University of Basel

A Glimpse Into the **High Performance Computing** Group's **Research Activities** at DMI

hpc.dmi.unibas.ch | dmi-hpc@unibas.ch | dmi-hpc.bsky.social | hpc-dmi-unibas

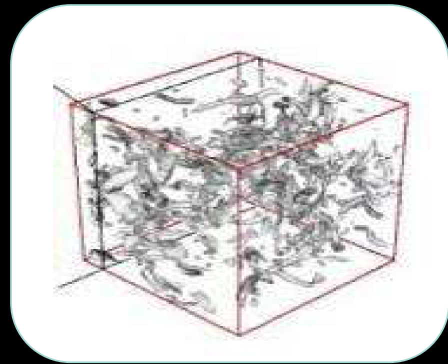
since 08/2015

Updated: 17.12.2024

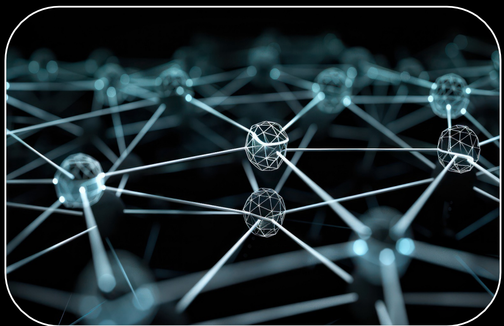
Teaching

Operating Systems
BSc course, 8 CP
Spring semester

Reproducibility and Performance of Privacy Enhancing Technologies
BSc seminar, 6 CP (with Isabel Wagner)
Fall semester

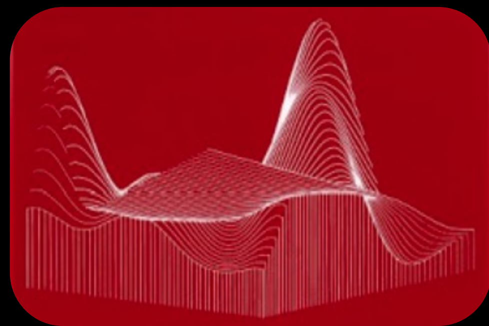


Applications of Computational Sciences
BSc course, 2 CP (with others)
Fall semester



Foundations of Distributed Systems
MSc course, 8 CP
(with I. Wagner, C. Tschudin, H. Schuldt)
Fall semester

