High Performance numerics for Exascale: The Exa-Soft Project

François Trahay

Context

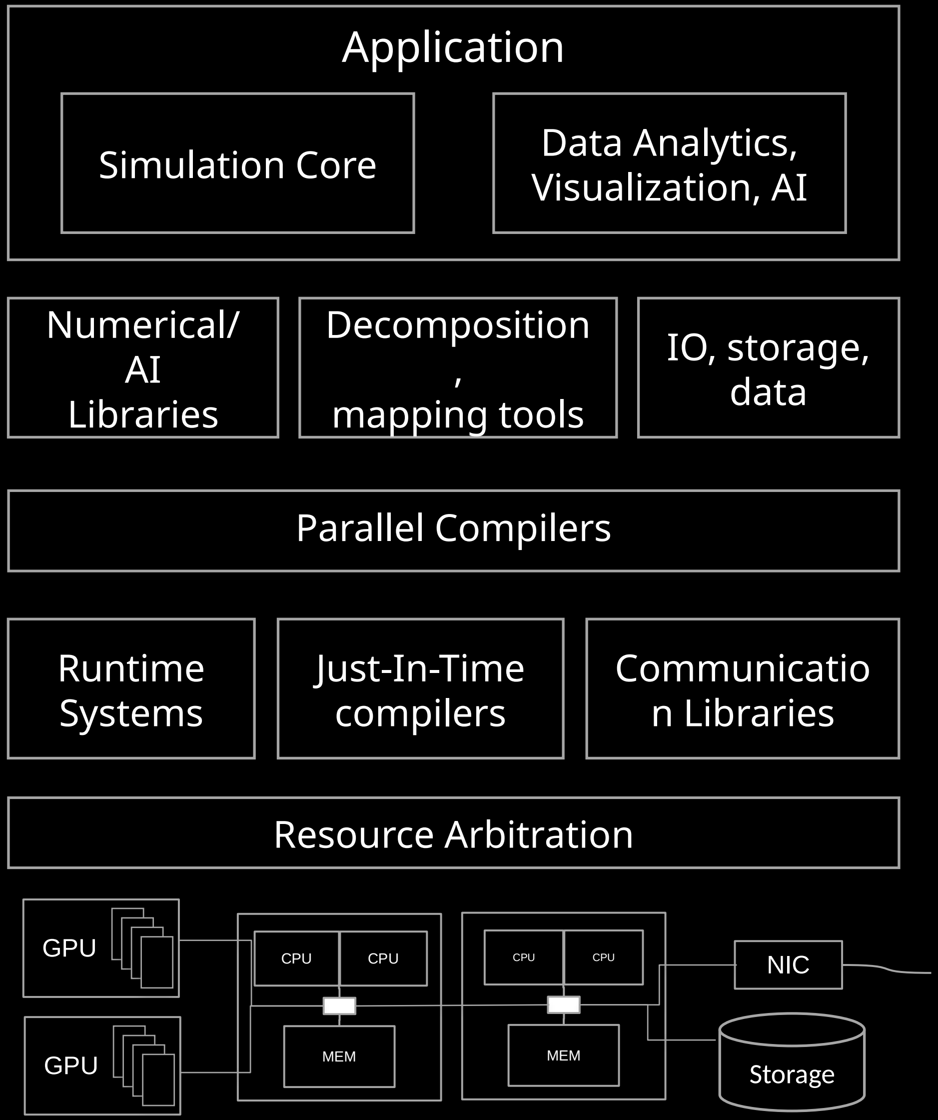
- PEPR NumPex
  - 40,8M€ project over 8 years
  - Goal: designing the software stack of Exascale supercomputers
- Main projects
  - PC1: Methods and Algorithms for Exascale – Exa-MA (Lead: Christophe Prud'homme, Helene Barucq)
  - PC2: HPC software and tools – Exa-SofT (Lead: Raymond Namyst, Alfredo Buttari)
  - PC3: Data-oriented Software and Tools for the Exascale – Exa-DoST (Lead: Gabriel Antoniu, Julien Bigot)
  - PC4: Architectures and Tools for Large-Scale Workflows – Exa-AtoW (Lead: François Bodin)
  - PC5: Development and integration project – Exa-DI (Lead: Jean Pierre Vilotte, Valérie Brenner)

