IN PARTNERSHIP WITH:
Institut Polytechnique de Bordeaux

Université de Bordeaux

Activity Report 2018

Project-Team TADAAM

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

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

RESEARCH CENTER
Bordeaux - Sud-Ouest

THEME
Distributed and High Performance Computing

Table of contents

Project-Team TADAAM

Creation of the Team: 2015 January 01, updated into Project-Team: 2017 December 01

Keywords:

Computer Science and Digital Science:

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, Visitors, External Collaborators

Research Scientists

Guillaume Aupy [Inria, Researcher]

Alexandre Denis [Inria, Researcher]
Brice Goglin [Inria, Researcher, HDR]
Emmanuel Jeannot [Team Leader, Inria, Senior Researcher, HDR]

Faculty Members
Guillaume Mercier [Institut National Polytechnique de Bordeaux, Associate Professor]
François Pellegrini [Univ de Bordeaux, Professor, HDR]

Post-Doctoral Fellow
Julien Herrmann [Inria, from Oct 2018]

PhD Students
Nicolas Denoyelle [Bull, until Nov 2018]
Valentin Honore [Univ de Bordeaux]
Benjamin Lorendeau [EDF, until Mar 2018]
Andres Xavier Rubio Proano [Inria, from Oct 2018]
Hugo Taboada [CEA, until Sep 2018]
Nicolas Vidal [Inria, from Oct 2018]

Technical staff
Cyril Bordage [Inria, until Mar 2018]
Clément Foyer [Inria, until Jan 2018]
Adrien Guilbaud [Inria, from Nov 2018]

Interns
Thibaut Lausecker [Inria, from Jun 2018 until Jul 2018]
Richard Sartori [Inria, from Jun 2018 until Sep 2018]
Nicolas Vidal [Inria, until Jun 2018]
Valentin Hoyet [Inria, from Oct 2018 (apprenti)]

Administrative Assistant
Catherine Cattaert Megrat [Inria]

2. Overall Objectives

2.1. 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.

[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

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. This is the case for at least two thirds of the applications selected in the 9^{th} PRACE. call [3], which concern quantum mechanics, fluid mechanics, climate, material physic, electromagnetism, etc.

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. Highlights of the Year

5.1. Highlights of the Year

Guillaume MERCIER is the chairman of the Hardware Topologies Management Working Group of the MPI Forum. This working group was created officially in December by Inria's impulse and has been rallied since by many institutions taking part in the MPI Forum. The goal of this working group is to standardize hardware topologies management mechanisms and abstractions in the MPI standard.

6. New Software and Platforms

6.1. Hsplit

Hierarchical communicators split

[3] http://www.prace-ri.eu/prace-9th-regular-call/

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: A new working group in the MPI Forum to champion the integration of this proposal in the MPI standard has been created. This working group includes Inria, CEA, Atos/Bull, Paratools, the University of Tennessee - Knoxville and many other institutions/companies are interested to join in.

- Participants: Guillaume Mercier, Brice Goglin, Emmanuel Jeannot and Farouk Mansouri
- Contact: Guillaume Mercier
- Publications: A hierarchical model to manage hardware topology in MPI applications - A Hierarchical Model to Manage Hardware Topology in MPI Applications
- URL: http://mpi-topology.gforge.inria.fr/

6.2. hwloc

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.

- Participants: Brice Goglin and Samuel Thibault
- Partners: Open MPI consortium - Intel - AMD
- Contact: Brice Goglin
- Publications: hwloc: a Generic Framework for Managing Hardware Affinities in HPC Applications - Managing the Topology of Heterogeneous Cluster Nodes with Hardware Locality (hwloc) - A Topology-Aware Performance Monitoring Tool for Shared Resource Management in Multicore Systems - Exposing the Locality of Heterogeneous Memory Architectures to HPC Applications - Towards the Structural Modeling of the Topology of next-generation heterogeneous cluster Nodes with hwloc - On the Overhead of Topology Discovery for Locality-aware Scheduling in HPC
- URL: http://www.open-mpi.org/projects/hwloc/

6.3. NetLoc

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.

- Participants: Brice Goglin, Clement Foyer and Cyril Bordage
- Contact: Brice Goglin
- Publications: netloc: Towards a Comprehensive View of the HPC System Topology - Netloc: a Tool for Topology-Aware Process Mapping
- URL: http://www.open-mpi.org/projects/netloc/

6.4. NewMadeleine

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.

- Participants: Alexandre Denis, Clement Foyer, Nathalie Furmento, Raymond Namyst and ADRIEN GUILBAUD
- Contact: Alexandre Denis
- Publications: NewMadeleine: a Fast Communication Scheduling Engine for High Performance Networks - Ordonnancement et qualité de service pour réseaux rapides - Improving Reactivity and Communication Overlap in MPI using a Generic I/O Manager - PIOMan : un gestionnaire d'entrées-sorties générique - A multithreaded communication engine for multicore architectures - A multicore-enabled multirail communication engine - About the interactions between communication and thread scheduling in clusters of multicore machines - An analysis of the impact of multi-threading on communication performance - A scalable and generic task scheduling system for communication libraries - A Generic and High Performance Approach for Fault Tolerance in Communication Library - A High-Performance Superpipeline Protocol for InfiniBand - A sampling-based approach

for communication libraries auto-tuning - High performance checksum computation for fault-tolerant MPI over InfiniBand - pioman: a Generic Framework for Asynchronous Progression and Multithreaded Communications - pioman: a pthread-based Multithreaded Communication Engine - Updating MadMPI to MPI-3: Remote Memory Access - Portage de StarPU sur la bibliothèque de communication NewMadeleine

- URL: http://pm2.gforge.inria.fr/newmadeleine/

6.5. PaMPA

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.

- Participants: Cécile Dobrzynski, Cedric Lachat and François Pellegrini
- Partners: Université de Bordeaux - CNRS - IPB
- Contact: François Pellegrini
- URL: http://project.inria.fr/pampa/

6.6. TreeMatch

KEYWORDS: Intensive parallel computing - High-Performance Computing - Hierarchical architecture - Placement
SCIENTIFIC DESCRIPTION: TreeMatch 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.

TreeMatch 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.

