Activity Report 2016

Team TADAAM

Topology-Aware System-Scale Data Management for High-Performance Computing

Inria teams are typically groups of researchers working on the definition of a common project, and objectives, with the goal to arrive at the creation of a project-team. Such project-teams may include other partners (universities or research institutions).

RESEARCH CENTER
Bordeaux - Sud-Ouest

THEME
Distributed and High Performance Computing

Table of contents

Team TADAAM

Creation of the Team: 2015 January 01

Keywords:

Computer Science and Digital Science:

1.1.1. - Multicore
1.1.2. - Hardware accelerators (GPGPU, FPGA, etc.)
1.1.3. - Memory models
1.1.4. - High performance computing
1.1.5. - Exascale
1.2. - Networks
2.1.7. - Distributed programming
2.2.2. - Memory models
2.2.3. - Run-time systems
2.6.1. - Operating systems
2.6.2. - Middleware
3.1.3. - Distributed data
6.2.7. - High performance computing
7.1. - Parallel and distributed algorithms
7.3. - Optimization
7.9. - Graph theory

Other Research Topics and Application Domains:

6.3.2. - Network protocols
6.5. - Information systems
9.4.1. - Computer science

1. Members

Research Scientists

Emmanuel Jeannot [Team leader, Inria, Senior Researcher, HDR]
Guillaume Aupy [Inria, Researcher, from Dec. 2016]
Alexandre Denis [Inria, Researcher]
Brice Goglin [Inria, Researcher, HDR]

Faculty Members

Guillaume Mercier [INP Bordeaux, Associate Professor]
François Pellegrini [Univ. Bordeaux, Professor, HDR]

Engineers

Clément Foyer [Inria, from Feb. 2016]
Cedric Lachat [Inria]
François Tessier [Inria, until Jan. 2016]

PhD Students

Remi Barat [CEA]
Raphaël Blanchard [ONERA, until Oct. 2016]
Nicolas Denoyelle [Bull, granted by CIFRE]

Benjamin Lorendeau [EDF, granted by CIFRE]
Romain Prou [Inria, until Oct. 2016]
Hugo Taboada [CEA]
Adèle Villiermet [Inria]

Post-Doctoral Fellows
Farouk Mansouri [Inria]
Cyril Bordage [Inria]

Visiting Scientist
Juan Luis Garcìa Zapata [University of Extremadura, from Sep. 2016 to Nov. 2016]

Administrative Assistants
Cecile Boutros [Inria]
Sylvie Embolla [Inria]

Others
Arnaud Bardoux [Intern from University of Strasbourg, from Apr. 2016 to Sep. 2016]
Ahmad Boissetri Binzagr [Intern from University of Bordeaux, from May 2016 to Jul. 2016]
Francois Candela [Inria, Intern from University of Bordeaux, from May 2016 to Jul. 2016]
Paul Jeanmaire [Intern from ENS Cachan, from Jun. 2016 to Jul. 2016]

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.

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.

[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

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

The NETLOC (See Section 6.1) tools have been run on one of the largest European supercomputers (the TGCC/Genci CURIE machine) and successfully modeled its 5200 nodes and its interconnection network (more than 800 switches). This is a joint work with CEA and the COLOC European project.

6. New Software and Platforms

6.1. NetLoc

Network Locality
FUNCTIONAL DESCRIPTION

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

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 application 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. NETLOC is distributed within HWLOC releases starting with HWLOC 2.0.

- Participants: Cyril Bordage and Brice Goglin
- Contact: Brice Goglin
- URL: http://www.open-mpi.org/projects/netloc/

6.2. NewMadeleine

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 Mad-MPI fully supports the `MPI_THREAD_MULTIPLE` multi-threading level.

- Participants: Alexandre Denis, Nathalie Furmento, Raymond Namyst and Clement Foyer
- Contact: Alexandre Denis
- URL: http://pm2.gforge.inria.fr/newmadeleine/

6.3. 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 43 millio elements to more than 1 billion elements 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.

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

6.4. SCOTCH

KEYWORDS: High-performance computing - Graph algorithms - Domain decomposition - Static mapping - Mesh partitioning - Sparse matrix ordering

SCIENTIFIC DESCRIPTION

SCOTCH is a software package and libraries for sequential and parallel graph partitioning, static mapping and clustering; sequential mesh and hypergraph partitioning; and sequential and parallel sparse matrix block ordering.

Its main use is to subdivise a scientific problem, expressed as a graph, into a set of subproblems as independent as possible from each other (in terms of connecting edges).

FUNCTIONAL DESCRIPTION

SCOTCH takes the form of a set of libraries, plus additional standalone programs. The sequential and parallel libraries provide a set of interfaces to describe centralized and distributed graphs to partition, the target architectures to map onto, the resulting centralized and distributed mapping and ordering structures, etc. SCOTCH takes advantage of Posix threads, and its parallel version, PT-SCOTCH, uses the MPI interface.

- Participants: François Pellegrini, Cédric Lachat, Rémi Barat and Cédric Chevalier
- Partners: CNRS - IPB - Region Aquitaine
- Contact: François Pellegrini
- URL: http://www.labri.fr/~pelegrin/scotch/

6.5. TreeMatch

KEYWORDS: Intensive parallel computing - High-Performance Computing - Hierarchical architecture - Placement

SCIENTIFIC DESCRIPTION

TreeMatch provides a permutation of the processes to the 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.

