IN PARTNERSHIP WITH:
Institut Polytechnique de Bordeaux

Université de Bordeaux

Activity Report 2019

Project-Team TADAAM

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

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

RESEARCH CENTER
Bordeaux - Sud-Ouest

THEME
Distributed and High Performance Computing

Table of contents

Project-Team TADAAM

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

Keywords:

Computer Science and Digital Science:

A1.1.1. - Multicore, Manycore
A1.1.2. - Hardware accelerators (GPGPU, FPGA, etc.)
A1.1.3. - Memory models
A1.1.4. - High performance computing
A1.1.5. - Exascale
A1.1.9. - Fault tolerant systems
A1.2. - Networks
A2.1.7. - Distributed programming
A2.2.2. - Memory models
A2.2.4. - Parallel architectures
A2.2.5. - Run-time systems
A2.6.1. - Operating systems
A2.6.2. - Middleware
A3.1.2. - Data management, quering and storage
A3.1.3. - Distributed data
A3.1.8. - Big data (production, storage, transfer)
A6.2.6. - Optimization
A6.2.7. - High performance computing
A6.3.3. - Data processing
A7.1.1. - Distributed algorithms
A7.1.2. - Parallel algorithms
A7.1.3. - Graph algorithms
A8.1. - Discrete mathematics, combinatorics
A8.2. - Optimization
A8.7. - Graph theory
A8.9. - Performance evaluation

Other Research Topics and Application Domains:

B6.3.2. - Network protocols
B6.3.3. - Network Management
B6.5. - Information systems
B9.5.1. - Computer science
B9.8. - Reproducibility

1. Team, Visitors, External Collaborators

Research Scientists

Emmanuel Jeannot [Team leader, Inria, Senior Researcher, HDR]

Alexandre Denis [Inria, Researcher]
Brice Goglin [Inria, Senior Researcher, HDR]
Guillaume Pallez Aupy [Inria, Researcher]

Faculty Members

Guillaume Mercier [Institut National Polytechnique de Bordeaux, Associate Professor, HDR]
François Pellegrini [U. Bordeaux, Professor, HDR]
Francieli Zanon-Boito [U. Bordeaux, Associate Professor, from Sep 2019]

Post-Doctoral Fellow

Julien Herrmann [Inria, until Sep 2019]

PhD Students

Valentin Honoré [U. Bordeaux]
Florian Reynier [CEA]
Philippe Swartvagher [Inria, from Oct 2019]
Andres Xavier Rubio Proano [Inria]
Nicolas Vidal [Inria]

Technical staff

Adrien Guilbaud [Inria, Engineer]

Interns and Apprentices

Fatima El Akkary [Inria, from Apr 2019 to Jun 2019]
Valentin Hoyet [Inria]
Amaury Jacques [Inria, from Feb 2019 to May 2019]
Philippe Swartvagher [Inria, from Feb 2019 to Jul 2019]

2. Overall Objectives

2.1. Overall Objectives

In TADAAM, we propose a new approach where we allow the application to explicitly express its resource needs about its execution. The application needs to express its behavior, but in a different way from the compute-centric approach, as the additional information is not necessarily focused on computation and on instructions execution, but follows a high-level semantics (needs of large memory for some processes, start of a communication phase, need to refine the granularity, beginning of a storage access phase, description of data affinity, etc.). These needs will be expressed to a service layer though an API. The service layer will be system-wide (able to gather a global knowledge) and stateful (able to take decision based on the current request but also on previous ones). The API shall enable the application to access this service layer through a well-defined set of functions, based on carefully designed abstractions.

Hence, **the goal of** TADAAM **is to design a stateful system-wide service layer for HPC systems, in order to optimize applications execution according to their needs**.

This layer will abstract low-level details of the architecture and the software stack, and will allow applications to register their needs. Then, according to these requests and to the environment characteristics, this layer will feature an engine to optimize the execution of the applications at system-scale, taking into account the gathered global knowledge and previous requests.

This approach exhibits several key characteristics:

- It is independent from the application parallelization, the programming model, the numerical scheme and, largely, from the data layout. Indeed, high-level semantic requests can easily be added to the application code after the problem has been modeled, parallelized, and most of the time after the data layout has been designed and optimized. Therefore, this approach is – to a large extent – orthogonal to other optimization mechanisms and does not require application developers to rewrite their code.
- Application developers are the persons who know best their code and therefore the needs of their application. They can easily (if the interface is well designed and the abstractions are correctly exposed), express the application needs in terms of resource usage and interaction with the whole environment.
- Being stateful and shared by all the applications in the parallel environment, the proposed layer will therefore enable optimizations that:
  - cannot be performed statically but require information only known at launch- or run-time,
  - are incremental and require minimal changes to the application execution scheme,
  - deal with several parts of the environment at the same time (e.g., batch scheduler, I/O, process manager and storage),
  - take into account the needs of several applications at the same time and deal with their interaction. This will be useful, for instance, to handle network contention, storage access or any other shared resources.

3. Research Program

3.1. Need for System-Scale Optimization

Firstly, in order for applications to make the best possible use of the available resources, it is impossible to expose all the low-level details of the hardware to the program, as it would make impossible to achieve portability. Hence, the standard approach is to add intermediate layers (programming models, libraries, compilers, runtime systems, etc.) to the software stack so as to bridge the gap between the application and the hardware. With this approach, optimizing the application requires to express its parallelism (within the imposed programming model), organize the code, schedule and load-balance the computations, etc. In other words, in this approach, the way the code is written and the way it is executed and interpreted by the lower layers drives the optimization. In any case, this approach is centered on how computations are performed. Such an approach is therefore no longer sufficient, as the way an application is executing does depend less and less on the organization of computation and more and more on the way its data is managed.

Secondly, modern large-scale parallel platforms comprise tens to hundreds of thousand nodes [1]. However, very few applications use the whole machine. In general, an application runs only on a subset of the nodes [2]. Therefore, most of the time, an application shares the network, the storage and other resources with other applications running concurrently during its execution. Depending on the allocated resources, it is not uncommon that the execution of one application interferes with the execution of a neighboring one.

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

[1] More than 22,500 XE6 compute node for the BlueWaters system; 5040 B510 Bullx Nodes for the Curie machine; more than 49,000 BGQ nodes for the MIRA machine.

[2] In 2014, the median case was 2048 nodes for the BlueWaters system and, for the first year of the Curie machine, the median case was 256 nodes

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

3.2. Scientific Challenges and Research Issues

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

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

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

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

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

Therefore, the second question we need to answer is: "**how to build a system-scale, stateful, shared layer that can gather applications needs expressed with a high-level semantic?**". This work will require not only to define a clean API where applications will express their needs, but also to define how such a layer will be shared across applications and will scale on future systems. The API will provide a simple yet effective way to express different needs such as: memory usage of a given portion of the code; start of a compute intensive part; phase where the network is accessed intensively; topology-aware affinity management; usage of storage (in read and/or write mode); change of the data layout after mesh refinement, etc. From an engineering point of view, the layer will have a hierarchical design matching the hardware hierarchy, so as to achieve scalability.

Once this has been done, the service layer, will have all the information about the environment characteristics and application requirements. We therefore need to design a set of **mechanisms to optimize applications execution**: communication, mapping, thread scheduling, data partitioning/mapping/movement, etc.

Hence, the last scientific question we will address is: "**How to design fast and efficient algorithms, mechanisms and tools to enable execution of applications at system-scale, in full a HPC ecosystem, taking into account topology and locality?**" A first set of research is related to thread and process placement according to the topology and the affinity. Another large field of study is related to data placement, allocation and partitioning: optimizing the way data is accessed and processed especially for mesh-based applications. The issues of transferring data across the network will also be tackled, thanks to the global knowledge we

have on the application behavior and the data layout. Concerning the interaction with other applications, several directions will be tackled. Among these directions we will deal with matching process placement with resource allocation given by the batch scheduler or with the storage management: switching from a best-effort application centric strategy to global optimization scheme.

