Activity Report 2015

Team POSTALE

Performance Optimization by Software Transformation and Algorithms & Librairies Enhancement

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

RESEARCH CENTER
Saclay - Île-de-France

THEME
Architecture, Languages and Compilation

Table of contents

Team POSTALE

Creation of the Team: 2014 January 01, end of the Team: 2015 December 31

Keywords:

Computer Science and Digital Science:

1. - Architectures, systems and networks
1.1.1. - Multicore
1.1.2. - Hardware accelerators (GPGPU, FPGA, etc.)
2.2. - Compilation
2.2.4. - Parallel architectures
6.2. - Scientific Computing, Numerical Analysis & Optimization
6.2.5. - Numerical Linear Algebra
6.2.7. - High performance computing

Other Research Topics and Application Domains:

6.1.1. - Software engineering

1. Members

Research Scientist

Christine Eisenbeis [Inria, Senior Researcher]

Faculty Members

Marc Baboulin [Team leader, Univ. Paris XI, Professor, HdR]
Joël Falcou [Univ. Paris XI, Associate Professor, HdR]
Amal Khabou [Univ. Paris XI, Associate Professor]
Lionel Lacassagne [Univ. Paris XI, Associate Professor until september 2015, then Univ. Paris 6, Professor, HdR]

Engineer

Konstantin Petrov [Research Engineer, part-time]

PhD Students

Lénaïc Bagnères [Univ. Paris XI]
Aygul Jamal [Univ. Paris XI]
Ian Masliah [Univ. Paris XI]
Adrien Rémy de Zotti [Univ. Paris XI, until Aug 2015]
Laurent Cabaret [École Centrale de Paris, Prag]
Aygül Jamal [Univ. Paris XI]
Jason Lambert [CEA List, Univ. Paris XI]
Antoine Tran Tan [Univ. Paris XI]

Administrative Assistant

Katia Evrat [Inria]

Other

Valentin Labourdette [M2 Internship, from May 2015 until Sep 2015]

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 [143] 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 [79], code generation [69] or communication-avoiding algorithms [37], [98].

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 [49], [50] and in the PhD thesis of Michael Kruse [99]. 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).

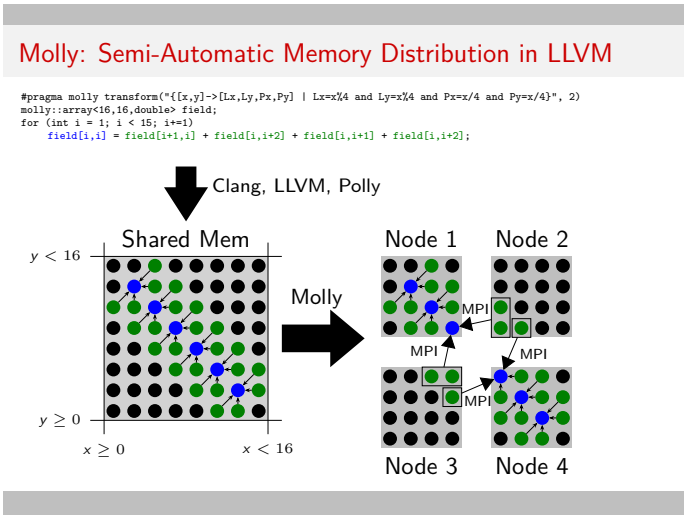

Figure 1. Automatic generation of subdomains using the Molly software.

Data layout is still a critical point that Postale will address. The DSL [69] 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 [79] 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 [50] 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 [51]. In particular we plan to address s-steps algorithms [119] in iterative methods as these program transformations 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 [47]. We have also used reversible computing in register allocation by enabling value rematerialization also by reverse computing [48]. 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 [36] 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 [144]. Especially on supercomputers or cloud framework it might be impossible to get this information. Besides making experiments on energy and power profiling [114], 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 [115].

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 [46]. 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 [122], SPADE [152] or GCC and its support for polyhedral compilation [125]. 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 [28], [95], image processing [56] to graph algorithms [138]. 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 [139] 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 [141] – 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** [151], 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** [62] 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).