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

6.6. hwloc

Hardware Locality
KEYWORDS: HPC - Topology - Open MPI - Affinities - GPU - Multicore - NUMA - 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: AMD - Intel - Open MPI consortium
- Contact: Brice Goglin
- URL: http://www.open-mpi.org/projects/hwloc/

7. New Results

7.1. Network Modeling

NETLOC (see Section 6.1) is a tool in HWLOC to discover the network topology. Our first work with NETLOC was to redesign it to be more efficient and more adapted to the needs. The code was cleaned and some dependencies were removed. We have added a display tool, that is able to show a network topology in a web browser where a user can interact with. It ran on one of the largest European supercomputer (the TGCC/Genci CURIE machine) and successfully modeled its 5200 nodes and its interconnection network (more than 800 switches).

Moreover, it is now possible to interact with Scotch from netloc. The first feature is to export a network topology, or even the current available topology given by the resource manager, into a SCOTCH architecture. Conversely, we can use SCOTCH tools in NETLOC for building a process mapping based on resources found by NETLOC and a process graph describing communications between processes. Tests conducted on a stencil mini-app have shown that the benefits are real and still needs more work.

7.2. Communication and computation overlap

To amortize the cost of communication in HPC application, programmers want to overlap communications with computation. To do so, they assume non-blocking MPI communications will progress in background. NewMadeleine, our communication library, is actually able to make communication progress in background so as to actually have overlap happen. However, not all MPI implementations are able to overlap communication and computation.

We have proposed [8] a benchmark to measure what really happens when trying to overlap non-blocking point-to-point communications with computation. The benchmark measures how much overlap happen in various cases: sender-side, receiver-side, datatypes likely to be offloaded onto NIC or not, multi-threaded computation, multi-threaded communication or not. We have benchmarked a wide panel of MPI libraries and hardware platforms, and thanks to low-level traces, explained the results.

7.3. Topology Aware Performance Monitoring

A tool has been developed to abstract performance metrics and map them onto the HWLOC (see Section 6.6) topology model of the system. During the year 2016, the tool has been entirely rewritten to release a more meaningful and stable programming abstraction, with off the shelf performance abstraction plugins and raw performance acquisition plugin [16]. A special effort has been carried out on output presentation by extending lstopo tool from hwloc into a library embedded in the monitoring tool to display performance metrics on the system topology. Another backend using R has also been developed for the purpose of post-mortem analysis and model extraction from abstract metrics of the topology.

7.4. Locality Aware Roofline Model

The years 2016 marked the achievement of our extension of the famous Cache Aware Roofline Model(CARM) and the associate tool. The latter model targets deep plateform and application analysis on multicore processors. Its model consist into a two-dimensions plane bound by several machine ceils and representative of scientific application workloads. Our extension validate the use of the CARM on emerging processors with heterogeneous memory subsystem, and extend the CARM methodology to encompass interconnection network, thus, enabling full modeling of shared memory systems [17]. This work is a collaboration with the INESC-ID research center under the NESUS project.

7.5. Performance Analysis of Electromagnetic Field Application on Large SMP Node

In the scope of the COLOC project we worked on understanding scalability issues of the efield application on a large shared memory system. Our analysis with above mentionned tools highlighted a potential bandwidth bottleneck. This problem can usually be tackled by the mean of threads and data mapping on respectively the machine cores and the memories. Unfortunately, those techniques can't be applied with this (closed source) application since the system does not allow to monitor memory accesses and traffic on the system.

7.6. Structural Modeling of Heterogeneous Memory Architectures

HWLOC (see Section 6.6) is the de facto standard tool for gathering information of parallel platform topologies. The advent of new memory architecture, with high-bandwidth and/or non-volatile memories cause the memory management subsytem complexity to increase. Indeed, besides taking care of allocating data buffers locally, developers also have to choose between different local memories with different performance and persistence characteristics. Moreover, the operating systems still cannot expose the full details about these technologies to applications. We modified the HWLOC tool to cope with these new needs in collaboration with Intel. This work led to the design a new structural model for platforms with heteregeneous memories [10].

7.7. Scalable Management of Platform Topologies

HWLOC (see Section 6.6) is used for gathering the topology of thousands of nodes in large clusters. Those nodes are now growing to hundreds of cores, making the overall amount of topology information non-negligible. We designed new ways to compress topologies, either lossless or lossy, for easier transfer between compute nodes and front nodes and more compact storage and manipulation [20]. We also studied the overhead of topology discovery on the overall execution time and showed that the Linux kernel is bottleneck on large nodes. It raised the need to use exported and/or abstracted topologies to factorize this overhead [11].

7.8. MPI One-side operations

MPI one-sided operations, aka Remote Memory Access (RMA), are direct read/write memory access to a remote node. Only one node (the origin) explicitely calls MPI operations, while communication progression is implicit for the other node (the target). These operations assume that the communication library is able to make communication progress in background.

Since MadMPI, the MPI implementation of NewMadeleine (see Section 6.2), extensively uses event-driven mechanism to reach asynchronous progression, we have [24] taken advantage of this property to implement MPI RMA operations in the library. This implementation keeps the overlap properties by asynchronously handle the messages exchanged by the applications. The addition also supports MPI_THREAD_MULTIPLE, for both shared and distributed memory contexts.

