Parallel Programming for *More than HPC*

Christian Terboven <terboven@itc.rwth-aachen.de>

September 15th, 2023

But first…

- … home come the title?

SC20, fully affected by the pandemic, sailed with two titles to illustrate how advanced computing plays a central role for not just research and development, but for everyone's life.

In my opinion, the range of programming models is a foundation of advanced computing. Given the increasing diversity of systems and applications, the set of topics that we are working on is broadening.

Reseach Directions + Agenda of this Talk

- Research Directions of RWTH's HPC Team:
  - Parallel Programming Models and Systems: OpenMP + MPI
  - Correctness Checking of Parallel Programs: MUST, Archer, OTF-CPT
  - Total Cost of Ownership in HPC
  - Analysis of Parallel Computer Architectures
  - High-level methodological HPC support

- First: three slides about RWTH and the IT Center / i12

- Agenda of this talk:

1. Selected contributions to Parallel Programming
2. Parallel Performance Engineering
3. Coupling HPC+AI Applications

RWTH Aachen University

A leading university with strong research

- One of the leading Technical Universities in Germany (TU9)
- One of eleven Germen Universities of Excellence
- Ranked among top 10 German universities in THE 2023
- One of the central nodes in the German Initiative for Research Data Management (NFDI)
- Host of many recognized centers: National High Performance Computing Center for Engineering Sciences (NHR4CES), ...

Studies and Teaching

Excellent Teaching, Learning and Assessment

- 47.269 Students
- 13.354 International Students
- 170 courses of study

Employees and Finances

- 10.249 Employees
- 1.108 Mio. Euro annual budget

IT Center @ RWTH Aachen University

Mission

IT-Service Provider for RWTH Aachen University

- From network infrastructure to HPC systems
- E-Learning and SLCM
- Responsible to support Research Data Management at RWTH

National Mission

- **HPC for Computational Engineering Sciences (NHR4CES)**
- **Important node of the NFDI network**

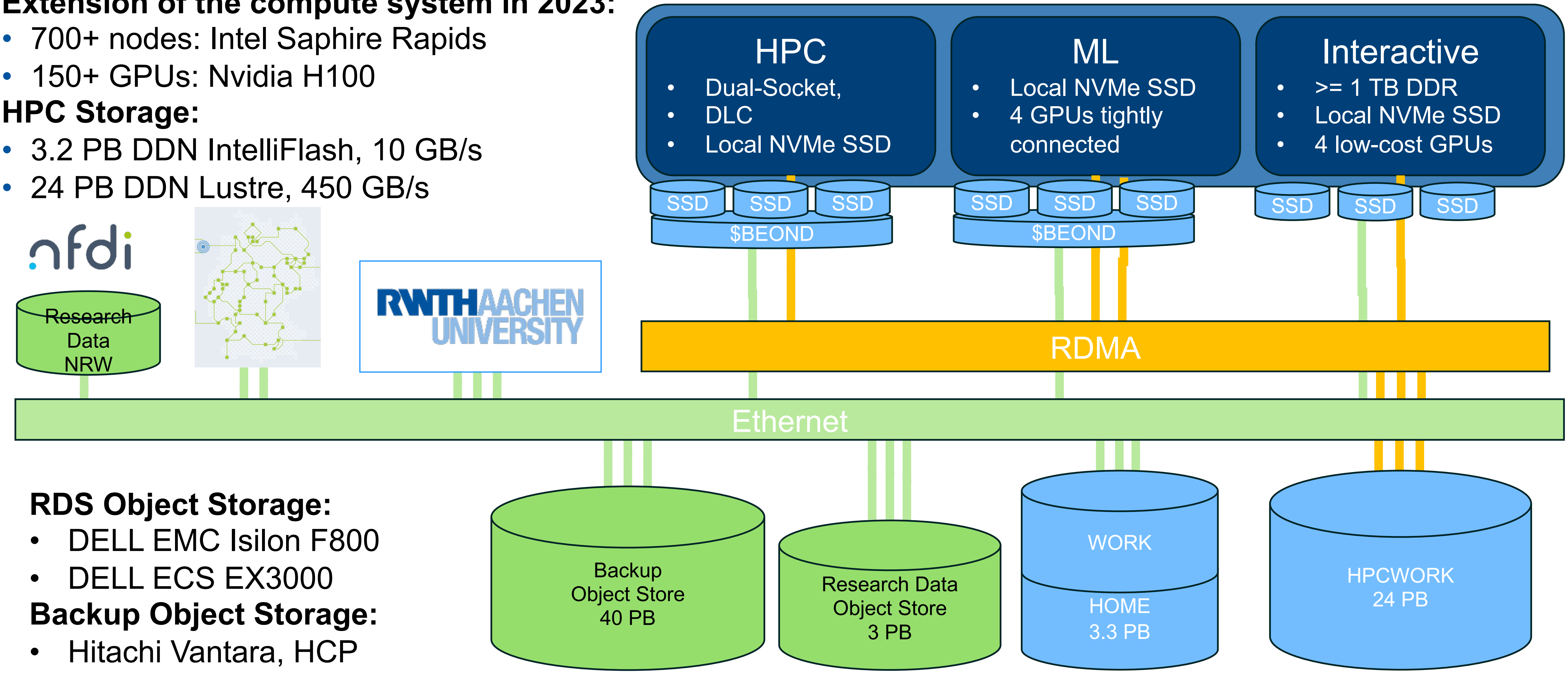
Staff and finances

- 360 employees
  (111 scientists, 130 staff, 46 apprentices, 74 students)
- About 42 M€ annual budget ,
  - 12M€ staff, 30M€ operations & invest

Compute and Storage for HPC and AI workflows: CLAIX

Extension of the compute system in 2023:
- 700+ nodes: Intel Saphire Rapids
- 150+ GPUs: Nvidia H100

HPC Storage:
- 3.2 PB DDN IntelliFlash, 10 GB/s
- 24 PB DDN Lustre, 450 GB/s



RDS Object Storage:
- DELL EMC Isilon F800
- DELL ECS EX3000

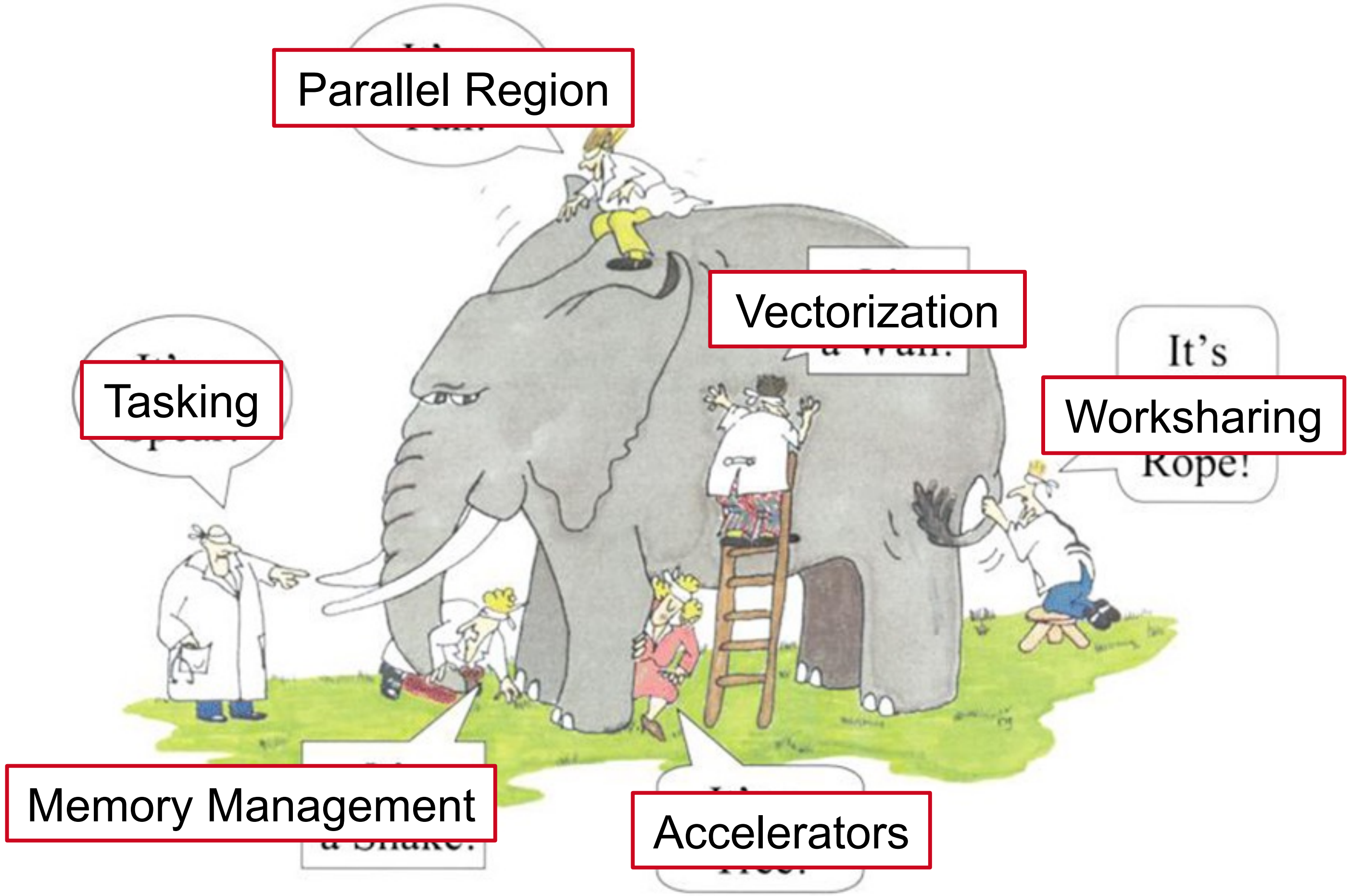
Backup Object Storage:
- Hitachi Vantara, HCP

Work on OpenMP

- Report from current work in the **Affinity** Subcommitee of the OpenMP Language Committee
- Credits: Jannis Klinkenberg (and others)

Is OpenMP as a programming model still alive?

