RESEARCH CENTRE

Inria Center at the University of Bordeaux

IN PARTNERSHIP WITH:

Institut Polytechnique de Bordeaux, Université de Bordeaux

2022
ACTIVITY REPORT

Project-Team

TADAAM

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

IN COLLABORATION WITH: Laboratoire Bordelais de Recherche en Informatique (LaBRI)

DOMAIN

Networks, Systems and Services, Distributed Computing

THEME

Distributed and High Performance Computing

Contents

Project-Team TADAAM

Creation of the Project-Team: 2017 December 01

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

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)

A6.1.2. – Stochastic Modeling

A6.2.3. – Probabilistic methods

A6.2.6. – Optimization

A6.2.7. – High performance computing

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

Other research topics and application domains

B6.3.2. – Network protocols

B6.3.3. – Network Management

B9.5.1. – Computer science

B9.8. – Reproducibility

1 Team members, visitors, external collaborators

Research Scientists

- Emmanuel Jeannot [Team leader, Inria, Senior Researcher, HDR]
- Alexandre Denis [Inria, Researcher]
- Brice Goglin [Inria, Senior Researcher, HDR]
- Guillaume Pallez [Inria, Researcher, until Aug 2022]

Faculty Members

- Guillaume Mercier [Bordeaux INP, Associate Professor, HDR]
- François Pellegrini [Univ. Bordeaux, Professor, HDR]
- Francieli Zanon-Boito [Univ. Bordeaux, Associate Professor]

Post-Doctoral Fellows

- Clément Foyer [Inria, until Aug 2022]
- Luan Teylo Gouveia Lima [Inria]
- Nicolas Vidal [Inria, from Feb 2022 until Aug 2022]

PhD Students

- Alexis Bandet [Inria]
- Robin Boezennec [Inria, from Sep 2022]
- Clément Gavoille [CEA]
- Florian Reynier [CEA, until Jun 2022]
- Julien Rodriguez [CEA]
- Richard Sartori [Atos]
- Philippe Swartvagher [Inria, until Nov 2022]
- Nicolas Vidal [Inria, until Jan 2022]

Technical Staff

- Clément Barthelemy [Inria, Engineer]
- Quentin Buot [Atos, Engineer, from Oct 2022]

Interns and Apprentices

- Robin Boezennec [Inria, from Feb 2022 until Jul 2022]
- Alessa Mayer [Inria, from Jun 2022 until Aug 2022]
- Corentin Mercier [Inria, from Mar 2022 until Aug 2022]
- Louis Peyrondet [Inria, from Jun 2022 until Aug 2022]

Administrative Assistant

- Catherine Cattaert Megrat [Inria]

Visiting Scientist

- Jannis Klinkenberg [RWTH Aachen University, from Jun 2022 until Aug 2022]

External Collaborator

- Elia Verdon [Univ. Bordeaux]

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 layer though an API. The service layer 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 this service layer through a well-defined set of functions, based on carefully designed abstractions.

Hence, **the goal of TADAAM is to design a stateful system-wide service layer for HPC systems, in order to optimize applications execution according to their needs**.

This layer 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, this layer will feature an engine 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 [1]. However, very few applications use the whole machine. In general, an application runs only on a subset of the nodes [2]. 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

[1] 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.

[2] 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

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 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

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, recent research in the team [28] 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 ≈ 88 hours on nine nodes, while the simulation results that made into the paper would take at least 569 days to run [28]. 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.

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 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

Several team members advocate for responsible use of digital tools in human activities. Members of the team contributed to a report on *Indicators for monitoring Inria's scientific activity* which includes high level discussions on the impact of evaluation of science. Members of the team participated to the writing of the *Inria global Action plan on F/M professional equality for 2021-2024.*

6 Highlights of the year

- Brice Goglin worked with AMD and HPE/Cray to improve the modeling of the Frontier supercomputer in the HWLOC software so that runtimes could better optimize task and data placement in this complex hybrid architecture. This work is used in production and helped breaking the Exaflop barrier.

- Guillaume Pallez was appointed SC'24 Technical Program Chair. SC'24 is the largest conference in the field with approximatively 13k participants. Guillaume Pallez will oversee the organisation of all technical content: papers, posters, workshops, awards, panels, etc. This is the first time a French person has been nominated to this position.

- A consortium is being created to foster the development of the SCOTCH software. Its call for members has been launched for the 30th anniversary of the software[3].
- Our paper [17] entitled “Modeling Memory Contention between Communications and Computations in Distributed HPC Systems” got the *best paper* award at the APDCM workshop.
- ICPP 2022 was organized (remotely) at Bordeaux this year[4]. Changes in the Inria processes have impaired our ability to manage the whole ICPP logistics. Together with other reasons, this forced us to make it virtual. Among the organizing committee, Emmanuel Jeannot and Guillaume Pallez were General chairs, Brice Goglin was finance chair and Clément Foyer proceedings chair.

7 New software and platforms

7.1 New software

7.1.1 Hsplit

Name: Hardware communicators split

Keywords: MPI communication, Topology, Hardware platform

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 o 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 o 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.

News of the Year: Our proposal forms the basis of a new feature that was voted in MPI 4.0 by the MPI Forum.

URL: `https://gitlab.inria.fr/hsplit/hsplit`

Publications: hal-01937123v2, hal-01621941, hal-01538002

Contact: Guillaume Mercier

Participants: Guillaume Mercier, Brice Goglin, Emmanuel Jeannot

[3] Call here
[4] Website here

7.1.2 hwloc

Name: Hardware Locality

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

Scientific Description: In 2022, the support for Intel GPUs in the L0 backend was improved with subdevice, memory and Xe fabric support. Heterogeneous memory description was also enhanced with a heuristics that guesses whether a NUMA node is DRAM, HBM or NVM, and some detection of future CXL memory expanders. Support for hybrid processors was also improved.

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.

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: Brice Goglin, Valentin Hoyet

Partners: Open MPI consortium, Intel, AMD, IBM

7.1.3 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.

News of the Year: NewMadeleine now features tag matching in constant time, allowing for a good scalability in number of requests. A dynamic multicast has been added to be used in conjunction with StarPU. The MPI I/O subsystem has been extended so as to be able to run HDF5 codes.

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, Clément Foyer, Nathalie Furmento, Raymond Namyst, Adrien Guilbaud, Florian Reynier, Philippe Swartvagher

7.1.4 TopoMatch

Keywords: Intensive parallel computing, High-Performance Computing, Hierarchical architecture, Placement

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.

News of the Year: We have worked on consolidating the software and on the fact that the mapping can be performed on arbitrary topologies.

URL: `https://gitlab.inria.fr/ejeannot/topomatch`

Publication: hal-03780662

Contact: Emmanuel Jeannot

Participants: Adele Villiermet, Emmanuel Jeannot, François Tessier, Guillaume Mercier, Pierre Celor

Partners: Université de Bordeaux, CNRS, IPB

7.1.5 SCOTCH

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

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: A consortium is being created to foster the development of Scotch. A call for founding members has been launched on 01 December 2022, for the 30th anniversary of the software. See: https://team.inria.fr/tadaam/call-for-founding-members-for-the-scotch-consortium/

URL: `http://www.labri.fr/~pelegrin/scotch/`

Publications: 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, Sébastien Fourestier, Jun-Ho Her, Cédric Chevalier, Amaury Jacques, Selmane Lebdaoui, Marc Fuentes

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

7.1.6 AGIOS

Name: Application-guided I/O Scheduler

Keywords: High-Performance Computing, Scheduling

Scientific Description: This library is being adapted in the context of the ADMIRE EuroHPC project.

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.

News of the Year: In 2022, a WFQ scheduling algorithm was added to the library, in order to support the implementation of the IO-Sets approach proposed by the Tadaam team for I/O bandwidth control.