4. Application Domains

4.1. Mesh-based applications

TADAAM targets scientific simulation applications on large-scale systems, as these applications present huge challenges in terms of performance, locality, scalability, parallelism and data management. Many of these HPC applications use meshes as the basic model for their computation. For instance, PDE-based simulations using finite differences, finite volumes, or finite elements methods operate on meshes that describe the geometry and the physical properties of the simulated objects. This is the case for at least two thirds of the applications selected in the 9^{th} PRACE. call [3], which concern quantum mechanics, fluid mechanics, climate, material physic, electromagnetism, etc.

Mesh-based applications not only represent the majority of HPC applications running on existing supercomputing systems, yet also feature properties that should be taken into account to achieve scalability and performance on future large-scale systems. These properties are the following:

Size Datasets are large: some meshes comprise hundreds of millions of elements, or even billions.

Dynamicity In many simulations, meshes are refined or coarsened at each time step, so as to account for the evolution of the physical simulation (moving parts, shockwaves, structural changes in the model resulting from collisions between mesh parts, etc.).

Structure Many meshes are unstructured, and require advanced data structures so as to manage irregularity in data storage.

Topology Due to their rooting in the physical world, meshes exhibit interesting topological properties (low dimensionality embedding, small maximum degree, large diameter, etc.). It is very important to take advantage of these properties when laying out mesh data on systems where communication locality matters.

All these features make mesh-based applications a very interesting and challenging use-case for the research we want to carry out in this project. Moreover, we believe that our proposed approach and solutions will contribute to enhance these applications and allow them to achieve the best possible usage of the available resources of future high-end systems.

5. Highlights of the Year

5.1. Highlights of the Year

- Guillaume PALLEZ was an invited speaker at the Royal Society https://royalsociety.org/science-events-and-lectures/2019/04/high-performance-computing/
- Brice GOGLIN is co-chair of the Architecture & Networks area of the SuperComputing 2020 conference.
- François PELLEGRINI has been re-appointed a member of the French *Commission Nationale de l'Informatique et des Libertés* (French data protection authority) by the President of the French Senate.

5.1.1. Awards

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

- Guillaume PALLEZ was one of the recipient of the IEEE Computer Society TCHPC Early Career Researchers Award for Excellence in High Performance Computing
- François PELLEGRINI was bestowed *Chevalier dans l'Ordre des Palmes Académiques* (Order of Academic Palms), promotion of July 2019.

6. New Software and Platforms

6.1. Hsplit

Hardware communicators split

KEYWORDS: MPI communication - Topology - Hardware platform

SCIENTIFIC DESCRIPTION: Hsplit is a library that implements an abstraction allowing the programmer using MPI in their parallel applications to access the underlying hardware structure through a hierarchy of communicators. Hsplit is based on the MPI_Comm_split_type routine and provides a new value for the split_type argument that specifically creates a hierarchy a subcommunicators where each new subcommunicator corresponds to a meaningful hardware level. The important point is that only the structure o the hardware is exploited and the number of levels or the levels names are not fixed so as to propose a solution independent from future hardware evolutions (such as new levels for instance). Another flavor of this MPI_Comm_split_type function is provided that creates a roots communicators at the same time a subcommunicator is produced, in order to ease the collective communication and/or synchronization among subcommunicators.

FUNCTIONAL DESCRIPTION: Hsplit implements an abstraction that allows the programmer using MPI in their parallel applications to access the underlying hardware structure through a hierarchy of communicators. Hsplit is based on the MPI_Comm_split_type routine and provides a new value for the split_type argument that specifically creates a hierarchy a subcommunicators where each new subcommunicator corresponds to a meaningful hardware level. The important point is that only the structure o the hardware is exploited and the number of levels or the levels names are not fixed so as to propose a solution independent from future hardware evolutions (such as new levels for instance). Another flavor of this MPI_Comm_split_type function is provided that creates a roots communicators at the same time a subcommunicator is produced, in order to ease the collective communication and/or synchronization among subcommunicators.

NEWS OF THE YEAR: Most of our proposal had been officially read in front of the MPI Forum at the last physical meeting in December at Albuquerque. This concerns hte guided and the unguided mode of the split function. This now has to pass two votes in the next physical meetings in 2020 to be part of the new version of the standard: MPI 4.0 that shall be ratified and released at the end of 2020. Since no other MPI library currently implements the unguided mode, Hsplit will be the only sofware that is currently able to provide it.

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

6.2. hwloc

Hardware Locality

KEYWORDS: NUMA - Multicore - GPU - Affinities - Open MPI - Topology - HPC - Locality

FUNCTIONAL DESCRIPTION: Hardware Locality (hwloc) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices. It builds a widely-portable abstraction of these resources and exposes it to applications so as to help them adapt their behavior to the hardware characteristics. They may consult the hierarchy of resources, their attributes, and bind task or memory on them.

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

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

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

6.3. NetLoc

Network Locality

KEYWORDS: Topology - Locality - Distributed networks - HPC - Parallel computing - MPI communication

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

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

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

Netloc is distributed within hwloc releases starting with hwloc 2.0.

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

6.4. NewMadeleine

NewMadeleine: An Optimizing Communication Library for High-Performance Networks

KEYWORDS: High-performance calculation - MPI communication

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

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

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

- Participants: Alexandre Denis, Clément Foyer, Nathalie Furmento, Raymond Namyst, Adrien Guilbaud, Florian Reynier and Philippe Swartvagher
- Contact: Alexandre Denis
- Publications: NewMadeleine: a Fast Communication Scheduling Engine for High Performance Networks - Ordonnancement et qualité de service pour réseaux rapides - Improving Reactivity and Communication Overlap in MPI using a Generic I/O Manager - PIOMan : un gestionnaire d'entrées-sorties générique - A multithreaded communication engine for multicore architectures - A multicore-enabled multirail communication engine - About the interactions between communication and thread scheduling in clusters of multicore machines - Scalability of the NewMadeleine Communication Library for Large Numbers of MPI Point-to-Point Requests - An analysis of the impact of multithreading on communication performance - A scalable and generic task scheduling system for communication libraries - A Generic and High Performance Approach for Fault Tolerance in Communication Library - A High-Performance Superpipeline Protocol for InfiniBand - A sampling-based approach for communication libraries auto-tuning - High performance checksum computation for fault-tolerant MPI over InfiniBand - pioman: a Generic Framework for Asynchronous Progression and Multithreaded Communications - pioman: a pthread-based Multithreaded Communication Engine - Updating MadMPI to MPI-3: Remote Memory Access - Portage de StarPU sur la bibliothèque de communication NewMadeleine
- URL: http://pm2.gforge.inria.fr/newmadeleine/

6.5. PaMPA

Parallel Mesh Partitioning and Adaptation

KEYWORDS: Dynamic load balancing - Unstructured heterogeneous meshes - Parallel remeshing - Subdomain decomposition - Parallel numerical solvers

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

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

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

NEWS OF THE YEAR: PaMPA has been used to remesh an industrial mesh of a helicopter turbine combustion chamber, up to more than 1 billion elements.

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

6.6. TopoMatch

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

SCIENTIFIC DESCRIPTION: TreeMatch embeds a set of algorithms to map processors/cores in order to minimize the communication cost of the application.

Important features are : the number of processors can be greater than the number of applications processes , it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) , it implements different placement algorithms that are switched according to the input size.

Some core algorithms are parallel to speed-up the execution. Optionally embeds scotch for fix-vertex mapping. enable exhaustive search if required. Several metric mapping are computed. Allow for oversubscribing of ressources. multithreaded.

TreeMatch is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI and MPICH2.

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

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

6.7. SCOTCH

KEYWORDS: Mesh partitioning - Domain decomposition - Graph algorithmics - High-performance calculation - Sparse matrix ordering - Static mapping

FUNCTIONAL DESCRIPTION: Scotch is a graph partitioner. It helps optimise the division of a problem, by means of a graph, into a set of independent sub-problems of equivalent sizes. These sub-problems can also be solved in parallel.

RELEASE FUNCTIONAL DESCRIPTION: Version 6.0 offers many new features:

sequential graph repartitioning

sequential graph partitioning with fixed vertices

sequential graph repartitioning with fixed vertices

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

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

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

embedded pseudo-random generator for improved reproducibility

and even more...

