

# 2025 Activity Report

RESEARCH CENTRE: Inria Centre at the University of Bordeaux  
IN PARTNERSHIP WITH: Bordeaux INP, Université de Bordeaux

---

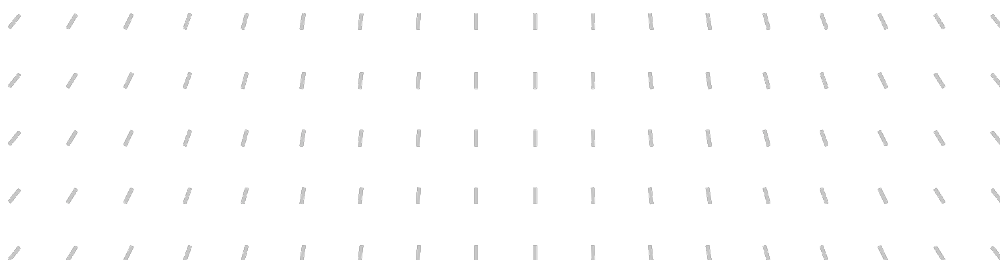
Project-Team

## TADAAM

Topology-aware system-scale data management for  
high-performance computing

---

*In collaboration with* Laboratoire Bordelais de Recherche en Informatique (LaBRI)



## **Project-Team TADAAM**

*Creation of the Project-Team: 2017 December 01*

Each year, Inria research teams publish an Activity Report presenting their work and results over the reporting period. These reports follow a common structure, with some optional sections depending on the specific team. They typically begin by outlining the overall objectives and research programme, including the main research themes, goals, and methodological approaches. They also describe the application domains targeted by the team, highlighting the scientific or societal contexts in which their work is situated. The reports then present the highlights of the year, covering major scientific achievements, software developments, or teaching contributions. When relevant, they include sections on software, platforms, and open data, detailing the tools developed and how they are shared. A substantial part is dedicated to new results, where scientific contributions are described in detail, often with subsections specifying participants and associated keywords. Finally, the Activity Report addresses funding, contracts, partnerships, and collaborations at various levels, from industrial agreements to international cooperations. It also covers dissemination and teaching activities, such as participation in scientific events, outreach, and supervision. The document concludes with a presentation of scientific production, including major publications and those produced during the year.

## Keywords

### Computer sciences and digital sciences

- A1.1.1. – Multicore, Manycore
- A1.1.2. – Hardware accelerators (GPGPU, FPGA, etc.)
- A1.1.3. – Memory models
- A1.1.4. – High performance computing
- A1.1.5. – Exascale
- A1.1.9. – Fault tolerant systems
- A1.2.4. – QoS, performance evaluation
- A1.6. – Green Computing
- A2.1.7. – Distributed programming
- A2.2.2. – Memory models
- A2.2.3. – Memory management
- A2.2.4. – Parallel architectures
- A2.2.5. – Run-time systems
- A2.6.1. – Operating systems
- A2.6.2. – Middleware
- A2.6.4. – Ressource management
- A3.1.2. – Data management, quering and storage
- A3.1.3. – Distributed data
- A3.1.8. – Big data (production, storage, transfer)
- A3.4. – Machine learning and statistics
- A6.1.2. – Stochastic Modeling
- A6.2.3. – Probabilistic methods
- A6.2.6. – Optimization
- A6.2.7. – HPC for machine learning
- A6.3.3. – Data processing
- A7.1.1. – Distributed algorithms
- A7.1.2. – Parallel algorithms
- A7.1.3. – Graph algorithms
- A8.1. – Discrete mathematics, combinatorics
- A8.2. – Optimization
- A8.7. – Graph theory
- A8.9. – Performance evaluation
- A9.2. – Machine learning

### Other research topics and application domains

- B6.3.2. – Network protocols
- B6.3.3. – Network Management
- B9.5.1. – Computer science
- B9.8. – Reproducibility

## Contents

<b>Project-Team TADAAM</b>	<b>1</b>
<b>1 Team members, visitors, external collaborators</b>	<b>5</b>
<b>2 Overall objectives</b>	<b>6</b>
<b>3 Research program</b>	<b>6</b>
3.1 Need for System-Scale Optimization . . . . .	6
3.2 Scientific Challenges and Research Issues . . . . .	7
<b>4 Application domains</b>	<b>8</b>
4.1 Mesh-based applications . . . . .	8
<b>5 Social and environmental responsibility</b>	<b>9</b>
5.1 Footprint of research activities . . . . .	9
5.2 Impact of research results . . . . .	9
5.3 Influence of team members . . . . .	9
<b>6 Highlights of the year</b>	<b>9</b>
<b>7 Latest software developments, platforms, open data</b>	<b>9</b>
7.1 Latest software developments . . . . .	9
7.1.1 hwloc . . . . .	9
7.1.2 Hsplit . . . . .	10
7.1.3 TopoMatch . . . . .	11
7.1.4 NewMadeleine . . . . .	11
7.1.5 IOPS . . . . .	12
7.1.6 AGIOS . . . . .	12
7.1.7 SCOTCH . . . . .	12
7.1.8 Raisin . . . . .	13
7.1.9 CORHPEX . . . . .	14
7.1.10 CERE . . . . .	15
7.2 New platforms . . . . .	15
7.2.1 PlaFRIM . . . . .	15
7.2.2 Abaca . . . . .	15
7.3 Open data . . . . .	16
<b>8 New results</b>	<b>16</b>
8.1 Predicting and Fixing Errors in Parallel Applications with AI . . . . .	16
8.2 Optimizing Performance and Energy with AI Guided Exploration . . . . .	17
8.3 IOPS: I/O Performance Evaluation Suite . . . . .	17
8.4 Prediction of HPC I/O Resources Usage with Machine Learning . . . . .	17
8.5 A Deep Look Into the Temporal I/O Behavior of HPC Applications . . . . .	18
8.6 On the Impact of Interference from Concurrent Jobs on Checkpointing Performance . . . . .	18
8.7 Checkpointing Optimisation to Prepare Future Exascale Plasma Turbulence Simulations . . . . .	18
8.8 A Weighted Bi-objective Strategy for Executing Scientific Workflows in Containerized Environments . . . . .	19
8.9 Towards a Novel Vertical Scaling Approach for Bursty Workloads in Kubernetes . . . . .	19
8.10 Network Topology Reconstruction . . . . .	20
8.11 A novel interface to enforce mapping policies . . . . .	20
8.12 Performance Projection for Design-Space Exploration on future HPC Architectures . . . . .	21
8.13 User-space interrupts for HPC communications . . . . .	21
8.14 Interrupt-safe data structures . . . . .	21
8.15 Management of InfiniBand memory registration with StarPU/NewMadeleine . . . . .	22

8.16	Composability of drivers and strategies in NewMadeleine	22
8.17	Improvement of the usability of SCOTCH and PT-SCOTCH	22
<b>9</b>	<b>Bilateral contracts and grants with industry</b>	<b>23</b>
9.1	Bilateral contracts with industry	23
<b>10</b>	<b>Partnerships and cooperations</b>	<b>24</b>
10.1	International initiatives	24
10.1.1	Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program	24
10.2	International research visitors	25
10.2.1	Visits of international scientists	25
10.3	European initiatives	25
10.3.1	H2020 projects	25
10.4	National initiatives	25
10.5	Public policy support	27
<b>11</b>	<b>Dissemination</b>	<b>28</b>
11.1	Promoting scientific activities	28
11.1.1	Scientific events: organisation	28
11.1.2	Scientific events: selection	28
11.1.3	Journal	28
11.1.4	Invited talks	28
11.1.5	Scientific expertise	29
11.1.6	Research administration	29
11.1.7	Standardization Activities	29
11.2	Teaching - Supervision - Juries	29
11.2.1	Teaching	29
11.2.2	Supervision	30
11.2.3	Juries	30
11.3	Popularization	30
11.3.1	Participation in Live events	30
<b>12</b>	<b>Scientific production</b>	<b>31</b>
12.1	Major publications	31
12.2	Publications of the year	31
12.3	Cited publications	33

# 1 Team members, visitors, external collaborators

## Research Scientists

- Brice Goglin [Team leader, INRIA, Senior Researcher, HDR]
- Alexandre Denis [INRIA, Researcher]
- Luan Teylo Gouveia Lima [INRIA, ISFP]
- Mihail Popov [INRIA, ISFP]

## Faculty Members

- Guillaume Mercier [BORDEAUX INP, Associate Professor Delegation, HDR]
- François Pellegrini [UNIV BORDEAUX, Professor, HDR]
- Francieli Zanon-Boito [UNIV BORDEAUX, Associate Professor Delegation, HDR]

## PhD Students

- Charles Goedefroit [BULL, CIFRE]
- Serge Meurrens [INRIA, from Dec 2025]
- Thibaut Pepin [CEA/DAM]
- Tristan Riehs [INRIA, from Sep 2025]
- Meline Trochon [DDN (DataDirect Networks), CIFRE]

## Technical Staff

- Mahamat Younous Abdraman [INRIA, Engineer]
- Pierre Clouzet [INRIA, Engineer]
- Ana Hourcau [INRIA, Engineer, from Oct 2025]
- Xavier Muller [INRIA, Engineer]

## Interns and Apprentices

- Laora Aimi [INRIA, Intern, from Feb 2025 until Sep 2025]
- Tanguy Chatelain [INRIA, Intern, from Feb 2025 until Jul 2025]
- Axel Malmgren [INRIA, until Jul 2025]
- Noureddine Tamssaout [INRIA, Intern, from Feb 2025 until Aug 2025]
- Gael Valade [INRIA, Intern, from Jun 2025 until Aug 2025]

## Administrative Assistant

- Fabienne Cuyollaa [INRIA]

## External Collaborators

- Iheb Becher [CNRS, from Apr 2025]
- Julien Rodriguez [DGA, until Aug 2025]

## 2 Overall objectives

In TADAAM, we propose a new approach where we allow the application to explicitly express its resource needs about its execution. The application needs to express its behavior, but in a different way from the compute-centric approach, as the additional information is not necessarily focused on computation and on instructions execution, but follows a high-level semantics (needs of large memory for some processes, start of a communication phase, need to refine the granularity, beginning of a storage access phase, description of data affinity, etc.). These needs will be expressed to a service through an API. These services will be system-wide (able to gather a global knowledge) and stateful (able to take decision based on the current request but also on previous ones). The API shall enable the application to access these services through a well-defined set of functions, based on carefully designed abstractions.

Hence, **the goal of TADAAM is to design stateful services for HPC systems, in order to optimize applications execution according to their needs.**

These layers will abstract low-level details of the architecture and the software stack, and will allow applications to register their needs. Then, according to these requests and to the environment characteristics, these services will feature engines to optimize the execution of the applications at system-scale, taking into account the gathered global knowledge and previous requests.

This approach exhibits several key characteristics:

- It is independent from the application parallelization, the programming model, the numerical scheme and, largely, from the data layout. Indeed, high-level semantic requests can easily be added to the application code after the problem has been modeled, parallelized, and most of the time after the data layout has been designed and optimized. Therefore, this approach is – to a large extent – orthogonal to other optimization mechanisms and does not require application developers to rewrite their code.
- Application developers are the persons who know best their code and therefore the needs of their application. They can easily (if the interface is well designed and the abstractions are correctly exposed), express the application needs in terms of resource usage and interaction with the whole environment.
- Being stateful and shared by all the applications in the parallel environment, the proposed layer will therefore enable optimizations that:
  - cannot be performed statically but require information only known at launch- or run-time,
  - are incremental and require minimal changes to the application execution scheme,
  - deal with several parts of the environment at the same time (e.g., batch scheduler, I/O, process manager and storage),
  - take into account the needs of several applications at the same time and deal with their interaction. This will be useful, for instance, to handle network contention, storage access or any other shared resources.