URL: `https://github.com/francielizanon/agios`

Publications: hal-02079899, hal-01247942, hal-03758890

Contact: Francieli Zanon-Boito

Participants: Luan Teylo Gouveia Lima, Alessa Mayer

7.1.7 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.

News of the Year: In 2022, Raisin was born. Several algorithms have been implemented to compute mappings in a multilevel framework: the initial partitioning algorithms DDFS and DBFS, a clustering algorithm, and the DKFM refinement algorithm.

Publication: hal-03604540v1

Contact: Julien Rodriguez

Participants: François Galea, François Pellegrini, Lilia Zaourar, Julien Rodriguez

7.2 New platforms

7.2.1 PlaFRIM

Participants: Brice Goglin.

Name: Plateforme Fédérative pour la Recherche en Informatique et Mathématiques

Website: 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 2021, 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.

8 New results

8.1 Relative performance projection on Arm architectures

Participants: Brice Goglin, Clément Gavoille, Emmanuel Jeannot.

With the advent of multi- many-core processors and hardware accelerators, choosing a specific architecture to renew a supercomputer can become very tedious. This decision process should consider the current and future parallel application needs and the design of the target software stack. It should also consider the single-core behavior of the application as it is one of the performance limitations in today's machines. In such a scheme, performance hints on the impact of some hardware and software stack modifications are mandatory to drive this choice. This paper proposes a workflow for performance projection based on execution on an actual processor and the application's behavior. This projection evaluates the performance variation from an existing core of a processor to a hypothetical one to drive the design choice. For this purpose, we characterize the maximum sustainable performance of the target machine and analyze the application using the software stack of the target machine. To validate this approach, we apply it to three applications of the CORAL benchmark suite: LULESH, MiniFE, and Quicksilver, using a single-core of two Arm-based architectures: Marvell ThunderX2 and Arm Neoverse N1. Finally, we follow this validation work with an example of design-space exploration around the SVE vector size, the choice of DDR4 and HBM2, and the software stack choice on A64FX on our applications with a pool of three source architectures: Arm Neoverse N1, Marvell ThunderX2, and Fujitsu A64FX.

This work [18] was performed in collaboration with CEA/DAM and ARM.

8.2 Using performance attributes for managing heterogeneous memory in HPC applications

Participants: Brice Goglin, Andrés Rubio Proaño.

The complexity of memory systems has increased considerably over the past decade. Supercomputers may now include several levels of heterogeneous and non-uniform memory, with significantly different properties in terms of performance, capacity, persistence, etc. Developers of scientific applications face a huge challenge: efficiently exploit the memory system to improve performance, but keep productivity high by using portable solutions.

In this work [19], we present a new API and a method to manage the complexity of modern memory systems. Our portable and abstracted API is designed to identify memory kinds and describe hardware characteristics using metrics, for example bandwidth, latency and capacity. It allows runtime systems, parallel libraries, and scientific applications to select the appropriate memory by expressing their needs for each allocation without having to re-modify the code for each platform. Furthermore we present a survey of existing ways to determine sensitivity of application buffers using static code analysis, profiling and benchmarking. We show in a use case that combining these approaches with our API indeed enables a portable and productive method to match application requirements and hardware memory characteristics.

8.3 Towards heuristics for data management in heterogeneous memory

Participants: Clément Foyer, Brice Goglin, Emmanuel Jeannot.

Currently, applications have to be heavily modified to use heterogeneous memory, often relying on vendor-specific APIs and software. Expecting that more and more machines in HPC will incorporate heterogeneous memory, there is a need for a portable, vendor-neutral solution to expose available kinds of memory and allocate data on those at runtime. Tools like *memkind* or *hwloc* already provide an API to identify memory (e.g., high bandwidth, large capacity) and allocate data on it.

More questions arise that are not trivial to answer. As memory with high bandwidth is limited in capacity, how to decide which data items to put in which kind of memory? And is it reasonable to consider such allocations optimal for the whole application duration? Further, how can we collect knowledge about data items and make that available for decision making?

We address these questions and challenges with the following approach. We review and extend methodologies to expose memory and associated memory characteristics. We present a runtime system that enables programmers to express hints about data and how to translate hints into memory placement decisions. Finally, we have been studying ways to identify memory access characteristics for data items both at compile- and at run-time.

This work [34] is performed in collaboration with RWTH Aachen in the context of the H2M ANR-DFG project.

8.4 Interferences between communications and computations in distributed HPC systems

Participants: Alexandre Denis, Emmanuel Jeannot, Philippe Swartvagher.

Parallel runtime systems such as MPI or task-based libraries provide models to manage both computation and communication by allocating cores, scheduling threads, executing communication algorithms. Efficiently implementing such models is challenging due to their interplay within the runtime system.

In [36, 38, 37, 39], we assess interferences between communications and computations when they run side by side. We study the impact of communications on computations, and conversely the impact of computations on communication performance. We consider two aspects: CPU frequency, and memory contention. We have designed benchmarks to measure these phenomena. We show that CPU frequency variations caused by computation have a small impact on communication latency and bandwidth. However, we have observed on Intel, AMD and ARM processors, that memory contention may cause a severe slowdown of computation and communication when they occur at the same time. We have designed a benchmark with a tunable arithmetic intensity that shows how interferences between communication and computation actually depend on memory pressure of the application. Finally we have observed up to 90 % performance loss on communications with common HPC kernels such as the conjugate gradient and general matrix multiplication.

Then we proposed [17, 8, 30, 23] a model to predict memory bandwidth for computations and for communications when they are executed side by side, according to data locality and taking contention into account. Elaboration of the model allowed to better understand locations of bottleneck in the memory system and what are the strategies of the memory system in case of contention. The model was evaluated on many platforms with different characteristics, and showed a prediction error in average lower than 4 %.

8.5 Tracing task-based runtime systems: feedbacks from the StarPU case

Participants: Alexandre Denis, Emmanuel Jeannot, Philippe Swartvagher.

Given the complexity of current supercomputers and applications, being able to trace application executions to understand their behaviour is not a luxury. As constraints, tracing systems have to be as little intrusive as possible in the application code and performances, and be precise enough in the collected data.

In an article currently under review, we present how we set up a tracing system to be used with the task-based runtime system StarPU. We study the different sources of performance overhead coming from the tracing system and how to reduce these overheads. Then, we evaluate the accuracy of distributed traces with different clock synchronization techniques. Finally, we summarize our experiments and conclusions with the lessons we learned to efficiently trace applications, and the list of characteristics each tracing system should feature to be competitive.

The reported experiments and implementation details comprise a feedback of integrating into a task-based runtime system state-of-the-art techniques to efficiently and precisely trace application executions. We highlight the points every application developer or end-user should be aware of to seamlessly integrate a tracing system or just trace application executions.

This work has been submitted to the journal *Concurrency and Computation: Practice and Experience*, and was highlighted in Guix-HPC report [29] as a usecase.

8.6 Use of dedicated core for nonblocking collective progression

Participants: Alexandre Denis, Emmanuel Jeannot, Florian Reynier.

Overlapping communications with computation is an efficient way to amortize the cost of communications of an HPC application. To do so, it is possible to utilize MPI nonblocking primitives so that communications run in background alongside computation. However, these mechanisms rely on communications actually making progress in background, which may not be true for all MPI libraries. Some MPI libraries leverage a core dedicated to communications to ensure communication progression. However, taking a core away from the application for such purpose may have a negative impact on the overall execution time. It may be difficult to know when such dedicated core is actually helpful.

We propose [16] a model for the performance of applications using MPI nonblocking primitives running on top of an MPI library with a dedicated core for communications. This model is used to understand the compromise between computation slowdown due to the communication core not being available for computation, and the communication speed-up thanks to the dedicated core; evaluate whether nonblocking communication is actually obtaining the expected performance in the context of the given application; predict the performance of a given application if run with a dedicated core.