NEWS OF THE YEAR: In 2019, several versions of Scotch have been released, from v6.0.7 up to v6.0.9. While they are mostly bugfix updates, several new features and API routines have been added, to increase its use by third-party software, notably routines handling target topologies. Also, code quality has been improved by the addition of many tests in the continuous integration process. A new graphical system has been developed by Amaury Jacques (Inria intern, Feb.-May 2019) to display differences in result quality across versions and builds. This system has been adopted by other Inria projects.

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

6.8. disk-revolve

KEYWORDS: Automatic differentiation - Gradients - Machine learning

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

- Authors: Guillaume Aupy and Julien Herrmann
- Contact: Julien Herrmann
- Publications: H-Revolve: A Framework for Adjoint Computation on Synchrone Hierarchical Platforms - Periodicity in optimal hierarchical checkpointing schemes for adjoint computations - Optimal Multistage Algorithm for Adjoint Computation
- URL: https://gitlab.inria.fr/adjoint-computation/disk-revolve-public

7. New Results

7.1. Management of heterogeneous and non-volatile memories in HPC

The emergence of non-volatile memory that may be used either as fast storage or slow high-capacity memory brings many opportunities for application developers.

We studied the impact of those new technologies on the allocation of resources in HPC platforms. We showed that co-scheduling HPC applications will possibly different needs in term of storage and memories brings constraints of the way non-volatile memory should be exposed by the hardware and operating system to bring both flexibility and performance. [21]

We also worked with Lawrence Livermore National Lab to propose an API to help application choose between the different kinds of available memory (high-bandwidth (HBM), normal (DDR), slow (non-volatile)). We exposed several useful criteria for selecting target memories as well as ways to rank them. [22]

7.2. Modeling and Visualizing Many-core HPC Platforms

As the number of cores keeps increasing inside processors, new kinds of hierarchy are added to organize and interconnect them. We worked with Intel to leverage new groups of cores such as *Dies* in newest Xeon Advanced Performance models. We also designed ways to clarify the modeling and visualisation of those many cores by factorizing identical parts of the platforms.

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

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

In collaboration with ENS Lyon and Georgia Tech, we looked at the interferences in the last level of cache (LLC) and use the *Cache Allocation Technology* (CAT) recently provided by Intel to partition the LLC and give each co-scheduled application their own cache area.

We considered m iterative HPC applications running concurrently and answer the following questions: (i) how to precisely model the behavior of these applications on the cache partitioned platform? and (ii) how many cores and cache fractions should be assigned to each application to maximize the platform efficiency? Here, platform efficiency is defined as maximizing the performance either globally, or as guaranteeing a fixed ratio of iterations per second for each application. Through extensive experiments using CAT, we demonstrated the impact of cache partitioning when multiple HPC application are co-scheduled onto CMP platforms. [2]

7.4. Modeling High-throughput Applications for in situ Analytics

In this work [3], we proposed to model HPC applications in the framework of in situ analytics. Typically, an HPC application is composed of a simulation tasks (data and compute intensive), and a set of analysis tasks that post-process the data. Currently, the performance of the I/O system in HPC platform prohibits the storage of all simulation data to process analysis post-mortem. Hence, in situ framework proposes to treat the data "on the fly", directly where it is produced. Hence, it leverages the amount of data to store as we only keep the result of analytics phase. However, simulation and analysis have to be scheduled in parallel and compete for shared resources. It generates resource conflicts and can lead to severe performance degradation for the simulation.

Hence, we proposed to model both platform (number of nodes and cores, memory, etc) and application (profile of each tasks) in order to optimize the execution of such applications. We propose a resource partitioning model that affects computational resources to the different tasks, as so as a scheduling of those tasks in order to maximize resource usage and minimize total application makespan. Tasks are assumed to be fully parallel to solve the partitioning problem.

We evaluated different scheduling heuristics combined to the resource partitioning model and show important features that influence in situ analytics performance.

This work is done in collaboration with Bruno RAFFIN from Inria team DATAMOVE of Inria Grenoble.

7.5. Modeling Non-Uniform Memory Access and Heterogeneous Memories on Large Compute Nodes with the Cache-Aware Roofline Model

The trend of increasing the number of cores on-chip is enlarging the gap between compute power and memory performance. This issue leads to design systems with heterogeneous memories, creating new challenges for data locality. Before the release of those memory architectures, the Cache-Aware Roofline Model [43] (CARM) offered an insightful model and methodology to improve application performance with knowledge of the cache memory subsystem.

With the help of the HWLOC library, we are able to leverage the machine topology to extend the CARM for modeling NUMA and heterogeneous memory systems, by evaluating the memory bandwidths between all combinations of cores and NUMA nodes. The new Locality Aware Roofline Model [6] (LARM) scopes most contemporary types of large compute nodes and characterizes three bottlenecks typical of those systems, namely contention, congestion and remote access. We also designed a hybrid memory bandwidth model to better estimate the roof when heterogeneous memories are involved or when read and write bandwidths differ.

We also developed an hybrid bandwidth model that combines the performance of different memories and their respective read/write bandwidth with the application memory access pattern to predict the performance of these accesses on heterogeneous memory platforms.

This work has been achieved in collaboration with the authors of the CARM from University of Lisbon.

7.6. Statistical Learning for Task and Data Placement in NUMA Architecture

Achieving high performance for multi-threaded application requires both a careful placement of threads on computing units and a thorough allocation of data in memory. Finding such a placement is a hard problem to solve, because performance depends on complex interactions in several layers of the memory hierarchy.

We proposed a black-box approach to decide if an application execution time can be impacted by the placement of its threads and data, and in such a case, to choose the best placement strategy to adopt [18]. We show that it is possible to reach near-optimal placement policy selection by looking at hardware performance counters, and at counters obtained from application instrumentation. Furthermore, solutions work across several recent processor architectures (from Haswell to Skylake), across several applications, and decisions can be taken with a single run of low overhead profiling.

This work has been achieved in collaboration with Thomas ROPARS from University of Grenoble.

7.7. On-the-fly scheduling vs. reservation-based scheduling for unpredictable workflows

Scientific insights in the coming decade will clearly depend on the effective processing of large datasets generated by dynamic heterogeneous applications typical of workflows in large data centers or of emerging fields like neuroscience. In this work [8], we show how these big data workflows have a unique set of characteristics that pose challenges for leveraging HPC methodologies, particularly in scheduling. Our findings indicate that execution times for these workflows are highly unpredictable and are not correlated with the size of the dataset involved or the precise functions used in the analysis. We characterize this inherent variability and sketch the need for new scheduling approaches by quantifying significant gaps in achievable performance. Through simulations, we show how on-the-fly scheduling approaches can deliver benefits in both system-level and user-level performance measures. On average, we find improvements of up to 35% in system utilization and up to 45% in average stretch of the applications, illustrating the potential of increasing performance through new scheduling approaches.

7.8. Scheduling strategies for stochastic jobs

Following the observations of made in 7.7, we studied stochastic jobs (coming from neuroscience applications) which we want to schedule on a reservation-based platform (e.g. cloud, HPC).

The execution time of jobs is modeled using a (known) probability distribution. The platform to run the job is reservation-based, meaning that the user has to request fixed-length time slots for its job to be executed. The aim of this project is to study efficient strategies of reservation for an user given the cost associated to the machine. These reservations are all paid until a job is finally executed.

As a first step we derived efficient strategies without any additional asumptions [15]. This allowed us to set up properly the problem. These strategies were general enough that they could take as input any probability distributions, and performed better than any more natural strategies. Then we extended our strategies by including checkpoint/restart to well-chosen reservations in order to avoid wasting the benefits of work during underestimated reservations [35]. We were able to develop a fully polynomial-time approximation for continuous distribution of job execution time whose performance we then experimentally studied.

The final works of this project focused on the case without checkpointing: we studied experimentally how the strategies developed in [15] would perform in a parallel setup and showed that they improve both system utilization and job response time. Finally we started to study the robustness of such solutions when the job distributions were not perfectly known [19] and observed that the performance were still correct even with a very low quantity of information.

