RESEARCH CENTRE

Bordeaux - Sud-Ouest

IN PARTNERSHIP WITH:

Institut Polytechnique de Bordeaux,
Université de Bordeaux

2021

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]

Faculty Members

- Guillaume Mercier [Institut National Polytechnique de Bordeaux, Associate Professor, HDR]
- François Pellegrini [Univ de Bordeaux, Professor, HDR]
- Francieli Zanon-Boito [Univ de Bordeaux, Associate Professor]

Post-Doctoral Fellows

- Clément Foyer [Inria, from Jul 2021]
- Luan Gouveia Lima [Inria, from Apr 2021]

PhD Students

- Alexis Bandet [Inria, from Oct 2021]
- Clément Foyer [Inria, from May 2021 until Jun 2021]
- Clement Gavoille [CEA]
- Florian Reynier [CEA]
- Julien Rodriguez [CEA]
- Andres Xavier Rubio Proano [Inria, until Oct 2021]
- Richard Sartori [Bull, CIFRE, from Apr 2021]
- Philippe Swartvagher [Inria]
- Nicolas Vidal [Inria]

Technical Staff

- Clement Barthelemy [Inria, Engineer, from Aug 2021]
- Marc Fuentes [Inria, Engineer, 40%]

Interns and Apprentices

- Valentin Hoyet [Inria, Apprentice, until Sep 2021]
- Selmane Lebdaoui [Inria, from Jun 2021 until Sep 2021]
- Florian Lecomte [Inria, from Jun 2021 until Sep 2021]
- Pierre Pavia [Inria, until Apr 2021]
- Lisa Weisbecker [Inria, from May 2021 until Jun 2021]

Administrative Assistants

- Catherine Cattaert Megrat [Inria, from Aug 2021]
- Roweida Mansour El Handawi [Inria, until Sep 2021]

External Collaborators

- Pierre Ferenbach [Univ de Bordeaux, until Jun 2021]
- Elia Verdon [Univ de 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.

However, this year is special in many respects. Due to the lockdowns and regional and national border restrictions implemented to cope with the Covid-19 pandemic, many activities have had to be performed on-line, mostly from home. Consequently, the footprint of travel of team members, and notably airline travel, has been significantly reduced. Consequently, in spite of the highly increased use of digital communication systems (video-conferencing, messaging systems, etc.), the overall footprint of our research activities has been globally less than that of the previous year, and possibly than ever.

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

- The 4.0 revision of the MPI standard was released in 2021, which includes our proposal for extending communicators with topology and hardware information.
- Version 7.0 of SCOTCH has been released. This major release is the fruition of six years of development. Notably, it implements several new multi-threaded algorithms, both for the centralized SCOTCH library and the distributed PT-SCOTCH library. Also, as the SCOTCH project is turning 30, the project of creating a consortium to perpetuate the development of SCOTCH has been launched by Inria.

- By the time of the decommissioning of Inria Gforge, SCOTCH was its sixth most downloaded software, with 278k+ recorded downloads.
- François PELLEGRINI has been elected a vice-president of the French *Commission nationale de l'informatique et des libertés*.

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 (to be released mid-2021) 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

7.1.2 hwloc

Name: Hardware Locality

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

Functional Description: Hardware Locality (hwloc) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices. It builds a widely-portable abstraction of these resources and

exposes it to applications so as to help them adapt their behavior to the hardware characteristics. They may consult the hierarchy of resources, their attributes, and bind task or memory on them.

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

News of the Year: hwloc 2.1 brought support for modern multi-die processors and memory-side caches. It also enhanced memory locality in heterogeneous memory architecture (e.g. with non-volatile memory DIMMs). The visualization of many-core platforms was also improved by factorizing objects when many of them are identical.

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

7.1.4 PaMPA

Name: Parallel Mesh Partitioning and Adaptation

Keywords: Dynamic load balancing, Unstructured heterogeneous meshes, Parallel remeshing, Subdomain decomposition, Parallel numerical solvers

Scientific Description: PaMPA is a parallel library for handling, redistributing and remeshing unstructured meshes on distributed-memory architectures. PaMPA dramatically eases and speeds-up the development of parallel numerical solvers for compact schemes. It provides solver writers with a distributed mesh abstraction and an API to: - describe unstructured and possibly heterogeneous meshes, on the form of a graph of interconnected entities of different kinds (e.g. elements, faces, edges, nodes), - attach values to the mesh entities, - distribute such meshes across processing elements, with an overlap of variable width, - perform synchronous or asynchronous data exchanges of values across processing elements, - describe numerical schemes by means of iterators over mesh entities and their connected neighbors of a given kind, - redistribute meshes so as to balance computational load, - perform parallel dynamic remeshing, by applying adequately a user-provided sequential remesher to relevant areas of the distributed mesh.