We describe the performance model and evaluate it on different applications. We compare the predictions of the model with actual executions.

8.7 A methodology for assessing computation/communication overlap of MPI nonblocking collectives

Participants: Alexandre Denis, Emmanuel Jeannot, Florian Reynier.

By allowing computation/communication overlap, MPI-3 nonblocking collectives (NBC) are supposed to be a way to improve application scalability and performance. However, it is known that to actually get overlap, the MPI library has to implement progression mechanisms in software or rely on the network hardware. These mechanisms may be present or not, adequate or perfectible, they may have an impact on communication performance or may interfere with computation by stealing CPU cycles.

Hence, from a user point of view, assessing and understanding the behavior of an MPI library concerning computation/communication overlap of NBC is difficult.

We propose [7] a complete and thorough methodology to assess the computation/communication overlap of NBC. We first propose new metrics to measure how much communication and computation do overlap, and to evaluate how they interfere with each other. We integrate these metrics into a complete methodology that covers: a set of benchmarks to measure them, evaluation of the metrics on real-life MPI libraries as well as a set of guidelines to interpret the results. We perform experiments on a large panel of MPI implementations and network hardware and show that the proposed methodology enables understanding and assessing communication/computation overlap of NBC: when and why it is efficient, nonexistent or even degrades performance. Last, we compare our methodology with state of the art metrics and show that they provide an incomplete and sometimes misleading information.

8.8 Task-based randomized singular value decomposition and multidimensional scaling

Participants: Alexandre Denis, Emmanuel Jeannot, Adrien Guilbaud.

The multidimensional scaling (MDS) is an important and robust algorithm for representing individual cases of a dataset out of their respective dissimilarities. However, heuristics, possibly trading-off with robustness, are often preferred in practice due to the potentially prohibitive memory and computational costs of the MDS. The recent introduction of random projection techniques within the MDS allowed it to be become competitive on larger test cases.

The goal of this work [27] is to propose a high-performance distributed-memory MDS based on random projection for processing data sets of even larger size (up to one million items). We propose a task-based design of the whole algorithm and we implement it within an efficient software stack including state-of-the-art numerical solvers, runtime systems and communication layers. The outcome is the ability to efficiently apply robust MDS to large data sets on modern supercomputers. We assess the resulting algorithm and software stack to the point cloud visualization for analyzing distances between sequences in metabarcoding.

8.9 IO-Sets: simple and efficient approaches for I/O bandwidth management

Participants: Luan Gouveia Lima, Guillaume Pallez, Nicolas Vidal, Francieli Zanon-Boito.

One of the main performance issues faced by high- performance computing platforms is the congestion caused by concurrent I/O from applications. When this happens, the platform's overall performance and utilization are harmed. From the extensive work in this field, I/O scheduling is the essential solution to this problem. However, one main drawback of current techniques is the amount of information needed about applications, which compromises their applicability.

Previous work has shown that when multiple applications perform I/O phases simultaneously, it is best to grant exclusive access to one at a time, limiting interference. That strategy is especially well adapted for applications with similar periods (they perform I/O phases with a similar frequency). From this observation, we investigate different strategies for grouping applications according to their I/O frequency and design IO-Sets: a novel method for I/O management in HPC systems.

In IO-Sets, the applications are sorted into sets according to their characteristic time (mean time between I/O phases). Applications from the same set do I/O exclusively (one at a time). Nonetheless, applications from different sets can do I/O accesses simultaneously, and, in this case, they share the available bandwidth. For each one of the sets, it is given a priority that defines the portion of the I/O bandwidth the applications will receive when doing I/O concurrently.

In [28, 22], we present the potential of IO-Sets through a scheduling heuristic called SET-10, which is simple and requires only minimal information. Our extensive experimental campaign shows the importance of IO-Sets and the robustness of SET-10 under various workloads. We also provide insights on using our proposal in practice. The I/O sets paper was submitted to the *IEEE Transactions on Parallel and Distributed Systems* journal and is currently under review.

8.10 Arbitration policies for I/O forwarding on HPC platforms

Participants: Alexis Bandet, Guillaume Pallez, Francieli Zanon-Boito.

I/O forwarding is an established and widely-adopted technique in HPC to reduce contention and improve performance in the access to shared storage infrastructure. The typical approach is to statically assign I/O nodes to applications depending on the number of compute nodes they use, which is not always necessarily related to their I/O requirements. In [1], we investigated arbitration policies to assign I/O nodes to applications (i.e. to decide how many I/O nodes an application should use) while considering their characteristics. We proposed a policy based on the Multiple-Choice Knapsack problem that seeks to maximize global bandwidth by giving more I/O nodes to applications that will benefit the most. Furthermore, we proposed a user-level I/O forwarding solution as an on-demand service capable of applying different allocation policies at runtime for machines where this layer is not present. We demonstrated our approach's applicability through extensive experimentation and showed it can transparently improve global I/O bandwidth by up to 85% in a live setup compared to the default static policy.

In 2022 we continued working on this topic in the group. The previously proposed MCKP algorithm targets situations where the number of currently active applications is smaller than the number of available I/O nodes, and thus does not consider the option of having applications sharing I/O nodes. Avoiding it is a good idea because concurrent applications may suffer from interference while sharing resources. However, this characteristic limits the applicability of the technique, because i) in a machine we often have more running applications than I/O nodes, and ii) even if the number of applications actually doing I/O will often be smaller, detecting and identifying those applications is a challenge. Therefore, we are working towards a placement strategy that decides which applications should share I/O nodes depending on their I/O intensities.

8.11 The role of storage target allocation in applications' I/O performance with BeeGFS

Participants: Luan Gouveia Lima, Guillaume Pallez, Francieli Zanon-Boito.

Parallel file systems are at the core of HPC I/O infrastructures. Those systems minimize the I/O time of applications by separating files into fixed-size chunks and distributing them across multiple storage targets. Therefore, the I/O performance experienced with a PFS is directly linked to the capacity to retrieve these chunks in parallel. We conduct an in-depth evaluation of the impact of the stripe count (the number of targets used for striping) on the write performance of BeeGFS, one of the most popular parallel file systems today. We consider different network configurations and show the fundamental role played by this parameter, in addition to the number of compute nodes, processes and storage targets.

Through a rigorous experimental evaluation, we directly contradict conclusions from related work. Notably, we observed that the performance is not limited by the number of used storage targets when they are faster than the used network links. In that case, the most important aspect is the load balance between the different servers. On the other hand, when storage targets and not the network limit performance, the more targets, the better. We also show that sharing I/O targets does not lead to performance degradation.

We published the results of this study in [15], where we give several recommendations that can significantly improve the overall write performance of BeeGFS deployments and provide valuable information for future work on storage target allocation and stripe count tuning.

8.12 I/O performance of multiscale finite element simulations on HPC environments

Participants: Francieli Zanon-Boito, Louis Peyrondet, Luan Gouveia Lima.

We study the I/O performance of multiscale simulations. More specifically, we consider the class of simulations based on multiscale finite element (FE) methods. From this study, we presented in [14] MSLIO a code that mimics the I/O behaviour of the simulations (also called an I/O Kernel). Such an I/O kernel is useful for HPC research, as it can be executed more easily and efficiently than the full simulations when researchers are only interested in the I/O load. We validate MSLIO by comparing it to the I/O performance of an actual simulation, and we illustrate the usefulness of the I/O kernel by using it to propose and evaluate different access pattern adaptations aiming at improving the I/O performance. As a case study, we used the family of Multiscale Hybrid Mixed (MHM) FE methods, which have been used for modelling and simulating several different multiscale physical phenomena.

8.13 Implementation of a Weighted Fair Queuing (WFQ) I/O request scheduler

Participants: Mayer Alessa, Francieli Zanon-Boito, Luan Gouveia Lima.