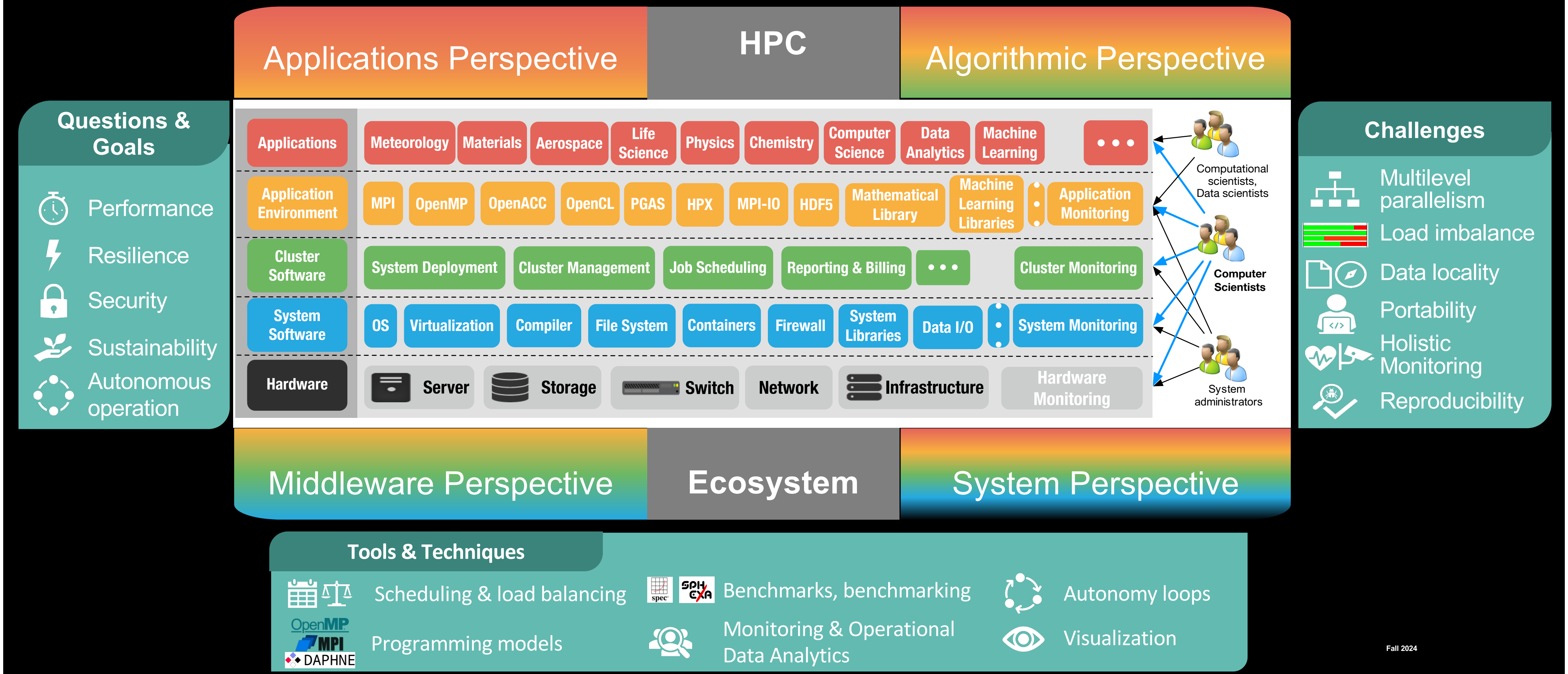
High Performance Computing
MSc course, 4 CP (soon 6CP)
Spring semester



Advanced Methods in Computational Sciences
PhD course, 3 CP (with others)
Spring semester

High Performance Computing Group at DMI

"How to exploit all parallelism efficiently across levels and devices?"



Fall 2024

Topics for Bachelor Theses

Selection (December 2024)

1. Porting Benchmarks and Applications to Trusted Execution Environments
2. Study of Benefits, Challenges, Security Properties of SELinux vs AppArmor on Ubuntu/Rocky Linux
3. Single Board Computers with Neural Compute Sticks for Machine Learning Model Inference
4. Continuous Performance Evaluation of Astrophysical Applications Through CI/CD Pipelines
5. CPU and GPU Frequency Sensitivity Analysis of HPC Workloads
6. Integrating Zoltan into SPH-EXA for Dynamic Load Balancing Studies
7. Explore LLM-as-a-Judge for Verification and Validation of OpenACC & OpenMP Scheduling
8. Automated Survey of Academic Papers through Local Large Language Models

1. Porting Benchmarks and Applications to Trusted Execution Environments

Context

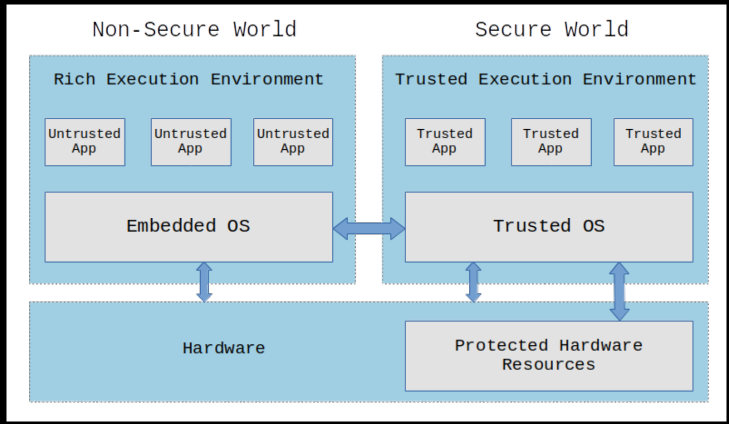
- Trusted Execution Environments (TEEs) provide secure and isolated execution environments for applications, protecting sensitive code and data from untrusted system components.
- Increasingly used in HPC, clouds, security-critical domains

