University of Basel

DMI HPC

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

hpc.dmi.unibas.ch | dmi-hpc@unibas.ch | @HPC_DMI_UniBas | @hpc_dmi_unibas

since 08/2015

Updated: 16.12.2022

University of Basel > Faculty of Science > Department of Mathematics and Computer Science > **High Performance Computing Group**

In a Nutshell (December 2022)

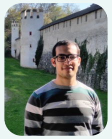
Team and Talents

Florina Ciorba
Lead

Robert Frank
IT administration
(shared)

Yvonne Walser
Secretary
(shared)

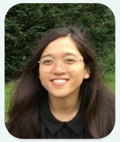
Ruben Cabezon
Senior scientist
(external, sciCORE)



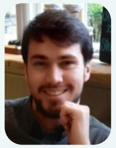
Ahmed Eleliemy
Postdoc

Jonas Müller Korndörfer
PhD student

Thomas Jakobsche
PhD student



Gabrielle Poerwawinata
(alumnus)



Osman Simsek
Postdoc

PhD dissertations (2)

BSc Theses (7+)

MSc Theses (5+)

MSc Projects (12+)

Student workers (3)

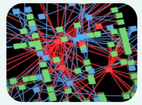
Postdoc
Open Position

PhD student
Open Position

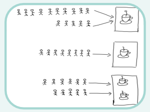
Lectures and Teaching

Operating Systems
BSc course, 8 CP
Spring 2022-present

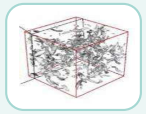
High Performance Computing
MSc course, 4 CP
Spring 2016-present



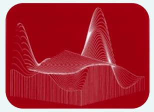
Foundations of Distributed Systems
MSc course, 8 CP
(with C. Tschudin, H. Schuldt)
Fall 2016-present



Algorithms, Concurrency, Parallelism
BSc seminar, 6 CP
Fall 2022-present



Applications of Computational Sciences
BSc course, 2 CP
(with others)
Fall 2019-present



Advanced Methods in Computational Sciences
PhD course, 3 CP
(with others)
Spring 2020-present

Research and Projects

DAPHNE

Integrating Data Analytics Pipelines for Large-Scale DM, HPC, and ML
01.12.2020-30.11.2024

3BEARS: Broad Bundle of BEnchmARks for Scheduling in HPC, Big Data, and ML
01.01.2021-31.12.2022

FNSNF

The DIALOGUE Study: Using digital health to improve care for families with predisposition to HBOC
01.11.2019-31.10.2023

pasc
Platform for Advanced Scientific Computing

SPH-EXA2: Smoothed Particle Hydrodynamics at Exascale
01.07.2021-30.06.2024

SKACH

SKACH: Sky to Observations
01.09.2021-31.12.2024

FNSNF

MLS2: Integrated Multilevel Scheduling for HPC
2023-2027
(in preparation)

FNSNF

MODA4HPC: Monitoring and Operational Data Analysis for Improving HPC Operations and Research
2023-2026
(in preparation)

Universität Basel

μ-Cluster: Visualizing Everything Parallel
01.03.2017-present

miniHPC: Research & teaching HPC cluster
01.01.2017-present

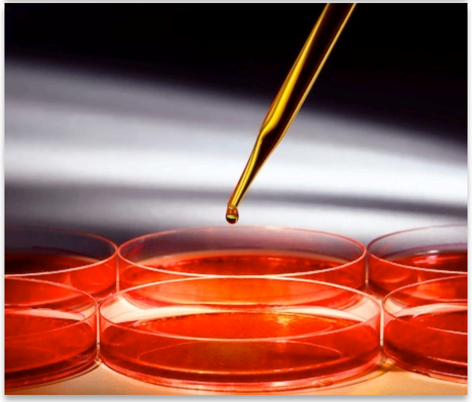
What is HPC?
#HPCMatters to our everyday lives

https://youtu.be/PuCx50FdSic (2’01”)