The IO-Sets approach proposed in [28] requires a fine control of the amount of I/O bandwidth concurrent jobs will receive during a concurrent I/O operation. However, such control is not supported by current parallel file systems. This work represented our first steps toward that implementation.

As described in [33], after a careful study, we concluded that the Weighted Fair Queuing (WFQ) algorithm would meet IO-sets demands. Then, we implement the algorithm into AGIOS, an I/O-scheduling library meant to be easily integrated into existing parallel file systems and I/O nodes. The implementation is available in the last version of AGIOS.

8.14 Dynamic Scheduling Strategies for Firm Semi-Periodic Real-Time Tasks.

Participants: Guillaume Pallez.

In this project [32, 10] we have studied strategies to schedule firm semi-periodic real-time tasks. Jobs are released periodically and have the same relative deadline. Job execution times obey an arbitrary probability distribution and can take either bounded or unbounded values. We have proposed and discussed several optimization criteria to study such scenarios. We have introduced new control parameters to dynamically decide whether to interrupt a job at any given time.

8.15 Doing better for jobs that failed: node stealing from a batch scheduler's perspective.

Participants: Guillaume Pallez.

After a machine failure, batch schedulers typically re-schedule the job that failed with a high priority. This is fair for the failed job but still requires that job to re-enter the submission queue and to wait for enough resources to become available. The waiting time can be very long when the job is large and the platform highly loaded, as is the case with typical HPC platforms. We studied [31] another strategy: when a job J fails, if no platform node is available, we steal one node from another job J_0 , and use it to continue the execution of J despite the failure. In this work, we give a detailed assessment of this node stealing strategy using traces from the Mira supercomputer at Argonne National Laboratory. The main conclusion is that node stealing improves the utilization of the platform and dramatically reduces the flow of large jobs, at the price of a slight increases the flow of small jobs.

8.16 Optimal checkpointing strategies for iterative applications

Participants: Guillaume Pallez.

This work provides an optimal checkpointing strategy to protect iterative applications from fail-stop errors. [9] We consider a general framework, where the application repeats the same execution pattern by executing consecutive iterations, and where each iteration is composed of several tasks. These tasks have different execution lengths and different checkpoint costs. Some naive and Young/Daly strategies are suboptimal. Our main contribution is to show that the optimal checkpoint strategy is globally periodic, and to design a dynamic programming algorithm that computes the optimal checkpointing pattern. This pattern may well checkpoint many different tasks, and this across many different iterations. We show through simulations, both from synthetic and real-life application scenarios, that the optimal strategy outperforms the naive and Young/Daly strategies.

More generally, we have presented an overview of Young/Daly strategies [13].

8.17 Process mapping on any topology with TopoMatch

Participants: Emmanuel Jeannot.

Process mapping (or process placement) is a useful algorithmic technique to optimize the way applications are launched and executed onto a parallel machine. By taking into account the topology of the machine and the affinity between the processes, process mapping helps reducing the communication time of the whole parallel application. In [11], we presented TopoMatch, a generic and versatile library and algorithm to address the process placement problem. We describe its features and characteristics, and we report different use-cases that benefit from this tool. We also studied the impact of different factors: sparsity of the input affinity matrix, trade-off between the speed and the quality of the mapping procedure as well as the impact of the uncertainty (noise) onto the input.

8.18 Mapping circuits onto multi-FPGA platforms

Participants: Julien Rodriguez, François Pellegrini.

The work of Julien Rodriguez concerns the placement of digital circuits onto a multi-FPGA platform, in the context of a PhD directed by François Pellegrini, in collaboration with François Galea and Lilia Zaourar at CEA Saclay. Its aim is to design and implement mapping algorithms that do not minimize the cut, as it is the case in most partitioning toolboxes, but the length of the longest part between sets of vertices. This metric strongly correlates to the critical path that signals have to traverse during a circuit compute cycle, hence to the maximum frequency at which a circuit can operate.

This problem has been modeled using a dedicated hypergraph model, in the form of red-black Directed Acyclic Hypergraphs (DAHs). Subsequently, a graph partitioning framework has been designed and implemented, consisting of initial partitioning and refinement algorithms. This work, which has been presented in [21], results in the development of a dedicated software called Raisin (see Section 7.1.7).

A side work consists in an exploration of quantum computing approaches to solve the hypergraph partitioning problem on two kinds of quantum computers: adiabatic and gate-based. This preliminary work [20] allowed to start a collaboration with members of the quantum team in CEA LIST, on solving higher order binary optimization problems.

9 Bilateral contracts and grants with industry

9.1 Bilateral contracts with industry

CEA

Participants: Alexandre Denis, Clément Gavoille, Brice Goglin, Emmanuel Jeannot, François Pellegrini, Florian Reynier, Julien Rodriguez.

- CEA/DAM granted the funding of the PhD thesis of Florian Reynier on non-blocking MPI collectives.
- CEA/LIST (Saclay) granted the funding of the PhD thesis of Julien Rodriguez on the mapping of digital circuits onto multi-FPGA platforms.
- CEA/DAM granted the funding of the PhD thesis of Clément Gavoille on the perdiction of performance on future ARM HPC platforms.

ATOS

Participants: Emmanuel Jeannot, Guillaume Mercier, Richard Sartori.

- ATOS/Bull is funding the CIFRE PhD Thesis of Richard Sartori on the determination of optimal parameters for MPI applications deployment on parallel architectures

9.2 Bilateral Grants with Industry

Intel

Participants: Brice Goglin.

Intel granted $30k and provided information about future many-core platforms and memory architectures to ease the design and development of the HWLOC software with early support for next generation hardware.

10 Partnerships and cooperations

10.1 International initiatives

10.1.1 Inria associate team not involved in an IIL or an international program

HPCProSol

Title: Next-generation HPC PROblems and SOLutions

Duration: from 2021 to 2023.

Coordinator: Francieli Zanon-Boito and Carla Osthoff (osthoff@lncc.br)

Partner: Laboratório Nacional de Computação Científica Petrópolis (Brésil)

Summary: In the context of the convergence of HPC and big data, the notion of scientific application is evolving into a scientific workflow, composed of cpu-intensive and data-intensive tasks. In this new scenario, the already challenging problems of efficiently managing resources are expected to become worse and should be tackled by better scheduling at application and system levels, and consider applications' characteristics to avoid issues such as interference. We propose a collaboration between the TADaaM Inria team and the LNCC to study and characterize the new HPC workload, represented by a set of scientific applications that are important to the LNCC. This will guide the proposal of monitoring and profiling techniques for applications, and the design of new coordination mechanisms to arbitrate resources in HPC environments.

10.1.2 Participation in other International Programs

JLESC

Participants: Alexis Bandet, Luan Teylo Gouveia Lima, Emmanuel Jeannot, Philippe Swartvagher.

JLESC Joint-Lab on Extreme Scale Computing

- Coordinators: Franck Cappello (general) and Yves Robert (Inria coordinator).
- Other partners: Argonne National Lab, University of Urbanna Champaign (NCSA), Tokyo Riken, Jülich Supercomputing Center, Barcelona Supercomputing Center (BSC).
- Abstract: The purpose of the Joint Laboratory for Extreme Scale Computing (JLESC) is to be an international, virtual organization whose goal is to enhance the ability of member organizations and investigators to make the bridge between Petascale and Extreme computing. The founding partners of the JLESC are INRIA and UIUC. Further members are ANL, BSC, JSC and RIKEN-AICS.
- Duration: since 2014.
- Since 2022, Alexis Bandet is the French student ambassador for JLESC.

10.2 International research visitors

10.2.1 Visits of international scientists

Other international visits to the team

- Bruno Bodin, Yale-NUS, Singapour, visited the team the week of november 2, 2022.
- Jannis Klinkenberg, RWTH Aachen University, visited the team from June to August, 2022.
- Carla Osthoff and Kary Ocana, from the LNCC (Brazil) visited the team from November 2 to 10 2022.