7.9. Online Prediction of Network Utilization

Stealing network bandwidth helps a variety of HPC runtimes and services to run additional operations in the background without negatively affecting the applications. A key ingredient to make this possible is an accurate prediction of the future network utilization, enabling the runtime to plan the background op- erations in advance, such as to avoid competing with the application for network bandwidth. In this work [23], we have proposed a portable deep learning predictor that only uses the information available through MPI introspection to construct a recurrent sequence-to-sequence neural network capable of forecasting network utilization. We leverage the fact that most HPC applications exhibit periodic behaviors to enable predictions far into the future (at least the length of a period). Our online approach does not have an initial training phase, it continuously improves itself during application execution without incurring significant computational overhead. Experimental results show better accuracy and lower computational overhead compared with the state-of-the-art on two representative applications.

7.10. An Introspection Monitoring Library

In this work [36] we have described how to improve communication time of MPI parallel applications with the use of a library that enables to monitor MPI applications and allows for introspection (the program itself can query the state of the monitoring system). Based on previous work, this library is able to see how collective communications are decomposed into point-to-point messages. It also features monitoring sessions that allow suspending and restarting the monitoring, limiting it to specific portions of the code. Experiments show that the monitoring overhead is very small and that the proposed features allow for dynamic and efficient rank reordering enabling up to 2-time reduction of communication parts of some program.

7.11. Tag matching in constant time

Tag matching is the operation, inside an MPI library, of pairing a packet arriving from the network, with its corresponding receive request posted by the user. This operation is not straightforward given that matching criterions are the communicator, the source of the message, a user-supplied tag, and since there are wildcards for tag and source. State of the art algorithms are linear with the number of pending packets and requests, or don't support wildcards.

We proposed [17] an algorithm that is able perform the matching operation in constant time, in all cases, even with wildcard requests. We implemented the algorithm in our NEWMADELEINE communication library, and demonstrated it actually improves performance of Cholesky factorization with CHAMELEON running on top of STARPU.

7.12. Dynamic broadcasts in StarPU/NewMadeleine

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

We proposed [42], a dynamic broadcast algorithm that runs without synchronization among participants, and where only the root node needs to know the others. Recipient don't even have to know whether the message will arrive as a plain send/receive or through a dynamic broadcast, which allows for a seamless integration in StarPU. We implemented the algorithm in our NewMadeleine communication library, leveraging its event-based paradigm and background progression of communications. Preliminary experiments using Cholesky factorization from the Chameleon library show a sensible performance improvement.

7.13. Task based asynchronous MPI collectives optimisation

Asynchronous collectives are more complex than plain non-blocking point-to-point communications. They need specific mechanisms for progression. Task based progression is a good way to improve the performance of applications with overlap.

We worked on a benchmarking tool [41] measuring specific collective overlapping, taking into account time shift between different nodes. Using this tool, we were able to experiment with different task execution policies in the NewMadeleine communication library.

We propose a progression policy consisting of a dedicated a core for progression tasks; modern processors have more and more cores, so it is profitable on that kind of processors. The only function of this core is to progress communications, so we use a particularly aggressive algorithm for this progression.

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

To amortize the cost of MPI collective operations, non-blocking collectives have been proposed so as to allow communications to be overlapped with computation. Unfortunately, collective communications are more CPU-hungry than point-to-point communications and running them in a communication thread on a single dedicated CPU core makes them slow. On the other hand, running collective communications on the application cores leads to no overlap. To address these issues, we proposed [5] an algorithm for tree-based collective operations that splits the tree between communication cores and application cores. To get the best of both worlds, the algorithm runs the short but heavy part of the tree on application cores, and the long but narrow part of the tree on one or several communication cores, so as to get a trade-off between overlap and absolute performance. We provided a model to study and predict its behavior and to tune its parameters. We implemented it in the MPC framework, which is a thread-based MPI implementation. We have run benchmarks on manycore processors such as the KNL and Skylake and got good results both in terms of performance and overlap.

7.15. Dynamic placement of Hybrid MPI +X coupled applications

We continued our collaboration with CERFACS in order to propose the Hippo software that addresses the issue of dynamic placement of computing kernels that feature each their own placement/mapping/binding policy of MPI processes and OpenMP threads. In such a case, enforcing a global placement policy for the whole application composed of several such kernels may be detrimental to the overall performance. Hippo (based on our Hsplit library and the hwloc software) is able to make the selection of the relevent resource on which some master MPI processes are going to execute and spawn OpenMP parallel sections while the remaining MPI processes are put in a "quiescence" state. Hippo is currently at the prototype stage and the interface and the set of provided functionnalities need some refinement, however, preliminary results are very encouraging, especially on climate modelling applications from Météo France.

7.16. Scheduling on Two Unbounded Resources with Communication Costs

Heterogeneous computing systems are popular and powerful platforms, containing several heterogeneous computing elements (e.g. CPU+GPU). In [13], we consider a platform with two types of machines , each containing an unbounded number of elements. We want to execute an application represented as a Directed Acyclic Graph (DAG) on this platform. Each task of the application has two possible execution times,

depending on the type of machine it is executed on. In addition we consider a cost to transfer data from one platform to the other between successive tasks. We aim at minimizing the execution time of the DAG (also called makespan). We show that the problem is NP-complete for graphs of depth at least three but polynomial for graphs of depth at most two. In addition, we provide polynomial-time algorithms for some usual classes of graphs (trees, series-parallel graphs).

7.17. H-Revolve: A Framework for Adjoint Computation on Synchrone Hierarchical Platforms

In this work [38], we study the problem of checkpointing strategies for adjoint computation on synchrone hierarchical platforms. Specifically we consider computational platforms with several levels of storage with different writing and reading costs. When reversing a large adjoint chain, choosing which data to checkpoint and where is a critical decision for the overall performance of the computation. We introduce H-Revolve, an optimal algorithm for this problem. We make it available in a public Python library along with the implementation of several state-of-the-art algorithms for the variant of the problem with two levels of storage. We provide a detailed description of how one can use this library in an adjoint computation software in the field of automatic differentiation or backpropagation. Finally, we evaluate the performance of H-Revolve and other checkpointing heuristics though an extensive campaign of simulation.

7.18. Sizing and Partitioning Strategies for Burst-Buffers to Reduce IO Contention

Burst-Buffers are high throughput and small size storage which are being used as an intermediate storage between the PFS (Parallel File System) and the computational nodes of modern HPC systems. They can allow to hinder to contention to the PFS, a shared resource whose read and write performance increase slower than processing power in HPC systems. A second usage is to accelerate data transfers and to hide the latency to the PFS. In this work [14], we concentrate on the first usage. We propose a model for Burst-Buffers and application transfers. We consider the problem of dimensioning and sharing the Burst-Buffers between several applications. This dimensioning can be done either dynamically or statically. The dynamic allocation considers that any application can use any available portion of the Burst-Buffers. The static allocation considers that when a new application enters the system, it is assigned some portion of the Burst-Buffers, which cannot be used by the other applications until that application leaves the system and its data is purged from it. We show that the general sharing problem to guarantee fair performance for all applications is an NP-Complete problem. We propose a polynomial time algorithms for the special case of finding the optimal buffer size such that no application is slowed down due to PFS contention, both in the static and dynamic cases. Finally, we provide evaluations of our algorithms in realistic settings. We use those to discuss how to minimize the overhead of the static allocation of buffers compared to the dynamic allocation.

7.19. Optimal Memory-aware Backpropagation of Deep Join Networks

Deep Learning training memory needs can prevent the user to consider large models and large batch sizes. In our work [4] (extended version [34]), we propose to use techniques from memory-aware scheduling and Automatic Differentiation (AD) to execute a backpropagation graph with a bounded memory requirement at the cost of extra recomputations. The case of a single homogeneous chain, i.e. the case of a network whose all stages are identical and form a chain, is well understood and optimal solutions have been proposed in the AD literature. The networks encountered in practice in the context of Deep Learning are much more diverse, both in terms of shape and heterogeneity. In this work, we define the class of backpropagation graphs, and extend those on which one can compute in polynomial time a solution that minimizes the total number of recomputations. In particular we consider join graphs which correspond to models such as Siamese or Cross Modal Networks.

