RESEARCH CENTRE

Bordeaux - Sud-Ouest

IN PARTNERSHIP WITH:

**Institut Polytechnique de Bordeaux,
Université de Bordeaux**

2020
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 Team: 2015 January 01, updated into 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. – Networks

A2.1.7. – Distributed programming

A2.2.2. – Memory models

A2.2.4. – Parallel architectures

A2.2.5. – Run-time systems

A2.6.1. – Operating systems

A2.6.2. – Middleware

A3.1.2. – Data management, quering and storage

A3.1.3. – Distributed data

A3.1.8. – Big data (production, storage, transfer)

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

B6.5. – Information systems

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]

PhD Students

- Valentin Honoré [Univ de Bordeaux, until Oct 2020]
- Florian Reynier [CEA]
- Julien Rodriguez [CEA, from Oct 2020]
- Andres Xavier Rubio Proano [Inria]
- Philippe Swartvagher [Inria]
- Nicolas Vidal [Inria]

Technical Staff

- Adrien Guilbaud [Inria, Engineer]

Interns and Apprentices

- Valentin Hoyet [Inria, Apprentice]
- Pierre Pavia [Inria, from Oct 2020]

Administrative Assistants

- Catherine Cattaert Megrat [Inria, until Oct 2020]
- Roweida Mansour El Handawi [Inria, from Oct 2020]

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

[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

resources with other applications running concurrently during its execution. Depending on the allocated resources, it is not uncommon that the execution of one application interferes with the execution of a neighboring one.

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

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

3.2 Scientific Challenges and Research Issues

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

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

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

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

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

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

6.1 Awards

- Our proposal for hardware-based, topology-aware MPI communicators was accepted in the MPI standard and shall be part of the MPI 4.0 specifications (to be released mid-2021).
- The hwloc software project was nominated for the *Prix coup de cœur académique* award of the *Systematic* open-source hub.

6.2 Impact of the covid crisis

The year 2020 was marked by the covid crisis and its impact on society and its overall activity. The world of research was also greatly affected with an unequal impact depending on the responsibilities and roles: faculty members have seen their teaching load increase significantly; PhD students and post-docs have often had to deal with a worsening of their working conditions, as well as with reduced interactions with their supervisors and colleagues; most scientific collaborations have been greatly affected, with many international activities cancelled or postponed to dates still to be defined.

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

Author: Guillaume Mercier

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

Authors: Brice Goglin, Samuel Thibault

Contact: Brice Goglin

Participants: Brice Goglin, Valentin Hoyet

Partners: Open MPI consortium, Intel, AMD, IBM

7.1.3 NetLoc

Name: Network Locality

Keywords: Topology, Locality, Distributed networks, HPC, Parallel computing, MPI communication

Functional Description: Netloc (Network Locality) is a library that extends hwloc to network topology information by assembling hwloc knowledge of server internals within graphs of inter-node fabrics such as Infiniband, Intel OmniPath or Cray networks.

Netloc builds a software representation of the entire cluster so as to help applications properly place their tasks on the nodes. It may also help communication libraries optimize their strategies according to the wires and switches.

Netloc targets the same challenges as hwloc but focuses on a wider spectrum by enabling cluster-wide solutions such as process placement. It interoperates with the Scotch graph partitioner to do so.

Netloc is distributed within hwloc releases starting with hwloc 2.0.

URL: http://www.open-mpi.org/projects/netloc/

Publications: hal-01010599, hal-01614437

Contacts: Brice Goglin, Cyril Bordage

Participants: Brice Goglin, Clément Foyer, Cyril Bordage

7.1.4 NewMadeleine

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

Keywords: High-performance calculation, MPI communication

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

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

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

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: http://pm2.gforge.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

Authors: Alexandre Denis, Olivier Aumage, Raymond Namyst, Elisabeth Brunet, François Trahay, Nathalie Furmento

Contact: Alexandre Denis

Participants: Alexandre Denis, Clément Foyer, Nathalie Furmento, Raymond Namyst, Adrien Guilbaud, Florian Reynier, Philippe Swartvagher

7.1.5 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/`

Authors: Cedric Lachat, François Pellegrini, Cécile Dobrzynski, Cedric Lachat

Contacts: Cedric Lachat, François Pellegrini, Cécile Dobrzynski

Participants: Cécile Dobrzynski, Cedric Lachat, François Pellegrini

Partners: Université de Bordeaux, CNRS, IPB

7.1.6 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`

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

Contact: Emmanuel Jeannot

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

Partners: Université de Bordeaux, CNRS, IPB

7.1.7 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: Branch v6 offers many new features:

sequential graph repartitioning

sequential graph partitioning with fixed vertices

sequential graph repartitioning with fixed vertices

new, fast, direct k-way partitioning and mapping algorithms

multi-threaded, shared memory algorithms in the (formerly) sequential part of the library

exposure in the API of many centralized and distributed graph handling routines

embedded pseudo-random generator for improved reproducibility

and even more...