PaMPA runs concurrently multiple sequential remeshing tasks to perform dynamic parallel remeshing and redistribution of very large unstructured meshes. E.g., it can remesh a tetrahedral mesh from 43Melements to more than 1Belements on 280 Broadwell processors in 20 minutes.

Functional Description: Parallel library for handling, redistributing and remeshing unstructured, heterogeneous meshes on distributed-memory architectures. PaMPA dramatically eases and speeds-up the development of parallel numerical solvers for compact schemes.

News of the Year: PaMPA has been used to remesh an industrial mesh of a helicopter turbine combustion chamber, up to more than 1 billion elements.

URL: http://project.inria.fr/pampa/

Contact: François Pellegrini

Participants: Cécile Dobrzynski, Cedric Lachat, François Pellegrini

Partners: Université de Bordeaux, CNRS, IPB

7.1.5 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.

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

Contact: Emmanuel Jeannot

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

Partners: Université de Bordeaux, CNRS, IPB

7.1.6 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: In 2021, Scotch switched to branch v7.0. Its major addition is the complete refactoring of the thread management subsystem, which now provides fully dynamic thread management. Many time-consuming algorithms have subsequently been parallelized, both in the Scotch centralized/sequential library and in the PT-Scotch distributed-memory version, which now implements MPI+thread hybrid parallelism.

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

7.1.7 H-Revolve

Keywords: Automatic differentiation, Gradients, Machine learning

Functional Description: This software provides several algorithms (Disk-Revolve, 1D-Revolve, Periodic-Disk-Revolve,...) computing the optimal checkpointing strategy when executing a adjoin chain with limited memory. The considered architecture has a level of limited memory that is free to access (writing and reading costs are negligible) and a level of unlimited memory with non-negligible access costs. The algorithms describe which data should be saved in the memory to minimize the number of re-computation during the execution.

URL: https://gitlab.inria.fr/adjoint-computation/H-Revolve

Publications: hal-02080706, hal-01654632, hal-01354902

Authors: Guillaume Pallez, Julien Herrmann

Contact: Guillaume Pallez

7.2 New platforms

7.2.1 PlaFRIM

Participants: Brice Goglin.

Name: Plateforme Fédérative pour la Recherche en Informatique et Mathématiques

Website: https://www.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 Using Bandwidth Throttling to Quantify Application Sensitivity to Heterogeneous Memory

Participants: Clément Foyer, Brice Goglin.

In the dawn of the exascale era, the memory management is getting increasingly harder but also of primary importance. The plurality of processing systems along with the emergence of heterogeneous memory systems require more care to be put into data placement. Yet, in order to test models, designs and heuristics for data placement, the programmer has to be able to access these expensive systems, or find a way to emulate them.

In [15], we propose to use the *Resource Control* features of the Linux kernel and x86 processors to add heterogeneity to a homogeneous memory system in order to evaluate the impact of different bandwidths on application performance. We define a new metric to evaluate the sensibility to bandwidth throttling as a way to investigate the benefits of using high-bandwidth memory (HBM) for any given application, without the need to access a platform offering this kind of memory. We evaluated 6 different well-known benchmarks with different sensitivity to bandwidth on a AMD platform, and validated our results on two Intel platforms with heterogeneous memory, Xeon Phi and Xeon with NVDIMMs. Although representing an idealized version of HBM, our method gives reliable insight of potential gains when using HBM.

Finally, we envision a design based on *Resource Control* using both bandwidth restriction and cache partitioning to simulate a more complex heterogeneous environment that allows for hand-picked data placement on emulated heterogeneous memory. We believe our approach can help develop new tools to test reliably new algorithms that improve data placement for heterogeneous memory systems.

8.2 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 [13, 25, 19, 24], 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 worked on a model to predict performances of computations and communications when they are executed in parallel. A paper about it will be submitted in the future.

8.3 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.

8.4 Profiles of upcoming HPC Applications and their Impact on Reservation Strategies

Participants: Brice Goglin, Guillaume Pallez.