## 3 Research program

### 3.1 Need for System-Scale Optimization

Firstly, in order for applications to make the best possible use of the available resources, it is impossible to expose all the low-level details of the hardware to the program, as it would make impossible to achieve portability. Hence, the standard approach is to add intermediate layers (programming models, libraries, compilers, runtime systems, etc.) to the software stack so as to bridge the gap between the application and the hardware. With this approach, optimizing the application requires to express its parallelism (within the imposed programming model), organize the code, schedule and load-balance the computations, etc. In other words, in this approach, the way the code is written and the way it is executed and interpreted by the lower layers drives the optimization. In any case, this approach is centered on how computations are performed. Such an approach is therefore no longer sufficient, as the way an application is executing does depend less and less on the organization of computation and more and more on the way its data is managed.

Secondly, modern large-scale parallel platforms comprise tens to hundreds of thousand nodes<sup>1</sup>. However, very few applications use the whole machine. In general, an application runs only on a subset of the nodes<sup>2</sup>. Therefore, most of the time, an application shares the network, the storage and other resources with other applications running concurrently during its execution. Depending on the allocated resources, it is not uncommon that the execution of one application interferes with the execution of a neighboring one.

Lastly, even if an application is running alone, each element of the software stack often performs its own optimization independently. For instance, when considering an hybrid MPI/OpenMP application, one may realize that threads are concurrently used within the OpenMP runtime system, within the MPI library for communication progression, and possibly within the computation library (BLAS) and even within the application itself (pthreads). However, none of these different classes of threads are aware of the existence of the others. Consequently, the way they are executed, scheduled, prioritized does not depend on their relative roles, their locations in the software stack nor on the state of the application.

The above remarks show that in order to go beyond the state-of-the-art, it is necessary to design a new set of mechanisms allowing cross-layer and system-wide optimizations so as to optimize the way data is allocated, accessed and transferred by the application.

### 3.2 Scientific Challenges and Research Issues

In TADAAM, we will tackle the problem of efficiently executing an application, at system-scale, on an HPC machine. We assume that the application is already optimized (efficient data layout, use of effective libraries, usage of state-of-the-art compilation techniques, etc.). Nevertheless, even a statically optimized application will not be able to be executed at scale without considering the following dynamic constraints: machine topology, allocated resources, data movement and contention, other running applications, access to storage, etc. Thanks to the proposed layer, we will provide a simple and efficient way for already existing applications, as well as new ones, to express their needs in terms of resource usage, locality and topology, using a high-level semantic.

It is important to note that we target the optimization of each application independently but also several applications at the same time and at system-scale, taking into account their resource requirement, their network usage or their storage access. Furthermore, dealing with code-coupling application is an intermediate use-case that will also be considered.

Several issues have to be considered. The first one consists in providing relevant **abstractions and models to describe the topology** of the available resources **and the application behavior**.

Therefore, the first question we want to answer is: **“How to build scalable models and efficient abstractions enabling to understand the impact of data movement, topology and locality on performance?”** These models must be sufficiently precise to grasp the reality, tractable enough to enable efficient solutions and algorithms, and simple enough to remain usable by non-hardware experts. We will work on (1) better describing the memory hierarchy, considering new memory technologies; (2) providing an integrated view of the nodes, the network and the storage; (3) exhibiting qualitative knowledge; (4) providing ways to express the multi-scale properties of the machine. Concerning abstractions, we will work on providing general concepts to be integrated at the application or programming model layers. The goal is to offer means, for the application, to express its high-level requirements in terms of data access, locality and communication, by providing abstractions on the notion of hierarchy, mesh, affinity, traffic metrics, etc.

In addition to the abstractions and the aforementioned models we need to **define a clean and expressive API in a scalable way**, in order for applications to express their needs (memory usage, affinity, network, storage access, model refinement, etc.).

Therefore, the second question we need to answer is: **“how to build a system-scale, stateful, shared layer that can gather applications needs expressed with a high-level semantic?”** This work will require not only to define a clean API where applications will express their needs, but also to define how such a layer will be shared across applications and will scale on future systems. The API will provide a simple yet effective way to express different needs such as: memory usage of a given portion of the code; start of a compute intensive part; phase where the network is accessed intensively; topology-aware affinity management; usage

<sup>1</sup>More than 22,500 XE6 compute node for the BlueWaters system; 5040 B510 Bullx Nodes for the Curie machine; more than 49,000 BGQ nodes for the MIRA machine.

<sup>2</sup>In 2014, the median case was 2048 nodes for the BlueWaters system and, for the first year of the Curie machine, the median case was 256 nodes

of storage (in read and/or write mode); change of the data layout after mesh refinement, etc. From an engineering point of view, the layer will have a hierarchical design matching the hardware hierarchy, so as to achieve scalability.

Once this has been done, the service layer, will have all the information about the environment characteristics and application requirements. We therefore need to design a set of **mechanisms to optimize applications execution**: communication, mapping, thread scheduling, data partitioning / mapping / movement, etc.

Hence, the last scientific question we will address is: “**How to design fast and efficient algorithms, mechanisms and tools to enable execution of applications at system-scale, in full a HPC ecosystem, taking into account topology and locality?**” A first set of research is related to thread and process placement according to the topology and the affinity. Another large field of study is related to data placement, allocation and partitioning: optimizing the way data is accessed and processed especially for mesh-based applications. The issues of transferring data across the network will also be tackled, thanks to the global knowledge we have on the application behavior and the data layout. Concerning the interaction with other applications, several directions will be tackled. Among these directions we will deal with matching process placement with resource allocation given by the batch scheduler or with the storage management: switching from a best-effort application centric strategy to global optimization scheme.

## 4 Application domains

TADAAM focuses on data management hence on data intensive applications, ranging from HPC to AI applications, that require lots of data movement, between cores, storage, etc.

Mesh-based applications were the main focus when TADAAM was created but it is now only one of our focuses among data intensive applications, especially since the emergence of AI and data analytics.

### 4.1 Mesh-based applications

TADAAM targets scientific simulation applications on large-scale systems, as these applications present huge challenges in terms of performance, locality, scalability, parallelism and data management. Many of these HPC applications use meshes as the basic model for their computation. For instance, PDE-based simulations using finite differences, finite volumes, or finite elements methods operate on meshes that describe the geometry and the physical properties of the simulated objects.

Mesh-based applications not only represent the majority of HPC applications running on existing supercomputing systems, yet also feature properties that should be taken into account to achieve scalability and performance on future large-scale systems. These properties are the following:

**Size** Datasets are large: some meshes comprise hundreds of millions of elements, or even billions.

**Dynamicity** In many simulations, meshes are refined or coarsened at each time step, so as to account for the evolution of the physical simulation (moving parts, shockwaves, structural changes in the model resulting from collisions between mesh parts, etc.).

**Structure** Many meshes are unstructured, and require advanced data structures so as to manage irregularity in data storage.

**Topology** Due to their rooting in the physical world, meshes exhibit interesting topological properties (low dimensionality embedding, small maximum degree, large diameter, etc.). It is very important to take advantage of these properties when laying out mesh data on systems where communication locality matters.

All these features make mesh-based applications a very interesting and challenging use-case for the research we want to carry out in this project. Moreover, we believe that our proposed approach and solutions will contribute to enhance these applications and allow them to achieve the best possible usage of the available resources of future high-end systems.

## 5 Social and environmental responsibility

### 5.1 Footprint of research activities

Team members make common use of small to large-scale high performance computing platforms, which are energy consuming.

For this reason, previous research in the team [35] leveraged an existing consolidated simulation tool — SimGrid — for the bulk of experiments, using an experimental platform for validation only. For comparison, the validation experiments required  $\approx 88$  hours on nine nodes, while the simulation results that made into the paper would take at least 569 days to run. Although using and adapting the simulation tool took a certain effort, it allowed for more extensive evaluation, in addition to decreasing the footprint of this research. A similar strategy is being used in other projects since then ([34]).

Brice Goglin is involved in an emerging French initiative towards extending the lifetime of computing infrastructure [26].

### 5.2 Impact of research results

The digital sector is an ever-growing consumer of energy. Hence, it is of the utmost importance to increase the efficiency of use of digital tools. Our work on performance optimization, whether for high-end, energy consuming supercomputers, or more modest systems, aims at reducing the footprint of computations.

Because the aim of these machines is to be used at their maximum capacity, given their high production cost to amortize, we consider that our research results will not lead to a decrease in the overall use of computer systems; however, we expect them to lead to better modeling the energy consumption of application and hence a usage of their energy, hence resulting in “more science per watt”. Of course it is always hard to evaluate the real impact as a possible rebound effect is for more users to run on these machines, or users deciding to run extra experiments “because it is possible”.

### 5.3 Influence of team members

Members of the team participated to the writing of the *Inria global Action plan on F/M professional equality for 2021-2024*.

Moreover, Méline TROCHON, Ph.D. student in the team, is a member of the *Groupe de Travail Parité-Egalité* from the Inria Center at the University of Bordeaux ([project.inria.fr/pariteegalitebordeaux/](http://project.inria.fr/pariteegalitebordeaux/)).

## 6 Highlights of the year

- TADAAM members published 4 papers at the IPDPS 2025 conference in Milan, one of the major conference in our research community (as well as 2 papers in IPDPS workshops).
- TADAAM also received the IPDPS 2025 Open Source Contribution Award in collaboration with STORM.
- Revision 5.0 of the MPI standard for communication in parallel applications was published. Guillaume Mercier is an editor, chapter leader and working group leader.
- A first industrial partner joined the Scotch consortium.
- Francieli Boito defended her HDR habilitation.

## 7 Latest software developments, platforms, open data

### 7.1 Latest software developments

#### 7.1.1 hwloc

**Name:** Hardware Locality

**Keywords:** NUMA, Multicore, GPU, Affinities, Open MPI, Topology, HPC, Locality

**Functional Description:** Hardware Locality (hwloc) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices. It builds a widely-portable abstraction of these resources and exposes it to applications so as to help them adapt their behavior to the hardware characteristics. They may consult the hierarchy of resources, their attributes, and bind task or memory on them.

hwloc targets many types of high-performance computing applications, from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers, and multiple other parallel libraries already use hwloc.

**News of the Year:** In 2024, the support for heterogeneous memory was further improved to ease the selection of best memory targets in a more portable way. Support for GPUs from several vendors was also enhanced. Newest discovery and binding features in different operating systems were also leveraged in hwloc. Many internal changes were implemented to prepare the 3.0 major release in 2025.

**URL:** <http://www.open-mpi.org/projects/hwloc/>

**Publications:** [inria-00429889](#), [hal-00985096](#), [hal-01183083](#), [hal-01330194](#), [hal-01400264](#), [hal-01402755](#), [hal-01644087](#), [hal-02266285](#)

**Contact:** Brice Goglin

**Participants:** Samuel Thibault, Brice Goglin, an anonymous participant

**Partners:** Open MPI consortium, Intel, AMD, IBM, Eviden

### 7.1.2 Hsplit

**Name:** Hardware communicators split

**Keyword:** Topology

**Scientific Description:** Hsplit is a library that implements an abstraction allowing the programmer using MPI in their parallel applications to access the underlying hardware structure through a hierarchy of communicators. Hsplit is based on the `MPI_Comm_split_type` routine and provides a new value for the `split_type` argument that specifically creates a hierarchy a subcommunicators where each new subcommunicator corresponds to a meaningful hardware level. The important point is that only the structure of the hardware is exploited and the number of levels or the levels names are not fixed so as to propose a solution independent from future hardware evolutions (such as new levels for instance). Another flavor of this `MPI_Comm_split_type` function is provided that creates a roots communicators at the same time a subcommunicator is produced, in order to ease the collective communication and/or synchronization among subcommunicators.