Major concerns

- Thinking ”scalable”
- Exploiting heterogeneous, multi-GPU platforms
    - (Dynamic) code generation
    - Scheduling of computations
    - Data management
- Writing portable and composable code
- Providing performance/energy feedback to the user

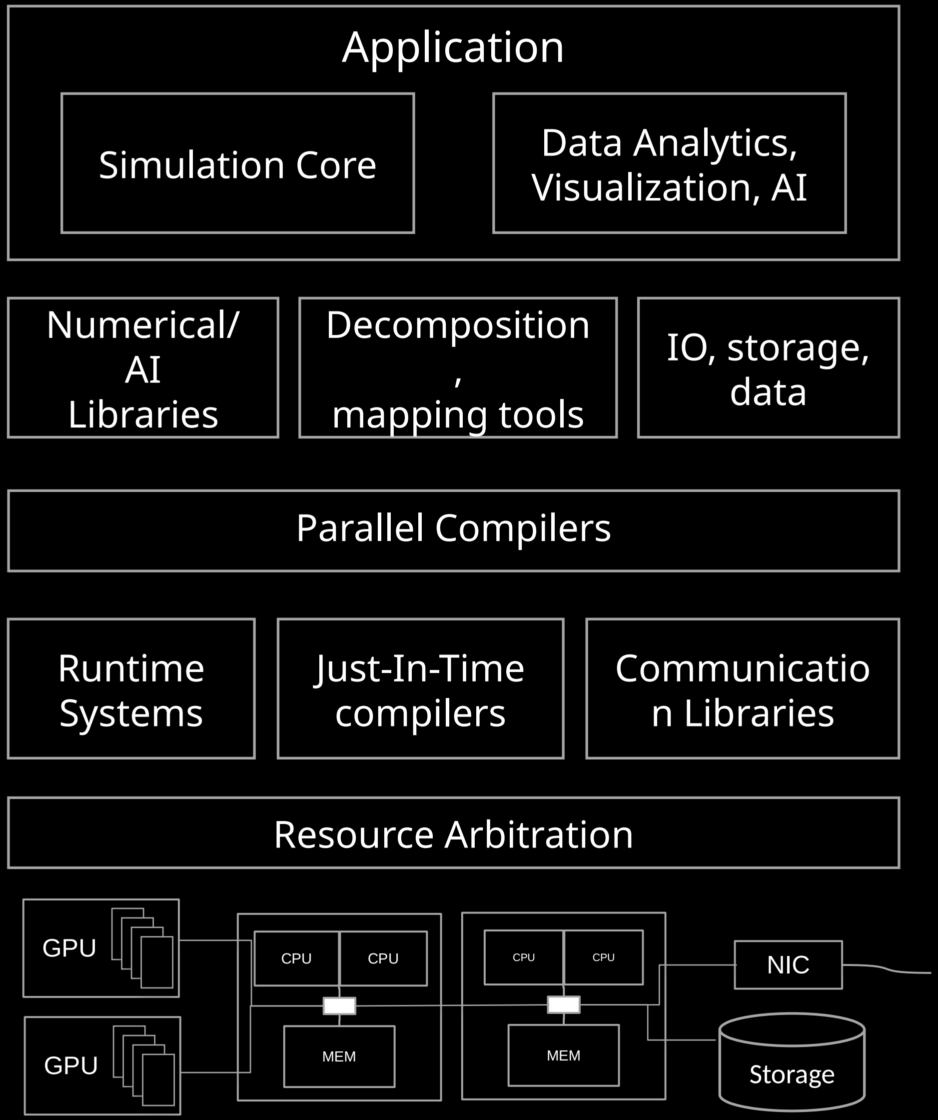
Vision

- Holistic approach
  - Contribute to a sound, consistent software stack
    - Most components should fit together!
  - Bridge the gap between existing languages/software/tools
  - Integrate state-of-the-art research results
  - Demonstrate relevance on representative applications