7.20. Optimal checkpointing for heterogeneous chains: how to train deep neural networks with limited memory

This work [33] introduces a new activation checkpointing method which allows to significantly decrease memory usage when training Deep Neural Networks with the back-propagation algorithm. Similarly to checkpoint-ing techniques coming from the literature on Automatic Differentiation, it consists in dynamically selecting the forward activations that are saved during the training phase, and then automatically recomputing missing activations from those previously recorded. We propose an original computation model that combines two types of activation savings: either only storing the layer inputs, or recording the complete history of operations that produced the outputs (this uses more memory, but requires fewer recomputations in the backward phase), and we provide an algorithm to compute the optimal computation sequence for this model. This paper also describes a PyTorch implementation that processes the entire chain, dealing with any sequential DNN whose internal layers may be arbitrarily complex and automatically executing it according to the optimal checkpointing strategy computed given a memory limit. Through extensive experiments, we show that our implementation consistently outperforms existing checkpoint-ing approaches for a large class of networks, image sizes and batch sizes.

7.21. I/O scheduling strategy for HPC applications

With the ever-growing need of data in HPC applications, the congestion at the I/O level becomes critical in supercomputers. Architectural enhancement such as burst buffers and pre-fetching are added to machines, but are not sufficient to prevent congestion. Recent online I/O scheduling strategies have been put in place, but they add an additional congestion point and overheads in the computation of applications.

In this project, we studied application pattern (such as periodicity), in order to develop efficient scheduling strategies [7], [32] for their I/O transfers.

7.22. A New Framework for Evaluating Straggler Detection Mechanisms in MapReduce

In this work [10] we present a new framework for evaluating straggler detection mechanisms in MapReduce. We then show how to use it efficiently.

7.23. Clarification of the MPI semantics

In the framework of the MPI Forum, we have been involved in several active working groups, in particular the “Terms and Conventions” Working Group. The work carried out in this group has lead to a timely study and proposed clarifications, revisions, and enhancements to the Message Passing Interface’s (MPI’s) Semantic Terms and Conventions. To enhance MPI, a clearer understanding of the meaning of the key terminology has proven essential, and, surprisingly, important concepts remain underspecified, ambiguous and, in some cases, inconsistent and/or conflicting despite 26 years of standardization. This work [16] addresses these concerns comprehensively and usefully informs MPI developers, implementors, those teaching and learning MPI, and power users alike about key aspects of existing conventions, syntax, and semantics. This work will also be a useful driver for great clarity in current and future standardization and implementation efforts for MPI.

7.24. Adaptive Request Scheduling for the I/O Forwarding Layer using Reinforcement Learning

I/O optimization techniques such as request scheduling can improve performance mainly for the access patterns they target, or they depend on the precise tune of parameters. In this work [40], we propose an approach to adapt the I/O forwarding layer of HPC systems to the application access patterns by tuning a request scheduler. Our case study is the TWINS scheduling algorithm, where performance improvements depend on the time window parameter, which depends on the current workload. Our approach uses a

reinforcement learning technique — contextual bandits — to make the system capable of learning the best parameter value to each access pattern during its execution, without a previous training phase. We evaluate our proposal and demonstrate it can achieve a precision of 88% on the parameter selection in the first hundreds of observations of an access pattern. After having observed an access pattern for a few minutes (not necessarily contiguously), we demonstrate that the system will be able to optimize its performance for the rest of the life of the system (years).

8. Bilateral Contracts and Grants with Industry

8.1. Bilateral Grants with Industry

8.1.1. Intel

INTEL granted $30k and provided information about future many-core platforms and memory architectures to ease the design and development of the HWLOC software with early support for next generation hardware.

8.1.2. EDF

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

8.1.3. CEA

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

9. Partnerships and Cooperations

9.1. Regional Initiatives

9.1.1. CRA HPC Scalable Ecosystem, 2018-2021

2018 - 2021 (36 months)

Coordinator: Emmanuel AGULLO

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

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

9.2. National Initiatives

9.2.1. ANR

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

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

Coordinator: Laurent Simon (LaBRI)

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

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

ANR DASH Data-Aware Scheduling at Higher scale (https://project.inria.fr/dash/).

AP générique JCJC 2017, 03/2018 - 02/2022 (48 months)

Coordinator: Guillaume PALLEZ (Tadaam)

Abstract: This project focuses on the effecient execution of I/O for High-Performance applications. The idea is to take into account some knowledge on the behavior of the different I/O steps to compute efficient schedules, and to update them dynamically with the online information.

ANR Solharis SOLvers for Heterogeneous Architectures over Runtime systems, Investigating Scalability .

AAPG ANR 2019, 2019 - 2023 (48 months)

Coordinator: Alfredo BUTTARI (IRIT-INPT)

Abstract: The Solharis project aims at producing scalable methods for the solution of large sparse linear systems on large heterogeneous supercomputers, using the STARPU runtime system, and to address the scalability issues both in runtime systems and in solvers.

9.2.2. ADT - Inria Technological Development Actions

ADT Gordon

10/2018 - 09/2020 (24 months)

Coordinator: Emmanuel JEANNOT

Other partners: Storm, HiePACS, PLEIADE (Inria Bordeaux)

Abstract: Teams HiePACS, Storm and Tadaam develop each a brick of an HPC software stack, namely solver, runtime, and communication library. The goal of the Gordon project is to consolidate the HPC stack, to improve interfaces between each brick, and to target a better scalability. The bioinformatics application involved in the project has been selected so as to stress the underlying systems.

9.2.3. IPL - Inria Project Lab

High-Performance computing and BigData

Participants: Guillaume Pallez, Emmanuel Jeannot, Nicolas Vidal, Francieli Zanon-Boito

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

9.2.4. Collaboration with CERFACS

Developments on the HIPPO software

Participants: Brice Goglin, Guillaume Mercier

A Memorandum of Understanding is currently being negociated between Inria and CERFACS to organize the collaboration between both entities pertaining to the developements on the HIPPO software. The goal is to provide a portable solution to address the issue of dynamic placement of hybrid coupled MPI + OpenMP applications, especially for climate modelling. Météo France is one of the targer of this work but other teams/institutes around the globe have expressed an interest in HIPPO. Therefore we want to create a solution that would match the needs of the community on the whole.

9.3. European Initiatives

9.3.1. Collaborations with Major European Organizations

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

Subject 1: Application modeling for hierarchical memory system

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

Subject 2: I/O Scheduling

9.4. International Initiatives

9.4.1. Inria International Labs

Joint-Lab on Extreme Scale Computing (JLESC):

Coordinators: Franck Cappello (general) and Yves Robert (Inria coordinator).

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

Abstract: The purpose of the Joint Laboratory for Extreme Scale Computing (JLESC) is to be an international, virtual organization whose goal is to enhance the ability of member organizations and investigators to make the bridge between Petascale and Extreme computing. The founding partners of the JLESC are Inria and UIUC. Further members are ANL, BSC, JSC and RIKEN-AICS.

9.4.2. Inria International Partners

9.4.2.1. Informal International Partners

Partner 1: Argonne National Lab

Subject 1: Binomial Checkpointing Strategies for Machine Learning (recipient of a FACCTS grant, 2018-2020) as well as network performance prediction.

Partner 2: Vanderbilt University

Subject 2: Scheduling for Neurosciences 7.8

Partner 3: ICL at University of Tennessee

Subject 3: on instrumenting MPI applications and modeling platforms (works on HWLOC take place in the context of the Open MPI consortium) and MPI and process placement

Partner 4: Lawrence Livermore National Laboratory

Subject 4: Exposing Heterogeneous Memory Characteristics to HPC Applications 7.1

9.5. International Research Visitors

9.5.1. Visits of International Scientists

- Ana Gainaru, Reseach Assistant Professor at U. Vanderbilt, visited the team for one week in December 2019.

10. Dissemination

10.1. Promoting Scientific Activities

10.1.1. Scientific Events: Organisation

10.1.1.1. General Chair, Scientific Chair

- Brice GOGLIN and Emmanuel JEANNOT organized (with Didem UNAT from Ko c University, Turkey), PADAL 2019 (Fifth Workshop on Programming Abstractions for Data Locality) in Bordeaux (workshop by invitation): 25 participants from 10 different countries.
- François PELLEGRINI was, along with Nataliia BIELOVA (from Inria team PRIVATICS), the co-chair of this year's jury of the CNIL-Inria European prize awarded to research scientific papers on the subject of data protection and privacy.