- Parallel Region & Worksharing
- Tasking
- SIMD / Vectorization
- Accelerator Programming
- Memory Management
- …



Memory Management (since OpenMP 5.0)

- Did you know that you can … allocate in high-bandwidth memory?

```
#include <omp.h>
double *x = omp_alloc(N * sizeof(double), omp_high_bw_mem_alloc);
```

- Recent work:
  - New allocator traits for finer placement control
    - `partition`: partitioning of allocated memory over storage resources:
      `environment`, `nearest`, `blocked`, `interleaved`, `user` (allows writing and specifying custom partitioner)
    - `part_size`: specifies the size of parts allocated over storage resources
  - Allow upper bound and stride for `OMP_PLACES` together with abstract names
    - Examples: `OMP_PLACES=cores(4)` or `OMP_PLACES=ll_caches(1:2)`
  - Unify allocator and target memory runtime routines
    - Capability to allocate device memory with OpenMP allocators: new routines returning target memory spaces
    - Memory space containing storage resources accessible by all devices as requested

Experiments with Heterogeneous Memory

- **Memory Performance Characteristics: Bandwidth & Latency**
  - Interplay with NUMA effects
  - System: Intel Cascade Lake + Intel Optane

- **Bandwidth Benchmark: STREAM**
  - Clearly displays NUMA effects
  - Using `numactl` to specify
    - Specify where to run (`--cpunodebind`)
    - Specify which memory to use (`--membind`)
  - Evaluated different number of threads

- **Latency Benchmark: Intel Memory Latency Checker or Lmbench**
  - Pointer chasing (avoids HW prefetching)

Bandwidth Results – Cascade Lake + Optane (Regular STREAM Triad)

Architecture

```
CPU: Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz
Freq Govenor: performance
---------------------
available: 4 nodes (0-3)
node 0 cpus: 0 2 4 6 8 10 12 14 16 18
             20 22 24 26 28 30 32 34 36 38
node 0 size: 191936 MB
node 0 free: 178709 MB
node 1 cpus: 1 3 5 7 9 11 13 15 17 19 21 23
             25 27 29 31 33 35 37 39
node 1 size: 192016 MB
node 1 free: 179268 MB
node 2 cpus:
node 2 size: 759808 MB
node 2 free: 759794 MB
node 3 cpus:
node 3 size: 761856 MB
node 3 free: 761851 MB
node distances:
node   0   1   2   3
  0:  10  21  17  28
  1:  21  10  28  17
  2:  17  28  10  28
  3:  28  17  28  10
```

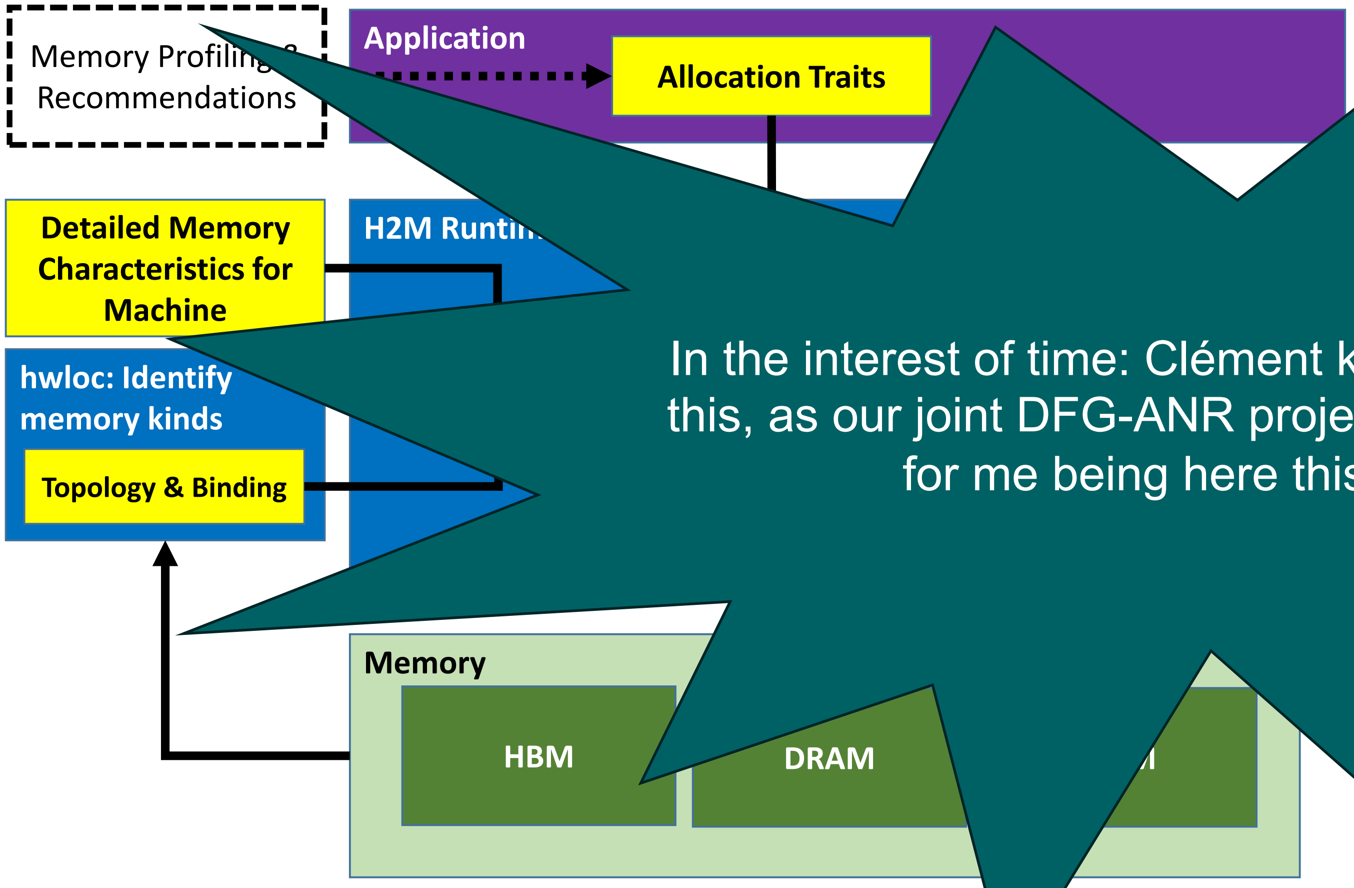
DRAM + Optane

Results for CPU-Domain 0 on Socket 0 [MB/s]

Threads	Mem-Domain 0	Mem-Domain 1	Mem-Domain 2	Mem-Domain 3	DRAM - Local vs Remote	NVM / DRAM
1	10484,30	5720,93	5156,73	2817,33	1,8326	2,0331
2	20258,73	11180,27	9700,57	4672,83	1,8120	2,0884
3	29931,40	16419,10	12629,97	6402,63	1,8230	2,3699
4	39393,77	21381,30	14952,13	7777,47	1,8424	2,6347
5	47635,00	26099,27	16738,10	8996,57	1,8251	2,8459
6	56124,63	30449,43	18069,27	9937,73	1,8432	3,1061
7	63814,83	34368,80	19117,40	10682,77	1,8568	3,3380
8	71127,77	37621,47	19992,70	11237,80	1,8906	3,5577
9	77052,30	40462,83	20548,63	11665,90	1,9043	3,7498
10	82760,67	42491,03	21132,23	11578,80	1,9477	3,9163
11	87170,37	43757,17	21255,03	11052,03	1,9921	4,1012
12	90497,07	44515,83	21544,50	10421,80	2,0329	4,2005
13	92723,13	45005,23	21687,73	9807,03	2,0603	4,2754
14	94877,07	45303,67	21752,83	8900,00	2,0942	4,3616
15	96342,97	45459,00	21711,43	7855,93	2,1193	4,4374
16	97184,43	45486,57	21658,70	6677,27	2,1366	4,4871
17	97578,23	45499,37	21555,20	5649,77	2,1446	4,5269
18	97749,70	45490,17	21565,00	4597,50	2,1488	4,5328
19	97817,47	45475,37	21562,40	3602,27	2,1510	4,5365
20	97713,80	45477,97	21374,57	2999,00	2,1486	4,5715

DRAM Sockets (Mem-Domain 0, 1) | **Optane Socket 0** (Mem-Domain 2) | **Optane Socket 1** (Mem-Domain 3)

H2M: Workflow and Concept



In the interest of time: Clément knows enough about this, as our joint DFG-ANR project H2M is the reason for me being here this week ☺

- **Allocation Strategy**
  - Generates placement decision / recommendation based on trait set for a single entity
- **imit Strategy**
  - Receives multiple allocation ...) and decides ... llocation order
  - ... ke use of allocation strategy
  - ... o able to modify traits or ... ion
- **ration Strategy**
  - ceives information about ... stered allocations and their ... t trait sets (that might have been ... pdated in the meantime) and decides how and where to move data

Data/Thread-to-Device Affinity (OpenMP 6.0) / 1

- **Idea:** Find devices that are close to the current thread

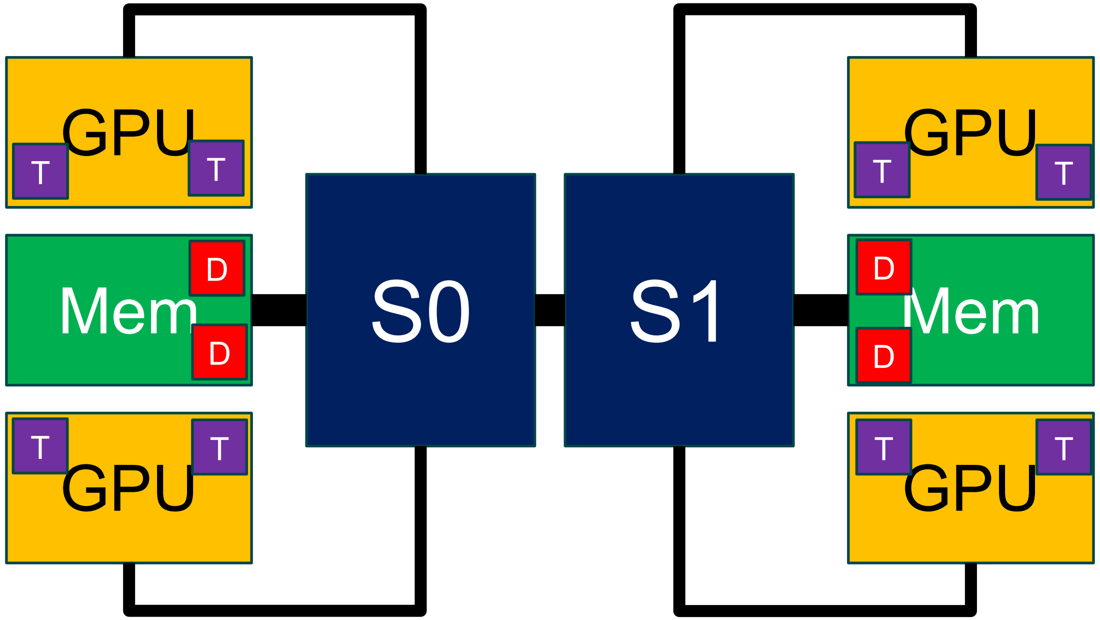
1. **Find devices that are close to the current thread**

```
int n=20;          // desired number of devices
int n_dev_found;   // actual number of devices
int dev_ids[n];
n_dev_found = omp_get_devices_in_order
                   (n, dev_ids, <trait_lowest_distance>);

#pragma omp target device(dev_ids[0]) // closest device
...
#pragma omp target device(dev_ids[n_dev_found-1])
```

