IN PARTNERSHIP WITH:
CNRS

Université de Bordeaux

Activity Report 2013

Project-Team RUNTIME

Efficient runtime systems for parallel architectures

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

RESEARCH CENTER
Bordeaux - Sud-Ouest

THEME
Distributed and High Performance Computing

Table of contents

Project-Team RUNTIME

Keywords: High Performance Computing, Scheduling, Runtime Systems, Multicore And Gpu, Programming Languages

Creation of the Project-Team: 2004 October 07*, updated into Team:* 2014 January 01.

1. Members

Research Scientists
Olivier Aumage [Inria, Researcher]
Alexandre Denis [Inria, Researcher]
Brice Goglin [Inria, Researcher]
Emmanuel Jeannot [Inria, Senior Researcher, HdR]

Faculty Members
Raymond Namyst [Team leader, Univ. Bordeaux I, Professor, HdR]
Denis Barthou [IPB, Professor, HdR]
Marie-Christine Counilh [Univ. Bordeaux I, Associate Professor]
Guillaume Mercier [IPB, Associate Professor]
Samuel Thibault [Univ. Bordeaux I, Associate Professor]
Pierre-André Wacrenier [Univ. Bordeaux I, Associate Professor]

External Collaborator
Julien Jaeger [Univ. Versailles, until Jan 2013]

Engineers
Cyril Bordage [Inria, granted by ANR FP3C project, until Dec 2013]
Nathalie Furmento [CNRS]
Tamara Meunier [Univ. Bordeaux I, from Jan 2013 until Dec 2013]

PhD Students
Paul-Antoine Arras [STMicroElectronics, granted by CIFRE]
Emmanuel Cieren [CEA, granted by CIFRE]
Sylvain Henry [Inria, granted by ANR FP3C project, until Oct 2013]
Andra Hugo [Inria]
Suraj Kumar [Inria, from Dec 2013]
Pei Li [Telecom Sud Paris, granted by Telecom Sud Paris]
Bertrand Putigny [Inria]
Corentin Rossignon [Total, granted by CIFRE]
Emmanuelle Saillard [CEA, granted by CIFRE]
Marc Sergent [Inria, granted by CEA/Region project]
François Tessier [Univ. Bordeaux I]
Gregory Vaumourin [CEA, from Oct 2013]
Soufiane Baghdadi [Telecom Sud Paris, granted by Telecom Sud Paris]

Post-Doctoral Fellow
Lilia Ziane [Inria, granted by ANR -SONGS-CEPAGE project, from Jul 2013]

Visiting Scientists
Marcelo Alaniz [invited researcher, from Sep 2013 until Sep 2013]
Sébastien Frémal [invited researcher, from Apr 2013 until Jun 2013]
Sergio Nesmachnow Canovas [invited researcher, from Sep 2013 until Sep 2013]

Administrative Assistant
Sylvie Embolla [Inria]

2. Overall Objectives

2.1. Designing Efficient Runtime Systems

Parallel, Runtime, Environment, Heterogeneity, SMP, Multicore, NUMA, HPC, High-Speed Networks, Protocols, MPI, Scheduling, Thread, OpenMP, Compiler Optimizations

The RUNTIME research project takes place within the context of High Performance Computing. It seeks to explore the design, the implementation and the evaluation of novel mechanisms needed by **runtime systems** for parallel computers. *Runtime systems* are intermediate software layers providing parallel programming environments with specific functionalities left unaddressed by the operating system. Runtime systems serve as a target for parallel language compilers (e.g. OpenMP), numerical libraries (e.g. Basic Linear Algebra Routines), communication libraries (e.g. MPI) or high-level programming environments (e.g. Charm++).

Runtime systems can thus be seen as functional extensions of operating systems, but the boundary between these layers is rather fuzzy since runtime systems often bypass (or redefine) functions usually implemented at the OS level. The increasing complexity of modern parallel hardware makes it even more necessary to postpone essential decisions and actions (scheduling, optimizations) at run time. Since runtime systems are able to perform dynamically what cannot be done statically, they indeed constitute an essential piece in the HPC software stack. The typical duties of a runtime system include task/thread scheduling, memory management, intra and extranode communication, synchronization, support for trace generation, topology discovery, etc. **The core of our research activities aims at improving algorithms and techniques involved in the design of runtime systems tailored for modern parallel architectures.**

One of the main challenges encountered when designing modern runtime systems is to provide powerful abstractions, both at the programming interface level and at the implementation level, to ensure **portability of performance** on increasingly complex hardware architectures. Consequently, even if the design of efficient algorithms obviously remains an important part of our research activity, the main challenge is to find means to transfer knowledge from the application down to the runtime system. It is indeed crucial to keep and take advantage of information about the application behavior at the level where scheduling or transfer decisions are made. We have thus devoted significant efforts in **providing programming environments with portable ways to transmit hints** (eg. scheduling hints, memory management hints, etc.) to the underlying runtime system.