News of the Year: In 2020, Scotch switched to branch v6.1. While many changes from last year are mostly bugfixes, several modules have undergone thorough rewriting, such as the graph partitioning-with-overlap (Wgraph) module. The continuous integration process has been further developed.

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

Authors: François Pellegrini, Cédric Chevalier

Contacts: François Pellegrini, Marc Fuentes

Participants: François Pellegrini, Sébastien Fourestier, Jun-Ho Her, Cédric Chevalier, Amaury Jacques

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

7.1.8 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 Aupy, Julien Herrmann

Contacts: Julien Herrmann, Guillaume Aupy

8 New results

8.1 Dynamic broadcasts in task-based runtime systems

We worked on the improvement of broadcast performance in StarPU runtime with NewMadeleine. Although StarPU supports MPI, its distributed and asynchronous model to schedule tasks makes it impossible to use MPI optimized routines, such as `MPI_Bcast`. Indeed these functions need that all nodes participating in the collective are synchronized and know each others, which makes it unusable in practice for StarPU.

We proposed [11] a dynamic broadcast algorithm that runs without synchronization among participants, and where only the root node needs to know the others. Recipients do not even have to know whether the message will arrive as a plain send/receive or through a dynamic broadcast, which allows for a seamless integration in StarPU. We implemented the algorithm in our NewMadeleine communication library, leveraging its event-based paradigm and background progression of communications. We performed benchmarks using the Cholesky factorization that is known to use broadcasts and observed up to 30% improvement of its total execution time.

8.2 Assessing progression of MPI nonblocking collectives

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 proposed 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 [27] 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.

8.3 Distributed MPI synchronized clocks

When working on performance of collective operations, we need a way to measure time in a synchronized way accross all nodes. The precision of such a clock is of paramount importance. However, it is not straightforward, since local clocks of each node have a natural drift and are not synchronized with each other.

We have proposed and implemented a novel algorithm to get a synchronized clock that exhibits minimal drift, even on long periods. It is especially suitable to be used for long benchmarks or traces, so as to avoid the interference of a frequent re-synchronization of the clocks. The algorithm is implemented and is working. A paper about it will be submitted in the future.

8.4 Use of dedicated core for Non-blocking collective progression

The effective progression of non-blocking collective is difficult to perform in real HPC applications. These programs have various amount of communication and independent computation. HPC cluster also vary in hardware configuration. Finding the good configuration for a real performance gain is hard to define and sometimes implies the total redesign of applications.

In [21] we propose a general model of the parallelization of several applications to evaluate the impact of stealing a core to the application and measure the gains by progressing the non-blocking collectives on this core. We then applied this model on a real case and managed to gain time on a specific configuration. This work need to be extended to find the precise needs to overlap efficiently applications.

8.5 Reinforcement Learning for Dynamic DAG Scheduling

In practice, it is quite common to face combinatorial optimization problems which contain uncertainty along with non-determinism and dynamicity. These three properties call for appropriate algorithms; reinforcement learning (RL) is dealing with them in a very natural way. Today, despite some efforts, most real-life combinatorial optimization problems remain out of the reach of reinforcement learning algorithms. In this work [14], we have proposed a reinforcement learning approach to solve a realistic scheduling problem, and apply it to an algorithm commonly executed in the high performance computing community, the Cholesky factorization. On the contrary to static scheduling, where tasks are assigned to processors in a predetermined ordering before the beginning of the parallel execution, our method is dynamic: task allocations and their execution ordering are decided at runtime, based on the system state and unexpected events, which allows much more flexibility. To do so, our algorithm uses graph neural networks in combination with an actor-critic algorithm (A2C) to build an adaptive representation of the problem on the fly. We have showed that this approach is competitive with state-of-the-art heuristics used in high-performance computing runtime systems. Moreover, our algorithm does not require an explicit model of the environment, but we demonstrate that extra knowledge can easily be incorporated and improves performance.

8.6 MPI Introspection Monitoring

We described in [15] how to improve communication time of MPI parallel applications with the use of a library that enables to monitor MPI applications and allows for introspection (the program itself can query the state of the monitoring system). Based on previous work, this library is able to see how collective communications are decomposed into point-to-point messages. It also features monitoring sessions that allow suspending and restarting the monitoring, limiting it to specific portions of the code. Experiments show that the monitoring overhead is very small and that the proposed features allow for dynamic and efficient rank reordering enabling up to 2-time reduction of communication parts of some program.

8.7 Adaptive request scheduling for the I/O forwarding layer using reinforcement learning

I/O optimization techniques such as request scheduling can improve performance mainly for the access patterns they target, or they depend on the precise tune of parameters. In [8], we proposed an approach to adapt the I/O forwarding layer of HPC systems to the application access patterns by tuning a request scheduler.

Our case study was the TWINS scheduling algorithm, where performance improvements depend on the time window parameter, which in turn depends on the current workload. Our approach used a reinforcement learning technique — contextual bandits — to make the system capable of learning the best parameter value to each access pattern during its execution, without a previous training phase.

