Activity Report 2014

Team POSTALE

Performance Optimization by Software Transformation and Algorithms & Librairies Enhancement

RESEARCH CENTER
Saclay - Île-de-France

THEME
Architecture, Languages and Compilation

Table of contents

Team POSTALE

Keywords: High Performance Computing, Computer Architectures, Generative Programming, Program Optimization, Numerical Algorithms, Auto-tuning, Numerical Software

Creation of the Team: 2014 January 01.

1. Members

Research Scientists

Christine Eisenbeis [Inria, Senior Researcher]
Grigori Fursin [Inria, Researcher, until Oct 2014]

Faculty Members

Marc Baboulin [Team leader, Univ. Paris XI, Professor]
Daniel Etiemble [Univ. Paris XI, Professor]
Joël Falcou [Univ. Paris XI, Associate Professor, HdR]
Amal Khabou [Univ. Paris XI, Associate Professor]
Lionel Lacassagne [Univ. Paris XI, Associate Professor, HdR]

Engineers

Konstantin Petrov [Inria, Research Engineer, part-time]
Taj Muhammad Khan [Inria, Univ. Paris XI, until Aug 2014]

PhD Students

Lénaïc Bagnères [Inria]
Laurent Cabaret [École Centrale de Paris, Prag]
Alessandro Ferreira Leite [Inria, Univ. Paris XI]
Aygül Jamal [Univ. Paris XI]
Michael Kruse [Inria, Univ. Paris XI, until September 2014]
Jason Lambert [CEA List, Univ. Paris XI]
Ian Masliah [Univ. Paris XI]
Adrien Remy de Zotti [Univ. Paris XI, ATER]
Antoine Tran Tan [Univ. Paris XI]

Administrative Assistant

Katia Evrat [Inria]

2. Overall Objectives

2.1. Overall Objectives

Postale is an Inria Saclay Île-de-France team in the area of high-performance computing (HPC), parallel architectures and compilation. The Postale acronym stands for "Performance Optimization by Software Transformation and Algorithms & Libraries Enhancement". Postale focuses on providing software and hardware means to help programmers to deal with the ever growing complexity of programming state-of-the-art parallel and distributed architectures and to develop optimized HPC applications. The Postale team involves researchers from Laboratoire de Recherche en Informatique (LRI) - University Paris-Sud - and have expertise in various domains including algorithms for HPC, programming languages, compilers, and architectures. The project is structured around two main research issues:

- Develop methods and software for program transformations/optimizations for a given algorithm/application and take advantage of programmer knowledge to develop efficient codes through programmer/compiler interface and domain specific languages (DSL),
- Provide innovative algorithms and efficient implementations in high-performance computing libraries for current highly parallel and heterogeneous or embedded architectures, and explore current barriers to performance.

Following the Inria terminology, the Postale team belongs to the field “Algorithmics, Programming, Software and Architecture” in the category ‘Architecture and Compiling”. The specificity of this project among other Inria teams addressing similar topics is that it does not focus only on architecture caracteristics and low level aspects of program execution but it takes into account the dimension of the user or program developer and their specific domain of application. In particular it aims at developing paths between programmers at the application level and computing resources. The targeted applications are high-performance scientific or image processing applications that require efficient use of ever developing highly parallel and heterogeneous systems. Since the applications are at the heart of our research, the members of the Postale team share the common goal of providing users with the most adequate compiler/user interface and software for their scientific application. In this project, we address issues which are transverse to most research objectives but with a different point of view, depending on if we work at the compiler or at the algorithm level. Namely, these issues are related to minimizing energy consumption and the amount of communication or synchronizations, optimizing performance and data locality, proposing user interfaces as close as possible to application domains.

3. Research Program

3.1. Architectures and program optimization

In this research topic, we focus on optimizing resources in a systematic way for the programmer by addressing fundamental issues like optimizing communication and data layout, generating automatically optimized codes via Domain Specific Languages (DSL), and auto-tuning of computer systems.

3.1.1. Optimization techniques for data and energy

3.1.1.1. Scientific context

Among the main challenges encountered in the race towards performance for supercomputers are energy (consumption, power and heat dissipation) and the memory/communication wall. This research topic addresses more specialized code analysis and optimization techniques as well as algorithmic changes in order to meet these two criteria, both from an expert - meaning handmade code transformations - or automatic - meaning compile time or run time - point of view.

Memory/communication wall means that processor elementary clock cycle decreases more rapidly over years than data transfer whether vertically between memory-ies and CPU (memory access) or horizontally between processors (data transfer). Moreover current architectures include complex memory features such as deep memory hierarchies, shared caches between cores, data alignment constraints, distributed memories etc. As a result data communication and data layout are becoming the bottleneck to performance and most program transformations aim at organizing them carefully and possibly avoiding or minimizing them. Energy consumption is also a limitation for today’s processor performance. Then the options are either to design processors that consume less energy or, at the software level, to design energy-saving compilers and algorithms.

In general, the memory and energy walls are tackled with the same kind of program transformations that consist of avoiding as much as possible data communication [158] but considering these issues separately offers a different perspective. In this research axis, we focus on data/memory and energy/power optimization that include handmade or automatic compiler, code and algorithm optimizations. The resulting tools are expected to be integrated in other Postale topics related to auto-tuning [93], code generation [83] or communication-avoiding algorithms [51], [112].

3.1.1.2. Activity description and recent achievements

3.1.1.2.1. Optimization for data:

Program data transformation - data layout, data transfers. Postale has been addressing these issues in the past ANR PetaQCD project described in [63], [64] and in the PhD thesis of Michael Kruse [113]. The latter describes handmade data layout optimizations for optimizing a 4D stencil computation taking into account the BlueGene Q features. It also presents the Molly software based on the LLVM (Low Level Virtual Machine) Polly optimizing compiler that automatically generates code for MPI data transfers (see Figure 1 that shows an example of code generating a decomposition of a stencil computation into 4 subdomains and how data are exchanged between subdomains).

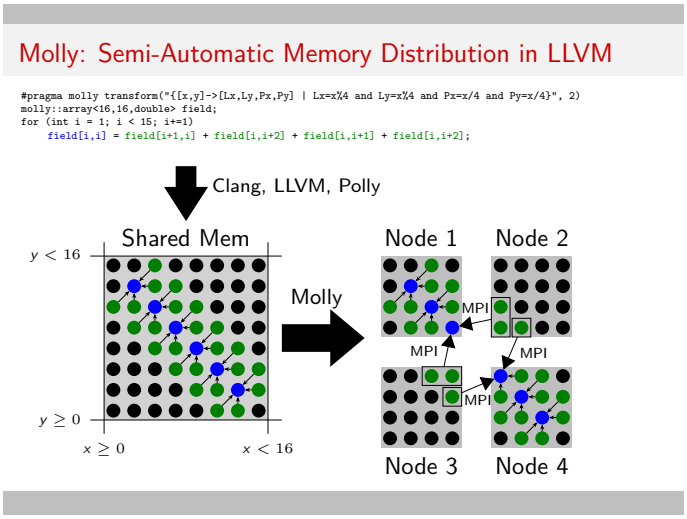

Figure 1. Automatic generation of subdomains using the Molly software.

Data layout is still a critical point that Postale will address. The DSL [83] approach allows us to consider data layout globally, providing then an opportunity to study aggressive layouts without transformation penalty. We will also seize this opportunity to investigate the data layout problem as a new dimension of the CollectiveMind [93] optimization topic.

Algorithm transformation - automating communication avoiding algorithms. This part is related to the Postale work on numerical algorithms. It originates from a research grant application elaborated with the former PetaQCD [64] team and the Inria Alpine project-team. One essential research direction consists of providing a set of high level optimizations that are generally out of reach from a traditional compiler approach. Among these optimizations, we consider communication-avoiding transformations and address the current open question of integrating these transformations in the polyhedral model in order to make them available in most software environments. Communication-avoiding algorithms improve parallelism and decrease communication requirements by ignoring some of dependency constraints at the frontiers of subdomains. Integrating communication-avoiding transformations is challenging first because these transformations change code semantics, which is unsual in program transformations, second because the validity of these transformations relies on numerical properties of the underlying transformed algorithms. This requires both compiler and algorithm skills since these transformations have important impact on the numerical stability and convergence of algorithms. Tools for the automatic generation of these transformed algorithms have two kinds of application. First, they accelerate the fastidious task of reprogramming for testing numerical properties. They may even be incorporated in an iterative tool for systematically evaluating these properties. Second, if these transformations are formalized we can consider generating different versions on line at run time, to adapt automatically algorithms to run time values [65]. In particular we plan to address s-steps algorithms [133] in iterative methods as these program trans-

formations are similar to loop unrolling and ghosting (inverse of loop peeling). These are aggressive transformations and special preconditioning is needed in order to ensure convergence.

3.1.1.2.2. Optimizing energy:

In this topic there are two main research directions. The first one is about reversible computing based on the Landauer's conjecture that heat dissipation is produced by information erasing. The second one is on actual measurements of energy/power of program execution and on understanding which application features are the most likely to save or consume energy.

Regarding **reversible computing**, the Landauer's hypothesis - still in discussion among physicists - says that erasing one bit of information dissipates energy, independently from hardware. This implies that energy saving algorithms should avoid as much as possible erasing information: it should be possible to recover values of variables at any time in program execution. In a previous work we have analyzed the impact of making computing DAG (Directed Acyclic Graphs) reversible [61]. We have also used reversible computing in register allocation by enabling value rematerialization also by reverse computing [62]. We are now working on characterizing algorithms by the amount of input and output data that have to be added to make algorithms reversible. We also plan to analyze mixed precision numerical algorithms [50] from this perspective.

Another research direction concerns **energy and power profiling and optimizing**. Understanding and monitoring precise energetic behavior of current programs is still a not easy task for the programmer or the compiler. One can measure it with wattmeters, or perform processor simulations or use hardware counters or sensors, or approximate it by the number of data that are communicated [159]. Especially on supercomputers or cloud framework it might be impossible to get this information. Besides making experiments on energy and power profiling [128], this research axis also includes the analysis of programming features that are the key parameters for saving energy. The ultimate goal is to have a cost model that describes the program energetic behavior of programs for the programmer or compiler being able to control it. One obvious key parameter is the count of memory accesses but one can also think of regularity features such as constant strides memory access, whether the code is statically or dynamically controlled, regularity/predictability conditional branchs. We have already performed this kind of analysis in the context of value prediction techniques where we designed entropy based criteria for estimating the predictibility of the sequence of values of some variables [129].

3.1.1.3. Research tracks for the 4 next years

Short term objectives are related to handmade or semi-automatic profiling and optimization of current scientific or image processing challenging applications. This gives a very good insight and expertise over state of the art applications and architectures. This know-how can be exploited under the form of libraries. This includes performance profiling, analysis of the energetic behavior of applications, and finding hot spots and focus optimization on these parts. This also implies to implement new numerical algorithms such as the communication-avoiding algorithms. Mid term objectives are to go forward to the automatization or semi-automatization of these techniques. Long term objectives are to understand the precise relationship between physics and computation both in programs as in reversible computing and in algorithms like in algorithmic thermodynamics [60]. The path is to define a notion of energetic complexity, which we intend to do it with the Galac team at Laboratoire de Recherche en Informatique.

3.1.2. Generative programming for new parallel architectures

3.1.2.1. Scientific context

Design, development and maintenance of high-performance scientific code is becoming one of the main issue of scientific computing. As hardware is becoming more complex and programming tools and models are proposed to satisfy constantly evolving applications, gathering expertise in both any scientific field and parallel programming is a daunting task. The natural conclusion is then to provide software design tools such that non-experts in computer science are able to produce non-trivial yet efficient codes on modern hardware architectures at their disposal. These tools can be divided in two types:

- **Compilers**. Compilers can be designed to either automatically derive parallel version of sequential codes or to support specific annotations to do so. Various successful examples include ISPC [137], SPADE [167] or GCC and its support for polyhedral compilation [140]. By offloading these tasks to compilers, the performance of the resulting codes is free of any overhead and the amount of user input is minimized. However, the scope and applicability of these techniques are fragile and can be hindered by complex code flow, inadequate data types or the use of high level languages features.
- **Libraries**. The unability of compilers to handle complex semantic is often mitigated by the design of libraries. Libraries can expose an arbitrary high level of abstraction through abstract data types and functions operating on them. User code is then expressed as a combination of function calls over instances of these data types. Different level of abstraction for parallel systems are available ranging from linear algebra [42], [109], image processing [70] to graph algorithms [153]. The main limitation of this approach is the lack of inter-procedural optimizations and the inherent divergence in API among vendors and targeted systems.

One emerging solution is to combine aspects of both solutions by designing systems which are able to provide abstraction and performance. One such approach is the design and development of **Domain Specific Languages** (or DSL) and more precisely, **Domain Specific Embedded Languages** (DSEL). DSLs [154] are non-general purpose, declarative language that simplify development by allowing users to express "the problem to solve" instead of "how to solve it". Actual code generation is then left to a proper compiler, interpreter or code generator that use high-level abstraction analysis and potential knowledge about target hardware to ensure performance. SCALA – and more precisely the FORGE tool [156] – is one of the most successful attempt at applying such techniques to parallel programming. DSELs differ from regular DSLs in the fact that they exist as a subset of an existing general purpose language. Often implemented as **Active Libraries** [166], they perform high-level optimizations based on a semantic analysis of the code before any real compilation process.

3.1.2.2. Activity description and recent achievements

In this research, we investigate the impact and applicability of software design methods based on DSELs to parallel programming and we study the portability and forward scalability of such programs. To do so, we investigate **Generative Programming** [76] applied to parallel programming.
Generative Programming is based on the hypothesis that any complex software system can be split into a list of interchangeable components (with clearly identified tasks) and a series of generators that combine components by following rules derived from an a priori domain specific analysis. In particular, we want to show that integrating the architectural support as another generative component of the set of tools leads to a better performance and an easier development on embedded or custom architecture targets (see Figure 2).

The application of Generative Programming allows us to build active libraries that can be easily re-targeted, optimized and deployed on a large selection of hardware systems. This is done by decoupling the abstract description of the DSEL from the description of hardware systems and the generation of hardware agnostic software components.
Current applications of this methodology include:

- BOOST.SIMD [84] is a C++ library for portable SIMD computations. It uses architecture aware generative programming to generate zero-overhead SIMD code on a large selection of platforms (from SSE to AVX2, Xeon Phi, PowerPC and ARM). Its interface is made so it is totally integrated into modern C++ design strategy based on the use of generic code and calls to the standard template libraries. In most cases, BOOST.SIMD delivers performance on the par with hand written SIMD code or with autovectorizers.
- NT^2 [83], [89] is a C++ library which implements a DSEL similar to MATLAB while providing automatic parallelization on SIMD systems, multicores and GPGPUs. NT^2 uses the high level of abstraction brought by the MATLAB API to detect, analyze and generate efficient loop nests taking care of every level of parallel hardware available. NT^2 eases the design of scientific computing application prototypes while delivering a significant percentage of the peak performance.