7.9. Topology and affinity aware hierarchical and distributed load-balancing

The evolution of massively parallel supercomputers make palpable two issues in particular: the load imbalance and the poor management of data locality in applications. Thus, with the increase of the number of cores and the drastic decrease of amount of memory per core, the large performance needs imply to particularly take care of the load-balancing and as much as possible of the locality of data. One mean to take into account this locality issue relies on the placement of the processing entities and load balancing techniques are relevant in order to improve application performance. With large-scale platforms in mind, we developed a hierarchical and distributed algorithm which aim is to perform a topology-aware load balancing tailored for Charm++ applications. This algorithm is based on both LibTopoMap for the network awareness aspects and on Treematch to determine a relevant placement of the processing entities. We show that the proposed algorithm improves the overall execution time in both the cases of real applications and a synthetic benchmark as well. For this last experiment, we show a scalability up to one millions processing entities [12].

7.10. Topology-Aware Data Aggregation for Intensive I/O on Large-Scale Supercomputers

Reading and writing data efficiently from storage systems is critical for high performance data-centric applications. These I/O systems are being increasingly characterized by complex topologies and deeper memory hierarchies. Effective parallel I/O solutions are needed to scale applications on current and future supercomputers. Data aggregation is an efficient approach consisting of electing some processes in charge of aggregating data from a set of neighbors and writing the aggregated data into storage. Thus, the bandwidth use can be optimized while the contention is reduced. In [13], we have taken into account the network topology for mapping aggregators and we propose an optimized buffering system in order to reduce the aggregation cost. We have validated our approach using micro-benchmarks and the I/O kernel of a large-scale cosmology simulation. We have showed improvements up to 15× faster for I/O operations compared to a standard implementation of MPI I/O.

7.11. Communication monitoring in OpenMPI

Monitoring data exchanges is critical when it comes to optimize process placement in a large scale environment. We participated in adding in Open-MPI, which is one of the major MPI implementation, a fine grain, point-to-point monitoring component that keeps track of message exchanges. Unlike implementations using PMPI operations, the layer in which this monitoring acts allow us to record at a lower level the effective data communications, for example, after the covering tree has been calculated. This component has been enriched with a complete coverage of collectives, point-to-point and one-sided communications. This component also reports informations about message sizes distribution. Monitored informations can be accessed by using MPI_Tools interface, or by dumping data in files.

7.12. Process Placement with TreeMatch

We released TreeMatch ver 0.4 in August. The new feature are: a new API, the handling oversubscribing (being able to map more processes that computing resources), fast exhaustive search (for small cases), K-partitioning in case of large arity of the tree, and a set of extensive tests.

7.13. Topology Aware Resource Management

SLURM is a Resource and Job Management System, a middleware in charge of delivering computing power to applications in HPC systems. Our goal is to take in account in SLURM placement process hardware topology but application communication pattern too. We have a new [9], [19] selection option for the cons_res plugin in SLURM. In this case the usually best_fit algorithm used to choose nodes is replaced by TreeMatch, an algorithm to find the best placement among the free nodes list in light of a given application communication matrix. We plan to release this work in the next release SLURM 17.02.

Fragmentation in cluster is one of the criteria important for administrator. Indeed, the way jobs are allocated impacts the global resource usage. Usually it is observed throught utilization of a cluster for a fixed load rate, but no metrics dedicated to fragmentation exist in litterature. Hence we construct several metrics to measure it. Our goal is to study the impact of our selection algorithm on fragmentation in comparison with other.

7.14. Impact of progress threads placement for MPI Non-Blocking Collectives

MPI Non-Blocking Collectives (NBC) allow communication overlap with computation. A good overlapping ratio is obtained when computation and communication are running in parallel. To achieve this, some implementations use progress threads to manage communication tasks. These threads should be bound on different cores to maximize the overlap. Thus, we elaborate several threads placement algorithms. These algorithms have been implemented within the MPC framework, using the HWLOC software to get a global view of the machine topology. We propose [18] a thread placement algorithm taking into account the NUMA topology of the machine in order to improve the overlapping ratio of non-blocking collective communications.

7.15. Hierarchical Communication Management in MPI

MPI, in its current state provides only a very limited set of functionnalities so as to allow the programmer to effectively leverage the physical characteristics of the underlying hardaware, such as the potentially complex memory hierarchy. The MPI philosophy being to be a hardware-agnostic interface, the challenge is therefore to propose an interface extension that offers the programmer significant control over the hardawre without dwelving too much into hardware details. We seek the right level of abstraction for this interface and the goal is push this proposal to the MPI Forum. This new interface is based on the concept of communicators, expands an already existing function available in the standard and also introduces a couple of helper functions. We have prototyped and drafted our proposal for the 2017 meetings of the forum.

7.16. Fully-abstracted approach for efficient thread binding in task-based model of programming

Task-based models and runtimes are quite popular in the HPC community. They help to implement applications with a high level of abstraction while still applying different types of optimizations. An important optimization target is hardware affinity, which concerns to match application behavior (thread, communication, data) to the architecture topology (cores, caches, memory). In fact, realizing a well adapted placement of threads is a key to achieve performance and scalability, especially on NUMA-SMP machines. However, this type of optimization is difficult: architectures become increasingly complex and application behavior changes with implementations and input parameters, *e.g* problem size and number of thread. Thus, by themselves task based runtimes often deal badly with this optimization and leave a lot of fine-tuning to the user. In this work [21], [25], [14], we propose a fully automatic, abstracted and portable affinity module. It produces and implements an optimized affinity strategy that combines knowledge about application characteristics and the

architecture's topology. Implemented in the backend of our task-based runtime ORWL, our approach was used to enhance the performance and the scalability of several unmodified ORWL-coded applications: matrix multiplication, a 2D stencil (Livermore Kernel 23), and a video tracking real world application. On two SGI SMP machines with quite different hardware characteristics, our tests show spectacular performance improvements for this unmodified application code due to a dramatic decrease of cache misses. A comparison to reference implementations using OpenMP confirms this performance gain of almost one order of magnitude.