Experimental Research in * and in HPC

* can be Biology, Chemistry, Physics, Pharmaceutics, Environment

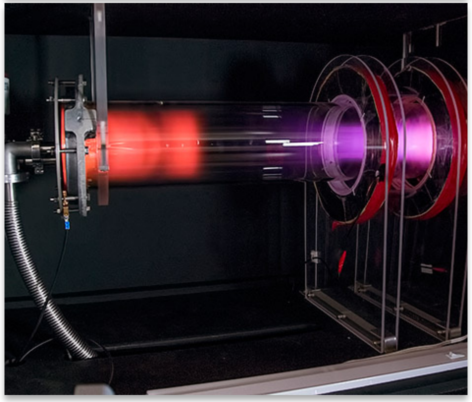
Biology



Chemistry

Pharmaceutics

Physics



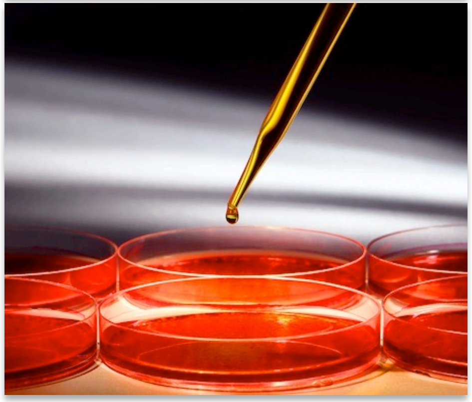
Environment

HPC

Experimental Research in * and in HPC involves Direct Experiments

* can be Biology, Chemistry, Physics, Pharmaceutics, Environment

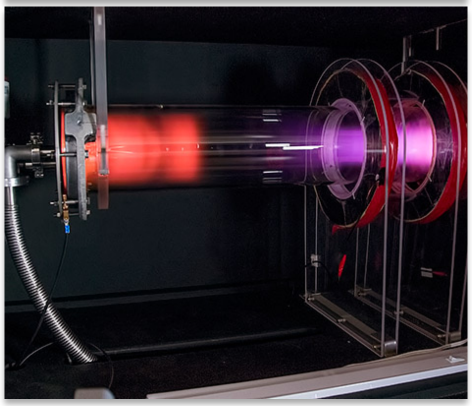
Biology



Chemistry

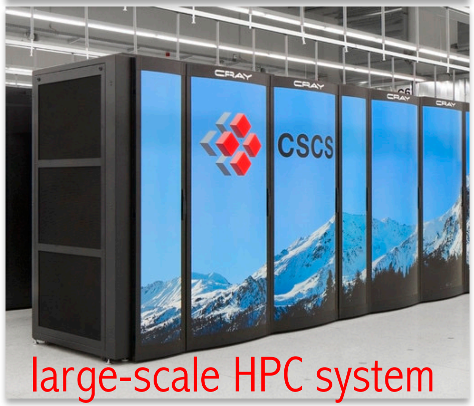
Pharmaceutics

Physics



Environment

HPC



Computational Research in * and in HPC involves Simulations

* can be Biology, Chemistry, Physics, Pharmaceutics, Environment

Biology

Chemistry

Pharmaceutics

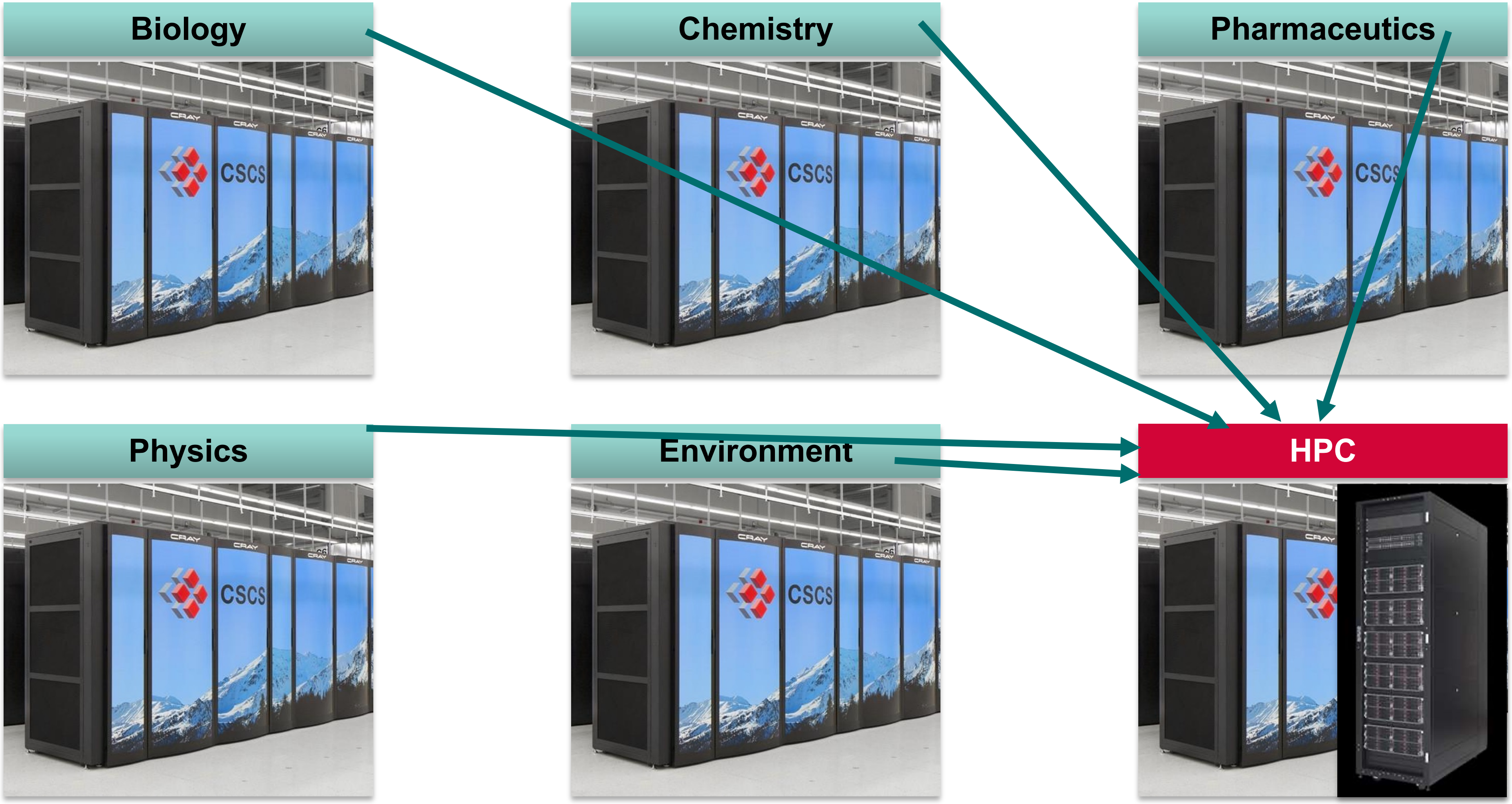
Physics

Environment

HPC

Each Simulation from * (incl. Computer Science) is an HPC Application

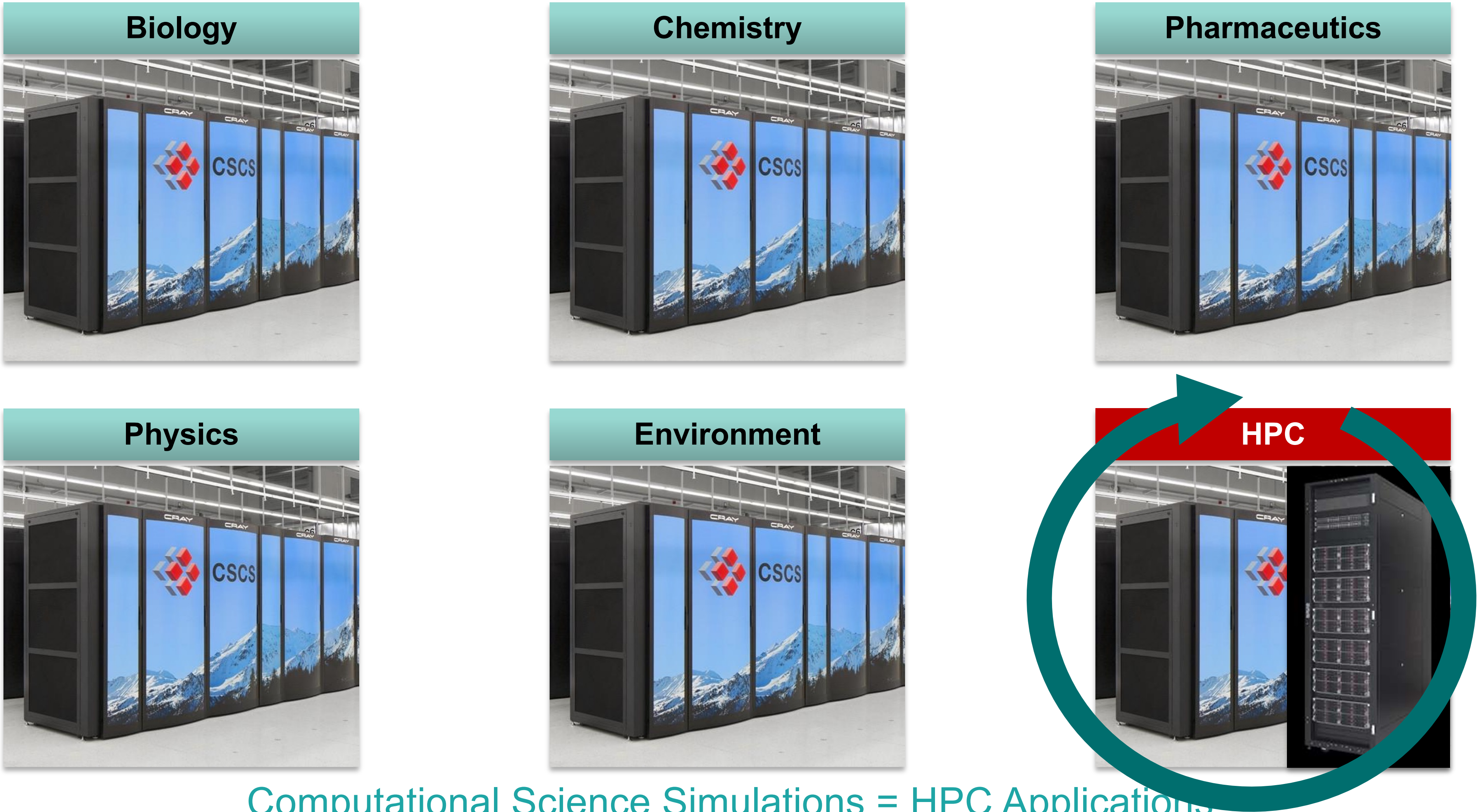
With Which We Perform Direct HPC Performance Experiments



Computational Science Simulations = HPC Applications

But We Also Abstract Computers Into Models which we Simulate

To Study, Improve, and Compare Our Simulations of Computers and Applications Against Real Code Executions (Direct Experiments) on the same Computers

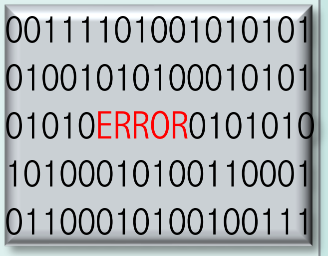


Computational Science Simulations = HPC Applications

High Performance Computing
What is important?

Goals

Performance

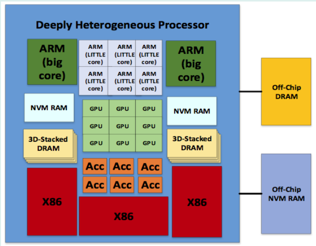


Resiliency

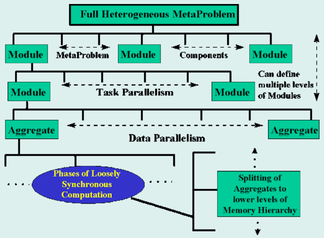
Energy

Energy

Challenges



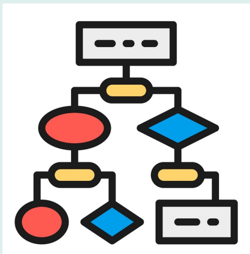
Heterogeneity



Multi-level parallelism

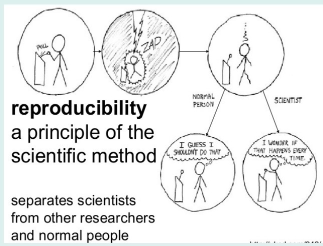
Load imbalance

Programming



Algorithms

Extreme scalability



Reproducibility

Convergence HPC+Data+AI

High Performance Computing Group
We Balance Theory and Practice with Interdisciplinary Applications

Vision & Mission
Improve **theory**, broaden **practice**, expand **application** of HPC & HPDA methods.
To accelerate scientific discovery.

Applications
- Astrophysics
- Cosmology
- Data analysis pipelines

Parallel Programming
- OpenMP — Loops & Tasks
- MPI — Placement & Decomposition
- DAPHNE — DSL & Runtime

Scheduling & Load Balancing
- Scheduling & Load Balancing Algorithms
- Load balancing libraries
- Runtime systems

Monitoring & Operational Data Analysis
- Monitoring
- System health
- Usage & Security Threats
- HPDA for operations

Benchmarking
- spec — SPEChpc 2021
- SPH-EXA — SPH-EXA2
- DAPHNE — IDA pipelines

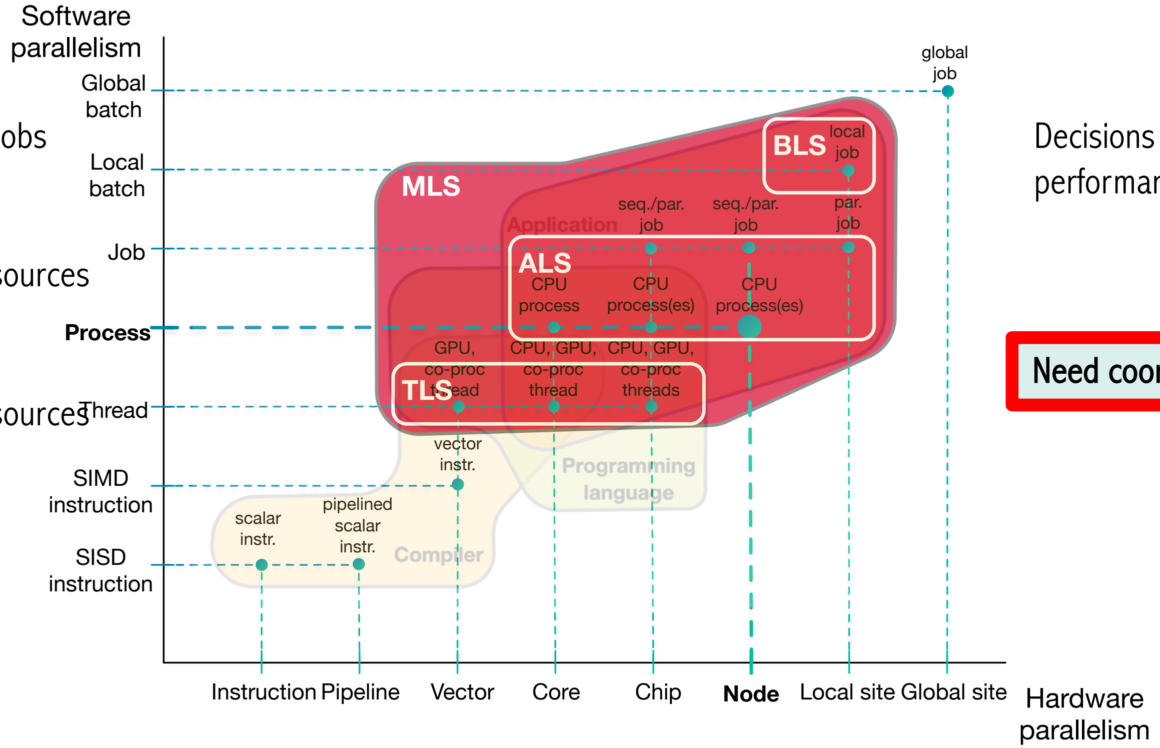
MLS: Multilevel Scheduling in Large Scale High Performance Computing
Current State: Disjoint Batch, Application, and Thread Level Scheduling

BLS: resources allocated to jobs

ALS: application processes scheduled onto allocated resources

TLS: application threads scheduled onto allocated resources

All scheduling decisions taken independently of each other



Decisions at higher levels impact & may limit performance envelope at lower levels

⇩

Need coordination & information exchange

BLS: Batch level scheduling
ALS: Application level scheduling
TLS: Thread level scheduling

MLS: Multilevel Scheduling for Large-Scale HPC 2017-2021

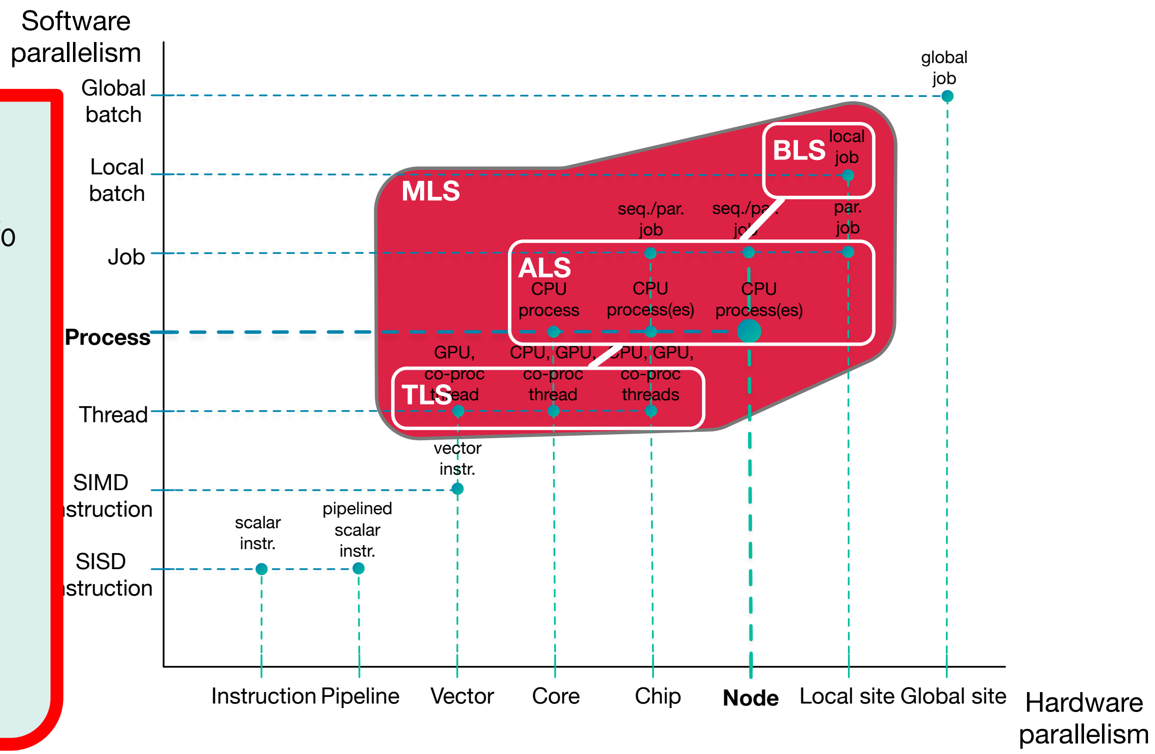
MLS: Multilevel Scheduling in Large Scale High Performance Computing
DMI-HPC Solution: Bridged Batch↔Application↔Thread Level Scheduling