- Participants: Adele Villiermet, Emmanuel Jeannot, François Tessier, Guillaume Mercier and Pierre Celor
- Partners: Université de Bordeaux - CNRS - IPB
- Contact: Emmanuel Jeannot
- URL: http://treematch.gforge.inria.fr/

6.7. 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 FUNCTIONAL DESCRIPTION: Version 6.0 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...

- Participants: François Pellegrini, Sébastien Fourestier, Jun-Ho Her and Cédric Chevalier
- Partners: Université de Bordeaux - IPB - CNRS - Region Aquitaine
- Contact: François Pellegrini
- Publications: Process Mapping onto Complex Architectures and Partitions Thereof - Multi-criteria Graph Partitioning with Scotch - Adaptation au repartitionnement de graphes d'une méthode d'optimisation globale par diffusion - Contributions au partitionnement de graphes parallèle multi-niveaux - A parallelisable multi-level banded diffusion scheme for computing balanced partitions with smooth boundaries - PT-Scotch: A tool for efficient parallel graph ordering - Design and implementation of efficient tools for parallel partitioning and distribution of very large numerical problems - Improvement of the Efficiency of Genetic Algorithms for Scalable Parallel Graph Partitioning in a Multi-Level Framework - PT-Scotch : Un outil pour la renumérotation parallèle efficace de grands graphes dans un contexte multi-niveaux - PT-Scotch: A tool for efficient parallel graph ordering
- URL: http://www.labri.fr/~pelegrin/scotch/

6.8. disk-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.

- Authors: Guillaume Aupy and Julien Herrmann
- Contact: JULIEN HERRMANN
- Publications: Periodicity in optimal hierarchical checkpointing schemes for adjoint computations - Optimal Multistage Algorithm for Adjoint Computation
- URL: https://gitlab.inria.fr/adjoint-computation/disk-revolve-public

7. New Results

7.1. Checkpointing Strategies for Adjoint Computation on Hierarchical Platforms

The Adjoint Computation problem can be split in two phases: the forward phase where functions are successively evaluated on a particular input, and a backward phase computing the gradient descent. In the backward phase, the outputs of the forward phase are used* for the corresponding backward computation. On very large problems, all the forward outputs can not be kept in the memory at the same time, and one has to decide which output should be checkpointed and which output should be recomputed later on. The goal is to minimize the number of recomputation when reversing an Adjoint Computation Graph.

Griewank and Walther proved that, for a given number of available checkpoints with negligible writing and reading costs, the schedule that minimizes the amount of recomputation uses a binomial checkpointing strategy. We have designed an optimal algorithm to tackle the more general problem where we don't have only one level of memory with negligible access cost, but a hierarchical storage architecture. Each level of memory has its own size, writing and reading cost. The problem becomes more complex, since, not only we have to decide if an output should be checkpointed, be we have to decide in which level of the memory it should be kept. A trade-off must be found between the cost of memory accesses and that of recomputations.

We have designed an exact algorithm providing the optimal checkpointing strategy for a given Adjoint Computation Graph size and a description of the Hierarchical Platform; as well as heuristics. These algorithms can be found in the Software DISK-REVOLVE and a paper describing them is under writing process.

7.2. Modeling Non-Uniform Memory Access on Large Compute Nodes with the Cache-Aware Roofline Model

The trend of increasing the number of cores on-chip is enlarging the gap between compute power and memory performance. This issue leads to design systems with heterogeneous memories, creating new challenges for data locality. Before the release of those memory architectures, the Cache-Aware Roofline Model [33] (CARM) offered an insightful model and methodology to improve application performance with knowledge of the cache memory subsystem.

With the help of the HWLOC library, we are able to leverage the machine topology to extend the CARM for modeling NUMA and heterogeneous memory systems, by evaluating the memory bandwidths between all combinations of cores and NUMA nodes. The new Locality Aware Roofline Model [5] (LARM) scopes most contemporary types of large compute nodes and characterizes three bottlenecks typical of those systems, namely contention, congestion and remote access. We also designed a hybrid memory bandwidth model to better estimate the roof when heterogeneous memories are involved or when read and write bandwidths differ.

This work has been achieved in collaboration with the authors of the CARM from Universidade de Lisboa.

7.3. Cross Platform Classification for Detecting Locality Sensitivity and Selecting Data and Threads Placement Strategy

Individual nodes composing High Performance Computing (HPC) systems embed complex multicore and manycore processors. At this scale, compute tasks and data placement can double or halve execution times with respectively trivial are wise placements. While state of the art placement solutions can offer good performance improvements, they failed to set up as standards in supercomputers software stack. Current solutions are rather directed toward data or thread driven static policies. Among existing or promising future placement solutions a deep evaluation of applications response to these had yet to be done in order to wisely choose the best one.

With a set of 37 HPC representative applications, three different HPC processors, and 51 state of the art characterization metrics we built thousands models to evaluate applications response to data and threads placement policies. Thanks to a thorough methodology, our models were able to predict applications sensitivity to locality and their preferred placement policy both on new platforms and new applications. In the first case we were able to achieve more than 75% accuracy while preferred policy predictions approach optimal speedups in the second case.

This work was conducted using the PlaFRIM experimental testbed, in collaboration with Thomas Ropars from Laboratoire d'Informatique de Grenoble.

Several leads can be taken toward an extension of this work. For instance, predictions can be improved with benchmark directed learning. Models interpretation can also be furthered studied to refine the design of application characterization metrics.

7.4. Co-scheduling HPC workloads on cache-partitioned CMP platforms

Co-scheduling techniques are used to improve the throughput of applications on chip multiprocessors (CMP), but sharing resources often generates critical interferences.

In collaboration with ENS Lyon and Georgia Tech, we looked at the interferences in the last level of cache (LLC) and use the *Cache Allocation Technology* (CAT) recently provided by Intel to partition the LLC and give each co-scheduled application their own cache area. We considered m iterative HPC applications running concurrently and answer the following questions: (i) how to precisely model the behavior of these applications on the cache partitioned platform? and (ii) how many cores and cache fractions should be assigned to each application to maximize the platform efficiency? Here, platform efficiency is defined as maximizing the performance either globally, or as guaranteeing a fixed ratio of iterations per second for each application. Through extensive experiments using CAT, we demonstrated the impact of cache partitioning when multiple HPC application are co-scheduled onto CMP platforms. [13]

7.5. Memory Footprint of Locality Information on Many-Core Platforms

Exploiting the power of HPC platforms requires knowledge of their increasingly complex hardware topologies. Multiple components of the software stack, for instance MPI implementations or OpenMP runtimes, now perform their own topology discovery to find out the available cores and memory, and to better place tasks based on their affinities.