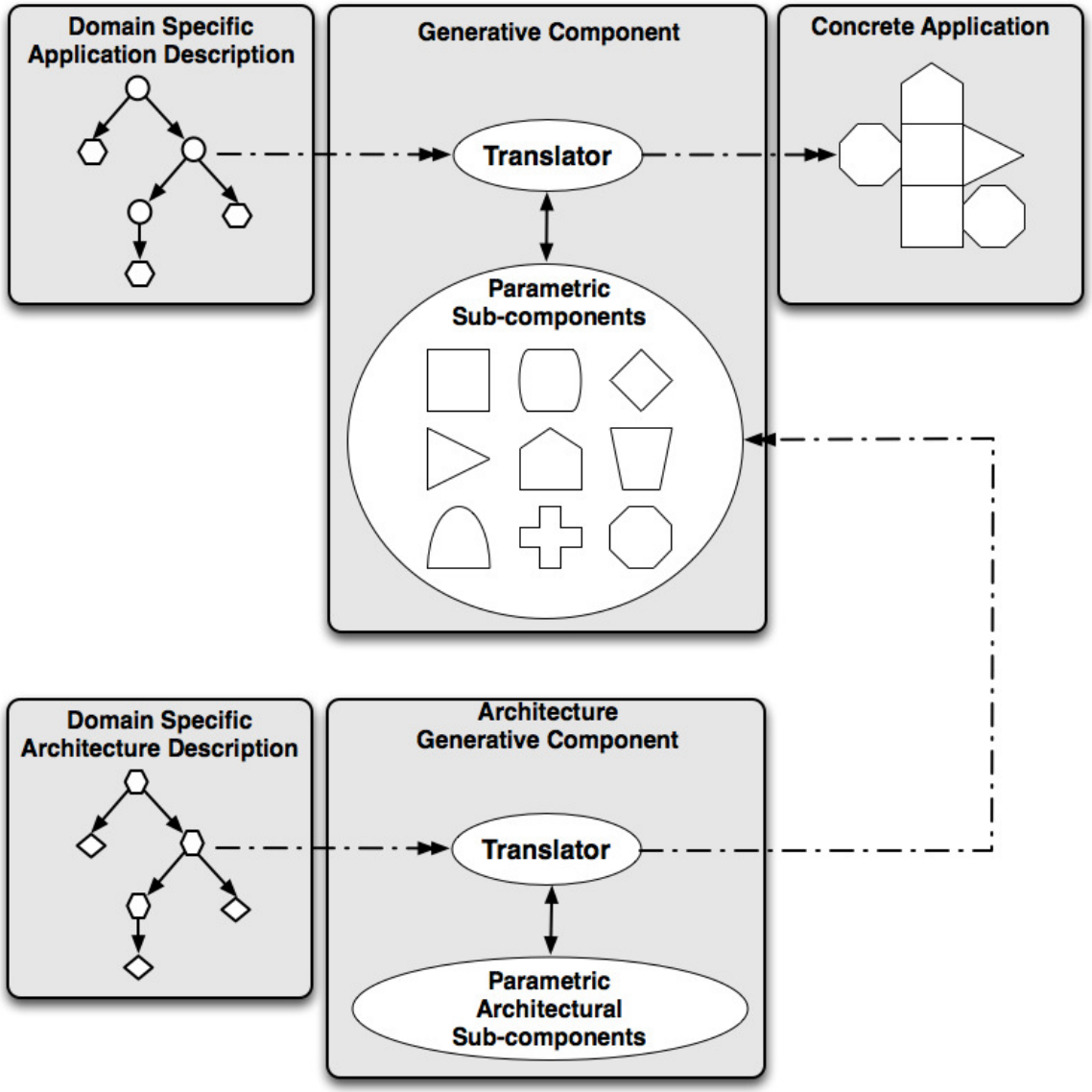

Figure 2. Principles of Architecture Aware Generative Programming

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 [70] 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 [69], [75] 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.

Our work uses a methodogy similar to SCALA [120], and more specifically, the DeLITE [142] 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 [137], which shares a MATLAB-like API with our work, Blaze [55] which supports a similar cost based system for optimizing code and Eigen [86]. 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 [71]) 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 [85] or to other runtime systems including Cloud computing systems and JIT support. Application to other performance metric like power consumption is also planned [156].

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 [27], [25], [32], [66].

3.1.3.2. Activity description and recent achievements

For the european project MILEPOST (2006-2009) [26], 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) [26], [77], [78]. 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 [25], [66].

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.

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 [92], [112]) 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 [59] as well as our existing technology, repository of knowledge and experience, and a growing community [77], [78], [79]. 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 [76].