Bridge BLS↔ALS↔TLS

to coordinate & exchange info
to better

expose
⇩
express
⇩
exploit

multilevel & heterog.
parallelism



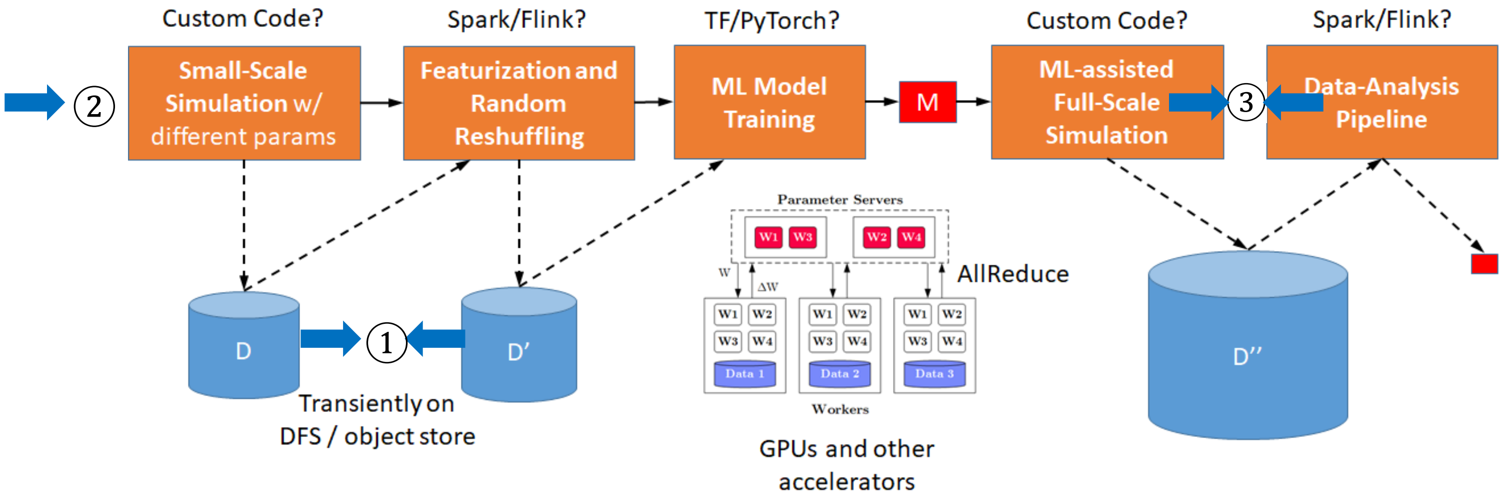
DMI HPC
hpc.dmi.unibas.ch

FNSNF

MLS: Multilevel Scheduling for Large-Scale HPC 2017-2021

DAPHNE: Integrated Data Analysis Pipelines for Large-Scale DM, HPC, ML

Example of an Integrated Data Analysis Pipeline: From Classical Simulation to ML-Assisted Simulation



Opportunity ①: Why not fuse the data generation (D) into the subsequent ML training (D') to avoid unnecessary data transfer?

Opportunity ②: Why not change the simulation parameters to yield better convergence or generalization of the ML model?

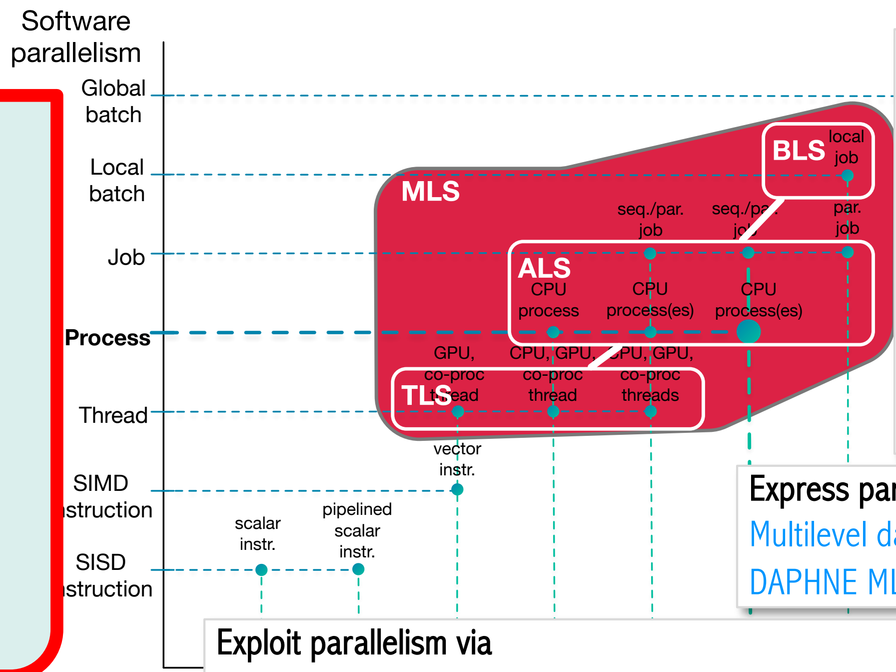
Opportunity ③: Why not fuse the final full-scale simulation with the data analysis pipeline to avoid unnecessary materialization?

DAPHNE
Integrating Data Analytics Pipelines for Large-Scale DM, HPC, and ML
01.12.2020-30.11.2024

MLS and DAPHNE
Multilevel Scheduling of Integrated Data Analysis Pipelines (IDAs)

Opportunity to
expose
⇩
express
⇩
exploit

multilevel & heterog. parallelism
for integrated data analysis pipelines



Expose parallelism via
Multilevel parallelism abstractions
- job (batch) parallelism (HPC, ML)
- application parallelism
  - model parallelism (ML)
  - data parallelism (DM, HPC, ML)
  - task parallelism (asynchronicity: HPC, ML)
- operator parallelism (DM, HPC, ML)

DAPHNE Use Cases and Benchmarks

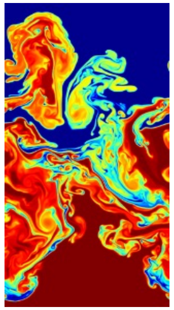
Express parallelism via
Multilevel data & program representation
DAPHNE MLIR dialect

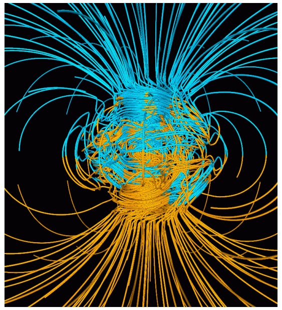
Exploit parallelism via
Multilevel compilation and execution: joined, fine-grained pipelines
Runtime system: workflows
Hierarchical scheduling and multi-tenant resource sharing
Efficient storage and HW acceleration

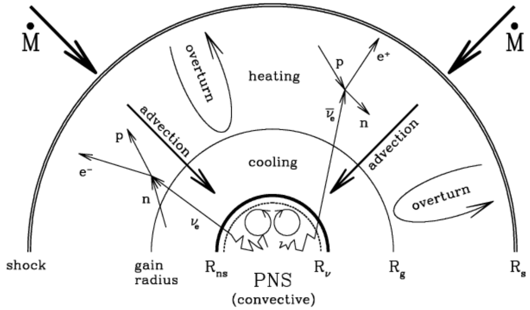
DM: Data Management
HPC: High Performance Computing
ML: Machine Learning
MLIR: Multilevel Intermediate Representation

DAPHNE
Integrating Data Analytics Pipelines
for Large-Scale
DM, HPC, and ML
01.12.2020-30.11.2024

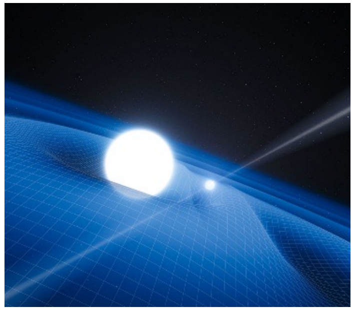
SPH-EXA: Smoothed Particle Hydrodynamics at Exascale


Hydrodynamics


Magnetic fields



Neutrino transport


General Relativity


Nuclear Physics

Vision

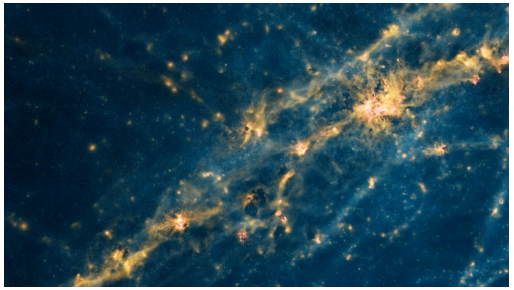
First **trillion particle galaxy formation** with SPH, gravity, and radiation using the scalable and fault tolerant SPH framework, SPH-EXA on Exascale computers.

Philosophy

Interdisciplinary **codesign** between computer scientists, astrophysicists, cosmologists, and visualization specialists to implement the SPH method (and additional physics) **for Exascale**, instead of optimizing already existing codes.

SPH-EXA Strategy

- State-of-the-art SPH method, leveraging implementations from existing codes
- Composable framework written in C++20
- MPI-based communication, OpenMP | OpenACC | CUDA | HIP parallelization
- Supports various types of hardware architectures, in-situ visualization
- No external dependencies beyond core compiler/language components & MPI
- Dynamic scheduling, adaptive load balancing, and fault-tolerance
- Designed for easy extensibility with additional physical effects and observables
- Extensive test coverage with unit and integration tests


Formation of a galaxy in the GigaERIS simulation (with ChaNGa, Mayer et al.)

PASC Platform for Advanced Scientific Computing
SKACH

SPH EXA project

(1) PASC SPH-EXA1 (2017-2021)
(2) PASC SPH-EXA2 (2021-2024)
(3) SKACH (2021-2024)

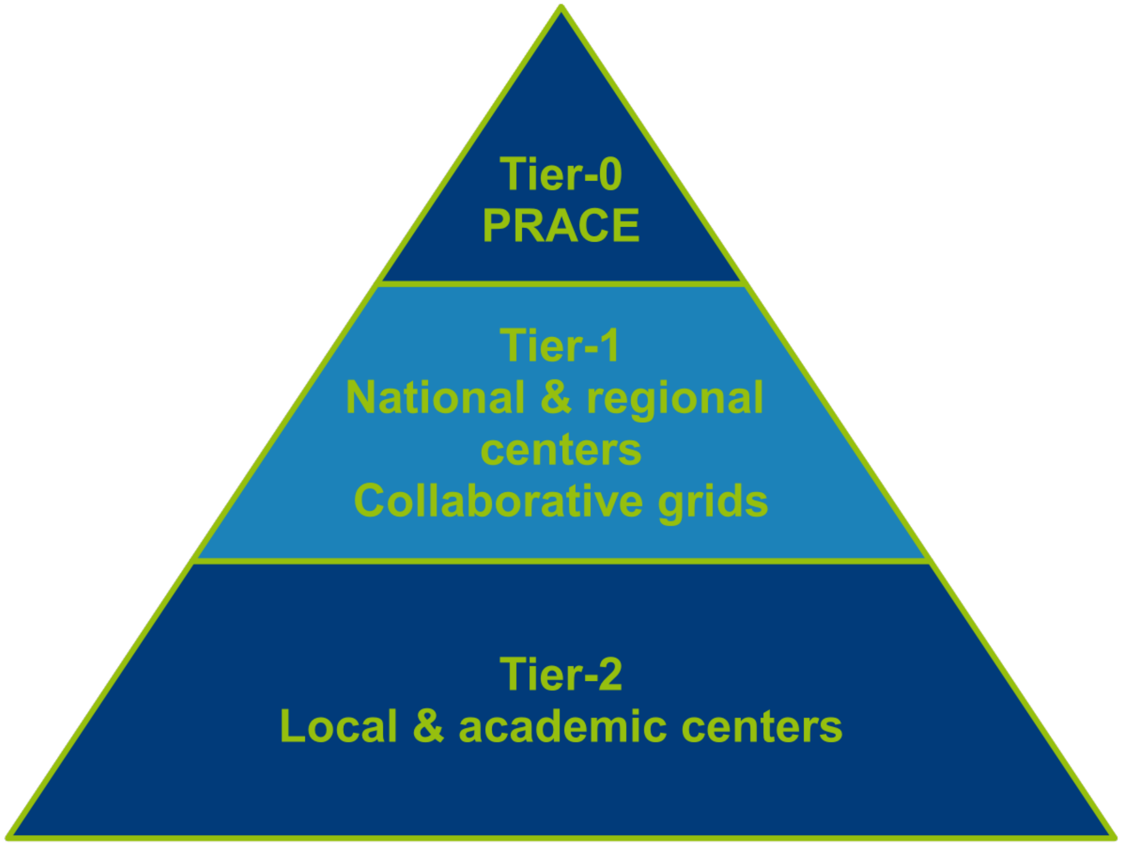
SPH-EXA1: https://hpc.dmi.unibas.ch/en/research/sph-exa/
SPH-EXA2: https://hpc.dmi.unibas.ch/en/research/pasc-sph-exa2/
SPH-EXA SKACH: https://hpc.dmi.unibas.ch/en/research/skach/
SPH-EXA download: https://github.com/unibas-dmi-hpc/SPH-EXA

