Activity Report 2015

Team TADAAM

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

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
2.2.3. - 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
7.9. - 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]
Alexandre Denis [Inria, Researcher]
Brice Goglin [Inria, Researcher, HdR]
Thomas Ropars [Inria, Starting Research position, until Aug 2015]

Faculty Members

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

Engineers

Cedric Lachat [Inria]
François Tessier [Inria]

PhD Students

Remi Barat [CEA]
Raphaël Blanchard [ONERA]
Nicolas Denoyelle [Bull, granted by CIFRE]
Benjamin Lorendeau [EDF, granted by CIFRE]
Romain Prou [Inria, from Apr 2015]
Hugo Taboada [CEA, from Oct 2015]

Adele Villiermet [Inria]

Post-Doctoral Fellows

Cyril Bordage [Inria, from Dec 2015]
Farouk Mansouri [Inria, from Nov 2015]

Visiting Scientist

Ivan Cores [Univ. étrangère, from Feb 2015 until May 2015]

Administrative Assistants

Sylvie Embolla [Inria, from Jun 2015]
Flavie Tregan [Inria, until Jun 2015]

Others

Guillaume Beauchamp [Inria, intern, from Jun 2015 until Aug 2015]
David Phan [Inria, intern, from May 2015 until Aug 2015]
Paul-Antoine Arras [Inria, Post-PHD contract, until Feb 2015]

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.

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.

[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

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 9th 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. New Software and Platforms

5.1. Hardware Locality

KEYWORDS: Topology - 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 the application so as to help them adapt their behavior to the hardware characteristics. HWLOC also offers monitoring abilities to identify application bottlenecks [12]. Moreover it focuses on modeling the network topology by embedding the NETLOC subproject in its future releases.

HWLOC targets many types of high-performance computing applications [1], [2], from thread scheduling to placement of MPI processes. Most existing MPI implementations, many resource managers, task schedulers and parallel libraries already use HWLOC.

HWLOC is developed in collaboration within the OPEN MPI consortium. The core development is carried out by Brice GOGLIN and other members of TADAAM team-project, with external contribution from many academic and industrial partners.

HWLOC is composed of 100,000 lines of C.

- Participants: Brice Goglin, Nicolas Denoyelle, Cyril Bordage
- Contact: Brice Goglin
- URL: http://www.open-mpi.org/projects/hwloc/

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

5.2. NewMadeleine

KEYWORDS: High-performance computing - 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
- Contact: Alexandre Denis
- URL: http://pm2.gforge.inria.fr/newmadeleine/

5.3. PaMPA

KEYWORDS: parallel numerical solvers, unstructured heterogeneous meshes, dynamic load balancing, subdomain decomposition, parallel remeshing
FUNCTIONAL DESCRIPTIONPAMPA ("Parallel Mesh Partitioning and Adaptation") is a library dedicated to the management of distributed meshes. Its purpose is to relieve solver writers from the tedious and error-prone task of writing again and again service routines for mesh handling, data communication and exchange, remeshing, and data redistribution. It is based on a distributed data structure that represents meshes as a set of entities (elements, faces, edges, nodes, etc.), linked by relations (that is, computation dependencies).

PAMPA interfaces with SCOTCH for mesh redistribution, and with MMG3D for parallel remeshing of tetrahedral elements. Other sequential remeshers can be plugged-in, in order to handle other types of elements.

Version 1.0 of PAMPA allows users to declare distributed meshes, to declare values attached to the entities of the meshes (e.g. temperature attached to elements, pressures to the faces, etc.), to exchange values between overlapping entities located at the boundaries of subdomains assigned to different processors, to iterate over the relations of entities (e.g. iterate over the faces of elements), to remesh in parallel the areas of a mesh that need to be remeshed, and to redistribute evenly the remeshed mesh across the processors of the parallel architecture. PAMPA is already used as the data structure manager for the AeroSol solver developed by teams CARDAMOM and CAGIRE.

Version 2.0 of PAMPA features parallel remeshing using any sequential remesher. At the time being, PAMPA is interfaced with the MMG3D tetrahedral remesher designed within team CARDAMOM. Its coupling with Tetgen is in progress. Meshes above one billion elements are generated with a quality similar to that offered by the sequential remesher alone. More than 1 billion of elements are obtained on a cluster with 600 processors in less than 20 minutes. Tests show a quite good weak scalability up to 600 processors, limited by cluster size. Extensive scalability testing will be performed during year 2016. An collaboration with Dassault Aviation demonstrated the use of PAMPA with their meshes during solver computations.

- Participants: Cédric Lachat, François Pellegrini, Cécile Dobrzynski.
- Contact: Cédric Lachat, François Pellegrini
- URL: project.inria.fr/pampa

5.4. Scotch

KEYWORDS: parallel graph partitioning, parallel static mapping, parallel sparse matrix block ordering, graph repartitioning, fixed vertices, mesh partitioning.
FUNCTIONAL DESCRIPTION

SCOTCH is a software package for parallel and sequential sparse matrix ordering, parallel and sequential graph partitioning, as well as sequential static mapping and remapping, without and with fixed vertices, and mesh and hypergraph partitioning.