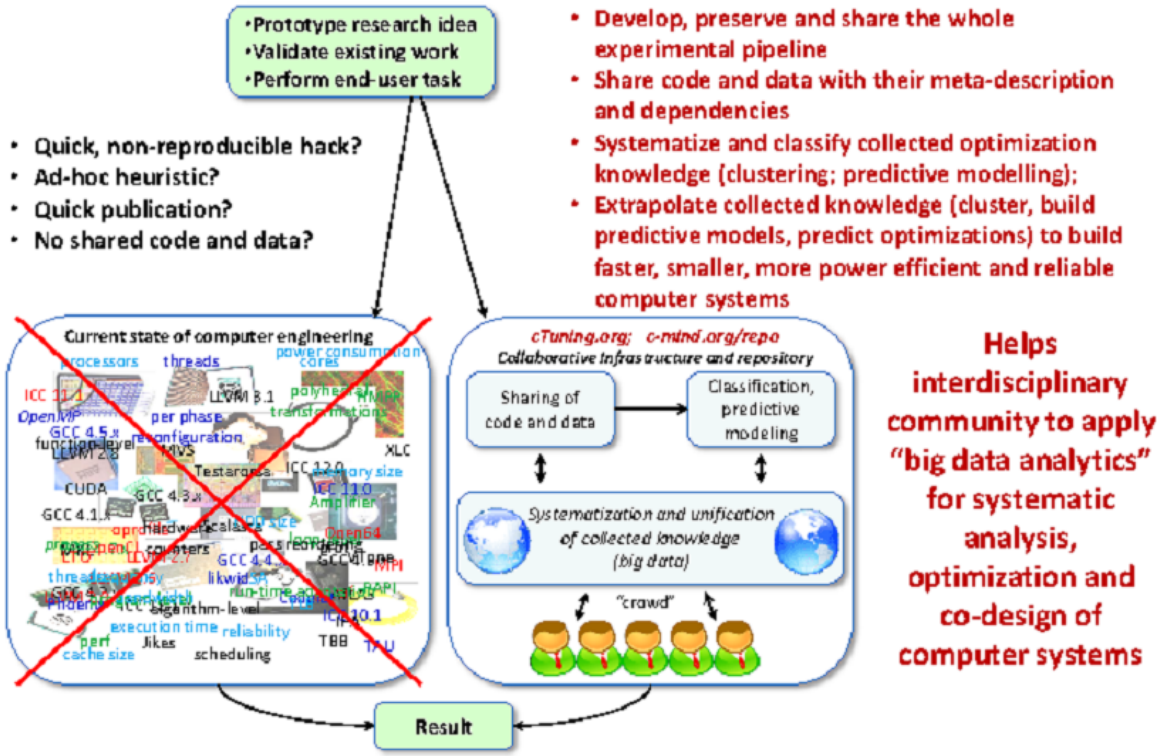

Figure 3. *Considering program optimization and run-time adaptation as a "big data problem"*

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 [42], [149], 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 [83] while the update of the trailing submatrix is performed by the GPU [37]. In another algorithm, we use a random preconditioning (see also Section 3.2.2) of the original matrix to avoid pivoting [40]. 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 [133], when compared to the hybrid multicore/GPU LU algorithm as it is implemented in the public domain library 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 [75] 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 [36] since, to our knowledge, such a solver cannot be found in current public

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

domain libraries (Sca)LAPACK [29], [54], PLASMA [150] and MAGMA [38]. 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 [98]. This algorithm uses a new pivoting strategy based on strong rank revealing QR factorization [84]. 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 [64].

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 [37].

3.2.1.2.3. Application to numerical fluid mechanics:

In an ongoing PhD thesis [153], [154], 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).

3.2.2. 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 [33], low-rank matrix approximation [116], or general matrix decompositions [87].

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 [40], [52] based on random butterfly transformations (RBT) [121] that can be applied to accelerate the solution of general or symmetric indefinite (dense) linear systems for multicore [35] or distributed architectures [34]. 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.

[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)

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 [44] statistical techniques based on the small sample theory [97] to estimate the condition number of linear system/linear least squares solvers [31], [39], [43]. 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 [45], 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).

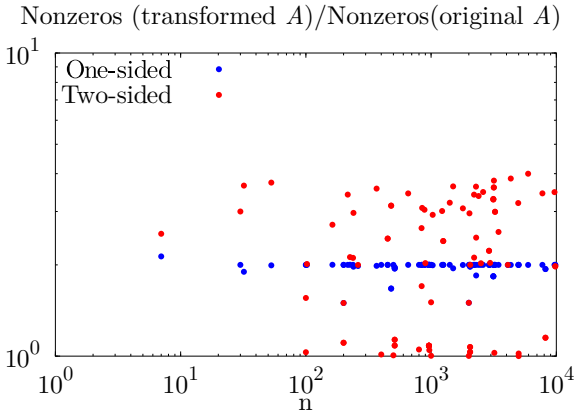

Figure 4. Evaluation of fill-in for one-sided RBT (90 matrices sorted by size).

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

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.

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 [96]. 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 [34]. 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) [94]. Optimal tuning of the code will be obtained using holistic approach developed in the Postale team [79].

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 [97]. 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 [41]. 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 [53], [63], [65], [91]. They are also used in some finance and statistics applications. For example the random orthogonal matrix (ROM) simulation [113] 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 [90].

Stewart [140] 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 [132] 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*) [57], [58] still outperforms other existing CCL algorithms [82], [89], [145] (the first versions of our algorithm appeared in 2009 [105], [106]).

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 [107], [131] with some accuracy tradeoffs. The first one is based on mean shift tracking [80], [81] and the second one relies on covariance matching and tracking [128], [129], [130].

These applications are used in video-surveillance: they perform motion detection [104], motion analysis [146], [147], 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 [108].
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 [117] which converts 32-bit floating point signal processing operators into fixed-point ones.
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 [109], [110], [111], [126], [127].
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 [88] in the image processing area that a fully parallelized code (SIMD+OpenMP) can be accelerated by a factor $\times 80$ on a multicore processor [101]. 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 [155], [156].
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].