**Functional Description:** Hsplit implements an abstraction that allows the programmer using MPI in their parallel applications to access the underlying hardware structure through a hierarchy of communicators. Hsplit is based on the `MPI_Comm_split_type` routine and provides a new value for the `split_type` argument that specifically creates a hierarchy a subcommunicators where each new subcommunicator corresponds to a meaningful hardware level. The important point is that only the structure of the hardware is exploited and the number of levels or the levels names are not fixed so as to propose a solution independent from future hardware evolutions (such as new levels for instance). Another flavor of this `MPI_Comm_split_type` function is provided that creates a roots communicators at the same time a subcommunicator is produced, in order to ease the collective communication and/or synchronization among subcommunicators.

**URL:** <https://gitlab.inria.fr/hsplit/hsplit>

**Publications:** [hal-01937123v2](#), [hal-01621941](#), [hal-01538002](#)

**Contact:** Guillaume Mercier

**Participants:** Guillaume Mercier, Brice Goglin, Emmanuel Jeannot, Thibaut Pepin

### 7.1.3 TopoMatch

**Keyword:** High performance computing

**Scientific Description:** TopoMatch embeds a set of algorithms to map processors/cores in order to minimize the communication cost of the application.

Important features are : the number of processors can be greater than the number of applications processes , it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) , it implements different placement algorithms that are switched according to the input size.

Some core algorithms are parallel to speed-up the execution. Optionally embeds scotch for fix-vertex mapping. enable exhaustive search if required. Several metric mapping are computed. Allow for oversubscribing of ressources. multithreaded.

TopoMatch is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI and MPICH2.

**Functional Description:** TreeMatch is a library for performing process placement based on the topology of the machine and the communication pattern of the application.

**URL:** <https://gitlab.inria.fr/ejeannot/topomatch>

**Publication:** hal-03780662

**Contact:** Emmanuel Jeannot

**Participants:** Emmanuel Jeannot, François Tessier, Guillaume Mercier, 2 anonymous participants

### 7.1.4 NewMadeleine

**Name:** NewMadeleine: An Optimizing Communication Library for High-Performance Networks

**Keywords:** High-performance calculation, MPI communication

**Functional Description:** NewMadeleine is the fourth incarnation of the Madeleine communication library. The new architecture aims at enabling the use of a much wider range of communication flow optimization techniques. Its design is entirely modular: drivers and optimization strategies are dynamically loadable software components, allowing experimentations with multiple approaches or on multiple issues with regard to processing communication flows.

The optimizing scheduler SchedOpt targets applications with irregular, multi-flow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance. SchedOpt itself is easily extensible through the concepts of optimization strategies (what to optimize for, what the optimization goal is) expressed in terms of tactics (how to optimize to reach the optimization goal). Tactics themselves are made of basic communication flows operations such as packet merging or reordering.

The communication library is fully multi-threaded through its close integration with PIOMan. It manages concurrent communication operations from multiple libraries and from multiple threads. Its MPI implementation MadMPI fully supports the MPI\_THREAD\_MULTIPLE multi-threading level.

**URL:** <https://pm2.gitlabpages.inria.fr/newmadeleine/>

**Publications:** inria-00127356, inria-00177230, inria-00177167, inria-00327177, inria-00224999, inria-00327158, tel-00469488, hal-02103700, inria-00381670, inria-00408521, hal-00793176, inria-00586015, inria-00605735, hal-00716478, hal-01064652, hal-01087775, hal-01395299, hal-01587584, hal-02103700, hal-02407276, hal-03012097, hal-03118807

**Contact:** Alexandre Denis

**Participants:** Alexandre Denis, 6 anonymous participants

### 7.1.5 IOPS

**Name:** IOPS - A generic benchmark orchestration framework

**Keywords:** I/O, HPC, Benchmarking

**Functional Description:** IOPS is a benchmark orchestration framework for automated parametric experiments. Users define YAML configurations specifying parameters to sweep, commands to execute, and metrics to extract. IOPS generates execution plans, runs tests locally or on SLURM clusters, and produces interactive HTML reports. It supports exhaustive, random, and Bayesian search strategies, with execution caching to avoid redundant tests. While originally designed for I/O characterization, it now serves as a general-purpose tool for systematic exploration of configuration spaces.

**Contact:** Luan Teylo Gouveia Lima

**Participants:** Luan Teylo Gouveia Lima, Francieli Zanon-Boito, Mahamat Younous Abdraman

### 7.1.6 AGIOS

**Name:** Application-guided I/O Scheduler

**Keywords:** High-Performance Computing, Scheduling

**Functional Description:** A user-level I/O request scheduling library that works at file level. Any service that handles requests to files (parallel file system clients and/or data servers, I/O forwarding frameworks, etc) may use the library to schedule these requests. AGIOS provides multiple scheduling algorithms, including dynamic options that change algorithms during the execution. It is also capable of providing many statistics in general and per file, such as average offset distance and time between requests. Finally, it may be used to create text-format traces.

**URL:** <https://github.com/francielizanon/agios>

**Publications:** [hal-03758890](#), [hal-01994677](#), [hal-02079899](#), [hal-01247942](#)

**Contact:** Francieli Zanon-Boito

**Participants:** Luan Teylo Gouveia Lima, 2 anonymous participants

### 7.1.7 SCOTCH

**Name:** Scotch / PT-Scotch

**Keywords:** Mesh partitioning, Domain decomposition, Graph algorithmics, High-performance calculation, Sparse matrix ordering, Static mapping

**Scientific Description:** The aim of the Scotch project is to tackle the problems of partitioning and mapping very large graphs, by way of algorithms that rely only on graph topology, and to devise efficient shared-memory, distributed-memory, and hybrid parallel algorithms for this purpose.

**Functional Description:** Scotch is a graph partitioner. It helps optimise the division of a problem, by means of a graph, into a set of independent sub-problems of equivalent sizes. These sub-problems can also be solved in parallel.

**Release Contributions:** SCOTCH has many interesting features:

- Its capabilities can be used through a set of stand-alone programs as well as through the libSCOTCH library, which offers both C and Fortran interfaces.
- It provides algorithms to partition graph structures, as well as mesh structures defined as node-element bipartite graphs and which can also represent hypergraphs.

- The SCOTCH library dynamically takes advantage of POSIX threads to speed-up its computations. The PT-SCOTCH library, used to manage very large graphs distributed across the nodes of a parallel computer, uses the MPI interface as well as POSIX threads.
- It can map any weighted source graph onto any weighted target graph. The source and target graphs may have any topology, and their vertices and edges may be weighted. Moreover, both source and target graphs may be disconnected. This feature allows for the mapping of programs onto disconnected subparts of a parallel architecture made up of heterogeneous processors and communication links.
- It computes amalgamated block orderings of sparse matrices, for efficient solving using BLAS routines.
- Its running time is linear in the number of edges of the source graph, and logarithmic in the number of vertices of the target graph for mapping computations.
- It can handle indifferently graph and mesh data structures created within C or Fortran programs, with array indices starting from 0 or 1.
- It offers extended support for adaptive graphs and meshes through the handling of disjoint edge arrays.
- It is dynamically parametrizable thanks to strategy strings that are interpreted at run-time.
- It uses system memory efficiently, to process large graphs and meshes without incurring out-of-memory faults,
- It is highly modular and documented. Since it has been released under the CeCILL-C free/libre software license, it can be used as a testbed for the easy and quick development and testing of new partitioning and ordering methods.
- It can be easily interfaced to other programs..
- It provides many tools to build, check, and display graphs, meshes and matrix patterns.
- It is written in C and uses the POSIX interface, which makes it highly portable.

**News of the Year:** The Member's contract for the Scotch Consortium has been finalized. A full-time core software engineer has been hired.

**URL:** <http://www.labri.fr/~pelegrin/scotch/>

**Publications:** [hal-04404141](#), [hal-01671156](#), [hal-01968358](#), [hal-00648735](#), [tel-00540581](#), [hal-00301427](#), [hal-00402893](#), [tel-00410402](#), [hal-00402946](#), [hal-00410408](#), [hal-00410427](#)

**Contact:** François Pellegrini

**Participants:** François Pellegrini, Marc Fuentes, Clément Barthelemy, Xavier Muller, 6 anonymous participants

**Partners:** Université de Bordeaux, IPB, CNRS, Region Aquitaine

### 7.1.8 Raisin

**Keywords:** Hypergraph, Partitioning, Graph algorithmics, Static mapping, FPGA

**Functional Description:** Raisin is a multi-valued oriented hypergraph partitioning software whose objective function is to minimize the length of the longest path between some types of vertices while limiting the number of cut hyper-arcs.

**Release Contributions:** Raisin has been designed to solve the problem of circuit placement onto multi-FPGA architectures. It models the circuit to map as a set of red-black, directed, acyclic hypergraphs (DAHs). Hypergraph vertices can be either red vertices (which represent registers and external I/O ports) or black vertices (which represent internal combinatorial circuits). Vertices bear multiple weights, which define the types of resources needed to map the circuit (e.g., registers, ALUs, etc.). Every hyper-arc comprises a unique source vertex, all other ends of the hyper-arcs being sinks (which models the transmission of signals through circuit wiring). A circuit is consequently represented as set of DAHs that share some of their red vertices.

Target architectures are described by their number of target parts, the maximum resource capacities within each target part, and the connectivity between target parts.

The main metric to minimize is the length of the longest path between two red vertices, that is, the critical path that signals have to traverse during a circuit compute cycle, which correlates to the maximum frequency at which the circuit can operate on the given target architecture.

Raisin computes a partition in which resource capacity constraints are respected and the critical path length is kept as small as possible, while reducing the number of cut hyper-arcs. It produces an assignment list, which describes, for each vertex of the hypergraphs, the part to which the vertex is assigned.

Raisin has many interesting features:

- It can map any weighted source circuit (represented as a set of red-black DAHs) onto any weighted target graph.
- It is based on a set of graph algorithms, including a multi-level scheme and local optimization methods of the “Fiduccia-Mattheyses” kind.
- It contains two greedy initial partitioning algorithms that have a computation time that is linear in the number of vertices. Each algorithm can be used for a particular type of topology, which can make them both complementary and efficient, depending on the problem instances.
- It takes advantage of the properties of DAHs to model path lengths with a weighting scheme based on the computation of local critical paths. This weighting scheme allows to constrain the clustering algorithms to achieve better results in smaller time.
- It can combine several of its algorithms to create dedicated mapping strategies, suited to specific types of circuits.
- It provides many tools to build, check and convert red-black DAHs to other hypergraph and graph formats.
- It is written in C.

**Publications:** [hal-03596218](#), [hal-04008677](#), [hal-04379716](#), [hal-03604540v1](#)

**Contact:** Julien Rodriguez

**Participants:** François Pellegrini, Julien Rodriguez, 2 anonymous participants

### 7.1.9 CORHPEX

**Name:** COmpiler, Runtime and Hardware Parameter EXplorer

**Keywords:** Energy, Performance, Optimization, Design space exploration

**Scientific Description:** COmpiler, Runtime and Hardware Parameter EXplorer (CORHPEX), is a framework to explore performance optimization spaces for HPC applications.