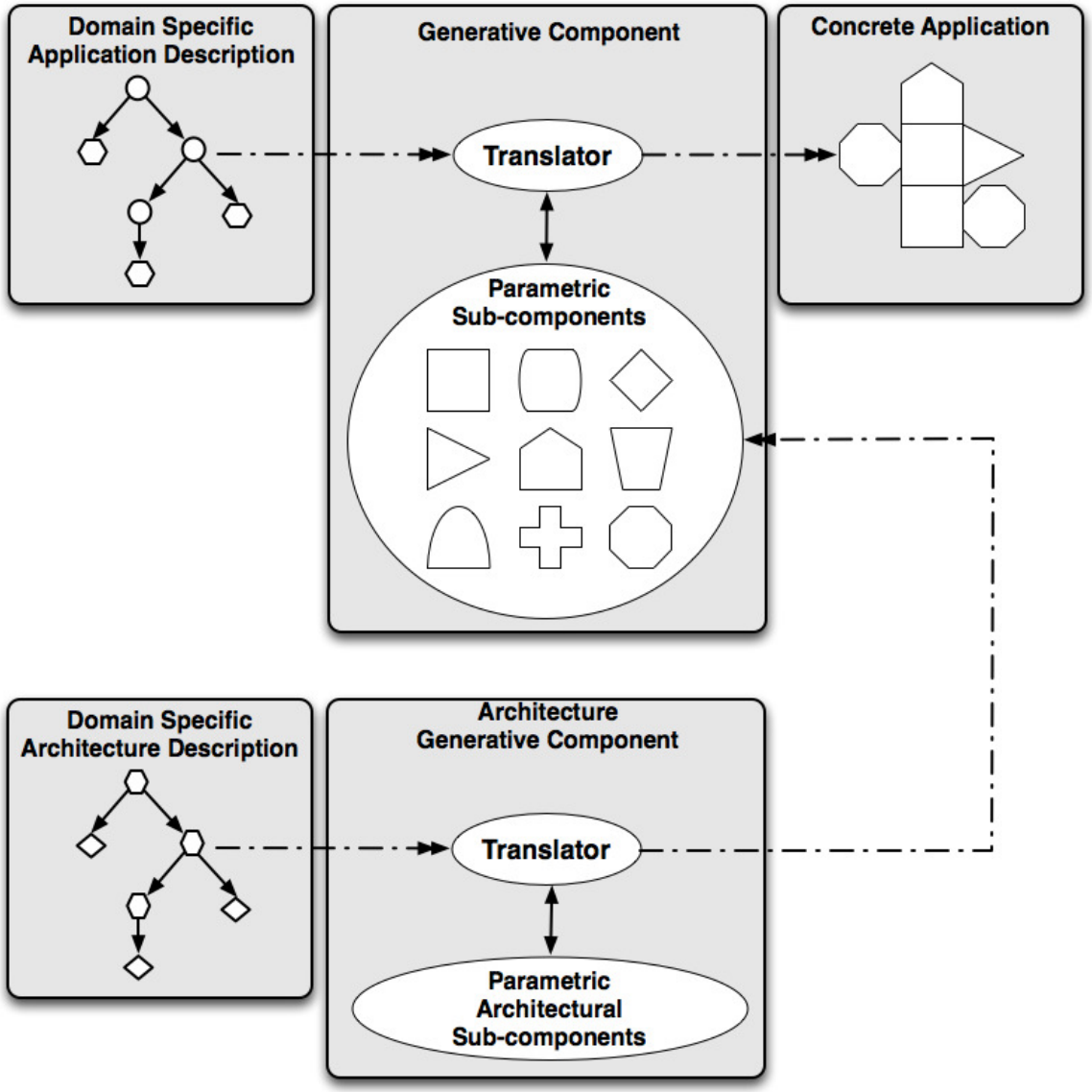

Figure 2. Principles of Architecture Aware Generative Programming

Our work uses a methodogy similar to SCALA [134], and more specifically, the DeLITE [157] toolset. Both approach rely on extracting high level, domain specific information from user code to optimize HPC applications. If our approach tries to maximize the use of compile-time optimization, DeLITE uses a runtime approach due to its reliance on the JAVA language.
In terms of libraries, various existing Scientific Computing library in C++ are actually available. The three most used are Armadillo [152], which shares a MATLAB-like API with our work, Blaze [69] which supports a similar cost based system for optimizing code and Eigen [100]. Our main feature compared to these solutions is the fact that hardware support is built-in the library core instead of beign tacked on the existing library, thus allowing us to support a larger amount of hardware.

3.1.2.3. Research tracks for the 4 next years

At short term, research and development on BOOST.SIMD and NT^2 will explore the applicability of our code generation methodology on distributed system, accelerators and heterogeneous systems. Large system support like Blue Gene/Q and other similar super-computer setup has been started.
Another axis of research is to apply generative programing to other scientific domain and to propose other domain specific tools using efficient code generators. Such a work has been started to explore the impact of generative programming on the design of portable linear algebra algorithms with an going PhD thesis on automatic generation of linear algebra software.
A mid-term objective is to bridge the gap with the Data Analytics community in order to both extract new expertise on how to make Big Data related issues scalable on modern HPC hardware and to provide tools for Data Analytics practitioners based on this collaboration.
On a larger scope, the implication of our methodology on language design will be explored. First by proposing evolution to C++ (as for example with our SIMD proposal [85]) so that generative programming can become a first class citizen in the language itself. Second by exploring how this methodology can be extended to other languages [99] or to other runtime systems including Cloud computing systems and JIT support. Application to other performance metric like power consumption is also planned [171].

3.1.3. Systematizing and automating program optimization

3.1.3.1. Scientific context

Delivering faster, more power efficient and reliable computer systems is vital for our society to continue innovation in science and technology. However, program optimization and hardware co-design became excessively time consuming, costly and error prone due to an enormous number of available design and optimization choices, and complex interactions between all software and hardware components. Worse, multiple characteristics have to be always balanced at the same time including execution time, power consumption, code size, memory utilization, compilation time, communication costs and reliability using a growing number of incompatible tools and techniques with many ad-hoc and intuition based heuristics. As a result, nearly peak performance of the new systems is often achieved only for a few previously optimized and not necessarily representative benchmarks while leaving most of the real user applications severely underperforming. Therefore, users are often forced to resort to a tedious and often non-systematic optimization of their programs for each new architecture. This, in turn, leads to an enormous waste of time, expensive computing resources and energy, dramatically increases development costs and time-to-market for new products and slows down innovation [41], [39], [46], [80].

3.1.3.2. Activity description and recent achievements

For the european project MILEPOST (2006-2009) [40], we, for the first time to our knowledge, attempted to address above challenges in practice with several academic and industrial partners including IBM, CAPS, ARC (now Synopsys) and the University of Edinburgh by combining automatic program optimization and tuning, machine learning and a public repository of experimental results. As a part of the project, we established a non-profit cTuning association (cTuning.org) that persuaded the community to voluntarily support our open source tools and repository while sharing benchmarks, data sets, tools and machine learning models even after the project. This approach, highly prised by the European Commission, Inria and the international community, helped us to substitute and automatically learn best compiler optimization heuristics by crowdsourcing auto-tuning (processing a large amount of performance statistics or "big data" collected from many users to classify application and build predictive models) [40], [91], [92]. However, it also exposed even more fundamental challenges including:

- Lack of common, large and diverse benchmarks and data sets needed to build statistically meaningful predictive models;
- Lack of common experimental methodology and unified ways to preserve, systematize and share our growing optimization knowledge and research material from the community including benchmarks, data sets, tools, tuning plugins, predictive models and optimization results;
- Problem with continuously changing, “black box” and complex software and hardware stack with many hardwired and hidden optimization choices and heuristics not well suited for auto-tuning and machine learning;
- Difficulty to reproduce performance results from the cTuning.org database submitted by the community due to a lack of full software and hardware dependencies;
- Difficulty to validate related auto-tuning and machine learning techniques from existing publications due to a lack of culture of sharing research artifacts with full experiment specifications along with publications in computer engineering.

As a result, we spent a considerable amount of our “research” time on re-engineering existing tools or developing new ones to support auto-tuning and learning. At the same time, we were trying to somehow assemble large and diverse experimental sets to make our research and experimentation on machine learning and data mining statistically meaningful. We spent even more time when struggling to reproduce existing machine learning-based optimization techniques from numerous publications. Worse, when we were ready to deliver auto-tuning solutions at the end of such tedious developments, experimentation and validation, we were already receiving new versions of compilers, third-party tools, libraries, operating systems and architectures. As a consequence, our developments and results were already potentially outdated even before being released while optimization problems considerably evolved.

We believe that these are major reasons why so many promising research techniques, tools and data sets for auto-tuning and machine learning in computer engineering have a life span of a PhD project, grant funding or publication preparation, and often vanish shortly after. Furthermore, we witness diminishing attractiveness of computer engineering often seen by students as “hacking” rather than systematic science. Many recent long-term research visions acknowledge these problems for computer engineering and many research groups search for “holy grail” auto-tuning solutions but no widely adopted solution has been found yet [39], [80].

3.1.3.3. Research tracks for the 4 next years

In this project, we will be evaluating the first, to our knowledge, alternative, orthogonal, interdisciplinary, community-based and big-data driven approach to address above problems. We are developing a knowledge management system for computer engineering (possibly based on GPL-licensed cTuning and BSD-licensed Collective Mind) to preserve and share through the Internet the whole experimental (optimization) setups with all related artifacts and exposed meta-description in a unified way including behavior characteristics (execution time, code size, compilation time, power consumption, reliability, costs), semantic and dynamic features, design and optimization choices, and a system state together with all software and hardware dependencies besides just performance data. Such approach allows community to consider analysis, design and optimization of computer systems as a unified, formalized and big data problem while taking advantage of mature R&D methodologies from physics, biology and AI.

During this project, we will gradually structure, systematize, describe and share all research material in computer engineering including tools, benchmarks, data sets, search strategies and machine learning models. Researchers can later take advantage of shared components to collaboratively prototype, evaluate and improve various auto-tuning techniques while reusing all shared artifacts just like LEGO™pieces, and applying machine learning and data mining techniques to find meaningful relations between all shared material. It can also help crowdsource long tuning and learning process including classification and model building among many participants.

At the same time, any unexpected program behavior or model mispredictions can now be exposed to the community through unified web-services for collaborative analysis, explanation and solving. This, in turn, enables reproducibility of experimental results naturally and as a side effect rather than being enforced - interdisciplinary community needs to gradually find and add missing software and hardware dependencies to the Collective Mind (fixing processor frequency, pinning code to specific cores to avoid contentions) or improve analysis and predictive models (statistical normality tests for multiple experiments) whenever abnormal behavior is detected.

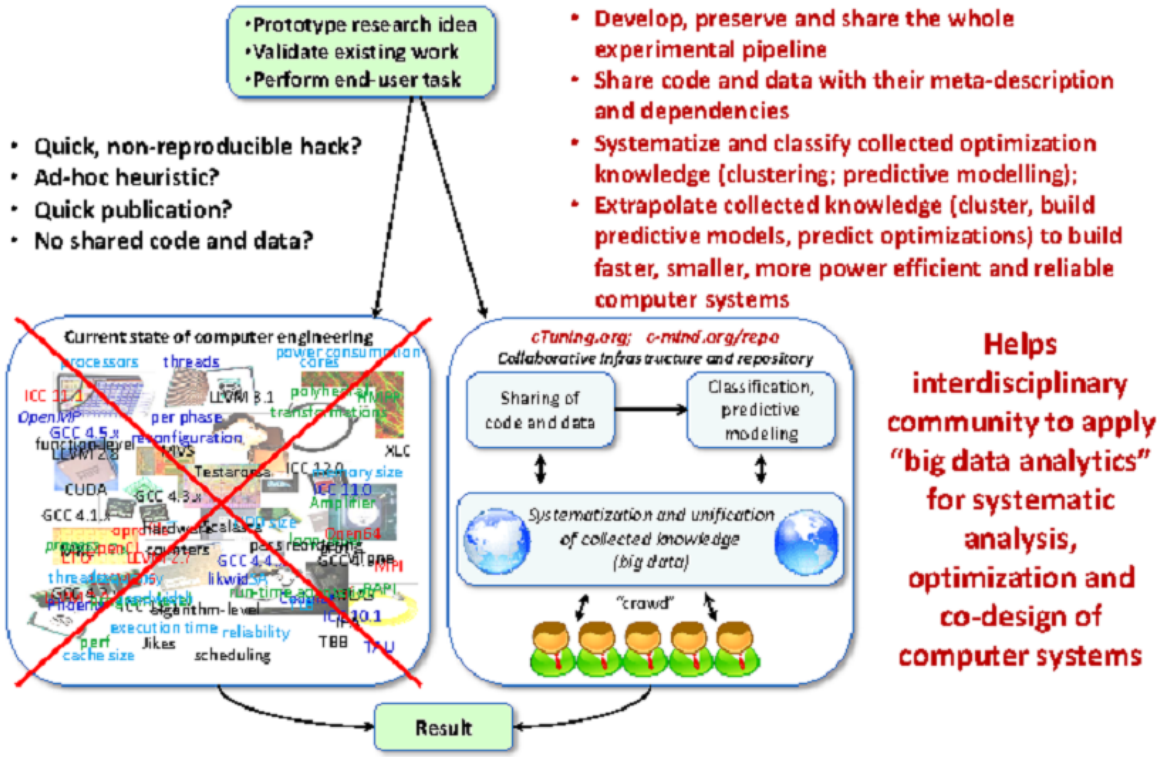

Figure 3. Considering program optimization and run-time adaptation as a "big data problem"

We hope that our approach will eventually help the community collaboratively evaluate and derive the most effective optimization strategies. It should also eventually help the community collaboratively learn complex behavior of all existing computer systems using top-down methodology originating from physics. At the same time, continuously collected and systematized knowledge ("big data") should allow community make quick and scientifically motivated advice about how to design and optimize the future heterogeneous HPC systems (particularly on our way towards extreme scale computing) as conceptually shown in Figure 3.

Similar systematization, formalization and big data analytics already revolutionized biology, machine learning, robotics, AI, and other important scientific fields in the past decade. Our approach also started revolutionizing computer engineering making it more a science rather than non-systematic hacking. It helps us effectively deal with the rising complexity of computer systems while focusing on improving classification and predictive models of computer systems' behavior, and collaboratively find missing features (possibly using new deep learning algorithms and even unsupervised learning [106], [126]) to improve optimization predictions, rather than constantly reinventing techniques for each new program, architecture and environment.

Our approach is strongly supported by a recent Vinton G. Cerf's vision for computer engineering [73] as well as our existing technology, repository of knowledge and experience, and a growing community [91], [92], [93]. Even more importantly, our approach already helped to promote reproducible research and initiate a new publication model in computer engineering supported by ACM SIGPLAN where all experimental results and related research artifacts with their meta-description and dependencies are continuously shared along with publications to be validated and improved by the community [90].

3.2. High-level HPC libraries and applications

In this research topic, we focus on developing optimized algorithms and software for high-performance scientific computing and image processing.

3.2.1. Taking advantage of heterogeneous parallel architectures

3.2.1.1. Activity description

In recent years and as observed in the latest trends from the Top 500 list [1], heterogeneous computing combining manycore systems with accelerators such as Graphics Processing Units (GPU) or Intel Xeon Phi coprocessors has become a *de facto* standard in high performance computing. At the same time, data movements between memory hierarchies and/or between processors have become a major bottleneck for most numerical algorithms. The main goal of this topic is to investigate new approaches to develop linear algebra algorithms and software for heterogeneous architectures [56], [164], with also the objective of contributing to public domain numerical linear algebra libraries (e.g., MAGMA [2]).
Our activity in the field consists of designing algorithms that minimize the cost of communication and optimize data locality in numerical linear algebra solvers. When combining different architectures, these algorithms should be properly "hybridized". This means that the workload should be balanced throughout the execution, and the work scheduling/mapping should ensure matching of architectural features to algorithmic requirements.

In our effort to minimize communication, an example concerns the solution of general linear systems (via LU factorization) where the main objective is to reduce the communication overhead due to pivoting. We developed several algorithms to achieve this objective for hybrid CPU/GPU platforms. In one of them the panel factorization is performed using a communication-avoiding pivoting heuristic [97] while the update of the trailing submatrix is performed by the GPU [51]. In another algorithm, we use a random preconditioning (see also Section 3.2.2) of the original matrix to avoid pivoting [54]. Performance comparisons and tests on accuracy showed that these solvers are effective on current hybrid multicore-GPU parallel machines. These hybrid solvers will be integrated in a next release of the MAGMA library.

Another issue is related to the impact of non-uniform memory accesses (NUMA) on the solution of HPC applications. For dense linear systems, we illustrated how an appropriate placement of the threads and memory on a NUMA architecture can improve the performance of the panel factorization and consequently accelerate the global LU factorization [148], when compared to the hybrid multicore/GPU LU algorithm as it is implemented in the public domain library MAGMA.

[1] http://www.top500.org/
[2] Matrix Algebra on GPU and Multicore Architectures, http://icl.cs.utk.edu/magma/

3.2.1.2. Research tracks for the 4 next years

3.2.1.2.1. Towards automatic generation of dense linear solvers:

In an ongoing research, we investigate a generic description of the linear system to be solved in order to exploit numerical and structural properties of matrices to get fast and accurate solutions with respect to a specific type of problem. Information about targeted architectures and resources available will be also taken into account so that the most appropriate routines are used or generated. An application of this generative approach is the possibility of prototyping new algorithms or new implementations of existing algorithms for various hardware.