[6] Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur
[7] smart memory allocator and management for 2D and 3D image processing

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* [30], [100] for realtime image segmentation using the Horowitz-Pavlidis quad-tree decomposition [93]. Such an algorithm could be useful for accelerating image decomposition like *Fast Level Set Transform* algorithm [118].

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 [72], [103], [102], [124] 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 [60], [67], [68] 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 [69], MPI [73] or image applications on the Cell processor [61], [74], [123], [134], [135], [136], [148].

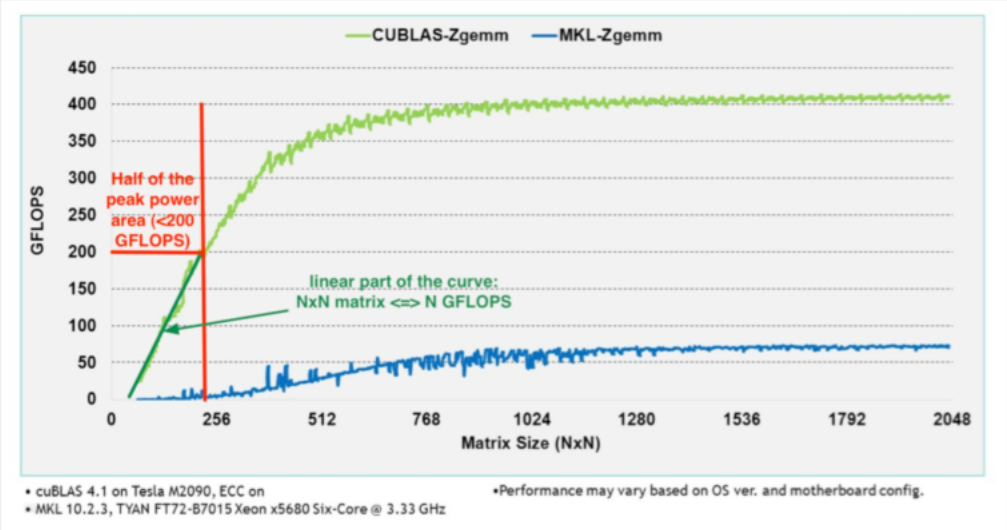

Figure 5. Nvidia cuBLAS performance versus Intel MKL: both have poor performance for small N

[8] agile realtime multi-target tracking algorithm, co-developped with Michèle Gouiffès at LIMSI

4. Highlights of the Year

4.1. Highlights of the Year

Marc Baboulin was invited plenary speaker at the HPCSE conference, Solan, Czech Republic, May 25-28, 2015.

The Random Butterfly Transformations developed by Postale are now available in the MAGMA library for GPU (release 1.6) and Intel Xeon Phi (release 1.3).

Marc Baboulin is general vice-chair of the HPC Symposium to be held in April 2016, Pasadena, CA.

5. New Software and Platforms

5.1. Boost.SIMD

FUNCTIONAL DESCRIPTION

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.

- Contact: Joël Falcou
- URL: http://www.github.com/MetaScale/nt2

5.2. CovTrack

FUNCTIONAL DESCRIPTION

CovTrack: agile realtime multi-target tracking algorithm.

- Contact: Lionel Lacassagne

5.3. Dohko

FUNCTIONAL DESCRIPTION

Dohko is a goal-oriented cloud architecture that aims to simplify the cloud for the users through a declarative strategy. It implements the autonomic properties: self-configuration, self-healing, and context-awareness. In Dohko, the users specify the applications and the requirements (e.g., number of CPU cores, maximal financial cost per hour, among others), and the system automatically (a) selects the resources (i.e., VMs) that meet the constraints, (b) configures and installs the applications in the clouds, (c) handles resource failures, and (d) executes the applications.

- Contact: Alessandro Ferreira Leite
- URL: http://dohko.io/

5.4. Molly

FUNCTIONAL DESCRIPTION

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.

- Contact: Michael Kruse

5.5. MyNRC

FUNCTIONAL DESCRIPTION

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

- Contact: Lionel Lacassagne

5.6. NT2

Numerical Template Toolbox
FUNCTIONAL DESCRIPTION

The Numerical Template Toolbox (NT2) is an Open Source C++ library aimed at simplifying the development, debugging and optimization of high-performance computing applications by providing a Matlab like syntax that eases the transition between prototype and actual application.

- Participants: Joël Falcou, Pierre Estérie and Ian Masliah
- Contact: Joël Falcou
- URL: https://github.com/jfalcou/nt2

6. New Results

6.1. Parallel light speed labeling: the world's fastest connected component labeling for multicore processors

Participants: Lionel Lacassagne, Laurent Cabaret, Daniel Etiemble.

We have designed a parallel version of the Light Speed Labeling for shared-memory multicore processor. This algorithm outperforms the best algorithm by a factor x10. We are now working on the design of algorithms for GPU and manycore embedded processor and especially the TSAR architecture of LIP6 laboratory. More information is available at

- TSAR architecture: https://www-soc.lip6.fr/trac/tsar
- ALMOS operating system: https://www-soc.lip6.fr/trac/almos
- GIET-VM system: https://www-soc.lip6.fr/trac/giet-vm

The paper [20] introduces the parallel version of the Light Speed Labeling (LSL) and compares it with the parallel versions of the competitors. A benchmark shows that the parallel Light Speed Labeling is at least ×1.9 faster than all the other algo- rithms for random images. This factor reach ×3.6 for structured random images. More important, we show that thanks to its run-based processing (segments), LSL is intrinsically more efficient than all pixel-based algorithms.