The initial purpose of SCOTCH was to compute high-quality static mappings of valuated graphs representing parallel computations, onto target architectures of arbitrary topologies. Taking into account the topology and heterogeneity of the target architecture, in terms of processor speed and link bandwidth, allows SCOTCH to provide partitions that maximize communication locality.

This feature, which was meant for the NUMA machines of the 1980's, has not been widely used in the past because high performance computers in the 1990's were UMA architectures, thanks to hardware advances. As today's high-end architectures are NUMA again, the mapping feature is regaining popularity.

The SCOTCH package consists of two libraries: the sequential SCOTCH library, and the parallel PT-SCOTCH library (for "Parallel Threaded SCOTCH") that operates according to the distributed memory paradigm, using MPI. SCOTCH was the first full 64-bit implementation of a general purpose graph partitioner.

Version `6.0`, released on December 2012, corresponding to the 20$^{\text{th}}$ anniversary of SCOTCH, offers many new features: static mapping with fixed vertices, static remapping, and static remapping with fixed vertices. Several critical algorithms of the formerly strictly sequential SCOTCH library can now run in a multi-threaded way. All of these features, which exist only in the sequential version, will be ported to the parallel PT-SCOTCH library in the upcoming `6.1` major release.

A recent work on branch `6.0` enables SCOTCH to map onto (possibly disconnected) parts of a regular architecture, thanks to the new `sub` target architecture. This feature aims at allowing programs to distribute their data so as to maximize locality, according to the assignment of nodes performed by the batch scheduler, which do not always represent a regular, nor a connected, part of a parallel machine.

SCOTCH has been integrated into numerous third-party software, which indirectly contribute to its diffusion.

- Participants: François Pellegrini, Cédric Lachat, Astrid Casadei.
- Contact: François Pellegrini
- URL: https://gforge.inria.fr/projects/scotch/

5.5. TreeMatch

KEYWORDS: High-performance computing. Process Placement
FUNCTIONAL DESCRIPTION

TreeMatch is a library for performing process placement based on the topology of the machine and the communication pattern of the application. 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.

- Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.
- Contact: Emmanuel Jeannot
- URL: http://treematch.gforge.inria.fr

5.6. Platforms

5.6.1. Platform: The Daltons

The Daltons are a set of machines available for the team members and the STORM team members enabling fast prototyping and benchmarking of our solutions, algorithms and software. It features up-to-date hardware (e.g. latest infiniband or GPU card) with the most recent processors.

6. New Results

6.1. TreeMatch Development

This year we have modified the TreeMatch API in order to enable better integration inside application with higher-level abstractions more precise semantic. We also introduced the “over subscribing” features that allow to map more than one process on a given processing unit. We also added new metrics to measure the performance of the proposed placement. We now have three metrics: The sum of the communication cost, the maximum of the communication cost and the hope-byte.

6.2. Affinity Abstraction

This year we worked on the affinity abstraction. Often, the affinity between two processes or threads is measured by the a matrix where a high entry represent a high affinity. Such example of matrices are the number of messages and the number of bytes exchanged between processing units. However, such matrix hide many characteristics of the application such as computation/communication overlap, network contention, etc. First, we have developed a new OpenMPI PML module to gather communication matrix of a running application. Then, we have conducted an extensive study of the minighost application to understand how such communication matrix actually measure the affinity between processes. On this application it appears that the size metric better matches the performances and that the performance of process placement is highly correlated to the proportion of communication in the application.

6.3. Locality for Application Using Locks on Clusters of Multicore Platforms

The aim of this post-doc work is to study the locality for applications based on read-write locks on clusters of multi-core platforms. We focused on the implementation of the video tracking application [25] using the Ordered Read Write Locks (ORWL) [20] model of programming on multi-cores architecture. For several uses, such as, human-computer interaction, security or traffic control, the tracking application aim to locate multiple moving objects over time using a camera. Its processing can be a time consuming process due to the amount of data that is contained in high definition video which leads to decrease the throughput. To overcome this problem it is possible to speed up the processing by exploiting task parallelism of ORWL model. Indeed, the model proposes abstractions of the decomposition in parallel parts (tasks), the synchronization of and the communication between threads. However, we noted some problems which decrease the parallelism scaling thus we introduced different optimizations: stream multiplexing, multiple buffering, etc. We are now working on parallelizing long-running tasks.

6.4. Topology Aware Malleability of MPI programs

Current parallel environments aggregate large numbers of computational resources with a high rate of change in their availability and load conditions. In order to obtain the best performance in this type of infrastructures, parallel applications must be able to adapt to these changing conditions.

In collaboration with Universidade da Coruña, Spain, we have worked on automatically and transparently adapt MPI applications to available resources is proposed. The solution relies on application-level migration approach, incorporating a new scheduling algorithm, based on TreeMatch and Hwloc, to obtain well balanced nodes while preserving performance as much as possible.