7.17. Multi-criteria graph partitioning for multi-physics simulations load balancing

A new set of algorithms has been designed to compute multi-criteria static mappings for the load balancing of multi-phisics simulations. The multi-criteria graph partitioning is known to be NP-hard, and there exist very few multi-criteria graph partitioners. Moreover, they focus on the edge-cut minimization instead of enforcing load balance. In practice, this strategy often leads to very unbalanced partitions, which are not useful for multi-physics simulations.

We have designed algorithms that focus on balancing several criteria at the same time to ensure that our results always match all balance criteria. We have implemented a prototype in Python to test these different heuristics. One of them, called PIERE, obtained good results [15], in term of balance as well as communication costs. PIERE uses the classic multilevel framework, but implements a new initial partitioning algorithm, which allows to find a balanced partition of the graph. The partition is then refined by local optimization heuristics that ensure the balance is kept for all criteria. This allow us to return a partition respecting the balance constraints. In [15], we compare against well-known partitioners that are SCOTCH and METIS, and highlight that, for a small mesh, the results exhibit a high discrepancy: each tool lacks of robustness.

PIERE outperformed the existing software METIS in our test cases, but there is room for improvement. We also verified the superiority of the hypergraph model over the graph model used by most partitioners. Meanwhile, we studied the source code of well known partitioners, namely METIS and SCOTCH, and we have identified a lot of algorithmic choices and internal parameters that are not described in their documentations. Carefully analyzing them helps us to clearly understand the differences of the different algorithms.

7.18. Scotch

In order to prepare for the inclusion of multi-criteria graph partitioning algorithms in SCOTCH, in the context of the PhD thesis of Rémi Barat, a new branch has been created in the SCOTCH repository. This new branch, labeled as 6.1, is the basis for the next main release of SCOTCH. The sequential graph structure has been adapted to handle graphs with multiple loads per vertex, and all the related algorithms have been adapted to take into account multiple vertex loads. This resulted in minimal updates in the interface of Scotch, with ful ascending compatibility. All of these modifications have been performed so as not to slow down significantly the algorithms in the most common case of graphs with single vertex loads.

7.19. PaMPA

Parallel remeshing has been improved. PaMPA coupled with Mmg (v5) remeshed a tetrahedral mesh from 43Melements to more than 1Belements on 280 Broadwell processors in 20 minutes. The resulted mesh, used by CERFACS, permitted one of the most finest simulation computed with LES (Large Eddy Simulation) on combustion.

The scalability of PT-SCOTCH scalability has been tested on the Curie cluster and compared to that of PARMETIS. These tests used DARI resources.

7.20. Originality of software works

Most judges have very little, if not none, knowledge on software developement. This results in misconceptions and mistakes regarding the application of copyright/author right (*droit d'auteur*) in court cases related to software. More generally, the concept of originality is misunderstood. While this criterion is meant in theory to separate works of the mind that are personal to an author (e.g., literary works), from creations of form that cannot, by nature, reflect the personality of their creator (e.g. mathematical tables), it is often used to qualify the degree of similarity between two different works, in the context of plagiarism. Also, the distinction between the realm of programs, that is, works of the mind, and that of algorithms, is not mastered. Algorithms belong to the *fonds commun*, a French term that has no equivalent in English and might be translated as "common pool". In order to help judges and lawmakers in understanding these notions, and articulate them, we have proposed a methodology for ruling software disputes. This methodology is solely based on the study of similarities in software code, since author right exclusively pertains to the level of the form [22].

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Contract with CEA

CEA is granting the PhD thesis of Hugo Taboada on specialized thread management in the context of multi programming models, and the PhD thesis of Rémi Barat on multi-criteria graph partitioning.

8.2. Bilateral Grant with Bull/Atos

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

8.3. Bilateral Grant with Onera

Onera is granting the PhD thesis of Raphaël Blanchard on the parallelization and data distribution of discontinuous Galerkin methods for complex flow simulations.

8.4. Bilateral Grant with EDF

EDF is granting the CIFRE PhD thesis of Benjamin Lorendeau on new programming models and optimization of Code Saturn.

8.5. Bilateral Grant with Intel

Intel is granting $30k and providing 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.

9. Partnerships and Cooperations

9.1. National Initiatives

9.1.1. ANR

ANR MOEBUS Scheduling in HPC (http://moebus.gforge.inria.fr/doku.php).

ANR INFRA 2013, 10/2013 - 9/2017 (48 months)

Coordinator: Denis Trystram (Inria Rhône-Alpes)

Other partners: Inria Bordeaux Sud-Ouest, Bull/ATOS

Abstract: This project focuses on the efficient execution of parallel applications submitted by various users and sharing resources in large-scale high-performance computing environments

ANR SATAS SAT as a Service.

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.

9.1.2. IPL - Inria Project Lab

MULTICORE - Large scale multicore virtualization for performance scaling and portability

Participants: Emmanuel Jeannot and Farouk Mansouri.

Multicore processors are becoming the norm in most computing systems. However supporting them in an efficient way is still a scientific challenge. This large-scale initiative introduces a novel approach based on virtualization and dynamicity, in order to mask hardware heterogeneity, and to let performance scale with the number and nature of cores. It aims to build collaborative virtualization mechanisms that achieve essential tasks related to parallel execution and data management. We want to unify the analysis and transformation processes of programs and accompanying data into one unique virtual machine. We hope delivering a solution for compute-intensive applications running on general-purpose standard computers.