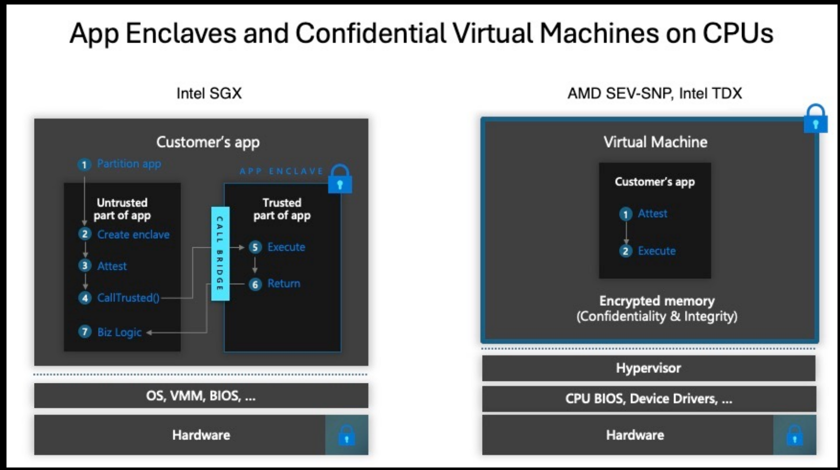
Motivation

- Evaluate feasibility, challenges, overhead of secure execution.

Objectives

- Port the selected benchmarks and applications to TEEs (e.g., Intel SGX, AMD SEV)
- Evaluate challenges of porting applications: code refactoring, dependency management, and enclave-specific constraints
- Measure and analyze performance of ported benchmarks and applications within TEEs against execution on native systems
- Provide insights into the trade-offs between performance, security, and ease of porting when using TEEs





https://learn.microsoft.com/en-us/azure/confidential-computing/trusted-execution-environment

2. Study of Benefits, Challenges, Security Properties of SELinux vs AppArmor on Ubuntu/Rocky Linux

Context

- Security-Enhanced Linux (SELinux) and AppArmor are Mandatory Access Control (MAC) systems that enhance Linux system security
- SELinux is popular in RHEL, Rocky Linux, AppArmor in Ubuntu

Motivation

- Uncover their security strengths, operational trade-offs, and deployment challenges
- Evaluate effectiveness, performance overhead, and ease of use •

Objectives

- Compare default configurations, usability, maintenance of SELinux and AppArmor on Ubuntu and Rocky Linux.
- Analyze the security properties of each system: access control granularity, policy enforcement, and attack surface reduction.
- Evaluate performance of SELinux and AppArmor using benchmarks (and real-world applications).

https://felixrante.com/advanced-security-selinux-and-apparmor-explained

Technology	Type Enforcement	MLS/MCS	Policy generator	Generator for containers
AppArmor	Yes	No	Yes	No
SELinux	Yes	Yes	No*	Yes

https://www.redhat.com/en/blog/apparmor-selinux-isolation

3. Single Board Computers with Neural Compute Sticks for Machine Learning Model Inference

Context

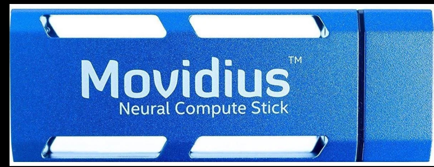
- Single Board Computers (SBC): Compact, low-cost computers with integrated essential components (ODROID, Raspberry Pi)
- Neural Compute Sticks (NCS): USB-powered hardware accelerators for ML inference - using a trained model to make predictions about new, unseen data.