With the expected convergence between HPC, BigData and AI, new applications with different profiles are coming to HPC infrastructures. We aim at better understanding the features and needs of these applications in order to be able to run them efficiently on HPC platforms. In [7] we proposed a bottom-up approach: we study thoroughly an emerging application, Spatially Localized Atlas Network Tiles (SLANT, originating from the neuroscience community) to understand its behavior. Based on these observations, we derive a generic, yet simple, application model (namely, a linear sequence of stochastic jobs). We expect this model to be representative for a large set of upcoming applications from emerging fields that start to require the computational power of HPC clusters without fitting the typical behavior of large-scale traditional applications. In a second step, we show how one can use this generic model in a scheduling framework. Specifically we consider the problem of making reservations (both time and memory) for an execution on an HPC platform based on the application expected resource requirements. We derive solutions using the model provided by the first step of this work. We experimentally show the robustness of the model, even with very few data points or using another application, to generate the model, and provide performance gains with regards to standard and more recent approaches used in the neuroscience community.

8.5 Multi-threaded centralized and distributed graph partitioning

Participants: François Pellegrini.

Based on the dynamic thread management framework which has been designed for the newest version of Scotch, new graph partitioning algorithms have been designed and implemented to benefit from the use of multiple threads. These algorithms concern several phases of a typical multi-level graph partitioning framework: the matching of graph vertices, the building of coarsened graphs deriving from a given matching (both for centralized and distributed graphs), and the progression of recursive graph bipartitioning methods.

For graph matching, a deterministic multi-threaded algorithm has been designed, to enable computational reproducibility while using threads. This algorithm, which implements a multi-threaded distance-2 coloring, is indeed highly scalable; yet, for graphs which do not have very small degrees, it is much more expensive than a sequential counterpart, up to several tens of threads.

For graph coarsening, while the centralized algorithm is quite straightforward, the distributed algorithm is more complex: it relies on the pre-computation of per-thread ranges of edges to be transferred, to

infer the location where coarsened edges arrays will be created for each thread. This algorithm takes advantage of the graph model implemented in SCOTCH, which allow for the management of "non-compact" graphs, in which the start and end indices of the adjacency arrays can be stored in two different arrays.

All these algorithms have been implemented in version 7.0 of SCOTCH. Altogether, they can bring up to ten-fold speed improvements when partitioning large graphs on a multiple-core, modern workstation.

8.6 Mapping circuits onto multi-FPGA platforms

Participants: Julien Rodriguez, François Pellegrini.

In the context of the PhD work of Julien RODRIGUEZ on the placement of digital circuits onto a multi-FPGA platform, prototype mapping algorithms have been designed and implemented, using a Python testbed. When mapping circuits to registers and combinatorial blocks located on separate FPGAs, these algorithms aim at preserving minimum critical path length rather than the number of wires cut.

8.7 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 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.

This work has been submitted for publication in CCGrid 2022.

8.8 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 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.

8.9 Using application grouping to improve I/O scheduling

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

Previous work has shown that, when multiple applications perform I/O phases at the same time, it is best to grant exclusive access to one of them at a time, which limits interference. That strategy is especially well adapted for a situation where applications have similar periods (they perform I/O phases with a similar frequency). However, when that is not the case, applications with shorter I/O phases present a higher stretch. We have been investigating a strategy where applications are grouped according to their I/O frequency. The idea is that applications from the same group should be executed one at a time, while different groups should share the available bandwidth. We are also working to determine a good priority-assigning policy.

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 [12], 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 2021 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 On the allocation of storage targets in parallel file systems

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

We have conducted experiments in PlaFRIM (using its parallel file system BeeGFS) aiming at providing a deeper understanding about I/O bandwidth and how it is affected by concurrent applications. We have 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 performance is limited by storage targets and not the network, then the more the merrier. We have not observed bandwidth congestion when sharing storage targets, which indicates that may not be an important resource to arbitrate between applications.

8.12 An International Survey on MPI Users

Participants: Emmanuel Jeannot.

The Message Passing Interface (MPI) plays a crucial part in the parallel computing ecosystem, a driving force behind many of the high-performance computing (HPC) successes. To maintain its relevance to the user community-and in particular to the growing HPC community at large-the MPI standard needs to identify and understand the MPI users' concerns and expectations, and adapt accordingly to continue to efficiently bridge the gap between users and hardware. A questionnaire survey was conducted using two online questionnaire frameworks and has gathered more than 850 answers from 42 countries since February 2019. Some of preceding surveys of MPI uses are questionnaire surveys like ours, while others are conducted either by analyzing MPI programs to reveal static behavior or by using profiling tools to analyze the dynamic runtime behavior of MPI jobs. Our survey is different from other questionnaire surveys in terms of its larger number of participants and wide geographic spread. As a result, it is possible to illustrate the current status of MPI users more accurately and with a wider geographical distribution. In [8], we have shown show some interesting findings, compare the results with preceding studies when possible, and provide some recommendations for MPI Forum based on the findings.