As detailed in the following sections, our research group has been developing a large spectrum of research topics during the last four years, ranging from low-level code optimization techniques to high-level task-based programming interfaces. The originality of our approach lies in the fact that we try to address these issues following a global approach, keeping in mind that all the achievements are intended to be eventually integrated together within a unified software stack. This led us to cross-study differents topics and co-design several pieces of software.

Our research project centers on three main directions:

Mastering large, hierarchical multiprocessor machines

- Thread scheduling over multicore machines
- Task scheduling over GPU heterogeneous machines
- Exploring parallelism orchestration at compiler and runtime level
- Improved interactions between optimizing compiler and runtime
- Modeling performance of hierarchical multicore nodes

Optimizing communication over high performance clusters

- Scheduling data packets over high speed networks
- New MPI implementations for Petascale computers
- Optimized intra-node communication

- Message passing over commodity networking hardware
- Influence of process placement on parallel applications performance

Integrating Communications and Multithreading

- Parallel, event-driven communication libraries
- Communication and I/O within large multicore nodes

Beside those main research topics, we obviously intend to work in collaboration with other research teams in order to validate our achievements by integrating our results into larger software environments (MPI, OpenMP) and to join our efforts to solve complex problems.

Among the target environments, we intend to carry on developing the successor to the PM^2 software suite, which would be a kind of technological showcase to validate our new concepts on real applications through both academic and industrial collaborations (CEA/DAM, Bull, IFP, Total, Exascale Research Lab.). We also plan to port standard environments and libraries (which might be a slightly sub-optimal way of using our platform) by proposing extensions (as we already did for MPI and Pthreads) in order to ensure a much wider spreading of our work and thus to get more important feedback.

Finally, as most of our work proposed is intended to be used as a foundation for environments and programming tools exploiting large scale, high performance computing platforms, we definitely need to address the numerous scalability issues related to the huge number of cores and the deep hierarchy of memory, I/O and communication links.

2.2. Highlights of the Year

- The hwloc software 5.2 is used for node topology discovery and process binding by the most popular MPI implementations, including MPICH2 and OPEN MPI and all their derivatives such as Intel MPI.
- The StarPU software 5.6 is used for dynamic scheduling by EADS for his hmatrix solver.

3. Research Program

3.1. Runtime Systems Evolution

parallel,distributed,cluster,environment,library,communication,multithreading,multicore

This research project takes place within the context of high-performance computing. It seeks to contribute to the design and implementation of parallel runtime systems that shall serve as a basis for the implementation of high-level parallel middleware. Today, the implementation of such software (programming environments, numerical libraries, parallel language compilers, parallel virtual machines, etc.) has become so complex that the use of portable, low-level runtime systems is unavoidable.

Our research project centers on three main directions:

Mastering large, hierarchical multiprocessor machines With the beginning of the new century, computer makers have initiated a long term move of integrating more and more processing units, as an answer to the frequency wall hit by the technology. This integration cannot be made in a basic, planar scheme beyond a couple of processing units for scalability reasons. Instead, vendors have to resort to organize those processing units following some hierarchical structure scheme. A level in the hierarchy is then materialized by small groups of units sharing some common local cache or memory bank. Memory accesses outside the locality of the group are still possible thanks to bus-level consistency mechanisms but are significantly more expensive than local accesses, which, by definition, characterizes NUMA architectures.

Thus, the task scheduler must feed an increasing number of processing units with work to execute and data to process while keeping the rate of penalized memory accesses as low as possible. False sharing, ping-pong effects, data vs task locality mismatches, and even task vs task locality mismatches between tightly synchronizing activities are examples of the numerous sources of overhead that may arise if threads and data are not distributed properly by the scheduler. To avoid these pitfalls, the scheduler therefore needs accurate information both about the computing platform layout it is running on and about the structure and activities relationships of the application it is scheduling.