The experimental evaluation shows successful and efficient operation, with an overhead of less than 1 second for the proposed scheduling algorithm, and of only a few seconds for the complete reconfiguration, which will be negligible in large applications with a realistic reconfiguration frequency.

6.5. Topology Aware Load Balancing

Charm++ is a message-passing based programming environment that uses an object-oriented approach. However, where MPI considers processes in its model, Charm++ uses finer-grain migratable objects called chares. Brought together with an adaptive runtime system, Charm++ allows to perform dynamic load balancing considering the CPU load of each chare. Our work on data locality and process placement lead us to add the benefits of our TreeMatch algorithm in a load balancing solution. Thus we developed few months ago a topology-aware load-balancer in Charm++ using TreeMatch to reduce the communication costs. During the last months, we significantly improved this load-balancer and its scalability. Particularly, our load balancing algorithm is now hierarchical and distributed. To validate this approach, we have begun to carry out experiments with a cosmological application on the Blue Waters supercomputer. The results will be published soon.

6.6. Topology Aware Resource Management

SLURM [24] 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 as well as application communication pattern. We have a new selection option for the cons_res plugin in SLURM. In this case the usually BestFit algorithm used to choose nodes is replaced by TreeMatch to find the best placement among the free nodes list in light of a given application communication matrix.

We updated this plugin based on SLURM 2.6.5 for last version SLURM 15.08. To decrease the overhead due to our algorithm we also implemented an alternative to use a subtree of the global topology. We ran experiments to compare these different solutions using our plugin with or without subtree and the current algorithm topology-aware in SLURM.

6.7. Topology Aware Performance Monitoring

While system's scale is growing exponentially, memory hierarchy is getting larger, at various levels. Hence optimizing applications to reach an optimal usage of a machine may involve a large spectrum of performance metrics interacting at different level of the system's hierarchy. Memory bound applications showing irregular patterns lead to locality issues. Addressing those issues and getting a good schedule on complex systems is a NP hard problem and can therefore only be solved with heuristics. Although powerful algorithms using the most intuitive heuristics such as communications path reduction and/or cache contention reduction may show good results on some cases, there are still room for improvements in this direction so much the configuration of applications, systems, software stack vary and impact the execution time.

In order to step in this direction we developped a highly extendible tool to gather asynchronously performance data from different sources. This information is then aggregated into different topology objects (cache, node, processing unit, ...) in order to give a synthetic and topology aware information to drive optimization.

In brief the tool works this way: The user provide a description file with arithmetic expression of performance counters(defined into performance data plugins), and topology objects where to map the expression. A pair (expression,object) defines a monitor which will sample performance data and stored them into an history. Then others monitors can be defined as a combination of the previous. For instance we can attach a process and record on each core its L3 cache miss counter, and then add each of those monitor into an upper monitor located on the L3 cache. Several aggregation functions are already available but we aim to provide several statistical function to extend the possibility of data interpretation. Such functions allow to aggregate results in a meaningful way. Then we add a locality insight using lstopo tool from hwloc to draw the results on a topology. This has been published in [12]

6.8. Memory Hierarchy Aware Roofline Model

The increasing complexity of computer architectures, makes challenging to fully exploit computer systems' capabilities. The cost of tuning applications on such machines can raise quickly. Therefore, linking the information about a machine performance bounds and applications performance results respectively to those bounds can help finding the bottlenecks and motivate code optimization.

In 2009 the Roofline model [23] throws those bases by ploting on a 2 dimensional diagram, application performance (GFlop/s) and arithmetic intensity (Flop/Byte) with respect to the main memory bandwidth (GByte/s) and peak floating point performance (GFlop/s). In 2014 the model extended by Alexandar Illic, take into account the data movement inside the cache hierarchy to provide a finer analyse by showing application's performance results with respect to the differents cache bandwidths.

With the cooperation of the Cache Aware Roofline Model authors, we have worked on extending this model to the whole memory hierarchy at NUMA scale in order to drive optimisations on next generation processors embeding different memory technologies and different memory configurations like Intel's KNL does.

While we are designing a tool based on hwloc and micro-kernels to empirically extract and validate machines bottlenecks, we also want to show with real NUMA applications that the model may be extended to such hierarchy levels, still providing insightful representation.

6.9. Topology Management and Standardization

We continued to work on the diffusion of our software and ideas in existing programming interfaces and standards tailored for HPC and parallel computing. In particular, we did integrate our TreeMatch algorithm in the Open MPI implementation of the Message Passing Interface, so as to provide enhanced Virtual Topology routines in MPI allowing the user to effectively create parallel applications taking into account both their behaviour and the characteristics of the underlying hardware. Our code is available in the master repository and should be available in an Open MPI distribution at some point in the next year (2016).

We also drafted and submitted a proposal to modify the MPI interface so that information regarding the underlying physical topology could be made available at the MPI application level. We plan to push our ideas during the next year so that our proposal can eventually make its way into the MPI standard.

6.10. Modeling Next-Generation Memory Architectures

We initiated a research topic on modeling next generation memory architectures that will mix different kinds of memories. Indeed the arrival of high-bandwidth and non-volatile memories cause computing cores to have different local memory banks with different characteristics.