**Functional Description:** CORHPEX enables application developers to discover the influence of configurations of hardware, compilers and run-time options on optimization targets such as performance and energy consumption by proposing the efficient collection and exploration of metrics in application design spaces.

**Release Contributions:** First public diffusion

**URL:** <https://gitlab.inria.fr/corhpex/corhpex>

**Publications:** [hal-05311328](#), [hal-05224515](#), [hal-04969854](#)

**Contact:** Mihail Popov

**Participants:** Mihail Popov, 4 anonymous participants

**Partner:** IFPEN

### 7.1.10 CERE

**Name:** Codelet Extractor and REplayer

**Keywords:** Checkpointing, Profiling

**Functional Description:** CERE finds and extracts the hotspots of an application as isolated fragments of code, called codelets. Codelets can be modified, compiled, run, and measured independently from the original application. Code isolation reduces benchmarking cost and allows piecewise optimization of an application.

**URL:** <https://benchmark-subsetting.github.io/cere/>

**Contact:** Mihail Popov

**Participant:** Mihail Popov

**Partners:** Université de Versailles St-Quentin-en-Yvelines, Exascale Computing Research

## 7.2 New platforms

### 7.2.1 PlaFRIM

**Participants:** Brice Goglin.

**Name:** Plateforme Fédérative pour la Recherche en Informatique et Mathématiques

**Website:** [plafrim.fr](http://plafrim.fr)

**Description:** PlaFRIM is an experimental platform for research in modeling, simulations and high performance computing. This platform has been set up from 2009 under the leadership of Inria Bordeaux Sud-Ouest in collaboration with computer science and mathematics laboratories, respectively LaBRI and IMB with a strong support in the region Aquitaine.

It aggregates different kinds of computational resources for research and development purposes. The latest technologies in terms of processors, memories and architecture are added when they are available on the market. As of 2023, it contains more than 6,000 cores, 50 GPUs and several large memory nodes that are available for all research teams of Inria Bordeaux, Labri and IMB.

Brice Goglin is in charge of PlaFRIM since June 2021.

### 7.2.2 Abaca

**Participants:** Brice Goglin.

**Name:** Abaca

**Website:** [abaca.inria.fr](http://abaca.inria.fr)

**Description:** Abaca is Inria's mutualized computing infrastructure. It gathers computing resources from Inria research centers across the country into a uniform software environment. The platform currently contains more than 15 000 CPU cores and 600 000 GPU cores.

Brice Goglin is a member of the executive committee since 2021,- and the Product Owner since June 2025.

## 7.3 Open data

### LLM4DiCE dataset

**Contributors:** Asia Auville, Tim Jammer, Eric Petit, Pablo de Oliveira Castro, Emmanuelle Saillard, and Mihail Popov

**Description:** Dataset containing both real world programs and mutants. Mutants are created from the original files by inserting MPI errors using a mutation tool developed in collaboration with Technische Universität Darmstadt. The files are scrapped from GitHub’s popular MPI projects.

**Dataset PID (DOI,...):** <https://doi.org/10.5281/zenodo.17286943>

**Project link:** <https://zenodo.org/records/17286943>

**Publications:** under review.

**Contact:** asia.auville@inria.fr

### I/O Traces from SDumont and PlaFRIM

**Contributors:** Francieli Boito, Luan Teylo, Mihail Popov, Theo Jolivel, François Tessier, Jakob Luetzgau, Julien Monniot, Ahmad Tarraf, André Carneiro, and Carla Osthoff.

**Description:** This data set contains the PlaFRIM and Santos Dumont traces we made available after the study from the “A Deep Look Into the Temporal I/O Behavior of HPC Applications” paper, including all code used to analyze them.

**Dataset PID (DOI,...):** <https://doi.org/10.5281/zenodo.14965920>

**Project link:** <https://zenodo.org/records/14965920>

**Publications:** [9, 27]

**Contact:** Luan Teylo, luan.teylo@inria.fr

## 8 New results

### 8.1 Predicting and Fixing Errors in Parallel Applications with AI

**Participants:** Asia Auville, Emmanuelle Saillard, Mihail Popov.

Investigating if parallel applications are correct is a very challenging task. Yet, recent progress in ML and text embedding show promising results in characterizing source code or the compiler intermediate representation to identify optimizations. We propose to transpose such characterization methods to the context of verification. In particular, we train ML models that take as labels the code correctness along with intermediate representations or source code embeddings as features. Results over small MPI verification benchmarks including MBI and DataRaceBench demonstrate that we can train models that detect if a code is correct with 90% accuracy and up to 75% over new unseen errors. This work [37] is a collaboration with the Iowa State University.

In the context of Asia Auville’s Ph.D. thesis, we are investigating the prediction capabilities of ML models to detected and fix errors in real world applications. Through Github repositories crawling and compiler mutations, we are creating complex application examples with MPI errors to train detection models. This work, in collaboration with the University of Versailles, Intel, and Technische Universität Darmstadt is currently under review.

## 8.2 Optimizing Performance and Energy with AI Guided Exploration

**Participants:** Lana Scravaglieri, Mihail Popov, Pierre Clouzet, Laércio Lima Pilla, Nouredine Tamssaout.

HPC systems expose configuration options that help users optimize their applications' execution. Questions related to the best thread and data mapping, number of threads, or cache prefetching have been posed for different applications, yet they have been mostly limited to a single optimization objective (e.g., performance) and a fixed application problem size. Unfortunately, optimization strategies that work well in one scenario may generalize poorly when applied in new contexts.

In the context of Lana Scravaglieri's Ph.D. thesis and in collaboration with IFP Energies nouvelles (IFPEN), we carried this research further, by focusing on the exploration of SIMD transformations over carbon storage applications. To do so, we are designing a more general exploration infrastructure, CORHPEX, that can easily incorporate more diverse optimization knobs and applications. This work was awarded the IPDPS'25 Best Open-Source Contribution Award. CORHPEX was also utilized on Software Defined Radio (SDR) applications in collaboration with Inria Topal, demonstrating the potential performance and energy gains when considering the target hardware [31].

Finally, in collaboration with the University of Uppsala, we are also investigating the hardware prefetching interaction with the new hybrid architectures (Intel's Efficiency- and Performance-cores or ARM big.LITTLE). Preliminary results [38, 36] showcase how energy gains can be achieved by tuning the system to the applications. This work is currently under review. we plan to further pursue the heterogeneous capabilities by adding memory effects to the AI based exploration.

## 8.3 IOPS: I/O Performance Evaluation Suite

**Participants:** Mahamat Abdraman, Francieli Boito, Mihail Popov, Luan Teylo.

In high-performance computing, I/O operations can become a bottleneck when dealing with large-scale data processing tasks. In these systems, where files are distributed by a parallel file system (PFS) across multiple object storage targets (OSTs), the performance of the I/O operations is influenced by several interdependent parameters, such as the number of computing nodes, network performance, the number of OSTs, and the access pattern of the application, among others. Therefore, understanding how the combination of these parameters affects I/O performance is crucial for identifying anomalies, verifying expected performance from a storage system, and improving application I/O performance. In this work [18], we present IOPS, a tool that allows users to profile an HPC I/O infrastructure. IOPS is designed to be easy to use, flexible, and extensible. We demonstrate the capabilities of IOPS by evaluating the performance of BeeGFS, a popular parallel file system, using different configurations and access patterns. The results show that IOPS can automate the process of finding the best parameter combinations.

In addition to the paper presenting IOPS, in 2025 we extended it to cover other benchmarking situations, such as generating data sets for training AI models (see Section 8.4) and evaluating interference. Moreover, we explored the use of Bayesian Optimization to navigate a large space of parameters.

## 8.4 Prediction of HPC I/O Resources Usage with Machine Learning

**Participants:** Mahamat Abdraman, Laora Aimi, Francieli Boito, Mihail Popov, Luan Teylo.

During previous work on heuristics for allocation of I/O resources to HPC applications [34], we observed that the best algorithm requires to know the number of resources that maximize application I/O performance. Nonetheless, this information is not typically available, and obtaining it would involve running the application multiple times with multiple configurations. Instead, we then focused on finding a good estimate of the

number of I/O resources (e.g., OSTs and I/O nodes) that provides the maximal bandwidth while minimizing the system occupation and taking into account the natural I/O variability. We used machine learning techniques to do so, focusing on intrinsic application features and system configurations. Preliminary results were published as a pre-print on HAL in 2024 [33].

In 2025, during the internship of Laora Aimi, we continued this work by improving the methodology for the evaluation of models' accuracy and further studying the importance of different parameters. Furthermore, we investigated approaches to decrease the amount of data required for training the model, and improved the IOPS tool so it can be used to obtain such data. This work is expected to result in a publication in 2026.

## 8.5 A Deep Look Into the Temporal I/O Behavior of HPC Applications

**Participants:** Francieli Boito, Mihail Popov, Luan Teylo.

We studied the temporal I/O behavior of over 440,000 jobs running on four HPC systems, all different in terms of infrastructure, scale, and users, covering several time periods over the last 11 years. The data we analyzed came either from parallel file systems (system-side traces) or from I/O monitoring tools (application-side traces).

The aim of analyzing these traces is to provide an in-depth study of data accesses by HPC applications in the wild. We have thus identified and addressed a number of questions dealing with the temporality of I/Os, their periodicity, the existence and prevalence of certain patterns, I/O concurrency between applications or user practices. We also proposed a classification of temporal I/O behaviors, which shows a few patterns are able to represent a vast majority of jobs. Overall, the results of this study provide relevant information for anyone working to improve high-performance I/O. They also serve as a basis for future research into both behavior detection tools and the use of trace analysis, particularly for scheduling and application optimization.

This work was submitted in 2024 and published at IPDPS'2025 [9] (an extended version was published as a technical report [27]) and is the result of a collaboration between Inria Bordeaux, Inria Rennes, the Technical University of Darmstadt, and the National Laboratory for Scientific Computing (LNCC).

We are currently extending our AI analysis on the dataset by considering more automated techniques. Our goal is to scale the I/O trace pattern detection to ease any new optimization.

## 8.6 On the Impact of Interference from Concurrent Jobs on Checkpointing Performance

**Participants:** Francieli Boito, Brice Goglin, Luan Teylo, Méline Trochon.

Among the most I/O-intensive operations in HPC systems is checkpointing, which is necessary to save the state of an application and allow it to be restarted at an advanced stage of computation. However, near the parallel file system, concurrency prevents checkpoint phases from reaching the best I/O performance. In this work, we studied I/O interference in this specific context: we look at performance of a checkpoint phase when faced with different interference patterns, exploring aspects such as scale, number of processes, operation, number of files, etc. Through an extensive experimentation, in two systems, we showed the impact of these aspects on checkpoint. Moreover, we showed that some configurations — namely, an application that does random accesses — lead to degraded system I/O performance. This work was published as pre-print in 2025 [28] and is expected to result in a publication in 2026, as it provides an important background for any effort into mitigating I/O interference.