As quoted by Gao *et al.* [43], we believe it is important to expose domain-specific knowledge semantics to the various software components in order to organize computation according to the application and architecture. Indeed, the whole software stack, from the application to the scheduler, should be involved in the parallelizing, scheduling and locality adaptation decisions by providing useful information to the other components. Unfortunately, most operating systems only provide a poor scheduling API that does not allow applications to transmit valuable *hints* to the system.

This is why we investigate new approaches in the design of thread schedulers, focusing on high-level abstractions to both model hierarchical architectures and describe the structure of applications' parallelism. In particular, we have introduced the *bubble* scheduling concept [7] that helps to structure relations between threads in a way that can be efficiently exploited by the underlying thread scheduler. *Bubbles* express the inherent parallel structure of multithreaded applications: they are abstractions for grouping threads which "work together" in a recursive way. We are exploring how to dynamically schedule these irregular nested sets of threads on hierarchical machines [3], the key challenge being to schedule related threads as closely as possible in order to benefit from cache effects and avoid NUMA penalties. We are also exploring how to improve the transfer of scheduling hints from the programming environment to the runtime system, to achieve better computation efficiency.

This is also the reason why we explore new languages and compiler optimizations to better use domain specific information. We propose a new domain specific language, QIRAL, to generate parallel codes from high level formulations for Lattice QCD problems. QIRAL describes the formulation of the algorithms, of the matrices and preconditions used in this domain and generalizes languages such as SPIRAL used in auto-tuning library generator for signal processing applications. Lattice QCD applications require huge amount of processing power, on multinode, multi-core with GPUs. Simulation codes require to find new algorithms and efficient parallelization. So far, the difficulties for orchestrating parallelism efficiently hinder algorithmic exploration. The objective of QIRAL is to decouple algorithm exploration with parallelism description. Compiling QIRAL uses rewriting techniques for algorithm exploration, parallelization techniques for parallel code generation and potentially, runtime support to orchestrate this parallelism. Results of this work have been published in [9]. A similar approach, this time targeting methods to solve matrix equations, has been proposed [17]. Hydra focuses on systems of equations involving regular shaped matrices (such as upper triangular for instance) and finds automatically a parallel method to solve this system. The approach, using to a divide and conquer technique, works for several equations such as LU decomposition, Sylvester equation and has been shown to be comparable or outperforming Intel MKL library on multicores. Hydra relies on StarPU.

For parallel programs running on multicores, thread affinity and data locality is essential for performance. We investigated in [23] how thread pinning strategies could impact performance and performance stability and compared the efficiency of several profile-guided strategies with compile-time strategies. Following this effort, in MAQAO, we developped a language to ease the instrumentation of parallel codes, in particular for capturing memory traces [16]. Through the combined analysis of the code behavior, at compile time and at runtime, MAQAO can then help users to better pinpoint and quantify performance issues in OpenMP codes, find load imbalance between threads, size of working sets, false sharing situations... The MAQAO instrumentation language has been used successfully in other tools, such as TAU. Besides, we proposed in [15] to combine static and dynamic dependence analysis for the detection of vectorization opportunities. MAQAO then estimates the potential gain that could be reached through vectorization and identifies the required code transformations, either by changing loop control or data layout.

Aside from greedily invading all these new cores, demanding HPC applications now throw excited glances at the appealing computing power left unharvested inside the graphical processing units (GPUs). A strong demand is arising from the application programmers to be given means to access this power without bearing an unaffordable burden on the portability side. Efforts have already been made by the community in this respect but the tools provided still are rather close to the hardware, if not to the metal. Hence, we decided to launch some investigations on addressing this issue. In particular, we have designed a programming environment named StarPU that enables the programmer to offload tasks onto such heterogeneous processing units and gives that programmer tools to fit tasks to processing units capability, tools to efficiently manage data moves to and from the offloading hardware and handles the scheduling of such tasks all in an abstracted, portable manner. The challenge here is to take into account the intricacies of all computation unit: not only the computation power is heterogeneous among the machine, but data transfers themselves have various behavior depending on the machine architecture and GPUs capabilities, and thus have to be taken into account to get the best performance from the underlying machine. As a consequence, StarPU not only pays attention to fully exploit each of the different computational resources at the same time by properly mapping tasks in a dynamic manner according to their computation power and task behavior by the means of scheduling policies, but it also provides a distributed shared-memory library that makes it possible to manipulate data across heterogeneous multicore architectures in a high-level fashion while being optimized according to the machine possibilities. In addition to this, the scheduling policy of StarPU has been modularized; this makes it easy to experiment with state of the art theoretical scheduling strategies. Last but not least, StarPU works over clusters, by extending the shared-memory view over the MPI communication library. This allows, with the same sequential-looking application source code, to tackle all architectures from small multicore systems to clusters of heterogeneous systems.

