, and staging are necessary

Learning outcome: Compare the differences between different deployment environments, and what are their specific requirements

Skills: SD3.4

Estimated commitment: 4 hours

SE.SEP3 EVITA course: Tests and programming paradigm **(not available for proposals)**

SE.SEP3-SD4.1 EVITA module: **Test driven development and agile testing**

Learning outcome: Describe the challenges of testing scientific applications in the contect of HPC applications

Learning outcome: Characterize what constitutes a test strategy

Skills: SD4.1

Estimated commitment: 2 hours

SE.SEP3-SD4.2 EVITA module: **Extreme Programming**

Learning outcome: Discuss the principles of extreme programming and when to apply it

Learning outcome: Apply the principles in the context of an HPC project

Skills: SD4.2

Estimated commitment: 2 hours

SE.SEP3-SD4.3 EVITA module: **SCRUM**

Learning outcome: Desribe the concepts of SCRUM, e.g. Sprint, Backlog, Planning, Daily meetings/Stand up meeting, and project velocity

Learning outcome: Apply practices of SCRUM

Skills: SD4.3

Estimated commitment: 4 hours

SE.SEP4 EVITA course: Software quality **(not available for proposals)**

SE.SEP4-SD5.1 EVITA module: **Software Quality and conding standards**

Learning outcome: Apply software engineering methods and practices especially in the context of high-performance computing

Learning outcome: Develop parallel programs and to apply software engineering methods and best practices

Skills: SD5.1

Estimated commitment: 2 hours

SE.SEP4-SD5.2 EVITA module: **Code refactoring**

Learning outcome: Apply common code refactorings in order to improve code quality, such as extract method, extract class, rename class and when it is suitable to apply which refactoring

Learning outcome: Apply refactoring that are specific to programming languages (e.g. Fortran)

Skills: SD5.2

Estimated commitment: 2 hours

SE.SEP4-SD5.3 EVITA module: **Code Review**

Learning outcome: Use a review system like Gerrit to organize the code reviews

Learning outcome: Document code review results and resulting tasks in an issue tracking system (for example Jira)

Skills: SD5.3

Estimated commitment: 2 hours

SE.SEP5 EVITA course: Software design and architecture **(not available for proposals)**

SE.SEP5-SD6.1 EVITA module: **Requirements Elicitation and Analysis**

Learning outcome: Describe the role of requirements elicitation and analysis during software development

Learning outcome: Identify functional and non-functional requirements of the software to be implemented

Skills: SD6.1

Estimated commitment: 2 hours

SE.SEP5-SD6.2 EVITA module: **Object-Oriented Design**

Learning outcome: Discuss and use the main concepts of object-orientation (classes, interfaces, polymorphism) to design and implement a program

Learning outcome: Write modular, reusable code by applying software design principles like Separation of concerns, loose coupling, information hiding, DRY, KISS etc. (following best practices like Clean Code by Robert C. Martin)

Skills: SD6.2

Estimated commitment: 3 hours

SE.SEP5-SD6.3 EVITA module: **Software Architecture of existing projects**

Learning outcome: Describe the importance and impact of software architecture during software development

Learning outcome: Analyze software architecture principles

Skills: SD6.3

Estimated commitment: 2 hours

SE.SEP5-SD6.4 EVITA module: **Software Architecture design from the ground up**

Learning outcome: Design the application as a plugin architecture so that the functionality can be extended more easily

Learning outcome: Design the software architecture of the system based on software architecture patterns

Skills: SD6.4

Estimated commitment: 4 hours

SE.SEP6 EVITA course: Documentation of software and deployment **(not available for proposals)**

SE.SEP6-SD7.1 EVITA module: **Requirements Documentation**

Learning outcome: Describe which information needs to be captured in a requirements document

Learning outcome: Apply the IEEE standard for software requirements specification for a structured requirement specification

Skills: SD7.1

Estimated commitment: 2 hours

Prerequisites: Shared-memory programming, Distributed-memory programming

SE.SEP6-SD7.2 EVITA module: **Software architecture and design documentation**

Learning outcome: Document the different views of the software architecture according to a specific documentation framework, e.g. 4+1 views, Views and Beyond, architecture decision frameworks (e.g. Taylor, Olaf Zimmermann)

Learning outcome: Apply a modeling language for documenting the design and the architecture, e.g. Unified Modeling Language (UML)

Skills: SD7.2

Estimated commitment: 2 hours

Prerequisites: Requirements documentation

SE.SEP6-SD7.3 EVITA module: **Source Code Documentation**

Learning outcome: Document source code using documentation generators like doxygen, pydoc, or sphinx

Learning outcome: Produce a consistent source code documentation according to guidelines and best practices

Skills: SD7.3

Estimated commitment: 2 hours

Prerequisites:

SE.SEP6-SD7.4 EVITA module: **Documentation for Reproducibility**

Learning outcome: Document all necessary information for end-users so that they are able to reproduce the results

Learning outcome: Use tools for literate programming like activepapers knitr, or jupyter to document all necessary information for end-users so that they are able to reproduce the results especially in the context of concurrency

Skills: SD7.4

Estimated commitment: 3 hours

Prerequisites:

10. Conclusions

In this report, we provide the list of courses that are currently incorporated in the EVITA CQF.

11. References

[1] **EUMaster4HPC**, The European Master for HPC, https://eumaster4hpc.eu, accessed 17/12/2025.

[2] **The European Master for HPC Curriculum** *Pascal Bouvry, Mats Brorsson, Ramon Canal, et al.* Journal of Parallel and Distributed Computing, Volume 201, July 2025, 105081. https://doi.org/10.1016/j.jpdc.2025.105081