9.2. European Initiatives

9.2.1. Collaborations in European Programs, Except FP7 & H2020

COLOC: the Concurrency and Locality Challenge (http://www.coloc-itea.org).

Program: ITEA2

Project acronym: COLOC

Project title: The Concurrency and Locality Challenge

Duration: November 2014 - November 2017

Coordinator: BULL/ATOS

Other partners: BULL/ATOS (France); Dassault Aviation (France) ; Enfeild AB (Sweden); Scilab entreprise (France); Teratec (France); Inria (France); Swedish Defebnse Research Agency - FOI (France); UVSQ (France).

Abstract: The COLOC project aims at providing new models, mechanisms and tools for improving applications performance and supercomputer resources usage taking into account data locality and concurrency.

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.

9.2.2. Collaborations with Major European Organizations

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

Subject 1: Application modeling for for hierarchical memory system

Partner 2: Argonne National Lab

Subject 2: Topology-aware data aggregation for I/O intensive application

Partner 3: BSC, Barcelona (Spain)

Subject 3: High-performance communication on new architectures; load-balancing and meshing: improve the distribution of data accross the processors for a flow and particle simulation in the human nasal cavity.

Partner 4: University of Liege (Belgium), Université Catholique de Louvain (Belgium), Weierstrass Institute for Applied Analysis and Stochastics (WIAS) (Germany)

Subject 4: Coupling sequential remeshers with PaMPA began in 2016. The work [23] is in progress and it concerns Tetgen developped by Hang Si, and Gmsh by Christophe Geuzaine and Jean-François Remacle.

9.3. International Initiatives

9.3.1. Inria International Labs

Joint-Lab on Extreme Scale Computing (JLESC):

Coordinators: Franck Cappello and Marc Snir.

Other partners: Argonne National Lab, University of Urbanna Champaign, Tokyo Riken, Jülich Supercomputing Center, Barcelona Supercomputing Center.

Abstract: The Joint Laboratory is based at Illinois and includes researchers from Inria, and the National Center for Supercomputing Applications, ANL, Riken, Jülich, and BSC. It focuses on software challenges found in extreme scale high-performance computers.

9.3.2. Inria International Partners

9.3.2.1. Declared Inria International Partners

Partner 1: AMD Research

Subject 1: Managing locality in the Heterogeneous System Architecture.

AMD provided hardware and details about its future architectures and programming models (HSA) to improve locality support for its products in the HWLOC software.

9.3.2.2. Informal International Partners

Partner 1: ICL at University of Tennessee

Subject 1: 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

Partner 2: Cisco Systems

Subject 2: network topologies and platform models

Partner 3: University of Tokyo and RIKEN

Subject 3: Adaptation of MPI and runtime systems to lightweight kernels used on clusters of manycores. This action has been submitted as a JLESC project proposal, currently beeing evaluated.

Partner 4: Lawrence Livermore National Laboratory

Subject 4: Testing of the mapping features of SCOTCH on very large process graphs (more than two billion vertices) and very large target architectures (more than 200,000 parts).

Partner 5: Sandia National Lab

Subject 5: Topology-aware management and allocation of computing resources in runtime systems.

9.4. International Research Visitors

9.4.1. Visits of International Scientists

- Balazs Gerofi from RIKEN visited us to present his work on micro-kernels for HPC. His visit led to a project proposal for JLESC.
- Jose-Luiz Garcìa Zapata, stayed for three months in the team to work on spectral partitioning and mapping. He implemented a spectral bipartitioning method in SCOTCH.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events Organisation

10.1.1.1. General Chair, Scientific Chair

Guillaume AUPY was the Technical Program vice-chair of SC'17.

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.

10.1.2. Scientific Events Selection

10.1.2.1. Chair of Conference Program Committees

Guillaume AUPY was the co-chair of the Parallel and Distributed Algorithms track of ICA3PP'17.

Emmanuel JEANNOT was the Program chair of the Heterogeneity in Computing Workshop (HCW'17).

Emmanuel JEANNOT was the Program chair of the track parallelism of COMPAS 2016.

10.1.2.2. Member of the Conference Program Committees

Alexandre DENIS was a member of the program committee of Compas'16 and CCGrid 2017.

Brice GOGLIN was a member of the program committee of CCGrid 2016, Cluster 2016, EuroMPI 2017, HotInterconnect 24 and of the Exacomm workshop.

Emmanuel JEANNOT was a member of the program committee of IPDPS 2017, CCGRID 2017,

Guillaume MERCIER was a member of the program committee of EuroMPI 2016 and EuroMPI 2017.

10.1.2.3. Reviewer

Cyril BORDAGE was reviewer for Cluster 2016.

Alexandre DENIS was a reviewer for Cluster 2016.

Brice GOGLIN was a reviewer for IEEE Micro.

Farouk MANSOURI was a reviewer for Cluster 2016.

Guillaume MERCIER was a reviewer for IPDPS 2017.

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 & Distibuted Systems (IJPEDS).

Guillaume MERCIER is editor of the EuroMPI 2016 Special issue of the Journal of High Performance Computing Applications (IJHPCA).

10.1.3.2. Reviewer - Reviewing Activities

Guillaume AUPY was a reviewer for EURASIP Journal of Embedded Systems, Cluster Computing and Transactions on Parallel and Distributed Systems (TPDS).