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

- Brice GOGLIN is the *Architecture & Networks* area co-chair of SuperComputing 2020.
- Emmanuel JEANNOT is the track program chair of Cluster 2020 (area: application, algorithms, and libraries)
- Emmanuel JEANNOT was the program chair of the COLOC workshop (collocated with Euro-Par).
- Emmanuel JEANNOT was the program chair of the RADR workshop (collocated with IPDPS).
- François PELLEGRINI was a co-chair (along with Roberto DI COSMO) of the workshop on *Software and Open Science: issues and opportunities*, National days on Open science (JNSO 2019), Paris (https://jnso2019.sciencesconf.org/resource/page/id/2).

10.1.2.2. Member of Conference Program Committees

- Alexandre DENIS was a member of the program committee of CCGrid 2019.
- Brice GOGLIN was a member of the program committee of ICPP 2019, EuroMPI 2019, HotInterconnects 26, ROME 2019, ROSS 2019, RADR 2019.
- Emmanuel JEANNOT was member of the program committee of SuperComputing 2019, Euro-MPI 2019, ROSS 2019, Heteropar 2019.
- Guillaume MERCIER was a member of the programm committee of CCGrid 2019 and EuroMPI 2019.
- Guillaume PALLEZ was a member of the program committee of SC 2019 (Tutorials), ICPP 2019, IPDPS 2020, ICA3PP 2019, PMBS 2019.
- François PELLEGRINI was a member of the program committee of ENISA's "EU Privacy Forum".

10.1.3. Journal

10.1.3.1. Member of the Editorial Boards

- Emmanuel JEANNOT is associate editor of the International Journal of Parallel, Emergent & Distributed Systems (IJPEDS).

10.1.3.2. Reviewer - Reviewing Activities

- Emmanuel JEANNOT was a reviewer for JPDC, Parallel Computing, Transaction on Computers.
- Guillaume MERCIER was a reviewer for IEEE Transactions on Computers and for Cluster Computing.
- François PELLEGRINI was a reviewer for journal Terminal.

10.1.4. Invited Talks

- Emmanuel JEANNOT was invited to the panel *Heterogeneous Computing for Energy Efficiency* of 10th International Green and Sustainable Computing Conference (Alexandia VA, USA, octobre 2019, https://www.igscc.org/copy-of-schedule).

- Emmanuel JEANNOT was invited to the panel *Resilience in High Performance Computing* to the HiPEAC event (at Bilbao Spain, October 2019, https://www.hipeac.net/csw/2019/bilbao/#/schedule/).
- Guillaume MERCIER was invited to make a presentation at the PADAL workshop 2019.
- Guillaume PALLEZ was invited to give a talk at the Royal Society of London, UK, as a part of the event *Numerical algorithms for high-performance computational science* [11] (https://royalsociety.org/science-events-and-lectures/2019/04/high-performance-computing/).
- François PELLEGRINI was invited to deliver a common talk with Emmanuel NETTER, "*Is code law?*", at the Symposium of the Agorantic research federation, University of Avignon (https://agorantic.univ-avignon.fr/symposium/symposium-agorantic-28-janvier-2019/).

10.1.5. Scientific Expertise

- Emmanuel JEANNOT was a member of the hiring committee of an Inria junior researcher position at Nancy and at the National Level.
- Guillaume PALLEZ is an elected member of the Inria evaluation committee.
- François PELLEGRINI is co-chair (along with Roberto DI COSMO) of the workgroup on Free software of the Permanent Secretariat for Open Science (SPSO) of the French Ministry of Higher Education (MENESR).
- François PELLEGRINI was heard in an expert panel commissioned by the *Information mission on digital identity* of the French *Assemblée nationale*, chaired by Mrs Marietta KARAMANLI, assisted by Mrs Paula FORTEZA and Christine HENNION.

10.1.6. Research Administration

- Alexandre DENIS is head of the Inria Bordeaux CUMI-R (IT users committee).
- Brice GOGLIN and Guillaume MERCIER are elected members of the Inria Bordeaux center committee.
- Emmanuel JEANNOT is deputy head of science of the Inria Bordeaux research center.
- Emmanuel JEANNOT is member of the Inria evaluation committee
- Emmanuel JEANNOT is member of LaBRI scientific council and head of the Satanas team.
- Guillaume PALLEZ is a worker representative at the Prevention, Health, Security committee (CHSCT) for the Inria center of Bordeaux.

10.1.7. Standardization Activities

TADAAM attended the MPI Forum meetings on behalf of Inria (where the MPI standard for communication in parallel applications is developed and maintained). Guillaume MERCIER leads the *Hardware Topologies* working group. Part of the HSPLIT proposal was discussed and read at the last physical meeting in December 2019 in Albuquerque and has been approved to enter the voting process for an eventual inclusion in the next revision (4.0) of the MPI standard. This voting process will take place in the first semester of 2020 and the release of the 4.0 revision is expected for the end of 2020. Guillaume MERCIER is also the chair of the standard chapter committee *Groups, Contexts, Communicators, Caching* and member of several other chapter committees.

10.2. Teaching - Supervision - Juries

10.2.1. Teaching

Members of the TADAAM project gave hundreds of hours of teaching at Université de Bordeaux and the Bordeaux INP engineering school, covering a wide range of topics from basic use of computers, introduction to algorithmics and C programming to advanced topics such as probabilities and statistics, scheduling, computer architecture, operating systems, parallel programming and high-performance runtime systems, as well as software law and personal data.

- Brice GOGLIN gave courses about Operating Systems to teachers as part of the *Diplôme Inter Universitaire* to prepare them for teaching the new Computer Science track in high-school.
- François PELLEGRINI did a training session on "*Information science, digital technologies and law*" for the continuous education of magistrates, École nationale de la magistrature (National School for Magistrates), Paris.
- François PELLEGRINI did two training sessions on "*Strategic issues of information technologies*" and "*Personal data law*" to a group of administration heads and civil society activists of several French-speaking west-African countries, in the context of FFGI 2019 at Ouagadougou, Burkina Faso.

10.2.2. Supervision

PhD: Benjamin LORENDEAU, Amélioration des performances via un parallélisme multi-niveaux sur un code CFD en maillages non structurés. Defense at Université de Bordeaux on December 16th. Advisors : Yvan FOURNIER and Emmanuel JEANNOT.

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

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

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

PhD started: Philippe SWARTVAGHER, Interactions at large scale between high performance communication libraries and task-based runtime, started in October 2019. Advisors: Alexandre DENIS and Emmanuel JEANNOT.

PhD started: Florian REYNIER, Task-based communication progression, started in January 2019. Advisors: Alexandre DENIS and Emmanuel JEANNOT.

PhD started: Pierre FERENBACH, The legal regime of video games, started in January 2019. Advisors: Xavier DAVERAT and François PELLEGRINI.

Master: Léa CHEVALIER, M2 student at Université Paris Nanterre supervised by François PELLEGRINI, won the Disney–Microsoft–Orange–TF1 prize on Media Law for her master thesis on "*Artistic creations generated by automated processing: are they works like others?*".

10.2.3. Juries

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

- Hugo BRUNIE, U. Bordeaux (Member);
- Jean-Baptiste KECK, U. Grenoble Alpes (Reviewer);
- Hamza DEROUI, Insa Rennes and U. Rennes (Reviewer);
- Arthur LOUSSERT, U. Bordeaux (Member).

FrançoisPELLEGRINI was member of the Ph.D defense jury of:

- Maximilien LANNA, U. Paris II Panthéon Assas (Member).

10.3. Popularization

10.3.1. Internal or external Inria responsibilities

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

10.3.2. Articles and contents

- Guillaume PALLEZ wrote a blog article for *Binaire* on autonomous vehicles [37].
- François PELLEGRINI was interviewed on the subject of "*Resisting algorithmic governance*" (cover page), Expertises droit/technologies/prospectives, nr 443, Feb. 2019 (https://www.expertises.info/#anciens-numeros).