The hwloc software 5.1 is being extended in collaboration with CPU vendors such as Intel and AMD to better represent these new memory technologies. We are working with Bull in the context of Nicolas Denoyelle's PhD on developing abstractions for deciding where to allocate the application buffers.

6.11. Modeling Affinity of Multithreaded Applications

With the increasing complexity and scale of multi-core processors, optimizing thread placement becomes more and more challenging. Our goal is to better understand which characteristics of a multi-threaded application can have an impact on a placement decision for a given architecture. To this end, we analyze the performance of a set of applications under different placement strategies and we try to relate the obtained results to characteristics of the applications such as the data footprint of each thread, the amount of data shared between threads, or the reuse distance.

To collect information about the characteristics of multi-threaded applications, we developed a set of tools based on the PIN dynamic binary instrumentation tools. PIN allows us to get information about all instructions executed and memory location accessed by each thread of an application during its execution, and this without modifying the source code of the application.

We used our PIN-based tools to study a representative set of applications taken from two well-known benchmark suites, namely the Mantevo benchmark suites (HPC applications based on OpenMP) and the Parsec benchmark suites (general-purpose applications based on pthreads). Analyzing the results of all our tests is an ongoing work.

6.12. Thread placement and threads policy on a multicore machine with NUMA effects.

Threads placement on multicore machine with NUMA effects is inevitable to have better performances. Threads must bind on cores to avoid thread migration and to have better cache locality. MPI non-blocking collectives can generate progress threads to complete communications. These additional threads can disturb computational threads. That is why we have implemented several thread placement algorithms into the MPC framework [22]. These algorithms allow to dedicate resources only for progress threads. Thus computational threads are not disturb. We test them with our own benchmarks which test all the MPI non-blocking collectives to compare the performances with different thread placement. We observe an improvement when resources are dedicated to progress threads and take NUMA effects into account.

We want to include a mechanism into MPC to specify thread kinds (MPI, OpenMP,...). These mechanism will allow the MPC scheduler to take threads specificity into account to improve the scheduling policy. Our goal is to increase runtime performances considering each type specific needs. We have begun to implement this mechanism.

Several MPC framework bugs have been corrected, thus we contribute to its stability.

6.13. Multithreaded Communications

To program clusters of multicores, hybrid models mixing MPI+threads, and in particular MPI+OpenMP are gaining popularity. This imposes new requirements on communication libraries, such as the need for MPI_THREAD_MULTIPLE level of multi-threading support. Moreover, the high number of cores brings new opportunities to parallelize communication libraries, so as to have proper background progression of communication and communication/computation overlap.

We have proposed PIOMan [11], a generic framework to be used by MPI implementations, that brings seamless asynchronous progression of communication by opportunistically using available cores. It uses system threads and thus is composable with any runtime system used for multithreading. Through various benchmarks, we demonstrated that our pioman-based MPI implementation exhibits very good properties regarding overlap, progression, and multithreading, and outperforms state-of-art MPI implementations.

6.14. RDMA-based Communications

High-performance network hardware is nowadays dominated by RDMA-oriented technologies. The software stack is moving too towards Remote Memory Access. However, most communication libraries stil use send/receive paradigm as a common denominator. We have proposed to study a software stack for networks the is based on remote memory access from the hardware up to the enduser API, where RDMA is first class citizen and not a compatibility layer. It is expected to obtain better performance, better scalability with regard to number of communication flows or threads, and better asynchronous progression, while optimization strategies on the packet flows such as aggregation as proposed in NewMadeleine are still possible. Work has begun as a Masters thesis [18] and continues as Romain Prou Ph.D. thesis.

6.15. Network Modeling

Netloc is a tool for hwloc [1] to find the topology of a supercomputer. For that, it discovers all the networks by exploring them by using tools specifying to the network type. The exploration gives all the machines and all the switches, with all the links between them. We improved netloc by adding the visualization of the topologies discovered. The visualization is dynamic and the user can interact with it, to get some information about the machines, the switches or the link such as the physical address, the hostname or the speed of the link. In order to be able to do optimizations that can be helping process placement, we started to class the different topologies. For now, we only handle Clos networks [21] and we are able to transform them into fat trees. The categorization in classes permits to have a clean graph and then interact with graph partitioners.

To have a complete tool, we need to handle all major classes of topologies such as meshes, torus or hypercubes. When the graph partitioning will be integrated with tools such as SCOTCH, we will be able to find a good mapping for the processes of a job. It could also helps the resource scheduling to optimize the resource sharing between jobs. The visualization can be improved by showing the architecture information retrieved by hwloc for each machine. We can complete the visualization by giving more information especially when the original graph was transformed to simplify it, as we did to Clos networks to obtain fat trees.