A track for generating efficient code is to develop new functionalities in the C++ library NT^2 [89] which is developed in the Postale team. This approach will enable us to generate optimized code that support current processor facilities (OpenMP and TBB support for multicores, SIMD extensions...) and accelerators (GPU, Intel Xeon Phi) starting from an API (Application Programming Interface) similar to Matlab. By analyzing the properties of the linear algebra domain that can be extracted from numerical libraries and combining them with architectural features, we have started to apply the generic approach mentioned in Section 3.1.2 to solve dense linear systems on various architectures including CPU and GPU. As an application, we plan to develop a new software that can run either on CPU or GPU to solve least squares problems based on semi-normal equations in mixed precision [50] since, to our knowledge, such a solver cannot be found in current public domain libraries (Sca)LAPACK [43], [68], PLASMA [165] and MAGMA [52]. This solver aims at attaining a performance that corresponds to what state-of-the-art codes achieve using mixed precision algorithms.

3.2.1.2.2. Communication avoiding algorithms for heterogeneous platforms:

In previous work, we focused on the LU decomposition with respect to two directions that are numerical stability and communication issue. This research work has lead to the development of a new algorithm for the LU decomposition, referred to as LU_PRRP: LU with panel rank revealing pivoting [112]. This algorithm uses a new pivoting strategy based on strong rank revealing QR factorization [98]. We also design a communication avoiding version of LU_PRRP, referred to as CALU_PRRP, which aims at overcoming the communication bottleneck during the panel factorization if we consider a parallel version of LU_PRRP. Thus CALU_PRRP is asymptotically optimal in terms of both bandwidth and latency. Moreover, it is more stable than the communication avoiding LU factorization based on Gaussian elimination with partial pivoting in terms of growth factor upper bound [78].

Due to the huge number and the heterogeneity of computing units in future exascale platforms, it is crucial for numerical algorithms to exhibit more parallelism and pipelining. It is thus important to study the critical paths of these algorithms, the task decomposition and the task granularity as well as the scheduling techniques in order to take advantage of the potential of the available platforms. Our goal here is to adapt our new algorithm CALU_PRRP to be scalable and efficient on heterogeneous platforms making use of the available accelerators and coprocessors similarly to what was achieved in [51].

3.2.1.2.3. Application to numerical fluid mechanics:

In an ongoing PhD thesis [168], [169], we apply hybrid programming techniques to develop a solver for the incompressible Navier-Stokes equations with constant coefficients, discretized by the finite difference method. In this application, we focus on solving large sparse linear systems coming from the discretization of Helmholtz and Poisson equations using direct methods that represent the major part of the computational time for solving the Navier-Stokes equations which describe a large class of fluid flows. In the future, our effort in the field will concern how to apply hybrid programming techniques to solvers based on iterative methods. A major task will consist of developing efficient kernels and choosing appropriate preconditioners. An important aspect is also the use of advanced scheduling techniques to minimize the number of synchronizations during the execution. The algorithms developed during this research activity will be validated on physical data provided by the physicists either form the academic world (e.g., LIMSI/University Paris-Sud [3] or industrial partners (e.g., EDF, ONERA). This research is currently performed in the framework of the CALIFHA project [4] and will be continued in an industrial contract with EDF R&D (starting October 2014).

[3] Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur, http://www.limsi.fr/

[4] CALculations of Incompressible Fluids on Heterogeneous, funded by Région Île-de-France and Digitéo (http://www.digiteo.fr)

3.2.2. Randomized algorithms in HPC applications

Activity description

Randomized algorithms are becoming very attractive in high-performance computing applications since they are able to outperform deterministic methods while still providing accurate results. Recent advances in the field include for instance random sampling algorithms [47], low-rank matrix approximation [130], or general matrix decompositions [101].

Our research in this domain consists of developing fast algorithms for linear algebra solvers which are at the heart of many HPC physical applications. In recent works, we designed randomized algorithms [54], [66] based on random butterfly transformations (RBT) [135] that can be applied to accelerate the solution of general or symmetric indefinite (dense) linear systems for multicore [49] or distributed architectures [48]. These randomized solvers have the advantage of reducing the amount of communication in dense factorizations by removing completely the pivoting phase which inhibits performance in Gaussian Elimination.

We also studied methods and software to assess the numerical quality of the solution computed in HPC applications. The objective is to compute quantities that provide us with information about the numerical quality of the computed solution in an acceptable time, at least significantly cheaper than the cost for the solution itself (typically a statistical estimation should require $\mathcal{O}(n^2)$ flops while the solution of a linear system involves at least $\mathcal{O}(n^3)$ flops, where n is the problem size). In particular, we recently applied in [58] statistical techniques based on the small sample theory [111] to estimate the condition number of linear system/linear least squares solvers [45], [53], [57]. This approach reduces significantly the number of arithmetic operations in estimating condition numbers. Whether designing fast solvers or error analysis tools, our ultimate goal is to integrate the resulting software into HPC libraries so that these routines will be available for physicists. The targeted architectures are multicore systems possibly accelerated with GPUs or Intel Xeon Phi coprocessors.

This research activity benefits from the Inria associate-team program, through the **associate-team R-LAS** [5], created in 2014 between Inria Saclay/Postale team and University of Tennessee (Innovative Computing Laboratory) in the area of randomized algorithms and software for numerical linear algebra. This project is funded from 2014 to 2016 and is lead jointly by Marc Baboulin (Inria/University Paris-Sud) and Jack Dongarra (University of Tennessee).

Research tracks for the 4 next years

3.2.2.1. Extension of random butterfly transformations to sparse matrices:

We recently illustrated how randomization via RBT can accelerate the solution of dense linear systems on multicore architectures possibly accelerated by GPUs. We recently started to extend this method to sparse linear systems arising from the discretization of partial differential equations in many physical applications. However, a major difficulty comes from the possible fill-in introduced by RBT. One of our first task consists of performing experiments on a collection of sparse matrices to evaluate the fill-in depending on the number of recursions in the algorithm. In a recent work [59], we investigated the possibility of using another form of RBT (one-side RBT instead of two-sided) in order to minimize the fill-in and we obtain promising preliminary results (Figure 4 shows that the fill-in is significantly reduced when using one-side RBT).

Another track of research is related to iterative methods for solving large sparse linear systems, and more particularly preconditioned Krylov subspace methods implemented in the solver ARMS (Algebraic Recursive Multilevel Solver (pARMS for its parallel distributed version). In this solver, our goal is to find the last level of preconditioning and then replace the original ILU factorization by our RBT preprocessing. A PhD thesis (supervised by Marc Baboulin) started in October 2014 on using randomization techniques like RBT for sparse linear systems.

[5] Randomized Linear Algebra Software, https://www.lri.fr/~baboulin/presentation_r-las.html/

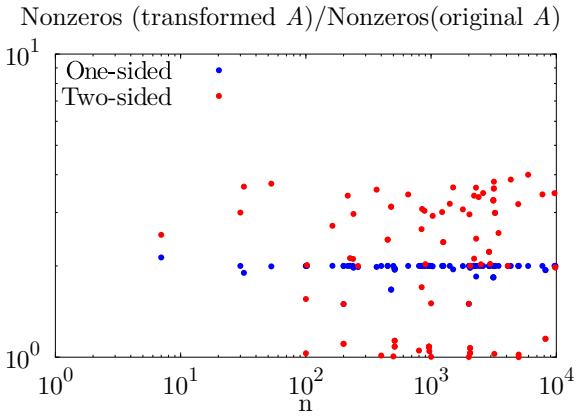

Figure 4. Evaluation of fill-in for one-sided RBT (90 matrices sorted by size).

3.2.2.2. Randomized algorithms on large clusters of multicore:

A major challenge for the randomized algorithms that we develop is to be able to solve very large problems arising in real-world physical simulations. As a matter of fact, large-scale linear algebra solvers from standard parallel distributed libraries like ScaLAPACK often suffer from expensive inter-node communication costs. An important requirement is to be able to schedule these algorithms dynamically on highly distributed and heterogeneous parallel systems [110]. In particular we point out that even though randomizing linear systems removes the communication due to pivoting, applying recursive butterflies also requires communication, especially if we use multiple nodes to perform the randomization. Our objective is to minimize this communication in the tiled agorithms and to use a runtime that enforces a strict data locality scheduling strategy [48]. A state of the art of possible runtime systems and how they can be combined with our randomized solvers will be established. Regarding the application of such solver, a collaboration with Pr Tetsuya Sakurai (University of Tsukuba, Japan) and Pr Jose Roman (Universitat Politècnica de València, Spain) will start in December 2014 to apply RBT to large linear systems encountered in contour integral eigensolver (CISS) [108]. Optimal tuning of the code will be obtained using holistic approach developed in the Postale team [93].

3.2.2.3. Extension of statistical estimation techniques to eigenvalue and singular value problems:

The extension of statistical condition estimation techniques can be carried out for eigenvalue/singular value calculations associated with nonsymmetric and symmetric matrices arising in, for example, optimization problems. In all cases, numerical sensitivity of the model parameters is of utmost concern and will guide the choice of estimation techniques. The important class of componentwise relative perturbations can be easily handled for a general matrix [111]. A significant outcome of the research will be the creation of high-quality open-source implementations of the algorithms developed in the project, similarly to the equivalent work for least squares problems [55]. To maximize its dissemination and impact, the software will be designed to be extensible, portable, and customizable.

3.2.2.4. Random orthogonal matrices:

Random orthogonal matrices have a wide variety of applications. They are used in the generation of various kinds of random matrices and random matrix polynomials [67], [77], [79], [105]. They are also used in some finance and statistics applications. For example the random orthogonal matrix (ROM) simulation [127] method uses random orthogonal matrices to generate multivariate random samples with the same mean and covariance as an observed sample.

The natural distribution over the space of orthogonal matrices is the Haar distribution. One way to generate a random orthogonal matrix from the Haar distribution is to generate a random matrix A with elements from the standard normal distribution and compute its QR factorization $A = QR$, where R is chosen to have nonnegative diagonal elements; the orthogonal factor Q is then the required matrix [104].

Stewart [155] developed a more efficient algorithm that directly generates an $n \times n$ orthogonal matrix from the Haar distribution as a product of Householder transformations built from Householder vectors of dimensions $1, 2, \cdots, n-1$ chosen from the standard normal distribution. Our objective is to design an algorithm that significantly reduces the computational cost of Stewart's algorithm by relaxing the property that Q is exactly Haar distributed. We also aim at extending the use of random orthogonal matrices to other randomized algorithms.

3.2.3. Embedded high-performance systems & computer vision

Scientific context

High-performance embedded systems & computer vision address the design of efficient algorithms for parallel architectures that deal with image processing and computer vision. Such systems must enforce realtime execution constraint (typically 25 frames per second) and power consumption constraint. If no COTS (*Component On The Shelf*) architecture (e.g., SIMD multicore processor, GPU, Intel Xeon Phi, DSP) satisfy the constraints, then we have to develop a specialized one.

A more and more important aspect when designing an embedded system is the tradeoff between speed (and power consumption) and numerical accuracy (and stability). Such a tradeoff leads to 16-bit computation (and storage) and to the design of less accurate algorithms. For example, the final accuracy for stabilizing an image is $10-1$ pixel, which is far from the maximum accuracy of (10^{-7}) available using the 32-bit IEEE format.

3.2.3.1. Activity description and recent achievements

Concerning image processing, our efforts concern the redesign of data-dependent algorithms for parallel architectures. A representative example of such an algorithm is the connected-component labeling (CCL) algorithm [147] which is used in industrial or medical imaging and classical computer vision like optical character recognition. As far as we know our algorithm (*Light Speed Labeling*) [71], [72] still outperforms other existing CCL algorithms [96], [103], [160] (the first versions of our algorithm appeared in 2009 [119], [120]).

Concerning computer vision (smart camera, autonomous robot, aerial drone), we developed in collaboration with LIMSI [6] two applications that run in realtime on embedded parallel systems [121], [146] with some accuracy tradeoffs. The first one is based on mean shift tracking [94], [95] and the second one relies on covariance matching and tracking [143], [144], [145].

These applications are used in video-surveillance: they perform motion detection [118], motion analysis [161], [162], motion estimation and multi-target tracking. Depending on the image nature and size, some algorithmic transforms (integral image, cumulative differential sum) can be applied and combined with hybrid arithmetic (16-bit / 32-bit / 64-bit). Finally, to increase the algorithm robustness color, space optimization is also used [122].

Usually one tries to convert 64-bit computations into 32-bit. But sometimes 16-bit floating point arithmetic is sufficient. As 16-bit numbers are now normalized by IEEE (754-2008) and are available in COTS processors like GPU and GPP (AVX2 for storage in memory and conversion into 32-bit numbers), we can run such kind of code on COTS processors or we can design specialized architectures like FPGA (*Field-Programmable gate array*) and ASIC (*Application-specific integrated circuit*) to be more efficient. This approach is complementary to that of [131] which converts 32-bit floating point signal processing operators into fixed-point ones.

[6] Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur

By extension to computer vision, we also address *interactive sensing HPC applications*. One CEA thesis funded by CEA and co-supervised by Lionel Lacassagne addresses the parallelization of Non Destructive Testing applications on COTS processors (super-charged workstation with GPUs and Intel Xeon Phi manycore processor). This PhD thesis deals with irregular computations with sparse-addressing and load-balacing problems. It also deals with floating point accuracy, by finding roots of polynomials using Newton and Laguerre algorithms. Depending on the configuration, 64-bit is required, but sometime 32-bit computations are sufficient with respect to the physics. As the second application focuses on interactive sensing, one has to add a second level of tradeoff for physical sampling accuracy and the sensor displacement [123], [124], [125], [141], [142].
In order to achieve realtime execution on the targeted architectures, we develop *High Level Transforms* (HLT) that are algorithmic transforms for memory layout and function re-organization. We show on a representative algorithm [102] in the image processing area that a fully parallelized code (SIMD+OpenMP) can be accelerated by a factor $\times 80$ on a multicore processor [115]. A CIFRE thesis (defended in 2014) funded by ST Microelectronics and supervised by Lionel Lacassagne has led to the design of very efficient implementations into an ASIC thanks to HLT. We show that the power consumption can be reduced by a factor 10 [170], [171].
All these applications have led to the development of software libraries for image processing that are currently under registration at APP (Agence de Protection des Programmes): `myNRC 2.0` [7] and `covTrack` [8].

3.2.3.2. Future: system, image & arithmetic

Concerning image processing we are designing new versions of CCL algorithms. One version is for parallel architectures where graph merging and efficient transitive closure is a major issue for load balancing. For embedded systems, *time prediction* is as important as execution time, so a specialized version targets embedded processors like ARM processors and Texas Instrument VLIW DSP C6x.

We also plan to design algorithms that should be less data-sensitive (the execution time depends on the nature of the image: a structured image can be processed quickly whereas an unstructured image will require more time). These algorithms will be used in even more data-dependent algorithms like *hysteresis thresholding* for image binarization, *split-and-merge* [44], [114] for realtime image segmentation using the Horowitz-Pavlidis quad-tree decomposition [107]. Such an algorithm could be useful for accelerating image decomposition like *Fast Level Set Transform* algorithm [132].

Concerning Computer vision we will study 16-bit floating point arithmetic for image processing applications and linear algebra operators. Concerning image processing, we will focus on iterative algorithms like optical flow computation (for motion estimation and image stabilization). We will compare the efficiency (accuracy and speed) of 16-bit floating point [86], [117], [116], [139] with fixed-point arithmetic. Concerning linear algebra, we will study efficient implementation for very small matrix inversion (from 6×6 up to 16×16) for our covariance-tracking algorithm.