Alexandre DENIS was a reviewer for the Journal of Parallel and Distributed Computing (JPDC).

Emmanuel JEANNOT was reviewer of IEEE TPDS.

Guillaume MERCIER was a reviewer for the EuroMPI 2016 Special Issue of the Parallel Computing journal.

François PELLEGRINI was a reviewer for SIAM Journal on Scientific Computing (SISC).

10.1.4. Invited Talks

Brice GOGLIN gave a talk about managing hardware locality in HPC during an AMD Tech Talk at AMD Research (Austin, TX).

Emmanuel JEANNOT gave a talk about topology-aware data management at the Workshop on Clusters, Clouds, and Data for Scientific Computing (CCDSC 2016).

Emmanuel JEANNOT gave a talk about metrics and models for process placement at the Third Workshop on Programming Abstractions for Data Locality (PADAL'16).

François PELLEGRINI delivered a keynote speech on freedom in the digital age, during the annual congress of *Société informatique de France*, Strasbourg.

François PELLEGRINI gave a talk on software law at Université de Nice Sophia-Antipolis.

François PELLEGRINI participated in a round-table on *Big data, compliance and personal data* during the JInov meeting, Paris.

François PELLEGRINI gave a talk on *Free software, a tool for sustainable development in countries of the Souths* law at the *Colloque international sur le logiciel libre dans les pays du Sud*, organized by Université Moulay Ismaïl & École nationale supérieure d'arts et métiers de Meknès.

François PELLEGRINI delivered a talk on freedom in the digital age, during the Defense Security Cyber summer school organized by Université de Bordeaux.

François PELLEGRINI delivered a talk on freedom and the ethics of informatics during the summer school for young researchers on the ethics of informatics, organized by CERNA and Allistene in Arcachon.

François PELLEGRINI participated in a round-table on the legal crtieria for software originality in the colloquium on protection and infringement of software : the notion of digital common pool, organized by AFDIT at Conseil national des barreaux, Paris.

François PELLEGRINI delivered a talk on the issues of rights on immaterial goods for digital development, during the international seminal of training for trainers on internet and information systems governance, organised by ITICC with the support of Organisation Internationale de la Francophonie and ARCEP-BF, in Ouagadougou.

François PELLEGRINI delivered the opening conference on the legal and economic bases the digital economy, for a training seminar for Members of the Parliament of Benin on the issues of laws on digital matters, organized by Organisation Internationale de la Francophonie at Grand-Popo.

François PELLEGRINI gave a talk on the operational solutions to digital security issues, during the 4th NGO forum organized by the French embassy in Moscow.

François PELLEGRINI delivered a keynote speech on the governance of open and free innovation, at the invitation of the French ministry of Foreign affairs, during the workshop on open innovation which took place within the French-German inter-governmental conference on digital issues, in Berlin.

Adèle VILLIERMET has been invited to give a talk at the summer school of GDR RO.

10.1.5. Scientific Expertise

Emmanuel JEANNOT was member of the hiring committee for an assistant professor position in informatics at Université de Bordeaux.

Brice GOGLIN was also a member of the hiring committee for Inria Bordeaux - Sud-Ouest research scientists.

François PELLEGRINI was a member of the hiring committee for a full professor position in informatics at Université de Nice Sophia-Antipolis (PR27-327). He also reviewed a PR1 promotion file at Université de Bordeaux.

10.1.6. Standardization Activities

TADAAM attends the MPI Forum meetings on behalf of Inria (where the MPI standard for communication in parallel applications is developed and maintained).

A proposal in currently under early discussion for submission to the forum 7.15.

10.1.7. Tutorials

Brice GOGLIN gave a tutorial about managing hardware affinities on hierarchical platforms with HWLOC during a PRACE Advanced Training Center session.

François PELLEGRINI gave a “hands-on” tutorial on SCOTCH during a meeting of the European projet COLOC.

10.1.8. Research Administration

Emmanuel JEANNOT is member of the scientific committee of the Labex IRMIA (Universite´ de Strasbourg).

Emmanuel JEANNOT is the head of the young researcher commission of Inria Bordeaux Sud-Ouest in charge of supervising the hiring of the PhDs and post-doc of the center.

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 and C programming to advanced topics such as computer architecture, operating systems, parallel programming and high-performance runtime systems, as well as software law.

10.2.2. Supervision

PhD in progress: Remi Barat, multi-criteria graph partitioning, started in 2014. Advisor: François Pellegrini.

PhD in progress: Raphaël Blanchard, parallelization and data distribution of discontinuous Galerkin methods for complex flow simulations, started in 2013. Advisor: François Pellegrini.

PhD in progress: Nicolas Denoyelle, advanced memory hierarchies and new topologies, started in 2015. Advisor: Brice Goglin and Emmanuel Jeannot.

PhD in progress: Benjamin Lorendeau, new programming models and optimization of Code Saturn, started in 2015. Advisor: Yvan Fournier and Emmanuel Jeannot.

PhD in progress: Hugo Taboada, communication progression in runtime systems, started in 2015. Advisor: Alexandre Denis and Emmanuel Jeannot.

PhD in progress: Adèle Villiermet, topology-aware resource management, started in 2014. Advisor: Emmanuel Jeannot and Guillaume Mercier.

PhD stopped: Romain Prou, communication management based on remote memory access, student resigned in october 2016. Advisor: Alexandre Denis and Emmanuel Jeannot.

10.2.3. Juries

Brice GOGLIN was member of the PhD defense committee of:

- Mohamed Lamine Karaoui (Université Pierre et Marie Curie, Reviewer).