We studied the impact of this topology discovery in terms of memory footprint. Storing locality information wastes an amount of physical memory that is becoming an issue on many-core platforms on the road to exascale.

We demonstrated that this information may be factorized between processes by using a shared-memory region. Our analysis of the physical and virtual memories in supercomputing architectures showed that this shared region can be mapped at the same virtual address in all processes, hence dramatically simplifying the software implementation. [19]

Our implementation in HWLOC and Open MPI showed a memory footprint that does not increase with the number of MPI ranks per node anymore. Moreover the job launch time decreased by more than a factor of 2 on an Intel Knights Landing Xeon Phi and on a 96-core NUMA platform.

7.6. New abstraction to manage hardware topologies in MPI applications

Since the end of year 2016, we have been working on new abstractions and mechanisms that can allow the programmer to take advantage of the underlying hardware topology in their parallel applications developed in MPI. For instance, taking into account the intricate network/memory hierarchy can lead to substantial improvements in communication performance and reduce altogether the overall execution time of the application. However, it is important to find the relevant level of abstraction, as too much details are not usable practically because the programmer is not a a hardware specialist most of the time. Also, MPI being hardware-agnostic, it is important to find means to use the hardware specifics without being tied to a particular architecture or hardware design.

With these goals in mind, we proposed the HSPLIT (see Section 6.1) libray that implements a solution based on a well-known MPI concept, the *communicators* (that can be seen as groups of communicating processes) [7], [19]. With HSPLIT, each level in the hardware hierarchy is accessible through a dedicated communicator. In this way, the programmer can leverage the underlying hierarchy in their application quite simply . The current implementation of HSPLIT is based on both HWLOC and NETLOC.

This work led to the creation of a new active working group within the MPI Forum, coordinated and led by Inria.

Also, this work has lead to the joint developement of the Hippo software with the CERFACS. Thanks to this piece of software, hybrid OpenMP/MPI applications can leverage the underlying physical hierarchy in order to better place MPI processes and OpenMP threads. This is particularly useful in a context where the application is composed of several kernels that use their own placement and mapping policy for processes and threads to achieve the best performance. Thanks to HSPLIT and HWLOC, CERFACS is now able to write codes in a more portable fashion without to solely rely on interactions of OpenMP and MPI runtimes for mapping and binding of processes/threads management.

7.7. Scheduling Parallel Tasks under Multiple Resources: List Scheduling vs. Pack Scheduling

Scheduling in High-Performance Computing (HPC) has been traditionally centered around computing resources (e.g., processors/cores). The ever-growing amount of data produced by modern scientific applications start to drive novel architectures and new computing frameworks to support more efficient data processing, transfer and storage for future HPC systems. This trend towards data-driven computing demands the scheduling solutions to also consider other resources (e.g., I/O, memory, cache) that can be shared amongst competing applications. In this paper, we study scheduling HPC applications while exploring the availability of multiple resources that could impact their performance. The goal is to minimize the overall execution time, or makespan, for a set of moldable tasks under multi-resource constraints. Two scheduling paradigms, namely, list scheduling and pack scheduling, are compared through both theoretical analyses and experimental evaluation. Theoretically, we prove, for several algorithms falling in the two scheduling paradigms, tight approximation ratios that increase linearly with the number of resource types. As the complexity of the direct solutions grows exponentially with the number of resource types, we also design a strategy to indirectly solve the problem via a transformation to a single-resource problem, which can significantly reduce the algorithms' running times without compromising their approximation ratios. Experiments conducted on Intel Knights Landing with two types of resources (processor cores and high-bandwidth memory) and simulations designed on more resource types confirm the benefit of the transformation strategy and show that pack-based scheduling, despite having a slightly worse theoretical bound, offers a practically promising and easy-to-implement solution, especially when managing a large number of resources. [20]

7.8. Sizing Burst-Buffers efficiently

Burst-Buffers are high throughput, small size intermediate storage systems typically based on SSDs or NVRAM that are designed to be used as a potential buffer between the computing nodes of a supercomputer and its main storage system consisting of hard drives. Their purpose is to absorb the bursts of I/O that many HPC applications experience (for example for saving checkpoints or data from intermediate results). In this paper, we propose a probabilistic model for evaluating the performance of Burst-Buffers. From a model of application and a data management strategy, we build a Markov-chain-based model of the system, that allows us to quickly answer issues about dimensioning of the system: for a given set of applications, and for a given Burst-Buffers size and bandwidth, how often does the buffer overflow? We also provide extensive simulation results to validate our modeling approach. [12], [25]

7.9. Scheduling for Neurosciences

In this project in collaboration with the Vanderbilt University, we are interested in scheduling stochastic jobs (originating from Neuroscience applications) on a reservation-based platform. Specifically, we consider jobs whose execution time follows a known probability distribution. The platform is reservation-based, meaning that the user has to request fixed-length time slots. The cost depends on both the request duration and the actual execution time of the job. A reservation strategy is a sequence of increasing-length reservations, which are paid for until one of them allows the job to successfully complete. The goal is to minimize the total expected cost of the strategy. We provide some properties of the optimal solution, which we characterize up to the length of the first reservation. We evaluate these heuristics using two different platform models and cost functions: The first one targets a cloud-oriented platform (e.g., Amazon AWS) using jobs that follow a large number of usual probability distributions (e.g., Uniform, Exponential, LogNormal, Weibull, Beta), and the second one is based on interpolating traces from a real neuroscience application executed on an HPC platform. [14], [27]

7.10. Process Affinity, Metrics and Impact on Performance

Process placement, also called topology mapping, is a well-known strategy to improve parallel program execution by reducing the communication cost between processes. It requires two inputs: the topology of the target machine and a measure of the affinity between processes. In the literature, the dominant affinity measure is the communication matrix that describes the amount of communication between processes. The goal of this work is to study the accuracy of the communication matrix as a measure of affinity. We have done an extensive set of tests with two fat-tree machines and a 3d-torus machine to evaluate several hypotheses that are often made in the literature and to discuss their validity. First, we have checked the correlation between algorithmic metrics and the performance of the application. Then, we have checked whether a good generic process placement algorithm never degrades performance. And finally, we have seen whether the structure of the communication matrix can be used to predict gain [16].