We evaluated our proposal and demonstrated it can achieve a precision of 88% on the parameter selection in the first hundreds of observations of an access pattern. After having observed an access pattern for a few minutes (not necessarily contiguously), we showed that the system will be able to optimize its performance for the rest of the life of the system (years).

8.8 Arbitration policies for on-demand I/O forwarding on HPC platforms

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 [17], we investigated the effects of I/O forwarding on performance by considering the application and system characteristics. For this evaluation, we implemented FORGE, a user-level forwarding emulator, and used the MareNostrum 4 (Spain) and Santos Dumont (Brazil) supercomputers. Our results demonstrated the importance of considering applications I/O characteristics when arbitrating the access to these resources, and the potential improvement that could come from an arbitration policy considering this information.

In [9], we investigated such arbitration policies. 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.

For such an allocation technique to be successful in practice, it is important to properly estimate the impact of different numbers of I/O nodes on each application's performance. In [31], we explored the idea of predicting application performance by extracting information from a coarse-grained aggregated trace from a previous execution, and then using this information to match each of the application's I/O phases to an equivalent benchmark, for which we could have performance results. We tested this idea by applying it to five different applications over three case studies, and found a mean error of approximately 20%. We extensively discussed the obtained results and limitations to the approach, pointing at future work opportunities.

8.9 New methods for graph partitioning with overlap

Among the graph partitioning features of SCOTCH is the ability to compute k-way partition with overlap of a graph. In this kind of partitioning, all parts comprise an internal core and a boundary layer, such that vertices of the boundary layer may belong to several neighboring parts at the same time. The objective function is to balance the load of the parts, boundary layer included, while minimizing the overall size of this boundary layer.

This kind of partitioning was originally designed for *ab nitio* quantum chemistry computations, to balance the computational workload of parallel solvers dealing with overlapping square matrices. Here, the boundary layer across domains modeled the overlapping parts of the matrices, and matrices were of very small sizes (less than a few tens of thousands of columns), hence resulting in small graphs to be managed, and partitioned into a rather small number of parts.

However, as a new user experimented with this module on very large graphs in the domain of oil reservoir simulations, the existing code, which was produced during the post-doc of Jun-Ho HER, exhibited some limitations in terms of performance and robustness. Hence, a complete rewriting of this module has been carried out, notably to implement optimized algorithms for performing a k-way version of the Fiduccia-Mattheyses algorithm, for this particular graph partitioning model. The resulting code has been released in version v6.1.0 of the SCOTCH software.

8.10 Improved methods for ordering weighted graphs

One of the main features of the SCOTCH partitioning software is to compute nested-dissection reorderings of sparse matrices, in order to reduce fill-in during the solving of large sparse systems of linear equations. In this context, graph vertices represent matrix variables (that is, row and column indices), and graph edges represent non-diagonal elements that connect variables together.

To reduce the cost of time-consuming matrix operations, it may be useful to compress matrices and their associated graphs: strongly inter-related variables are merged together into “super-variables”, to reduce graph size and compute time. However, the overall memory and time taken afterwards to process super-variables is higher than for uncompressed variables. Not taking this into account perturbs fill-in and load balancing estimators and strategies.

Consequently, in the case of compressed graphs, vertex graphs must be weighted, to reflect the number of variables that have been aggregated into a single super-variable, and vertex weights must be taken into account in all graph partitioning and load balancing routines. This work has been carried out in collaboration with the developers of the MUMPS solver (for which a dedicated interface exists in SCOTCH). The resulting code has been released in version v6.1.0 of the SCOTCH software.

8.11 Mapping circuits onto multi-FPGA platforms

An FPGA (Field Programmable Gate Array) is an integrated circuit comprising a large number of programmable and interconnectable logic resources, which allows one to implement, by programming, a digital electronic circuit such as a microprocessor, a compute accelerator or a complex hybrid system-on-chip. However, some circuits are too big to be implemented on a single FPGA. To address this issue, several FPGAs can be interconnected on a dedicated hardware platform. However, available circuit placement tools do not account for all the constraints of the placement problem to be solved in order to map efficiently a circuit onto a multi-FPGA platform. For example, cost functions are not designed to minimize signal propagation times between FPGA registers, nor do they take into account the capacity constraints induced by the routing of connections across FPGAs.

In order to solve these issues and to design and implement efficient algorithms for mapping very large circuits onto multi-FPGA platformes, a PhD work has started in October 2020, in collaboration with CEA/LIST. The PhD candidate, Julien RODRIGUEZ, has started reviewing the existing bibliography and is currently designing models, based on hypergraphs, to represent the circuits and the FPGA platforms.

8.12 Exposing the characteristics of heterogeneous memory architectures to parallel applications

The complexity of memory system has increased. Supercomputers may now include memory at several levels, that is heterogeneous and non-uniform memory, with significantly different properties. In that sense developers of scientific applications face a huge challenge: efficiently exploit the memory system.