We extended OpenCL capabilities by proposing to use, transparently, StarPU as an OpenCL device [35]. A functional approach to StarPU has been proposed besides in [18].

Optimizing communications over high performance clusters and grids Using a large panel of mechanisms such as user-mode communications, zero-copy transactions and communication operation offload, the critical path in sending and receiving a packet over high speed networks has been drastically reduced over the years. Recent implementations of the MPI standard, which have been carefully designed to directly map *basic* point-to-point requests onto the underlying low-level interfaces, almost reach the same level of performance for very basic point-to-point messaging requests. However more complex requests such as non-contiguous messages are left mostly unattended, and even more so are the irregular and multiflow communication schemes. The intent of the work on our NewMadeleine communication engine, for instance, is to address this situation thoroughly. The NewMadeleine optimization layer delivers much better performance on *complex* communication schemes with negligible overhead on basic single packet point-to-point requests. Through Mad-MPI, our proof-of-concept implementation of a subset of the MPI API, we intend to show that MPI applications can also benefit from the NewMadeleine communication engine.

The increasing number of cores in cluster nodes also raises the importance of intra-node communication. Our KNem software module aims at offering optimized communication strategies for this special case and let the above MPI implementations benefit from dedicated models depending on process placement and hardware characteristics.

Moreover, the convergence between specialized high-speed networks and traditional Ethernet networks leads to the need to adapt former software and hardware innovations to new message-passing stacks. Our work on the Open-MX software is carried out in this context.

Regarding larger scale configurations (clusters of clusters, grids), we intend to propose new models, principles and mechanisms that should allow to combine communication handling, threads scheduling and I/O event monitoring on such architectures, both in a portable and efficient way. We particularly intend to study the introduction of new runtime system functionalities to ease the development of code-coupling distributed applications, while minimizing their unavoidable negative impact on the application performance.

Integrating Communications and Multithreading Asynchronism is becoming ubiquitous in modern communication runtimes. Complex optimizations based on online analysis of the communication schemes and on the de-coupling of the request submission vs processing. Flow multiplexing or transparent heterogeneous networking also imply an active role of the runtime system request submit and process. And communication overlap as well as reactiveness are critical. Since network request cost is in the order of magnitude of several thousands CPU cycles at least, independent computations should not get blocked by an ongoing network transaction. This is even more true with the increasingly dense SMP, multicore, SMT architectures where many computing units share a few NICs. Since portability is one of the most important requirements for communication runtime systems, the usual approach to implement asynchronous processing is to use threads (such as Posix threads). Popular communication runtimes indeed are starting to make use of threads internally and also allow applications to also be multithreaded. Low level communication libraries also make use of multithreading. Such an introduction of threads inside communication subsystems is not going without troubles however. The fact that multithreading is still usually optional with these runtimes is symptomatic of the difficulty to get the benefits of multithreading in the context of networking without suffering from the potential drawbacks. We advocate the importance of the cooperation between the asynchronous event management code and the thread scheduling code in order to avoid such disadvantages. We intend to propose a framework for symbiotically combining both approaches inside a new generic I/O event manager.

Moreover, the design of distributed parallel code, integrating both MPI and OpenMP, is complex and error-prine. Deadlock situations may arise and are difficult to detect. We proposed an original approach, based on static (compile-time) analysis and runtime verification in order to detect deadlock situation but also to pinpoint the cause of such deadlock [27]. This work first focuses on MPI communication alone, the extension to hybrid MPI/OpenMP codes is in progress.

4. Application Domains

4.1. Application Domains

HPC, simulation

The RUNTIME group is working on the design of efficient runtime systems for parallel architectures. We are currently focusing our efforts on High Performance Computing applications that merely implement numerical simulations in the field of Seismology, Weather Forecasting, Energy, Mechanics or Molecular Dynamics. These time-consuming applications need so much computing power that they need to run over parallel machines composed of several thousands of processors.