6.2. Opening Polyhedral Compiler's Black Box

Participants: Lénaïc Bagnères, Oleksandr Zinenko, Stéphane Huot, Cédric Bastoul.

While compilers offer a fair trade-off between productivity and executable performance in single-threaded execution, their optimizations remain fragile when addressing compute-intensive code for parallel architectures with deep memory hierarchies. Moreover, these optimizations operate as black boxes, impenetrable for the user, leaving them with no alternative to time-consuming and error-prone manual optimization in cases where an imprecise cost model or a weak analysis resulted in a bad optimization decision. To address this issue, we propose a technique allowing to automatically translate an arbitrary polyhedral optimization, used internally by loop-level optimization frameworks of several modern compilers, into a sequence of comprehensible syntactic transformations as long as this optimization focuses on scheduling loop iterations. With our approach, we open the black box of the polyhedral frameworks enabling users to examine, refine, replay and even design complex optimizations semi-automatically in partnership with the compiler. [17]

6.3. Automating Resource Selection and Configuration in Inter-clouds through a Software Product Line Method

Participants: Alexandro Ferreira Leite, Vladimir Castro Alves, Genaina Nunes Rodrigues, Claude Tadonki, Christine Eisenbeis, Alba Cristina Alves de Melo.

Nowadays, cloud users face three important problems: (a) choosing one or more appropriate cloud provider(s) to run their application(s), (b) selecting appropriate cloud resources, which implies having enough information about the available resources, including their characteristics and constraints, and (c) configuring the cloud resources. These problems are mostly due to the wide range of resources. These resources usually have distinct dependencies, and they are offered at various clouds' layers. In this complex scenario, the users often have to handle cloud resources and their dependencies manually. This is an error-prone and time-consuming activity, even for skilled cloud users and system administrators. In this context, this paper proposes a software product line engineering (SPLE) method and a tool to deal with these issues. Our SPL-based engineering method enables a declarative and goal-oriented strategy. Furthermore, it allows resource selection and configuration in inter-cloud environments. In our proposal, the cloud users specify their applications and requirements, and our tool automatically selects and configures a suitable computing environment, taking into account temporal and functional dependencies. Experimental results on Amazon EC2 and Google Compute Engine (GCE) show that our approach enables unskilled users to have access to advanced inter-cloud computing configurations, without being concerned with the characteristics of each cloud. [18]

6.4. A Randomized LU-based Solver Using GPU and Intel Xeon Phi Accelerators

Participants: Marc Baboulin, Amal Khabou, Adrien Rémy de Zotti.

We present a fast hybrid solver for dense linear systems based on LU factorization. To achieve good performance, we avoid pivoting by using random butterfly transformations for which we developed efficient implementations on heterogeneous architectures. We used both Graphics Processing Units and Intel Xeon Phi as accelerators. The performance results show that the pre-processing due to randomization is negligible and that the solver outperforms the corresponding routines based on partial pivoting. [16]

6.5. Metaprogramming dense linear algebra solvers. Applications to multi and many-core architectures

Participants: Ian Masliah, Marc Baboulin, Joël Falcou.

The increasing complexity of new parallel architectures has widened the gap between adaptability and efficiency of the codes. As high performance numerical libraries tend to focus more on performance, we wish to address this issue using a C++ library called NT2. By analyzing the properties of the linear algebra domain that can be extracted from numerical libraries and combining them with architectural features, we developed a generic approach to solve dense linear systems on various architectures including CPU and GPU. We have then extended our work with an example of a least squares solver based on semi-normal equations in mixed precision that cannot be found in current libraries. For the automatically generated solvers, we report performance comparisons with state-of-the-art codes, and show that it is possible to obtain a generic code with a high-level interface (similar to MATLAB) which runs either on CPU or GPU without generating a significant overhead. [21] [23]

6.6. Using Random Butterfly Transformations in Parallel Schur Complement-Based Preconditioning

Participants: Marc Baboulin, Aygul Jamal, Masha Sosonkina.

We propose to use a randomization technique based on Random Butterfly Transformations (RBT) in the Algebraic Recursive Multilevel Solver (ARMS) to improve the preconditioning phase in the iterative solution of sparse linear systems. We integrated the RBT technique into the parallel version of ARMS (pARMS). The preliminary experimental results on some matrices from the Davis' collection show an improvement of the convergence and accuracy of the results when compared with existing implementations of the pARMS preconditioner. [15]

6.7. LU Preconditioning for Overdetermined Sparse Least Squares Problems

Participants: Gary Howell, Marc Baboulin.