In [22] we present a method to better manage the complexity of modern memory systems by extending the hwloc programming interface by an API. To do so, we consider the capabilities of hwloc to expose the memory hierarchy and the characterization of the memories by means of a set of attributes and metrics that allow us to obtain and ordering of memory targets to then use most suitable memory target for each allocation. This attributes we consider to be obtained in two ways. The use of hardware information given by the ACPI Heterogeneous Memory Attribute (HMAT) and also as an alternative or complemetary manner by benchmarking.

Our API can be used by runtime systems and parallel libraries to provide memory allocator and placement policies that finely respect the affinities and needs of computational tasks.

8.13 New Interfaces for Management of Topologies in parallel applications

In [18], we exposed some ideas in order to unify and rationalize the way virtual topologies can be managed in MPI. In the current version of MPI, several types of virtual topologies exist with their dedicated sets of functions. Our main idea is to unify and rationalize this management by letting the nature of the underlying object guide the behaviour of a function this is applied to it. By doing so, the current interface could be leaner and offer more functionnalities. We plan to expand this work so as to also encompass hardware topologies management and create a holistic and sensible approach to the issue of topologies management in MPI. We also plan to make a concrete proposal to the MPI Forum for a future version of the standard.

8.14 Mapping and Scheduling HPC Applications for Optimizing I/O

In [10] we addressed the issue of mapping applications to alleviate I/O contention. I/O contention can represent a performance bottleneck. The access to bandwidth can be split in two complementary yet distinct problems. The mapping problem and the scheduling problem. The mapping problem consists in selecting the set of applications that are in competition for the I/O resource. The scheduling problem consists then, given I/O requests on the same resource, in determining the order to these accesses to minimize the I/O time. In this work we propose to couple a novel bandwidth-aware mapping algorithm to I/O list-scheduling policies to develop a cross-layer optimization solution.We study this solution experimentally using an I/O middleware: CLARISSE. We show that naive policies such as FIFO perform relatively well in order to schedule I/O movements, and that the important part to reduce congestion lies mostly on the mapping part. We evaluate the algorithm that we propose using a simulator that we validated experimentally. This evaluation shows important gains for the simple, bandwidth-aware mapping solution that we provide compared to its non bandwidth-aware counterpart. The gains are both in terms of machine efficiency (makespan) and application efficiency (stretch). This stresses even more the importance of designing efficient, bandwidth-aware mapping strategies to alleviate the cost of I/O congestion.

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

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 [30] 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.

In addition, we showed how one could derive quasi-optimal scheduling solutions given this type of application model in [13].

8.16 Robustness of the Young/Daly formula for stochastic iterative applications

Following the previous study and the realization of stochastic behavior in application performance, we studied how the strategies at the foundations of resilience hold in the presence of stochastic behavior in [12]. The Young/Daly formula for periodic checkpointing is known to hold for a divisible load application where one can checkpoint at any time-step. In an nutshell, the optimal period is $P = \sqrt{2\mu C}$ where μ is the Mean Time Between Failures (MTBF) and C is the checkpoint time. This work assesses the accuracy of the formula for applications decomposed into computational iterations where: (i) the duration of an iteration is stochastic, i.e., obeys a probability distribution law of mean μ; and (ii) one can checkpoint only at the end of an iteration. We were able to establish that the relevance of the formula goes well beyond its original framework.

8.17 Application-Driven Requirements for Node Resource Management in Next-Generation Systems

Emerging workloads on supercomputing platforms are pushing the limits of traditional high-performance computing software environments. Multi-physics, coupled simulations, big-data processing and machine learning frameworks, and multi-component workloads pose serious challenges to system and application developers. At the heart of the problem is the lack of cross-stack coordination to enable flexible resource management among multiple runtime components. In collaboration with RIKEN, CEA, LLNL and Intel, we analyzed in [16] seven real-world applications that represent emerging workloads and illustrate the scope and magnitude of the problem. We then extract several themes from these applications that highlight next-generation requirements for node resource managers. Finally, using these requirements, we propose a general, cross-stack coordination framework and outline its components and functionality.

9 Partnerships and cooperations

9.1 International Initiatives

9.1.1 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.

9.1.2 Inria Associate Team not involved in an IIL

Informal International Partners

Argonne National Lab: Binomial Checkpointing Strategies for Machine Learning (recipient of a FACCTS grant, 2018-2020) as well as network performance prediction and study of symmetries in process/thread mapping.

Vanderbilt University: Scheduling for Neurosciences 8.15.

ICL at University of Tennessee: on instrumenting MPI applications and modeling platforms (works on HWLOC take place in the context of the Open MPI consortium) and MPI and process placement.

Lawrence Livermore National Laboratory: Exposing Heterogeneous Memory Characteristics to HPC Applications 7.1

9.2 International Research Visitors

None this year, due to the covid crisis.