8.13 Narrowing the Search Space of Applications Mapping on Hierarchical Topologies

Participants: Emmanuel Jeannot.

Processor architectures at exascale and beyond are expected to continue to suffer from nonuniform access issues to in-die and node-wide shared resources. Mapping applications onto these resource hierarchies is an on-going performance concern, requiring specific care for increasing locality and resource sharing but also for ensuing contention. Application-agnostic approaches to search efficient mappings are based on heuristics. Indeed, the size of the search space makes it impractical to find optimal solutions nowadays and will only worsen as the complexity of computing systems increases over time. In this work [14] we leverage the hierarchical structure of modern compute nodes to reduce the size of this search space. As a result, we facilitate the search for optimal mappings and improve the ability to evaluate existing heuristics. Using widely known benchmarks, we show that permuting thread and process placement per node of a hierarchical topology leads to similar performances. As a result, the mapping search space can be narrowed down by several orders of magnitude when performing exhaustive search. This reduced search space will enable the design of new approaches, including exhaustive search or automatic exploration. Moreover, it provides new insights into heuristic-based approaches, including better upper bounds and smaller solution space.

8.14 Reinforcement Learning for Dynamic DAG Scheduling

Participants: Emmanuel Jeannot.

In this work [17], we have proposed READYS, a reinforcement learning algorithm for the dynamic scheduling of computations modeled as a Directed Acyclic Graph (DAGs). Our goal is to develop a scheduling algorithm in which allocation and scheduling decisions are made at runtime, based on the state of the system, as performed in runtime systems such as StarPU or ParSEC. Reinforcement Learning is a natural candidate to achieve this task, since its general principle is to build step by step a strategy that, given the state of the system (the state of the resources and a view of the ready tasks and their successors in our case), makes a decision to optimize a global criterion. Moreover, the use of Reinforcement Learning is natural in a context where the duration of tasks (and communications) is stochastic. We propose READYS that combines Graph Convolutional Networks (GCN) with an Actor-Critic Algorithm (A2C): it builds an adaptive representation of the scheduling problem on the fly and learns a scheduling strategy, aiming at minimizing the makespan. A crucial point is that READYS builds a general scheduling strategy which is neither limited to only one specific application or task graph nor one particular problem size, and that can be used to schedule any DAG. We focus on different types of task graphs originating from linear algebra factorization kernels (CHOLESKY, LU, QR) and we consider heterogeneous platforms made of a few CPUs and GPUs. We first propose to analyze the performance of READYS when learning is performed on a given (platform, kernel, problem size) combination. Using simulations, we show that the scheduling agent obtains performances very similar or even superior to algorithms from the literature, and that it is especially powerful when the scheduling environment contains a lot of uncertainty. We additionally demonstrate that our agent exhibits very promising generalization capabilities. To the best of our knowledge, this is the first paper which shows that reinforcement learning can really be used for dynamic DAG scheduling on heterogeneous resources.

8.15 Optimal Checkpointing Strategies for Iterative Applications

Participants: Guillaume Pallez.

This work provides an optimal checkpointing strategy to protect iterative applications from fail-stop errors. [6] 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.

8.16 Scheduling periodic I/O access with bi-colored chains: models and algorithms

Participants: Emmanuel Jeannot, Guillaume Pallez, Nicolas Vidal.

Observations show that some HPC applications periodically alternate between (i) operations (computations, local data-accesses) executed on the compute nodes, and (ii) I/O transfers of data and this behavior can be predicted beforehand. While the compute nodes are allocated separately to each application, the storage is shared and thus I/O access can be a bottleneck leading to contention. To tackle this issue, we design new static I/O scheduling algorithms that prescribe when each application can access

the storage. To design a static algorithm, we emphasize on the periodic behavior of most applications. Scheduling the I/O volume of the different applications is repeated over time. This is critical since often the number of application runs is very high. In the following report, we develop a formal background for I/O scheduling. First, we define a model, bi-colored chain scheduling, then we go through related results existing in the literature and explore the complexity of this problem variants. Finally, to match the HPC context, we perform experiments based on use-cases matching highly parallel applications or distributed learning framework. [9]

8.17 Exploring the Impacts of Workload Characterizations on Mapping Strategies

Participants: Emmanuel Jeannot, Guillaume Pallez, Nicolas Vidal.

In high performance, computing concurrent applications are sharing the same file system. However, the bandwidth which provides access to the storage is limited. Therefore, too many I/O operations performed at the same time lead to conflicts and performance loss due to contention. This scenario will become more common as applications become more data intensive. In this work, we discuss two simple and practical strategies to mitigate such performance loss. They are based on the idea of scheduling I/O access so as not to exceed some prescribe I/O bandwidth. More precisely, we compare two approaches: one grouping applications into packs that will be run independently (i.e pack scheduling), the other one scheduling greedily applications using a predefined order (i.e. list scheduling).