Because the lifetime of HPC applications often spreads over several years and because they are developed by many people, they have strong portability constraints. Thus, these applications are mostly developed on top of standard APIs (e.g. MPI for communications over distributed machines, OpenMP for shared-memory programming). That explains why we have long standing collaborations with research groups developing parallel language compilers, parallel programming environments, numerical libraries or communication software. Actually, all these "clients" are our primary target.

Although we are currently mainly working on HPC applications, many other fields may benefit from the techniques developed by our group. Since a large part of our efforts is devoted to exploiting multicore machines and GPU accelerators, many desktop applications could be parallelized using our runtime systems (e.g. 3D rendering, etc.).

5. Software and Platforms

5.1. Common Communication Interface

Participant: Brice Goglin.

- The *Common Communication Interface* aims at offering a generic and portable programming interface for a wide range of networking technologies (Ethernet, InfiniBand, ...) and application needs (MPI, storage, low latency UDP, ...).
- CCI is developed in collaboration with the *Oak Ridge National Laboratory* and several other academics and industrial partners.
- CCI is in early development and currently composed of 19 000 lines of C.
- http://www.cci-forum.org

5.2. Hardware Locality

Participants: Brice Goglin, Samuel Thibault.

- *Hardware Locality* (HWLOC) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.
- It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics.
- HWLOC targets many types of high-performance computing applications [2], from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers already use HWLOC.
- HWLOC is developed in collaboration with the OPEN MPI project. The core development is still mostly performed by Brice GOGLIN and Samuel THIBAULT from the RUNTIME team-project, but many outside contributors are joining the effort, especially from the OPEN MPI and MPICH2 communities.
- HWLOC is composed of 30 000 lines of C.
- http://runtime.bordeaux.inria.fr/hwloc/

5.3. Network Locality

Participant: Brice Goglin.

- *Netloc Locality* (HWLOC) is a library that extends hwloc to network topology information by assembling hwloc knowledge of server internals within graphs of inter-node fabrics such as Ethernet or Infiniband.
- HWLOC targets the same challenges as hwloc but focuses on a wider spectrum by enabling cluster-wide solutions such process placement.
- HWLOC is developed in collaboration with University of Wisconsin in LaCrosse and Cisco, within the OPEN MPI project.
- NETLOC is composed of 15 000 lines of C.
- http://netloc.org

5.4. KNem

Participant: Brice Goglin.

- KNEM (*Kernel Nemesis*) is a Linux kernel module that offers high-performance data transfer between user-space processes.
- KNEM offers a very simple message passing interface that may be used when transferring very large messages within point-to-point or collective MPI operations between processes on the same node.
- Thanks to its kernel-based design, KNEM is able to transfer messages through a single memory copy, much faster than the usual user-space two-copy model.
- KNEM also offers the optional ability to offload memory copies on INTEL I/O AT hardware which improves throughput and reduces CPU consumption and cache pollution.
- KNEM is developed in collaboration with the MPICH2 team at the Argonne National Laboratory and the OPEN MPI project. These partners already released KNEM support as part of their MPI implementations.
- KNEM is composed of 8 000 lines of C. Its main contributor is Brice GOGLIN.
- http://runtime.bordeaux.inria.fr/knem/

5.5. Open-MX

Participant: Brice Goglin.

- The OPEN-MX software stack is a high-performance message passing implementation for any generic ETHERNET interface.
- It was developed within our collaboration with Myricom, Inc. as a part of the move towards the convergence between high-speed interconnects and generic networks.
- OPEN-MX exposes the raw ETHERNET performance at the application level through a pure message passing protocol.
- While the goal is similar to the old GAMMA stack [42] or the recent iWarp [41] implementations, OPEN-MX relies on generic hardware and drivers and has been designed for message passing.
- OPEN-MX is also wire-compatible with Myricom MX protocol and interface so that any application built for MX may run on any machine without Myricom hardware and talk other nodes running with or without the native MX stack.
- OPEN-MX is also an interesting framework for studying next-generation hardware features that could help ETHERNET hardware become legacy in the context of high-performance computing. Some innovative message-passing-aware stateless abilities, such as multiqueue binding and interrupt coalescing, were designed and evaluated thanks to OPEN-MX [5].
- Brice GOGLIN is the main contributor to OPEN-MX. The software is already composed of more than 45 000 lines of code in the Linux kernel and in user-space.
- http://open-mx.org/

5.6. StarPU

Participants: Olivier Aumage, Andra Hugo, Nathalie Furmento, Raymond Namyst, Marc Sergent, Samuel Thibault, Pierre-André Wacrenier.