9.3 European Initiatives

9.3.1 Collaborations in European Programs, except FP7 and H2020

PRACE 6IP

- Title: PRACE Sixth Implementation Phase (PRACE-6IP) project
- See also: https://cordis.europa.eu/project/id/823767
- Duration: May 2019 - December 2021
- Partners: see https://cordis.europa.eu/project/id/823767
- 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).

9.3.2 Collaborations with Major European Organizations

RWTH Aachen University (Germany): HPC Group (Christian Terboven) within the H2M ANR-DFG project.

9.4 National Initiatives

ANR

ANR DASH Data-Aware Scheduling at Higher scale (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 — SOLvers for Heterogeneous Architectures over Runtime systems, Investigating Scalability.

- 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.

ANR-DFG H2M — H2M: Heuristics for Heterogeneous Memory

- 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.

ADT - Inria Technological Development Actions

ADT Gordon

- 10/2018 - 09/2020 (24 months)
- Coordinator: Emmanuel JEANNOT
- Other partners: Storm, HiePACS, PLEIADE (Inria Bordeaux)
- Abstract: Teams HiePACS, Storm and Tadaam develop each a brick of an HPC software stack, namely solver, runtime, and communication library. The goal of the Gordon project is to consolidate the HPC stack, to improve interfaces between each brick, and to target a better scalability. The bioinformatics application involved in the project has been selected so as to stress the underlying systems.

9.5 Regional Initiatives

HPC-Ecosystem

- Participants : Alexandre DENIS, Emmanuel JEANNOT, Guillaume PALLEZ, Philippe SWARTVAGHER, Nicolas VIDAL.
- Grant: Regional council
- Dates: 2018 – 2020
- Partners: EPIs STORM , HIEPACS from Inria Bordeaux Sud-Ouest, Airbus, CEA-CESTA, INRA
- Overview: The goal of this project is to develop a unified Execution Support (SE) for large-scale numerical simulation and the processing of large volumes of data. We identified four Application Challenges (DA) identified by the Nouvelle-Aquitaine region that we propose to carry over this unified support. We will finally develop four Methodological Challenges (CM) to evaluate the impact of the project. This project will make a significant contribution to the emerging synergy on the convergence between two yet relatively distinct domains, namely High Performance Computing (HPC) and the processing, management of large masses of data (Big Data); this project is therefore clearly part of the emerging field of High Performance Data Analytics (HPDA).

10 Dissemination

10.1 Promoting Scientific Activities

10.1.1 Scientific Events: Organisation

General Chair, Scientific Chair

- François PELLEGRINI co-chaired the *CNIL-Inria prize on privacy protection*, granted on January 2020 during the CPDP conference.
- François PELLEGRINI co-chaired on October 2020 the *Convergences du Droit et du Numérique* transdisciplinary event, where scientists in the fields of law and informatics work together on common subjects. https://cdn.u-bordeaux.fr/
- Emmanuel JEANNOT and Guillaume PALLEZ are the general chair of the ICPP'22 conference.

Member of the Organizing Committees

- Emmanuel JEANNOT is member of the steering committee of the Euro-Par conference.
- Emmanuel JEANNOT is member of the steering committee of the Cluster conference (until Sept. 2020).
- Guillaume PALLEZ was the finance liaison to Technical Program of SC'20.
- Guillaume PALLEZ is the finance chair (executive committee) of the IEEE Cluster'23 conference.

10.1.2 Scientific Events: Selection

Chair of Conference Program Committees

- Brice GOGLIN was the co-chair of the area "Architecture and Networking" of the 2020 SuperComputing conference.
- Emmanuel JEANNOT was chair of the track "Application, Algorithms and Library" of the 2020 Cluster conference.
- Emmanuel JEANNOT was chair of the RADR Workshop on Resource. Arbitration for Dynamic Runtimes (in conjunction with IPDPS)
- GuillaumePALLEZ was chair of the track "Algorithm" of the 2021 ICPP conference.

Member of the Conference Program Committees

- Emmanuel JEANNOT was member of the following program committees: SuperComputing 2020, IPDPS 2021, ROSS 2021, HiPC 2020, SBAC-PAD 2020.
- Brice GOGLIN was member of the following program committees: RADR 2020, COLOC 2020.
- Guillaume MERCIER was member of the following program committees: EuroMPI 2020, ISC 2020.
- Guillaume PALLEZ was member of the following program committees: ICPP 2020, IPDPS 2021, SC 2020 (tutorial).
- Alexandre DENIS was member of the following program committees: Cluster 2020, Cluster 2021, IPDPS 2021.
- Francieli ZANON BOITO was member of the following program committees: Cluster 2020, SC 2020 (tutorial), Cluster 2021.

Reviewer

- Alexandre DENIS was a reviewer for SC 2020 and ISC 2021.
- Francieli ZANON BOITO was a reviewer for ICS 2020.