## 8.7 Checkpointing Optimisation to Prepare Future Exascale Plasma Turbulence Simulations

**Participants:** Méline Trochon.

The advent of exascale computing has revolutionized high-performance computing (HPC) and enabled unprecedented advancements in nuclear fusion research. Simulations of plasma turbulence dynamics, such as the GYSELA code, now achieve unparalleled precision and complexity. However, this progress is accompanied by significant challenges in managing the exponential growth of data generated by these simulations. Traditional input/output (I/O) methods struggle to handle the massive data volumes, heightened concurrency, and fault-tolerance requirements inherent to exascale systems. This work [17] investigated the I/O bottlenecks inherent in exascale computing, with a particular focus on the checkpointing mechanisms of GYSELA. These mechanisms are critical for ensuring fault tolerance and must handle several terabytes of data efficiently to avoid undermining computational performance. We analyze the current implementation of GYSELA's checkpointing mechanism managed via the PDI data interface, identifying its limitations and proposing two alternative approaches aimed at enhancing scalability and resilience. Experiments conducted on pre-exascale architectures validate the efficiency of these methods through both strong and weak scaling benchmarks.

We reduced the checkpointing execution time by a factor of four, achieving near-optimal bandwidth utilisation, and we have identified implementations well-suited for exascale architectures. Our findings suggest the potential for notable performance improvements and offer insights that could help optimise I/O operations in exascale simulations.

This work was started during Méline Trochon's internship at CEA, and will be continued in 2026. Indeed, GYSELA is one of the demonstrator applications of the NumPEX Exa-DoST project.

## 8.8 A Weighted Bi-objective Strategy for Executing Scientific Workflows in Containerized Environments

**Participants:** Wesley Ferreira, Liliane Kunstmann, Yuri Frota, Luan Teylo, Daniel de Oliveira.

Scientific workflows support the execution of complex simulation-based experiments across heterogeneous computing environments. Containerization technologies, such as Docker, improve portability by encapsulating tasks together with their dependencies. However, they also introduce challenges in resource management, as containers incur additional memory and CPU overhead and may execute concurrently on the same virtual or physical machine. These challenges are particularly critical in memory-constrained environments, where inefficient scheduling can lead to performance degradation or even task failures. To address this issue, we propose a weighted bi-objective scheduling strategy that considers memory consumption and execution time, allowing users to prioritize one objective or achieve a balance between the two. Experimental evaluations with both synthetic and real-world workflows demonstrate that our approach enhances performance and resource utilization.

This study [11] was done in collaboration with the Federal Fluminense University in Brazil, through the Equipe Associée DecoHPC, and was published in the *Simpósio em Sistemas Computacionais de Alto Desempenho (SSCAD)*, 2025.

## 8.9 Towards a Novel Vertical Scaling Approach for Bursty Workloads in Kubernetes

**Participants:** Miguel De Lima, Luan Teylo, Lúcia Drummond.

Traditional static computational resource allocation in cloud or on-premises clusters often results in inefficient overprovisioning. Users frequently lack precise knowledge of the memory and processors their applications require, leading them to request excess resources. This causes wasted capacity, higher costs,

and, in shared environments, longer queue waiting times. Dynamic resource allocation through autoscaling addresses this issue by adjusting resources at runtime. Kubernetes, a widely used container orchestration platform, supports autoscaling via Horizontal and Vertical Pod Autoscalers. However, its default restart-based scaling can disrupt stateful, long-running workloads without checkpointing. This work leverages Kubernetes' new in-place scaling, which resizes resources without restarts, to propose the Dynamic Resizing Strategy (DRS), a novel autoscaling approach that proactively manages contention by temporarily throttling co-located pods to prioritize a bursting application. We evaluate it with NAS Parallel Benchmarks and synthetic workloads in co-execution scenarios, showing improved efficiency and stability, increasing success rates and reducing global average wait time by over 18% compared to the Burstable QoS class.

This study [14] was done in collaboration with the Federal Fluminense University in Brazil, through the Equipe Associée DecoHPC, and was published in the 18th IEEE/ACM International Conference on Utility and Cloud Computing, 2025.

## 8.10 Network Topology Reconstruction

**Participants:** Brice Goglin, Guillaume Mercier, Thibaut Pépin.

With the increase in size and complexity of supercomputers, it has become crucial to match applications and communication libraries to the underlying physical topology. While new functionalities were recently added to the MPI standard regarding the access of topological information of computing nodes of the system, there is still a lack of tools to retrieve the network topology information.

We previously implemented the prototype for a tool allowing the reconstruction of the network topology using latency measurements[15]. We continued to build upon this tool by improving our runtime MPI rank attribution library. The purpose of this library is to better define the execution environment of a succession of communications and to use this information to improve the performance of MPI collective communications. To this end, we defined a set of metrics to add upon the information commonly used by collective communication algorithms. Our first implementation validated the usefulness of the use of these metrics to improve the data locality as well as the communication load balancing. Experimental results using our tool show a great potential for performance improvements for a string of communications. The results analysis is still ongoing.

We also started to adapt node-aware topological collective communication algorithms to use the network topology information. This adaptation requires to shift from a balanced topology (the node), to an unbalanced one (the network) as well as adding the handling of multiple topology layers.

Last, on the communication side, we presented our work on runtime rank attribution at the COMPAS conference in Bordeaux in July 2025 and our article on network topology reconstruction was accepted for the HPCAsia conference, which will take place in January 2026

## 8.11 A novel interface to enforce mapping policies

**Participants:** Guillaume Mercier.

We are taking part in the design and development of the QUOVADIS software, which aims at providing application developers with an interface to easily enforce mapping policies for processes and threads. Indeed, in a context of multi-kernel applications where various policies could be applied to improve performance, changing dynamically such policies is currently difficult and awkward. QUOVADIS' goal is to help hybrid applications in making efficient use of heterogeneous hardware, to ease programmability in the presence of multiple programming abstractions, and to enable portability across systems. QUOVADIS' core interface is based on a split-like operation for processes that partitions hardware resources into an arbitrary number of pieces and assigns processes to these pieces, enabling concurrency and avoiding resource interference. Similar operations are also available for threads [30] (OpenMP and POSIX threads are both supported), enabling hybrid applications to fully take advantage of managing resources at a level of abstraction that

computational scientists can employ rapidly. The QUOVADIS thread interface features similar semantics — and syntax — to the process interface allowing users to leverage a single-semantics model for partitioning and assignment of resources to workers. When combined with application-specific heuristics, QUOVADIS enforces tailored execution policies by using dynamic hardware affinities exposed through a straightforward stack semantics (push/pop). An arbitrary number of binding policies can be stacked that correspond to the stacked composition of coupled components in a QUOVADIS-enabled application.

## 8.12 Performance Projection for Design-Space Exploration on future HPC Architectures

**Participants:** Clément Gavoille, Brice Goglin, Emmanuel Jeannot.

To address the growing need for performance from future HPC machines, their processor designs are constantly evolving. Assessing the impact of changes in hardware, software stack, and applications on performance is crucial in a codesign process. Here, we propose a performance projection workflow to facilitate the initial exploration of design space for multicore nodes and multi-threaded applications. For this purpose, we analyze the architectural efficiency of an accessible source machine and determine the maximum sustainable flop/s performance of a hypothetical target machine based on its software stack on a per-thread basis. Finally, we use these characterizations to project the performance evolution from the source machine to the target machine.

In this work, we assess the strengths and weaknesses of our approach by integrating it into the Fugaku-Next Feasibility Study. We compare the accuracy and overhead of our approach with the gem5 cycle-level simulations and a fast exploration methodology based on Machine Code Analyzer (MCA), using NAS Parallel benchmarks and CCS-QCD, a quantum chromodynamics miniapp. The study demonstrates that, compared to gem5, our approach has a prediction deviation of 5% for most cases and up to 30% for extreme cases. Additionally, it exhibits an execution overhead an order of magnitude bigger than MCA but orders of magnitude smaller than gem5.

Finally, we demonstrate our approach's capability to study larger scale and more representative applications than gem5, such as QWS and Genesis, two applications of RIKEN optimized for Fugaku.

This work [12] was performed in collaboration with CEA/DAM and RIKEN.

## 8.13 User-space interrupts for HPC communications

**Participants:** Alexandre Denis, Brice Goglin, Charles Goedefroit.

In HPC, network are programmed directly from user space, since system call have a significant cost with low latency networks. Usually, the user performs polling: the network is polled at regular interval to check whether a new message has arrived. However, it wastes some resources. Another solution is to rely on interrupts instead of polling, but since interrupts are managed by the kernel, they involve system calls we are precisely willing to avoid.

Intel introduced user-level interrupts on its latest Sapphire Rapids CPUs, allowing to use interrupts from user space. These user space interrupts may be a viable alternative to polling, by using interrupts without the cost of systems calls.

We have extended [20, 13] user-space interrupts to be able to trigger them from a device and not only from the CPU. We work with Eviden on their BXI network to make it trigger user-space interrupts so as to benefit from uintr in inter-node communications.

## 8.14 Interrupt-safe data structures

**Participants:** Alexandre Denis, Charles Goedefroit.

With the addition of interrupt-based communication in NewMadeleine, synchronization issues have emerged in some data structures. NewMadeleine relies on lock-free queues for a lot of its activities: progression through Pioman, submission queue, completion queue, deferred tasks. However, our implementation of lock-free queues was not non-blocking and was not suitable for use in an interrupt handler.

Other implementations found in the literature target scalability but exhibit high latency in the uncontended case. We have shown that, since latency of network and queues are different by several orders of magnitude, even highly contended network operation do not impose a high pressure on queues.

We have proposed [10] a new non-blocking queue algorithm that is optimized for low contention, while degrading nicely in case of higher contention. We have shown that it exhibits the best performance in NewMadeleine when compared to 15 other queue designs on four different architectures.

### 8.15 Management of InfiniBand memory registration with StarPU/NewMadeleine

**Participants:** Alexandre Denis, Tanguy Chatelain, Tristan Riehs.

Until now, StarPU allocated memory to receive messages when tasks were submitted. This may cause a useless large consumption of memory. A better strategy would be to allocate memory just in time, when the sender is ready to send data.

However, networks used in HPC, like InfiniBand, are programmed from user space, and thus require most of the time memory to be registered; this is a costly operation that involves a system call. If we allocate memory just in time, memory registration delays the actual posting of the receive operation.

We have proposed [29, 19] to perform memory registration in advance, to remove it from the communication critical path: on the sender side, when we begin computation for a task, we know that we will have to send it later; we register memory upfront at the beginning of computation, before the data is even available (we only need the pointer). At the same time, when a task starts, the sender sends a request to the receiver so that it may allocate and register the buffer for the receive operation, so that allocation and registration are completed when the message is ready to be sent. We have observed that these mechanisms save a lot of memory, thus allowing to run larger datasets, while improving performance.

### 8.16 Composability of drivers and strategies in NewMadeleine

**Participants:** Alexandre Denis, Gael Valade.

The NewMadeleine communication library is built with software component, making it modular. Components are used for drivers and for the optimizing strategy. However, strategies are built as a single monolithic component. Therefore, the user has to choose between the strategy that implement message aggregation, priority-based scheduling or multi-rail, but cannot *compose* them.

We have worked [32] on a new structure that dispatch these features in virtual driver components, so as to make them composable.

### 8.17 Improvement of the usability of SCOTCH and PT-SCOTCH

**Participants:** Clément Barthélemy, Mark Fuentes, Xavier Muller, François Pellegrini.

The SCOTCH software has undergone continuous development. A first axis of work concerned the implementation of a full set of multilevel vertex bipartitioning algorithms that aim at minimizing the vertex cut (*i.e.*, the separator size), while also balancing the weights of halo vertices across both subdomains. This work, as a continuation of Astrid Casadei's work, described in her Ph.D. thesis, aims at reducing communication overhead when solving sparse linear systems.