Vision

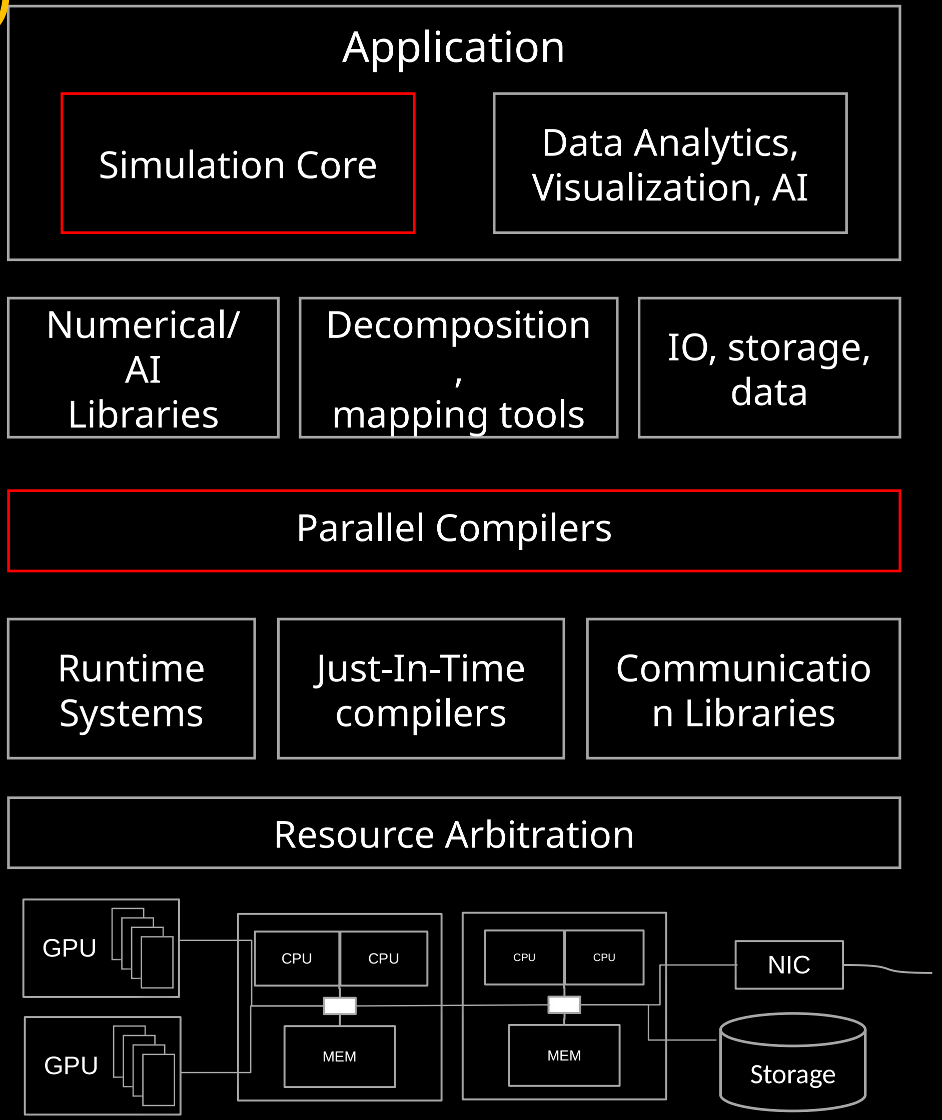
- Focus on modern, C++-based programming approaches
- Rely on runtime systems for improved efficiency and portability
- Use in-the-loop performance analysis
- Favor research & development between different teams
  - E.g. PhD subjects spreading across multiple WP



Work plan

- High-level approaches for developing efficient and composable parallel software (WP1)
- Just-in-Time code optimization with continuous feedback loop (WP2)
- Runtime Systems at Exascale (WP3)
- Portable, scalable numerical building blocks and software (WP4)
- Performance analysis and prediction (WP5)
- Energy profiling and control (WP6)

