Electives
Master Automotive Software Engineering
Summer Semester

Faculty Computer
Science Date: 12.02.2025

Table of Contents

Technische Hochschule Deggendorf
Dieter-Görlitz-Platz 1
94469 Deggendorf

Tel.: +49 991 3615-0
Fax: +49 991 3615-297

www.th-deg.de
info@th-deg.de

Electives
Artificial Intelligence and Data Science
Summer Semester

Faculty Computer
Science Date: 12.02.2025

Generell Information

This module handbook applies to electives offered in summer semester 2025. The listed modules can be chosen for Elective 1 or 2. For the winter semester, there is a separate list/module handbook

Important: If German is specified as the language of instruction for an elective course, then the examination will also be conducted in German!

Technische Hochschule Deggendorf
Dieter-Görlitz-Platz 1
94469 Deggendorf

Tel.: +49 991 3615-0
Fax: +49 991 3615-297

www.th-deg.de
info@th-deg.de

AIX-M-2 Datacenter Network Programming

Module code	AIX-M-2
Module coordination	Prof. Dr. Andreas Kassler
Course number and name	AIX-2 Datacenter Network Programming
Lecturer	Prof. Dr. Andreas Kassler
Semester	2
Duration of the module	1 semester
Module frequency	annually
Course type	compulsory course
Level	
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 90 hours self-study: 60 hours Total: 150 hours
Type of Examination	written ex. 90 min.
Duration of Examination	90 min.
Weighting of the grade	5/210
Language of Instruction	English

Module Objective

Students acquire understanding and hands-on experience of how the data plane of modern datacenter networking equipment can be programmed using the high-level and popular programming language P4 (see http://p4.org). They learn the basic concepts of the P4 language and understand, how offloading simple computational tasks to the data plane of programmable networking devices (such as datacenter routers or network cards) can be used to speed up the performance of Deep Learning, Big Data Analytics use-cases within modern datacenters. They understand, how the data plane can be used to accelerate distributed high-performance computing (HPC) building blocks including distributed key-value stores, where load-balancing and network monitoring of the datacenter networking fabric is important for achieving high speed and low latency.

They setup their own development environment in the network emulator Mininet and implement simple data plane programs in the P4 language. They know how to use P4 to parse packet headers, apply different actions and modify packets before forwarding them. They know basic P4 constructs, how to store stateful information (e.g. parts of a neural network) and how to perform simple computational tasks in the data plane.
Based on this knowledge and understanding, students implement a small-scale project in a team. They use their acquired knowledge on P4 and programmable datacenter networking. They evaluate the results of other project groups and get evaluated by other groups. For this project work, they have used standard tools (Mininet, P4 toolchain, command line interface) for programming the data plane of an (emulated) datacenter router.
After finishing this module, students can design, implement and evaluate their own P4 programs using the network emulator Mininet.

Specifically, students will have achieved the following outcomes upon completion of the module:
Subject competency
Students understand the significance of datacenter network programming and how datacenter network programming can be used to improve the performance of distributed high-performance applications such as distributed training and inference of large-scale ML models.
Methodological competency
Students select the most appropriate P4 constructs for solving a concrete practical problem and use it to implement a concrete use-case of a data plane program.
Personal competency
Students understand complex theoretical concepts and apply them to problems arising in practice.
Social competency
Students communicate clearly, argue and criticize logically and constructively, contribute to reasoned, team-oriented problem-solving processes in the group.

Applicability in this and other Programs

This Module is suitable for the following programs:
- Master in Angewandte Informatik/Infotronik
- Master in Artificial Intelligence and Data Science
- Master in High Performance Computing/Quantum Computing

Entrance Requirements

Students should have basic understanding of Network Technologies and/or Communication Networks. Basic knowledge of Programming and basic knowledge in Python helps in the Project Part of the course.