SPH-EXA 2

SPH-EXA2: Smoothed Particle Hydrodynamics at Exascale
01.07.2021-30.06.2024

SKAO-C: Sky to Observations
01.09.2021-31.12.2024

- **Main Goal:** PRACE Tier-0 allocation
  (>1 million Nh + GPUs)



SPH-EXA 2

SPH-EXA2: Smoothed Particle Hydrodynamics at Exascale
01.07.2021-30.06.2024

SKAO-C: Sky to Observations
01.09.2021-31.12.2024

- **Main Goal:** PRACE Tier-0 allocation
  (>1 million Nh + GPUs)

ChaNGa

PHOEBOS-MR + HR simulation

30 billion particles + 240 billion particles
Precursor simulations for SKAO
No scaling further than 3,000 nodes (no GPUs)
But HR will need to run on 4’898 nodes

SPH EXA

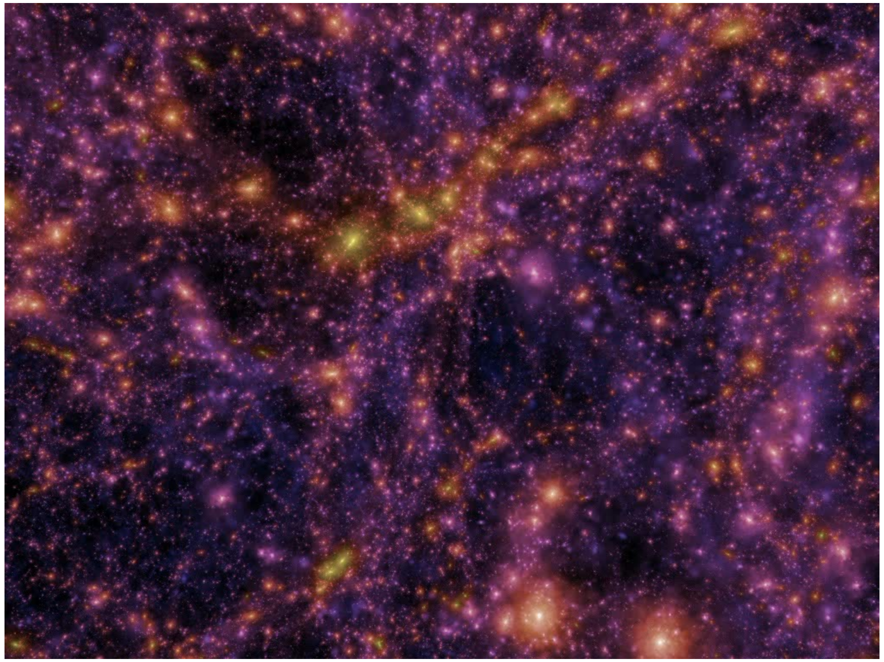
PHOEBOS-MR + HR simulation

30 billion particles + 240 billion particles
Validation & Verification against ChaNGa
Scaling further than 4,000 nodes (with GPUs)

SPH EXA

EXA-PHOEBOS simulation

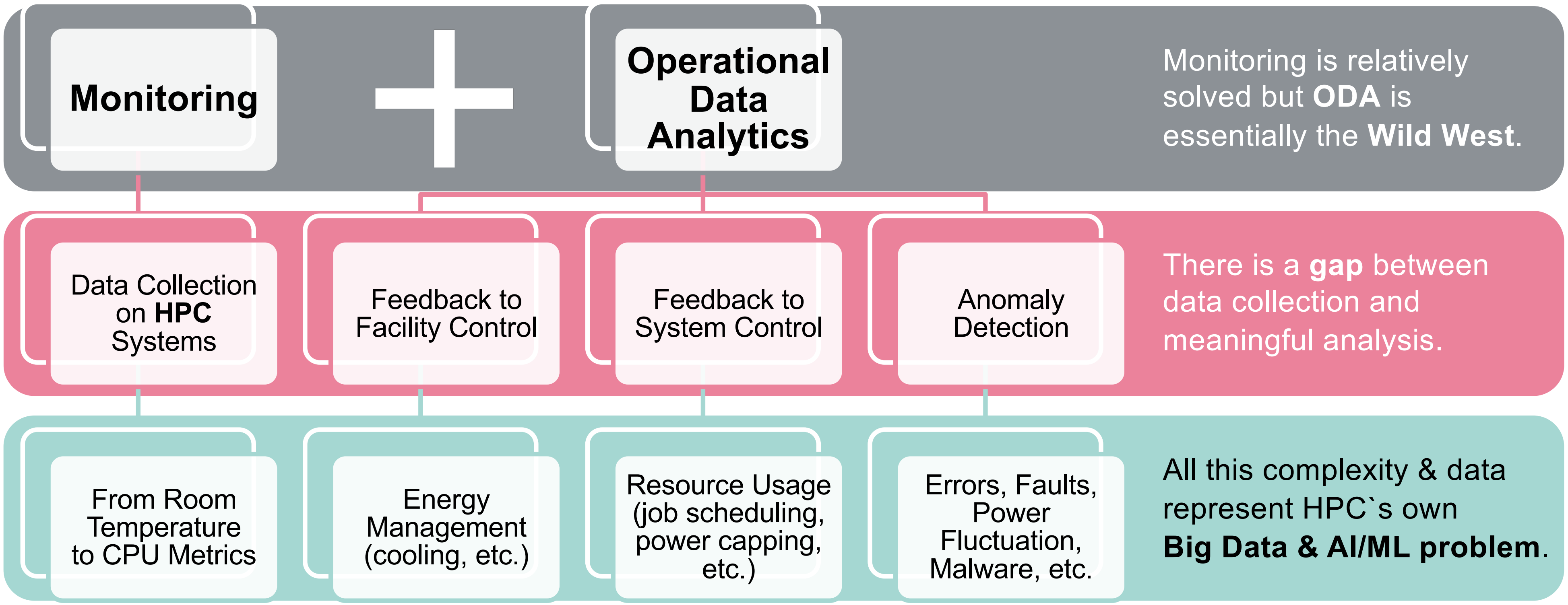
1 trillion particles
Flagship simulation for SKAO
Scaling at full machine and beyond (with GPUs)



Source: Millenium simulation (ESO) (now Illustris)

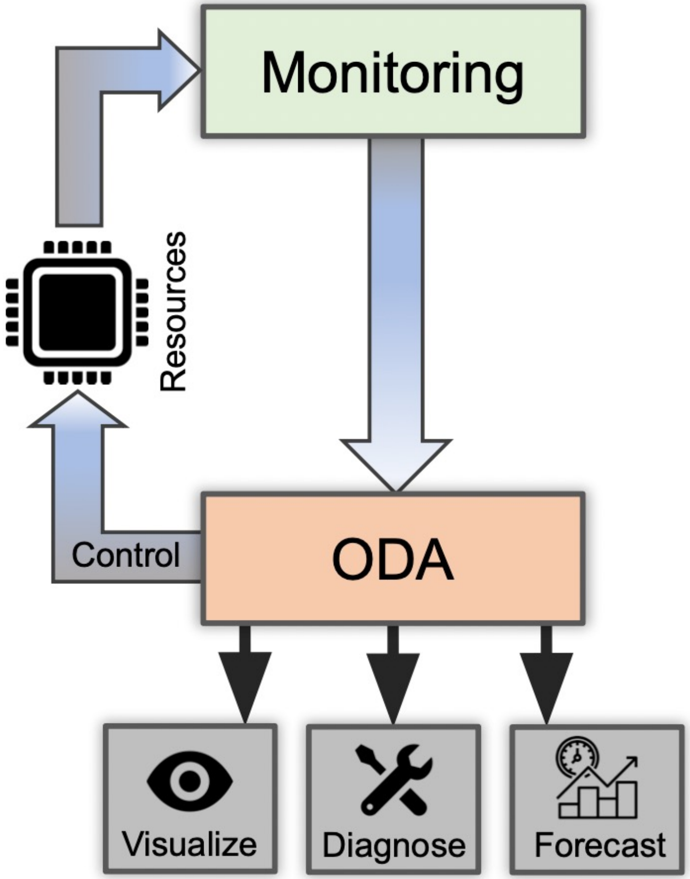
MODA4HPC: Monitoring and Operational Data Analytics
Insight through Eyes on the Inside

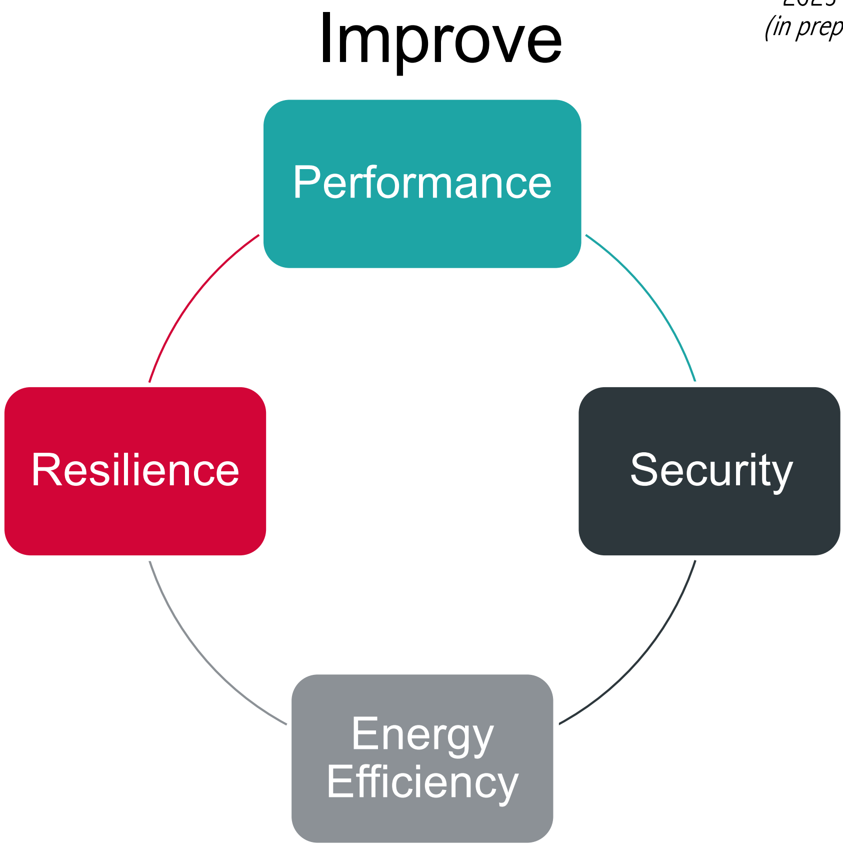
MODA4HPC: Monitoring and Operational Data Analysis for Improving HPC Operations and Research 2023-2026 *(in preparation)*



MODA4HPC: Monitoring and Operational Data Analytics
Goals

MODA4HPC: Monitoring and Operational Data Analysis for Improving HPC Operations and Research 2023-2026 *(in preparation)*





Recently: security in HPC

- ➢ **Avoiding misuse by e.g. cryptocurrency miners!**

Adapted from: A. Netti, W. Shin, M. Ott, R. Palumbo, S. Dolas, T. Wilde, K. Yamamoto. "A Conceptual Framework for HPC Operational Data Analytics". SC21 Operational Data Analytics BoF. 2021.