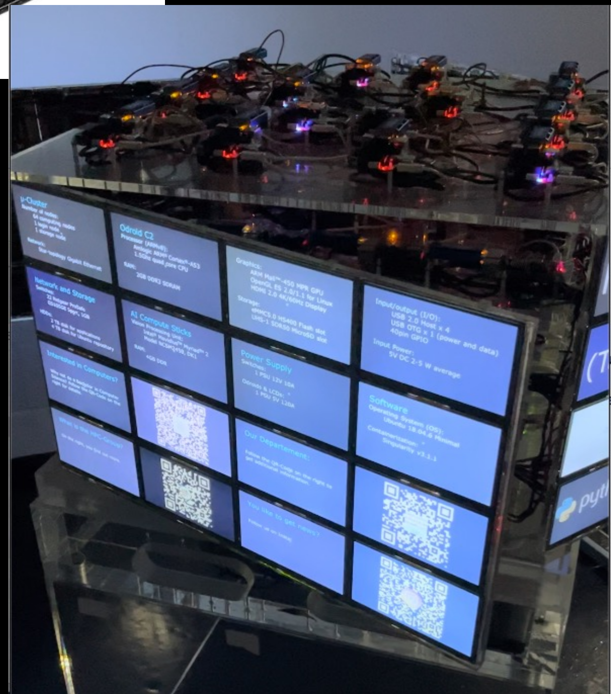
Motivation

- Enable SBCs NCS for efficient ML model inference
- Benchmark inference performance

Objectives

- Understand hardware acceleration for ML inference
- Connect and configure the compute stick on an SBC
- Run pretrained ML models on the compute stick
- Compare inference performance on SBC+NCS vs. general-purpose CPU (laptop, compute node)





4. Continuous Performance Evaluation of Astrophysical Applications Through CI/CD Pipelines

Context

- CI/CD pipelines are useful for faster delivery of high-quality software
- Such software powers scientific computing applications (e.g., SPH-EXA)
- Automation of building and testing software exists, but performance measurements do not

Motivation

- Software verification from the performance perspective
- Continuous measurement of performance improvements

Objectives

- Understand the use of CI/CD pipelines (e.g., R
- Learn about performance measurement of software
- Integrate performance monitoring to CI/CD (e.g., ReFrame)
- Visualize software performance for each build

5. CPU and GPU Frequency Sensitivity Analysis of HPC Workloads

Context

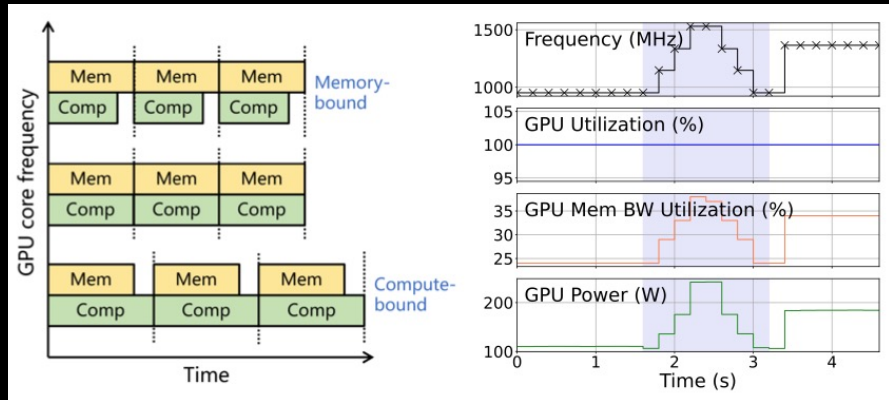
- Energy-efficiency is a major concern for HPC
- DVFS method can decrease consumption, but reduces performance