Learning Content

The Module is decomposed into two parts:
Part I: ?Introduction to Datacenter Network Programming? and Part II ?Project in Datacenter Network Programming?
Content Part I:
(1) Introduction to Programming the Data Plane of a Datacenter networking device:
- Difference between Data and Control Plane
- Introduction to P4 language
- P4 programming model
- Compiling and deploying P4 programs
- P4 Targets: Behavioral Model (BMv2), Programmable Switching ASIC Intel Tofino, Mellanox Bluefield DPU, Netronome SmartNIC
- Basic P4 concepts: header parsing, applying tables and actions, header rewriting.
- Workshop: Setup Development environment with Mininet and Command Line Interface (CLI), implement, test and debug simple P4 language constructs and programs using the Mininet network emulator
(2) Datacenter Networking and Load Balancing:
Faculty Computer Science
Artificial Intelligence and Data Science
Date: 22.11.2022
- Datacenter networking fundamentals, routing and forwarding within the datacenter networking fabric
- Workshop: Advanced P4 concepts: stateful information, register arrays, counters and meters.
- Loadbalancing in Datacenter networks, Equal Cost Multipath Routing, Conga, Hula
- Workshop: Implementing ECMP in P4
(3) In Network support for Monitoring and Caching:
- Active and passive network monitoring
- Inband Network Telemetry (INT) for fine-granular network monitoring
- Accelerating Distributed Key-value stores in the data plane of the data center
- Using telemetry for fine-grained loadbalancing
- Workshop: Implementing Hula and INT in P4
(4) In Network support for Distributed Machine Learning:
- Role of the datacenter network for distributed training and inference
- In network support for Distributed Machine Learning Inference for in-switch traffic classification
- Mapping trained machine learning models (decision trees, SVMs, neural networks) to programmable data plane devices
- In network support for distributed training within a datacenter network
Content Part II:

Project: Implementation of your own small dataplane program in P4 and testing it in the Mininet network emulator.

Teaching Methods

- Interactive lectures
- Practical workshop style exercises using the network emulator Mininet
- Software implementation of your own P4 program

Remarks

The module is comprised of two parts. The second part (project work) can be done in groups of max. 3 students.

Recommended Literature

Recommended Literature will be provided at the start of the course by a set of research and practical oriented articles that are available online.

LSI-12 Data Visualization

Module code	LSI-12
Module coordination	Prof. Dr. Phillipp Torkler
Course number and name	LSI-12 Data Visualization
Lecturers	Prof. Dr. Phillipp Torkler Prof. Dr. Javier Valdes
Semester	2
Duration of the module	1 semester
Module frequency	annually
Course type	required course
Level	Postgraduate
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 45 hours virtual learning: 45 hours Total: 150 hours
Type of Examination	written student research project
Weighting of the grade	5/90
Language of Instruction	English

Module Objective

Data Visualization is the graphic representation of a data analysis to achieve clear and effective communication of results and insights. Complex ideas are presented in charts and graphs with the goal of quickly and easily disseminating key, actionable information. Data visualization is an essential part of data science and analytics, especially when working with large, complicated data sets like sequencing data. The visualization tells a story, whether as a stand-along graph or combined with other graphs, charts and design elements in an infographic or dashboard.

After completing the Data Visualization module, students will have obtained the following learning competencies:

Professional competence

After successfully completing the module, students will:

- know the data visualization principles.
- be familiar with file formats and their usage in the different analysis approaches.
- know about common data analysis workflows and be able to interpret and visualize the achieved results.

Methodological competence

After successfully completing the module, students will:

- know how to use ggplot2 in R to create custom plots.
- know how to use matplotlib and Python to create custom plots.

Social competence

- Interdisciplinary and interpersonal collaboration when working together in small groups on developing R and Python scripts for data analysis and data visualization.
- Working together with fellow-students in small groups on designing and developing biostatistical validation of biomedical datasets within R and/or Python.