High-level approaches for developing efficient and composable parallel software (WP1)



- **Pilots:** Christian Perez (Inria Lyon), Marc Pérache (CEA/DAM/DIF)
- **Tasks**
- High level programming model for heterogeneous architectures
  - Milestones: M24: Initial version of C++ extension, M24: Start porting demonstrators on IFPEN applications, M60: Final version of C++ extension
- Tools for parallel heterogeneous scientific application at scale
- Foundation of an HPC Composition Model
- High level data description and partitioning for reusable parallel building blocks
  - M48: Final version of an HPC composition framework including data description and integration with runtimes

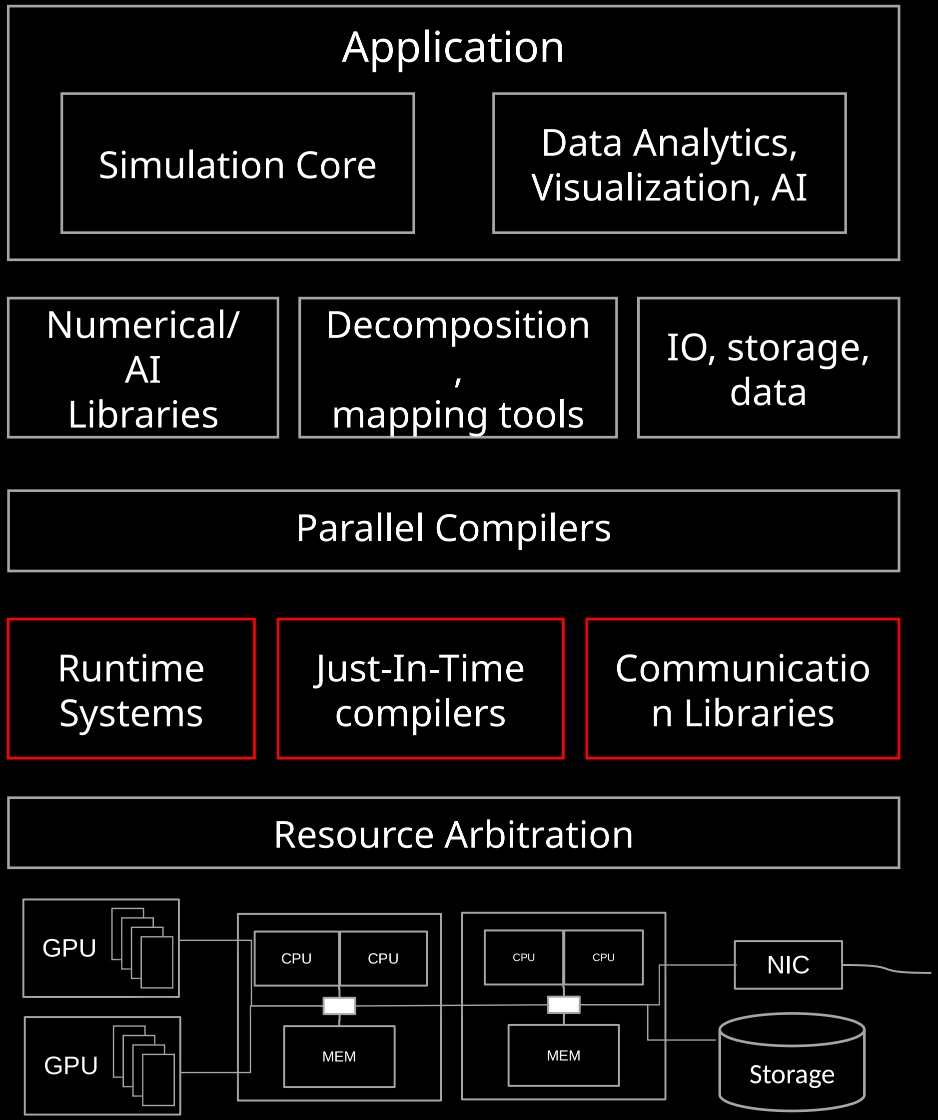
Just-in-Time code optimization with continuous feedback loop (WP2)