Motivation

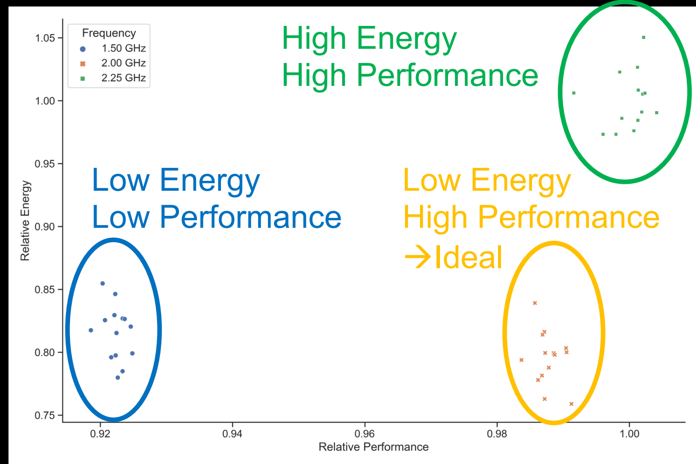
- Explore which workloads can benefit from DVFS more without sacrificing performance
- How do CPU and GPU workloads behave under DVFS

Objectives

- Perform frequency sensitivity analysis
- Identify frequency sensitive workloads
- Model the effect of frequency changes to voltage output
- Establish a frequency sensitivity threshold



Improving GPU Energy Efficiency through an Application-transparent Frequency Scaling Policy with Performance Assurance, Zhang, Wang, Lin, Xu, Wang, EuroSys 2024



https://www.archer2.ac.uk/news/2022/12/12/CPUFreq.html

6. Integrating Zoltan into SPH-EXA for Dynamic Load Balancing Studies

https://www.youtube.com/watch?v=elcMD_oFJt8

Context

- SPH-EXA is a high-performance, particle-based smoothed particle hydrodynamics (SPH) simulation framework used for modeling complex astrophysical phenomena (galaxy formation, star evolution, cosmological structure formation)
- Zoltan is a suite of advanced load balancing and data management tools designed for scalable parallel applications

Motivation

- Integrate Zoltan into SPH-EXA, to improve the dynamic load balancing of particles across processors, increase parallel efficiency, reduce communication overhead, and optimize resource utilization in large-scale simulations with SPH-EXA

Objectives

- Evaluate current load balancing
- Integrate Zoltan
- Evaluate different load balancing algorithms in Zoltan
- Analyze Performance of SPH-EXA + Zoltan

Parallel computing concepts / Example: Smoothed particle hydrodynamics
https://edukamu.fi/elements-of-supercomputing

7. Explore LLM-as-a-Judge for Verification and Validation of OpenACC & OpenMP Scheduling in Compilers

Context

- OpenACC and OpenMP, are widely used to parallelize HPC applications, and scheduling thereof is critical in achieving high efficiency and performance across various platforms.
- Verifying and validating compiler implementations to ensure correct and optimal scheduling is a challenging task.

Motivation

- Large Language Models (LLMs) offer new ways to generate targeted test cases and automate validation processes.
- Need for systematic verification and validation methods to address scheduling-related functionalities in compilers.

Objectives

- Design Effective Prompts to generate test cases for scheduling, that address critical scheduling-related functionalities.
- Evaluate the correctness of generated test cases using outputs from different LLMs and validate their compliance with specifications.
- Execute and analyze the verified and validated test cases on diverse platforms to assess their performance and efficiency.

Goal:

Test **OpenMP's** `schedule` **clause** to ensure it divides work properly and correctly computes the value of π in parallel.