10.1.3 Journal

Member of the Editorial Boards

- Emmanuel JEANNOT is associate editor of the International Journal of Parallel, Emergent & Distributed Systems (IJPEDS).
- Guillaume PALLEZ is a Review Board Member for IEEE Transactions on Parallel and Distributed Systems (TPDS)

Reviewer - Reviewing Activities

- Emmanuel JEANNOT was reviewer for the following journals: Transactions on Computers, The Journal of Supercomputing, IEEE Transactions on Parallel and Distributed Systems, Parallel Computing, Concurrency and Computation: Practice and Experience.

10.1.4 Invited Talks

- Brice GOGLIN gave a keynote speech at the SBAC-PAD conference.
- François PELLEGRINI delivered a speech on "*The ethical issues of the protection of personal data*" at the scientific seminar "*Ethics: when digital sciences come close to the life of Humans*" of the Institute of Information Sciences and their Applications (INS2I) of CNRS.

10.1.5 Leadership within the Scientific Community

- François PELLEGRINI is a commissionner at CNIL, the French data protection authority.
- François PELLEGRINI is the co-pilot of the *Free/Libre software working group* of the *Committee for Open Science* (CoSO) of the French Ministry of Higher Education and Research.

10.1.6 Scientific Expertise

- Brice GOGLIN was a member of the hiring committee for a professor position at Université de Bordeaux.
- Emmanuel JEANNOT and Guillaume PALLEZ were members of the hiring committee of Inria junior researcher position at the national Level.
- François PELLEGRINI participated in the hearing by the Mission on monitoring of the Covid-19 emergency law of the French Senate, about contact-tracing applications.
- François PELLEGRINI evaluated a project for the Belgian FRS.

10.1.7 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 voted in the version 4 of the MPI specifications that 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.

10.1.8 Research Administration

- Emmanuel JEANNOT is head of science of the Bordeaux-Sud-Ouest Inria Research Centre.
- Emmanuel JEANNOT is member, Guillaume PALLEZ is elected member of the Inria evaluation committee.

10.2 Teaching - Supervision - Juries

10.2.1 Juries

- Brice GOGLIN was reviewer for the PhD defense of Raphael Ponsard (U. Grenoble).
- Brice GOGLIN was the president of the PhD defense jury of Li Han (ENS Lyon and U. Shanghai).

- Emmanuel JEANNOT was member of the PhD defense jury of Mainassara Chekaraou (U. Luxembourg).
- Guillaume PALLEZ was reviewer for the PhD defense (viva) of Najat Kukreja (Imperial College London).
- Francieli ZANON BOITO was member of the PhD defense jury of Tao Chang (Institut Polytechnique de Paris).

10.2.2 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 algorithmics 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.

- François PELLEGRINI did a course in English on “*Software Law*” and “*Personal data law*” to 23 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 2020 at Ouagadougou, Burkina Faso.
- François PELLEGRINI did an in-person 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.

10.2.3 Supervision

- PhD in progress: Andrès RUBIO, Management on heterogeneous and non-volatile memories, started in October 2018. 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.

10.3 Popularization

10.3.1 Internal or external Inria responsibilities

- Brice GOGLIN is in charge of the diffusion of the scientific culture for the Bordeaux-Sud-Ouest Inria Research Centre. He organized several popularization activities involving colleagues.

10.3.2 Articles and contents

- François PELLEGRINI released an on-line course on *Informatics and the Society*, aimed at first-year university students (Pix), high school pupils and the general public. This course is already used at Université de Bordeaux and by other universities (Lille, Avignon, etc.), and is freely available at : https://pedagotec.u-bordeaux.fr/informatique_et_societe/

10.3.3 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*.

10.3.4 Interventions

- Brice GOGLIN and Valentin HONORÉ were part of the speed dating event *Déclics* between researchers and teenagers in the *Camille Julian* high school in Bordeaux.
- Brice GOGLIN gave talks about research in computer science and high-performance computing at the *Monzie* high school in Bazas, as part of the *Fête de la Science* event and *Chiche* programme.
- Guillaume PALLEZ was invited to give a presentation and to intervene at the *Elie Faure* middle school as part of the *Sciences en collège* event led by *Cap Sciences*. The students then wrote a summary of the intervention https://echosciences.nouvelle-aquitaine.science/articles/a-quoi-servent-les-maths-dans-la-vie-de-tous-les-jours

11 Scientific production

11.1 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.

11.2 Publications of the year

International journals

[6] 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.

[7] J. Herrmann and G. Pallez. 'H-Revolve: A Framework for Adjoint Computation on Synchronous Hierarchical Platforms'. In: *ACM Transactions on Mathematical Software* (2020). DOI: 10.1145/3378672. URL: https://hal.inria.fr/hal-02080706.

[8] J. Luca Bez, F. Zanon Boito, R. Nou, A. Miranda, T. Cortes and P. O. Navaux. 'Adaptive Request Scheduling for the I/O Forwarding Layer using Reinforcement Learning'. In: *Future Generation Computer Systems* 112 (2020), pp. 1156–1169. DOI: 10.1016/j.future.2020.05.005. URL: https://hal.inria.fr/hal-01994677.

International peer-reviewed conferences

[9] 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, 17th May 2021. URL: https://hal.inria.fr/hal-03149582.