Jannis Klinkenberg

Status: PhD student

Institution of origin: RWTH Aachen University

Country: Germany

Dates: June to August, 2022 (3 months).

Context of the visit: research exchange in the framework of the H2M project.

Mobility program/type of mobility: research stay and common work with members of the team.

Bruno Bodin

Status: Assistant Professor

Institution of origin: YaleNUS College

Country: Singapore

Dates: November 2 – November 5, 2022

Context of the visit: research exchange and discussion for possible candidature.

Mobility program/type of mobility: research stay and lecture.

Carla Osthoff and Kary Ocana

Status researchers

Institution of origin: LNCC

Country: Brazil

Dates: November 2 – November 10, 2022

Context of the visit: collaboration in the HPCProSol joint team.

Mobility program/type of mobility: research stay and lecture.

10.3 European initiatives

10.3.1 H2020 projects

ADMIRE

Participants: Alexis Bandet, Clément Barthélémy, Luan Teylo Gouveia Lima, Emmanuel Jeannot, Guillaume Pallez, Francieli Zanon-Boito.

ADMIRE project on cordis.europa.eu

Title: Adaptive multi-tier intelligent data manager for Exascale

Duration: From April 1, 2021 to March 31, 2024

Partners:

- DataDirect Networks France (DDN Storage), France
- Institut National de Recherche en Informatique et Automatique (Inria), France
- Johannes Gutenberg-Universitat Mainz, Germany
- Kungliga Tekniska Hoegskolan (KTH), Sweden
- Forschungszentrum Julich GMBH (FZJ), Germany
- Universita degli Study di Napoli Parthenope (UNIPARTH), Italy
- Universita degli Studi di Torino (UNITO), Italy
- Instytut Chemii Bioorganicznej Polskiej Akademii Nauk, Poland
- Universita di Pisa (UNIPI), Italy
- E4 Computer Engineering SPA (E4), Italy
- Université de Bordeaux (UBx), France
- Universita degli Studi di Milano (UMIL), Italy
- Paratools SAS (Paratools SAS), France
- Technische Universitat Darmstadt, Germany
- Max-Planck-Gesellschaft zur Forderung der Wissenschaften EV (MPG), Germany
- CINECA Consorzio Interuniversitario (CINECA), Italy
- Universidad Carlos III de Madrid (UC3M), Spain
- Barcelona Supercomputing Center Centro Nacional de Supercomputacion (BSC CNS), Spain
- Consorzio Interuniversitario Nazionale per l'Informatica (CINI), Italy

Inria contact: Emmanuel JEANNOT

Coordinator: Jesus Carretero, UC3M, Spain.

Summary: The growing need to process extremely large data sets is one of the main drivers for building exascale HPC systems today. However, the flat storage hierarchies found in classic HPC architectures no longer satisfy the performance requirements of data-processing applications. Uncoordinated file access in combination with limited bandwidth make the centralised back-end parallel file system a serious bottleneck. At the same time, emerging multi-tier storage hierarchies come with the potential to remove this barrier. But maximising performance still requires careful control to avoid congestion and balance computational with storage performance. Unfortunately, appropriate interfaces and policies for managing such an enhanced I/O stack are still lacking.

The main objective of the ADMIRE project is to establish this control by creating an active I/O stack that dynamically adjusts computation and storage requirements through intelligent global coordination, malleability of computation and I/O, and the scheduling of storage resources along

> all levels of the storage hierarchy. To achieve this, we will develop a software-defined framework based on the principles of scalable monitoring and control, separated control and data paths, and the orchestration of key system components and applications through embedded control points.
>
> Our software-only solution will allow the throughput of HPC systems and the performance of individual applications to be substantially increased – and consequently energy consumption to be decreased – by taking advantage of fast and power-efficient node-local storage tiers using novel, European ad-hoc storage systems and in-transit/in-situ processing facilities. Furthermore, our enhanced I/O stack will offer quality-of-service (QoS) and resilience. An integrated and operational prototype will be validated with several use cases from various domains, including climate/weather, life sciences, physics, remote sensing, and deep learning.

Emmanuel Jeannot is the leader of WP6, concerned with the design and the implementation of the "intelligent controller", an instantiation of the service-layer envisioned at the beginning of the project. Clément Barthélémy has been hired in August 2021 as a research engineer to work specifically on this task. He has taken part in different ADMIRE activities, meetings and workshops, remotely and in-person. The controller has been designed as a client-server system, communicating via RPC. A client library has been implemented and tested with different ADMIRE modules, including FlexMPI, a malleable MPI implementation developed by partner UC3M. An interface to the Slurm resource manager has also been added, easing the dynamic addition or removal of nodes to a running job, a feature that can be used with, for example, MPI dynamic processes. As part of the global orchestration role of the work package, the Slurm command line interface has been augmented to get information needed to deploy the *ad-hoc* file systems under the responsibility of WP2. Finally, preliminary work has been done to check the feasibility of implementing I/O scheduling at the backend file system level.

Textarossa

Participants: Brice Goglin.

- Textarossa: Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale
- Program: H2020 EuroHPC
- Grant Agreement number: 956831 — TEXTAROSSA — H2020-JTI-EuroHPC-2019-1
- 2021-2024
- Partners: Fraunhofer Gesellschaft zur Foerderung der Angewandten Forshung E.V.; Consorzio Interuniversitario Nazionale per l'Informatica; Institut National de Recherche en Informatique et Automatique; Bull SAS; E4 Computer Engineering SPA; Barcelona Supercomputing Center; Instytut Chemii Bioorganicznej Polskiej; Istituto Nazionale di Fisica Nucleare; Consiglio Nazionale delle Ricerche; In Quattro SRL.
- To achieve high performance and high energy efficiency on near-future exascale computing systems, a technology gap needs to be bridged: increase efficiency of computation with extreme efficiency in HW and new arithmetics, as well as providing methods and tools for seamless integration of reconfigurable accelerators in heterogeneous HPC multi-node platforms. TEXTAROSSA aims at tackling this gap through applying a co-design approach to heterogeneous HPC solutions, supported by the integration and extension of IPs, programming models and tools derived from European research projects, led by TEXTAROSSA partners [6].
- Website: textarossa.eu
- TADaaM funding: 200k€

EUPEX

Participants: Brice Goglin.

- EUPEX: European Pilot for Exascale
- Program: H2020 EuroHPC
- Grant Agreement number: 101033975 – H2020-JTI-EuroHPC-2020-01
- 2022-2025
- 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
- TADaaM funding: 150k€

10.3.2 Other european programs/initiatives

ANR-DFG H2M

Participants: Clément Foyer, Brice Goglin, Emmanuel Jeannot.

- Title: Heuristics for Heterogeneous Memory
- Website: h2m.gitlabpages.inria.fr
- AAPG ANR 2020, 2021 - 2023 (36 months)
- Coordinator: Christian Terboven (German coordinator) and Brice Goglin (French coordinator).
- Abstract: H2M is a ANR-DFG project between the TADaaM team and the HPC Group at RWTH Aachen University (Germany) from 2021 to 2023. The overall goal is to leverage HWLOC's knowledge of heterogeneous memory up to programming languages such as OpenMP to ease the allocations of data sets in the appropriate target memories.

10.4 National initiatives

ANR DASH

Participants: Luan Gouveia Lima, Emmanuel Jeannot, Guillaume Pallez, Nicolas Vidal.

- Title: Data-Aware Scheduling at Higher scale

- Website: project.inria.fr/dash
- AP générique JCJC 2017, 03/2018 - 07/2023 (48 months, extended due to Covid)

- Coordinator: Guillaume PALLEZ (Tadaam)
- Abstract: This project focuses on the effecient execution of I/O for High-Performance applications. The idea is to take into account some knowledge on the behavior of the different I/O steps to compute efficient schedules, and to update them dynamically with the online information.