Results show that performances depend heavily on the I/O load and the homogeneity of the underlying workload. Finally, we introduce the notion of characteristic time, that represent information on the average time between consecutive I/O transfers. We show that it could be important to the design of schedulers and that we expect it to be easily obtained by analysis tools.

8.18 New interfaces for topologies management in parallel applications

Participants: Guillaume Mercier.

In [10], we present and discuss a unified view of and interface for collective communication in the MPI standard that in a natural way exploits MPI's orthogonality of concepts. We observed that the currently separate and different interfaces for sparse and global collective communication can be unified under the global collective communication interfaces, and at the same time lead to leaner and stronger support for stencil-like, sparse collective communication on Cartesian communicators. Our observations not only significantly reduce the number of concrete operation interfaces, but extend the functionality that can be supported by MPI while provisioning for possible future, much more wide-ranging functionality.

We suggest to (re)define communicators as the sole carriers of the topological structure over processes that determines the semantics of the collective operations, and to limit the functions that can associate topological information with communicators to the functions for distributed graph topology and intercommunicators creation. As a consequence, one set of interfaces for collective communication operations (in blocking, non-blocking, and persistent variants) will suffice, thereby explicitly eliminating the `MPI_Neighbor_` interfaces (in all variants) from the MPI standard. Topological structure will no longer be implied by Cartesian communicators, which in turn will have the sole role of naming processes in a (d-dimensional, Euclidean) geometric point-space. The geometric naming can be passed to the topology creating functions as part of the communicator, and guide process rank reordering and topological collective algorithm selection. We also explore ramifications of our proposal for one-sided communication.

Concretely, at the price of only one essential, additional function, our suggestion eliminates 10 concrete collective function interfaces from MPI 3.1, and 15 from MPI 4.0, while providing vastly more optimization scope for the MPI library implementation. Interfaces for Cartesian communicators can

likewise be simplified and/or eliminated from the MPI standard, as could the general active (post-start) synchronization mechanism for one-sided communication.

A prototype library is provided and we plan to make a concrete proposal to the MPI Forum for a future version of the standard (5.X) This proposal is currently under discussion in the Topologies Working Group of the MPI Forum.

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 International Labs

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.

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

Participants: Guillaume Pallez, Francieli Zanon-Boito.

HPCProSol

Title: Next-generation HPC PROblems and SOLutions

Expected duration: from 2021 to 2023.

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

Partner: Laboratório Nacional de Computação Científica (LNCC), Brazil

Situation: In 2021, no visits could be organized due to the sanitary situation. We are currently waiting for an answer regarding funding for 2022.

Summary: This joint team's main goal is to study and characterize the new HPC workload, represented by a set of scientific applications that are important to the LNCC because they are representative of its Santos Dumont machine's workload. This generated knowledge will guide the proposal of monitoring and profiling techniques for applications, and the design of new coordination mechanisms to arbitrate resources in HPC environments.

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

Informal International Partners

Argonne National Lab: Study of symmetries in process/thread mapping 8.13.

Vanderbilt University: Scheduling for Neurosciences 8.4.

10.2 International research visitors

None this year, due to the covid crisis.

10.3 European initiatives

10.3.1 FP7 & H2020 projects

Admire

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

- Admire: Adaptive multi-tier intelligent data manager for Exascale
- Program: H2020 EuroHPC
- Grant Agreement number: 956748 — ADMIRE — H2020-JTI-EuroHPC-2019-1
- 2021-2024

- Partners: University Carlos III of Madrid ; Johannes Gutenberg University Mainz ; Barcelona Supercomputing Center ; Technische Universitat Darmstadt ; DataDirect Networks France ; Inria ; ParaTools ; Forschungszentrum Julich GmbH ; Consorzio Interuniversitario Nazionale per l'Informatica ; CINECA ; E4 computer engineering ; Poznan Supercomputing and Networking Center ; Royal Institute of Technology ; The Max Planck Society.
- The main objective of the ADMIRE project is to establish a control at the system scale level to avoid congestion and balance computational with storage performance. More precisely, the goal is to create 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. TADaaM is responsible of the WP6 (Intelligent Controller) which is an instantiation of the service-layer that we envisioned at the beginning of the project.
- Clement Barthelemy has been hired in August 2021 as a research engineer to work specifically on this project. He has taken part in different ADMIRE activities, meetings and workshops and develops the WP6 main output, the distributed intelligent controller and its interfaces with applications, schedulers and decision engines.
- Website: https://www.admire-eurohpc.eu
- TADaaM funding: 640k€

Textarossa

Participants: Brice Goglin.