We investigate how to use an LU factorization with the classical LSQR routine for solving overdetermined sparse least squares problems. Usually L is much better conditioned than A and iterating with L instead of A results in faster convergence. When a runtime test indicates that L is not sufficiently well-conditioned, a partial orthogonalization of L accelerates the convergence. Numerical experiments illustrate the good behavior of our algorithm in terms of storage and convergence. [19]

6.8. Dense Symmetric Indefinite Factorization on GPU Accelerated Architectures

Participants: Marc Baboulin, Jack Dongarra, Adrien Rémy de Zotti, Stanimire Tomov, Ichitaro Yamazaki.

We study the performance of dense symmetric indefinite factorizations (Bunch-Kaufman and Aasen's algorithms) on multicore CPUs with a Graphics Processing Unit (GPU). Though such algorithms are needed in many scientific and engineering simulations, obtaining high performance of the factorization on the GPU is difficult because the pivoting that is required to ensure the numerical stability of the factorization leads to frequent synchronizations and irregular data accesses. As a result, until recently, there has not been any implementation of these algorithms on hybrid CPU/GPU architectures. To improve their performance on the hybrid architecture, we explore different techniques to reduce the expensive communication and synchronization between the CPU and GPU, or on the GPU. We also study the performance of a symmetric indefinite factorization with no pivoting combined with the preprocessing technique based on Random Butterfly Transformations. Though such transformations only have probabilistic results on the numerical stability, they avoid the pivoting and obtain a great performance on the GPU. [14]

6.9. Computing least squares condition numbers on hybrid multicore/GPU systems

Participants: Marc Baboulin, Jack Dongarra, Rémi Lacroix.

We present an efficient computation for least squares conditioning or estimates of it. We propose performance results using new routines on top of the multicore-GPU library MAGMA. This set of routines is based on an efficient computation of the variance-covariance matrix for which, to our knowledge, there is no implementation in current public domain libraries LAPACK and ScaLAPACK. [22]

6.10. Towards a High-Performance Tensor Algebra Package for Accelerators

Participants: Marc Baboulin, Veselin Dobrev, Jack Dongarra, Christopher Earl, Joël Falcou, Azzam Haidar, Ian Karlin, Tzanio Kolev, Ian Masliah, Stanimire Tomov.

Numerous important applications, e.g., high-order FEM simulations, can be expressed through tensors. Examples are computation of FE matrices and SpMV products expressed as generalized tensor contractions. Contractions by the first index can often be represented as tensor index reordering plus gemm, which is a key factor to achieve high-performance. We present ongoing work on the design of a high-performance package in MAGMA for Tensor algebra that includes techniques to organize tensor contractions, data storage, and parametrization related to batched execution of large number of small tensor contractions. We apply auto-tuning and code generation techniques to provide an architecture-aware, user-friendly interface. [24]

7. Bilateral Contracts and Grants with Industry

7.1. 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, Amal Khabou.

  It concerns two differents topics:
  - Enhancing performance of numerical solvers using accelerators (postdoc started in October 2014).
  - Studying numerical quality and reproducibility in HPC exascale applications (ongoing ANR submission).
- NumScale: 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.

8. Partnerships and Cooperations

8.1. 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.
- **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, Konstantin Petrov.

8.2. International Initiatives

8.2.1. Inria Associate Teams not involved in an Inria International Labs

8.2.1.1. R-LAS

Title: Randomized Linear Algebra Software

International Partner (Institution - Laboratory - Researcher):

University of Tennessee, Knoxville (United States) - Innovative Computing Laboratory (ICL) - Jack Dongarra

Start year: 2014

See also: https://www.lri.fr/~baboulin/r-las.html

The objective of the associate team between Inria and University of Tennessee is to develop a class of fast algorithms and software based on randomization to enhance linear algebra calculations in high-performance computing (HPC) applications. The first application will focus on FFT-like randomization techniques to avoid pivoting in dense and sparse matrix factorizations and thus removing the communication cost due to pivoting. The second application is related to the computation of statistical condition estimates for linear algebra problems in order to assess the numerical quality of solutions computed by HPC applications. The targeted architectures are large scale multicore systems with accelerators. The ultimate goal of the project is to make the randomized solvers designed by the associate team accessible to end-users thanks to a public domain software library.

8.3. International Research Visitors

8.3.1. Visits of International Scientists

- Masha Sosonkina, Old Dominion University, USA.
- Hartwig Anzt, University of Tennessee, USA.
- Nick Higham, University of Manchester, UK.
- Jean-Luc Gaudiot, UC Irvine, USA.