ANR Solharis

Participants: Alexandre Denis, Guillaume Pallez, Philippe Swartvagher, Nicolas Vidal.

- Title: SOLvers for Heterogeneous Architectures over Runtime systems, Investigating Scalability
- Website: www.irit.fr/solharis
- AAPG ANR 2019, 2019 - 2023 (48 months)

- Coordinator: Alfredo BUTTARI (IRIT-INPT)
- Abstract: The Solharis project aims at producing scalable methods for the solution of large sparse linear systems on large heterogeneous supercomputers, using the STARPU runtime system, and to address the scalability issues both in runtime systems and in solvers.

AEX: Repas

Participants: Robin Boezennec, Guillaume Pallez.

- Title: REPAS: New Portrayal of HPC Applications
- Inria Exploratory program 2022

- Coordinator: Guillaume PALLEZ (Tadaam)
- Abstract: What is the right way to represent an application in order to run it on a highly parallel (typically exascale) machine? The idea of project is to completely review the models used in the development scheduling algorithms and software solutions to take into account the real needs of new users of HPC platforms.

11 Dissemination

11.1 Promoting scientific activities

11.1.1 Scientific events: organisation

General chair, scientific chair

- Emmanuel Jeannot and Guillaume Pallez were the general chair of the ICPP'22 conference.
- Emmanuel Jeannot was general co-chair of The 14th BenchCouncil International Symposium On Benchmarking, Measuring And Optimizing (Bench 2022).

Member of the organizing committees

- Clément Foyer was the proceeding chair of the ICPP'22 conference.
- Brice Goglin was the finance chair of the ICPP'22 conference.
- Emmanuel Jeannot and Guillaume Pallez are members the ICPP steering committee.
- Guillaume Pallez was the IEEE Cluster'23 finance chair (executive committee).

11.1.2 Scientific events: selection

Chair of conference program committees

- Emmanuel Jeannot was chair of the 2022 RADR Workshop on Resource Arbitration for Dynamic Runtimes (in conjunction with IPDPS).
- Guillaume Pallez was the SC'22 Awards vice-chair and Cluster'22 Panel co-chair.

Member of the conference program committees

- Alexandre Denis was a member of the HiPC 2022 program committee.
- Brice Goglin was a member of the following program committees: Hot Interconnect 29, HeteroPar, ROME@ARCS.
- Emmanuel Jeannot was a member of the following program committees: Cluster 2022, HPCMALL 2022.
- Guillaume Mercier was a member of the following program committees: EuroMPI/USA 2022, CCGrid 2022.
- Guillaume Pallez was a member of the following program committees: IPDPS'23.
- Francieli Zanon-Boito was a member of the following program committees: DRBSD 2022 (SC workshop) and ISC 2022 Project Posters.

Reviewer

- Alexis Bandet was an external reviewer for: IPDPS 2023, SBAC-PAD 2023.
- Luan Teylo Gouveia Lima was an external reviewer for: IPDPS 2023, SBAC-PAD 2022, FGCS and JPDC.
- Philippe Swartvagher was an external reviewer for: IPDPS 2023 and Cluster 2022.

11.1.3 Journal

Member of the editorial boards

- Emmanuel Jeannot is member of the editorial board of the Journal of Parallel Emergent & Distributed Systems.

Reviewer - reviewing activities

- Emmanuel Jeannot has reviewed a submission for: ACM Computing Surveys; Concurrency and Computation: Practice and Experience.

11.1.4 Invited talks

- Emmanuel Jeannot was invited to give a talk: “Overlapping computation and communication: a runtime-system point of view” at CCDSC’22.
- Guillaume Pallez was invited to give a talk: “Model Accuracy in HPC System Software Algorithmic” at CCDSC’22
- François Pellegrini was invited to give a talk with Cédric Brun: “Software: between trust and transparency, what acceptability?” at the CNRS INS2I Software science: from idea to binary event.
- François Pellegrini participated in a roundtable on the social acceptability of automated processing, during the conference on “*Public decision support algorithms*” organized by the Société informatique de France.
- Philippe Swartvagher was invited to give a talk: “Interactions between Task-Based Runtime Systems and the Communication Layer” in the Minisymposium “Task-Based Programming for Distributed Memory Systems” at SIAM-PP 22.
- Philippe Swartvagher was invited to give a talk: "Using Guix for scientific, reproducible, and publishable experiments" at the workshop for the 10 years of Guix.
- Francieli Zanon-Boito was invited to give a talk: “The role of storage target allocation in applications’ I/O performance with BeeGFS” at the CHPC National Conference (South Africa).
- Francieli Zanon-Boito was invited to give a talk: “IO-SETS: Simple and efficient approaches for I/O” at the Per3S workshop.

11.1.5 Scientific expertise

- Brice Goglin was a member of the Khronos OpenCL Advisory Panel as well as the oneAPI Level Zero Technical Advisory Board.
- François Pellegrini is the vice-president of CNIL, the French data protection authority.
- François Pellegrini is a member of the ERC pool of experts on ethics, and participated in several ERC project evaluation committees (e.g., “Horizon 2020 PoC”, “Starting Grants 2021”, “Consolidator Grants 2021”, “Advanced Grants 2021”).

11.1.6 Standardization Activities

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 that now encompasses both physical and virtual topologies and participates also in serveral other Working Groups. He’s also an editor of the MPI Standard. For this year, two proposals are currently in the voting process :

- The first proposal is an expansion/enhancement of our hardware communicators proposal. We now propose to use process set names to guide the splitting of communicators. The benefits are twofold: first, new types of resources can be used. For instance, shared memory can be considered not necessarily as a hardware resource since it can be implemented through software. It therefore falls into a grey area between hardware and software. We want to be able to provide a flexible mechanism that will allow such support. Second, since some process set names can be that of hardware resource, we then propose a unifying mechanism to leverage hardware information at the MPI application level. Without such new feature, there would be two distinct mechanisms in the standard which can be confusing to the user.

- The second proposal in a companion feature to the one we already introduced. The original plan was to introduce them both at the same time but the interface needed more work to be accepted. This work in now over and we are confident that this will be voted in. Basically, hardware resource type names can be used to guide the splitting of hardware based on hardware criteria. Such names are implementation-defined and no mechanism currently exists to query such names in a standard fashion. Our proposal shall fill this gap.

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

11.1.7 Tutorials

- Philippe Swartvagher gave a talk at the "Midi de la bidouille" entitled *"Auto-hébergement : quoi, pourquoi, comment ?"*.

11.1.8 Research administration

- Emmanuel Jeannot is head of science of the Inria Bordeaux research center.
- Emmanuel Jeannot is a member and Guillaume Pallez is an elected member of the Inria evaluation committee.
- Brice Goglin is in charge of the computing infrastructures of the Inria Bordeaux research center.
- François Pellegrini is a co-pilot of the *Source code and Software* college within the Committee for Open Science (CoSO) of the French Ministry of Higher Education and Research.

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, 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 20 PhD students (in informatics, law, physics, medicine, etc.) of Université de Bordeaux.
- François Pellegrini participated in a training session on "*Information science, digital technologies and law*" for the continuous education of magistrates, *École nationale de la magistrature* (National School for Magistrates), Paris.

11.2.2 Supervision

- PhD in progress: Robin Boezennec, Vers de nouvelles représentations des applications hpc. Started in September 2022, co-advised with Datamove (Grenoble). Inria Advisors: Guillaume Pallez and Fanny DUFOSSÉ.
- PhD in progress: Clément Gavoille, the prediction of performance on future ARM HPC platforms. Started in January 2021, co-advised with CEA and ARM. Inria Advisors: Brice Goglin and Emmanuel Jeannot.
- PhD in progress: Julien Rodriguez, Circuit mapping onto multi-FPGA platforms, started in October 2020. Advisors: François Pellegrini, François GALEA and Lilia ZAOURAR.