- Textarossa: Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale
- Program: H2020 EuroHPC
- Grand 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.
- Website: https://textarossa.eu/
- Reference publication: [11]
- TADaaM funding: 200k€

10.3.2 Other european programs/initiatives

PRACE 6IP

Participants: Emmanuel Jeannot.

- Title: PRACE Sixth Implementation Phase (PRACE-6IP) project
- Website: https://cordis.europa.eu/project/id/823767
- Duration: May 2019 - December 2021
- Inria contact: Luc Giraud
- The objectives of PRACE-6IP are to build on and seamlessly continue the successes of PRACE and start new innovative and collaborative activities proposed by the consortium. We worked on impact of process mapping on energy consumption (in collaboration with Avalon).

ANR-DFG H2M

Participants: Clément Foyer, Brice Goglin, Emmanuel Jeannot, Andrès Rubio Proano.

- Title: Heuristics for Heterogeneous Memory
- Website: https://h2m.gitlabpages.inria.fr/
- AAPG ANR 2020, 2021 - 2023 (48 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: https://project.inria.fr/dash/
- AP générique JCJC 2017, 03/2018 - 02/2022 (48 months)

- 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: https://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.

11 Dissemination

11.1 Promoting scientific activities

11.1.1 Scientific events: organisation

General chair, scientific chair

- Emmanuel JEANNOT and Guillaume PALLEZ are the general chair of the ICPP'22 conference.

Member of the organizing committees

- Emmanuel JEANNOT was member of the steering committee of the Euro-Par conference until August 2021.
- Guillaume PALLEZ is the finance chair (executive committee) of the IEEE Cluster'23 conference.
- Clément FOYER is the proceeding chair of the ICPP'22 conference.

11.1.2 Scientific events: selection

Chair of conference program committees

- Emmanuel JEANNOT was chair of the 2021 RADR Workshop on Resource Arbitration for Dynamic Runtimes (in conjunction with IPDPS).
- Emmanuel JEANNOT was chair of the 2021 COLOC Workshop on Data Locality (in conjunction with EuroPar).
- Guillaume PALLEZ was chair of the track "Algorithm" of the 2021 ICPP conference.
- Emmanuel JEANNOT was publicity chair of the IEEE Cluster 2021 conference.
- Guillaume PALLEZ is co-chair of the Panel track of the 2022 Cluster conference.

Member of the conference program committees

- Emmanuel JEANNOT was member of the following program committees: IPDPS 2021, SC 2021, Cluster 2021, Heteropar 2021, ROSS 2021.
- Brice GOGLIN was member of the following program committees: ISC 2021, RADR 2021, 2021 Smoky Mountains Conference, Hot Interconnect 28, COLOC 2021.
- Alexandre DENIS was a member of the following program committees: IEEE Cluster 2021, IPDPS 2021.
- Guillaume MERCIER was a member of the following program committees: ISC 2021, ICPP 2021, EuroMPI 2021.
- Guillaume PALLEZ was a member of the following program committees: IEEE IPDPS 2021, SIAM ACDA 2021.
- Francieli ZANON-BOITO was a member of the following program committees: IEEE CLUSTER 2021, IEEE HPCC 2021, and ISC 2021 Research Posters

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.
- Guillaume PALLEZ was a member of the IEEE TPDS Review board.

Reviewer - reviewing activities

- Emmanuel JEANNOT has reviewed a submission for Concurrency and Computation: Practice and Experience.
- Clément FOYER has reviewed a submission for IEEE Transactions on Parallel and Distributed Systems (TPDS).
- Alexandre DENIS was a reviewer for the journal of Parallel and Distributed Computing (JPDC).
- Francieli ZANON-BOITO was a reviewer for the International Journal of High Performance Computing Applications (IJHPCA).

11.1.4 Scientific expertise

- Brice GOGLIN was a member of the hiring committee for ATER positions at Université de Bordeaux.
- Brice GOGLIN was a member of the Khronos OpenCL Advisory Panel.
- François PELLEGRINI was the vice-president for the selection committee of two full professor positions at Université de Bordeaux.

11.1.5 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. The core of our Hsplit proposal was released as part of the version 4 of the MPI specifications in 2021. It will lead to other proposals and developments in the future. Guillaume MERCIER is also the chair of the standard chapter committee *Groups, Contexts, Communicators, Caching* and member of several other chapter committees.