According to Nvidia (see Figure 5), the computation rate (Gflop/s) for ZGEMM (complex matrix-matrix multiplication with 64-bit precision – for small value of N – is linearly proportional to N. That means that, for a 6×6 matrix, we achieve around 6 Gflop/s on a Tesla M2090 (400 Gflop/s peak power). This represents 1.5 % of the peak power. For that reason, designing efficient parallel codes for embedded systems [74], [81], [82] is different and may be more complex than designing codes for classical HPC systems. Our `covTrack` software requires many hundreds of 6×6 matrix-matrix multiplications every frame.
Last point is to develop tools that help to automatically distribute or parallelize a code on an architecture code parallelization/distribution dealing with scientific computing [83], MPI [87] or image applications on the Cell processor [75], [88], [138], [149], [150], [151], [163].

[7] smart memory allocator and management for 2D and 3D image processing
[8] agile realtime multi-target tracking algorithm, co-developped with Michèle Gouiffès at LIMSI

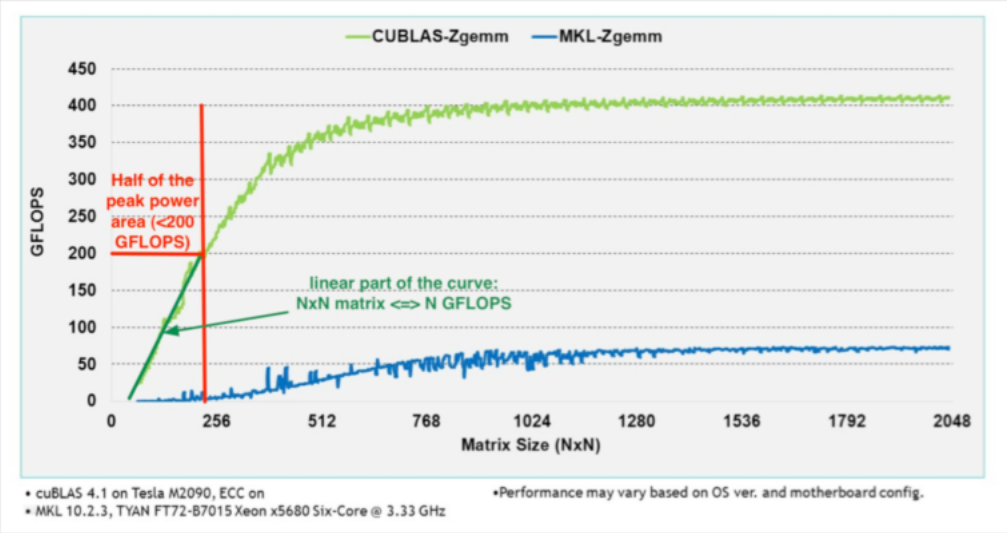

Figure 5. Nvidia cuBLAS performance versus Intel MKL: both have poor performance for small N

4. New Software and Platforms

4.1. New Software

4.1.1. MyNRC: image-oriented library for allocation and manipulation of SIMD 1D, 2D and 3D structures

Participant: Lionel Lacassagne.

MyNRC is multi-plateform library that can handle SSE, AVX, Neon and ST VECx registers.

4.1.2. CovTrack: agile realtime multi-target tracking algorithm

Participants: Michèle Gouiffès, Lionel Lacassagne, Florence Laguzet, Andrés Romero.

4.1.3. tmLQCD for Blue Gene/Q

Participant: Michael Kruse [correspondant].

tmLQCD is a program suite for lattice quantum chromodynamics (Lattice QCD) using the chirally twisted Wilson quarks to reduce discretization artifacts. This software is in productive use by the European Twisted Mass Collaboration (ETMC).

As to not waste precious computation time on the supercomputers it is running on, it is important to optimize the code in order to run as fast as possible. tmLQCD has already been customized for Intel Xeon processors, the Blue Gene/L and Blue Gene/P from IBM. For the latter's successor, the Blue Gene/Q, more profound changes to the program are necessary. With these changes, tmLQCD reaches a peak performance of up to 55% of the machines theoretical floating point performance.

The Blue Gene/Q optimized tmLQCD is available at: http://github.com/Meinersbur/tmLQCD

4.1.4. Molly

Participant: Michael Kruse [correspondant].

Using Polly extension, the LLVM compiler framework is able to automatically parallelize general programs for shared memory threading for by exploiting the powerful analysis and transformations of the polyhedral model.

Molly adds the ability to manage distributed memory using the polyhedral model and is therefore able to automatically parallelize even for the largest of today's supercomputer. Once the distribution of data between the computer's nodes is known, Molly determines the values that are required to be transferred between the nodes and chunks them into as few messages as possible. It also keeps tracks of the buffers required by the MPI interface. Transfers are asynchronous such that further computations take place while the data is being transferred.

Molly has not yet been officially released.

4.1.5. Dohko (http://dohko.io/)

Participant: Alessandro Ferreira Leite [correspondant].

Automating multi-cloud configuration is a difficult task. The difficulties are mostly due to clouds' heterogeneity and the lack of tools to coordinate cloud computing configurations automatically. As a result, virtual machine image (VMI) is the common approach to configure cloud environment. Although VMI can handle functional properties like minimum disk size, operating system, and software packages, it leads to a high number of configuration options, increasing the difficulty to select one that matches users' requirements. Moreover, the usage of VMI usually results in vendor lock-in. Furthermore, VMI leaves for the users the work of selecting a resource to deploy the image and for orchestrating them accordingly, i.e., the work of selecting and instantiating the VMI in each cloud. In addition, VMI migration across multiple clouds normally has a high cost due to network traffic. Finally, in case of cloud's failure, it may be difficult for users to re-create the failed environment in another cloud, since the image will be inaccessible.

Therefore, to overcome these issues, we developed a configuration management tool for cloud computing. Our tool, called Dohko, allows the users to configure a multi-cloud computing environment, following a declarative strategy. In this case, the users describe their applications and requirements and use our tool to select the resources and to set up the whole computing environment automatically, taking into account temporal and functional dependencies between the resources. Moreover, following a software product line (SPL) engineering method, Dohko captures the knowledge of configuring cloud environments in form of reusable assets. In this case, a product is a cloud computing environment that meets the user requirements, where the requirements can be either based on high or low-level descriptions. Examples of low-level descriptions include: virtualization type, disk technology, sustainable performance, among others, whereas high-level descriptions include the number of CPU cores, the RAM memory size, and the maximum monetary cost per hour. In this context, a cloud computing environment also matches cloud's configuration constraints. Besides that, thanks to the usage of an extended feature model (EFM) with attributes, our approach enables the description of the whole computing environment (i.e., hardware and software) independent of cloud provider and without requiring the usage of virtual machine image. In this case, it relies on an off-the-shelf constraint satisfaction problem (CSP) solver to implement the feature model and to select the resources.

By employing a declarative strategy, Dohko could execute a biological sequence comparison application in two distinct cloud providers (i.e., Amazon EC2 and Google Compute Engine) considering a single and a multi-cloud scenario, demanding minimal users' intervention to instantiate the whole cloud environment, as well as to execute the application. In particular, our solution tackles the lack of middleware prototypes that can support different scenarios when using services from many clouds. Moreover, it meets the functional requirements identified for multiple cloud-unaware systems [136] such as: (a) it provides a way to describe functional and non-functional requirements through the usage of an SPL engineering method; (b) it can aggregate services from distinct clouds; (c) it provides a homogeneous interface to access services of multiple clouds; (d) it allows the service selection of the clouds; (e) it can deploy its components across many clouds; (f) it provides automatic procedures for deployments; (g) it utilizes an overlay network to connect and to organize the resources; (h) it does not impose any constraint for the connected clouds.

4.2. Platforms

4.2.1. Fast linear system solvers in public domain libraries (http://icl.cs.utk.edu/magma/)

Participant: Marc Baboulin [correspondant].

Hybrid multicore+GPU architectures are becoming commonly used systems in high performance computing simulations. In this research, we develop linear algebra solvers where we split the computation over multicore and graphics processors, and use particular techniques to reduce the amount of pivoting and communication between the hybrid components. This results in efficient algorithms that take advantage of each computational unit [12]. Our research in randomized algorithms yields to several contributions to propose public domain libraries PLASMA and MAGMA in the area of fast linear system solvers for general and symmetric indefinite systems. These solvers minimize communication by removing the overhead due to pivoting in LU and $LDLT$ factorization. Different approaches to reduce communication are compared in [2].

See also the web page http://icl.cs.utk.edu/magma/.

4.2.2. cTuning Framework (http://cTuning.org): Repository and Tools for Collective Characterization and Optimization of Computing Systems

Participant: Grigori Fursin [correspondant].

Designing, porting and optimizing applications for rapidly evolving computing systems is often complex, ad-hoc, repetitive, costly and error prone process due to an enormous number of available design and optimization choices combined with the complex interactions between all components. We attempt to solve this fundamental problem based on collective participation of users combined with empirical tuning and machine learning.

We developed cTuning framework that allows to continuously collect various knowledge about application characterization and optimization in the public repository at cTuning.org. With continuously increasing and systematized knowledge about behavior of computer systems, users should be able to obtain scientifically motivated advices about anomalies in the behavior of their applications and possible solutions to effectively balance performance and power consumption or other important characteristics.

Currently, we use cTuning repository to analyze and learn profitable optimizations for various programs, datasets and architectures using machine learning enabled compiler (MILEPOST GCC). Using collected knowledge, we can quickly suggest better optimizations for a previously unseen programs based on their semantic or dynamic features [10].

We believe that such approach will be vital for developing efficient Exascale computing systems. We are currently developing the new extensible cTuning2 framework for automatic performance and power tuning of HPC applications.

For more information, see the web page http://cTuning.org.

4.2.3. NT2 (http://www.github.com/MetaScale/nt2)

Participants: Pierre Esterie, Joël Falcou, Mathias Gaunard, Ian Masliah, Antoine Tran Tan.

NT2 is a C++ high level framework for scientific computing.[18]

4.2.4. Boost.SIMD (http://www.github.com/MetaScale/nt2)

Participants: Pierre Esterie, Joël Falcou, Mathias Gaunard.

Boost.SIMD provides a portable way to vectorize computation on Altivec, SSE or AVX while providing a generic way to extend the set of supported functions and hardwares.

5. New Results

5.1. Highlights of the Year

CovTrack: Agile multi-target multi-threaded realtime tracker We have developed and highly optimized a multi-target tracking system based on covariance tracking algorithm. The complexity of the algorithm – connected to the number of features – can be tuned to fit the processor computation power (with/without SIMD). Moreover the features can be also selected from a large set of features to adapt the algorithm to the scene and the nature of tracking (indoor/outdoor, pedestrian/car,). Some software and algorithmic transforms have been also applied to accelerate the code for scalar/SIMD processors. [20]

The Light Speed Labeling (LSL) algorithm is still the world fastest connected component labeling (CCL) algorithm. We have proposed a new benchmark that performs fair comparisons for such a data-dependent algorithm (that involves Union-Find algorithm optimization combined with memory and control flow optimization). We show that thanks to its run-based approach and its line-relative labeling, LSL is intrinsically more efficient that all State-of-the-Art pixel-based algorithms, whatever the memory management.[23]

5.2. Excalibur: An Autonomic Cloud Architecture for Executing Parallel Applications

Participants: Alessandro Ferreira Leite, Claude Tadonki, Christine Eisenbeis, Tainá Raiol, Maria Emilia Walter, Alba Cristina de Melo.

IaaS providers often allow the users to specify many requirements for their applications. However, users without advanced technical knowledge usually do not provide a good specification of the cloud environment, leading to low performance and/or high monetary cost. In this context, the users face the challenges of how to scale cloud-unaware applications without re-engineering them. Therefore, in this paper, we propose and evaluate a cloud architecture, namely Excalibur, to execute applications in the cloud. In our architecture, the users provide the applications and the architecture sets up the whole environment and adjusts it at runtime accordingly. We executed a genomics workflow in our architecture, which was deployed in Amazon EC2. The experiments show that the proposed architecture dynamically scales this cloud-unaware application up to 10 instances, reducing the execution time by 73compared to the execution in the configuration specified by the user.[25]

5.3. A Fine-grained Approach for Power Consumption Analysis and Prediction

Participants: Alessandro Ferreira Leite, Claude Tadonki, Christine Eisenbeis, Alba Cristina de Melo.

Power consumption has became a critical concern in modern computing systems for various reasons including financial savings and environmental protection. With battery powered de-vices, we need to care about the available amount of energy since it is limited. For the case of supercomputers, as they imply a large aggregation of heavy CPU activities, we are exposed to a risk of overheating. As the design of current and future hardware is becoming more and more complex, energy prediction or estimation is as elusive as that of time performance. However, having a good prediction of power consumption is still an important request to the computer science community. Indeed, power consumption might become a common performance and cost metric in the near future. A good methodology for energy prediction could have a great impact on power-aware programming, compilation, or runtime monitoring. In this paper, we try to understand from measurements where and how power is consumed at the level of a computing node. We focus on a set of basic programming instructions, more precisely those related to CPU and memory. We propose an analytical prediction model based on the hypothesis that each basic instruction has an average energy cost that can be estimated on a given architecture through a series of micro-benchmarks. The considered energy cost per operation includes both the overhead of the embedding loop and associated (hardware/software) optimizations. Using these precalculated values,

we derive a linear extrapolation model to predict the energy of a given algorithm expressed by means of atomic instructions. We then use three selected appli-cations to check the accuracy of our prediction method by comparing our estimations with the corresponding measurements obtained using a multimeter. We show a 9.48% energy prediction on sorting.[27]

5.4. Automated Code Generation for Lattice Quantum Chromodynamics and beyond

Participants: Denis Barthou, Konstantin Petrov, Olivier Brand-Foissac, Olivier Pène, Gilbert Grosdidier, Michael Kruse, Romain Dolbeau, Christine Eisenbeis, Claude Tadonki.

This is ongoing work on a Domain Specific Language which aims to simplify Monte-Carlo simulations and measurements in the domain of Lattice Quantum Chromodynamics. The tool-chain, called Qiral, is used to produce high-performance OpenMP C code from LaTeX sources. We discuss conceptual issues and details of implementation and optimization. The comparison of the performance of the generated code to the well-established simulation software is also made.[17]

5.5. Switchable Scheduling for Runtime Adaptation of Optimization

Participants: Lénaïc Bagnères, Cédric Bastoul.

Parallel applications used to be executed alone until their termination on partitions of supercomputers: a very static environment for very static applications. The recent shift to multicore architectures for desktop and embedded systems as well as the emergence of cloud computing is raising the problem of the impact of the execution context on performance. The number of criteria to take into account for that pur-pose is significant: architecture, system, workload, dynamic parameters, etc. Finding the best optimization for every context at compile time is clearly out of reach. Dynamic optimization is the natural solution, but it is often costly in execution time and may offset the optimization it is en-abling. In this paper, we present a static-dynamic compiler optimization technique that generates loop-based programs with dynamic auto-tuning capabilities with very low overhead. Our strategy introduces switchable scheduling, a family of program transformations that allows to switch between optimized versions while always processing useful computation. We present both the technique to generate self-adaptive programs based on switchable scheduling and experimental evidence of their ability to sustain high-performance in a dynamic environment.[22]

5.6. Efficient distributed randomized algorithms for solving large dense symmetric indefinite linear systems

Participants: Marc Baboulin, Dulceneia Becker, George Bosilca, Anthony Danalis, Jack Dongarra.

Randomized algorithms are gaining ground in high-performance computing applications as they have the potential to outperform deterministic methods, while still providing accurate results. We propose a randomized solver for distributed multicore architectures to efficiently solve large dense symmetric indefinite linear systems that are encountered, for instance, in parameter estimation problems or electromagnetism simulations. Our contribution is to propose efficient kernels for applying random butterfly transformations (RBT) and a new distributed implementation combined with a runtime (PaRSEC) that automatically adjusts data structures, data mappings, and the scheduling as systems scale up. Both the parallel distributed solver and the supporting runtime environment are innovative. To our knowledge, the randomization approach associated with this solver has never been used in public domain software for symmetric indefinite systems. The underlying runtime framework allows seamless data mapping and task scheduling, mapping its capabilities to the underlying hardware features of heterogeneous distributed architectures. The performance of our software is similar to that obtained for symmetric positive definite systems, but requires only half the execution time and half the amount of data storage of a general dense solver. [15]

5.7. Solvers for 3D incompressible Navier-Stokes equations on hybrid CPU/GPU systems

Participants: Yushan Wang, Marc Baboulin, Karl Rupp, Olivier Le Maître, Yann Fraigneau.