- PhD in progress: Richard Sartori, Determination of optimal parameters for MPI applications deployment on parallel architectures. Started in April 2021, co-advised with ATOS/Bull in Grenoble. Inria Advisors: Guillaume Mercier and Emmanuel Jeannot.
- PhD in progress: Alexis Bandet, I/O characterization and monitoring of the new generation of HPC applications. Started in October 2021. Advisors: Francieli Zanon-Boito and Guillaume Pallez.
- PhD finished: Nicolas Vidal, IO scheduling strategies, started in October 2018 and defended in January 2022. Advisors: Guillaume Pallez and Emmanuel Jeannot.
- PhD finished: Philippe Swartvagher, Interactions at large scale between high performance communication libraries and task-based runtime, started in October 2019 and defended in November 2022. Advisors: Alexandre Denis and Emmanuel Jeannot.
- PhD finished: Florian Reynier, Task-based communication progression, started in January 2019 and defended in June 2022. Advisors: Alexandre Denis and Emmanuel Jeannot.

11.2.3 Juries

- Brice Goglin was reviewer for the PhD defense of John Gliksberg (Atos, University of Paris-Saclay and University of La Mancha).
- Brice Goglin was the president of the PhD defense jury of Celia Tassadit Ait Kaci (Atos and Inria).
- Emmanuel Jeannot was president of the PhD defense jury of Van Man Nguyen (Inria and CEA)
- Emmanuel Jeannot was reviewer of the PhD thesis and president of the defense jury of Hatem Elshazly (Univ. Polytecnica de Catalunya, Spain)
- Emmanuel Jeannot was reviewer of the PhD thesis of Francesco Gava (Insa Rouen)
- François Pellegrini was a member of the PhD defense jury of Marie Bastian (Univ. Nanterre).
- François Pellegrini was president of the PhD defense jury of Esragul Korkmaz (Univ. Bordeaux).
- Francieli Zanon-Boito was a member of the PhD defense jury of Yishu Du (ENS Lyon).
- Brice Goglin was a member of the hiring committee for ATER positions at Université de Bordeaux.
- Francieli Zanon-Boito was a member of the hiring committee for associate professor at Université de Bordeaux.
- Guillaume Pallez was a member of the hiring committee for CRCN positions at Inria Rennes and Inria Nancy. He also was a member of the CRHC promotion jury.

11.3 Popularization

- Brice Goglin gave talks about research in computer science and high-performance computing to high-school student as part of the *Fête de la Science* event, *Chiche* programme, and *Cordée de la Réussite* programme.
- Emmanuel Jeannot gave a lecture on NFTs and Blockchain at the Léognan mediathèque.
- François Pellegrini participated in two debates with students of high schools in the Bordeaux region, on facial recognition technologies and on personal data.
- François Pellegrini participated in several roundtable panels at the *Rencontres de l'Esprit Critique* event, in Toulouse, on "*law and zetetics: which interactions?*", "*Algo-literacy: for a chosen digital culture and less disinformation*", and "*Our personal data on the Internet in the service of political operations? The Cambridge Analytica affair*".

11.3.1 Articles and contents

- Brice Goglin's work in collaboration with AMD and HPE/Cray to help break the Exaflop barrier was described in an Inria news article[5].
- François Pellegrini was interviewed by the *Journal du CNRS* on personal data and profiling[6].
- François Pellegrini made an appearance in an episode of Sébastien Canévet's "*Vous avez le droit*" YouTube channel[7].
- François Pellegrini participated in two documentaries which are now freely available: "*Responsables du Numérique*"[8] (on sustainable digital policies) and "*LoL - Logiciel libre, une affaire sérieuse*"[9] (on free/libre software).
- François Pellegrini participated in the writing of two booklets on open science for PhD candidates: one on "*Codes and software*"[10] (as first author), and the other on "*Questions on open science*"[11]. The booklet on "*Codes and software*" is currently being translated in English.

11.3.2 Interventions

- Emmanuel Jeannot gave a talk at "Unithé ou Café" on NFTs and Blockchain with Damien Robert at the Bordeaux Inria center.
- Luan Teylo Gouveia Lima gave a talk about research in computer science and high-performance computing to undergraduate students from the Federal University of Mato Grosso in Brazil.

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] 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.

[3] 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. URL: https://hal.inria.fr/hal-01924951.

[4] 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. URL: https://hal.inria.fr/hal-03010676.

[5] 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. URL: https://hal.inria.fr/hal-01937123.

[5] link here
[6] link here
[7] link here
[8] link here
[9] link here
[10] link here
[11] link here

12.2 Publications of the year

International journals

[6] G. Agosta, M. Aldinucci, C. Alvarez, R. Ammendola, Y. Arfat, O. Beaumont, M. Bernaschi, A. Biagioni, T. Boccali, B. Bramas et al. 'Towards EXtreme scale technologies and accelerators for euROhpc hw/Sw supercomputing applications for exascale: The TEXTAROSSA approach'. In: *Microprocessors and Microsystems: Embedded Hardware Design* 95 (Nov. 2022), p. 104679. DOI: 10.1016/j.micpro.2022.104679. URL: https://hal.inria.fr/hal-03936864.

[7] A. Denis, J. Jaeger, E. Jeannot and F. Reynier. 'A methodology for assessing computation/communication overlap of MPI nonblocking collectives'. In: *Concurrency and Computation: Practice and Experience* 34.22 (10th Oct. 2022). DOI: 10.1002/cpe.7168. URL: https://hal.inria.fr/hal-03922777.

[8] A. Denis, E. Jeannot and P. Swartvagher. 'Predicting Performance of Communications and Computations under Memory Contention in Distributed HPC Systems'. In: *International Journal of Networking and Computing*. Special Issue on Workshop on Advances in Parallel and Distributed Computational Models 2022 13.1 (Jan. 2023), p. 30. URL: https://hal.inria.fr/hal-03871630.

[9] Y. Du, L. Marchal, G. Pallez and Y. Robert. 'Optimal Checkpointing Strategies for Iterative Applications'. In: *IEEE Transactions on Parallel and Distributed Systems* 33.3 (1st Mar. 2022), pp. 507–522. DOI: 10.1109/TPDS.2021.3099440. URL: https://hal.inria.fr/hal-03338278.

[10] Y. Gao, G. Pallez, Y. Robert and F. Vivien. 'Dynamic Scheduling Strategies for Firm Semi-Periodic Real-Time Tasks'. In: *IEEE Transactions on Computers* 72.1 (1st Jan. 2023), pp. 55–68. DOI: 10.1109/TC.2022.3208203. URL: https://hal.inria.fr/hal-03778357.

[11] E. Jeannot. 'Process mapping on any topology with TopoMatch'. In: *Journal of Parallel and Distributed Computing* 170 (Dec. 2022), pp. 39–52. DOI: 10.1016/j.jpdc.2022.08.002. URL: https://hal.inria.fr/hal-03780662.

[12] F. Pellegrini and E. Verdon. 'Personal data, the fuel of surveillance capitalism'. In: *L'Économie politique* 94 (May 2022), pp. 36–47. URL: https://hal.inria.fr/hal-03861709.

International peer-reviewed conferences

[13] A. Benoit, Y. Du, T. Herault, L. Marchal, G. Pallez, L. Perotin, Y. Robert, H. Sun and F. Vivien. 'Checkpointing à la Young/Daly: An Overview'. In: IC3 2022 - 2022 Fourteenth International Conference on Contemporary Computing. Noida, India: ACM, 4th Aug. 2022, pp. 701–710. DOI: 10.1145/3549206.3549328. URL: https://hal.inria.fr/hal-03830322.

[14] F. Boito, A. T. A. Gomes, L. Peyrondet and L. Teylo. 'I/O performance of multiscale finite element simulations on HPC environments'. In: WAMCA 2022 - 13th Workshop on Applications for Multi-Core Architectures. Bordeaux, France, 2nd Nov. 2022. URL: https://hal.inria.fr/hal-03808833.