2. **Use devices that are close to data used in target**

```
#pragma omp task affinity(data[start:len])
{
    #pragma omp target map(tofrom: data[start:len]) \
        device_affinity(data[start:len])
    {
        // content of the target task
    }
}
```

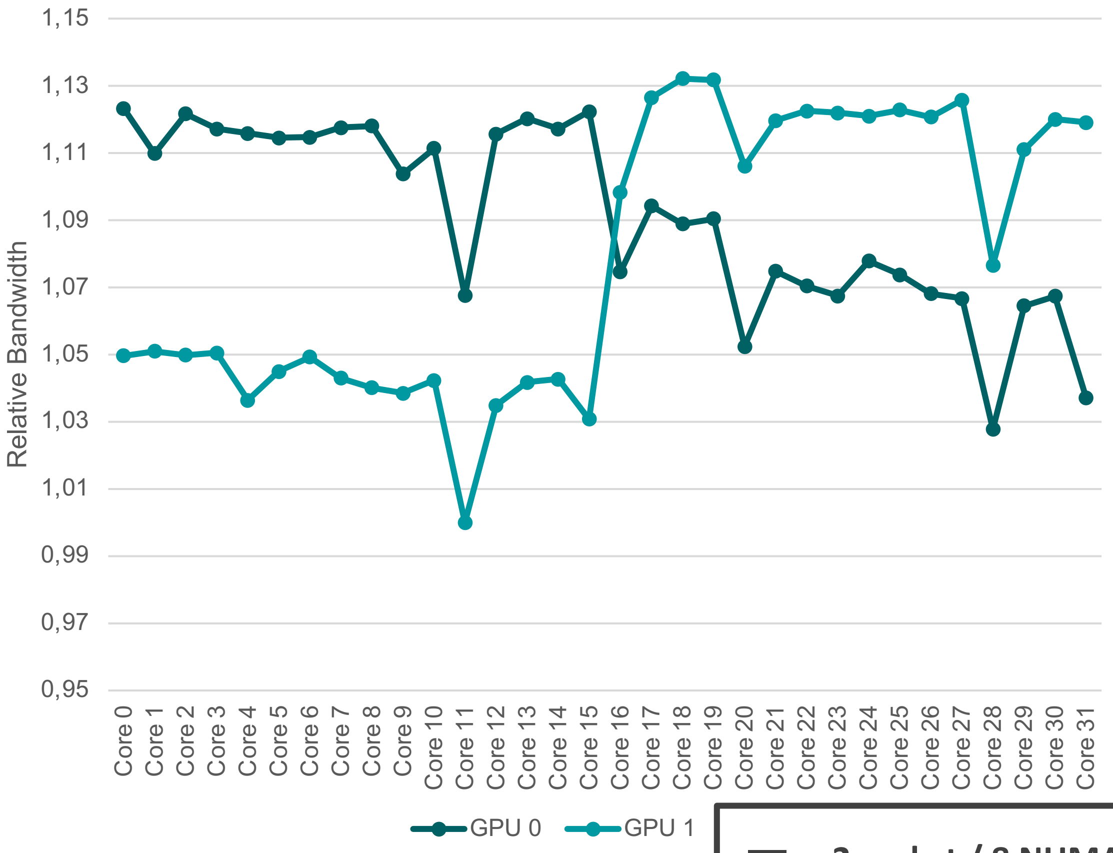


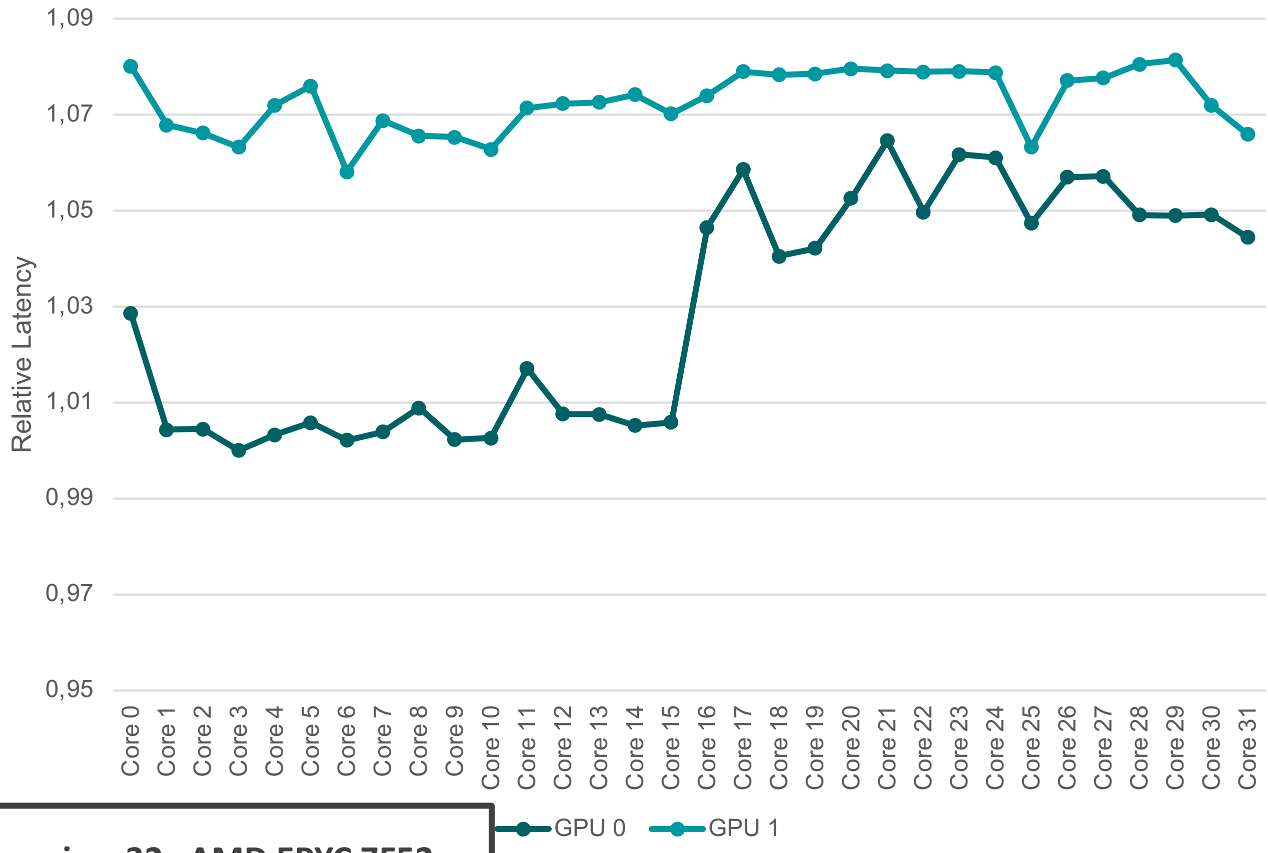
- **Scenario 1: Data not mapped to any device**
  → Use device that is close to data in host memory
- **Scenario 2: Offload to device that already holds part of required data**
  → Minimize data movement & reuse existing data

Data/Thread-to-Device Affinity (OpenMP 6.0) / 2

- **Question:** Does it matter?

Here: measurement on AMD MI210



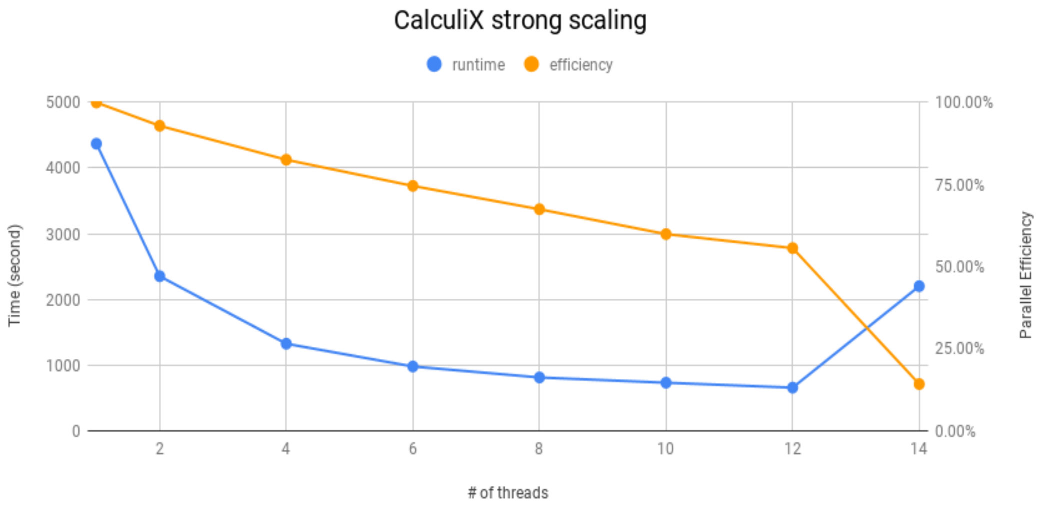


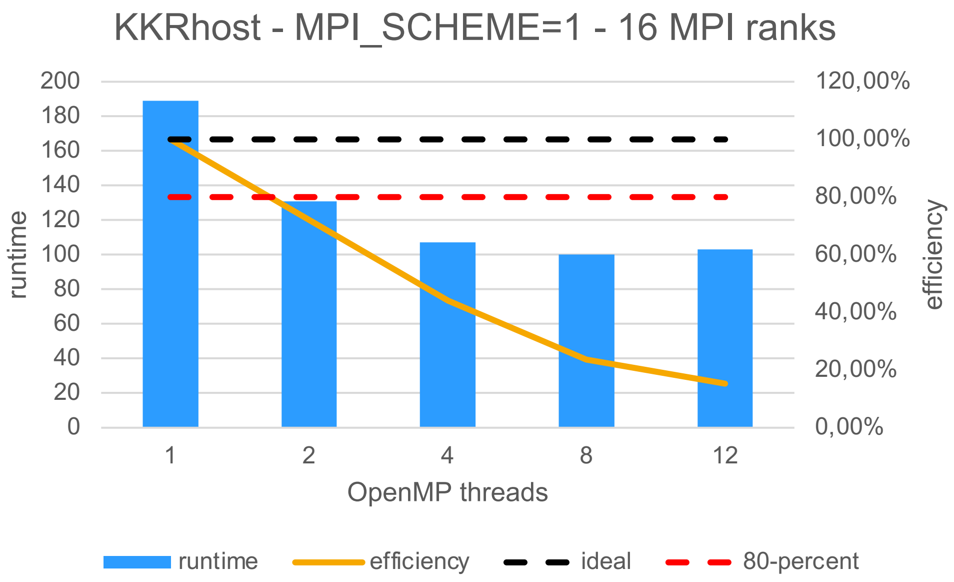
- 2 socket / 8 NUMA domains: 32x AMD EPYC 7F52
- 2x AMD MI210
  - GPU 0 connected to NUMA domain 2
  - GPU 1 connected to NUMA domain 7