10.3.3. Interventions

- Brice GOGLIN is the sponsor (*parrain*) of the *Edouard Vaillant* middle school (Bordeaux) for their scientific projects with the fondation *La main à la pâte*.
- Guillaume PALLEZ, Brice GOGLIN, Valentin HONORÉ, Philippe SWARTVAGHER and Nicolas VIDAL gave seminars and hands-on session about computer science to schools attending *Fete de la Science*, Oct. 2019.
- François PELLEGRINI participated in the round table "*GDPR and cyber-security*" during the OpenS'IAE event, Pau.
- François PELLEGRINI participated in the rountable on "*Legal aspects, GDPR and anonymization technologies*" during the annual congress of *Société informatique de France* (SIF), Bordeaux.
- François PELLEGRINI gave a conference on the security of personal data during the *Mars@Hack* event, in Mont-de-Marsan.
- François PELLEGRINI participated in the round table "*Is our legal framework IA-compatible?*" during the NAIA (*Nouvelle-Aquitaine Intelligence Artificielle*) event, Bordeaux.
- François PELLEGRINI participated in the round table "*What alternatives and what regulations in the GAFAM era?*" during the second edition of the *Rencontres Culture:Tech*, Assemblée nationale, Paris.
- François PELLEGRINI gave a conference on "*Freedom in the digital age*" to students of first and second year of all departments at ENS Cachan (250 people).
- François PELLEGRINI participated in a round table on "*The future of choice*" in the context of the "de-inauguration" of the exhibition textitUnder influence, the science of choice at Espace Pierre-Gilles de Gennes (ESPGG), Paris.
- François PELLEGRINI answered the public at the movie theater Utopia Bordeaux after the display of the movie *Meeting Snowden*.
- François PELLEGRINI gave a public conference on "*How does personal data processing interfere with our privacy?*" during the *IA Pau* conference, Pau.

10.3.4. Internal action

- François PELLEGRINI participated in a round table on "*Digital security: technical, ethical and legal aspects*", Inria scientific days 2019, Lyon (https://project.inria.fr/journeesscientifiques2019/).

10.3.5. Creation of media or tools for science outreach

- Brice GOGLIN was involved in the building of the MOOC *Sciences Numériques et Technologie* which focus at bringing basics about computer science to high-school teachers and general audience. More than 18 000 people registered to the course.
- François PELLEGRINI is one of the 14 people appearing in the documentary "*LOL: Logiciel libre, une affaire sérieuse*" ("LOL: Free software, a serious matter").

11. Bibliography

Publications of the year

Doctoral Dissertations and Habilitation Theses

[1] G. MERCIER. *Évolutions du passage de messages face aux défis de la gestion des topologies matérielles hiérarchiques*, Université de Bordeaux, December 2019, Habilitation à diriger des recherches, https://hal.inria.fr/tel-02412813

Articles in International Peer-Reviewed Journals

[2] G. AUPY, A. BENOIT, B. GOGLIN, L. POTTIER, Y. ROBERT. *Co-scheduling HPC workloads on cache-partitioned CMP platforms*, in "International Journal of High Performance Computing Applications", April 2019, vol. 33, n° 6, pp. 1221-1239 [*DOI :* 10.1177/1094342019846956], https://hal.inria.fr/hal-02093172

[3] G. AUPY, B. GOGLIN, V. HONORÉ, B. RAFFIN. *Modeling High-throughput Applications for in situ Analytics*, in "International Journal of High Performance Computing Applications", April 2019, vol. 33, n° 6, pp. 1185-1200, forthcoming [*DOI :* 10.1177/1094342019847263], https://hal.inria.fr/hal-02091340

[4] O. BEAUMONT, J. HERRMANN, G. PALLEZ, A. SHILOVA. *Optimal Memory-aware Backpropagation of Deep Join Networks*, in "Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences", 2019, forthcoming, https://hal.inria.fr/hal-02401105

[5] A. DENIS, J. JAEGER, E. JEANNOT, M. PÉRACHE, H. TABOADA. *Study on progress threads placement and dedicated cores for overlapping MPI nonblocking collectives on manycore processor*, in "International Journal of High Performance Computing Applications", May 2019, vol. 33, n° 6, pp. 1240-1254 [*DOI :* 10.1177/1094342019860184], https://hal.inria.fr/hal-02400422

[6] N. DENOYELLE, B. GOGLIN, A. ILIC, E. JEANNOT, L. SOUSA. *Modeling Non-Uniform Memory Access on Large Compute Nodes with the Cache-Aware Roofline Model*, in "IEEE Transactions on Parallel and Distributed Systems", June 2019, vol. 30, n° 6, pp. 1374–1389 [*DOI :* 10.1109/TPDS.2018.2883056], https://hal.inria.fr/hal-01924951

[7] A. GAINARU, V. LE FÈVRE, G. PALLEZ. *I/O scheduling strategy for periodic applications*, in "ACM Transactions on Parallel Computing", 2019, forthcoming, https://hal.inria.fr/hal-02141576

[8] A. GAINARU, H. SUN, G. AUPY, Y. HUO, B. A. LANDMAN, P. RAGHAVAN. *On-the-fly scheduling vs. reservation-based scheduling for unpredictable workflows*, in "International Journal of High Performance Computing Applications", 2019, forthcoming [*DOI :* 10.1177/1094342019841681], https://hal.inria.fr/hal-02058290

[9] F. PELLEGRINI. *Safety and digital hygiene of professionals*, in "Dalloz IP/IT", April 2019, n° 4, pp. 233-236, https://hal.inria.fr/hal-02113563

[10] T.-D. PHAN, G. PALLEZ, S. IBRAHIM, P. RAGHAVAN. *A New Framework for Evaluating Straggler Detection Mechanisms in MapReduce*, in "ACM Transactions on Modeling and Performance Evaluation of Computing Systems", April 2019, vol. X, pp. 1-22 [*DOI :* 10.1145/3328740], https://hal.inria.fr/hal-02172590

Invited Conferences

[11] G. PALLEZ. *Adjoint computation and Backpropagation*, in "Meeting of the Royal Society – Numerical algorithms for high-performance computational science", London, United Kingdom, April 2019, https://hal.inria.fr/hal-02400746

[12] F. PELLEGRINI. *Enjeux démocratiques de la protection des données à caractère personnel*, in "Journées scientifiques Inria", Lyon, France, Inria, June 2019, https://hal.inria.fr/hal-02150857

International Conferences with Proceedings

[13] M. A. ABA, A. MUNIER-KORDON, G. PALLEZ. *Scheduling on Two Unbounded Resources with Communication Costs*, in "Euro-Par - European Conference on Parallel Processing", Gottingen, Germany, August 2019, https://hal.inria.fr/hal-02141622

[14] G. AUPY, O. BEAUMONT, L. EYRAUD-DUBOIS. *Sizing and Partitioning Strategies for Burst-Buffers to Reduce IO Contention*, in "IPDPS 2019 - 33rd IEEE International Parallel and Distributed Processing Symposium", Rio de Janeiro, Brazil, May 2019, https://hal.inria.fr/hal-02141616

[15] G. AUPY, A. GAINARU, V. HONORÉ, P. RAGHAVAN, Y. ROBERT, H. SUN. *Reservation Strategies for Stochastic Jobs*, in "IPDPS 2019 - 33rd IEEE International Parallel and Distributed Processing Symposium", Rio de Janeiro, Brazil, IEEE, May 2019, pp. 166-175 [*DOI :* 10.1109/IPDPS.2019.00027], https://hal.inria.fr/hal-01968419

[16] P. BANGALORE, R. RABENSEIFNER, D. HOLMES, J. JAEGER, G. MERCIER, C. BLAAS-SCHENNER, A. SKJELLUM. *Exposition, clarification, and expansion of MPI semantic terms and conventions*, in "EuroMPI '19 - 26th European MPI Users' Group Meeting", Zürich, Switzerland, ACM Press, September 2019, pp. 1-10 [*DOI :* 10.1145/3343211.3343213], https://hal.inria.fr/hal-02413199

[17] A. DENIS. *Scalability of the NewMadeleine Communication Library for Large Numbers of MPI Point-to-Point Requests*, in "CCGrid 2019 - 19th Annual IEEE/ACM International Symposium in Cluster, Cloud, and Grid Computing", Larnaca, Cyprus, May 2019, https://hal.inria.fr/hal-02103700