We developed a hybrid multicore/GPU solver for the incompressible Navier-Stokes equations with constant coefficients, discretized by the finite difference method. By applying the prediction-projection method, the Navier-Stokes equations are transformed into a combination of Helmholtz-like and Poisson equations for which we describe efficient solvers. We propose a new implementation that takes advantage of GPU accelerators. We present numerical experiments on a current hybrid machine.

5.8. The Numerical Template toolbox: A Modern C++ Design for Scientific Computing

Participants: Pierre Esterie, Joël Falcou, Mathias Gaunard, Jean-Thierry Lapresté, Lionel Lacassagne.

The design and implementation of high level tools for parallel programming is a major challenge as the complexity of modern architectures increases. Domain Specific Languages (or DSL) have been proposed as a solution to facilitate this design but few of those DSL s actually take full advantage of said parallel architectures. In this paper, we propose a library-based solution by designing a C++ DSL s using generative programming: View the MathML source. By adapting generative programming idioms so that architecture specificities become mere parameters of the code generation process, we demonstrate that our library can deliver high performance while featuring a high level API and being easy to extend over new architectures. [18]

5.9. Boost.SIMD: generic programming for portable simdization

Participants: Pierre Esterie, Joël Falcou, Mathias Gaunard, Jean-Thierry Lapresté, Lionel Lacassagne.

Abstract SIMD extensions have been a feature of choice for processor manufacturers for a couple of decades. Designed to exploit data parallelism in applications at the instruction level, these extensions still require a high level of expertise or the use of potentially fragile compiler support or vendor-specific libraries. While a large fraction of their theoretical accelerations can be obtained using such tools, exploiting such hardware becomes tedious as soon as application portability across hardware is required. In this paper, we describe Boost.SIMD, a C++ template library that simplifies the exploitation of SIMD hardware within a standard C++programming model. Boost.SIMD provides a portable way to vectorize computation on Altivec, SSE or AVX while providing a generic way to extend the set of supported functions and hardwares. We introduce a C++standard compliant interface for the users which increases expressiveness by providing a high-level abstraction to handle SIMD operations, an extension-specific optimization pass and a set of SIMD aware standard compliant algorithms which allow to reuse classical C++ abstractions for SIMD computation. We assess Boost.SIMD performance and applicability by providing an implementation of BLAS and image processing algorithms.

5.10. Automatic Task-based Code Generation for High Performance Domain Specific Embedded Language

Participants: Antoine Tran Tan, Joël Falcou, Daniel Etiemble, Harmut Kaiser.

Providing high level tools for parallel programming while sustaining a high level of performance has been a challenge that techniques like Domain Specific Embedded Languages try to solve. In previous works, we investigated the design of such a DSEL-NT2- providing a Matlab-like syntax for parallel numerical computations inside a C++ library. In this paper, we show how NT2 has been redesigned for shared memory systems in an extensible and portable way.[28]

5.11. High Level Transforms for SIMD and low-level computer vision algorithms

Participants: Lionel Lacassagne, Daniel Etiemble, Alain Dominguez, Pascal Vezolle.

This paper presents a review of algorithmic transforms called High Level Transforms for IBM, Intel and ARM SIMD multi-core processors to accelerate the implementation of low level image processing algorithms. We show that these optimizations provide a significant acceleration. A first evaluation of 512-bit SIMD XeonPhi is also presented. We focus on the point that the combination of optimizations leading to the best execution time cannot be predicted, and thus, systematic benchmarking is mandatory. Once the best configuration is found for each architecture, a comparison of these performances is presented. The Harris points detection operator is selected as being *representative* of low level image processing and computer vision algorithms. Being composed of five convolutions, it is more complex than a simple filter and enables more opportunities to combine optimizations. The presented work can scale across a wide range of codes using 2D stencils and convolutions. Such High Level Transforms provide a speedup of x89 on a 2×4 core Intel Xeon processor versus a code that is already SIMDized and OPenMPized.[26]

5.12. What Is the World's Fastest Connected Component Labeling Algorithm?

Participants: Laurent Cabaret, Lionel Lacassagne.

Optimizing connected component labeling is cur- rently a very active research field. Some teams claim to have design the fastest algorithm ever designed. This paper presents a review of these algorithms and a enhanced benchmark that improve classical random images benchmark with a varying granularity set of random images in order to become closer to natural image behavior. Our algorithm, the Light Speed Labeling is from $\times 3.5$ up to $\times 5.3$ faster than the best State-of-the-Art competitor.[23]

5.13. Covariance tracking: architecture optimizations for embedded systems

Participants: Andrés Romero, Lionel Lacassagne, Michèle Gouiffès, Ali Hassan Zahraee.

Covariance matching techniques have recently grown in interest due to their good performances for object retrieval, detection, and tracking. By mixing color and texture information in a compact representation, it can be applied to various kinds of objects (textured or not, rigid or not). Unfortunately, the original version requires heavy computations and is difficult to execute in real time on embedded systems. This article presents a review on different versions of the algorithm and its various applications; our aim is to describe the most crucial challenges and particularities that appeared when implementing and optimizing the covariance matching algorithm on a variety of desktop processors and on low-power processors suitable for embedded systems. An application of texture classification is used to compare different versions of the region descriptor. Then a comprehensive study is made to reach a higher level of performance on multi-core CPU architectures by comparing different ways to structure the information, using single instruction, multiple data (SIMD) instructions and advanced loop transformations. The execution time is reduced significantly on two dual-core CPU architectures for embedded computing: ARM Cortex-A9 and Cortex-A15 and Intel Penryn-M U9300 and Haswell-M 4650U. According to our experiments on covariance tracking, it is possible to reach a speedup greater than 2 on both ARM and Intel architectures, when compared to the original algorithm, leading to real-time execution. [20]

6. Bilateral Contracts and Grants with Industry

6.1. Bilateral Contracts with Industry

- **EDF R& D**: this is a collaboration with the department SINETICS of EDF in the area of high-performance computing.
  Participants: Marc Baboulin, Grigori Fursin, Amal Khabou.

  It concerns two differents topics:

  - Enhancing performance of numerical solvers using accelerators (postdoc starting in October 2014) and vectorization techniques (internship starting in November 2014).
  - Studying numerical quality and reproducibility in HPC exascale applications (ongoing ANR submission).

- **ARM Ltd**
  Participant: Grigori Fursin.

  UK: this collaboration is related to systematizing benchmarking of OpenCL programs for new ARM GPU architectures and applying machine learning to predict better optimizations (Grigori Fursin).
- **Collaboration with the small size company NumScale** (PME, 10 people) NumScale on C++ parallel code generation technology. NumScale is a start-up created in 2012 as the result of a Digiteo/University Paris Sud technological transfer program (Digiteo OMTE). NumScale exploits scientific results and tools based around code generation for parallel programs as well as advanced code optimization techniques developed by members of the team.

7. Partnerships and Cooperations

7.1. Regional Initiatives

- **CALIFHA project (DIM Digiteo 2011)**: CALculations of Incompressible Fluid flows on Heterogeneous Architectures. Funding for a PhD student. Collaboration with LIMSI/CNRS. Participants: Marc Baboulin (Principal Investigator), Joel Falcou, Yann Fraigneau (LIMSI), Laura Grigori, Olivier Le Maître (LIMSI), Laurent Martin Witkowski (LIMSI)

7.2. National Initiatives

- **EDF**: Contract with EDF on improving performance and designing algorithms of iterative solvers on parallel machines with accelerators (Marc Baboulin). This contract enables to hire a postdoc researcher in October 2014.
  Participants: Marc Baboulin, Amal Khabou.
- **Lal/In2P3/CERN** The collaboration with CERN and LAL/IN2P3 + LRI focuses on LHCb and Atlas tracker code optimization. Those experiments must analyze results in realtime (10ms for analyzing particle trajectography). Early results show that these tracking algorithms can run in real time on SIMD multicore General Purpose Processor and on Xeon-Phi.
  Participant: Lionel Lacassagne.
- **Inserm** Contract with Paris X / INSERM U669 (Christophe Genolini) in the R++ project. R++ is an open source effort to modernize and increase performance of the R language used by scientists to develop statistical analysis tools. Funding for one research engineer has been received to support this project.
  Participant: Joël Falcou.
- **followup of the ANR Cosinus project PetaQCD - Towards PetaFlops for Lattice Quantum ChromoDynamics** Collaboration with Lal (Orsay), LPT (Orsay), LABRI (Bordeaux). About the design of architecture, software tools and algorithms for Lattice Quantum Chromodynamics.
  Participants: Christine Eisenbeis, Michael Kruse, Konstantin Petrov.

7.3. European Initiatives

7.3.1. ITEA

Program: ITEA

Project acronym: MANY

Project title: Many-core Programming and Resource Management for High-Performance Embedded Systems

Duration: 09/2011 - 08/2014

Coordinator: XDIN

Other partners: France: Thales Communications and Security, CAPS Entreprise, Telecom SudParis; Spain: UAB; Sweden: XDIN; Korea: ETRI, TestMidas, SevenCore; Netherlands: Vector Fabrics, ST-Ericsson, TU Eindhoven; Belgium: UMONS.

Abstract: Adapting Industry for the for the disruptive landing of many-core processors in Embedded Systems in order to provide scalable, reusable and very fast sofware development.
Participants: Lénaïc Bagnères, Cédric Bastoul, Taj Muhammad Khan.

7.4. International Initiatives

7.4.1. Inria Associate Teams

Participants: Marc Baboulin, Jack Dongarra.

R-LAS is an Inria associate team with University of Tennessee,(https://www.lri.fr/~baboulin/r-las.html), 2014-2017.

This project is proposed in the context of developing a class of fast algorithms based on randomization for numerical linear algebra solvers. The funding was used in 2014 to cover exchange visits for researchers and PhD students from Inria and University of Tennessee.

7.4.1.1. Informal International Partners

- **Lawrence Berkeley National Laboratory** - USA: collaboration of Marc Baboulin with Sherry Li on application of randomization techniques to the solution of large sparse linear systems using direct methods (joint publications and co-organizations of mini-symposia for SIAM conferences).
- **Old Dominion University** - USA: Collaboration with Pr. Masha Sosonkina on locality optimization for numerical linear algebra solvers (joint publication) and preconditioned Krylov subspace methods (PhD thesis of Aygül Jamal, starting in October 2014).
- **Louisiana State University** - USA: collaboration of Joel Falcou with the STELLAR team in the framework of the HPX project (Hartmut Kaiser). It is mainly related to the design and implementation of a C++ asynchronous runtime system. In this framework, the STELLAR team hosted 2 PhD students of the Postale team for extended visits in 2013 and 2014.
- **Texas A&M University** - USA: collaboration of Joel Falcou with the PARASOL team in the framework of the STAPL project (Lawrence Rauchwerger). It is mainly related to the applicability of parallel skeletons inside STAPL on large scale parallel machines.
- **University of Illinois at Urbanna Champaign** (UIUC) - USA, in the context of the Inria Joint Laboratory for Petascale computing. Since 2011, we have initiated collaborations with researchers from UIUC (Wen-mei Hwu, Karl Rupp) in the area of numerical software.
- **University of Manchester**: collaboration with Professors Nick Higham and Françoise Tisseur on random orthogonal matrices and fault-tolerant linear algebra algorithms (Amal Khabou).
- **University of California - Irvine**: collaboration of Christine Eisenbeis with Professor Jean-Luc Gaudiot on Application Characterization for Modern Multicore Architectures

7.4.2. Participation In other International Programs

Stic AmSud: BioCloud-EEAmSud **Participants:** Christine Eisenbeis, Alessandro Ferreira Leite, Claude Tadonki.

BioCloud-EEAmSud is a cooperation project integrated by Brazil, Chile and France following the 2012 STIC-AmSud call. Partners in Brazil are Universidade de Brasilia, Universidade Federal Fluminense, and EMBRAPA-Genetic Resources and Biotechnology (CENARGEN), through the support of the Coordination of Improvement of Senior Staff of the Ministry of Education in Brazil (CAPES). In Chile, the main partner is Universidad de Santiago de Chile, through the support of the National Commission for Scientific and Technological Research of Chile (CONICYT). In France, the institutions involved are Mines ParisTech (CRI) and Inria-Saclay, through the support of the Ministry of Foreign and European Affairs (MAEE). The international project coordinator is Pr. Maria Emília Machado Telles Walter (UnB). Alessandro Ferreira Leite' thesis work is a joint University of Brazilia - université Paris-Sud 11 thesis and is partially supported by BioCloud-EEAmSud. Maria Emilia Machado Telles Walter and Alba Cristian de Melo visited Grand-Large in 2013, as well as Taina Rajol.

7.5. International Research Visitors

7.5.1. Visits of International Scientists

- Masha Sosokina (Professor, Old Dominion University, USA), June 10-13, 2014.
- Tingxing Tim Dong (PhD student, University of Tennessee, USA), August 25-26, 2014.
- Anthony Danalis (University of Tennessee, USA), December 15-16, 2014.
- Tetsuya Sakurai (University of Tsukuba, Japan), December 15-16, 2014.
- Jose Roman (University of Valencia, Spain), December 15-16, 2014.
- Jean-Luc Gaudiot, UCLA, Irvine, March 3rd, September 4th, November 24th, 2014.

8. Dissemination

8.1. Promoting Scientific Activities

Marc Baboulin Member of Steering Committee of ACM High Performance Computing Symposium (HPC'14), Tampa, Florida, April 13-16, 2014.

Marc Baboulin organizer of the workshop "Linear least squares and applications", 19th International Linear Algebra Society Conference (ILAS 2014), Seoul, South Korea, Aug. 06-09, 2014.

Marc Baboulin organizer of the minisymposium "Randomized algorithms in parallel matrix computations" at the SIAM Conference on Parallel Processing for Scientific Computing, Portland (OR), USA, Feb. 18-21, 2014.

Christine Eisenbeis IJPP (International Journal on Parallel Programming) editorial board.

Christine Eisenbeis comité d'organisation du colloque "Recherche et démocratie", 21-22 mai 2014, Orsay, (http://www.centre-dalembert.u-psud.fr/archives-colloques/2014-recherche-scientifique-et-democratie/).

Christine Eisenbeis 5th Workshop on applications for multi-core architectures, October 22-23, 2014, University Pierre et Marie Curie, Paris, France

Joël Falcou Membre du comité français de normalisation des langages C et C++ (JTC1/SC22/WG21) auprès de l'AFNOR depuis janvier 2011

Joël Falcou Co-chair of the C++NOW conference on C++ http://cppnow.org/

Joël Falcou Organizer and co-chair of the 2014 Workshop on Programming Models for SIMD/Vector Processing https://sites.google.com/site/wpmvp2014/home

Joël Falcou Membre fondateur du *User Group* C++ francophone (http://www.meetup.com/User-Group-Cpp-Francophone/)

Lionel Lacassagne reviewer for JRTIP and Eurasip Signal journals

Lionel Lacassagne co-chairman of the "Visual Scene Analysis on Hybrid Multicore" special session at DASIP 2014.

Lionel Lacassagne Program Committee member of DASIP 2014.

8.2. Teaching - Supervision - Juries

8.2.1. Teaching

Master : Christine Eisenbeis, coordinatrice du module “Optimisations et compilation” du M2 recherche NSI (“Nouveaux systèmes informatiques”) de l’université Paris-Sud 11. 3 heures de cours.

Master: M. Baboulin and J. Falcou teach in "Calcul Haute Performance" of M2 recherche NSI of University Paris Sud 11.

Lionel Lacassagne: Master 2 Research "Nouveau Systemes Informatique" (NSI) and Master 2 "Systemes Embarqués et Traitement de l’Information" (SETI). Computer Architecture: optimisations for multicore and SIMD processors.

Polytech 5th year: M. Baboulin and J. Falcou teach the "Parallel Computing" class.