Performance Engineering

- Report from current work in the EU CoE projects **POP**, POP2 and POP3
- Credits: Joachim Protze (and others)

Motivation





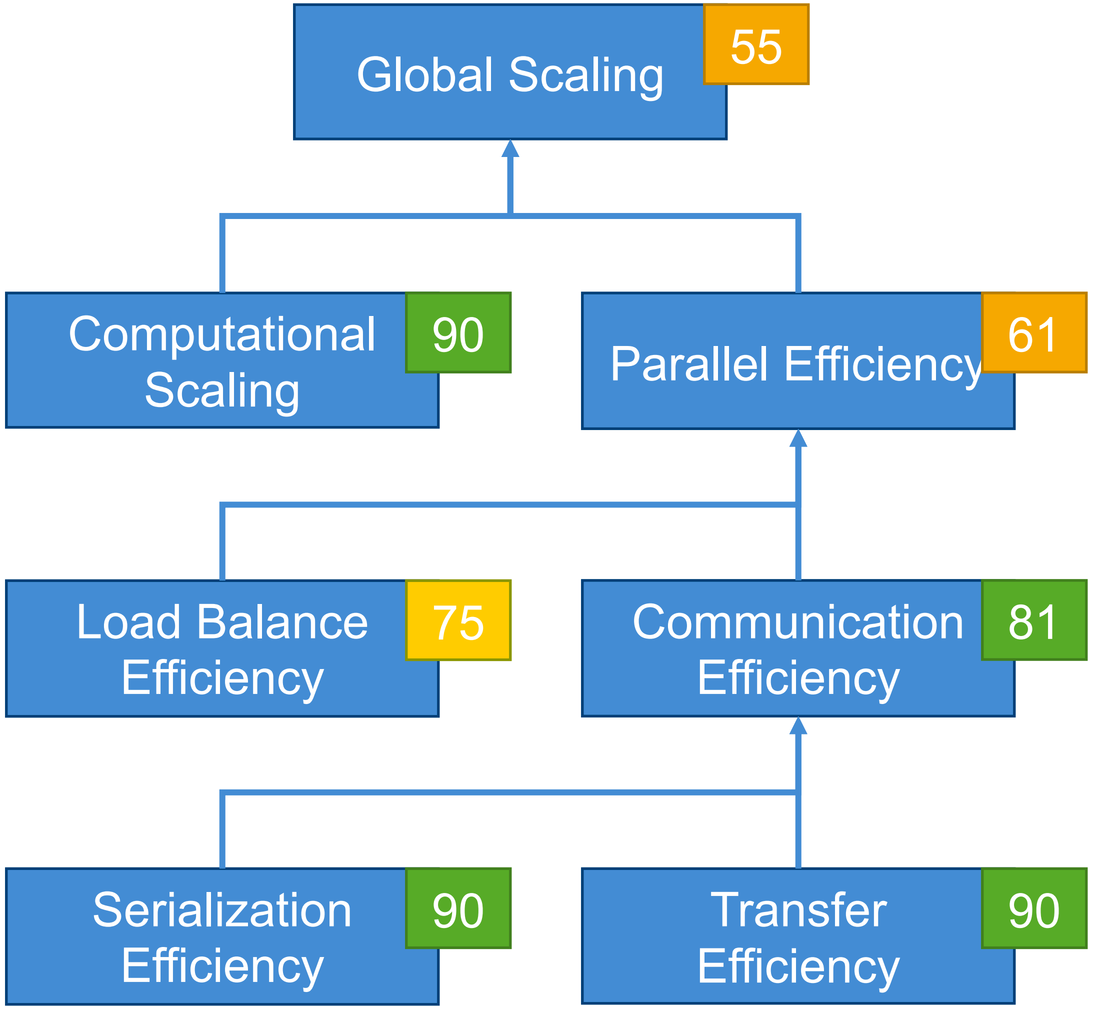
- Problem:
  - Why is my code getting inefficient at scale?
  - Multiple fundamental issues of (parallel) programming possible

- Solution: POP metrics
  - Standardized performance assessment independent of application / system
  - Goal: Enable simple verification of performance improvements

https://pop-coe.eu

Standard POP metrics

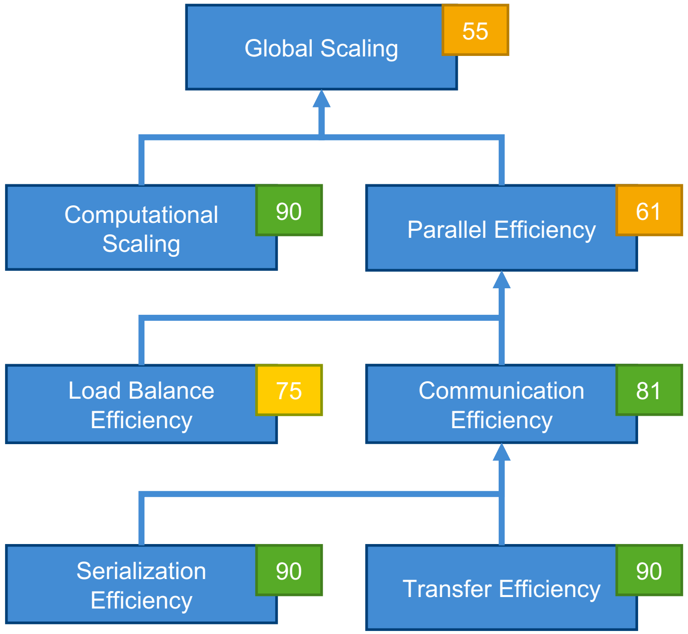
- Hierarchy of metrics
  - Aka *fundamental model factors*
- Highlight issues in the parallel structure of an application
- Parallel Efficiency breaks down into
  - Load balance
  - Serialization
  - Transfer
- Computational Scaling captures impact of scaling to node-level performance

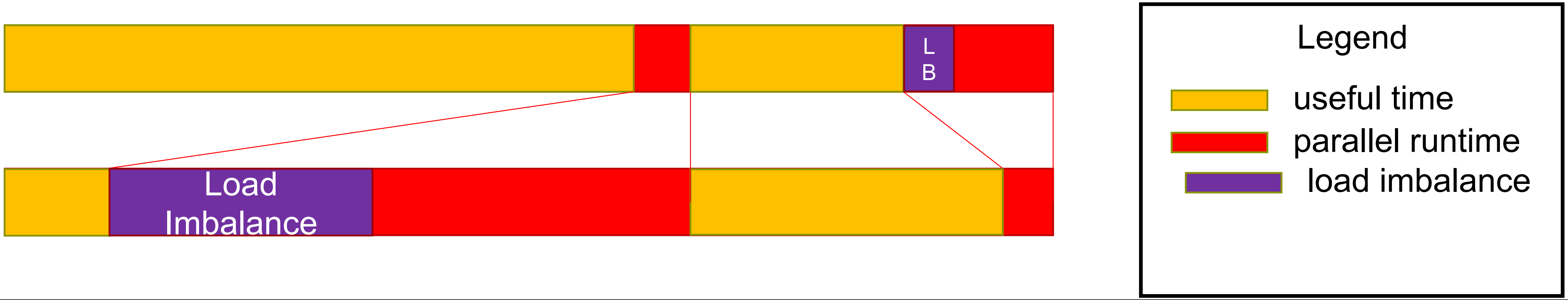


Standard POP metrics

Load Balance

- Reflects global imbalance of work between execution units
- $LB = \frac{avg(useful\ time)}{\max(useful\ time)}$
- *Useful time*: execution time outside parallel runtimes