Code Example:

c Copy

```c
#include <stdio.h>
#include <omp.h>

static long num_steps = 100000000; // Total number of steps
double step;

int main() {
    int i;
    double x, pi, sum = 0.0;
    double start_time, run_time;
    int num_threads = 4; // Number of threads to use

    step = 1.0 / (double)num_steps;

    // Measure time for parallel computation
    start_time = omp_get_wtime();

    #pragma omp parallel num_threads(num_threads)
    {
        double partial_sum = 0.0;

        #pragma omp for schedule(static) // OpenMP schedule clause (try 'dynamic' or 'guided'
        for (i = 0; i < num_steps; i++) {
            x = (i + 0.5) * step;
            partial_sum += 4.0 / (1.0 + x * x);
        }

        #pragma omp atomic
        sum += partial_sum;
    }

    pi = step * sum;

    run_time = omp_get_wtime() - start_time;

    printf("Computed PI = %.15f\n", pi);
    printf("Expected PI = 3.141592653589793\n");
    printf("Execution Time = %.5f seconds\n", run_time);

    return 0;
}
```

8. Automated Survey of Academic Papers through Local Large Language Models

Context

- Large collection of scientific papers about scheduling in HPC
- No clear overview of research gaps, trends, future needs

Motivation

- Automate the processing of large literature corpora
- Era of Natural Language Processing, LLM, and AI

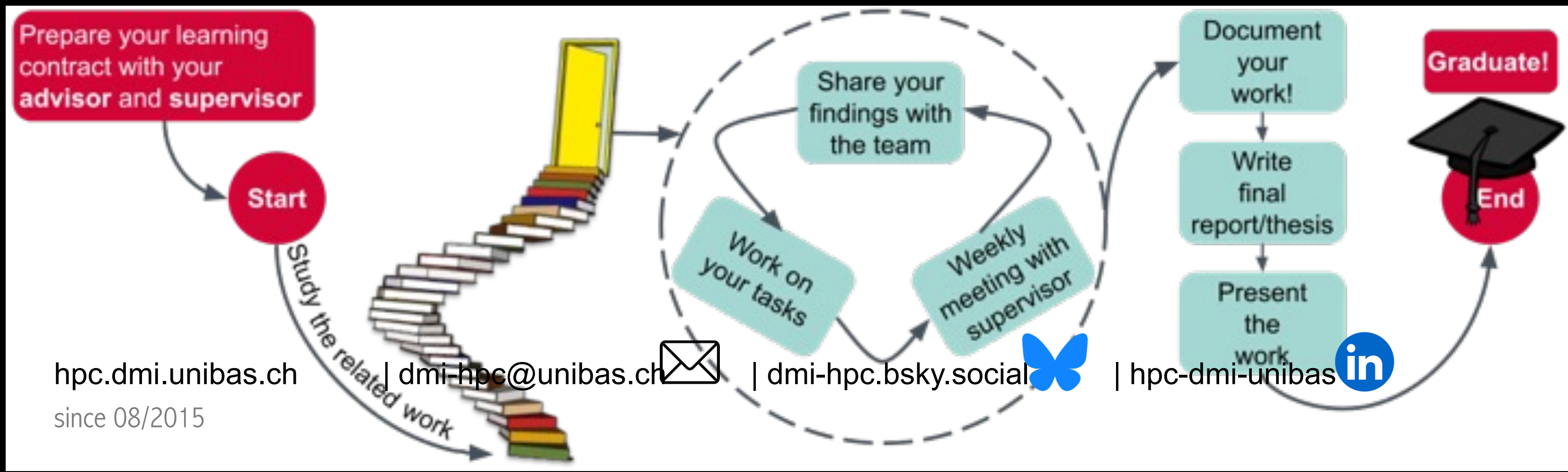
Objectives

- Survey the LLMs landscape to select a suitable open-source local LLM for this task
- Develop, test, and evaluate prompts to conduct a literature survey
- Establish an automate process for surveying papers
- Extract and analyze insights

You should choose this thesis if you are passionate about leveraging cutting-edge NLP techniques to automate literature surveys and uncover insights in the high-impact field of HPC scheduling.

High Performance Computing Group

Want to bring your own topic?



https://hpc.dmi.unibas.ch

dmi-hpc@unibas.ch

dmi-hpc.bsky.social

hpc-dmi-unibas

Selected topics online https://hpc.dmi.unibas.ch/en/theses/