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

11.1.6 Research administration

- Brice GOGLIN is in charge of the computing infrastructures of the Inria Bordeaux research center (since June 2021).
- 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.
- François PELLEGRINI is a co-pilot of the free/libre software experts group within the Committee for Open Science (CoSO) of the French Ministry of Higher Education.

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 architecture, operating systems, big data, parallel programming and high-performance runtime systems, as well as software law and personal data law.

- Brice GOGLIN gave courses about Operating Systems to teachers as part of the *Diplôme Inter Universitaire* to prepare them for teaching the new Computer Science track in high-school.
- 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 did two on-line training sessions on "*Strategic issues of information technologies*" and "*Personal data law*" to a group of administration heads and civil society activists of several French-speaking west-African countries, in the context of FFGI 2021 at Ouagadougou, Burkina Faso.
- Emmanuel JEANNOT gave a PhD-level introductory course on statistical tests.

11.2.2 Supervision

- PhD ended: Andrès RUBIO, Management on heterogeneous and non-volatile memories, defended in October 2021. Advisor: Brice GOGLIN.
- PhD in progress: Nicolas VIDAL, IO scheduling strategies, started in October 2018. Advisors: Guillaume PALLEZ and Emmanuel JEANNOT.
- PhD in progress: Philippe SWARTVAGHER, Interactions at large scale between high performance communication libraries and task-based runtime, started in October 2019. Advisors: Alexandre DENIS and Emmanuel JEANNOT.
- PhD in progress: Florian REYNIER, Task-based communication progression, started in January 2019. Advisors: Alexandre DENIS and Emmanuel JEANNOT.
- PhD started: Julien RODRIGUEZ, Circuit mapping onto multi-FPGA platforms, started in October 2020. Advisors: François PELLEGRINI, François GALEA and Lilia ZAOURAR.
- PhD started: 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 started: 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 started: 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.

11.2.3 Juries

- Brice GOGLIN was reviewer for the habilitation defense of François Trahay (Institut Polytechnique de Paris).
- Brice GOGLIN was the president of the Ph.D defense jury of Paul Beziau (U. Bordeaux).
- Emmanuel JEANNOT was member of jury of the “Isabelle Attali Prize”.

11.3 Popularization

11.3.1 Internal or external Inria responsibilities

- Brice GOGLIN was in charge of the diffusion of the scientific culture for the Inria Research Centre of Bordeaux until May 2021. He organized several popularization activities involving colleagues.

11.3.2 Articles and contents

- François PELLEGRINI was interviewed several times by newspapers and radio and television stations on the issues regarding biometry, e-voting, automated source code writing, so-called “artificial intelligence”, digital sovereignty, etc.
- François PELLEGRINI wrote the obituary of Philippe Aigrain for the bulletin of *Société informatique de France*.

11.3.3 Education

- Brice GOGLIN is the sponsor (*parrain*) of the *Edouard Vaillant* middle school (Bordeaux) for their scientific projects with the fondation *La main à la pâte*.
- François PELLEGRINI is a scientific and pedagogic advisor for the design of the new *Informatics* room at *Palais de la découverte*.

11.3.4 Interventions

- Emmanuel JEANNOT gave a talk at *the Ramiro Arrué* high school on "*Do Algorithms Control Us?*"
- Emmanuel JEANNOT gave a talk at "Unithé ou Café" on process mapping at the Bordeaux Inria center.
- Brice GOGLIN gave talks about research in computer science and high-performance computing at the *Odilon Redon* high school in Pauillac, as part of the *Fête de la Science* event and *Chiche* programme.
- Clément FOYER presented his research in the field of HPC to visiting high-school students, as part of their high-school internship programme which provides them with an early opportunity to discover the world of research.
- François PELLEGRINI gave a talk about the social issues of “artificial intelligence” at Forum NAIA-R, Bordeaux.

12 Scientific production

12.1 Major publications

[1] J. L. Bez, A. Miranda, R. Nou, F. Zanon Boito, T. Cortes and P. Navaux. 'Arbitration Policies for On-Demand User-Level I/O Forwarding on HPC Platforms'. In: *IEEE International Parallel & Distributed Processing Symposium (IPDPS 2021)*. Portland, Oregon, United States, 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, 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.

12.2 Publications of the year

International journals

[6] 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.

[7] 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.

[8] A. Hori, E. Jeannot, G. Bosilca, T. Ogura, B. Gerofi, J. Yin and Y. Ishikawa. 'An International Survey on MPI Users'. In: *Parallel Computing* (2021). URL: https://hal.inria.fr/hal-03347652.