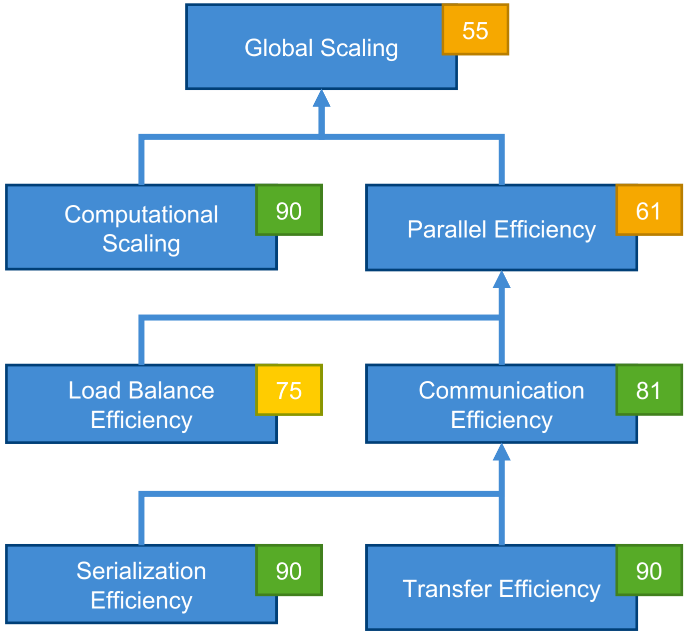


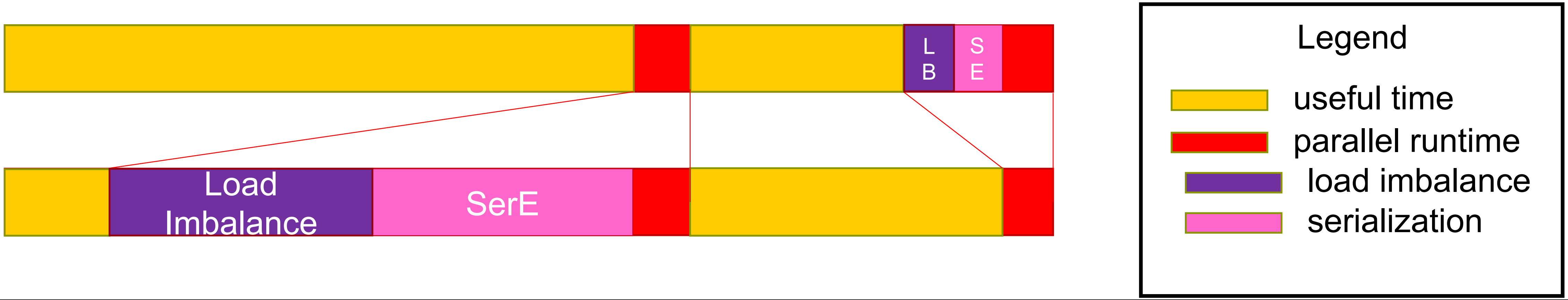


Standard POP metrics

Serialization Efficiency

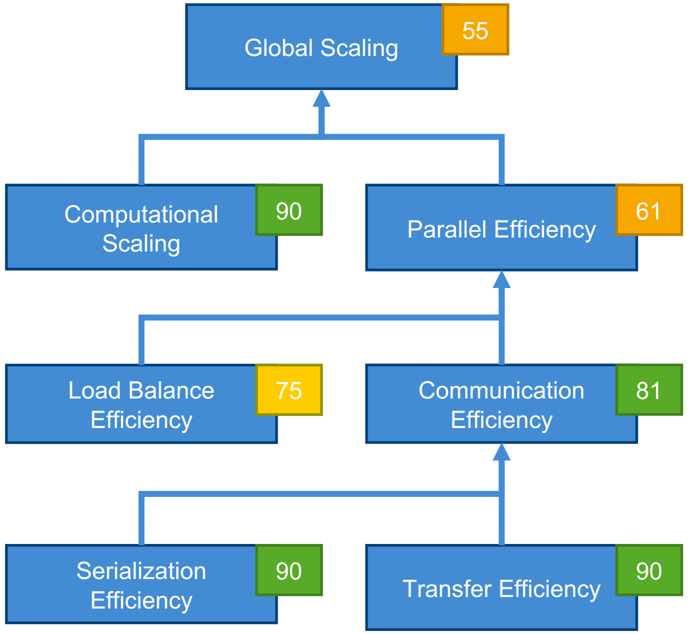
- Reflects moving imbalance of work between execution units, resp., alternating dependencies
- $SerE = \frac{max(useful\ time)}{ideal\ runtime}$
- *Ideal runtime*: execution time on an ideal machine with 0 communication cost (inf. BW / 0 lat)

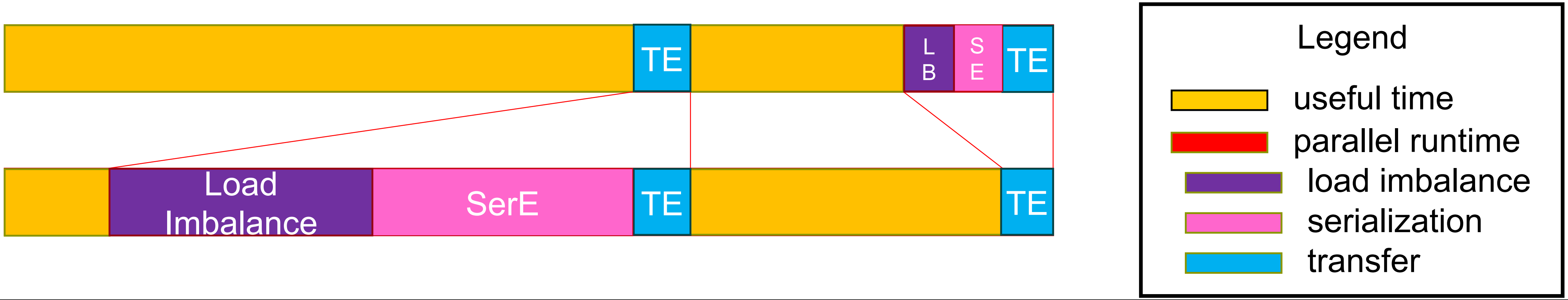




Transfer Efficiency

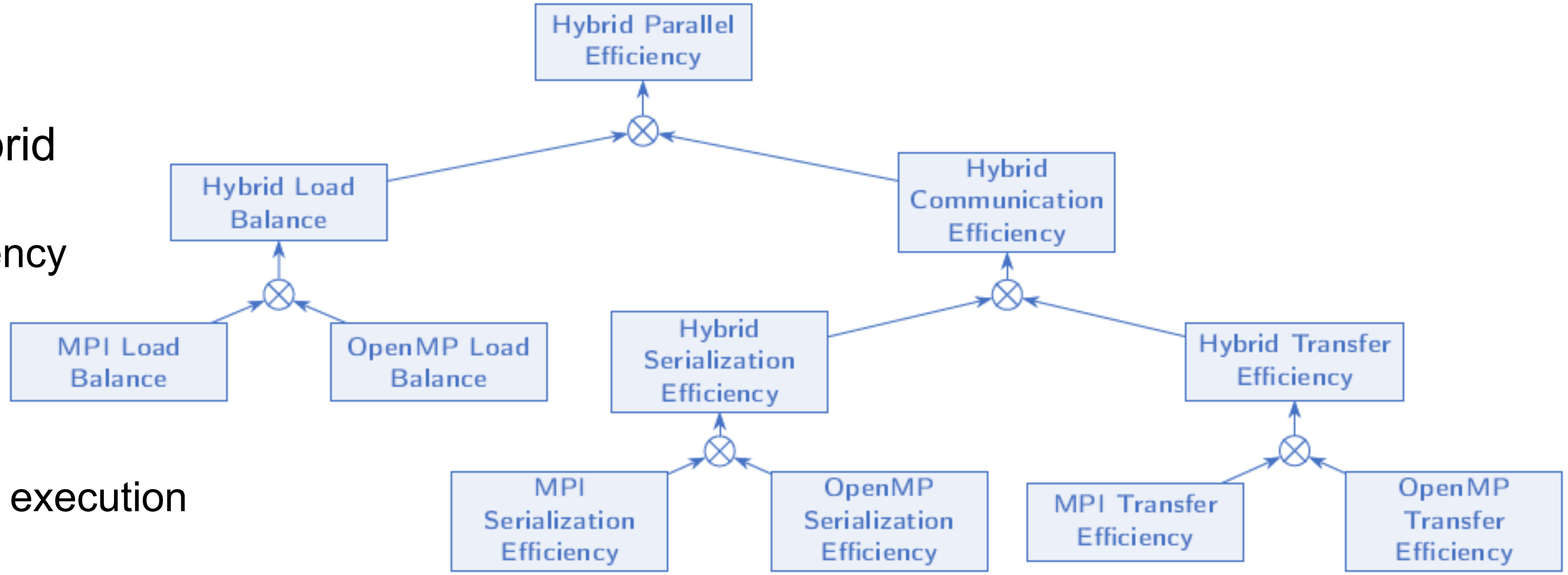
- Cost of transfer/communication/synchronization
- $TE = \frac{ideal\ runtime}{real\ runtime}$
- *Real runtime*: observed execution time





Critical path-based model

- Generalization of multiplicative hybrid metrics
  - Hybrid split of Communication Efficiency into programming models



- Idea:
  - Critical path = event path in program execution with longest duration
  - $runtime_{ideal} \approx critical\ path\ of\ useful\ compution$

- Prototype tool for „on-the-fly“ calculation of hybrid metrics
  - Enables metric calculation for applications with non hierarchical communication (e.g. MPI-Detach with detached tasks)

Reference: J. Protze, F. Orland, K. Haldar, T. Koritzius, C. Terboven, „On-the-Fly Calculation of Model Factors for Multiparadigm Applications“, Euro-Par 2022

Coupling HPC+AI

- Report from current work in the NHR4CES **Cross-sectional group** Parallelism & Performance
- Credits: Fabian Orland (and others)

Challenges at Scale (or: Exascale) / 1

Tasking may be employed to provide efficient and scalable coupling of SW components

- CFD simulations cannot live without modeling approaches
  - Becomes worse in multi-physics and multi-scale phenomena, or with interactions such as combustion
  - Will be complemented with data-based models