Pilots: Philippe Clauss (Univ. Strasbourg), Thierry Gautier (Inria Lyon)

Directions: tighter integration of runtime systems and just-in-time compilers

Tasks

- Runtime multi-versioning of parallel tasks
- Resource-aware tasks generation
- Specialization-based dynamic parallelization of sparse codes
- Integration and unification of runtime mechanisms

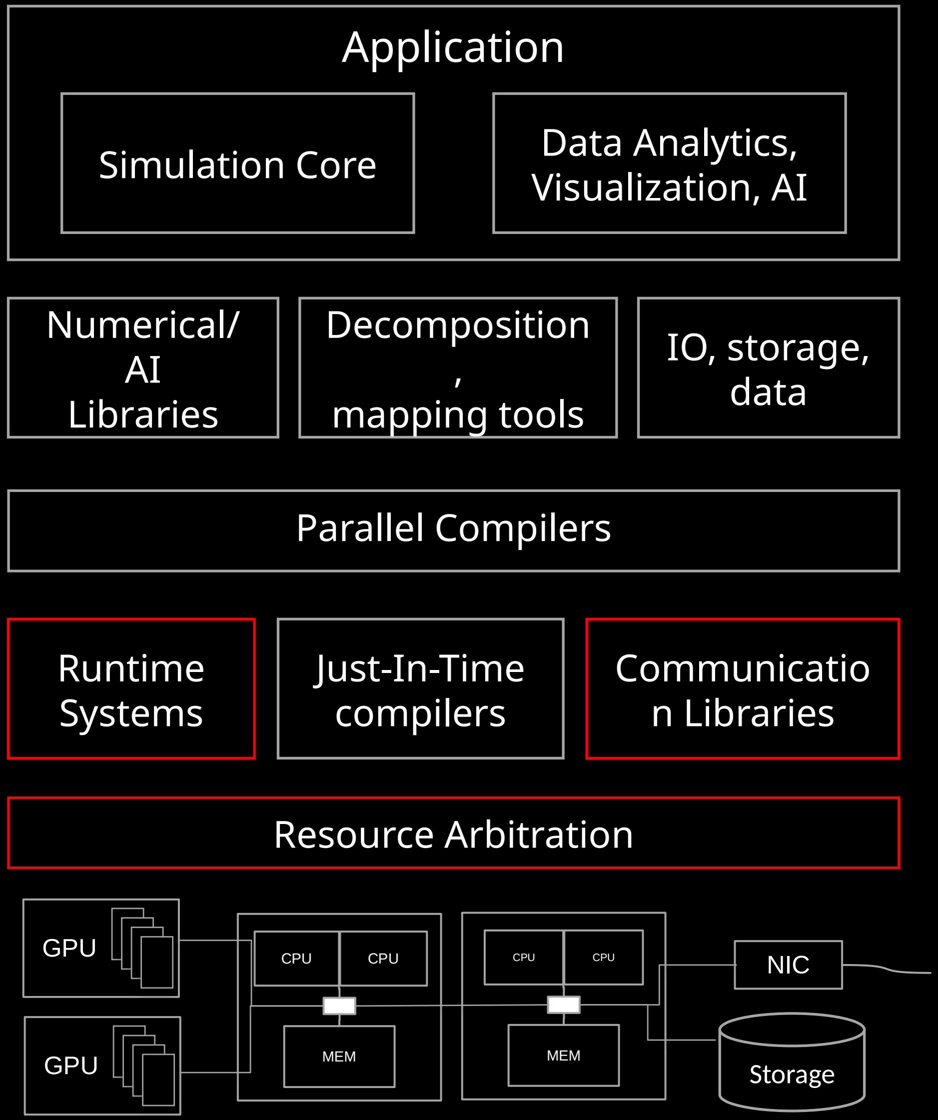


Runtime Systems at Exascale (WP3)

Pilot: Samuel Thibault (Univ Bordeaux)

Tasks

- Integration of asynchronous network communications scheduling and local task scheduling
- High-level data description and partitioning mechanisms
- Data placement in heterogeneous memory levels
- Fault Tolerance for large-scale systems