8.2.2. Supervision

PhD in progress: Ian Masliah, Automatic code generation in high-performance computing numerical libraries, University Paris Sud 11, Supervisors: M. Baboulin and J. Falcou

PhD in progress: Antoine Tran Tan, Automatic task-based code generation by C++ meta-programming, University Paris-Sud 11, Supervisor: Joël Falcou

PhD in progress: Adrien Rémy, Solving dense linear systems on accelerated multicore architectures, University Paris Sud 11, Supervisors: M. Baboulin and B. Rozoy

PhD in progress: Yushan Wang, Numerical simulations of incompressible fluid flows on heterogeneous parallel architectures, University Paris Sud 11, Supervisors: M. Baboulin and O. Le Maître

PhD in progress: Lénaïc Bagnères, université Paris-Sud 11, supervisors: Cédric Bastoul and Christine Eisenbeis

PhD defended, december 2014: Alessandro Leite, université Paris-Sud 11, supervisors: Alba de Melo (university of Brazilia), Claude Tadonki (CRI, école des Mines de Paris), Christine Eisenbeis

PhD defended, september 2014, Michael Kruse, Polytopic memory layout optimization, université Paris-Sud 11, supervisor: Christine Eisenbeis

PhD in progress: Jason Lambert, Interactive Non-Destructive Testing algorithm on SIMD multicore processors and GPU. CEA List/Disc funding, université Paris-Sud 11, supervisor: Lionel Lacassagne

8.2.3. Juries

- Marc Baboulin, Thèse de Viviana Siless (08/07/2014)). Fonction exercée: président du jury
- Marc Baboulin, HDR de Christophe Denis (04/07/2014). Fonction exercée: rapporteur
- Marc Baboulin, Thèse de Jesus Camacho Rodriguez (25/09/2014). Fonction exercée: président du jury
- Marc Baboulin, HDR de Sylvie Boldo (06/10/2014). Fonction exercée: président du jury
- Marc Baboulin, Thèse de Philippe Théveny (31/10/2014). Fonction exercée: président du jury
- Marc Baboulin, Thèse de Vincent Reverdy (12/11/2014). Fonction exercée: examinateur
- Christine Eisenbeis, membre du jury de la thèse de Michael Kruse, "Lattice QCD Optimization and Polytopic Representations of Distributed Memory", vendredi 26 septembre 2014, université Paris-Sud

- Christine Eisenbeis, membre du jury de la thèse de Alessandro Ferreira Leite, "A User-Centered and Autonomic Multi-Cloud Architecture for High Performance Computing Applications", mardi 2 décembre 2014, université Paris-Sud
- Lionel Lacassagne, membre du jury de la thèse de Haixiong Ye, "Impact of High Level Transforms on High Level Synthesis: application to signal and image processing", ST Microelectronics funding, 20 mai 2014, université Paris-Sud

8.3. Popularization

Christine Eisenbeis est membre du conseil scientifique des programmes du centre d'Alembert, Centre Interdisciplinaire d'Étude de l'Évolution des Idées, des Sciences et des Techniques (CIEEIST), de l'université Paris-Sud. À ce titre, elle a fait partie du comité d'organisation du colloque "Recherche et démocratie", 21-22 mai 2014, Orsay, (http://www.centre-dalembert.u-psud.fr/archives-colloques/2014-recherche-scientifique-et-democratie/). Lors de l'animation "Livres au marché" de Malakoff, le 23 novembre 2014, elle a présenté, avec Jean-Pierre Archambault, le livre d'enseignement en terminale de l'ISN (Informatique et Sciences du Numérique) (principal auteur Gilles Dowek).

9. Bibliography

Major publications by the team in recent years

[1] M. BABOULIN, D. BECKER, J. DONGARRA. *A Parallel Tiled Solver for Dense Symmetric Indefinite Systems on Multicore Architectures*, in "Proceedings of IEEE International Parallel & Distributed Processing Symposium (IPDPS 2012)", 2012, pp. 14-24

[2] M. BABOULIN, S. DONFACK, J. DONGARRA, L. GRIGORI, A. RÉMY, S. TOMOV. *A class of communication-avoiding algorithms for solving general dense linear systems on CPU/GPU parallel machines*, in "International Conference on Computational Science (ICCS 2012)", Procedia Computer Science, Elsevier, 2012, vol. 9, pp. 17–26

[3] M. BABOULIN, J. DONGARRA, J. HERRMANN, S. TOMOV. *Accelerating linear system solutions using randomization techniques*, in "ACM Trans. Math. Softw.", 2013, vol. 39, n^{o} 2

[4] M. BABOULIN, S. GRATTON. *A contribution to the conditioning of the total least squares problem*, in "SIAM J. Matrix Anal. and Appl.", 2011, vol. 32, n^{o} 3, pp. 685–699

[5] M. BAHI, C. EISENBEIS. *Impact of Reverse Computing on Information Locality in Register Allocation for High Performance Computing*, in "International Journal of Parallel Programming", 2012, pp. 1–28

[6] D. BARTHOU, O. BRAND-FOISSAC, O. PENE, G. GROSDIDIER, R. DOLBEAU, C. EISENBEIS, M. KRUSE, K. PETROV, C. TADONKI. *Automated Code Generation for Lattice Quantum Chromodynamics and beyond*, in "Journal of Physics: Conference Series", 2014, vol. 510, 012005 p. , LPT-Orsay-13-142 [*DOI :* 10.1088/1742-6596/510/1/012005], http://hal.inria.fr/hal-00926513

[7] P. ESTERIE, J. FALCOU, M. GAUNARD, J.-T. LAPRESTÉ, L. LACASSAGNE. *The Numerical Template toolbox: A Modern C++ Design for Scientific Computing*, in "Journal of Parallel and Distributed Computing", July 2014 [*DOI :* 10.1016/J.JPDC.2014.07.002], https://hal.inria.fr/hal-01061305

[8] P. ESTERIE, M. GAUNARD, J. FALCOU, J.-T. LAPRESTÉ. *Exploiting Multimedia Extensions in C++: A Portable Approach*, in "Computing in Science & Engineering", 2012, vol. 14, n° 5, pp. 72–77

[9] A. FERREIRA LEITE. *A User-Centered and Autonomic Multi-Cloud Architecture for High Performance Computing Applications*, Paris-Sud XI ; Universidade de Brasília, December 2014, https://hal.inria.fr/tel-01097295

[10] G. FURSIN, Y. KASHNIKOV, A. W. MEMON, Z. CHAMSKI, O. TEMAM, M. NAMOLARU, E. YOM-TOV, B. MENDELSON, A. ZAKS, E. COURTOIS, F. BODIN, P. BARNARD, E. ASHTON, E. BONILLA, J. THOMSON, C. WILLIAMS, M. F. P. O'BOYLE. *Milepost GCC: Machine Learning Enabled Self-tuning Compiler*, in "International Journal of Parallel Programming", 2011, vol. 39, pp. 296-327, 10.1007/s10766-010-0161-2

[11] M. KRUSE. *Lattice QCD Optimization and Polytopic Representations of Distributed Memory*, Paris-Sud XI, September 2014, https://hal.inria.fr/tel-01078440

[12] S. TOMOV, J. DONGARRA, M. BABOULIN. *Towards dense linear algebra for hybrid GPU accelerated manycore systems*, in "Parallel Computing", 2010, vol. 36, n° 5&6, pp. 232–240

Publications of the year

Doctoral Dissertations and Habilitation Theses

[13] A. FERREIRA LEITE. *A User-Centered and Autonomic Multi-Cloud Architecture for High Performance Computing Applications*, Paris-Sud XI, December 2014, https://hal.inria.fr/tel-01097295

[14] M. KRUSE. *Lattice QCD Optimization and Polytopic Representations of Distributed Memory*, Paris-Sud XI, September 2014, https://hal.inria.fr/tel-01078440

Articles in International Peer-Reviewed Journals

[15] M. BABOULIN, D. BECKER, G. BOSILCA, A. DANALIS, J. DONGARRA. *An efficient distributed randomized algorithm for solving large dense symmetric indefinite linear systems*, in "Parallel Computing", July 2014, vol. 40, n° 7, pp. 213-223 [*DOI :* 10.1016/J.PARCO.2013.12.003], https://hal.inria.fr/hal-01024857

[16] M. BABOULIN, S. GRATTON, R. LACROIX, A. J. LAUB. *Statistical estimates for the conditioning of linear least squares problems*, in "Lecture notes in computer science", 2014, vol. 8384, pp. 124-133 [*DOI :* 10.1007/978-3-642-55224-3_13], https://hal.inria.fr/hal-00991710

[17] D. BARTHOU, O. BRAND-FOISSAC, O. PENE, G. GROSDIDIER, R. DOLBEAU, C. EISENBEIS, M. KRUSE, K. PETROV, C. TADONKI. *Automated Code Generation for Lattice Quantum Chromodynamics and beyond*, in "Journal of Physics: Conference Series", 2014, vol. 510, 012005 p. , LPT-Orsay-13-142 [*DOI :* 10.1088/1742-6596/510/1/012005], https://hal.inria.fr/hal-00926513

[18] P. ESTERIE, J. FALCOU, M. GAUNARD, J.-T. LAPRESTÉ, L. LACASSAGNE. *The Numerical Template toolbox: A Modern C++ Design for Scientific Computing*, in "Journal of Parallel and Distributed Computing", July 2014 [*DOI :* 10.1016/J.JPDC.2014.07.002], https://hal.inria.fr/hal-01061305

[19] G. FURSIN, R. MICELI, A. LOKHMOTOV, M. GERNDT, M. BABOULIN, A. D. MALONY, Z. CHAMSKI, D. NOVILLO, D. D. VENTO. *Collective mind: Towards practical and collaborative auto-tuning*, in "Scientific

Programming", July 2014, vol. 22, n^{o} 4, pp. 309-329 [*DOI :* 10.3233/SPR-140396], https://hal.inria.fr/hal-01054763

[20] A. ROMERO, L. LACASSAGNE, M. GOUIFFÈS, A. HASSAN ZAHRAEE. *Covariance tracking: architecture optimizations for embedded systems*, in "EURASIP Journal on Advances in Signal Processing", December 2014, 25 p. [*DOI :* 10.1186/1687-6180-2014-175], https://hal.inria.fr/hal-01094903

[21] M. SZYDLARSKI, P. ESTERIE, J. FALCOU, L. GRIGORI, R. STOMPOR. *Spherical harmonic transform on heterogeneous architectures using hybrid programming*, in "Concurrency and Computation Practice and Experience", March 2014, vol. 26, n^{o} 3, 28 p. [*DOI :* 10.1002/CPE.3038], https://hal.inria.fr/hal-01091256

International Conferences with Proceedings

[22] L. BAGNÈRES, C. BASTOUL. *Switchable Scheduling for Runtime Adaptation of Optimization*, in "Euro-Par 2014 Parallel Processing", Porto, Portugal, Lecture Notes in Computer Science, Springer International Publishing, August 2014, vol. 8632, pp. 222 - 233 [*DOI :* 10.1007/978-3-319-09873-9_19], https://hal.inria.fr/hal-01097200

[23] L. CABARET, L. LACASSAGNE. *What Is the World's Fastest Connected Component Labeling Algorithm?*, in "SiPS: IEEE International Workshop on Signal Processing Systems", Belfast, United Kingdom, IEEE, October 2014, 6 p. , https://hal.inria.fr/hal-01094905

[24] L. CABARET, L. LACASSAGNE, L. OUDNI. *A Review of World's Fastest Connected Component Labeling Algorithms: Speed and Energy Estimation*, in "International Conference on Design and Architectures for Signal and Image Processing", Madrid, Spain, October 2014, https://hal.inria.fr/hal-01081962

[25] A. FERREIRA LEITE, C. TADONKI, C. EISENBEIS, T. RAIOL, M. E. WALTER, A. C. ALVES DE MELO. *Excalibur: An Autonomic Cloud Architecture for Executing Parallel Applications*, in "Fourth International Workshop on Cloud Data and Platforms (CloudDP)", Amsterdam, Netherlands, April 2014 [*DOI :* 10.1145/2592784.2592786], https://hal-mines-paristech.archives-ouvertes.fr/hal-01087315

[26] L. LACASSAGNE, D. ETIEMBLE, A. HASSAN ZAHRAEE, A. DOMINGUEZ, P. VEZOLLE. *High Level Transforms for SIMD and Low-Level Computer Vision Algorithms*, in "Symposium on Principles and Practice of Parallel Programming / WPMVP", Orlando, Florida, United States, February 2014, 8 p. [*DOI :* 10.1145/2568058.2568067], https://hal.inria.fr/hal-01094906

[27] A. LEITE, C. TADONKI, C. EISENBEIS, A. DE MELO. *A Fine-grained Approach for Power Consumption Analysis and Prediction*, in "International Conference on Computational Science - ICCS", Cairns, Australia, June 2014 [*DOI :* 10.1016/J.PROCS.2014.05.211], https://hal.inria.fr/hal-01074959

[28] A. TRAN TAN, J. FALCOU, D. ETIEMBLE, H. KAISER. *Automatic Task-based Code Generation for High Performance Domain Specific Embedded Language*, in "HLPP 2014", Amsterdam, Netherlands, July 2014, https://hal.inria.fr/hal-01061423

[29] O. ZINENKO, C. BASTOUL, S. HUOT. *Manipulating Visualization, Not Codes*, in "International Workshop on Polyhedral Compilation Techniques (IMPACT)", Amsterdam, Netherlands, January 2015, 8 p. , https://hal.inria.fr/hal-01100974

Scientific Books (or Scientific Book chapters)

[30] A. RÉMY, M. BABOULIN, M. SOSONKINA, B. ROZOY. *Locality Optimization on a NUMA Architecture for Hybrid LU Factorization*, in "Advances in Parallel Computing", 2014, vol. 25, pp. 153-162 [*DOI :* 10.3233/978-1-61499-381-0-153], https://hal.inria.fr/hal-00987284

Research Reports

[31] M. BABOULIN, J. DONGARRA, R. LACROIX. *Computing least squares condition numbers on hybrid multicore/GPU systems*, February 2014, n° RR-8479, https://hal.inria.fr/hal-00947204

[32] M. BABOULIN, J. FALCOU, I. MASLIAH. *Towards an automatic generation of dense linear algebra solvers on parallel architectures*, Université Paris-Sud, October 2014, n° RR-8615, 20 p. , https://hal.inria.fr/hal-01075663

[33] M. BABOULIN, X. S. LI, F.-H. ROUET. *Using Random Butterfly Transformations to Avoid Pivoting in Sparse Direct Methods*, Inria, February 2014, n° RR-8481, Also appeared as Lapack Working Note 285, https://hal.inria.fr/hal-00950612

[34] G. FURSIN, C. DUBACH. *Experience report: community-driven reviewing and validation of publications*, June 2014, https://hal.inria.fr/hal-01006563

[35] A. RÉMY, M. BABOULIN, M. SOSONKINA, B. ROZOY. *Locality optimization on a NUMA architecture for hybrid LU factorization*, March 2014, n° RR-8497, https://hal.inria.fr/hal-00957673

Other Publications

[36] D. BARTHOU, O. BRAND-FOISSAC, R. DOLBEAU, G. GROSDIDIER, C. EISENBEIS, M. KRUSE, O. PENE, K. PETROV, C. TADONKI. *Automated Code Generation for Lattice Quantum Chromodynamics and beyond*, January 2014, https://hal.archives-ouvertes.fr/hal-00930288

[37] J. LAMBERT, H. CHOUH, G. ROUGERON, V. BERGEAUD, S. CHATILLON, L. LACASSAGNE, J.-C. IEHL, J.-P. FARRUGIA, V. OSTROMOUKHOV. *Interactive Ultrasonic Field Simulation For Non-Destructive Testing*, June 2014, vol. 33, n° 2, 25th Eurographics Symposium on Rendering, https://hal.inria.fr/hal-01093294

[38] J. LAMBERT, G. ROUGERON, L. LACASSAGNE. *Calcul de champ ultrasonore interactif pour le contrôle non destructif*, May 2014, Les Journées COFREND, https://hal.inria.fr/hal-01093131

References in notes

[39] *The HiPEAC vision on high-performance and embedded architecture and compilation (2012-2020)*, 2012, http://www.hipeac.net/roadmap