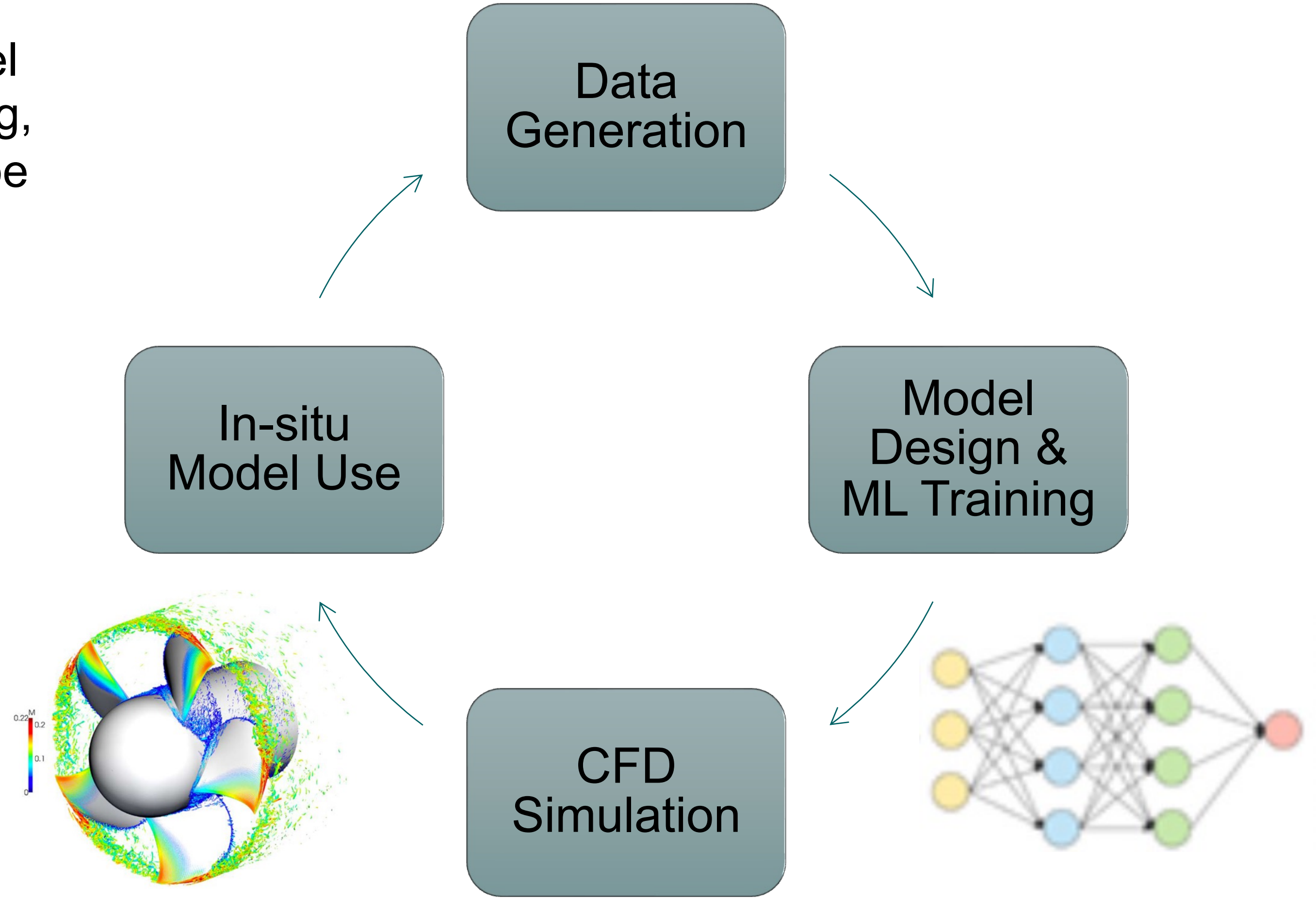
- At Exascale, the amount of data may exceed the Exabyte range for single simulation runs
  - In-situ data reduction, extraction and interpretation will hence be unavoidable

- To utilize HPC resources efficiently, software and workflows must scale to high CPU counts
  - In compute-drive applications, analyses are frequently a posteriori, necessitating to have the data on disk
  - As the field of parallel and scalable ML and DL is progressing, those algorithms become feasible to be intertwined with simulation codes implementing full loops

- Many pre-Exascale systems integrate homogeneous and heterogeneous compute nodes
  - ML and DL components can be accelerated

Tasking may be employed to provide efficient and scalable coupling of SW components

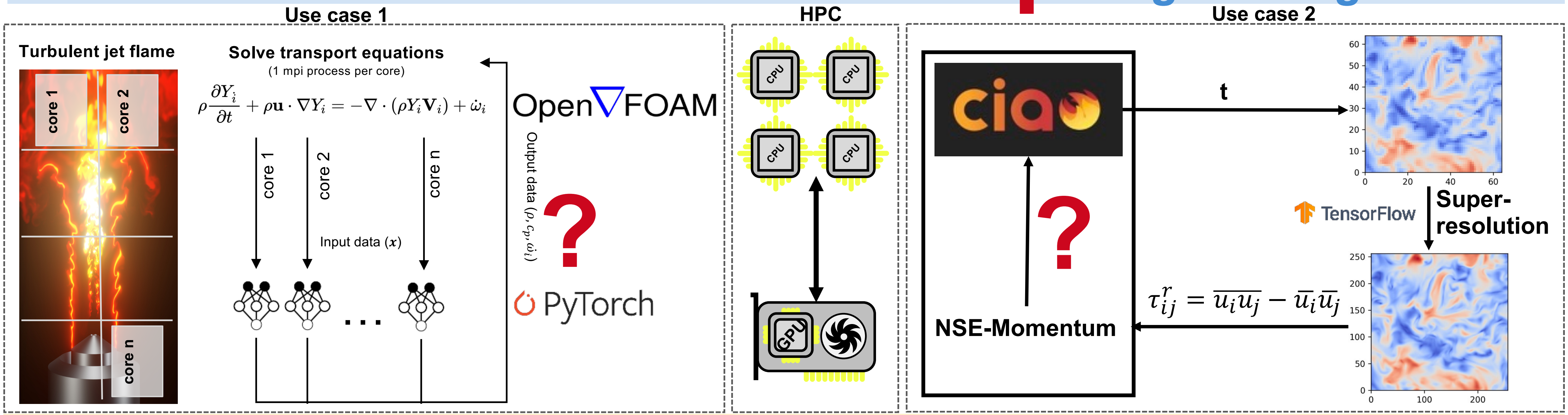
- Key expectation: As the field of parallel and scalable ML and DL is progressing, those algorithms become feasible to be intertwined with simulation codes implementing full loops



Research Questions

- A-posteriori evaluation of coupled CFD+ML simulation is open research ?

Engineering Science

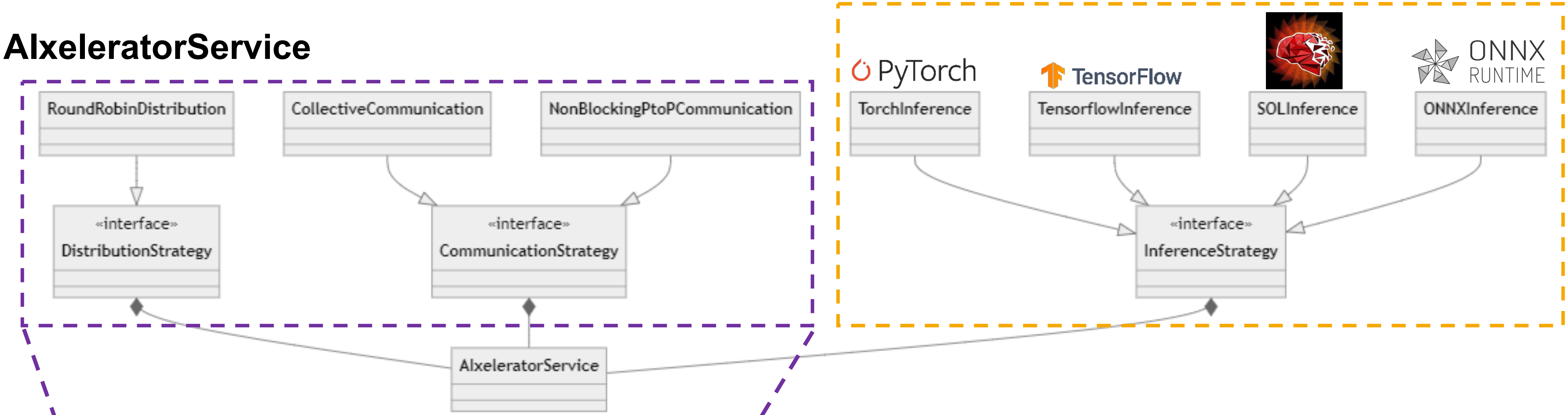


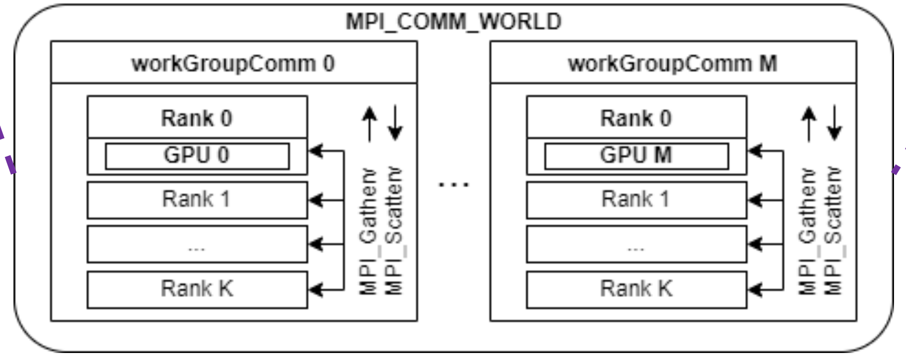
- How can we ***efficiently couple*** highly parallel (CFD) simulations with ML on ***heterogeneous architectures***?
- How can we ***model*** the performance of a coupled HPC-ML application?
- How can we ***optimize*** a coupled HPC-ML application?

HPC

Coupling

AIxeleratorService



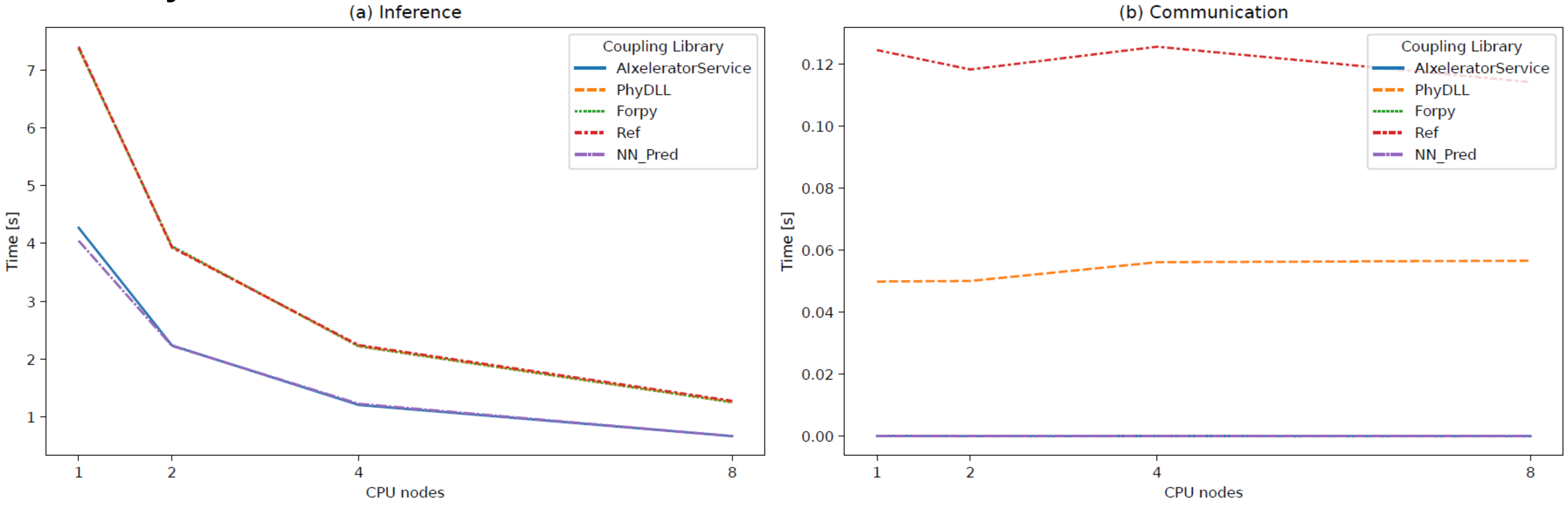


- C++ library with C & Fortran interfaces
- Hides MPI complexity from the user
- Supports multiple AI frameworks
- Different inference modes:
  - CPU only (no MPI communication)
  - GPU only
  - Hybrid CPU+GPU (MLP ✔ CNN ⚠)