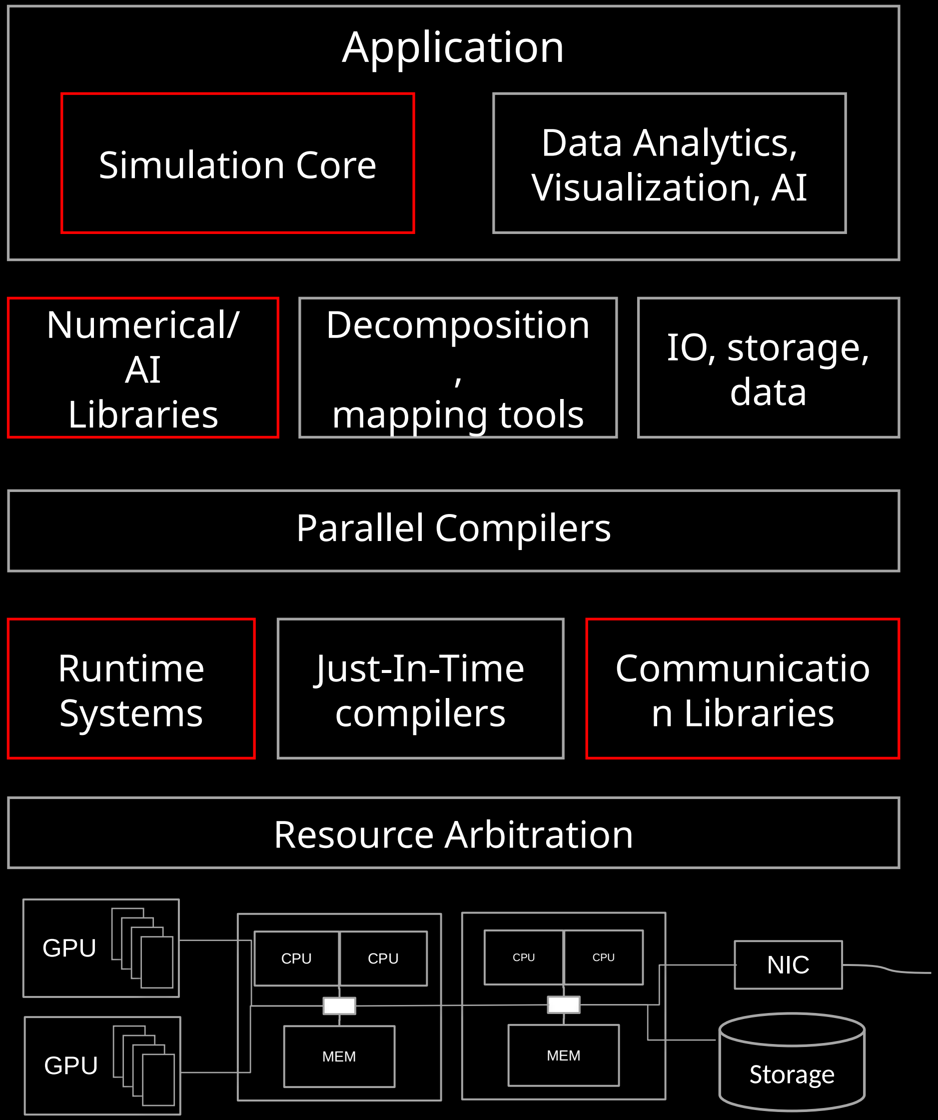


Portable, scalable numerical building blocks and software (WP4)

Pilots: Marc Baboulin (Univ. Paris-Saclay), Abdou Guermouche (Univ. Bordeaux)

Tasks

- Composability of numerical libraries
- Expression of scalable algorithms for dense and sparse (direct) linear algebra using task-based programming
- Efficient implementation of approximate computing algorithms
- Sparse and dense tensor computations using task-based algorithms
- Extension of Chameleon to small dimension tensors for large distributed systems with applications to deep neural networks



Performance analysis and prediction (WP5)