7.11. Scheduling bi-colored-chains

In high performance computing, platform are shared by concurrent applications, each able to work with immense amount of data. As the file system is shared, we need to tackle congestion problems. One way to avoid increased I/O duration is to schedule the tasks with regards to their requests.

We proposed a theoretical model, bi-colored chains, that models applications with two alternating phases on distinct resources. After showing that minimizing the makespan with this model is a NP-complete problem in most cases. We studied particular cases, especially periodic applications and periodic schedule and provided approximation algorithms. This model will be developed in a PhD that started this fall, and enrich with practical data from simulations.

An extended intership report is available here: [31].

7.12. Experimenting task-based runtimes on a legacy Computational Fluid Dynamics code with unstructured meshes

Advances in high performance computing hardware systems lead to higher levels of parallelism and optimizations in scientific applications and more specifically in computational fluid dynamics codes. To reduce the level of complexity that such architectures bring while attaining an acceptable amount of the parallelism offered by modern clusters, the task-based approach has gained a lot of popularity recently as it is expected to deliver portability and performance with a relatively simple programming model. In this work, we have reviewed and presented the process of adapting part of Code Saturne, a legacy code at EDF R&D into a task-based form using the PARSEC (Parallel Runtime Scheduling and Execution Control) framework. We have first shown show the adaptation of our prime algorithm to a simpler form to remove part of the complexity of our code and then present its task-based implementation. We then have compared performance of various forms of our code and discuss the perks of task-based runtimes in terms of scalability, ease of incremental deployment in a legacy CFD code, and maintainability [8].

7.13. Progress threads placement for overlapping MPI non-blocking collectives using simultaneous multi-threading

Non-blocking collectives have been proposed so as to allow communications to be overlapped with computation in order to amortize the cost of MPI collective operations. To obtain a good overlap ratio, communications and computation have to run in parallel. To achieve this, different hardware and software techniques exists. Using dedicated cores for progress threads is one of them. However, some CPUs provide Simultaneous Multi-Threading, which is the ability for a core to have multiple hardware threads running simultaneously, sharing the same arithmetic units. We propose [18], [3] to use SMT to run progress threads to avoid dedicated cores allocation. We have run benchmarks on Haswell processors, using its Hyper-Threading capability, and get good results for both performance and overlap for inter-node communications. However, we have shown that Simultaneous Multi-Threading for intra-communications leads to bad performances due to contention on cache.

7.14. Dynamic placement of progress thread for overlapping MPI non-blocking collectives on manycore processor

To amortize the cost of MPI collective operations, non-blocking collectives have been proposed so as to allow communications to be overlapped with computation. Unfortunately, collective communications are more CPU-hungry than point-to-point communications and running them in a communication thread on a single dedicated CPU core makes them slow. On the other hand, running collective communications on the application cores leads to no overlap. To address these issues, we proposed [28], [17], [21], [3] an algorithm for tree-based collective operations that splits the tree between communication cores and application cores. To get the best of both worlds, the algorithm runs the short but heavy part of the tree on application cores, and the long but narrow part of the tree on one or several communication cores, so as to get a trade-off between overlap and absolute performance. We provided a model to study and predict its behavior and to tune its parameters. We implemented it in the MPC framework, which is a thread-based MPI implementation. We have run benchmarks on manycore processors such as the KNL and Skylake and got good results both in terms of performance and overlap.

7.15. Multi-criteria graph partitioning

The inclusion of multi-constraint graph partitioning algorithms in SCOTCH resulted in the obtainment of balanced multi-constraint partitions for a simulation software used in an industrial context [15]. This protype version is being transferred into the trunk of the SCOTCH package. Much of this year's software development has been devoted to the refactoring of the multi-threading management of the sequential version of the SCOTCH library.

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Grants with Industry

8.1.1. Intel

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.

8.1.2. Bull/Atos

Bull/ATOS granted the CIFRE PhD thesis on Nicolas Denoyelle on advanced memory hierarchies and new topologies.

8.1.3. EDF

With Yvan Fournier from EDF R&D we co-advise the PhD thesis of Benjamin Lorendeau under a CIFRE funding.

8.1.4. CEA

CEA/DAM granted the CIFRE PhD thesis of Hugo Taboada on non-blocking MPI collectives.

9. Partnerships and Cooperations

9.1. Regional Initiatives

9.1.1. CRA HPC Scalable Ecosystem, 2018-2021

2018 - 2021 (36 months)

Coordinator: Emmanuel AGULLO

Other partners: INRA, Institut Pprime, UPPA, Airbus, CEA, CATIE

Abstract: The goal is to design a unified runtime-system for numerical simulation at large-scale and with a large amount of data. We aim at contributing significantly to the convergence between HPC and BigData. TADAAM is involved in scheduling data access and managing communication efficiently on large-scale system.

9.2. National Initiatives

9.2.1. PIA ELCI, Environnement Logiciel pour le Calcul Intensif, 2014-2018

The ELCI PIA project is coordinated by BULL with several partners: CEA, Inria, SAFRAN, UVSQ.

This project aims to improve the support for numerical simulations and High Performance Computing (HPC) by providing a new generation software stack to control supercomputers, to improve numerical solvers, and pre- and post computing software, as well as programming and execution environment. It also aims at validating the relevance of these developments by demonstrating their capacity to deliver better scalability, resilience, modularity, abstraction, and interaction on some application use-cases. TADAAM is involved in WP1 and WP2 ELCI Work Packages. Emmanuel JEANNOT is the Inria representative in the ELCI steering committee.

9.2.2. ANR

ANR SATAS SAT as a Service (http://www.agence-nationale-recherche.fr/Project-ANR-15-CE40-0017).

AP générique 2015, 01/2016 - 12/2019 (48 months)

Coordinator: Laurent Simon (LaBRI)

Other partners: CRIL (Univ. Artois), Inria Lille (Spirals)

Abstract: The SATAS project aims to advance the state of the art in massively parallel SAT solving. The final goal of the project is to provide a "pay as you go" interface to SAT solving services and will extend the reach of SAT solving technologies, daily used in many critical and industrial applications, to new application areas, which were previously considered too hard, and lower the cost of deploying massively parallel SAT solvers on the cloud.

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 AUPY (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.

9.2.3. 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.