3BEARS: Broad Bundle of BEnchmARks for
Scheduling in HPC, Big Data, and ML
01.01.2021-31.12.2022

3BEARS: Broad Bundle of BEnchmARks for Allocation of Resources & Scheduling in Parallel & Distributed Computing

Motivation

- The scheduling context keeps changing with every progress in technology. New scheduling algorithms may be expected to emerge and other may become obsolete.
- No scheduling benchmark exist for the fair and reproducible evaluation of scheduling algorithms.

Objectives

Create the 3BEARS Benchmark Suite: A set of scheduling and resource allocation test applications, freely available to the community to enable the fair and reproducible evaluation of the state-of-the-art & to design novel strategies for scheduling and resource allocation techniques for emerging HPC architectures.

Approach

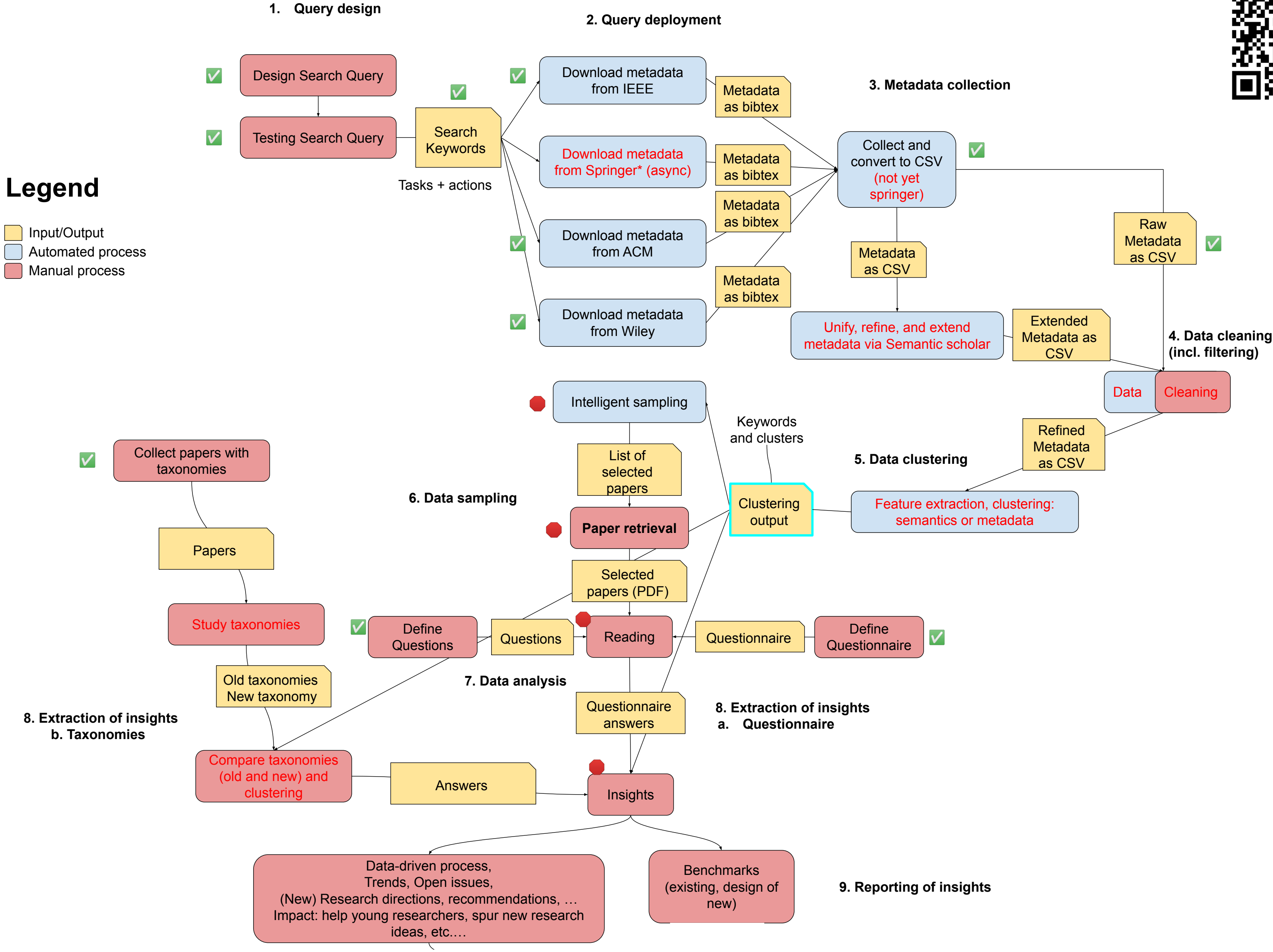
- Survey the state of the art in resource allocation and scheduling in order to identify the individual benchmarks and mini-apps used by the researchers of this community
- Characterize the existing benchmarks and mini-apps (often not scheduling-oriented) regarding their features and limitations for resource allocation and scheduling experiments
- Design and develop the 3BEARS Benchmark Suite
- Test, reproduce, and verify the developed 3BEARS Suite using state of the art scheduling and resource allocation algorithms

Outcomes

A set of application benchmarks' code, parameters, analysis, and guidelines that can be used as a starting point for the design and development of novel scheduling and resource allocation techniques.

3BEARS: Broad Bundle of BEnchmARks for Scheduling in HPC, Big Data, and ML
01.01.2021-31.12.2022

3BEARS: Broad Bundle of BEnchmARks for Allocation of Resources & Scheduling in Parallel & Distributed Computing



Topics for Bachelor Theses (15 CP)

High Performance Computing Group | December 2022

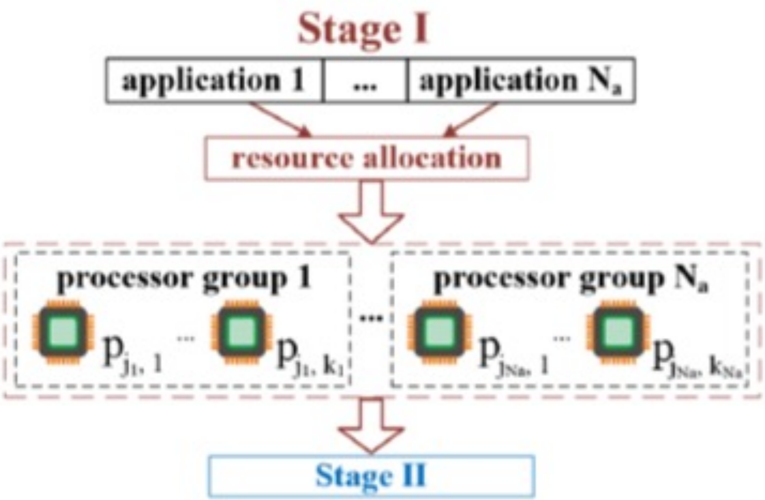
Topics for Bachelor Theses
Selection (December 2022)

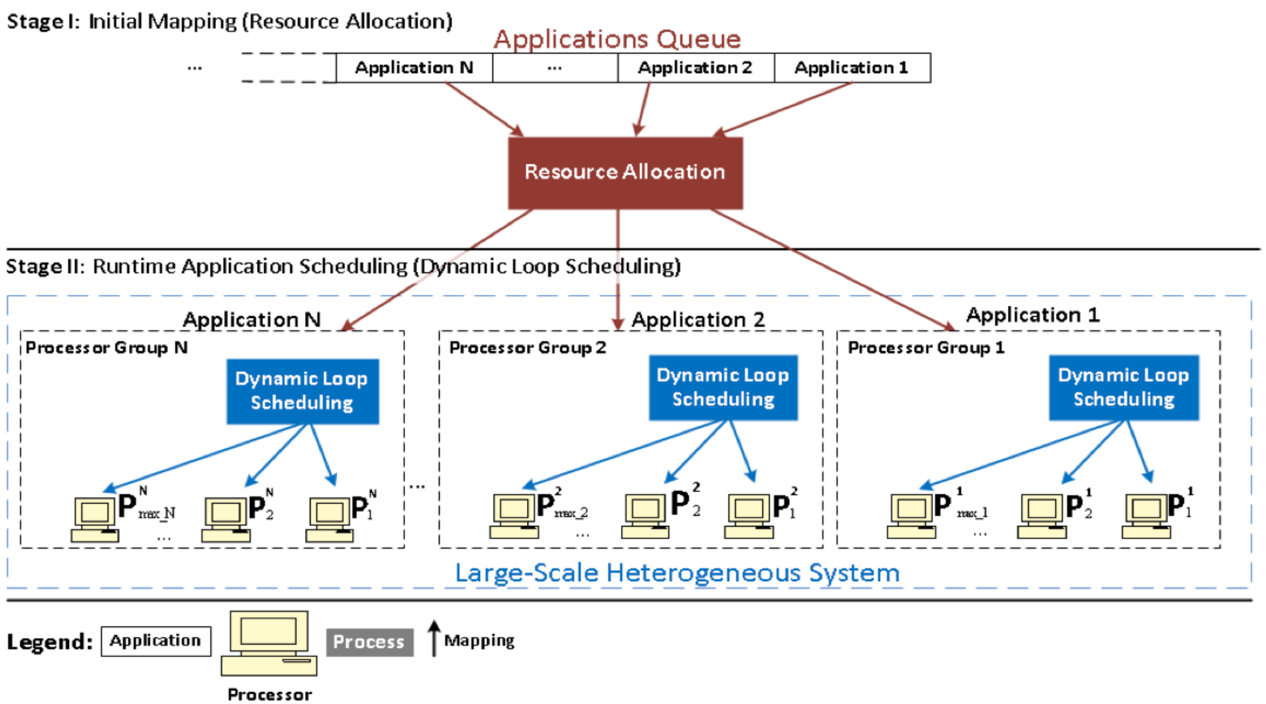
1. Explore the Robustness of Resource Allocation Heuristics in HPC
2. Automated Generation of Job Submission Scripts for Scientific Applications
3. Individual ML-based Assistant for Collocation & Backfilling of User's HPC Jobs
4. Custom Visualization of Generic Performance Data with the Performance Analysis Portal
5. Analysis of Energy Consumption of Busy vs. Idle Computing Nodes
6. Performance and Energy Measurements of HPC Applications
7. In-situ Visualization of Cosmological Simulations
8. Exploration of Mixed Precision Computations in Cosmological Simulations
9. Injection and Detection of Anomalies in HPC systems
10. Cyclomatic Complexity Analysis of HPC Application Codes

1. Explore the Robustness of Resource Allocation Heuristics in HPC

Goal: Allocate heterogeneous computing resources to a *batch of moldable parallel jobs* in such a way that all applications within a batch *finish before a certain deadline* in the presence of *unpredictable perturbations* that may arise during execution