[15] F. Boito, G. Pallez and L. Teylo. 'The role of storage target allocation in applications' I/O performance with BeeGFS'. In: CLUSTER 2022 - IEEE International Conference on Cluster Computing. Heidelberg, Germany, 6th Sept. 2022. URL: https://hal.inria.fr/hal-03753813.

[16] A. Denis, J. Jaeger, E. Jeannot and F. Reynier. 'One core dedicated to MPI nonblocking communication progression? A model to assess whether it is worth it'. In: International Symposium on Cluster, Cloud and Internet Computing (CCGRID). Taormina, Italy, 16th May 2022. URL: https://hal.inria.fr/hal-03695835.

[17] A. Denis, E. Jeannot and P. Swartvagher. 'Modeling Memory Contention between Communications and Computations in Distributed HPC Systems'. In: IPDPS - 2022 - IEEE International Parallel and Distributed Processing Symposium Workshops. Lyon / Virtual, France, 30th May 2022, p. 10. DOI: 10.1109/IPDPSW55747.2022.00086. URL: https://hal.inria.fr/hal-03682199.

[18] C. Gavoille, H. Taboada, P. Carribault, F. Dupros, B. Goglin and E. Jeannot. 'Relative performance projection on Arm architectures'. In: The 28th International Euro-Par Conference 2022. Glasgow, United Kingdom, 22nd Aug. 2022. URL: https://hal.inria.fr/hal-03887202.

[19] B. Goglin and A. Rubio Proaño. 'Using Performance Attributes for Managing Heterogeneous Memory in HPC Applications'. In: PDSEC 2022 - 23rd IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing, held in conjunction with the IPDPS 2022 - 36th IEEE International Parallel and Distributed Processing Symposium. Lyon / Virtual, France, 30th May 2022. URL: https://hal.inria.fr/hal-03599360.

[20] J. Rodriguez. 'Quantum algorithms for hypergraph bi-partitioning'. In: 23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision. Villeurbanne - Lyon, France, 23rd Feb. 2022. URL: https://hal.science/hal-03595234.

[21] J. Rodriguez, F. Galea, F. Pellegrini and L. Zaourar. 'Partitionnement d'un ensemble connexe d'hypergraphes orientés sans cycle avec minimisation de chemin'. In: 23ème congrès annuel de la Société Française de Recherche Opérationnelle et d'Aide à la Décision. Villeurbanne - Lyon, France, 23rd Feb. 2022. URL: https://hal.science/hal-03596218.

Conferences without proceedings

[22] F. Boito, G. Pallez, L. Teylo and N. Vidal. 'IO-SETS: Simple and efficient approaches for I/O bandwidth management'. In: COMPAS 2022 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Amiens, France, 5th July 2022. URL: https://hal.inria.fr/hal-03773392.

[23] P. Swartvagher. 'Interactions entre calculs et communications au sein des systèmes HPC distribués : évaluation et modélisation.' In: COMPAS 2022 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Amiens, France, 5th July 2022. URL: https://hal.inria.fr/hal-03719612.

Doctoral dissertations and habilitation theses

[24] F. Reynier. 'A study on progression of MPI communications using dedicated resources'. Université de Bordeaux, June 2022.

[25] P. Swartvagher. 'On the Interactions between HPC Task-based Runtime Systems and Communication Libraries'. Université de Bordeaux, 29th Nov. 2022. URL: https://theses.hal.science/tel-03989856.

[26] N. Vidal. 'Data-Aware Scheduling at Higher scale'. Université de Bordeaux, 31st Jan. 2022. URL: https://theses.hal.science/tel-03892821.

Reports & preprints

[27] E. Agullo, O. Coulaud, A. Denis, M. Faverge, A. Franc, J.-M. Frigerio, N. Furmento, A. Guilbaud, E. Jeannot, R. Peressoni, F. Pruvost and S. Thibault. *Task-based randomized singular value decomposition and multidimensional scaling*. RR-9482. Inria Bordeaux - Sud Ouest; Inrae - BioGeCo, 9th Sept. 2022, p. 37. URL: https://hal.inria.fr/hal-03773985.

[28] F. Boito, G. Pallez, L. Teylo and N. Vidal. *IO-SETS: Simple and efficient approaches for I/O bandwidth management*. 5th May 2022. URL: https://hal.inria.fr/hal-03648225.

[29] P.-A. Bouttier, L. Courtès, Y. Dupont, M. Felšöci, F. Gruber, K. Hinsen, A. Isaac, P. Prins, P. Swartvagher, S. Tournier and R. Wurmus. *Guix-HPC Activity Report 2020-2021: Reproducible software deployment for high-performance computing*. Inria Bordeaux - Sud-Ouest; Université Grenoble - Alpes; Université Paris, 3rd Feb. 2022. URL: https://hal.inria.fr/hal-03565692.

[30] A. Denis, E. Jeannot and P. Swartvagher. *Modeling Memory Contention between Communications and Computations in Distributed HPC Systems (Extended Version)*. RR-9451. INRIA Bordeaux, équipe TADAAM, 10th Feb. 2022, p. 34. URL: https://hal.inria.fr/hal-03564751.

[31] Y. Du, L. Marchal, G. Pallez and Y. Robert. *Doing better for jobs that failed: node stealing from a batch scheduler's perspective*. 15th Apr. 2022. URL: https://hal.inria.fr/hal-03643403.

[32] Y. Gao, G. Pallez, Y. Robert and F. Vivien. *Scheduling Strategies for Overloaded Real-Time Systems*. RR-9455. Inria - Research Centre Grenoble – Rhône-Alpes, Feb. 2022, pp. 1–48. URL: https://hal.inria.fr/hal-03580853.

[33] A. Mayer, L. Teylo and F. Boito. *Implementation and Test of a Weighted Fair Queuing (WFQ) I/O Request Scheduler*. RR-9480. Inria, 23rd Aug. 2022, p. 12. URL: https://hal.inria.fr/hal-03758890.

Other scientific publications

[34] J. Klinkenberg, A. Kozhokanova, C. Terboven, C. Foyer, B. Goglin and E. Jeannot. 'H2M: Towards Heuristics for Heterogeneous Memory'. In: IEEE Cluster 2022 - 2022 IEEE International Conference on Cluster Computing. Heidelberg, Germany, 5th Sept. 2022. URL: https://hal.inria.fr/hal-03886110.

[35] C. Mercier. 'Gestion de l'énergie sur la plate-forme de calcul scientifique PlaFRIM'. Université de bordeaux, 31st Aug. 2022. URL: https://hal.inria.fr/hal-03770831.

12.3 Cited publications

[36] A. Denis, E. Jeannot and P. Swartvagher. 'Interferences between Communications and Computations in Distributed HPC Systems'. In: *ICPP 2021 - 50th International Conference on Parallel Processing*. Chicago / Virtual, United States, Aug. 2021, p. 11. DOI: 10.1145/3472456.3473516. URL: https://hal.inria.fr/hal-03290121.

[37] P. Swartvagher. 'Interactions entre calculs et communications au sein des systèmes HPC distribués'. In: *COMPAS 2021 - Conférence francophone d'informatique en Parallélisme, Architecture et Système*. Lyon, France, July 2021. URL: https://hal.inria.fr/hal-03290074.

[38] P. Swartvagher. *Interferences between Communications and Computations in Distributed HPC Systems*. Journée de l'École Doctorale Mathématiques et Informatique. Poster. May 2021. URL: https://hal.inria.fr/hal-03292004.

[39] P. Swartvagher. *Interferences between Communications and Computations in Distributed HPC Systems*. Euro-Par - 27th International European Conference on Parallel and Distributed Computing. Poster. Aug. 2021. URL: https://hal.inria.fr/hal-03333852.