9.2.4. IPL - Inria Project Lab

High-Performance computing and BigData

Participants: Guillaume Aupy, Emmanuel Jeannot, Julien Herrmann and Nicolas Vidal.

HPC and Big Data evolved with their own infrastructures (supercomputers versus clouds), applications (scientific simulations versus data analytics) and software tools (MPI and OpenMP versus Map/Reduce or Deep Learning frameworks). But Big Data analytics is becoming more compute-intensive (thanks to deep learning), while data handling is becoming a major concern for scientific computing. The goal of this HPC-BigData IPL is to gather teams from the HPC, Big Data and Machine Learning (ML) areas to work at the intersection between these domains. Research is organized along three main axes: high performance analytics for scientific computing applications, high performance analytics for big data applications, infrastructure and resource management

9.3. European Initiatives

9.3.1. Collaborations in European Programs, Except FP7 & H2020

NESUS: Network for Ultrascale Computing (http://www.nesus.eu)

Program: COST

Project acronym: NESUS

Project title: Network for Ultrascale Computing

Duration: April 2014 - April 2018

Coordinator: University Carlos III de Madrid

Other partners: more than 35 countries

Abstract: Ultrascale systems are envisioned as large-scale complex systems joining parallel and distributed computing systems that will be two to three orders of magnitude larger that today's systems. The EU is already funding large scale computing systems research, but it is not coordinated across researchers, leading to duplications and inefficiencies. The goal of the NESUS Action is to establish an open European research network targeting sustainable solutions for ultrascale computing aiming at cross fertilization among HPC, large scale distributed systems, and big data management. The network will contribute to glue disparate researchers working across different areas and provide a meeting ground for researchers in these separate areas to exchange ideas, to identify synergies, and to pursue common activities in research topics such as sustainable software solutions (applications and system software stack), data management, energy efficiency, and resilience. Some of the most active research groups of the world in this area are members of this proposal. This Action will increase the value of these groups at the European-level by reducing duplication of efforts and providing a more holistic view to all researchers, it will promote the leadership of Europe, and it will increase their impact on science, economy, and society. Emmanuel JEANNOT is the vice-chair of this Action.

9.3.2. Collaborations with Major European Organizations

Partner 1: INESC-ID, Lisbon, (Portugal)

Subject 1: Application modeling for hierarchical memory system

Partner 2: University Carlos III de Madrid, (Spain)

Subject 2: I/O Scheduling

9.4. International Initiatives

9.4.1. Inria International Labs

Joint-Lab on Extreme Scale Computing (JLESC):

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.

9.4.2. Inria International Partners

9.4.2.1. Informal International Partners

Partner 1: Argonne National Lab

Subject 1: Binomial Checkpointing Strategies for Machine Learning (recipient of a FACCTS grant, 2018-2020)

Partner 2: Vanderbilt University

Subject 2: Scheduling for Neurosciences

Partner 3: ICL at University of Tennessee

Subject 3: 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

9.5. International Research Visitors

9.5.1. Visits of International Scientists

- Sri Hari Krishna Narayanan (Argonne National Lab), April 2018
- Sri Hari Krishna Narayanan(Argonne National Lab), Jan Hückelheim and Navjot Kukreja (Imperial College), June 2018
- Jesus Carretero, David E. Singh (Univ. Carlos III, Madrid), Dean Chester (Univ. of Warwick), Raymond Nou (BSC), October 2018
- Paul Hovland (Argonne National Lab) and Navjot Kukreja (Imperial College), December 2018

9.5.2. Visits to International Teams

9.5.2.1. Research Stays Abroad

Valentin HONORÉ visited Ana Gainaru, Padma Raghavan, and Hongyang Sun at Vanderbilt University (USA) for three months in the context of the project "Scheduling for Neurosciences".

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events Organisation

10.1.1.1. General Chair, Scientific Chair

In the context of ANR Dash, Guillaume AUPY organized a workshop on I/O in Europe https://project.inria.fr/dash/events/. He was the co-chair of the FTS workshop at Cluster 2018.

10.1.1.2. Member of the steering committee

Emmanuel JEANNOT is member of the steering committee of Euro-Par and the Cluster international conference.

10.1.1.3. Member of the Organizing Committees

- Guillaume AUPY was a member of the Organizing Committee of Per3S 2019.

10.1.2. Scientific Events Selection

10.1.2.1. Chair of Conference Program Committees

- Guillaume AUPY was workshop Chair at SC 2018, Inclusivity Vice-Chair at SC 2018, Algorithm track vice-chair at ICPP 2018.
- Emmanuel JEANNOT was the program chair of the COLOC workshop.

10.1.2.2. Member of the Conference Program Committees

- Emmanuel JEANNOT was member of the program committee of IPDPS 2019, HPML 2018, Heteropar 2018, Compas 2018.
- Brice GOGLIN was a member of the program committee of EuroMPI 2018, SuperComputing 2018 (posters), and of the COLOC, ExaComm and ROME Workshops.
- Alexandre DENIS was a member of the program committee of CCGrid 2018, CCGrid 2019, HiPC 2018, SC 2018 (workshops).
- Guillaume AUPY was a member of the program committee of SuperComputing 2018 (Doctoral Showcase), HPML 2018, CEBDA 2018 (IPDPS workshop).
- Guillaume MERCIER was a member of the programm committe of SuperComputing 2018 (Performance Measurement, Modeling, and Tools Track), EuroMPI 2018, CCGrid 2018, HPCS 2018 and Compas 2018

10.1.2.3. Reviewer

- Alexandre DENIS was a reviewer for IPDPS 2018.

10.1.3. Journal

10.1.3.1. Member of the Editorial Boards

- Emmanuel JEANNOT is associate editor of the International Journal of Parallel, Emergent & Distributed Systems (IJPEDS).
- Emmanuel JEANNOT was an invited editor for the Special Issue of *Concurrency and Computation: Practice and Experience* for best papers of HCW 2018.

10.1.3.2. Reviewer - Reviewing Activities

- Guillaume AUPY was a reviewer for IEEE TPDS, Cluster Computing.
- Emmanuel JEANNOT was a reviewer for IEEE TPDS, Journal of Computational Science.
- Alexandre DENIS was a reviewer for IEEE TPDS.

10.1.4. Invited Talks

- Guillaume AUPY was invited to give a talk at Per3S 2018.
- Guillaume AUPY and Emmanuel JEANNOT were invited to give a talk at CCDSC 2018.