- Implement a library of resource allocation heuristics (from literature)
- Assess their performance in various simulation scenarios with multiple applications (multi-tenant), resource types, and perturbations

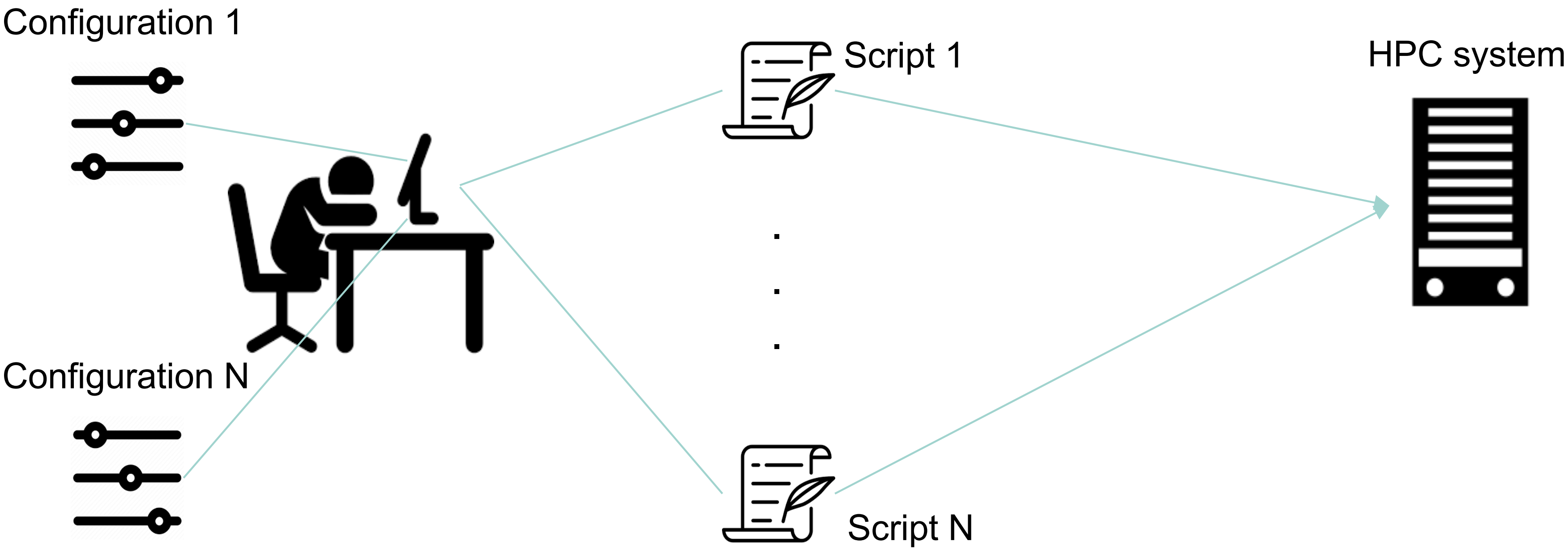




2. Automated Generation of Job Submission Scripts for Scientific Applications

Goal: Facilitate the use of HPC systems for computational scientists for factorial experimental campaigns

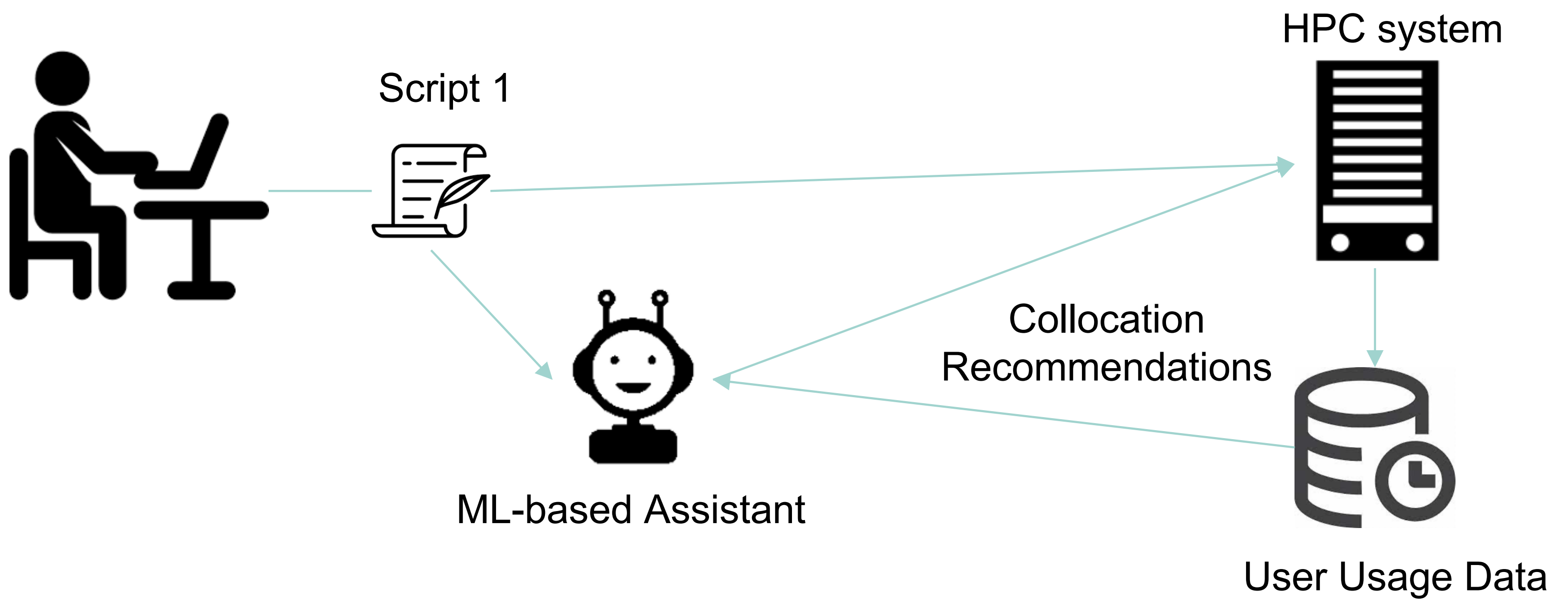
- Current challenge: preparing experiments requires writing, handling, and maintaining a very large number of job submission scripts
- Objective: Design and implement a web-based tool (Python-Flask or JavaScript-NodeJS) that automatically generates and submits customized job scripts of scientific applications.



3. Individual ML-based Assistant for Collocation & Backfilling of a User's HPC Jobs

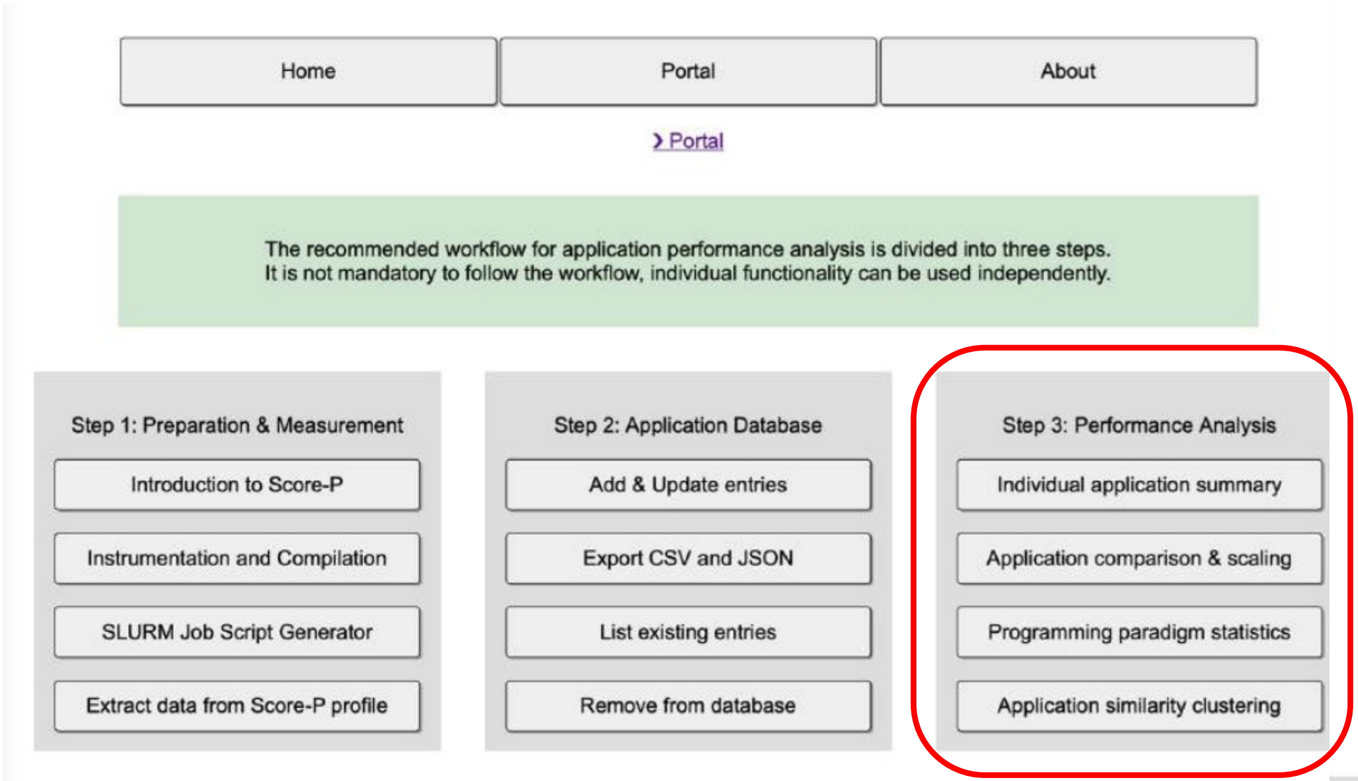
Goal: Reduce waiting time in the queue for jobs of an HPC user via collocation

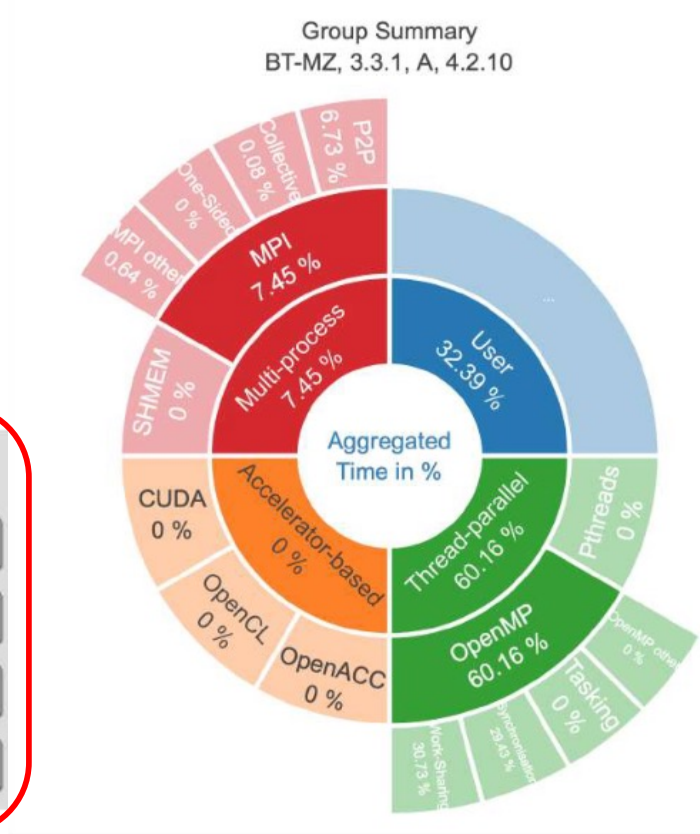
- Record and store information about applications execution (execution time, CPU, and memory usages) and learn a model on these data
- Design a command-line tool to give ML-based recommendations to a user for collocating and/or backfilling their jobs on the HPC system to improve system utilization
- Testing the tool by running benchmarks, measure performance and system utilization with/without collocation and backfilling
- Technology used will include python and clustering techniques