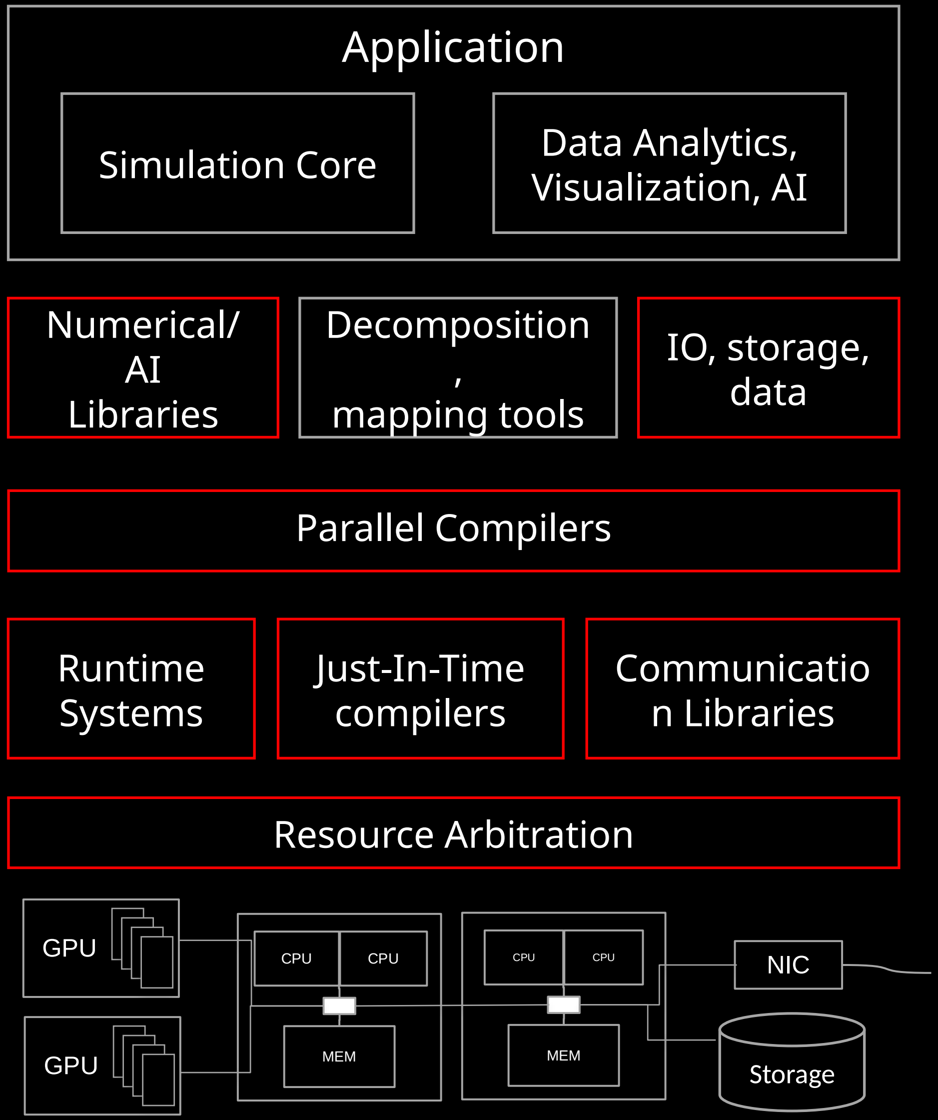
Pilot: François Trahay (Telecom SudParis)

Directions

Scalable tool suite for energy measurement and management:

Tasks

- Scalable tracing tool (TSP)
- System-wide post-mortem trace analysis (Polaris)
- Fine-grained energy measurements (TSP+STORM)
- On-the-fly performance analysis that guides the runtime system (TSP)



Energy profiling and control (WP6)