[40] *European Union Framework Program 6 MILEPOST project No 035307 (MachIne Learning for Embedded PrOgramS opTimization)*, http://cordis.europa.eu/project/rcn/79763_en.html

[41] *PRACE: Partnership for Advanced Computing in Europe*, http://www.prace-project.eu

[42] AMD. *AMD Core Math Library*, http://developer.amd.com/libraries/acml/

[43] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. D. CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, D. SORENSEN. *LAPACK Users' Guide*, SIAM, 1999, Third edition

[44] K. ANEJA, F. LAGUZET, L. LACASSAGNE, A. MERIGOT. *Video rate image segmentation by means of region splitting and merging*, in "IEEE International Conference on Signal and Image Processing Applications (ICSIPA)", 2009

[45] M. ARIOLI, M. BABOULIN, S. GRATTON. *A partial condition number for linear least-squares problems*, in "SIAM J. Matrix Anal. and Appl.", 2007, vol. 29, n^{o} 2, pp. 413–433

[46] K. ASANOVIC. *The landscape of parallel computing research: a view from Berkeley*, Electrical Engineering and Computer Sciences, University of California at Berkeley, December 2006, n^{o} UCB/EECS-2006-183, http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.pdf

[47] A. AVRON, P. MAYMOUNKOV, S. TOLEDO. *Blendenpick: Supercharging LAPACK's least-squares solvers*, in "SIAM J. Sci. Comput.", 2010, vol. 32, pp. 1217–1236

[48] M. BABOULIN, D. BECKER, G. BOSILCA, A. DANALIS, J. DONGARRA. *An efficient distributed randomized algorithm for solving large dense symmetric indefinite linear systems*, in "Parallel Computing", 2014, vol. 40, n^{o} 7, pp. 213–223

[49] M. BABOULIN, D. BECKER, J. DONGARRA. *A Parallel Tiled Solver for Dense Symmetric Indefinite Systems on Multicore Architectures*, in "Proceedings of IEEE International Parallel & Distributed Processing Symposium (IPDPS 2012)", 2012, pp. 14-24

[50] M. BABOULIN, A. BUTTARI, J. DONGARRA, J. KURZAK, J. LANGOU, J. LANGOU, P. LUSZCZEK, S. TOMOV. *Accelerating scientific computations with mixed precision algorithms*, in "Computer Physics Communications", 2009, vol. 180, n^{o} 12, pp. 2526–2533

[51] M. BABOULIN, S. DONFACK, J. DONGARRA, L. GRIGORI, A. RÉMY, S. TOMOV. *A class of communication-avoiding algorithms for solving general dense linear systems on CPU/GPU parallel machines*, in "International Conference on Computational Science (ICCS 2012)", Procedia Computer Science, Elsevier, 2012, vol. 9, pp. 17–26

[52] M. BABOULIN, J. DONGARRA, J. DEMMEL, S. TOMOV, V. VOLKOV. *Enhancing the performance of dense linear algebra solvers on GPUs in the MAGMA project*, November 15, 2008, http://www.lri.fr/~baboulin/SC08.pdf

[53] M. BABOULIN, J. DONGARRA, S. GRATTON, J. LANGOU. *Computing the conditioning of the components of a linear least squares solution*, in "Numerical Linear Algebra with Applications", 2009, vol. 16, n^{o} 7, pp. 517–533

[54] M. BABOULIN, J. DONGARRA, J. HERRMANN, S. TOMOV. *Accelerating linear system solutions using randomization techniques*, in "ACM Trans. Math. Softw.", 2013, vol. 39, n^{o} 2

[55] M. BABOULIN, J. DONGARRA, R. LACROIX. *Computing least squares condition numbers on hybrid multicore/GPU systems*, in "Proceedings of the International Conference of Applied Mathematics, Modeling and Computational Science (AMMCS 2013)", 2013

[56] M. BABOULIN, J. DONGARRA, S. TOMOV. *Some Issues in Dense Linear Algebra for Multicore and Special Purpose Architectures*, in "9th International Workshop on State-of-the-Art in Scientific and Parallel Computing (PARA'08)", Lecture Notes in Computer Science, Springer-Verlag, 2008, vol. 6126-6127

[57] M. BABOULIN, S. GRATTON. *A contribution to the conditioning of the total least squares problem*, in "SIAM J. Matrix Anal. and Appl.", 2011, vol. 32, n^{o} 3, pp. 685–699

[58] M. BABOULIN, S. GRATTON, R. LACROIX, A. J. LAUB. *Statistical estimates for the conditioning of linear least squares problems*, in "10th International Conference on Parallel Processing and Applied Mathematics (PPAM 2013)", Heidelberg, R. WYRZYKOWSKI (editor), Lecture Notes in Computer Science, Springer-Verlag, 2014, vol. 8384, pp. 124-133

[59] M. BABOULIN, X. S. LI, F.-H. ROUET. *Using Random Butterfly Transformations to Avoid Pivoting in Sparse Direct Methods*, in "Proceedings of VECPAR 2014", 2014

[60] J. C. BAEZ, M. STAY. *Algorithmic thermodynamics*, in "Mathematical Structures in Computer Science", 2012, vol. 22, n^{o} 5, pp. 771–787, http://dx.doi.org/10.1017/S0960129511000521

[61] M. BAHI, C. EISENBEIS. *Spatial complexity of reversibly computable DAG*, in "Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems", ACM, 2009, pp. 47–56

[62] M. BAHI, C. EISENBEIS. *Impact of Reverse Computing on Information Locality in Register Allocation for High Performance Computing*, in "International Journal of Parallel Programming", 2012, pp. 1–28

[63] D. BARTHOU, O. BRAND-FOISSAC, O. PENE, G. GROSDIDIER, R. DOLBEAU, C. EISENBEIS, M. KRUSE, K. PETROV, C. TADONKI. *Automated Code Generation for Lattice Quantum Chromodynamics and beyond*, in "Journal of Physics: Conference Series", 2014, vol. 510, 012005, LPT-Orsay-13-142 [*DOI :* 10.1088/1742-6596/510/1/012005], http://hal.inria.fr/hal-00926513

[64] D. BARTHOU, G. GROSDIDIER, C. EISENBEIS, P. GUICHON, M. KRUSE, O. PENE, K. PETROV, C. TADONKI. *PetaQCD: En Route for the automatic code generation for lattice QCD*, in "Proceedings of the 29th International Symposium on Lattice field theory (Lattice 2011)", 2011, vol. 2011

[65] P. BASU, S. WILLIAMS, B. V. STRAALEN, A. VENKAT, L. OLIKER, M. HALL. *Compiler Generation and Autotuning of Communication-Avoiding Operators for Geometric Multigrid*, in "High Performance Computing Conference (HiPC)", december 2013

[66] D. BECKER, M. BABOULIN, J. DONGARRA. *Reducing the amount of pivoting in symmetric indefinite systems*, in "9th International Conference on Parallel Processing and Applied Mathematics (PPAM 2011)", Heidelberg, R. WYRZYKOWSKI (editor), Lecture Notes in Computer Science, Springer-Verlag, 2012, vol. 7203, pp. 133–142

[67] T. BETCKE, N. J. HIGHAM, V. MEHRMANN, C. SCHRÖDER, F. TISSEUR. *NLEVP: A Collection of Nonlinear Eigenvalue Problems*, in "ACM Trans. Math. Software", February 2013, vol. 39, n^{o} 2, pp. 7:1-7:28 [*DOI :* 0.1145/2427023.2427024]

[68] L. BLACKFORD, J. CHOI, A. CLEARY, E. D'AZEVEDO, J. DEMMEL, I. DHILLON, J. DONGARRA, S. HAMMARLING, G. HENRY, A. PETITET, K. STANLEY, D. WALKER, R. WHALEY. *ScaLAPACK Users' Guide*, SIAM, 1997, pp. 58–60

[69] BLAZE. *The Blaze Library*, 2014, https://code.google.com/p/blaze-lib/

[70] G. BRADSKI. *The OpenCV Library*, in "Dr. Dobb's Journal of Software Tools", 2000

[71] L. CABARET, L. LACASSAGNE. *A Review of Worlds Fastest Connected Component Labeling Algorithms : Speed and Energy Estimation*, in "IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)", 2014, pp. 1-8

[72] L. CABARET, L. LACASSAGNE. *What is the world fastest Connected Component Labeling Algorithm ?*, in "IEEE International Workshop on Signal Processing Systems (SiPS)", 2014, pp. 1-6

[73] V. G. CERF. *Where is the science in computer science?*, in "Communications of the ACM", 2012, vol. 55, n^{o} 10, pp. 5-5

[74] M. O. CHEEMA, L. LACASSAGNE, O. HAMMAMI. *System-Platforms-Based SystemC TLM Design of Image Processing Chains for Embedded Applications*, in "EURASIP Journal on Embedded Systems", 2007, pp. 1-14 [*DOI :* 10.1155/2007/71043]

[75] P. COURBIN, A. PÉDRON, T. SAIDANI, L. LACASSAGNE. *Parallélisation d'opérateurs de TI: multi-coeurs, Cell ou GPU ?*, in "GRETSI", 2009

[76] K. CZARNECKI, U. W. EISENECKER, R. GLÜCK, D. VANDEVOORDE, T. L. VELDHUIZEN. *Generative Programming and Active Libraries*, in "Generic Programming", 1998, pp. 25-39

[77] P. I. DAVIES, N. J. HIGHAM. *Numerically Stable Generation of Correlation Matrices and their Factors*, in "BIT", 2000, vol. 40, n^{o} 4, pp. 640-651

[78] J. W. DEMMEL, L. GRIGORI, M. HOEMMEN, J. LANGOU. *Communication-optimal parallel and sequential QR and LU factorizations*, in "SIAM Journal on Scientific Computing", 2012, vol. 34, n^{o} 1, pp. 206–239

[79] J. W. DEMMEL, A. MCKENNEY. *A Test Matrix Generation Suite*, Mathematics and Computer Science Division, Argonne National LaboratoryArgonne, IL, USA, March 1989, n^{o} MCS-P69-0389, 16 p. , LAPACK Working Note 9

[80] J. DONGARRA ET.AL.. *The International Exascale Software Project roadmap*, in "Int. J. High Perform. Comput. Appl.", February 2011, vol. 25, n^{o} 1, pp. 3–60, http://dx.doi.org/10.1177/1094342010391989

[81] A. ELOUARDI, S. BOUAZIZ, A. DUPRET, L. LACASSAGNE, J. KLEIN, R. REYNAUD. *A smart sensor based vision system: implementation and evaluation*, in "Journal of Applied Physics", 2006, vol. 39, pp. 1694-1705 [*DOI :* 10.1088/0022-3727/39/8/033]

[82] A. ELOUARDI, S. BOUAZIZ, A. DUPRET, L. LACASSAGNE, J. KLEIN, R. REYNAUD. *A Smart Architecture for Low-Level Image Computing*, in "International Journal of Computer Sciences and Application", 2008, vol. 5,3, pp. 1-19

[83] P. ESTERIE, J. FALCOU, M. GAUNARD, J.-T. LAPRESTÉ, L. LACASSAGNE. *The numerical template toolbox: A modern C++ design for scientific computing*, in "Journal of Parallel and Distributed Computing", 2014

[84] P. ESTERIE, M. GAUNARD, J. FALCOU, J.-T. LAPRESTÉ. *Exploiting Multimedia Extensions in C++: A Portable Approach*, in "Computing in Science & Engineering", 2012, vol. 14, n^{o} 5, pp. 72–77

[85] P. ESTÉRIE, M. GAUNARD, J. FALCOU. *A proposal to add single instruction multiple data computation to the standard library*, in "N3561", 2013

[86] D. ETIEMBLE, S. PISKORSKI, L. LACASSAGNE. *Performance evaluation of Altera C2H compiler on image processing benchmarks*, in "TCHA: Workshop on Tools And Compiler for Hardware Acceleration", 2006

[87] J. FALCOU, L. LACASSAGNE, S. SCHAETZ. *Cell MPI: Mastering the Cell Broadband Engine architecture through a Boost based parallel communication library*, in "Boost Conference", 2011

[88] J. FALCOU, T. SAIDANI, L. LACASSAGNE, D. ETIEMBLE. *Programmation par squelettes algorithmiques pour le processeur Cell*, in "SYMPA", 2008

[89] J. FALCOU, J. SÉROT, L. PECH, J.-T. LAPRESTÉ. *Meta-programming applied to automatic SMP parallelization of linear algebra code*, in "Euro-Par 2008–Parallel Processing", Springer Berlin Heidelberg, 2008, pp. 729–738

[90] G. FURSIN, C. DUBACH. *Experience report: community-driven reviewing and validation of publications*, in "Proceedings of the 1st Workshop on Reproducible Research Methodologies and New Publication Models in Computer Engineering (ACM SIGPLAN TRUST'14)", ACM, 2014, http://dx.doi.org/10.1145/2618137.2618142

[91] G. FURSIN. *Collective Tuning Initiative: automating and accelerating development and optimization of computing systems*, in "Proceedings of the GCC Developers' Summit", June 2009

[92] G. FURSIN, Y. KASHNIKOV, A. W. MEMON, Z. CHAMSKI, O. TEMAM, M. NAMOLARU, E. YOM-TOV, B. MENDELSON, A. ZAKS, E. COURTOIS, F. BODIN, P. BARNARD, E. ASHTON, E. BONILLA, J. THOMSON, C. WILLIAMS, M. F. P. O'BOYLE. *Milepost GCC: Machine Learning Enabled Self-tuning Compiler*, in "International Journal of Parallel Programming", 2011, vol. 39, pp. 296-327, 10.1007/s10766-010-0161-2

[93] G. FURSIN, R. MICELI, A. LOKHMOTOV, M. GERNDT, M. BABOULIN, A. D. MALONY, Z. CHAMSKI, D. NOVILLO, D. D. VENTO. *Collective Mind: towards practical and collaborative auto-tuning*, in "Special issue on Automatic Performance Tuning for HPC Architectures, Scientific Programming Journal", 2014

[94] M. GOUIFFÈS, F. LAGUZET, L. LACASSAGNE. *Color Connectedness Degree For Mean-Shift Tracking*, in "IEEE International Conference on Pattern Recognition (ICPR)", 2010

[95] M. GOUIFFÈS, F. LAGUZET, L. LACASSAGNE. *Projection Histogram For Mean-Shift Tracking*, in "IEEE International Conference on Image Processing (ICIP)", 2010

[96] C. GRANA, D. BORGHESANI, R. CUCCHIARA. *Connected Component Labeling Techniques on Modern Architectures*, in "ICIAP", IEEE, 2009, pp. 816-824

[97] L. GRIGORI, J. DEMMEL, H. XIANG. *CALU: a communication optimal LU factorization algorithm*, in "SIAM J. Matrix Anal. and Appl.", 2011, vol. 32, pp. 1317-1350

[98] M. GU, S. C. EISENSTAT. *Efficient Algorithms for Computing a Strong Rank-revealing QR Factorization*, in "SIAM Journal on Scientific Computing", July 1996, vol. 17, n° 4, pp. 848–869, http://dx.doi.org/10.1137/0917055

[99] S. GUELTON, J. FALCOU, P. BRUNET. *Exploring the vectorization of python constructs using pythran and boost SIMD*, in "Proceedings of the 2014 Workshop on Workshop on programming models for SIMD/Vector processing", ACM, 2014, pp. 79–86

[100] G. GUENNEBAUD, B. JACOB. *Eigen v3*, 2010, http://eigen.tuxfamily.org

[101] N. HALKO, P. G. MARTINSSON, J. A. TROPP. *Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions*, in "SIAM Review", 2011, vol. 53, pp. 217–288

[102] C. HARRIS, M. STEPHENS. *A combined corner and edge detector*, in "4th ALVEY Vision Conference", Editions Hermes, Paris, 1988

[103] L. HE, Y. CHAO, K. SUZUKI. *A run-based two-scan labeling algorithm*, in "ICIAR", LNCS 4633, 2007, pp. 131-142

[104] R. M. HEIBERGER. *Algorithm AS 127: Generation of Random Orthogonal Matrices*, in "J. Roy. Statist. Soc. Ser. C (Applied Statistics)", 1978, vol. 27, n° 2, pp. 199-206