Emmanuel JEANNOT was member of the PhD defense committee of:

- Loïc Thiébault (Université de Versailles Saint-Quentin, Reviewer).

François PELLEGRINI was member of the PhD defense committee of:

- Karl-Eduard Berger (Université de Versailles Saint-Quentin);
- Alessandro Fanfarillo (Università degli Studi di Roma Tor Vergata, Reviewer);
- Thomas Hume, Université de Bordeaux;
- Sébastien Morais (Université Évry Val d'Essonne, Reviewer).

10.3. Popularization

Brice GOGLIN is in charge of the diffusion of the scientific culture for the Inria Research Center of Bordeaux. He organized several popularization activities in the center. He also gave several talks about computer architecture, high performance computing, and research careers to general public audience, school students, teachers, or even to non-expert Inria colleagues.

Brice GOGLIN was involved in the design of the section about fondamentals of computer science in the 2017 massive open online course that will help teachers of the new ICN section in schools (*Informatique et Création Numérique*). It was filmed for 10 video sequences (about an hour in total).

François PELLEGRINI was filmed during a 3-hour conference on author's rights, in the context of the MAPI'Days, to serve as an on-line training for personnel and students of Université de Bordeaux (https://fad.u-bordeaux.fr/course/view.php?id=740).

François PELLEGRINI is the author of an opinion piece on digital sovereignty in newspaper Le Monde (http://www.lemonde.fr/idees/article/2016/06/24/la-souverainete-numerique-passe-par-le-logiciel-libre_4957781_3232.html).

François PELLEGRINI is the co-author of a booklet on free/libre software licenses edited by Pôle Systematic Paris Région & Pôle Aquinetic, which is now in its second edition (http://systematic-paris-region.org/sites/default/files/content/page/attachments/LivretBleu_Juridique_GT-LogicielLibre_Systematic_Mai2016_web.pdf).

In the context of the decree authorizing the TES (*Titres Électroniques Sécurisés*) file, François PELLEGRINI published a set of three blog posts (starting with http://www.pellegrini.cc/2016/11/la-biometrie-des-honnetes-gens/), which have been cited and linked by several French newspapers (Libération, Mediapart, NextInpact). He also participated in a debate on the same subject, organised by the Ligue des droits de l'Homme de Gironde (http://ldh-gironde.org/jeudi-15-decembre-2016-a-18h30-rencontre-debat-autour-du-mega-fichier-tes/).

François PELLEGRINI delivered a talk on *Freedom and the ethics of informatics* during a seminar on *Technologies, ethics and cognition* organized by the bouddhist group Dhagpo Bordeaux, in partnership with Cap Sciences and Université de Bordeaux (http://www.dhagpo-bordeaux.org/seminaire-technologies-ethique-cognition/).

François PELLEGRINI was filmed, during an interview on *Innovation and free/libre licenses*, for the *ULab Innov+* MOOC.

11. Bibliography

Publications of the year

Articles in International Peer-Reviewed Journals

[1] L.-C. CANON, E. JEANNOT. *Correlation-Aware Heuristics for Evaluating the Distribution of the Longest Path Length of a DAG with Random Weights*, in "IEEE Transactions on Parallel and Distributed Systems", 2016, https://hal.inria.fr/hal-01412922

Articles in Non Peer-Reviewed Journals

[2] J. CARRETERO, R. ČIEGIS, E. JEANNOT, L. LEFÈVRE, G. RÜNGER, D. TALIA, Ž. JULIUS. *HeteroPar 2014, APCIE 2014, and TASUS 2014 Special Issue*, in "Concurrency and Computation: Practice and Experience", 2016, 2 p. , https://hal.inria.fr/hal-01253278

Invited Conferences

[3] F. PELLEGRINI. *L'enjeu du big data pour la gouvernance*, in "Journée d'Etude : Transition numérique et action publique : focus sur la loi pour une République numérique", Paris, France, Centre d'études et de Recherches de sciences administratives et politiques de l'Université Paris II and Chaire Mutations de l'Action Publique et du Droit Public, November 2016, https://hal.inria.fr/hal-01418990

[4] F. PELLEGRINI. *La production d'un intérêt général dans la gouvernance polycentrique de l'Internet*, in "3è Colloque international du Centre de Droit Public Comparé de l'Université Panthéon-Assas Paris-II", Paris, France, Centre de Droit Public Comparé de l'Université Panthéon-Assas Paris-II, May 2016, https://hal.inria.fr/hal-01418989

[5] F. PELLEGRINI. *Liberté à l'ère numérique*, in "Les métamorphoses des droits fondamentaux à l'ère du numérique", Bordeaux, France, Forum Montesquieu, université de Bordeaux and CERCCLE and CRDEI and Institut Léon Duguit, November 2016, https://hal.inria.fr/hal-01418991

International Conferences with Proceedings

[6] P.-A. ARRAS, D. FUIN, E. JEANNOT, S. THIBAULT. *DKPN: A Composite Dataflow/Kahn Process Networks Execution Model*, in "24th Euromicro International Conference on Parallel, Distributed and Network-based processing", Heraklion Crete, Greece, February 2016, https://hal.inria.fr/hal-01234333

[7] I. CORES, P. GONZÁLEZ, E. J. JEANNOT, M. J. MARTÍN, G. RODRÍGUEZ. *An application-level solution for the dynamic reconfiguration of MPI applications*, in "12th International Meeting on High Performance Computing for Computational Science", Porto, Portugal, June 2016, https://hal.inria.fr/hal-01424263