10.1.5. Scientific Expertise

- Emmanuel JEANNOT have been reviewer for the PRACE 6IP call (WP8).
- Emmanuel JEANNOT was a member of the hiring committee of an Inria junior researcher position at Bordeaux.
- François PELLEGRINI has been appointed as co-pilot of the project group on free/libre software at *Comité pour la science ouverte* (CoSO), an arm of the CODORNUM of the French Ministry of Higher Education and Research.
- François PELLEGRINI was a member of the former *Comité d'Orientation sur l'Information Scientifique et Technique* (CORIST) of *Institut National de la Recherche Agronomique* (INRA).
- François PELLEGRINI participated in a roundtable on "*the societal impact of digital identity*" during the *Assises de l'identité numérique* organized by the French ministries of the Interior and Justice, and the State Secretariat for digital issues.
- François PELLEGRINI was heard by members of the Law Commission of the French Senate, on e-voting.
- François PELLEGRINI was a member of the hiring committee for an assistant professor position at Université de Pau et des Pays de l'Adour.

10.1.6. Research Administration

- Emmanuel JEANNOT is deputy head of science of the Inria Bordeaux research center.
- Emmanuel JEANNOT is member of the Inria evaluation committee
- Emmanuel JEANNOT is member of LaBRI scientific coouncil and head of the Satanas team.
- Alexandre DENIS is head of the Inria Bordeaux CUMI-R (IT users committee).
- Brice GOGLIN and Guillaume MERCIER are elected members of the research centre committee.

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). TADAAM also created of a new working group in the MPI Forum, dedicated to hardware topologies management and currently leads this working group. The HSPLIT proposal in currently under early discussions for submission to the forum and eventual inclusion in the MPI standard.

10.2. Teaching - Supervision - Juries

10.2.1. 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 algorithmics and C programming to advanced topics such as probabilities and statistics, scheduling, computer architecture, operating systems, parallel programming and high-performance runtime systems, as well as software law and personal data.

Brice GOGLIN participated in the section about fondamentals of computer science in the MOOCs *Informatique et Création Numérique* and *Sciences Numériques et Technologie* which focus at bringing basics about computer science to high-school teachers.

François PELLEGRINI gave a doctoral course at the University of Luxembourg on "*Law and freedom(s) in the digital age*". He also teached at Télécom Sud Paris and École Nationale de la Magistrature.

10.2.2. Supervision

PhD: Nicolas DENOYELLE, advanced memory hierarchies and new topologies, defended on November 6th 2018. Advisor: Brice GOGLIN and Emmanuel JEANNOT.

PhD: Hugo TABOADA, *Recouvrement des Collectives MPI Non-bloquantes sur Processeur Many-core*, defended on 11 December 2018. Advisors: Alexandre DENIS and Emmanuel JEANNOT.

PhD in progress: Benjamin LORENDEAU, new programming models and optimization of Code Saturn, started in 2015. Advisors: Yvan FOURNIER and Emmanuel JEANNOT.

PhD in progress: Valentin Honoré, Partitioning Strategies for high throughput Applications, started in November 2017. Advisors: Guillaume AUPY and Brice GOGLIN.

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

PhD started: Nicolas VIDAL, IO scheduling strategies, started in October 2018. Advisors: Guillaume AUPY and Emmanuel JEANNOT.

10.2.3. Juries

Guillaume AUPY was a reviewer of the mid-PhD defense of Massinissa Ait Aba (CEA, supervisors: Alix Munier, Safia Kedad Sidhoum).

Emmanuel JEANNOT was member of the Ph.D defense jury of:

- Yann Barsamian (University of Strasbourg, Reviewer)
- Loic Pottier (University of Lyon, ENS Lyon, Reviewer)
- Philippe Virouleau (University of Grenoble-Alpes, Reviewer)

François PELLEGRINI was member of the Ph.D defense jury of:

- Grégoire Pichon (University of Bordeaux)

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 Inria Research Centre of Bordeaux. He organized several popularization activities involving colleagues.

Guillaume AUPY co-organized (with Marthe Bonamy) a 2-day visit of Inria & Labri for undergrad students from ENS Lyon.

10.3.2. Education

- Brice GOGLIN was involved in the MOOC *Informatique et Création Numérique* which focuses at bringing basics about computer science to high-school teachers. He answered numerous questions on the forum. More than 19 000 people registered to the course, and more than 1 500 successfully finished it.
- Brice GOGLIN was involved in the building of the MOOC *Sciences Numériques et Technologie* which focus at bringing basics about computer science to high-school teachers.
- François PELLEGRINI, created a set of ten animated short videos on the digital revolution and its consequences, for high-school pupils and freshman students (Pix/C2i), in collaboration with the services of Université de Bordeaux.

10.3.3. Interventions

- Valentin HONORÉ and Brice GOGLIN went to the St Genes middle school in Bordeaux in March for *Semaine des Maths* to give hands-on sessions about basics of algorithmics and computer science.
- TADAAM presented its research to the general public during the 10th anniversary of the research centre on September 27th.
- Emmanuel JEANNOT was the roundtable presenter of the topic HPC and scientific computing at the Inria Bordeaux 10 years event on September 27th.
- Valentin HONORÉ and Brice GOGLIN presented TADAAM research during the research centre open day on October 13th.
- Guillaume AUPY, Valentin HONORÉ, Nicolas VIDAL and Brice GOGLIN gave seminars and hands-on session about computer science to schools attending *Fete de la Science* in October.
- Guillaume AUPY went to the Sainte-Foy-La-Grande middle school in the context of Maths-en-Jeans to talk about finding a way to share messages in class and how it relates to the internet.
- Brice GOGLIN introduced research, research carriers, high performance computing and data centers to middle-school interns on December 17th.
- François PELLEGRINI delivered a conference entitled "*Tous pirates ?*", at the National Theater of Bordeaux-Aquitaine (TNBA), in relation with the play of same name created by the Traverse and OS'O artist collectives.
- François PELLEGRINI participated in a conference and roundtable "*Ingénieurs, éthique et valeurs face à l'industrie 4.0*" organized by Fondation Anthony Mainguené at École Nationale Supérieure des Arts et Métiers.
- François PELLEGRINI was member of the jury during a fake trial of a self-driving artificial intelligence (*Carambolage du siècle*") at the Appeal Court of Paris, during the *Nuit du droit*.
- François PELLEGRINI delivered a conference on the digital revolution at Le Bar Commun during the *Week of digital Freedoms*.