[105] N. J. HIGHAM. *J-Orthogonal Matrices: Properties and Generation*, in "SIAM Rev.", September 2003, vol. 45, n° 3, pp. 504-519 [*DOI :* 10.1137/S0036144502414930]

[106] G. E. HINTON, S. OSINDERO. *A fast learning algorithm for deep belief nets*, in "Neural Computation", 2006, vol. 18

[107] S. HOROWITZ, T. PAVLIDIS. *Picture segmentation by a tree traversal algorithm*, in "Journal of the ACM", 1976, vol. 23, pp. 368-388

[108] T. IKEGAMI, T. SAKURAI, U. NAGASHIMA. *A filter diagonalization for generalized eigenvalue problems based on the Sakurai-Sugiura projection method*, in "Journal of Computational and Applied Mathematics", 2010, vol. 233, n° 8, pp. 1927–1936

[109] INTEL. *Math Kernel Library*, http://developer.intel.com/software/products/mkl/

[110] V. JIMENEZ, I. GELADO, L. VILANOVA, M. GIL, G. FURSIN, N. NAVARRO. *Predictive runtime code scheduling for heterogeneous architectures*, in "Proceedings of the International Conference on High Performance Embedded Architectures & Compilers (HiPEAC 2009)", January 2009

[111] C. S. KENNEY, A. J. LAUB. *Small-sample statistical condition estimates for general matrix functions*, in "SIAM J. Sci. Comput.", 1994, vol. 15, pp. 36–61

[112] A. KHABOU, J. DEMMEL, L. GRIGORI, M. GU. *LU Factorization with Panel Rank Revealing Pivoting and Its Communication Avoiding Version*, in "SIAM Journal on Matrix Analysis and Applications", 2013, vol. 34, n^{O} 3, pp. 1401-1429, http://epubs.siam.org/doi/abs/10.1137/120863691

[113] M. KRUSE. *Lattice QCD Optimization and Polytopic Representations of Distributed Memory*, Université Paris-Sud 11, September, 26th 2014

[114] T. KUNLIN, L. LACASSAGNE, A. MÉRIGOT. *A Fast image segmentation scheme*, in "International Conference on Information and Communication Technologies", IEEE, 2004

[115] L. LACASSAGNE, D. ETIEMBLE, A. HASSAN ZAHRAEE, A. DOMINGUEZ, P. VEZOLLE. *High Level Transforms for SIMD and low-level computer vision algorithms*, in "ACM Workshop on Programming Models for SIMD/Vector Processing (PPoPP)", 2014, pp. 49-56

[116] L. LACASSAGNE, D. ETIEMBLE, S. KABLIA. *16-bit Floating Point Instructions for embedded Multimedia Applications*, in "CAMP: Computer Architecture and Machine Perception", IEEE, 2005

[117] L. LACASSAGNE, D. ETIEMBLE. *16-bit floating point operations for low-end and high-end embedded processors*, in "ODES: Optimizations for DSP and Embedded Systems", IEEE/ACM, 2005

[118] L. LACASSAGNE, A. MANZANERA, J. DENOULET, A. MÉRIGOT. *High Performance Motion Detection: Some trends toward new embedded architectures for vision systems*, in "Journal of Real Time Image Processing", october 2008, pp. 127-148 [*DOI :* 10.1007/s11554-008-0096-7]

[119] L. LACASSAGNE, A. B. ZAVIDOVIQUE. *Light Speed Labeling for RISC architectures*, in "IEEE International Conference on Image Analysis and Processing (ICIP)", 2009

[120] L. LACASSAGNE, B. ZAVIDOVIQUE. *Light Speed Labeling: efficient connected component labeling on RISC architectures*, in "Journal of Real-Time Image Processing", 2011, vol. 6, n^{O} 2, pp. 117-135

[121] F. LAGUZET, M. GOUIFFÈS, L. LACASSAGNE. *Automatic color space switching for robust tracking*, in "IEEE International Conference on Signal and Image Processing Applications (ICSIPA)", 2011

[122] F. LAGUZET, A. ROMERO, M. GOUIFFÈS, L. LACASSAGNE, D. ETIEMBLE. *Color tracking with contextual switching: Real-time implementation on CPU*, in "Journal of Real-Time Image Processing", 2013, pp. 1-18

[123] J. LAMBERT, L. LACASSAGNE, G. ROUGERON, S. L. BERRE, S. CHATILLON. *High Performance simulation of ultrasonic fields for Non Destructive Testing*, in "International Symposium in Nuclear Application and Monte-Carlo", 2013

[124] J. LAMBERT, A. PÉDRON, G. GENS, F. BIMBARD, L. LACASSAGNE, E. IAKOVLEVA, S. L. BERRE. *Analysis of multicore CPU and GPU toward parallelization of Total Focusing Method ultrasound reconstruction*, in "IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)", 2012

[125] J. LAMBERT, G. ROUGERON, L. LACASSAGNE, S. CHATILLON. *A fast ultrasonic simulation tool based on massively parallel implementations*, in "Review of Progress of Quantitative Nondestructive Evaluation", 2013

[126] Q. LE, M. RANZATO, R. MONGA, M. DEVIN, K. CHEN, G. CORRADO, J. DEAN, A. NG. *Building high-level features using large scale unsupervised learning*, in "International Conference in Machine Learning", 2012

[127] W. LEDERMANN, C. ALEXANDER, D. LEDERMANN. *Random Orthogonal Matrix Simulation*, in "Linear Algebra Appl.", 2011, vol. 434, n^{o} 6, pp. 1444-1467 [*DOI :* 10.1016/J.LAA.2010.10.023]

[128] A. LEITE, C. TADONKI, C. EISENBEIS, A. DE MELO. *A Fine-grained Approach for Power Consumption Analysis and Prediction*, in "Procedia Computer Science", 2014, vol. 29, pp. 2260–2271

[129] S. LIU, C. EISENBEIS, J.-L. GAUDIOT. *A theoretical framework for value prediction in parallel systems*, in "Parallel Processing (ICPP), 2010 39th International Conference on", IEEE, 2010, pp. 11–20

[130] M. W. MAHONEY. *Randomized algorithms for matrices and data*, in "Foundations and Trends in Machine Learning", 2011, vol. 3, n^{o} 2, pp. 123–224

[131] D. MENARD, R. SERIZEL, R. ROCHER, O. SENTIEYS. *Accuracy Constraint Determination in Fixed-Point System Design*, in "Journal on Embedded Systems (JES),", 2008, vol. 2008, pp. 1-12 [*DOI :* 10.1155/2008/242584]

[132] P. MONASSE, F. GUICHARD. *Fast computation of contrast-onvariant image representation*, in "Transaction on", 2000, vol. 9,5, pp. 860-872

[133] S. MOUFAWAD. *Demmel type communication-avoiding generalized minimal residual method (CA-GMRES) on multicore hardwares: an application in QCD*, American university of BeirutBeirut, Libanon, june 2011, defended on 2010, June 10th

[134] M. ODERSKY. *An Overview of the SCALA Programming Language*, EPFL Lausanne, Switzerland, 2004, n^{o} IC/2004/64

[135] D. S. PARKER. *Random Butterfly Transformations with Applications in Computational Linear Algebra*, Computer Science Department, UCLA, 1995, n^{o} CSD-950023

[136] D. PETCU. *Consuming Resources and Services from Multiple Clouds*, in "Journal of Grid Computing", 2014, pp. 1–25

[137] M. PHARR, W. R. MARK. *ISPC: A SPMD Compiler for High-Performance CPU Programming*, in "Innovative Parallel Computing (InPar)", 2012

[138] S. PISKORSKI, L. LACASSAGNE, D. ETIEMBLE. *IPLG: un outil pour la fusion d'opérateurs en Traitement d'Images*, in "SYMPA", 2009

[139] S. PISKORSKI, L. LACASSAGNE, M. KIEFFER, D. ETIEMBLE. *Efficient floating point interval processing for embedded systems and applications*, in "SCAN - International Symposium of Scientific computing, Computer Arithmetic and Validated Numerics", 2006, 2006 p.

[140] S. POP, A. COHEN, C. BASTOUL, S. GIRBAL, G. A. SILBER, N. VASILACHE. *GRAPHITE: Loop optimizations based on the polyhedral model for GCC*, in "Proc. of the 4th GCC Developper's Summit", June 2006, pp. 179–198

[141] A. PÉDRON, L. LACASSAGNE, V. BARBILLON, F. BIMBARD, G. ROUGERON, S. L. BERRE. *Performance analysis of an ultrasound reconstruction algorithm for non destructuve testing*, in "IEEE International Conference on Parallel Computing (ParCo)", 2011

[142] A. PÉDRON, L. LACASSAGNE, F. BIMBARD, S. L. BERRE. *Parallelization of an ultrasound reconstruction algorithm for non destructive testing on multicore CPU and GPU*, in "IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)", 2011

[143] A. ROMERO, M. GOUIFFÈS, L. LACASSAGNE. *Feature Points tracking adaptative to Saturation*, in "IEEE International Conference on Signal and Image Processing Applications (ICSIPA)", 2011

[144] A. ROMERO, M. GOUIFFÈS, L. LACASSAGNE. *Covariance Descriptor Multiple Object Tracking and Re-Identification with Colorspace Evaluation*, in "IEEE ACCV - Workshop on Detection and Tracking in Challenging Environnements", 2012

[145] A. ROMERO, M. GOUIFFÈS, L. LACASSAGNE. *Enhanced Local Binary Covariance Matrices (ELBCM) for texture analysis and object tracking*, in "ACM International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications", 2013

[146] A. ROMERO, L. LACASSAGNE, M. GOUIFFÈS. *Real-time covariance tracking algorithm for embedded systems*, in "IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)", 2013

[147] A. ROSENFELD, J. PLATZ. *Sequential operator in digital pictures processing*, in "Journal of ACM", 1966, vol. 13,4, pp. 471-494

[148] A. RÉMY, M. BABOULIN, M. SOSONKINA, B. ROZOY. *Locality optimization on a NUMA architecture for hybrid LU factorization*, in "International Conference on Parallel Computing (PARCO 2013)", Advances in Parallel Computing, IOS Press, 2014, vol. 25, pp. 153-162

[149] T. SAIDANI, J. FALCOU, C. TADONKI, L. LACASSAGNE, D. ETIEMBLE. *Algorithmic Skeletons within an Embedded Domain Specific Language for the Cell Processor*, in "Parallel Architectures and Compilation Techniques, PACT", 2009, pp. 67-76

[150] T. SAIDANI, L. LACASSAGNE, S. BOUAZIZ, T. M. KHAN. *Parallelization Strategies for the Points of Interests Algorithm on the Cell Processor*, in "Lecture Notes in Computer Science", Springer, 2007, pp. 104-112 [*DOI :* 10.1007/978-3-540-74742-0]

[151] T. SAIDANI, S. PISKORSKI, L. LACASSAGNE, S. BOUAZIZ. *Parallelization Schemes for Memory Optimization on the Cell Processor: A Case Study of Image Processing Algorithm*, in "PACT-MEDEA", 2007, pp. 15-19

[152] C. SANDERSON. *Armadillo: An open source C++ linear algebra library for fast prototyping and computationally intensive experiments*, in "Report Version", 2010, vol. 2

[153] J. SIEK, L.-Q. LEE, A. LUMSDAINE. *Boost Random Number Library*, June 2000, http://www.boost.org/libs/graph/

[154] D. SPINELLIS. *Notable design patterns for domain-specific languages*, in "Journal of Systems and Software", 2001, vol. 56, n° 1, pp. 91 - 99 [*DOI :* 10.1016/S0164-1212(00)00089-3], http://www.sciencedirect.com/science/article/pii/S0164121200000893

[155] G. W. STEWART. *The Efficient Generation of Random Orthogonal Matrices With an Application to Condition Estimators*, in "SIAM J. Numer. Anal.", 1980, vol. 17, n° 3, pp. 403-409

[156] A. K. SUJEETH, A. GIBBONS, K. J. BROWN, H. LEE, T. ROMPF, M. ODERSKY, K. OLUKOTUN. *Forge: Generating a High Performance DSL Implementation from a Declarative Specification*, in "12th International Conference on Generative Programming: Concepts and Experiences", 2013

[157] A. K. SUJEETH, T. ROMPF, K. J. BROWN, H. LEE, H. CHAFI, V. POPIC, M. WU, A. PROKOPEC, V. JOVANOVIC, M. ODERSKY, K. OLUKOTUN. *Composition and Reuse with Compiled Domain-Specific Languages*, in "ECOOP'13: European Conference on Object-Oriented Programming", 2013

[158] V. SUNDRIYAL, M. SOSONKINA, A. GAENKO, Z. ZHANG. *Energy saving strategies for parallel applications with point-to-point communication phases*, in "Journal of Parallel and Distributed Computing", 2013 [*DOI :* 10.1016/J.JPDC.2013.03.006]

[159] V. SUNDRIYAL, M. SOSONKINA, Z. ZHANG. *Automatic runtime frequency-scaling system for energy savings in parallel applications*, in "The Journal of Supercomputing", 2014, vol. 68, n° 2, pp. 777–797

[160] K. SUZUKI, I. HORIBA, N. SUGIE. *Linear-time connected component labeling based on sequential local operations*, in "Computer Vision and Image Understanding", january 2003, vol. 89, n° 1, pp. 1-23 [*DOI :* 10.1016/S1077-3142(02)00030-9]

[161] H. TABIA, M. GOUIFFÈS, L. LACASSAGNE. *Motion histogram quantification for human action recognition*, in "IEEE International Conference on Pattern Recognition (ICPR)", 2012

[162] H. TABIA, M. GOUIFFÈS, L. LACASSAGNE. *Motion modeling for abnormal event detection in crowd scenes*, in "IEEE International Conference on Pattern Recognition (ISCIVC)", 2012

[163] C. TADONKI, L. LACASSAGNE, T. SAÏDANI, J. FALCOU, K. HAMIDOUCHE. *The Harris algorithm revisited on the Cell processor*, in "International Workshop on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART)", 2010

[164] S. TOMOV, J. DONGARRA, M. BABOULIN. *Towards dense linear algebra for hybrid GPU accelerated manycore systems*, in "Parallel Computing", 2010, vol. 36, n° 5&6, pp. 232–240

[165] UNIVERSITY OF TENNESSEE. *PLASMA Users' Guide, Parallel Linear Algebra Software for Multicore Architectures, Version 2.3*, 2010

[166] T. L. VELDHUIZEN. *Active Libraries and Universal Languages*, Indiana University Computer Science, May 2004, http://www.ubietylab.net/ubigraph/content/Papers/pdf/VeldhuizenThesis.pdf

[167] H. WANG, H. ANDRADE, B. GEDIK, K.-L. WU. *A Code Generation Approach for Auto-Vectorization in the Spade Compiler*, in "LCPC'09", 2009, pp. 383-390

[168] Y. WANG, M. BABOULIN, J. DONGARRA, J. FALCOU, Y. FRAIGNEAU, O. L. MAÎTRE. *A parallel solver for incompressible fluid flows*, in "International Conference on Computational Science (ICCS 2013)", Procedia Computer Science, Elsevier, 2013, vol. 18, pp. 439–448

[169] Y. WANG, M. BABOULIN, K. RUPP, O. LE MAÎTRE, Y. FRAIGNEAU. *Solving 3D Incompressible Navier-Stokes Equations on Hybrid CPU/GPU Systems*, in "Proceedings of the High Performance Computing Symposium", San Diego, CA, USA, HPC '14, Society for Computer Simulation International, 2014, pp. 12:1–12:8, http://dl.acm.org/citation.cfm?id=2663510.2663522

[170] H. YE, L. LACASSAGNE, D. ETIEMBLE, L. CABARET, J. FALCOU, O. FLORENT. *Impact of High Level Transforms on High Level Synthesis for motion detection algorithm*, in "IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)", 2012, pp. 1-8

[171] H. YE, L. LACASSAGNE, J. FALCOU, D. ETIEMBLE, L. CABARET, O. FLORENT. *High Level Transforms to reduce energy consumption of signal and image processing operators*, in "IEEE International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)", 2013, pp. 247-254