A second axis of work concerned the improvement of the edge partitioning/mapping of huge meshes representing cardiac tissue, in the context of the MICROCARD-2 Euro-HPC project. In this project, elements bearing the same tags must always be placed in the same parts. This result has been achieved by designing and implementing multi-threaded centralized and distributed graph quotienting routines, seen as extensions of preexisting graph coarsening algorithms, which coalesce all vertices of same tags into single weighted vertices of a coarser graph then to be partitioned. This method yielded the expected result, *i.e.*, well-balanced partitions enforcing the placement constraint.

All these improvements are already available to scientific partners in pre-release versions of Scotch, and will be provided to the global community in future public releases. Several bugfix releases took place in the mean time, up to version v7.0.10.

Also, version v1.0.0 of ScotchPy, a Python wrapper for Scotch and PT-Scotch, has been publicly released.

## 9 Bilateral contracts and grants with industry

### 9.1 Bilateral contracts with industry

#### CEA

**Participants:** Clément Gavaille, Brice Goglin, Guillaume Mercier, Thibaut Pépin.

- CEA/DAM granted the funding of the PhD thesis of Thibaut Pépin on communication on modular supercomputer architectures.
- CEA/DAM granted the fundind of the PhD thesis of Clément Gavaille, defended in January, which led to publication with RIKEN [12].

#### ATOS/Bull/Eviden

**Participants:** Quentin Buot, Alexandre Denis, Brice Goglin, Emmanuel Jeannot, Guillaume Mercier, Richard Sartori.

- ATOS/Bull/Eviden is funding the CIFRE PhD Thesis of Charles Goedefroit on Delivering Userspace Interrupts from the BXI network interface

#### IFPEN

**Participants:** Mihail Popov, Lana Scravaglieri.

- IFPEN funded the PhD Thesis of Lana Scravaglieri on the designs of models to optimize numerical simulations by adjusting the programs to the underline HPC systems.

#### DDN

**Participants:** Francieli Boito, Brice Goglin, Méline Trochon.

- DDN is funding the thesis of Méline Trochon (CIFRE) on improving checkpointing mechanisms in HPC to prevent network and I/O contention. She is advised by Francieli Boito and Brice Goglin in Bordeaux, François Tessier from Inria Rennes, and Jean-Thomas Acquaviva from DDN.

## 10 Partnerships and cooperations

### 10.1 International initiatives

#### 10.1.1 Associate Teams in the framework of an Inria International Lab or in the framework of an Inria International Program

##### DECoHPC

**Participants:** Luan Teylo, Francieli Boito.

**Title:** Data movement, Energy Consumption and performance in High-Performance Computing

**Partner Institution(s):** • National Laboratory for Scientific Computing (LNCC), Brazil

- Federal Fluminense University (UFF) Brazil
- Federal University of Rio Grande do Sul (UFRGS), Brazil
- Federal Center for Technological Education of Rio de Janeiro (CEFET-RJ), Brazil

**Date/Duration:** from 2024 to 2026

**Website:** [team.inria.fr/decohpc/](https://team.inria.fr/decohpc/)

**Additional info/keywords:** Supercomputers were conceived to efficiently run traditional HPC applications, namely numerical simulations. However, in the context of the convergence between HPC and big data, their workload is becoming more heterogeneous. In this new scenario, efficient application execution becomes more challenging. Moreover, energy consumption has emerged as an important concern for HPC and computer science in general. First, with the effects of climate change, environmental concerns have become a major focus across various scientific fields. Second, as more and more exascale machines emerge, the energy budget has become one of the main concerns for these machines, driven not only by environmental considerations but also by economic ones.

The previous HPCProSol associate team (2021–2023) provided us with performance insights about two kinds of representative applications from the Santos Dumont system from the LNCC (the largest supercomputer in Latin America): finite element methods (HPC) and bioinformatics workflows (HPDA). Moreover, we collaborated on advancing the system’s monitoring infrastructure by developing software to efficiently process it. Now, in the DECoHPC associate team, we aim to take these insights and tools and extend them towards our three main goals:

- (WP1) Based on the Santos Dumont’s traces (recently made available), to obtain a holistic view of the I/O behavior of HPC applications. We want to classify applications according to their behaviors — and on their different needs from the system.
- (WP2) To study and characterize the energy consumption of moving applications’ data through the network and I/O infrastructure.
- (WP3) To characterize the I/O performance and energy consumption of AI applications, which have not been explored in HPCProSol, but are now among one of the most important users of HPC facilities.

## 10.2 International research visitors

### 10.2.1 Visits of international scientists

In the context of the DECoHPC Associate Team, in 2025 the team received the visits of:

- André Carneiro, from LNCC, to continue our collaborative work on analyzing I/O traces (see Section 8.5).
- Miguel de Lima, from UFF, to work on containerization environments for HPC applications (see Section 8.9).

## 10.3 European initiatives

### 10.3.1 H2020 projects

#### EUPEX

**Participants:** Brice Goglin.

- EUPEX: European Pilot for Exascale
- Program: H2020 EuroHPC
- Grant Agreement number: 101033975 – H2020-JTI-EuroHPC-2020-01
- 2022-2026
- Partners: Atos, FZJ, CEA, GENCI, CINECA, E4, ICS-FORTH, Cini National Lab, ECMWF, IT4I, FER, ParTec, EXAPSYS, INGV, Goethe University, SECO, CybeleTech
- The EUPEX pilot brings together academic and commercial stakeholders to co-design a European modular Exascale-ready pilot system. Together, they will deploy a pilot hardware and software platform integrating the full spectrum of European technologies, and will demonstrate the readiness and scalability of these technologies, and particularly of the Modular Supercomputing Architecture (MSA), towards Exascale.  
EUPEX's ambition is to support actively the European industrial ecosystem around HPC, as well as to prepare applications and users to efficiently exploit future European exascale supercomputers.
- Website: [eupex.eu](http://eupex.eu)
- TADaaM funding: 150k€

## 10.4 National initiatives

### InriaSoft: Scotch Consortium

**Participants:** François Pellegrini, Clément Barthélemy.

- Scotch Consortium
- Program: InriaSoft
- 2024–
- Website: [gitlab.inria.fr/scotch/scotch](https://gitlab.inria.fr/scotch/scotch)
- Coordinator: François Pellegrini

- Abstract:

The Scotch Consortium, supported by InriaSoft<sup>3</sup>, has been created to bring together organizations interested in furthering the SCOTCH software currently developed within the TADAAM project. It will take care of the sustainability and development of the Scotch software environment, sharing the governance between its members. It will also allow every member to participate in the software roadmap, and to get adequate support. It will ensure SCOTCH stays permanently maintained, and available to the worldwide community under a free/libre software license.

While the consortium has not officially been launched, Inria has started populating the Scotch consortium engineering team by agreeing to hire a full-time core software engineer. Clément Barthélemy was recruited and started working on September 1<sup>st</sup>, joining Marc Fuentes, the part-time environment software engineer.

### Numpex PC2: Exa-Soft

**Participants:** Alexandre Denis.

- Exa-Soft: HPC softwares and tools
- Program: project PC2 in PEPR Numpex
- 2023-2029
- Partners: Université Paris-Saclay, Telecom SudParis, Bordeaux INP, ENSIIE, Université de Bordeaux, Université de Grenoble-Alpes, Université de Rennes 1, Université de Strasbourg, Université de Toulouse, CEA, CNRS, Inria.
- Website: [numpex.org/exasoft-hpc-software-and-tools](https://numpex.org/exasoft-hpc-software-and-tools)
- Coordinator: Raymond NAMYST (Storm)
- Abstract:

Though significant efforts have been devoted to the implementation and optimization of several crucial parts of a typical HPC software stack, most HPC experts agree that exascale supercomputers will raise new challenges, mostly because the trend in exascale compute-node hardware is toward heterogeneity and scalability: Compute nodes of future systems will have a combination of regular CPUs and accelerators (typically GPUs), along with a diversity of GPU architectures. Meeting the needs of complex parallel applications and the requirements of exascale architectures raises numerous challenges which are still left unaddressed. As a result, several parts of the software stack must evolve to better support these architectures. More importantly, the links between these parts must be strengthened to form a coherent, tightly integrated software suite. Our project aims at consolidating the exascale software ecosystem by providing a coherent, exascale- ready software stack featuring breakthrough research advances enabled by multidisciplinary collaborations between researchers. The main scientific challenges we intend to address are: productivity, performance portability, heterogeneity, scalability and resilience, performance and energy efficiency.

### Numpex PC3: Exa-DoST

**Participants:** Francieli Boito, Emmanuel Jeannot, Luan Teylo.

<sup>3</sup>Inria has launched the InriaSoft program to help support open source software products authored by Inria and its partners when their usage has gone beyond the academic circles of their initial research context and when some key users are willing to become involved to support future developments.

- Exa-DoST: Data-oriented Software and Tools for the Exascale
- Program: project PC3 in PEPR Numpex
- 2023-2029
- Partners: Université Paris-Saclay, Telecom SudParis, Bordeaux INP, ENSIIE, Université de Bordeaux, Université de Grenoble-Alpes, Université de Rennes 1, Université de Strasbourg, Université de Toulouse, CEA, CNRS, Inria.
- Website: [numpex.org/exadost-data-oriented-software-and-tools-for-the-exascale/](https://numpex.org/exadost-data-oriented-software-and-tools-for-the-exascale/)
- Coordinator: Gabriel ANTONIU (KerData)
- Abstract:

The advent of future Exascale supercomputers raises multiple data-related challenges. To enable applications to fully leverage the upcoming infrastructures, a major challenge concerns the scalability of techniques used for data storage, transfer, processing and analytics. Additional key challenges emerge from the need to adequately exploit emerging technologies for storage and processing, leading to new, more complex storage hierarchies. Finally, it now becomes necessary to support more and more complex hybrid workflows involving at the same time simulation, analytics and learning, running at extreme scales across supercomputers interconnected to clouds and edgebased systems. The Exa-DoST project will address most of these challenges, organized in 3 areas: 1. Scalable storage and I/O; 2. Scalable in situ processing; 3. Scalable smart analytics. As part of the NumPEX program, Exa-DoST will address the major data challenges by proposing operational solutions co-designed and validated in French and European applications. This will allow filling the gap left by previous international projects to ensure that French and European needs are taken into account in the roadmaps for building the data-oriented Exascale software stack.

### Inria Exploratory Action

**Participants:** Asia Auville, Emmanuelle Saillard, Mihail Popov.

- Title: Large Language Models for Detection and Correction of Errors
- Website: [LLM4DiCE](#)
- 2024 - 2027 (36 months)
- Coordinator: Emmanuelle Saillard and Mihail Popov
- Abstract: Large Language Models (LLMs) are a hot and rapidly evolving research topic. In particular, their recent successes in summarization, question-answering, and code generation with AI pair programming make them attractive candidates in the field of error verification. We propose to harness these LLMs capabilities with fine-tuning on carefully generated datasets through a novel clustering strategy based on Natural Language Processing (NLP) techniques and code embedding to assist bug detection and correction, targeting hard domains such as parallel program verification.

## 10.5 Public policy support

François Pellegrini was short-listed by a jury set-up by the European Commission, for the position of chair of the European Data Protection Supervisor, the independent data protection authority that supervises the use of personal data by the bodies of the European Union. He was subsequently heard before the European Parliament and the European Council. Due to a lack of agreement between these two institutions, the hiring process is currently on hold.