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Xeon Phi processors.
- STARPU offers a unified offloadable task abstraction named codelet.In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (e.g. one function for CUDA and one function for CPUs).

- STARPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. A high-level data management library enforces memory coherency over the machine: before a codelet starts (e.g. on an accelerator), all its data are transparently made available on the compute resource.
- STARPU obtains portable performances by efficiently (and easily) using all computing resources at the same time.
- STARPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.
- STARPU can also leverage existing parallel implementations, by supporting *parallel tasks*, which can be run concurrently over the machine.
- STARPU provides *scheduling contexts* which can be used to partition computing resources. Scheduling contexts can be dynamically resized to optimize the allocation of computing resources among concurrently running libraries.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- STARPU comes with a plug-in for the GNU Compiler Collection (GCC), which extends languages of the C family with syntactic devices to describe STARPU's main programming concepts in a concise, high-level way.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- http://runtime.bordeaux.inria.fr/StarPU/

5.7. NewMadeleine

Participant: Alexandre Denis.

- NEWMADELEINE is communication library for high performance networks, based on a modular architecture using software components.
- The NEWMADELEINE optimizing scheduler aims at enabling the use of a much wider range of communication flow optimization techniques such as packet reordering or cross-flow packet aggregation.
- NEWMADELEINE targets applications with irregular, multiflow communication schemes such as found in the increasingly common application conglomerates made of multiple programming environments and coupled pieces of code, for instance.
- It is designed to be programmable through the concepts of optimization *strategies*, allowing experimentations with multiple approaches or on multiple issues with regard to processing communication flows, based on basic communication flows operations such as packet merging or reordering.
- The reference software development branch of the NEWMADELEINE software consists in 90 000 lines of code. NEWMADELEINE is available on various networking technologies: Myrinet, Infiniband, Quadrics and ETHERNET. It is developed and maintained by Alexandre DENIS.
- http://runtime.bordeaux.inria.fr/newmadeleine/

5.8. PadicoTM

Participant: Alexandre Denis.

- PadicoTM is a high-performance communication framework for grids. It is designed to enable various middleware systems (such as CORBA, MPI, SOAP, JVM, DSM, etc.) to utilize the networking technologies found on grids.
- PadicoTM aims at decoupling middleware systems from the various networking resources to reach transparent portability and flexibility.
- PadicoTM architecture is based on software components. Puk (the PadicoTM micro-kernel) implements a light-weight high-performance component model that is used to build communication stacks.
- PadicoTM component model is now used in NEWMADELEINE. It is the cornerstone for networking integration in the projects “LEGO” and “COOP” from the ANR.
- PadicoTM is composed of roughly 60 000 lines of C.
- PadicoTM is registered at the APP under number IDDN.FR.001.260013.000.S.P.2002.000.10000.
- http://runtime.bordeaux.inria.fr/PadicoTM/

5.9. MAQAO

Participants: Denis Barthou, Olivier Aumage, Tamara Meunier.

- MAQAO is a performance tuning tool for OpenMP parallel applications. It relies on the static analysis of binary codes and the collection of dynamic information (such as memory traces). It provides hints to the user about performance bottlenecks and possible workarounds.
- MAQAO relies on binary codes for Intel x86 and ARM architectures. For x86 architecture, it can insert probes for instrumention directly inside the binary. There is no need to recompile. The static/dynamic approach of MAQAO analysis is the main originality of the tool, combining performance model with values collected through instrumentation.
- MAQAO has a static performance model for x86 and ARM architectures. This model analyzes performance of the codes on the architectures and provides some feed-back hints on how to improve these codes, in particular for vector instructions.
- The dynamic collection of data in MAQAO enables the analysis of thread interactions, such as false sharing, amount of data reuse, runtime scheduling policy, ...
- MAQAO is in the European FP7 project ”MontBlanc” and in the Samsung GRO project ”Gepetto”.
- http://www.maqao.org/

5.10. QIRAL

Participants: Denis Barthou, Olivier Aumage.