6.16. Scalable mapping onto (disconnected) parts of regular target architectures

Since its inception, SCOTCH allows one to map graphs onto so-called “algorithmically-defined” target architectures. They are regular architectures such as hypercube, multi-dimensional grids and tori, butterfly networks, etc., whose characteristics are defined by subroutines which are part of the SCOTCH library. However, on today’s large-scale computer systems, software jobs do not usually run on all of the machine, but on a set of nodes assigned by the batch scheduler. Consequently, one should be able to map a process graph onto (possibly disconnected) parts of an algorithmically-defined target architecture, which was not an available feature. Only “decomposition-defined” architectures (another way to represent target architectures in SCOTCH) supported this feature, but are not scalable above a few hundred processing elements.

In order to allow SCOTCH to provide mappings onto parts of an algorithmically-defined target architecture, a new meta target architecture, called “sub”, has been created. The sub architecture allows one to restrict a regular algorithmically-defined target architecture to a subset of its vertices. Instead of using a top-down approach to build a description of the target architecture, through a recursive bipartitioning algorithm, our new algorithm uses a bottom-up approach, based on recursive matching and coarsening of neighboring vertices, much like for graph coarsening. The clustering tree is pruned of branches that lead to parts of the machine that are not allowed mapping targets. Distance between subdomains is computed using the distance function of the underlying algorithmically-defined target architecture. Preliminary results have been presented at a SIAM CS&E conference workshop [14], and a beta-version of the upcoming release 6.0.5 of SCOTCH has been shipped to early testers at Lawrence Livermore National Laboratory.

6.17. Multi-Level Parallelism in a CFD code

Code_Saturne [19] is an industrial and open source Computational Fluid Dynamics software. Developed at EDF R&D, it solves the Navier-Stokes equations for 2D, 2D-axisymmetric and 3D flows, steady or unsteady, laminar or turbulent, incompressible or weakly dilatable, isothermal or not, with scalars transport if required.

Our goal is to evaluate different ways of improving and preparing this application for the future HPC architectures. We strengthened our application knowledge by using various instrumentation tools and provided a small topology instrumentation library. As instrumentation of a full code can be a tedious thing, we provided a mini application on which to perform our future experiments. We have run experiments to determine the potential gain of topology awareness on our code by using the graph mapping solutions of PT-SCOTCH. We have also run experiments on ghost cells numbering to see the impact of their locations on cache misses.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts 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.

7.2. Bilateral Grants with Bull/Atos

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

7.3. Bilateral Grants 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.

7.4. Bilateral Grants with EDF

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

8. Partnerships and Cooperations

8.1. National Initiatives

8.1.1. ANR

ANR Songs Simulation of next generation systems (http://infra-songs.gforge.inria.fr/).

ANR INFRA 2011, 01/2012 - 12/2015 (48 months)

Identification: ANR-11INFR01306

Coordinator: Martin Quinson (Inria Nancy)

Other partners: Inria Nancy, Inria Rhône-Alpes, IN2P3, LSIIT, Inria Rennes, I3S.

Abstract: The goal of the SONGS project is to extend the applicability of the SimGrid simulation framework from Grids and Peer-to-Peer systems to Clouds and High Performance Computation systems. Each type of large-scale computing system will be addressed through a set of use cases and lead by researchers recognized as experts in this area.

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.

8.1.2. IPL - Inria Project Lab

MULTICORE - Large scale multicore virtualization for performance scaling and portability

Participants: Emmanuel Jeannot.

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.

8.2. European Initiatives

8.2.1. 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.

8.2.2. Collaborations with Major European Organizations

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

Subject 1: Application modeling for for hierarchical memory system

Partner 2: ETH Zurich (Switzerland)

Subject 2: Topology mapping

Partner 3: BSC, Barcelona (Spain)

Subject 3: High-performance communication on new architectures; load-balancing and meshing.

8.3. International Initiatives

8.3.1. Inria International Labs

JLPC Inria joint-Lab on Extreme Scale Computing:

Coordinators: Franck Cappello and Marc Snir.

Other partners: Argonne National Lab, Inria, 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.

8.3.2. Inria International Partners

8.3.2.1. 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: UWLAX (Wisconsin)

Subject 3: network topology modeling

Partner 4: Intel

Subject 4: modeling many-core platforms and next-generation memory architectures

Partner 5: University of Tokyo and Riken

Subject 5: Adaptation of MPI and runtime systems to MIC processors.

Partner 6: Lawrence Livermore National Laboratory

Subject 6: 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).

8.4. International Research Visitors

8.4.1. Visits of International Scientists

8.4.1.1. Internships

- Ivan Cores from Universidade da Coruña, Spain, visited us for 4 months and have worked on topology-aware malleability of MPI programs.
- Guillaume Houzeaux and Mariano Vazquez from BSC visited us for several days to work on particule and mesh based applications and new architectures.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific events organisation

9.1.1.1. General chair, scientific chair

TADAAM has organized the 22nd European MPI Users' Group Meeting, also known as EuroMPI. EuroMPI is the preeminent meeting for users, developers and researchers to interact and discuss new developments and applications of message-passing parallel computing, in particular in and related to the Message Passing Interface (MPI).

TADAAM has also organized the MPI Forum meeting that took place just after the EuroMPI conference.