11. Bibliography

Major publications by the team in recent years

[1] R. BARAT, C. CHEVALIER, F. PELLEGRINI. *Multi-criteria Graph Partitioning with Scotch*, in "SIAM Workshop on Combinatorial Scientific Computing", Bergen, Norway, F. MANNE, P. SANDERS, S. TOLEDO (editors), Proceedings of the Seventh SIAM Workshop on CSC, Society for Industrial and Applied Mathematics, June 2018, pp. 66-75 [*DOI :* 10.1137/1.9781611975215.7], https://hal.inria.fr/hal-01968358

Publications of the year

Doctoral Dissertations and Habilitation Theses

[2] N. DENOYELLE. *From Software Locality to Hardware Locality in Shared Memory Systems with NUMA and Heterogenous Memory*, Université de Bordeaux, November 2018, https://tel.archives-ouvertes.fr/tel-01917364

[3] H. TABOADA. *MPI Non-Blocking Collective Overlap on Manycore Processor*, Université de bordeaux, December 2018, https://hal.archives-ouvertes.fr/tel-01960415

Articles in International Peer-Reviewed Journals

[4] G. AUPY, A. BENOIT, S. DAI, L. POTTIER, P. RAGHAVAN, Y. ROBERT, M. SHANTHARAM. *Co-scheduling Amdahl applications on cache-partitioned systems*, in "International Journal of High Performance Computing Applications", 2018, vol. 32, n° 1, pp. 123-138 [*DOI :* 10.1177/1094342017710806], https://hal.inria.fr/hal-01968422

[5] N. DENOYELLE, B. GOGLIN, A. ILIC, E. JEANNOT, 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", 2019 [*DOI :* 10.1109/TPDS.2018.2883056], https://hal.inria.fr/hal-01924951

[6] Y. GEORGIOU, E. JEANNOT, G. MERCIER, A. VILLIERMET. *Topology-Aware Job Mapping*, in "International Journal of High Performance Computing Applications", January 2018, vol. 32, n° 1, pp. 14-27 [*DOI :* 10.1177/1094342017727061], https://hal.inria.fr/hal-01621325

[7] B. GOGLIN, E. JEANNOT, F. MANSOURI, G. MERCIER. *Hardware topology management in MPI applications through hierarchical communicators*, in "Parallel Computing", August 2018, vol. 76, pp. 70 - 90 [*DOI :* 10.1016/J.PARCO.2018.05.006], https://hal.inria.fr/hal-01937123

[8] E. JEANNOT, Y. FOURNIER, B. LORENDEAU. *Experimenting task-based runtimes on a legacy Computational Fluid Dynamics code with unstructured meshes*, in "Computers and Fluids", 2018, vol. 173, pp. 51-58 [*DOI :* 10.1016/J.COMPFLUID.2018.03.076], https://hal.inria.fr/hal-01901975

[9] F. PELLEGRINI. *Portability of data and services*, in "Revue française d'administration publique", 2018, vol. 167, n° 3, pp. 513-523 [*DOI :* 10.3917/RFAP.167.0513], https://hal.inria.fr/hal-01975442

[10] S. SEO, A. AMER, P. BALAJI, C. BORDAGE, G. BOSILCA, A. BROOKS, P. CARNS, A. CASTELLO, D. GENET, T. HÉRAULT, S. IWASAKI, P. JINDAL, L. V. KALÉ, S. KRISHNAMOORTHY, J. LIFFLANDER, H. LU, E. MENESES, M. SNIR, Y. SUN, K. TAURA, P. BECKMAN. *Argobots: A Lightweight Low-Level Threading and Tasking Framework*, in "IEEE Transactions on Parallel and Distributed Systems", March 2018, vol. 29, n° 3, pp. 512 - 526 [*DOI :* 10.1109/TPDS.2017.2766062], https://hal.inria.fr/hal-01887586

Articles in Non Peer-Reviewed Journals

[11] J. G. BARBOSA, E. JEANNOT. *Foreword to the Special Issue of the Twenty Sixth International Heterogeneity in Computing Workshop (HCW) and to the Fifteenth International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms (HeteroPar)*, in "Concurrency and Computation: Practice and Experience", 2018, 2 p. , editorial [*DOI :* 10.1002/CPE.5007], https://hal.inria.fr/hal-01903118

International Conferences with Proceedings

[12] G. AUPY, O. BEAUMONT, L. EYRAUD-DUBOIS. *What Size Should your Buffers to Disks be?*, in "International Parallel and Distributed Processing Symposium (IPDPS)", Vancouver, Canada, IEEE, May 2018 [*DOI :* 10.1109/IPDPS.2018.00075], https://hal.inria.fr/hal-01623846

[13] G. AUPY, A. BENOIT, B. GOGLIN, L. POTTIER, Y. ROBERT. *Co-scheduling HPC workloads on cache-partitioned CMP platforms*, in "IEEE Cluster 2018", Belfast, United Kingdom, Proceedings the 20th IEEE Cluster Conference, September 2018, pp. 335-345, https://hal.inria.fr/hal-01874154

[14] G. AUPY, A. GAINARU, V. HONORÉ, P. RAGHAVAN, Y. ROBERT, H. SUN. *Reservation Strategies for Stochastic Jobs*, in "IPDPS 2019 - 33rd IEEE International Parallel and Distributed Processing Symposium", Rio de Janeiro, Brazil, May 2019, pp. 1-10, https://hal.inria.fr/hal-01968419

[15] R. BARAT, C. CHEVALIER, F. PELLEGRINI. *Multi-criteria Graph Partitioning with Scotch*, in "SIAM Workshop on Combinatorial Scientific Computing", Bergen, Norway, F. MANNE, P. SANDERS, S. TOLEDO (editors), Proceedings of the Seventh SIAM Workshop on CSC, Society for Industrial and Applied Mathematics, June 2018, pp. 66-75 [*DOI :* 10.1137/1.9781611975215.7], https://hal.inria.fr/hal-01968358

[16] C. BORDAGE, E. JEANNOT. *Process Affinity, Metrics and Impact on Performance: an Empirical Study*, in "18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (IEEE/ACM CCGrid)", Washington DC, United States, May 2018, https://hal.inria.fr/hal-01901988