- QIRAL is a high level language (expressed through LaTeX) that is used to described Lattice QCD problems. It describes matrix formulations, domain specific properties on preconditionings, and algorithms.
- The compiler chain for QIRAL can combine algorithms and preconditionings, checking validity of the composition automatically. It generates OpenMP parallel code, using libraries, such as BLAS.
- This code is developped in collaboration with other teams participating to the ANR PetaQCD project.

5.11. TreeMatch

Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.

- TREEMATCH is a library for performing process placement based on the topology of the machine and the communication pattern of the application.
- TREEMATCH provides a permutation of the processes to the processors/cores in order to minimize the communication cost of the application.
- Important features are : the number of processors can be greater than the number of applications processes ; it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speeds) ; it implements different placement algorithms that are switched according to the input size.
- Some core algorithms are parallel to speed-up the execution.
- TREEMATCH is integrated into various software such as the Charm++ programming environment as well as in both major open-source MPI implementations: Open MPI and MPICH2.
- TREEMATCH is available at: http://treematch.gforge.inria.fr.

6. New Results

6.1. SIMD Analysis Support in MAQAO

Either on ARM and x86 architectures, compilers and tools are needed for automatic and efficient vectorization. Although commercial compilers (e.g. IBM xlc, Intel icc, PGI pgcc) have made significant advances in auto-vectorization, a lot of source codes still remain too complicated for a compiler to vectorize, particularly when complex data structures are involved, or because of the lack of information at compile time. However, when vectorization fails, compilers leave the user with little clues about the cause of the failure, even though in certain cases moderate modifications could be applied on the source code to enable the compiler to vectorize.

Thus, the main objective for this work was to analyse SIMD vectorization potentials through loop detection. Parallelism detection is done through the instrumentation of the binary codes, capturing all memory streams in target loops and computing memory dependences using MAQAO. When combined to a static analysis for register dependences, this technique ensures that parallel slices of computation will be detected.

From a practical point of view, this work consists in the capture of the trace and its processing to extract memory reference patterns. To do so, we made use of the current state of the art MAQAO for instrumentation and trace capture on Intel architectures. We then implemented the dependence analysis on memory traces for performing loop pattern recognition. Finally, using this mechanism for loop pattern recognition, we can conclude about the vectorization potential of computation intensive loop nests. The dependence analysis does not depend on the target architecture, hence results computed for x86 architectures are valuable for ARM target as well.

6.2. NUMA-aware fine grain parallelization for multi-core architecture

Today, popular frameworks like Intel TBB or OpenMP offer a task based programming interface that allows to easily parallelize algorithms in shared memory. We have proposed some improvements to these task-based parallelization frameworks in order to cope with the problem of expressing an algorithm with a suitable task grain size and with the problem of Non Uniform Memory Accesses that degrades performance. In its current prototype state, our framework does not fully automate the selection of an optimal grain size. However, it significantly helps the programmer by proposing a simple interface to deal with DAG coarsening.

We have shown the benefits of this work on the parallelization of a sparse ILU preconditioner which is a challenging application with respect to task grain tuning and NUMA effect to an Intel TBB implementation. To improve even more the NUMA aspects, we are working on improving the task scheduler with cache-aware hierarchical scheduling support using a similar approach as the one implemented in the Bubblesched thread scheduler.

6.3. Task scheduling over heterogeneous architectures

We continued our work on extending STARPU to master exploitation of Heterogeneous Platforms through dynamic task scheduling, leading to the release of STARPU 1.1. We have extended our lightweight DSM to support out-of-core scheduling over disks. We have finished integrating STARPU with SIMGRID and obtained very accurate simulated times, which allows to experiment scheduling heuristics without having to actually execute the application on the target platform, thus tremendously reducing experimentation time and resource consumption.

We have modularized the scheduling part of STARPU, which permits to create complex schedulers by assembling simple scheduling components. This will allow theoreticians to work on writing the simple scheduling components without having to deal with the technical parts of the scheduling, performed in other scheduling components.

We have also collaborated with various research project to leverage the potential of STARPU: for instance, the PaStiX sparse matrix solver was ported over STARPU, so that we improved the dynamic task and management for applications with such fine-grain task size. This resulted with fair-enough performance on CPUs, compared to the hand-optimized static scheduler of PaStiX, and very promising performance on CPUs + GPUs. EADS ported its sparse hmatrix solver over STARPU, and we collaborated to work on adding STARPU support for communicating sparse data over MPI.