9.1.1.2. Member of the steering committee

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

9.1.2. Scientific events selection

9.1.2.1. Chair of conference program committees

Alexandre DENIS, Brice GOGLIN, Emmanuel JEANNOT and Guillaume MERCIER acted as Program Co-chairs as well as Proceeding Editors of Euro-MPI 2015.

9.1.2.2. Member of the conference program committees

Brice GOGLIN was a member of the program committee of HotInterconnect 23, EuroMPI 2015.

Emmanuel JEANNOT was a member of the program committee of IPDPS'2015, PPAM 2015, EuroMPI 2015, Heteropar'2015, Compas 201, HiPC'15

Alexandre DENIS was a member of the program committee of HeteroPar 2015, HiPC'15, EuroMPI 2015, and Compas'15.

Guillaume MERCIER was a member of the program committee of EuroMPI 2015.

9.1.2.3. Reviewer

Emmanuel JEANNOT was reviewer of SuperComputing.

Alexandre DENIS was a reviewer for CCGrid'2015.

Guillaume MERCIER was a reviewer for IPDPS 2015.

9.1.3. Journal

9.1.3.1. Member of the editorial boards

Emmanuel JEANNOT is associate editor of the International Journal of Parallel, Emergent and Distributed Systems

9.1.4. Editing activities

Emmanuel Jeannot edited the special issue following the HeteroPar 2014, APCIE 2014, and TASUS 2014 workshops [9].

9.1.4.1. Reviewer - Reviewing activities

Brice GOGLIN was a reviewer for the IEEE Micro journal.

Guillaume MERCIER was a reviewer for the Simulation and Modelling Practice and Theory Journal (SIMPAT) and Parallel Computing (PARCO)

Emmanuel JEANNOT was a reviewer for IEEE TPDS, Parallel Computing, JPDC, ACM TACO

François Pellegrini was a reviewer for Parallel Computing.

9.1.5. Invited talks

Alexandre DENIS gave a talk about overlap of communication and computation at *CEA/DAM*.

Brice GOGLIN gave a talk about managing locality in hierarchical computing platforms at *Maison de la Simulation*.

Emmanuel JEANNOT gave an invited talk system-scale optimisation of HPC applications at the PADAL workshop in Berkeley [4].

Emmanuel JEANNOT gave an invited talk on topology-aware data management at the Sandia National Lab working group.

Emmanuel JEANNOT gave the keynote speech at the Heteropar'2015 workshop in Vienna.

François PELLEGRINI was invited to participate in a round table on the law of the Internet of things at the Law faculty of Université d'Aix-Marseille.

François PELLEGRINI gave an invited talk on legal aspects of software creation during the *Journées nationales du GDR "Génie de la Programmation et du Logiciel"* [5] in Bordeaux.

François PELLEGRINI gave an invited talk on legal aspects of software creation during the *Journées du réseau du développement logiciel (DevLOG)* [6] in Bordeaux.

François PELLEGRINI was invited to participate in a round table on the use of personal data for the personalization of healthcare treatments, in the context of the 3rd *Rencontres du droit et de l'innovation* held by the Forum Montesquieu in Bordeaux.

François PELLEGRINI was invited to give a talk on Big data and personal data, in the context of the Juriconnexion meetings [7], at École française du Barreau, in Issy-les-Moulineaux.

9.1.6. Scientific expertise

Emmanuel JEANNOT was member of the hiring committee for the professor position in computer science of the university of Bordeaux.

François PELLEGRINI is a commissioner at CNIL, the French data privacy supervision authority, where he has been appointed by the President of the French Senate in December 2013.

François PELLEGRINI, as a commissioner at CNIL, is the representative for France in the data privacy supervision bodies of several European organizations that process personal data (Europol, the Schengen information system, etc.). He has been appointed as technical expert for on-site inspection missions of such data processing systems.

9.1.7. 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).

9.1.8. Tutorials

Brice GOGLIN gave tutorials about managing hardware affinities on hierarchical platforms with HWLOC during a PRACE Advanced Training Center session and during EuroMPI. He also gave a hands-on session on advance uses of the GIT version control system in Inria Bordeaux internal seminars.

Emmanuel JEANNOT gave a tutorial about optimizing process placement with TreeMatch during a PRACE Advanced Training Center session.

[4] https://sites.google.com/a/lbl.gov/padal-workshop/padal15

[5] http://gdr-gpl.cnrs.fr/node/196

[6] http://devlog.cnrs.fr/jdev2015/t4

[7] http://www.juriconnexion.fr/programme-de-la-journee-du-8-decembre-2015-2/

9.1.9. Research administration

Emmanuel JEANNOT is member of the scientific council of the Labex IRMIA (Université 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.

François PELLEGRINI is deputy vice-president of Université de Bordeaux in charge of digital issues. In this context, he participates in the definition and implementation of policies regarding aspects of research such as the development of the academic computing center of the Bordeaux area (MCIA), relationships with other universities (UPPA, Université de La Rochelle), etc.