Results – CIAO-AI DHIT

Ref + PhyDLL: 4 Python procs á 12 OpenMP threads per CPU node + 4-32 CIAO procs on additional node
Forpy, NN_pred, AIX: 4 CIAO procs á 12 OpenMP threads per CPU node

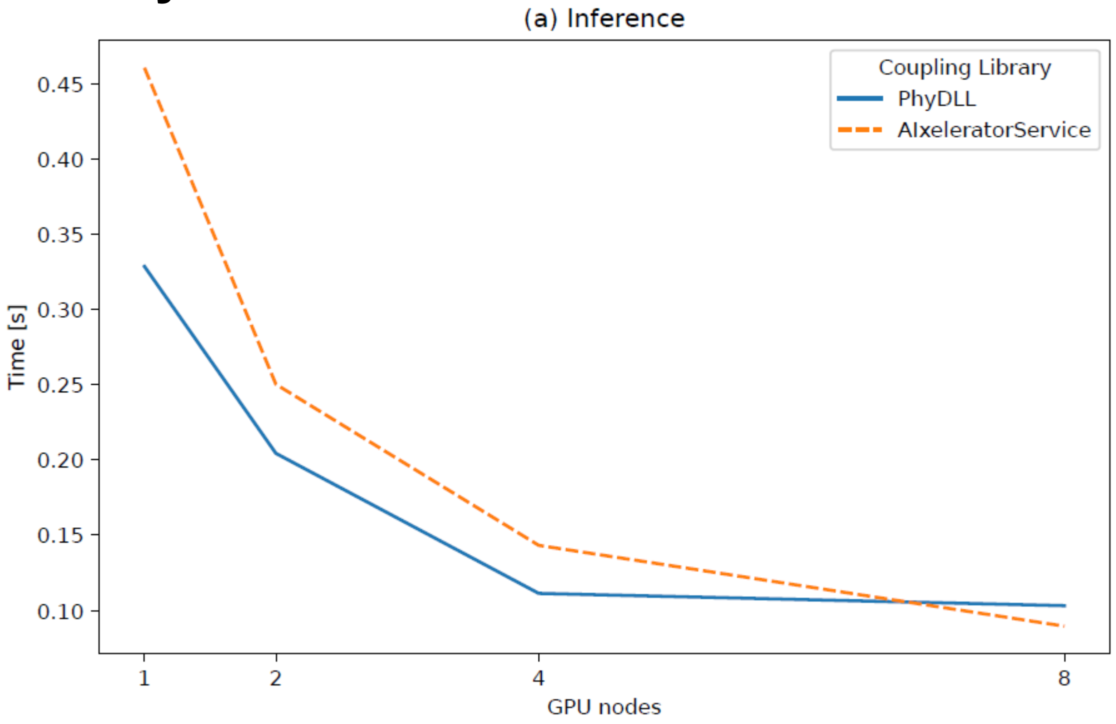
Scalability - CPU

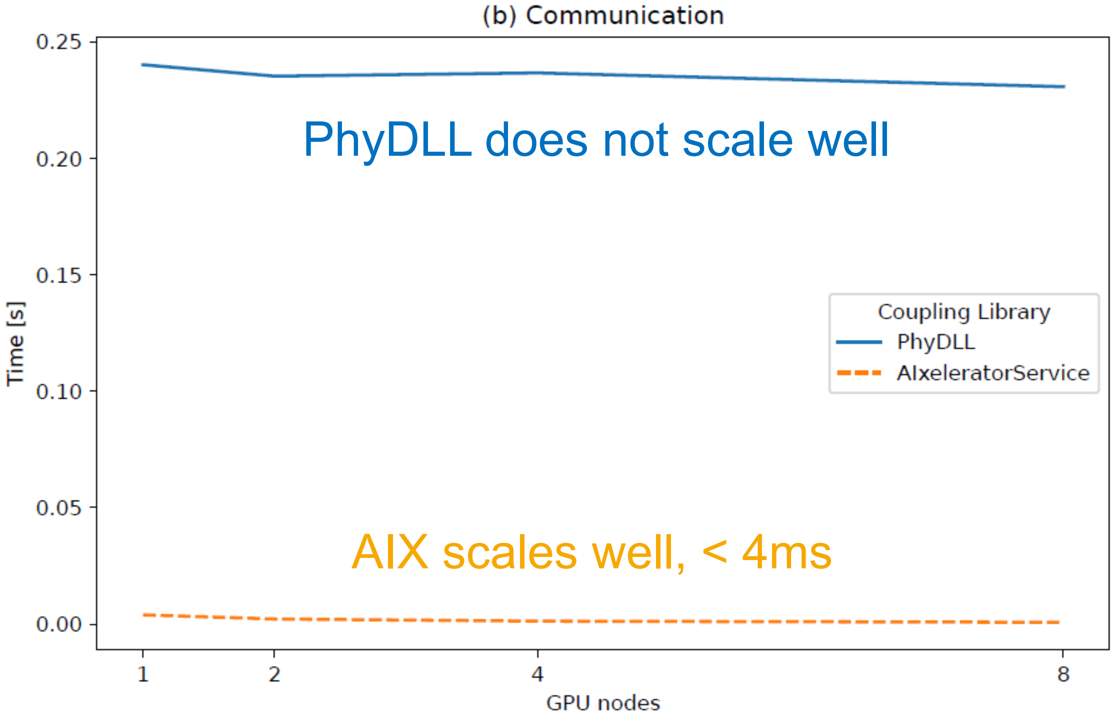


Results – CIAO-AI DHIT

4 CIAO procs per GPU node
(+ 2 Python procs per GPU node) for PhyDLL

Scalability - GPU





ENSIMA

TECHNISCHE UNIVERSITÄT DARMSTADT

Parallel Programming

RWTH AACHEN UNIVERSITY

JÜLICH Forschungszentrum

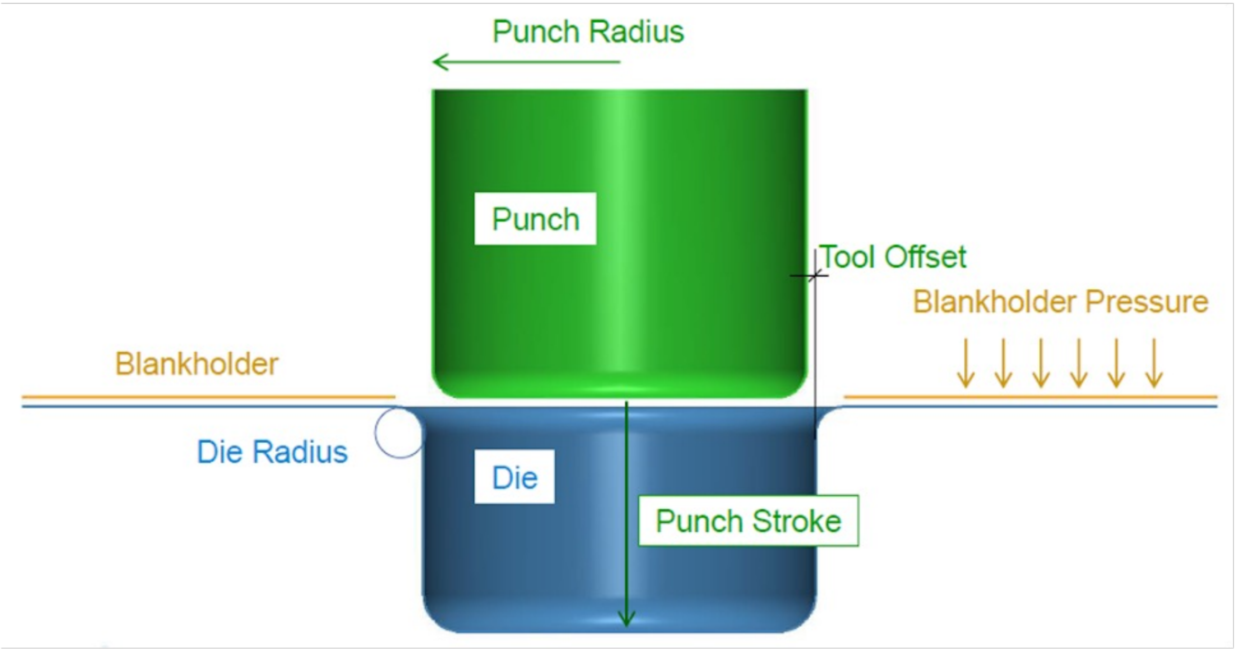
GESELLSCHAFT FÜR NUMERISCHE SIMULATION gns

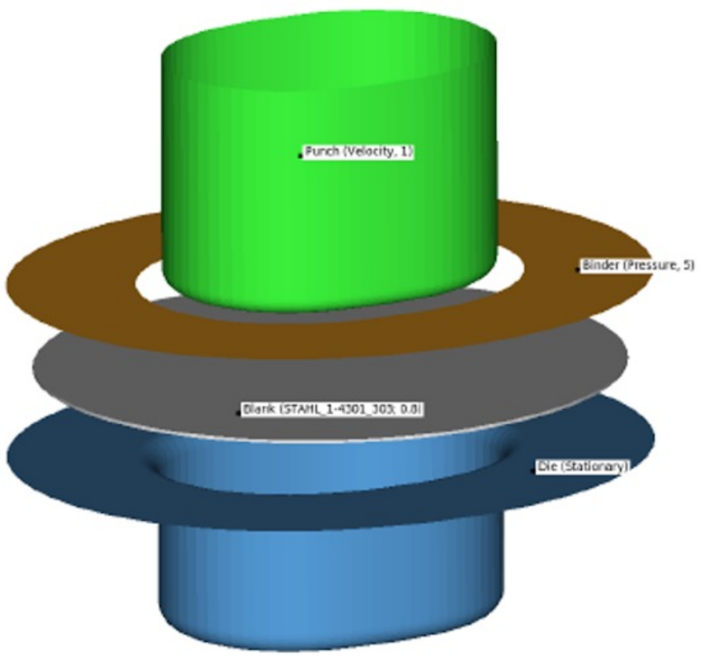
GNS Systems IT Services for Engineering

SIMCON GmbH
TH Würzburg-Schweinfurt

SPONSORED BY THE Federal Ministry of Education and Research

- Optimization of simulation-driven design processes – deep drawing of sheet metal