## 11 Dissemination

### 11.1 Promoting scientific activities

#### 11.1.1 Scientific events: organisation

##### Member of the organizing committees

- Francieli Boito is Deputy chair for the Workshops part of ISC 2026, and vice-chair for the Workshops part of IPDPS 2026. In both cases, the work started in 2025.

#### 11.1.2 Scientific events: selection

##### Chair of conference program committees

- Mihail Popov was co-chair for the HIPC 2025 System Software (High Performance Computing) track.

##### Member of the conference program committees

- Francieli Boito was a member of the program committees of HPCAsia 2025, ICPP 2025, Cluster 2025, HiPC 2025, Rex-IO (a workshop held with Cluster) 2025, and PDSW (a workshop held with Supercomputing) 2025.
- Brice Goglin was a member of the program committees of Cluster 2025 and ICPP 2025.
- Mihail Popov was a member of the program committees of IPDPSW GrAPL 2025 and ISCW 2025.
- Guillaume Mercier was a member of the program committees of EuroMPI 2025, HiPC 2025 and ICPP 2025.
- Luan Teylo was a member of the program committee of CCGRID 2025, PMBS 2025, WAMCA 2025, ISPDC 2025, ESSA 2025, HiPC 2025 and SBAC 2025.
- Alexandre Denis was a member of the program committees of APDCM 2025 and HiPC 2025.

#### 11.1.3 Journal

- Mihail Popov was a reviewer for TPDS, JPDC, Journal of Cloud Computing, and journal of Supercomputing.
- Guillaume Mercier served as a reviewer for the Journal of Parallel and Distributed Computing.
- Luan Teylo was a reviewer for TPDS.

##### Reviewer - reviewing activities

- Francieli Boito was an external reviewer for IPDPS 2025.
- Guillaume Mercier served as an external reviewer for SC 2025,
- Luan Teylo was an external reviewer for SC 2025 and Cluster 2025.
- Alexandre Denis was an external reviewer for Cluster 2025.

#### 11.1.4 Invited talks

- Francieli Boito gave an invited talk at the NHR Conference (a German national event) in September 2025 — “Investigating the temporal I/O behavior of HPC applications”.
- Francieli Boito gave a keynote during the ESSA workshop (held with IPDPS) in June 2025 — “Improving I/O Resource Usage in HPC”.
- Brice Goglin was invited to give a talk at the French Academy of Science in May as a followup to the 2024 "Innovation" award from l'Academie des Sciences, Dassault Systèmes and Inria.

### 11.1.5 Scientific expertise

- Brice Goglin was a member of the Khronos OpenCL Advisory Panel as well as the Unified Acceleration Foundation (former oneAPI) Hardware Abstraction SIG.
- Brice Goglin is involved in the expertise of HPC projects in Africa with IRD and AFD.

### 11.1.6 Research administration

- Francieli Boito is a member of the council of the SIN department of the University of Bordeaux since 2022.
- Brice Goglin is the product owner of the Inria' nation-wide computing infrastructure, Abaca, since June 2025.
- Brice Goglin is in charge of the computing infrastructures of the Inria Bordeaux research center.

### 11.1.7 Standardization Activities

#### Participation in the MPI Forum

- TADAAM attended the MPI Forum meetings on behalf of Inria (where the MPI standard for communication in parallel applications is developed and maintained). Guillaume Mercier leads the *Topologies* working group. He participates in several other Working Groups (Hybrid WG, ABI WG) and is also an editor of the MPI Standard, as a member of several chapter committees (Contexts, Topologies and Info object). He also serves as the Context chapter committee chair. This year, the version 5.0 of the standard was approved by the MPI forum, the major addition to the MPI standard being the introduction of an Abstract Binary Interface (ABI) that is expected to improve applications portability since several implementations of MPI do coexist. This ABI support will enable applications to switch (more or less) effortlessly from one implementation to another (or even from one implementation *version* to another).

#### Participation in the PMIx ASC

- TADAAM is a member of the Administrative Steering Committee of PMIx standard focused on orchestration of application launch and execution.

## 11.2 Teaching - Supervision - Juries

### 11.2.1 Teaching

Members of the TADAAM project gave hundreds of hours of teaching at Université de Bordeaux and the Bordeaux INP engineering school, covering a wide range of topics from basic use of computers, introduction to algorithmic and C programming to advanced topics such as probabilities and statistics, scheduling, computer networks, computer architecture, operating systems, big data, cryptography, parallel programming and high-performance runtime systems, as well as software law and personal data law.

- François Pellegrini did the introductory conference of the *Numerics* graduate program at Université de Bordeaux, on the ethical issues of automated data processing.
- François Pellegrini did a course in English on “*Software Law*” and “*Personal data law*” to 9 PhD students of Université de Bordeaux.
- François Pellegrini did a new course on personal data and cybersecurity to 35 professional (lawyers, IT managers, etc.) attending the University Degree (D.U.) of cyber-criminology of Université Bordeaux-Montaigne with support from Gendarmerie Nationale.
- François Pellegrini did a new course on "Technological and societal innovations" to the 15 students of the international master "Law for Innovation" of Université de Bordeaux.

- Luan Teylo taught a course on data visualization with Python to undergraduate students from various fields of study at the Université de Bordeaux.
- Alexandre Denis teaches a course on MPI+X at ENSEIRB-MATMECA.
- Mihail Popov is the head of the cryptography and parallel programming courses, both at ENSEIRB-MATMECA.

### 11.2.2 Supervision

- PhD finished: Lana Scravaglieri, Portable vectorization with numerical accuracy control for multi-precision simulation codes. Advisors: Olivier Aumage, Mihail Popov, Thomas Guignon (IFPEN) and Ani Anciaux-Sedrakian (IFPEN).
- PhD in progress: Asia Auville, Large Language Models for Detection and Correction of Errors in HPC Applications. Advisors: Emmanuelle Saillard, Mihail Popov, Pablo Oliveira (UVSQ) and Eric Petit (Intel).
- PhD in progress: Charles Goedefroit, Delivering userspace interrupts from the BXI network interface. co-advised with ATOS/Bull/Eviden. Started in March 2024. Advisors: Alexandre Denis and Brice Goglin.
- PhD in progress: Serge Meurrens, Application-aware I/O scheduling in HPC systems. Started in December 2025. Advisors: Francieli Boito, François Tessier (Inria Rennes), and Luan Teylo.
- PhD in progress: Thibaut Pepin, MPI communication on modular supercomputing architectures, started in May 2023. Advisors: Guillaume Mercier.
- PhD in progress: Tristan Riehs, Integration of communications and task scheduling. Started in October 2025. Advisors: Alexandre Denis, Philippe Swartvagher (TOPAL), Samuel Thibault (Storm).
- PhD in progress: Méline Trochon, Adaptive checkpointing strategies depending on the network load. Started in November 2024. Advisors: Francieli Boito, François Tessier, Brice Goglin and Jean-Thomas Acquaviva (DDN).

### 11.2.3 Juries

- Francieli Boito was a member of the PhD jury of Adrian KHELILI (University of Paris Saclay).
- Brice Goglin was reviewer for the PhD thesis of Ioannis Vardas (TUWien) and Mickael Boichot (TelecomSudParis and CEA/DAM).
- Brice Goglin was the president of the PhD defense jury of Radjasouria Vinayagame (Univ. Bordeaux and Eviden).

## 11.3 Popularization

### 11.3.1 Participation in Live events

- Brice Goglin gave talks about research in computer science and high-performance computing to high-school students as part of the *Chiche* programme and *Circuit Scientifique Bordelais*, and about science and research to elementary schools' classes.
- Brice Goglin and Mihail Popov presented research in HPC and AI to middle school interns.

## 12 Scientific production

### 12.1 Major publications

- [1] J. L. Bez, A. Miranda, R. Nou, F. Z. Boito, T. Cortes and P. Navaux. ‘Arbitration Policies for On-Demand User-Level I/O Forwarding on HPC Platforms’. In: IPDPS 2021 - 35th IEEE International Parallel and Distributed Processing Symposium. Portland, Oregon / Virtual, United States, 17th May 2021. URL: <https://hal.inria.fr/hal-03149582>.
- [2] F. Boito, G. Pallez, L. Teylo and N. Vidal. ‘IO-SETS: Simple and efficient approaches for I/O bandwidth management’. In: *IEEE Transactions on Parallel and Distributed Systems* 34.10 (15th Aug. 2023), pp. 2783–2796. DOI: [10.1109/TPDS.2023.3305028](https://doi.org/10.1109/TPDS.2023.3305028). URL: <https://inria.hal.science/hal-03648225>.
- [3] A. Denis. ‘Scalability of the NewMadeleine Communication Library for Large Numbers of MPI Point-to-Point Requests’. In: CCGrid 2019 - 19th Annual IEEE/ACM International Symposium in Cluster, Cloud, and Grid Computing. Larnaca, Cyprus, 14th May 2019. URL: <https://hal.inria.fr/hal-02103700>.
- [4] N. Denoyelle, B. Goglin, A. Ilic, E. Jeannot and L. Sousa. ‘Modeling Non-Uniform Memory Access on Large Compute Nodes with the Cache-Aware Roofline Model’. In: *IEEE Transactions on Parallel and Distributed Systems* 30.6 (June 2019), pp. 1374–1389. DOI: [10.1109/TPDS.2018.2883056](https://doi.org/10.1109/TPDS.2018.2883056). URL: <https://hal.inria.fr/hal-01924951>.
- [5] A. Gainaru, B. Goglin, V. Honoré and G. Pallez. ‘Profiles of upcoming HPC Applications and their Impact on Reservation Strategies’. In: *IEEE Transactions on Parallel and Distributed Systems* 32.5 (May 2021), pp. 1178–1190. DOI: [10.1109/TPDS.2020.3039728](https://doi.org/10.1109/TPDS.2020.3039728). URL: <https://hal.inria.fr/hal-03010676>.
- [6] B. Goglin, E. Jeannot, F. Mansouri and G. Mercier. ‘Hardware topology management in MPI applications through hierarchical communicators’. In: *Parallel Computing* 76 (Aug. 2018), pp. 70–90. DOI: [10.1016/j.parco.2018.05.006](https://doi.org/10.1016/j.parco.2018.05.006). URL: <https://hal.inria.fr/hal-01937123>.
- [7] L. Scravaglieri, M. Popov, L. Lima Pilla, A. Guermouche, O. Aumage and E. Saillard. ‘Optimizing Performance and Energy Across Problem Sizes Through a Search Space Exploration and Machine Learning’. In: *Journal of Parallel and Distributed Computing* 180 (28th June 2023), p. 104720. DOI: [10.1016/j.jpdc.2023.104720](https://doi.org/10.1016/j.jpdc.2023.104720). URL: <https://hal.science/hal-03810305>.
- [8] F. Zanon Boito, L. Teylo, M. Popov, T. Jolivel, F. Tessier, J. Luetzgau, J. Monniot, A. Tarraf, A. Carneiro and C. Osthoff. *A Deep Look Into the Temporal I/O Behavior of HPC Applications*. 15th Jan. 2025. URL: <https://inria.hal.science/hal-04887809>.