[18] N. DENOYELLE, B. GOGLIN, E. JEANNOT, T. ROPARS. *Data and Thread Placement in NUMA Architectures: A Statistical Learning Approach*, in "ICPP 2019 - 48th International Conference on Parallel Processing", Kyoto, Japan, ACM Press, August 2019, pp. 1-10 [*DOI :* 10.1145/3337821.3337893], https://hal.inria.fr/hal-02135545

[19] A. GAINARU, G. PALLEZ. *Making Speculative Scheduling Robust to Incomplete Data*, in "ScalA19: 10th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems", Denver, United States, November 2019, https://hal.inria.fr/hal-02336582

[20] A. GAINARU, G. PALLEZ, H. SUN, P. RAGHAVAN. *Speculative Scheduling for Stochastic HPC Applications*, in "ICPP 2019 - 48th International Conference on Parallel Processing", Kyoto, Japan, August 2019, https://hal.inria.fr/hal-02158598

[21] B. GOGLIN, A. RUBIO PROAÑO. *Opportunities for Partitioning Non-Volatile Memory DIMMs between Co-scheduled Jobs on HPC Nodes*, in "Euro-Par 2019: Parallel Processing Workshops", Göttingen, Germany, August 2019, https://hal.inria.fr/hal-02173336

[22] E. A. LEÓN, B. GOGLIN, A. RUBIO PROAÑO. *M&MMs: Navigating Complex Memory Spaces with hwloc*, in "Fifth International Symposium on Memory Systems Proceedings (MEMSYS19)", Washington, DC, United States, May 2019 [*DOI :* 10.1145/3357526.3357546], https://hal.inria.fr/hal-02266285

[23] S.-M. TSENG, B. NICOLAE, G. BOSILCA, E. JEANNOT, A. CHANDRAMOWLISHWARAN, F. CAPPELLO. *Towards Portable Online Prediction of Network Utilization using MPI-level Monitoring*, in "EuroPar'19: 25th International European Conference on Parallel and Distributed Systems", Goettingen, Germany, August 2019, https://hal.inria.fr/hal-02184204

Conferences without Proceedings

[24] A. HORI, G. BOSILCA, E. JEANNOT, T. OGURA, Y. ISHIKAWA. *Is Japanese HPC another Galapagos? - Interim Report of MPI International Survey* -, in "Summer United Workshops on Parallel, Distributed and Cooperative Processing", Kitami, Japan, July 2019, https://hal.inria.fr/hal-02193264

Scientific Books (or Scientific Book chapters)

[25] *Euro-Par 2018: Parallel Processing Workshop*, LNCS - Lecture Notes in Computer Science, Springer, Turin, Italy, May 2019, vol. 11339 [*DOI :* 10.1007/978-3-030-10549-5], https://hal.inria.fr/hal-02403078

[26] *Actes du colloque des Convergences du Droit et du Numérique*, Actes du colloque des Convergences du droit et du numérique, Université de Bordeaux, Bordeaux, France, July 2019, https://hal.inria.fr/hal-02195921

[27] J. CARRETERO, E. JEANNOT, A. ZOMAYA. *Ultrascale Computing Systems*, Institution of Engineering and Technology, January 2019 [*DOI :* 10.1049/PBPC024E], https://hal.inria.fr/hal-02402981

[28] G. D. COSTA, A. L. LASTOVETSKY, J. G. BARBOSA, J. C. D. MARTIN, J.-L. G. ZAPATA, M. JANETSCHEK, E. JEANNOT, J. LEITÃO, R. R. MANUMACHU, R. PRODAN, J. A. RICO-GALLEGO, P. V. ROY, A. SHOKER, A. V. D. LINDE. *Programming models and runtimes*, in "Ultrascale Computing Systems", Institution of Engineering and Technology, 2019 [*DOI :* 10.1049/PBPC024E_CH2], https://hal.inria.fr/hal-02403121

[29] E. JEANNOT, J. CARRETERO. *Conclusion*, in "Ultrascale Computing Systems", Institution of Engineering and Technology, 2019 [*DOI :* 10.1049/PBPC024E], https://hal.inria.fr/hal-02403088

[30] F. PELLEGRINI. *Security and digitization : Between fantasies of effectiveness and proven violations of fundamental rights*, in "La sécurité : mutations et incertitudes", M. AFROUKH, C. MAUBERNARD, C. VAL (editors), Collection Colloques & Essais, Institut universitaire Varenne, January 2019, n° 77, pp. 89-100, https://hal.inria.fr/hal-02069419

Research Reports

[31] M. AIT ABA, G. AUPY, A. MUNIER-KORDON. *Scheduling on Two Unbounded Resources with Communication Costs*, Inria, March 2019, n° RR-9264, https://hal.inria.fr/hal-02076473

[32] G. AUPY, E. JEANNOT, N. VIDAL. *Scheduling periodic I/O access with bi-colored chains: models and algorithms*, Inria, February 2019, n° RR-9255, 25 p. , https://hal.inria.fr/hal-02021070

[33] O. BEAUMONT, L. EYRAUD-DUBOIS, J. HERRMANN, A. JOLY, A. SHILOVA. *Optimal checkpointing for heterogeneous chains: how to train deep neural networks with limited memory*, Inria Bordeaux Sud-Ouest, November 2019, n° RR-9302, https://arxiv.org/abs/1911.13214 , https://hal.inria.fr/hal-02352969

[34] O. BEAUMONT, J. HERRMANN, G. PALLEZ, A. SHILOVA. *Optimal Memory-aware Backpropagation of Deep Join Networks*, Inria, May 2019, n° RR-9273, https://hal.inria.fr/hal-02131552

[35] A. GAINARU, B. GOGLIN, V. HONORÉ, G. PALLEZ, P. RAGHAVAN, Y. ROBERT, H. SUN. *Reservation and Checkpointing Strategies for Stochastic Jobs (Extended Version)*, Inria & Labri, Univ. Bordeaux ; Department

of EECS, Vanderbilt University, Nashville, TN, USA ; Laboratoire LIP, ENS Lyon & University of Tennessee Knoxville, Lyon, France, October 2019, n° RR-9294, https://hal.inria.fr/hal-02328013

[36] E. JEANNOT, R. SARTORI. *Improving MPI Application Communication Time with an Introspection Monitoring Library*, Inria, October 2019, n° RR-9292, 23 p. , https://hal.inria.fr/hal-02304515

Scientific Popularization

[37] G. PALLEZ. *Le non-sens écologique des voitures autonomes*, July 2019, Dans cet article de vulgarisation, je discute si l'avénement promis des véhicules autonomes serait ou non réelement un moyen de réduire la pollution (notamment dans les villes) (plutôt pas), https://hal.inria.fr/hal-02342636

Other Publications

[38] G. AUPY, J. HERRMANN. *H-Revolve: A Framework for Adjoint Computation on Synchrone Hierarchical Platforms*, March 2019, working paper or preprint, https://hal.inria.fr/hal-02080706

[39] P. BECKMAN, E. JEANNOT, S. PERARNAU. *Introduction to RADR 2019*, IEEE, May 2019, pp. 908-910, IPDPSW 2019 - IEEE International Parallel and Distributed Processing Symposium Workshops [*DOI :* 10.1109/IPDPSW.2019.00150], https://hal.inria.fr/hal-02403058

[40] J. LUCA BEZ, F. ZANON BOITO, R. NOU, A. MIRANDA, T. CORTES, P. O. NAVAUX. *Adaptive Request Scheduling for the I/O Forwarding Layer using Reinforcement Learning*, October 2019, working paper or preprint, https://hal.inria.fr/hal-01994677

[41] F. REYNIER. *Task based progression of asynchronous communications*, June 2019, COMPAS 2019 - Conférence d'informatique en Parallélisme, Architecture et Système, Poster, https://hal.inria.fr/hal-02407276

[42] P. SWARTVAGHER. *Opérations collectives dynamiques dans StarPU / NewMadeleine*, ENSEIRB-MATMECA, September 2019, https://hal.inria.fr/hal-02303822

References in notes

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