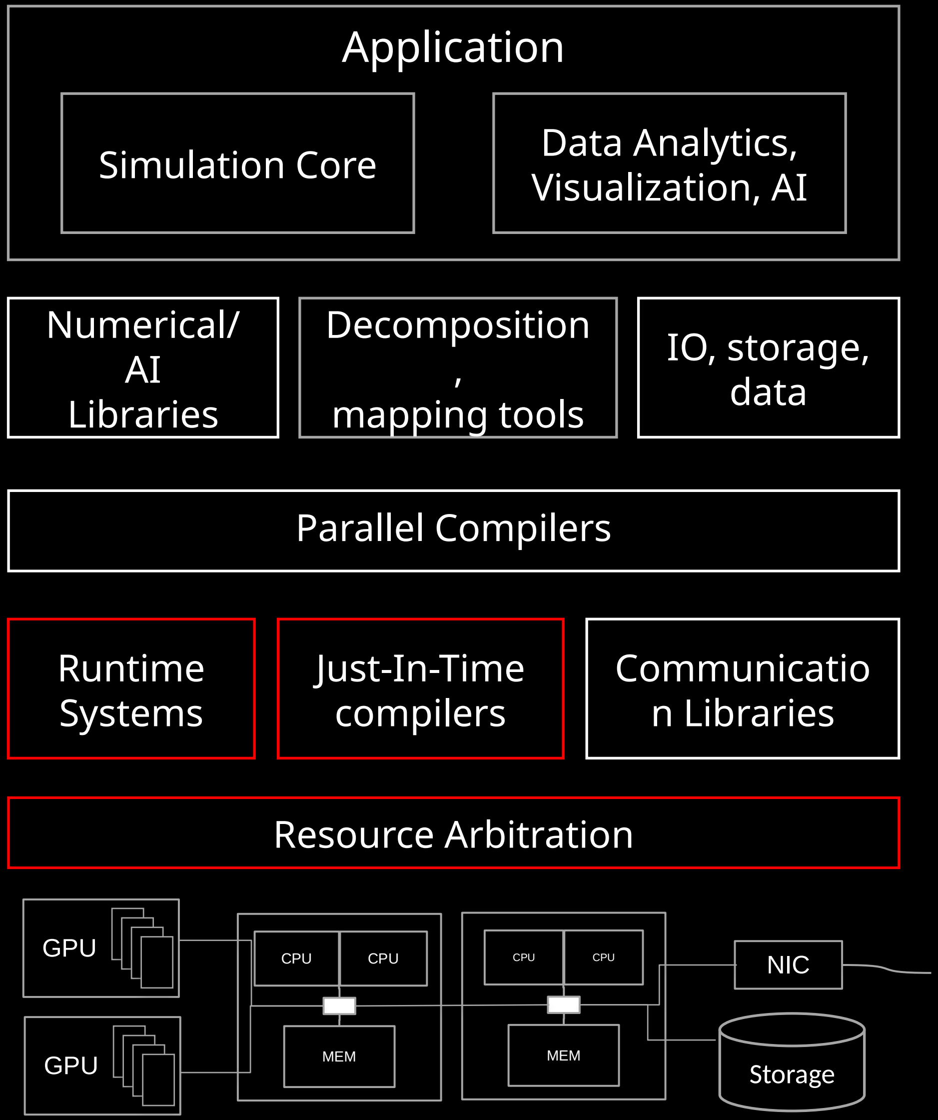
Pilots: George Da Costa (University of Toulouse), Amina Guermouche (Bordeaux-INP)

Directions

- Fine-grain energy monitoring
- Energy-aware task scheduling (at the node/application level)
- Energy-aware job scheduling (at the cluster level)

Tasks

- Fine-grained energy measurements
- Power and performance models
- Energy-aware scheduling algorithms
- Cluster-level power measurement
- Energy-aware job scheduling and feedback



Major challenges are ahead…
Exciting times!

Scalable tracing tool (T5.1)

Who: Hadrien Guelque, Valentin Honoré, François Trahay

Problem

Trace formats store events sequencially

→ Traces become huge, processing them takes hours

Proposal: Hierarchical trace format

- On the fly detection of function entry/exit, loops, etc.
- Lossless compression of counters (timestamps, hardware counters, etc.)
- Lossy compression of counters (timestamps, hardware counters, etc.)

→ small traces

→ fast processing of traces