[10] J. Carretero, E. Jeannot, G. Pallez, D. E. Singh and N. Vidal. 'Mapping and Scheduling HPC Applications for Optimizing I/O'. In: ICS2020 - 34th ACM International Conference on Supercomputing. Barcelona, Spain, 29th June 2020. URL: https://hal.archives-ouvertes.fr/hal-02559749.

[11] A. Denis, E. Jeannot, P. Swartvagher and S. Thibault. 'Using Dynamic Broadcasts to improve Task-Based Runtime Performances'. In: Euro-Par - 26th International European Conference on Parallel and Distributed Computing. Euro-Par 2020. Warsaw, Poland: https://2020.euro-par.org/, 24th Aug. 2020. DOI: 10.1007/978-3-030-57675-2_28. URL: https://hal.inria.fr/hal-02872765.

[12] Y. Du, L. Marchal, Y. Robert and G. Pallez. 'Robustness of the Young/Daly formula for stochastic iterative applications'. In: ICPP 2020 - 49th International Conference on Parallel Processing. Edmonton / Virtual, Canada, 17th Aug. 2020, pp. 1–11. DOI: 10.1145/3404397.3404419. URL: https://hal.inria.fr/hal-03024618.

[13] A. Gainaru, B. Goglin, V. Honoré, G. Pallez, P. Raghavan, Y. Robert and H. Sun. 'Reservation and Checkpointing Strategies for Stochastic Jobs'. In: IPDPS 2020 - 34th IEEE International Parallel and Distributed Processing Symposium. New Orleans, LA / Virtual, United States, 18th May 2020, pp. 1–26. URL: https://hal.inria.fr/hal-03029298.

[14] N. Grinsztajn, O. Beaumont, E. Jeannot and P. Preux. 'Geometric Deep Reinforcement Learning for Dynamic DAG Scheduling'. In: IEEE SSCI 2020 - Symposium Series on Computational Intelligence. SSCI 2020 proceedings. Canberra / Virtual, Australia, Dec. 2020. URL: https://hal.inria.fr/hal-03028981.

[15] E. Jeannot and R. Sartori. 'Improving MPI Application Communication Time with an Introspection Monitoring Library'. In: PDSEC 2020 - 21st IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing. New-Orleans, United States, May 2020, p. 10. URL: https://hal.inria.fr/hal-02906352.

[16] E. A. León, B. Gerofi, J. Jaeger, G. Mercier, R. Riesen, M. Takagi and B. Goglin. 'Application-Driven Requirements for Node Resource Management in Next-Generation Systems'. In: ROSS 2020 : International Workshop on Runtime and Operating Systems for Supercomputers. Atlanta, GA / Virtual, United States, 13th Nov. 2020. URL: https://hal.inria.fr/hal-02950635.

[17] J. Luca Bez, F. Zanon Boito, A. Miranda, R. Nou, T. Cortes and P. O. A. Navaux. 'Towards On-Demand I/O Forwarding in HPC Platforms'. In: PDSW 2020 : 5th IEEE/ACM International Parallel Data Systems Workshop. Atlanta, Georgia / Virtual, United States, 12th Nov. 2020. URL: https://hal.inria.fr/hal-03150024.

[18] J. L. Träff, S. Hunold, G. Mercier and D. Holmes. 'Collectives and Communicators: A Case for Orthogonality'. In: EuroMPI/USA '20: 27th European MPI Users' Group Meeting. Austin, TX / Virtual, United States, 21st Sept. 2020, pp. 31–38. DOI: 10.1145/3416315.3416319. URL: https://hal.inria.fr/hal-03117285.

Conferences without proceedings

[19] V. Honoré. 'Techniques d'ordonnancement pour les applications stochastiques sur plateformes HPC'. In: COMPAS 2020 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Lyon, France, 30th June 2020. URL: https://hal.inria.fr/hal-02635733.

[20] F. Pellegrini and F. Vallet. 'When sound processing meets data protection'. In: Linux Audio Conference 2020. Bordeaux / Virtual, France: https://lac2020.sciencesconf.org/, 25th Nov. 2020. URL: https://hal.inria.fr/hal-03096429.

[21] F. Reynier. 'Utilisation de cœurs dédiés pour la progression des communications collectives non bloquantes'. In: COMPAS 2020 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Lyon, France: https://2020.compas-conference.fr/, 30th June 2020. URL: https://hal.inria.fr/hal-03118807.

[22] A. Rubio Proaño. 'Exposer les caractéristiques des architectures à mémoires hétérogènes aux applications parallèles'. In: COMPAS 2020 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Lyon, France: https://2020.compas-conference.fr/, 29th June 2020. URL: https://hal.inria.fr/hal-02639607.

[23] P. Swartvagher. 'Amélioration des performances de supports d'exécution à tâches à l'aide de broadcasts dynamiques'. In: COMPAS 2020 - Conférence francophone d'informatique en Parallélisme, Architecture et Système. Lyon, France, 30th June 2020. URL: https://hal.inria.fr/hal-02580626.