8.3.2. Visits to International Teams

8.3.2.1. Research stays abroad

- Marc Baboulin,
  - Invitation at Old Dominion University, Norfolk, USA, (October 2015)
  - Invitation at National Institute of Informatics, Tokyo, Japan (August 2015)
  - Invitation at Académie des Sciences de Prague, République Tchèque (June 2015)
  - Invitation at Inria Bordeaux- équipe Hiepacs (March 2015)

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific events organisation

9.1.1.1. Member of the organizing committees

- Marc Baboulin,
  - Member of the steering committee of HPC 2015
  - Minisymposium “Randomized algorithms in numerical linear algebra.” Organizers: Marc Baboulin, Jack Dongarra (Univ. of Tennessee) et Sherry Li (Lawrence Berkeley National Laboratory), SIAM Conference on Computational Science and Engineering, Salt Lake City, USA, March 14-18, 2015.
- Christine Eisenbeis, comité d’organisation du colloque "Les métiers du travail scientifique : images et valeurs, réalités et défis", 20-21 mai 2015, Orsay, (http://www.centre-dalembert.u-psud.fr/2015-les-metiers-du-travail-scientifique-images-et-valeurs-realites-et-defis/).

9.1.2. Journal

9.1.2.1. Member of the editorial board

- Christine Eisenbeis, IJPP

9.1.3. Invited talks

- Marc Baboulin, invited plenary speaker at the HPCSE conference, Solan, Czech Republic, May 25-28, 2015.

9.1.4. Scientific expertise

- Marc Baboulin, comité de sélection pour un poste de maître de conférences en informatique, université Paris-Sud 11, mai 2015.
- Christine Eisenbeis, comité de sélection pour un poste de professeur d’informatique, université Paris-Sud 11, mai 2015.
- Joël Falcou, comité de sélection pour un poste de maître de conférences en informatique, université Paris-Sud 11, mai 2015.
- Lionel Lacassagne, comité de sélection pour un poste de maître de conférences en informatique, université Paris-Sud 11, mai 2015.

9.2. Teaching - Supervision - Juries

9.2.1. Supervision

PhD in progress: Lénaïc Bagnères, "Adaptation automatique et semi-automatique des optimisations de programmes", University Paris-Sud 11, supervisors: Cédric Bastoul and Christine Eisenbeis

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 defended: Jason Lambert, "Parallélisation de simulations interactives de champs ultrasonores pour le contrôle non destructif", CEA List/Disc funding, University Paris-Sud 11, July 3rd, 2015, Supervisor: Lionel Lacassagne

PhD defended: Adrien Rémy, "Solving dense linear systems on accelerated multicore architectures", University Paris Sud 11, July 8th, 2015, Supervisors: M. Baboulin and B. Rozoy

PhD defended: Antoine Tran Tan, "Squelettes algorithmiques asynchrones : application aux langages orientés domaine", University Paris-Sud 11, October 8th, 2015, Supervisor: Joël Falcou

PhD defended: Yushan Wang, "Solving Incompressible Navier-Stokes Equations on Heterogeneous Parallel Architectures", University Paris Sud 11, April 9th, 2015, Supervisors: M. Baboulin and O. Le Maître

9.2.2. Juries

- Marc Baboulin,

  - PhD defence of Michele Mangili, Politecnico di Milano (15/12/2015), chairman of the committee.
  - HDR defence of Stéphane Louise, CEA Saclay (29/06/2015), chairman of the committee.
- Christine Eisenbeis, rapporteuse de la thèse de Nicolas Triquenaux, « Energy Characterization and Savings in Single and Multiprocessor Systems : Understanding how much can be saved and how to achieve it in modern systems » , 18 septembre 2015, université de Versailles-Saint-Quentin.

9.3. 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 "Les métiers du travail scientifique : images et valeurs, réalités et défis", 20-21 mai 2015, Orsay, (http://www.centre-dalembert.u-psud.fr/2015-les-metiers-du-travail-scientifique-images-et-valeurs-realites-et-defis/).

10. 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, LPT-Orsay-13-142 article nb. 012005 [*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^{o} 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^{o} 5&6, pp. 232–240

Publications of the year

Doctoral Dissertations and Habilitation Theses

[13] A. RÉMY. *Solving dense linear systems on accelerated multicore architectures*, Université Paris-Sud, July 2015, https://hal.inria.fr/tel-01206837

International Conferences with Proceedings

[14] M. BABOULIN, J. DONGARRA, A. RÉMY, S. TOMOV, I. YAMAZAKI. *Dense Symmetric Indefinite Factorization on GPU Accelerated Architectures*, in "11th International Conference on Parallel Processing and Applied Mathematics (PPAM 2015)", Krakow, Poland, Lecture Notes in Computer Science, September 2015, https://hal.inria.fr/hal-01223022

[15] M. BABOULIN, A. JAMAL, M. SOSONKINA. *Using Random Butterfly Transformations in Parallel Schur Complement-Based Preconditioning*, in "8th Workshop on Computer Aspects of Numerical Algorithms (CANA'15)", Lodz, Poland, September 2015, https://hal.inria.fr/hal-01223090

[16] M. BABOULIN, A. KHABOU, A. RÉMY. *A Randomized LU-based Solver Using GPU and Intel Xeon Phi Accelerators*, in "HeteroPar'2015", Vienna, Austria, August 2015, https://hal.inria.fr/hal-01223018