9.2. Teaching - Supervision - Juries

9.2.1. 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.

9.2.2. Supervision

PhD: François Tessier, Placement of parallel applications according to the topology and the affinity, Université de Bordeaux, defended Jan. 26, 2015. Advisor: Emmanuel Jeannot and Guillaume Mercier.

PhD: Paul-Antoine Arras, Ordonnancement d'applications à flux de données pour les MPSOC, Université de Bordeaux, defended Feb. 3, 2015. Advisor: Emmanuel Jeannot and Samuel Thibault.

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: Romain Prou, communication management based on remote memory access, started in 2015. Advisor: Alexandre Denis 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.

9.2.3. Juries

Brice GOGLIN was member of the PhD defense of the following candidates:

- Surya Narayanan Khizahanchery Natarajan (Inria Rennes, Reviewer)

Emmanuel JEANNOT was member of the PhD defense of the following candidates:

- Ivan Cores, (Universidade Da Coruña, Reviewer)
- Emmanuelle Saillard (Université de Bordeaux, President)
- Farouk Mansouri (Université de Grenoble, Reviewer)
- Stéfano Drimon Kurz Mór (Federal University of Rio Grande do Sul, Reviewer)

François PELLEGRINI was member of the PhD defense of the following candidates:

- François Tessier (Université de Bordeaux, President)
- Astrid Casadei (Université de Bordeaux, Member)
- Karl-Eduard Berger (CEA & Paris Saclay, Reviewer)

Brice GOGLIN was also a member of the hiring committees for the Inria Bordeaux communication department head, for a communication assistant, and for the works council (AGOS) assistant.

Emmanuel JEANNOT was member of the hiring committee of a research team assistant.

9.3. Popularization

Brice GOGLIN is in charge of the diffusion of the scientific culture for the Inria Research Center of Bordeaux. He gave numerous talks about high performance computing and research careers to general public audience and school student. He is also involved in the popularization of computer programming and robotics programming and gave several wide audience seminar on these topics.

François PELLEGRINI was invited to give a talk about privacy and information security by the students of ENS Bretagne at Rennes, in the context of their “EntretiENS” conference program.

François PELLEGRINI was a member of the jury of the Student Demo Cup at the Paris Open-Source Summit.

10. Bibliography

Major publications by the team in recent years

[1] F. BROQUEDIS, J. CLET-ORTEGA, S. MOREAUD, N. FURMENTO, B. GOGLIN, G. MERCIER, S. THIBAULT, R. NAMYST. *hwloc: a Generic Framework for Managing Hardware Affinities in HPC Applications*, in "Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP2010)", Pisa, Italia, IEEE Computer Society Press, February 2010, pp. 180–186, http://hal.inria.fr/inria-00429889

[2] B. GOGLIN. *Managing the Topology of Heterogeneous Cluster Nodes with Hardware Locality (hwloc)*, in "Proceedings of 2014 International Conference on High Performance Computing & Simulation (HPCS 2014)", Bologna, Italy, July 2014, pp. 74–81, http://hal.inria.fr/hal-00985096

[3] E. JEANNOT, E. MENESES, G. MERCIER, F. TESSIER, G. ZHENG. *Communication and Topology-aware Load Balancing in Charm++ with TreeMatch*, in "IEEE Cluster", Indianapolis, IN, USA, September 2013, 8 p.

[4] E. JEANNOT, G. MERCIER, F. TESSIER. *Process Placement in Multicore Clusters: Algorithmic Issues and Practical Techniques*, in "IEEE Transactions on Parallel and Distributed Systems", April 2014, vol. 25, n^{o} 4, pp. 993–1002 [*DOI :* 10.1109/TPDS.2013.104]

[5] A. TATE, A. KAMIL, A. DUBEY, A. GRÖSSLINGER, B. CHAMBERLAIN, B. GOGLIN, H. C. EDWARDS, C. J. NEWBURN, D. PADUA, D. UNAT, E. JEANNOT, F. HANNIG, T. GYSI, H. LTAIEF, J. SEXTON, J. LABARTA, J. SHALF, K. FÜRLINGER, K. O’BRIEN, L. LINARDAKIS, M. BESTA, M.-C. SAWLEY, M. ABRAHAM, M. BIANCO, M. PERICÀS, N. MARUYAMA, P. H. J. KELLY, P. MESSMER, R. B. ROSS, R. CLEDAT, S. MATSUOKA, T. SCHULTHESS, T. HOEFLER, V. J. LEUNG. *Programming Abstractions for Data Locality*, PADAL Workshop 2014, April 28–29, Swiss National Supercomputing Center (CSCS), Lugano, Switzerland, November 2014, http://hal.inria.fr/hal-01083080

Publications of the year

Doctoral Dissertations and Habilitation Theses

[6] P.-A. ARRAS. *Scheduling of dynamic streaming applications on hybrid embedded MPSoCs comprising programmable computing units and hardware accelerators*, Université de Bordeaux, February 2015, https://tel.archives-ouvertes.fr/tel-01159519