4. Custom Visualization of Generic Performance Data with the Performance Analysis Portal (PAP)

Goal: Enable visualization and analysis of generic performance data in PAP
- Create an interface to allow *data characterization*
- Create an interface for insertion of custom metrics
- Update, modify, and create new visualizations for processing generic performance data



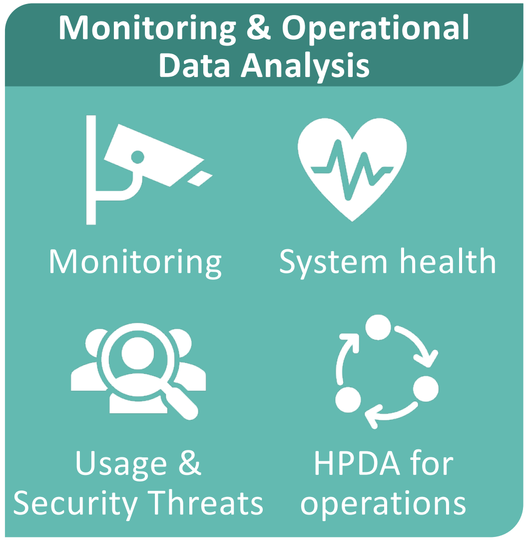


5. Analysis of Energy Consumption of Busy vs. Idle Computing Nodes

Goal: Investigate the energy consumption for busy and idle nodes of the HPC cluster "miniHPC" operated by the HPC research group to recommend power saving strategies.

- How much energy is the cluster using in total?
- How much energy is used by busy nodes that execute workloads?
- How much energy is used by idle nodes?
- Can we power down/shut off idle nodes and when should we do that, e.g. at night?

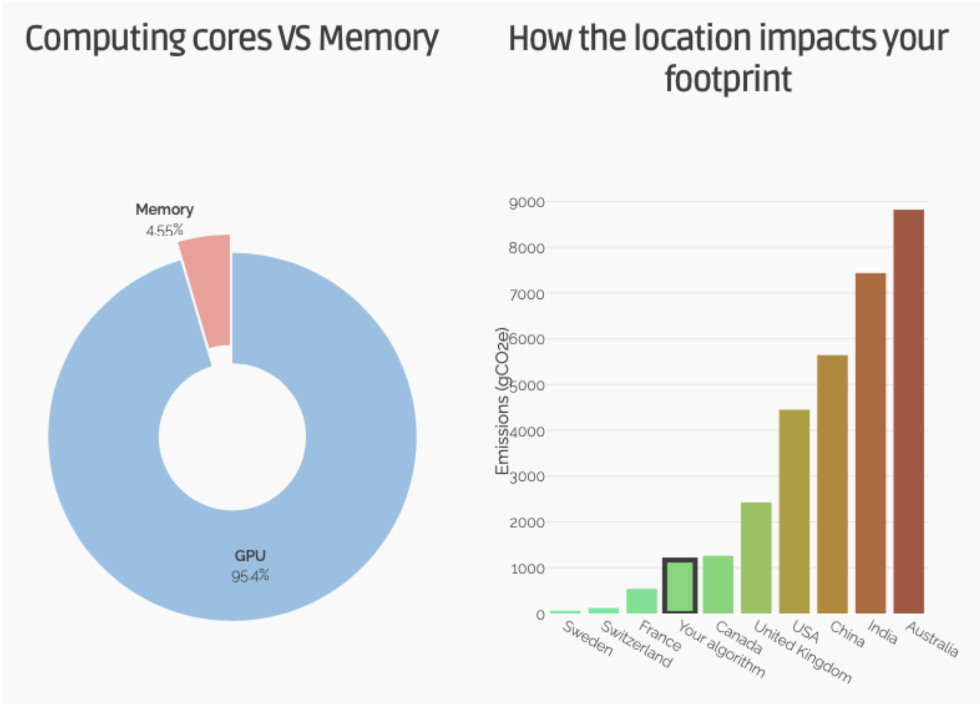
miniHPC system: left-switch and nodes, right-connections



6. Performance and Energy Measurements of HPC Applications

Goal: Explore performance vs. energy (and carbon footprint) of HPC applications using accurate energy measurements.

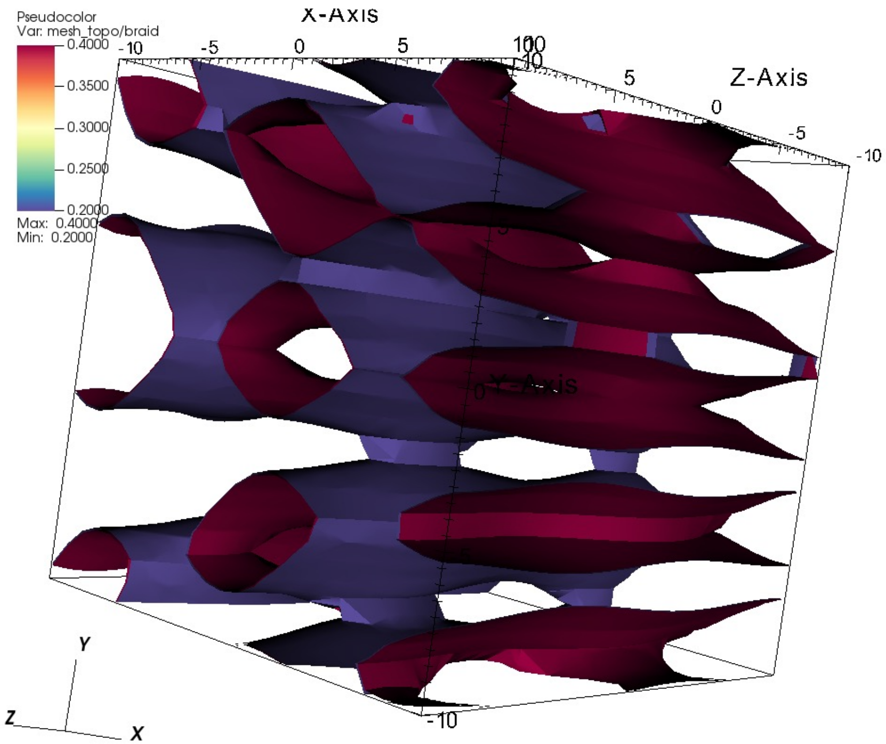
- Implementation of more accurate models for GA4HPC tool (www.green-algorithms.org)
- Analyze performance and energy characteristics of various HPC applications
- Identify configurations with reduced energy and highest performance

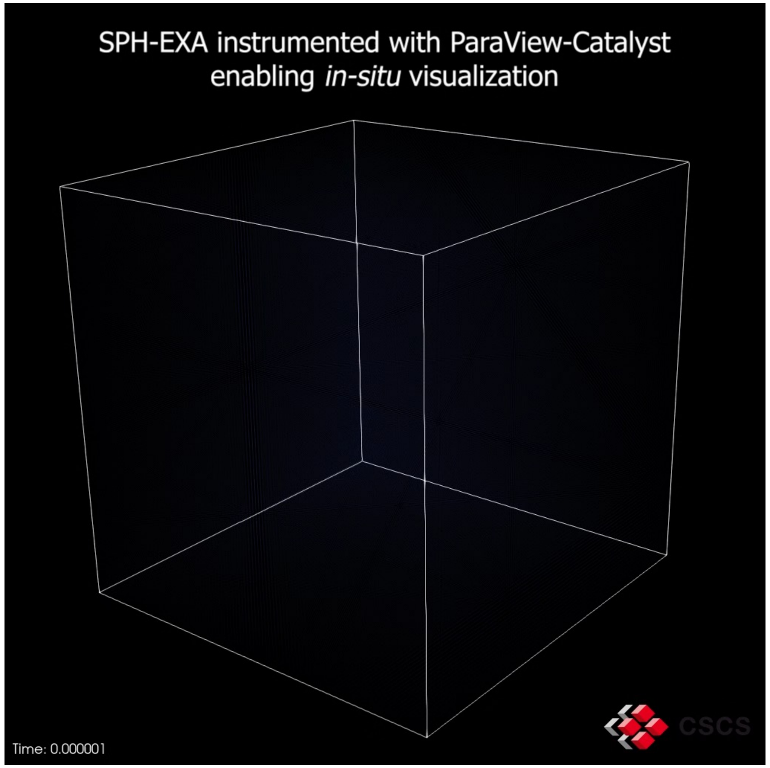


7. In-situ Visualization of Cosmological Simulations

Goal: Efficient visualization of computational characteristics of cosmological simulations.

- Develop support for visualizing *particle-to-computing node* correspondence
- Implement filters for efficient information visualization in 3-D space
- Create *videos* that show the evolution of particle distribution across computing nodes during execution

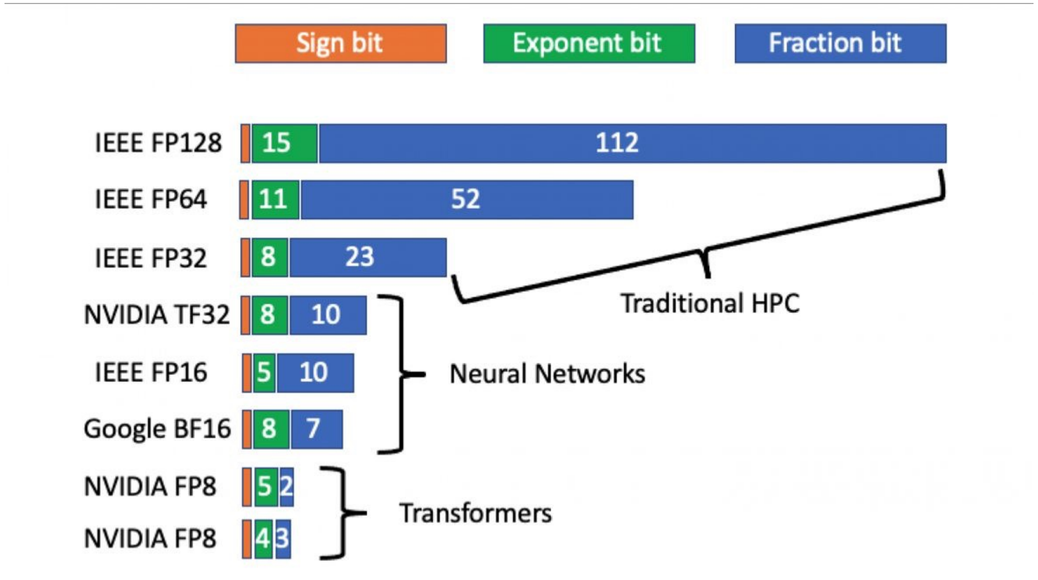


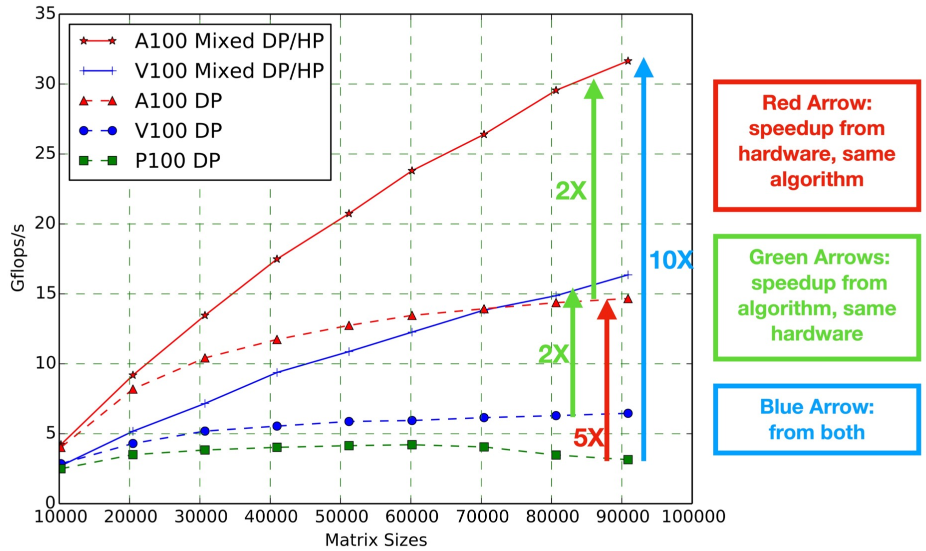


8. Exploration of Mixed Precision Computations in Cosmological Simulations

Goal: Evaluate *performance* vs. *accuracy* of mixed precision computations in cosmological simulations (very large, extremely long running, typically use FP64 precision)