Applicability in this and other Programs

master seminar, master thesis

Entrance Requirements

Recommended or advantageous:

Basic Knowledge in R

Module: LSI-04 *Biostatistics I*

Learning Content

1 R Packages for data visualization
2 Open access visualization tools
3 Matplotlib and other Python packages for data visualization
4 Theoretical Background
5 Perception And Interpretation

Teaching Methods

Tutorial, practical exercises, application examples

The module consists of an interactive theoretical part with blended learning components. Within the tutorial the students use example NGS datasets to perform the biomedical data visualization. In the practical part of the tutorial the students should learn to find various visualization tools, possibilities and methods and discuss their advantages and disadvantages to represent statisitical significance.

Remarks

The iLearn teaching and learning platform provides students with additional literature references and learning material to prepare for the lectures.

Recommended Literature

Detailed lecture notes are available online for preparation and follow-up work

- The Biostars Handbook: Bioinformatics Data Analysis Guide; 2019; https://www.biostarhandbook.com/

HPC-07 HPC/QC Technology

Module code	HPC-07
Module coordination	Prof. Dr. Helena Liebelt
Course number and name	HPC-7 HPC/QC Technology
Lecturer	Prof. Dr. Helena Liebelt
Semester	1
Duration of the module	1 semester
Module frequency	annually
Course type	required course
Level	postgraduate
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	StA
Weighting of the grade	5/90
Language of Instruction	English

Module Objective

In this module, students delve into the intricacies of High Performance Computing (HPC) and/or Quantum Computing (QC) systems, gaining a comprehensive understanding of the technological challenges specific to these cutting-edge fields. Through a blend of theoretical learning and hands-on experience, students familiarize themselves with the hardware technologies essential to these domains. Through practical sessions, they acquire invaluable skills in system assembly and configuration, empowering them to proficiently set up components of modern HPC systems. Additionally, students learn the nuances of system installation and configuration on a smaller scale, equipping them with the capability to effectively deploy and manage these advanced computing systems.

Professional skills

Professional Skills: Students develop a range of professional skills essential for success in the field of High Performance Computing (HPC) and/or Quantum Computing (QC) systems. They refine their ability to analyze complex technological issues specific to these domains, employing critical thinking and problem-solving techniques to address challenges effectively. Through hands-on experience in system assembly and configuration, students enhance their technical proficiency, ensuring they are adept at deploying and managing modern HPC systems. Furthermore, students cultivate strong communication skills, enabling them to articulate their ideas and solutions clearly to peers and industry professionals.

Methodological skills

This module hones students' methodological skills, providing them with a structured framework for approaching problems in HPC and/or QC systems. Students learn systematic approaches to system setup, installation, and configuration, ensuring precision and efficiency in their work. They develop robust methodologies for troubleshooting and debugging, equipping them with the ability to identify and resolve issues promptly. Additionally, students learn to conduct thorough research, staying abreast of the latest advancements in hardware technologies relevant to the field.

Social skills

In addition to technical expertise, students refine their social skills, recognizing the collaborative nature of the HPC and/or QC ecosystem. Through group projects and collaborative tasks, students learn to work effectively as part of a team, leveraging each other's strengths to achieve common goals. They develop interpersonal skills such as active listening, constructive feedback, and conflict resolution, fostering a positive and productive team dynamic. Furthermore, students engage in networking opportunities with industry professionals, enhancing their ability to build and maintain professional relationships within the field.

Personal skills

This module also focuses on the development of personal skills essential for professional growth and success. Students cultivate traits such as adaptability and resilience, learning to navigate the dynamic landscape of HPC and/or QC systems with confidence. They hone their time management and organization skills, balancing academic coursework with hands-on practical sessions effectively. Additionally, students foster a growth mindset, embracing challenges as opportunities for learning and growth. By prioritizing self-reflection and continuous improvement, students develop into well-rounded individuals prepared to excel in the rapidly evolving field of high-performance computing.