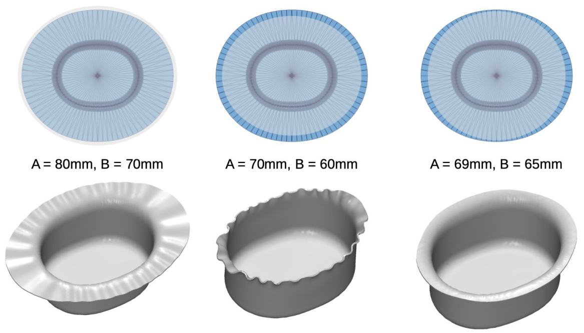


• OpenForm – Numerical simulation of deep drawing for design optimization

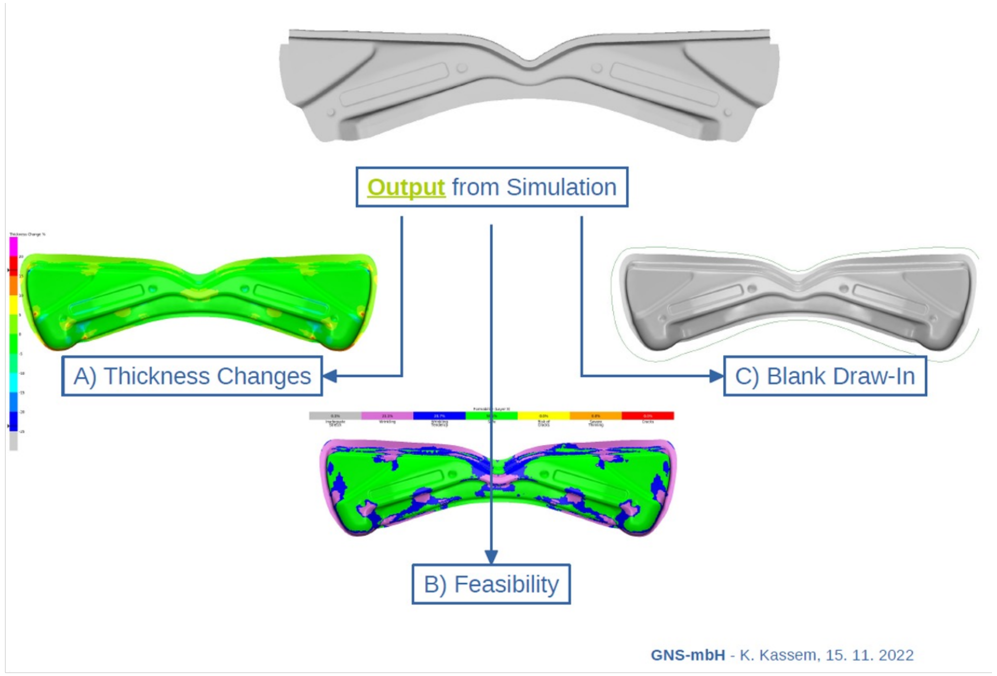
Simulation input

A: Geometry of the Forming Tools
Addendum Surfaces

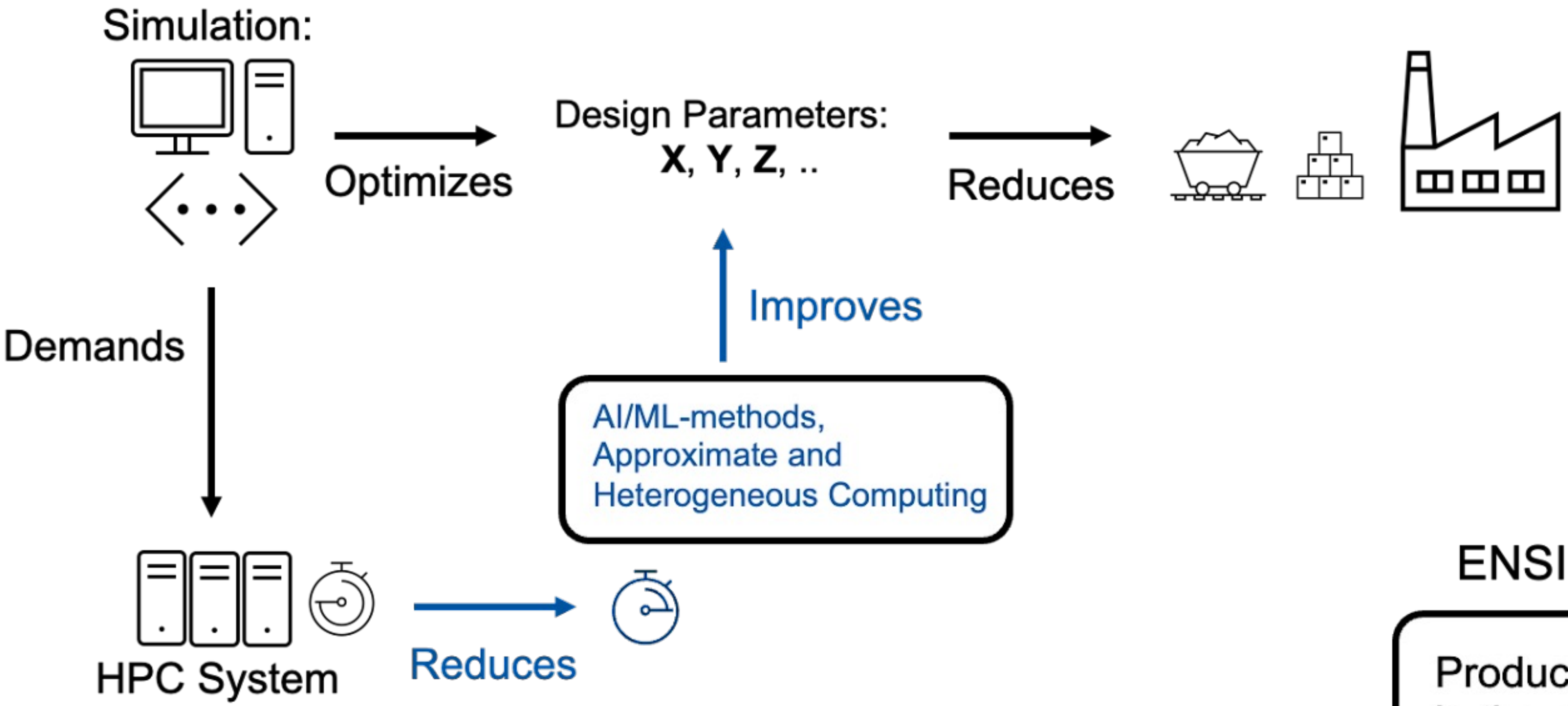
B: Initial Geometry and Properties of Blank
Outline
Thickness



Simulation output



Current Workflow

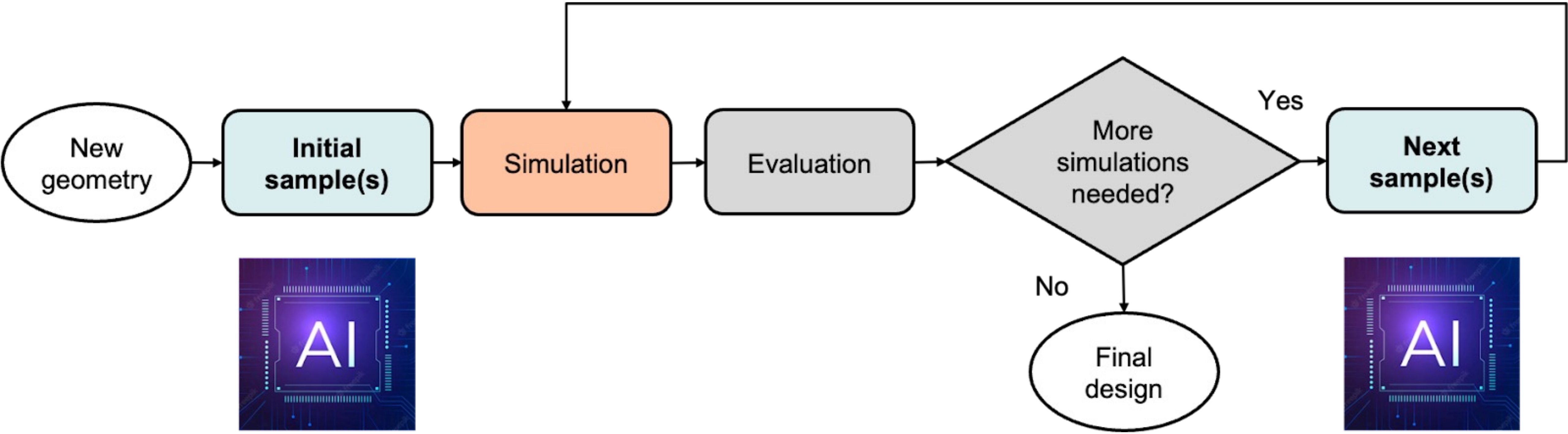


ENSIMA's use case:

Production of sheet metals
in the automotive industry

Trial and error: time and resource consuming

Resulting Optimized Workflow



Summary

Summary

- **Reims and Aachen are partner cities …**
  - https://aachen-reims.de/
  - https://amitiereimsaachen.blogspot.com/
  - … and we would be more than happy to partner with you on such topics ;-)

- **The compute architecture and memory subsystem are changing …**
  - … and “performance” becomes even more complex to achieve
  - … and “performance” becomes even more complex to assess

- **Integration of research results into OpenMP (and MPI): sustainability of research**

- **The applications are changing …**
  - … and require *More than HPC* to be made fit for the next decade!