Edition (books, proceedings, special issue of a journal)

[24] U. Schwardmann, C. Boehme, D. B. Heras, V. Cardellin, E. Jeannot, A. Salis, C. Schifanella, R. R. Manumachu, D. Schwamborn, L. Ricci, O. Sangyoon, T. Gruber, L. Antonelli and S. L. Scott. *Euro-Par 2019: Parallel Processing Workshops*. Vol. 11997. Lecture Notes in Computer Science. Göttingen, Germany, 2020. DOI: 10.1007/978-3-030-48340-1. URL: https://hal.inria.fr/hal-03146470.

Doctoral dissertations and habilitation theses

[25] V. Honoré. 'HPC - Big Data Convergence : Managing theDiversity of Application Profiles on HPC Facilities'. Université de Bordeaux, 15th Oct. 2020. URL: https://tel.archives-ouvertes.fr/tel-03003808.

Reports & preprints

[26] L. Brotcorne, A. Canteaut, A. C. Viana, C. Grandmont, B. Guedj, S. Huot, V. Issarny, G. Pallez, V. Perrier, V. Quema, J.-B. Pomet, X. Rival, S. Salvati and E. Thomé. *Indicateurs de suivi de l'activité scientifique de l'Inria*. Inria, 1st Dec. 2020. URL: https://hal.inria.fr/hal-03033764.

[27] A. Denis, J. Jaeger, E. Jeannot and F. Reynier. *Experiments for Assessing Computation/Communication Overlap of MPI Nonblocking Collectives*. Inria & Labri, Univ. Bordeaux; CEA, CEA/DAM/DIF, Bruyères-le-Châtel, France, Oct. 2020, p. 39. URL: https://hal.inria.fr/hal-03012097.

[28] Y. Du, L. Marchal, G. Pallez and Y. Robert. *Optimal Checkpointing Strategies for Iterative Applications*. Inria - Research Centre Grenoble – Rhône-Alpes, Oct. 2020. URL: https://hal.inria.fr/hal-02980455.

[29] Y. Du, L. Marchal, G. Pallez and Y. Robert. *Robustness of the Young/Daly formula for stochastic iterative applications*. Inria Grenoble Rhône-Alpes, Mar. 2020. URL: https://hal.inria.fr/hal-02514107.

[30] A. Gainaru, B. Goglin, V. Honoré and G. Pallez. *Profiles of upcoming HPC Applications and their Impact on Reservation Strategies*. Inria & Labri, Université Bordeaux, 25th Aug. 2020, p. 30. URL: https://hal.inria.fr/hal-02921487.

[31] F. Zanon Boito. *Estimation of the impact of I/O forwarding on application performance*. Inria, 16th Oct. 2020, p. 20. URL: https://hal.inria.fr/hal-02969780.

Other scientific publications

[32] P. Beckman, E. Jeannot and S. Perarnau. *Workshop on Resource Arbitration for Dynamic Runtimes (RADR)*. New-Orleans / Virtual, United States, 19th May 2020. DOI: 10.1109/IPDPSW50202.2020.00157. URL: https://hal.inria.fr/hal-03146446.

[33] F. Pellegrini. *Reflections on digital contact tracing tools: Contribution to the round table of academics organised on 23/04/2020 by the Senate's Law Commission in the context of the monitoring mission on the emergency law to combat the COVID-19 outbreak.* 30th Apr. 2020. URL: https://hal.inria.fr/hal-02554672.

11.3 Other

Softwares

[34] [SW] S. Archipoff, C. Augonnet, O. Aumage, G. Beauchamp, B. Bramas, A. Buttari, A. Cassagne, J. Clet-Ortega, T. Cojean, N. Collin, V. Danjean, A. Denis, L. Eyraud-Dubois, N. Furmento, S. Henry, A. Hugo, M. Juhoor, A. Juven, M. Keryell-Even, Y. Khorsi, T. Lambert, E. Leria, B. Lizé, M. Makni, S. Nakov, R. Namyst, L. Nesi Lucas, P. Joris, D. Pasqualinotto, S. Pitoiset, Q.-D. Nguyen, C. Roelandt, C. Sakka, C. Salingue, L. Mello Schnorr, M. Sergent, A. Simonet, L. Stanisic, B. Subervie, F. Tessier, S. Thibault, B. Videau, L. Villeveygoux and P.-A. Wacrenier, *StarPU* version 1.3.3, 20th Jan. 2020. HAL: ⟨hal-02443512⟩, URL: https://hal.inria.fr/hal-02443512, SWHID: ⟨swh:1:dir:b6e19d99449a78805e7a55a341fbaba2bc431973;origin=https://hal.archives-ouvertes.fr/hal-02443512;visit=swh:1:snp:c21d3dfbd96e4fb502c534e59644dba14c542100;anchor=swh:1:rev:31be198773f103324593d26369f135fbde5b97f8;path=/⟩.