[8] A. DENIS, F. TRAHAY. *MPI Overlap: Benchmark and Analysis*, in "International Conference on Parallel Processing", Philadelphia, United States, 45th International Conference on Parallel Processing, August 2016, https://hal.inria.fr/hal-01324179

[9] Y. GEORGIOU, E. JEANNOT, G. MERCIER, A. VILLIERMET. *Topology-aware resource management for HPC applications*, in "ICDCN 2017", Hyderabad, India, January 2017 [*DOI :* 10.1145/3007748.3007768], https://hal.inria.fr/hal-01414196

[10] B. GOGLIN. *Exposing the Locality of Heterogeneous Memory Architectures to HPC Applications*, in "1st ACM International Symposium on Memory Systems (MEMSYS16)", Washington, DC, United States, ACM, October 2016 [*DOI :* 10.1145/2989081.2989115], https://hal.inria.fr/hal-01330194

[11] B. GOGLIN. *On the Overhead of Topology Discovery for Locality-aware Scheduling in HPC*, in "Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP2017)", St Petersburg, Russia, Proceedings of the 25th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP2017), IEEE Computer Society, March 2017, 9 p. , https://hal.inria.fr/hal-01402755

[12] E. JEANNOT, G. MERCIER, F. TESSIER. *Topology and affinity aware hierarchical and distributed load-balancing in Charm++*, in "1st Workshop on Optimization of Communication in HPC runtime systems (IEEE COM-HPC16)", Salt-Lake City, United States, November 2016, https://hal.inria.fr/hal-01394748

[13] F. TESSIER, P. MALAKAR, V. VISHWANATH, E. JEANNOT, F. ISAILA. *Topology-Aware Data Aggregation for Intensive I/O on Large-Scale Supercomputers*, in "1st Workshop on Optimization of Communication in HPC runtime systems (IEEE COM-HPC16)", Salt-Lake City, United States, IEEE, November 2016, 9 p. , https://hal.inria.fr/hal-01394741

National Conferences with Proceedings

[14] F. MANSOURI, J. GUSTEDT. *Le modèle de programmation ORWL pour la parallélisation d'une application de suivi vidéo HD sur architecture multi-coeurs*, in "Conférence d'informatique en Parallélisme, Architecture et Système (COMPAS)", Lorient, France, July 2016, accepted for publication in Compas'16, https://hal.inria.fr/hal-01325850

Conferences without Proceedings

[15] R. BARAT, C. CHEVALIER, F. PELLEGRINI. *Multi-constraints graph partitioning for load balancing of multi-physics simulations*, in "Conférence d'informatique en Parallélisme, Architecture et Système (COMPAS)", Lorient, France, July 2016, https://hal.inria.fr/hal-01417532

[16] N. DENOYELLE. *Moniteurs hiérarchiques de performance, pour gérer l'utilisation des ressources partagées de la topologie*, in "Compas", Lorient, France, July 2016, https://hal.inria.fr/hal-01343152

[17] N. DENOYELLE, A. ILIC, B. GOGLIN, L. SOUSA, E. JEANNOT. *Automatic Cache Aware Roofline Model Building and Validation Using Topology Detection*, in "NESUS Third Action Workshop and Sixth Management Committee Meeting", Sofia, Bulgaria, Jesus Carretero, October 2016, vol. I, https://hal.inria.fr/hal-01381982

[18] H. TABOADA. *Impact du placement des threads de progression pour les collectives MPI non-bloquantes*, in "Compas 2016: conférence d'informatique en Parallélisme, Architecture et Système", Lorient, France, July 2016, https://hal.archives-ouvertes.fr/hal-01355140

Research Reports

[19] Y. GEORGIOU, E. JEANNOT, G. MERCIER, A. VILLIERMET. *Topology-aware resource management for HPC applications*, Inria Bordeaux Sud-Ouest ; Bordeaux INP ; LaBRI - Laboratoire Bordelais de Recherche en Informatique, February 2016, n^{o} RR-8859, 17 p. , https://hal.inria.fr/hal-01275270

[20] B. GOGLIN. *Towards the Structural Modeling of the Topology of next-generation heterogeneous cluster Nodes with hwloc*, Inria, November 2016, https://hal.inria.fr/hal-01400264

[21] J. GUSTEDT, E. JEANNOT, F. MANSOURI. *Fully-abstracted affinity optimization for task-based models*, Inria Nancy, December 2016, n^{o} RR-8993, https://hal.inria.fr/hal-01409101

[22] F. PELLEGRINI. *The originality of software works*, Inria Bordeaux Sud-Ouest ; Université de bordeaux, August 2016, n^{o} RR-8945, 13 p. , https://hal.inria.fr/hal-01352700

Other Publications

[23] A. BARDOUX. *Remaillage parallèle pour le calcul haute performance*, Université de strasbourg, August 2016, https://hal.inria.fr/hal-01417406

[24] C. FOYER. *Updating MadMPI to MPI-3: Remote Memory Access*, Inria Bordeaux, équipe TADAAM, September 2016, https://hal.inria.fr/hal-01395299

[25] J. GUSTEDT, E. JEANNOT, F. MANSOURI. *Optimizing Locality by Topology-aware Placement for a Task Based Programming Model*, September 2016, pp. 164 - 165, IEEE Cluster 2016 Conference, Poster [*DOI :* 10.1109/CLUSTER.2016.87], https://hal.archives-ouvertes.fr/hal-01416284