[17] A. DENIS, J. JAEGER, E. JEANNOT, M. PÉRACHE, H. TABOADA. *Dynamic Placement of Progress Thread for Overlapping MPI Non-Blocking Collectives on Manycore Processor*, in "EURO-PAR 2018 : 24th International European Conferenceon Parallel and Distributed Computing", Turin, Italy, Euro-Par 2018: Parallel Processing, Springer, August 2018, https://hal.archives-ouvertes.fr/hal-01888255

[18] A. DENIS, J. JAEGER, H. TABOADA. *Progress Thread Placement for Overlapping MPI Non-Blocking Collectives using Simultaneous Multi-Threading*, in "COLOC : 2nd workshop on data locality, in conjuction with EURO-PAR 2018", Turin, Italy, Euro-Par 2018: Parallel Processing Workshops, LNCS 11339 proceedings, August 2018, https://hal.archives-ouvertes.fr/hal-01888257

[19] B. GOGLIN. *Memory Footprint of Locality Information on Many-Core Platforms*, in "6th Workshop on Runtime and Operating Systems for the Many-core Era (ROME 2018), held in conjunction with IPDPS", Vancouver, BC, Canada, IEEE, May 2018, 10 p. [*DOI :* 10.1109/IPDPSW.2018.00201], https://hal.inria.fr/hal-01644087

[20] H. SUN, R. ELGHAZI, A. GAINARU, G. AUPY, P. RAGHAVAN. *Scheduling Parallel Tasks under Multiple Resources: List Scheduling vs. Pack Scheduling*, in "IPDPS 2018 - 32nd IEEE International Parallel & Distributed Processing Symposium", Vancouver, Canada, May 2018 [*DOI :* 10.1109/IPDPS.2018.00029], https://hal.inria.fr/hal-01951412

Conferences without Proceedings

[21] H. TABOADA. *Recouvrement des Collectives MPI Non-bloquantes sur Processeur Manycore*, in "Compas 2018: conférence d'informatique en Parallélisme, Architecture et Système", Toulouse, France, July 2018, https://hal.archives-ouvertes.fr/hal-01888249

Scientific Books (or Scientific Book chapters)

[22] G. AUPY, Y. ROBERT. *Scheduling for Fault-Tolerance: An Introduction*, in "Topic in parallel and distributed computing: Enhancing the Undergraduate Curriculum: Performance, Concurrency, and Programming on Modern Platforms", Springer International Publishing, September 2018, pp. 143-170, https://hal.inria.fr/hal-01968454

Books or Proceedings Editing

[23] G. AUPY, X. TANG (editors). *Parallel and distributed algorithms*, Wiley, April 2018, 3 p. [*DOI :* 10.1002/CPE.4663], https://hal.inria.fr/hal-01971630

[24] D. BLANCO HERAS, L. BOUGÉ, E. JEANNOT, R. SAKELLARIOU, R. M. BADIA, J. G. BARBOSA, L. RICCI, S. L. SCOTT, S. LANKES, J. WEIDENDORFER (editors). *Euro-Par 2017 International Workshops, Santiago de Compostela, Spain, August 28-29, 2017, Revised Selected Papers*, Lecture Notes in Computer Science, Springer, Santiago de Compostella, Spain, 2018, vol. 10659 [*DOI :* 10.1007/978-3-319-75178-8], https://hal.inria.fr/hal-01962797

Research Reports

[25] G. AUPY, O. BEAUMONT, L. EYRAUD-DUBOIS. *Sizing and Partitioning Strategies for Burst-Buffers to Reduce IO Contention*, Inria, October 2018, n° RR-9213, https://hal.inria.fr/hal-01904032

[26] G. AUPY, A. BENOIT, B. GOGLIN, L. POTTIER, Y. ROBERT. *Co-scheduling HPC workloads on cache-partitioned CMP platforms*, Inria, February 2018, n° RR-9154, https://hal.inria.fr/hal-01719728

[27] G. AUPY, A. GAINARU, V. HONORÉ, P. RAGHAVAN, Y. ROBERT, H. SUN. *Reservation Strategies for Stochastic Jobs (Extended Version)*, Inria & Labri, Univ. Bordeaux ; Department of EECS, Vanderbilt University, Nashville, TN, USA ; Laboratoire LIP, ENS Lyon & University of Tennessee Knoxville, Lyon, France, October 2018, n° RR-9211, pp. 1-37, https://hal.inria.fr/hal-01903592

[28] A. DENIS, J. JAEGER, E. JEANNOT, M. PÉRACHE, H. TABOADA. *Dynamic Placement of Progress Thread for Overlapping MPI Non-Blocking Collectives on Manycore Processor*, Inria Bordeaux Sud-Ouest, March 2018, n° RR-9160, pp. 1-12, https://hal.inria.fr/hal-01741787

[29] B. GOGLIN, E. JEANNOT, F. MANSOURI, G. MERCIER. *A Hierarchical Model to Manage Hardware Topology in MPI Applications*, Inria Bordeaux Sud-Ouest ; Bordeaux INP ; LaBRI - Laboratoire Bordelais de Recherche en Informatique, March 2018, n° RR-9077, 32 p. , https://hal.inria.fr/hal-01538002

[30] H. SUN, R. ELGHAZI, A. GAINARU, G. AUPY, P. RAGHAVAN. *Scheduling Parallel Tasks under Multiple Resources: List Scheduling vs. Pack Scheduling*, Inria Bordeaux Sud-Ouest, January 2018, n° RR-9140, https://hal.inria.fr/hal-01681567

[31] N. VIDAL. *Scheduling bi-colored chains*, Ecole Normale Supérieure de Lyon - ENS LYON, June 2018, https://hal.inria.fr/hal-01944993

Scientific Popularization

[32] F. PELLEGRINI, V. ANDRÉ. *L'ère du fichage généralisé : Identification biométrique et contrôle social*, in "Le Monde Diplomatique", April 2018, n^{o} 769, 3 p. , Version abrégée de https://hal.inria.fr/hal-01492431, https://hal.inria.fr/hal-01756475

References in notes

[33] A. ILIC, F. PRATAS, L. SOUSA. *Cache-aware Roofline model: Upgrading the loft*, in "IEEE Computer Architecture Letters", 2014, vol. 13, n^{o} 1, pp. 21–24