- Create configurations for changing the precision of different fields in the simulation code
- Prepare and run experiments with multiple mixed-precision configurations
- Compare against the performance and accuracy of original FP64 runs and find the best performing mixed-precision configurations within acceptable errors

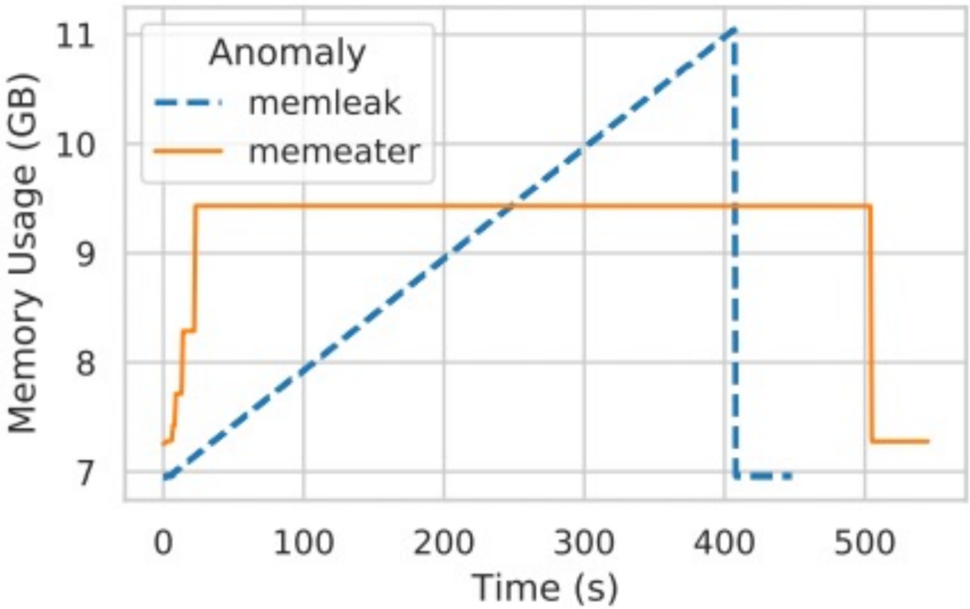


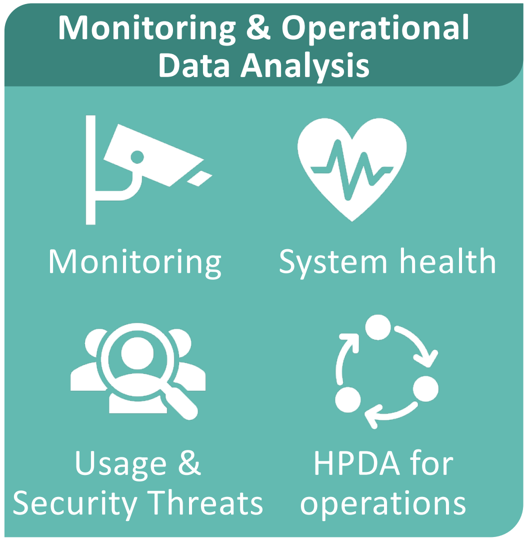


9. Injection and Detection of Anomalies in HPC Systems

Goal: Assess performance anomaly injection methods for HPC applications and investigate how to detect them automatically.

- Examine performance anomalies supported by the **H**PC **P**erformance **A**nomaly **S**uite
- *Inject* various performance anomalies using **HPAS**
- *Detect* the injected anomalies using Linux's **perf** (performance analysis tool)
- Explore the **automation** anomaly detection, e.g. based on rules





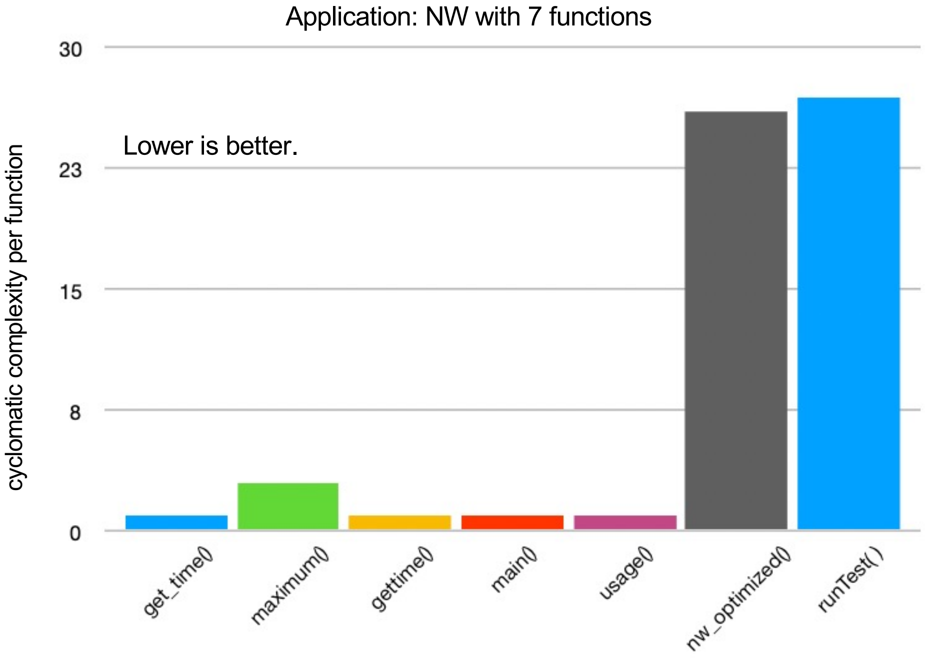
Emre Ates, Yijia Zhang, Burak Aksar, Jim Brandt, Vitus J. Leung, Manuel Egele, and Ayse K. Coskun. HPAS: An HPC Performance Anomaly Suite for Reproducing Performance Variations. In International Conference on Parallel Processing (ICPP), Aug. 2019

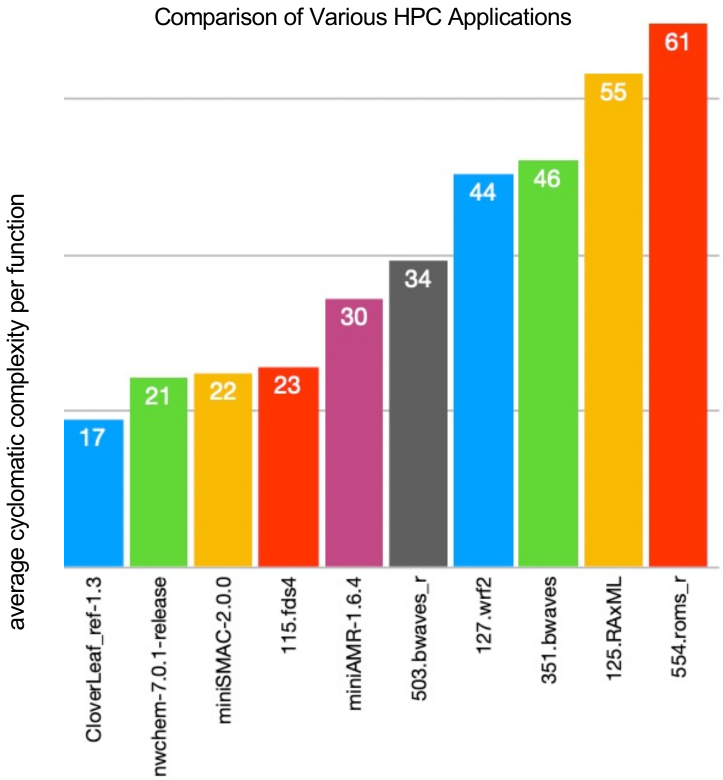
10. Cyclomatic Complexity Analysis of HPC Application Codes

Goal: Analyze the *cyclomatic complexity* (# of paths through a code) of HPC application codes. The objectives are to:

- *Compute cyclomatic complexity** for a large number of applications
- Investigate how *average cyclomatic complexity* compares for various applications
- Compare cyclomatic complexity for applications across programming languages & paradigms
- Quantify analysis insights through statistics and appropriate visualization

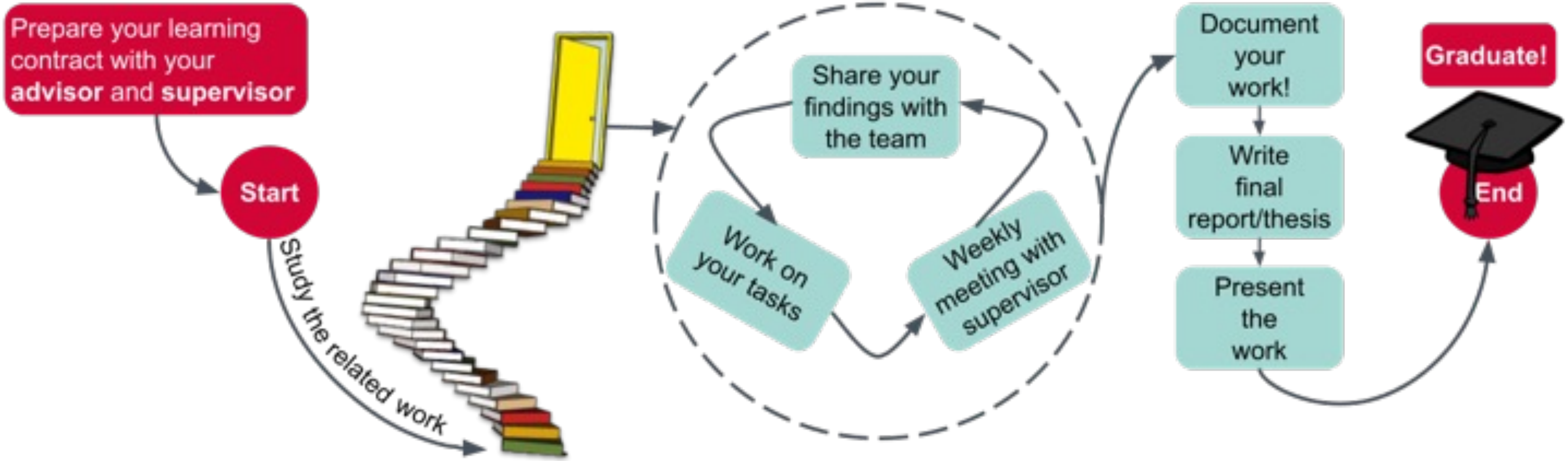
*with the Lizard[1] cyclomatic complexity analyzer.





[1]https://github.com/terryyin/lizard

High Performance Computing Group
Your own topic?



https://hpc.dmi.unibas.ch

dmi-hpc@unibas.ch

@HPC_DMI_UniBas

@hpc_dmi_unibas

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