[9] E. Jeannot, G. Pallez and N. Vidal. 'Scheduling periodic I/O access with bi-colored chains: models and algorithms'. In: *Journal of Scheduling* (2021). URL: https://hal.inria.fr/hal-03216844.

[10] J. L. Träff, S. Hunold, G. Mercier and D. Holmes. 'MPI collective communication through a single set of interfaces: A case for orthogonality '. In: *Parallel Computing* (2021). URL: https://hal.inria.fr/hal-03321274.

International peer-reviewed conferences

[11] G. Agosta, D. Cattaneo, W. Fornaciari, A. Galimberti, G. Massari, F. Reghenzani, F. Terraneo, D. Zoni, C. Brandolese, M. Celino et al. 'TEXTAROSSA: Towards EXtreme scale Technologies and Accelerators for euROhpc hw/Sw Supercomputing Applications for exascale'. In: DSD 2021 - 24th Euromicro Conference on Digital System Design. Palermo / Virtual, Italy, 1st Sept. 2021. URL: https://hal.inria.fr/hal-03329640.

[12] 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.

[13] 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, 9th Aug. 2021, p. 11. DOI: 10.1145/3472456.3473516. URL: https://hal.inria.fr/hal-03290121.

[14] N. Denoyelle, E. Jeannot, S. Perarnau, B. Videau and P. Beckman. 'Narrowing the Search Space of Applications Mapping on Hierarchical Topologies'. In: PMBS21 Workshop - 12th IEEE International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems, to be held in conjunction with SC21. Saint-Louis, United States, 2021. URL: https://hal.inria.fr/hal-03364531.

[15] C. Foyer and B. Goglin. 'Using Bandwidth Throttling to Quantify Application Sensitivity to Heterogeneous Memory'. In: MCHPC'21: Workshop on Memory Centric High Performance Computing. Saint-Louis, Missouri, United States, 14th Nov. 2021. URL: https://hal.inria.fr/hal-03356585.

[16] Y. Gao, G. Pallez, Y. Robert and F. Vivien. 'Work-in-Progress: Evaluating Task Dropping Strategies for Overloaded Real-Time Systems'. In: RTSS 2021 - 42nd IEEE Real-Time Systems Symposium. Dortmund, Germany: IEEE, 7th Dec. 2021, pp. 1–4. URL: https://hal.inria.fr/hal-03357422.

[17] N. Grinsztajn, O. Beaumont, E. Jeannot and P. Preux. 'READYS: A Reinforcement Learning Based Strategy for Heterogeneous Dynamic Scheduling'. In: IEEE Cluster 2021. Portland / Virtual, United States, 7th Sept. 2021. URL: https://hal.inria.fr/hal-03313229.

Conferences without proceedings

[18] F. Pellegrini. 'The originality of software works created by composition of pre-existing modules'. In: L'originalité du logiciel en question. Paris / Virtual, France, 13th Apr. 2021. URL: https://hal.inria.fr/hal-03202438.

[19] 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, 6th July 2021. URL: https://hal.inria.fr/hal-03290074.

Doctoral dissertations and habilitation theses

[20] A. Rubio Proaño. 'Data Placement Strategies for Heterogeneous and Non-Volatile Memories in High Performance Computing'. Université de Bordeaux, 7th Oct. 2021. URL: https://tel.archives-ouvertes.fr/tel-03431281.

Reports & preprints

[21] P.-A. Bouttier, L. Courtès, Y. Dupont, M. Felsöci, 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.

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

Other scientific publications

[23] F. Pellegrini. *The originality of software works created by composition of pre-existing components*. 13th Apr. 2021. URL: https://hal.inria.fr/hal-03201316.

[24] P. Swartvagher. 'Interferences between Communications and Computations in Distributed HPC Systems'. In: Euro-Par - 27th International European Conference on Parallel and Distributed Computing. Euro-Par 2021: Parallel Processing Workshops. Lisbon / Virtual, Portugal, 30th Aug. 2021. URL: https://hal.inria.fr/hal-03333852.

[25] P. Swartvagher. 'Interferences between Communications and Computations in Distributed HPC Systems'. In: Journée de l'École Doctorale Mathématiques et Informatique. Bordeaux, France, 20th May 2021. URL: https://hal.inria.fr/hal-03292004.

12.3 Other

Scientific popularization

[26] F. Pellegrini. 'Tribute to Philippe Aigrain'. In: *1024 : Bulletin de la Société Informatique de France* 18 (Nov. 2021), pp. 129–131. DOI: 10.48556/SIF.1024.18.129. URL: https://hal.inria.fr/hal-03457409.