### 12.2 Publications of the year

#### International peer-reviewed conferences

- [9] F. Boito, L. Teylo, M. Popov, T. Jolivel, F. Tessier, J. Luetzgau, J. Monniot, A. Tarraf, A. Carneiro and C. Osthoff. ‘A Deep Look Into the Temporal I/O Behavior of HPC Applications’. In: *39th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*. 39th IEEE International Parallel & Distributed Processing Symposium (IPDPS). Milan, Italy, 3rd June 2025. DOI: [10.1109/IPDPS64566.2025.00072](https://doi.org/10.1109/IPDPS64566.2025.00072). URL: <https://inria.hal.science/hal-04887809> (cit. on pp. 16, 18).
- [10] A. Denis and C. Goedefroit. ‘NBLFQ: a lock-free MPMC queue optimized for low contention’. In: IPDPS 2025 - 39th International Parallel & Distributed Processing Symposium. International Parallel & Distributed Processing Symposium. Milan, Italy: IEEE, June 2025. URL: <https://inria.hal.science/hal-04851700> (cit. on p. 22).

- [11] W. Ferreira, L. Kunstmann, Y. Frota, L. Teylo and D. D. Oliveira. ‘A Weighted Bi-objective Strategy for Executing Scientific Workflows in Containerized Environments’. In: *2025: Anais do XXVI Simpósio em Sistemas Computacionais de Alto Desempenho*. Simpósio em Sistemas Computacionais de Alto Desempenho. Bonito - MS, Brazil: Sociedade Brasileira de Computação, 28th Oct. 2025, pp. 350–361. DOI: [10.5753/sscad.2025.16730](https://doi.org/10.5753/sscad.2025.16730). URL: <https://inria.hal.science/hal-05467004> (cit. on p. 19).
- [12] C. Gavaille, H. Taboada, J. Domke, B. Goglin and E. Jeannot. ‘Performance Projection for Design-Space Exploration on future HPC Architectures’. In: *Proceedings of the 39th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IPDPS 2025 - 39th IEEE International Parallel & Distributed Processing Symposium. Milano, Italy: IEEE, June 2025. URL: <https://inria.hal.science/hal-04856139> (cit. on pp. 21, 23).
- [13] C. Goedefroit, A. Denis, M. Barbe, B. Goglin and G. Pichon. ‘Communication Notification through User-Level Interrupts for the BXI Network’. In: *Cluster 2025 - 27th IEEE International Conference on Cluster Computing*. Edinburgh (Scotland), United Kingdom, 2nd Sept. 2025. URL: <https://inria.hal.science/hal-05150209> (cit. on p. 21).
- [14] M. de Lima, L. Teylo and L. Drummond. ‘Towards a Novel Vertical Scaling Approach for Bursty Workloads in Kubernetes’. In: *UCC '25: Proceedings of the 18th IEEE/ACM International Conference on Utility and Cloud Computing*. IEEE/ACM UCC 2025 - 18th IEEE/ACM International Conference on Utility and Cloud Computing. Nantes, France, Dec. 2025, pp. 1–6. DOI: [10.1145/3773274.3774699](https://doi.org/10.1145/3773274.3774699). URL: <https://inria.hal.science/hal-05466865> (cit. on p. 20).
- [15] T. Pepin, J. Jaeger, G. Mercier and B. Goglin. ‘Toward unprivileged, portable and generic network topology discovery’. In: *SCA/HPCAsia '26: Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region*. SCA/HPCAsia 2026 - Supercomputing Asia and International Conference on High Performance Computing in Asia Pacific Region. Osaka, Japan: ACM, 25th Jan. 2026, pp. 271–283. DOI: [10.1145/3773656.3773657](https://doi.org/10.1145/3773656.3773657). URL: <https://hal.science/hal-05468186> (cit. on p. 20).
- [16] L. Scravaglieri, A. Anciaux-Sedrakian, O. Aumage, T. Guignon and M. Popov. ‘Compiler, Runtime, and Hardware Parameters Design Space Exploration’. In: *IPDPS 2025 - 39th IEEE International Parallel and Distributed Processing Symposium*. Milan, Italy, Feb. 2025. URL: <https://inria.hal.science/hal-04969854>.
- [17] M. Trochon, J. Bigot, V. Grandgirard and D. Midou. ‘Checkpointing Optimisation to Prepare Future Exascale Plasma Turbulence Simulations’. In: *IPDPSW 2025 - 39th IEEE International Parallel & Distributed Processing Symposium*. Milan, Italy, 3rd June 2025. URL: <https://hal.science/hal-05105811> (cit. on p. 19).

#### Conferences without proceedings

- [18] M. Abdraman, F. Boito and L. Teylo. ‘IOPS: I/O Performance Evaluation Suite’. In: *ESSA 2025 - 6th Workshop on Extreme-Scale Storage and Analysis*. Milan, Italy, 4th June 2025. URL: <https://hal.science/hal-05120034> (cit. on p. 17).
- [19] T. Chatelain. ‘Anticipation des communications réseau grâce à la connaissance du futur dans le parallélisme à tâches’. In: *COMPAS 2025 - Conférence francophone d’informatique en Parallélisme, Architecture et Système*. Bordeaux, France, 24th June 2025. URL: <https://hal.science/hal-05147860> (cit. on p. 22).
- [20] C. Goedefroit, M. Barbe, A. Denis, B. Goglin and G. Pichon. ‘Interruptions en espace utilisateur depuis le réseau BXI pour le recouvrement calcul/communications’. In: *COMPAS 2025 - Conférence francophone d’informatique en Parallélisme, Architecture et Système*. Bordeaux, France, 24th June 2025. URL: <https://inria.hal.science/hal-05085582> (cit. on p. 21).
- [21] B. Goglin, J. Jaeger, G. Mercier and T. Pépin. ‘Placement de tâches MPI pour minimiser la charge par carte réseau et améliorer la localité’. In: *COMPAS 2025*. Bordeaux (France), France, 24th June 2025. URL: <https://inria.hal.science/hal-05150331>.

- [22] M. Trochon. ‘Checkpointing optimisation to prepare future exascale plasma turbulence simulations’. In: Conférence francophone d’informatique en Parallélisme, Architecture et Système (COMPAS 2025). Bordeaux, France, 24th June 2025. URL: <https://hal.science/hal-05150262>.

#### Doctoral dissertations and habilitation theses

- [23] F. Boito. ‘Towards Better I/O Resource Usage in HPC’. Université de Bordeaux, 5th Dec. 2025. URL: <https://inria.hal.science/tel-05428671>.

#### Reports & preprints

- [24] S. Arias, M. Bergmann, F. Campillo, M.-A. Enard, C. Fabre, F. Garcia, B. Guedj, E. Jeannot, G. Neglia, D. Peurichard, D. Racoceanu, B. Sagot and G. Tworkowski. *Reflections on the Use of Generative AI for Research Professions*. Inria, 9th July 2025. URL: <https://inria.hal.science/hal-05188001>.
- [25] S. Arias, M. Bergmann, F. Campillo, M.-A. Enard, C. Fabre, F. Garcia, B. Guedj, E. Jeannot, G. Neglia, D. Peurichard, D. Racoceanu, B. Sagot and G. Tworkowski. *Réflexions sur l’usage de l’IA générative pour les métiers de la recherche*. Inria, 2025, pp. 1–10. URL: <https://inria.hal.science/hal-05187992>.
- [26] R. Boëzennec, F. Fernandes dos Santos, B. Goglin, A. Kritikakou, G. Pallez, E. Rohou, O. Sentieys and M. Traiola. *Increasing the Lifetime of HPC Machines: Issues, Implications, and Open Challenges*. 2025. URL: <https://hal.science/hal-05312072> (cit. on p. 9).
- [27] F. Boito, L. Teylo, M. Popov, T. Jolivel, F. Tessier, J. Luettgau, J. Monniot, A. Tarraf, A. Carneiro and C. Osthoff. *A deep look into the temporal I/O behavior of HPC applications -extended version*. RR-9577. Inria & Labri, Univ. Bordeaux, 5th Mar. 2025, pp. 1–42. URL: <https://inria.hal.science/hal-04978752> (cit. on pp. 16, 18).
- [28] M. Trochon, J.-T. Acquaviva, F. Boito, B. Goglin, F. Tessier and L. Teylo. *On the Impact of Interference from Concurrent Jobs on Checkpointing Performance*. 2025. URL: <https://hal.science/hal-05294610> (cit. on p. 18).

#### Other scientific publications

- [29] T. Chatelain. ‘Anticipation des communications réseau grâce à la connaissance du futur dans le parallélisme à tâche’. Enseirb-Matmeca, Sept. 2025. URL: <https://inria.hal.science/hal-05281620> (cit. on p. 22).
- [30] E. A. Leon, S. Gutiérrez and G. Mercier. ‘Improving Productivity of Threaded Scientific Applications with Quo Vadis’. In: PASC 2025 - Platform for Advanced Scientific Computing. Windisch, Switzerland, 16th June 2025. URL: <https://inria.hal.science/hal-05176401> (cit. on p. 20).
- [31] N. Tamssaout. ‘Full stack optimization for streaming applications’. Inria, 1st Oct. 2025. URL: <https://inria.hal.science/hal-05311328> (cit. on p. 17).
- [32] G. Valade. ‘Composabilité de drivers au sein de NewMadeleine’. Enseirb-Matmeca, 9th Oct. 2025. URL: <https://inria.hal.science/hal-05450657> (cit. on p. 22).

### 12.3 Cited publications

- [33] A. Bandet, F. Boito and G. Pallez. ‘Prediction and Interpretability of HPC I/O Resources Usage with Machine Learning’. working paper or preprint. 2024. URL: <https://inria.hal.science/hal-04698511> (cit. on p. 18).
- [34] A. Bandet, F. Boito and G. Pallez. ‘Scheduling Distributed I/O Resources in HPC Systems’. In: *Euro-Par 2024: Parallel Processing: 30th European Conference on Parallel and Distributed Processing, Madrid, Spain, August 26–30, 2024, Proceedings, Part I*. Madrid, Spain: Springer-Verlag, 2024, pp. 137–151. DOI: [10.1007/978-3-031-69577-3\\_10](https://doi.org/10.1007/978-3-031-69577-3_10). URL: [https://doi.org/10.1007/978-3-031-69577-3\\_10](https://doi.org/10.1007/978-3-031-69577-3_10) (cit. on pp. 9, 17).

- [35] F. Boito, G. Pallez, L. Teylo and N. Vidal. ‘IO-SETS: Simple and efficient approaches for I/O bandwidth management’. In: *IEEE Transactions on Parallel and Distributed Systems* 34.10 (Aug. 2023), pp. 2783–2796. DOI: [10.1109/TPDS.2023.3305028](https://doi.org/10.1109/TPDS.2023.3305028). URL: <https://inria.hal.science/hal-03648225> (cit. on p. 9).
- [36] B. Daneshkhah. *Exploring the Search Space of Hybrid E/P Cores Prefetch Configurations using Machine Learning*. 2025. URL: <https://www.diva-portal.org/smash/get/diva2:1985357/FULLTEXT02.pdf> (visited on 30/12/2025) (cit. on p. 17).
- [37] J. E. Karchi, H. Chen, A. Tehranijamsaz, A. Jannesari, M. Popov and E. Saillard. ‘MPI Errors Detection using GNN Embedding and Vector Embedding over LLVM IR’. In: *IPDPS 2024 - 38th International Symposium on Parallel and Distributed Processing*. San Francisco, United States, May 2024. URL: <https://inria.hal.science/hal-04724011> (cit. on p. 16).
- [38] A. Lorén. *Hybrid E/P Cores Prefetch Optimization*. 2024. URL: <https://www.diva-portal.org/smash/get/diva2:1888603/FULLTEXT01.pdf> (visited on 30/12/2024) (cit. on p. 17).