[17] L. BAGNÈRES, O. ZINENKO, S. HUOT, C. BASTOUL. *Opening Polyhedral Compiler's Black Box*, in "CGO 2016 - 14th Annual IEEE/ACM International Symposium on Code Generation and Optimization", Barcelona, Spain, March 2016, https://hal.inria.fr/hal-01253322

[18] A. FERREIRA LEITE, V. C. ALVES, G. NUNES RODRIGUES, C. TADONKI, C. EISENBEIS, A. C. MAGALHAES ALVES DE MELO. *Automating Resource Selection and Configuration in Inter-clouds through a Software Product Line Method*, in "8th International Conference on Cloud Computing (CLOUD), 2015 IEEE", New York City, United States, July 2015, pp. 726-733 [*DOI :* 10.1109/CLOUD.2015.101], https://hal-mines-paristech.archives-ouvertes.fr/hal-01252985

[19] G. W. HOWELL, M. BABOULIN. *LU Preconditioning for Overdetermined Sparse Least Squares Problems*, in "11th International Conference on Parallel Processing and Applied Mathematics (PPAM 2015)", Krakow, Poland, Lecture Notes in Computer Science, September 2015, https://hal.inria.fr/hal-01223069

[20] L. LACASSAGNE, L. CABARET, D. ETIEMBLE. *Parallel light speed labeling: the world's fastest connected component labeling for multicore processors*, in "International Conference on Image Processing", Quebec, Canada, IEEE, September 2015, 8 p. , https://hal.inria.fr/hal-01243310

[21] I. MASLIAH, M. BABOULIN, J. FALCOU. *Metaprogramming dense linear algebra solvers. Applications to multi and many-core architectures*, in "13th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA 2015)", Helsinki, Finland, August 2015, https://hal.inria.fr/hal-01221358

Scientific Books (or Scientific Book chapters)

[22] M. BABOULIN, J. DONGARRA, R. LACROIX. *Computing least squares condition numbers on hybrid multicore/GPU systems*, in "Interdisciplinary Topics in Applied Mathematics, Modeling and Computational Science", Springer International Publishing, 2015, vol. 117 [*DOI :* 10.1007/978-3-319-12307-3_6], https://hal.inria.fr/hal-01204804

Research Reports

[23] M. BABOULIN, J. FALCOU, I. MASLIAH. *Meta-programming and Multi-stage Programming for GPGPUs*, Inria Saclay Ile de France ; Paris-Sud XI, September 2015, n^{o} RR-8780, https://hal.inria.fr/hal-01204661

Other Publications

[24] M. BABOULIN, V. DOBREV, J. DONGARRA, C. EARL, J. FALCOU, A. HAIDAR, I. KARLIN, T. KOLEV, I. MASLIAH, S. TOMOV. *Towards a High-Performance Tensor Algebra Package for Accelerators*, August 2015, Smoky Mountains Computational Sciences and Engineering Conference (SMC 2015), Poster, https://hal.archives-ouvertes.fr/hal-01231234

References in notes

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

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

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

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

[29] 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

[30] 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

[31] 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

[32] 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

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

[34] 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

[35] 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

[36] 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

[37] 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

[38] 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

[39] 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

[40] 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

[41] 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

[42] 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

[43] 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

[44] 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

[45] 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

[46] 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

[47] 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

[48] 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

[49] 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

[50] 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

[51] 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

[52] 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

[53] 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]

[54] 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

[55] BLAZE. *The Blaze Library*, 2014, https://bitbucket.org/blaze-lib/blaze

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

[57] 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

[58] 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

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

[60] 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]

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

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

[63] 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

[64] 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

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

[66] 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

[67] 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]

[68] 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

[69] 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

[70] 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

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

[72] 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

[73] 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

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

[75] 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

[76] 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

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

[78] 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

[79] 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

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

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

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

[83] 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

[84] 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^{o} 4, pp. 848–869, http://dx.doi.org/10.1137/0917055

[85] 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

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

[87] 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

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

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

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

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

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

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

[94] 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^{O} 8, pp. 1927–1936

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

[96] 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

[97] 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

[98] 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

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

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

[101] 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

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

[103] 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

[104] 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]

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

[106] 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

[107] 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

[108] 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

[109] 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

[110] 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

[111] 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

[112] 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

[113] 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]

[114] 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

[115] 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

[116] 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

[117] 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]

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

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

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

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

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

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

[124] 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

[125] 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

[126] 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

[127] 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

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

[129] 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

[130] 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

[131] 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

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

[133] 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

[134] 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

[135] 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]

[136] 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

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

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

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

[140] 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^{o} 3, pp. 403-409

[141] 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

[142] 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

[143] 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]

[144] 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^{o} 2, pp. 777–797

[145] 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^{o} 1, pp. 1-23 [*DOI :* 10.1016/S1077-3142(02)00030-9]

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

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

[148] 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

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

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

[151] T. L. VELDHUIZEN. *Active Libraries and Universal Languages*, Indiana University Computer Science, May 2004, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.3916

[152] 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

[153] 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

[154] 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

[155] 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

[156] 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