Applicability in this and other Programs

Hardware / system design for complex modern computing systems

Entrance Requirements

Prerequisites for this module on a master's level would typically include:

1. Foundational Knowledge in Computer Science or Related Field: Students should have a solid understanding of computer science fundamentals, including data structures, algorithms, computer architecture, and operating systems.
2. Programming Proficiency: Proficiency in at least one programming language commonly used in high-performance computing, such as C/C++, Python, or Fortran, is essential. Students should be comfortable writing, debugging, and optimizing code.
3. Mathematical Background: A strong background in mathematics, particularly in areas such as linear algebra, calculus, and probability theory, is necessary for understanding the underlying principles of high-performance computing and quantum computing.
4. Understanding of Parallel Computing Concepts: Familiarity with parallel computing concepts and techniques is crucial, including parallel algorithms, parallel programming models (e.g., MPI, OpenMP, CUDA), and parallel computing architectures.
5. Basic Knowledge of Hardware Systems: Students should have a basic understanding of computer hardware components and architecture, including processors, memory systems, storage devices, and networking.
6. Prior Experience with Operating Systems: Familiarity with operating systems concepts and administration is beneficial, as students will be involved in setting up and configuring computing systems.
7. Experience with Command-Line Interface (CLI) Tools: Proficiency in using command-line interface tools and Unix/Linux operating systems is important for executing commands, managing files, and interacting with the computing environment.
8. Probability and Statistics: Some familiarity with probability and statistics is advantageous, especially for students interested in quantum computing, as it provides the foundation for understanding quantum algorithms and quantum information theory.
9. Critical Thinking and Problem-Solving Skills: Strong critical thinking and problem-solving skills are essential for analyzing complex technological issues and developing effective solutions in the context of high-performance and quantum computing systems.
10. Research Skills: Students should possess basic research skills, including the ability to gather, evaluate, and synthesize information from academic literature, technical documentation, and online resources related to high-performance and quantum computing.

These prerequisites ensure that students have the necessary background knowledge and skills to fully engage with the advanced topics covered in the module and to successfully complete hands-on practical sessions and assignments.

Learning Content

The aim of this course is to discover the technological paticularities of HPC and QC systems.

The module is divided into two parts, both covering theoretical as well as practical aspects including hands-on sessions:

- Hardware
  - Setting up a compute node
  - Rack technologies
  - Cooling aspects
- Software
  - Setting up an operating system
  - Middleware
  - Access and scheduling

Teaching Methods

Lecture with lab sessions / exercises

Recommended Literature

- Andrew S. Tanenbaum; Herbert Bos. Modern Operating Systems. Prentice Hall, 4th ed. 2014
- Evi Nemeth, Garth Snyder, Trent R. Hein et al. Unix and Linux System Administration Handbook. Addison-Wesley, 5th ed. 2018
- Christine Bresnahan, Richard Blum. Mastering Linux system administration. Wiley. 2021. https://ebookcentral.proquest.com/lib/th-deggendorf/detail.action?docID=6658986
- Further literature as indicated in the lecture

AIX-M-16 ChatGPT et al.: Generative AI with Transformers

Module code	AIX-M-16
Module coordination	Prof. Dr. Andreas Fischer
Course number and name	AIX-M-16 ChatGPT et al.: Generative AI with Transformers
Lecturers	Zineddine Bettouche Prof. Dr. Andreas Fischer
Semester	2
Duration of the module	1 semester
Module frequency	annually
Course type	compulsory course
Level	
Semester periods per week (SWS)	4
ECTS	5
Workload	Time of attendance: 60 hours self-study: 90 hours Total: 150 hours
Type of Examination	written student research project
Weighting of the grade	
Language of Instruction	English

Module Objective

Entrance Requirements

AIX-M-16 ChatGPT et al.: Generative AI with Transformers

Entrance Requirements

Substantial background in artificial intelligence

Learning Content

The module will give an introduction to the transformer technology which drives modern large language models. Covered topics are:

- Foundations of Language Models
- Word Embeddings
- Attention Mechanism
- Architectures of Transformer Models
- Popular Open Source Transformer Models
- Limitations of Large Language Models
- Applications of Transformers in and beyond NLP
- Optimization of Transformer Models

Type of Examination

written student research project

Methods

Seminaristic education

Recommended Literature

- Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
- Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

- Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).