[7] F. TESSIER. *Placement of parallel applications according to the topology and the affinity*, Université de Bordeaux, January 2015, https://tel.archives-ouvertes.fr/tel-01174693

Articles in International Peer-Reviewed Journals

[8] P.-A. ARRAS, D. FUIN, E. JEANNOT, A. STOUTCHININ, S. THIBAULT. *List Scheduling in Embedded Systems Under Memory Constraints*, in "International Journal of Parallel Programming", December 2015, vol. 43, n^{o} 6, pp. 1103-1128 [*DOI :* 10.1007/s10766-014-0338-1], https://hal.inria.fr/hal-01087067

Articles in Non Peer-Reviewed Journals

[9] 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

International Conferences with Proceedings

[10] 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

[11] A. DENIS. *pioman: a pthread-based Multithreaded Communication Engine*, in "Euromicro International Conference on Parallel, Distributed and Network-based Processing", Turku, Finland, March 2015, https://hal.inria.fr/hal-01087775

[12] N. DENOYELLE, B. GOGLIN, E. JEANNOT. *A Topology-Aware Performance Monitoring Tool for Shared Resource Management in Multicore Systems*, in "Proceedings of Euro-Par 2015: Parallel Processing Workshops", Vienna, Austria, Lecture Notes in Computer Science, Springer, August 2015, https://hal.inria.fr/hal-01183083

Conferences without Proceedings

[13] E. JEANNOT. *Topology Aware Process Placement and Data Management*, in "SIAM Conference on Computational Science & Engineering, SIAM CSE '15", Salt-Lake City, United States, March 2015, https://hal.inria.fr/hal-01252734

[14] F. PELLEGRINI, C. LACHAT. *Process Mapping onto Complex Architectures and Partitions Thereof*, in "Computer Science & Engineering", Salt Lake City, United States, SIAM, March 2015, https://hal.inria.fr/hal-01253509

Scientific Books (or Scientific Book chapters)

[15] *Proceedings of the 22nd European MPI Users' Group Meeting*, ACM, Bordeaux, France, September 2015, 149 p. , https://hal.inria.fr/hal-01252232

[16] P. Bouvry, G. L. Tsafack Chetsa, G. Da Costa, E. Jeannot, L. Lefèvre, J.-M. Pierson, F. Pinel, P. Stolf, S. Varrette. *Energy efficiency and high-performance computing*, in "Large-scale Distributed Systems and Energy efficiency", Wiley, 2015, https://hal.inria.fr/hal-01251988

[17] N. Denoyelle, M. Carrere, F. Pouget, T. Viéville, F. Alexandre. *From biological to numerical experiments in systemic neuroscience: a simulation platform*, in "Advances in Neurotechnology, Electronics and Informatics", A. Londral, P. Encarnação (editors), Biosystems & Biorobotics, Springer, November 2015, vol. 12, https://hal.inria.fr/hal-01227968

Other Publications

[18] R. Prou. *Gestion des communications centrée sur les accès mémoire à distance*, Université d'Orléans, September 2015, https://hal.inria.fr/hal-01252752

References in notes

[19] F. Archambeau, N. Méchitoua, M. Sakiz. *Code Saturne: A finite volume code for the computation of turbulent incompressible flows-Industrial applications*, in "International Journal on Finite Volumes", 2004, vol. 1, n^{O} 1, pp. http–www

[20] P.-N. Clauss, J. Gustedt. *Iterative computations with ordered read–write locks*, in "Journal of Parallel and Distributed Computing", 2010, vol. 70, n^{O} 5, pp. 496 - 504 [*DOI :* 10.1016/j.jpdc.2009.09.002], http://www.sciencedirect.com/science/article/pii/S0743731509001671

[21] C. Clos. *A Study of Non-Blocking Switching Networks*, in "Bell System Technical Journal", 1953, vol. 32, n^{O} 2, pp. 406–424

[22] M. Pérache, H. Jourdren, R. Namyst. *MPC: A Unified Parallel Runtime for Clusters of NUMA Machines*, in "Proceedings of the 14th International Euro-Par Conference on Parallel Processing", Berlin, Heidelberg, Euro-Par '08, Springer-Verlag, 2008, pp. 78–88, http://dx.doi.org/10.1007/978-3-540-85451-7_9

[23] S. Williams, A. Waterman, D. Patterson. *Roofline: an insightful visual performance model for multicore architectures*, in "Communications of the ACM", 2009, vol. 52, n^{O} 4, pp. 65–76

[24] A. B. Yoo, M. A. Jette, M. Grondona. *Slurm: Simple linux utility for resource management*, in "Job Scheduling Strategies for Parallel Processing", Springer, 2003, pp. 44–60

[25] R. Zhang, J. Ding. *Object Tracking and Detecting Based on Adaptive Background Subtraction*, in "Procedia Engineering", 2012, vol. 29, pp. 1351 - 1355, 2012 International Workshop on Information and Electronics Engineering [*DOI :* 10.1016/j.proeng.2012.01.139], http://www.sciencedirect.com/science/article/pii/S187770581200149X