6.4. Task Size Control with XcalableMP/StarPU

On the work sharing among GPUs and CPU cores on GPU equipped clusters, it is a critical issue to select the task computational weights suited to these heterogeneous computing resources. We have been developing a solution for this problem, based on the cooperation of a PGAS language named XcalableMP (developed at the University of Tsukuba) together with a runtime sytem named XMP-dev/StarPU building on the work of the University of Tsukuba and on the StarPU platform developed by the Inria Runtime Team. Through the development, we found the necessity of adaptive task weight control for the GPU/CPU work sharing to achieve the best performance for various application codes. In particular, the language was extended to add a new feature allowing to alter the task size to be assigned to these heterogeneous resources dynamically during application execution. As a result of performance evaluation on several benchmarks, we confirmed the proposed feature correctly works and perform well even for relatively small size of problems.

6.5. Scheduling contexts for StarPU

Scheduling context is an extension of STARPU that allows multiple parallel codes to run concurrently with minimal interference. A scheduling context encapsulates an instance of the runtime system, and runs on top of a subset of the available processing units (i.e. regular cores or GPU accelerators). In order to maximize the overall efficiency of applications, contexts can be dynamically shrunk or expanded by a *hypervisor* that periodically gathers performance statistics inside each context (e.g. resource utilization, computation progress) and tries to determine how resources should be assigned to contexts so as to minimize the overall execution time. We have demonstrated the relevance of this approach using benchmarks invoking multiple high performance linear algebra kernels simultaneously on top of heterogeneous multicore machines. We have shown that our mechanism can dramatically improve the overall application run time (-34%), most notably by reducing the average cache miss ratio (-50%).

6.6. Load-balancing with TreeMatch

In the context of the Joint Laboratory for Petascale Computing (JLPC) included Inria and the University of Illinois at Urbana, we developed two load balancers for Charm++.

The two load-balancers we wrote take into account both the computing power and the hierarchical topology depending on the fact that the application is compute-bound or communication-bound. This work is based on our TREEMATCH library that computes process placement in order to reduce an application communication costs based on the hardware topology. Compared to some other solutions based on weighted topologies (latency, bandwidth, ...), ours is fully dynamic because we use only a qualitative approach for our representation of the hardware architecture.

The first load balancer is designed for compute-bound applications as it favours the leveling of CPU loads. The second load balancer focuses on communication-bound applications as it first reduces the congestion on the upper links in the topology tree.

These two load balancers gave us improvements for some applications up to 10% of the execution time.

6.7. List scheduling in embedded systems taking into account memroy constraints

Video decoding and image processing in embedded systems are subject to strong resource constraints, particularly in terms of memory. List-scheduling heuristics with static priorities (HEFT, SDC, etc.) being the often-cited solution due to both their good performance and their low complexity, we propose a method aimed at introducing the notion of memory into them. Moreover, we show that through appropriate adjustment of task priorities and judicious resort to insertion-based policy, speedups up to 20% can be achieved. Lastly, we show that our technique allows to prevent deadlock and to substantially reduce the required memory footprint compared to classic list-scheduling heuristics.

6.8. NewMadeleine generic multi-threading

The PIOMan progression engine utilized in NewMadeleine used to rely on the Marcel specific multi-threading library, with dedicated hooks and close co-operation between libraries. It restricted the target platforms and applications, and was considered as a constraint by users. We have designed mechanisms to make communication progress without hooks in the thread scheduler, able to run on any system with a `pthread` library. We have re-written PIOMan from the ground up to implement these mechanisms, and based on lock-free structures, with scalability in mind. A proof-of-concept port to the Intel Xeon Phi has been implemented in cooperation with the University of Tokyo, using the DCFA (Direct Communication Facility for manycore-based Accelerators) library to access InfiniBand boards from the Xeon Phi.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Contracts with Industry

SAMSUNG We have signed a contract with the Samsung company to work on the *Generation of Parallel Patterns based programs for hybrid CPU-GPU architectures* from october 2012 to september 2013.

7.2. Bilateral Grants with Industry

STMicroelectronics STMicroelectronics is granting the CIFRE PhD Thesis of Paul-Antoine Arras on *The development of a flexible heterogeneous system-on-chip platform using a mix of programmable processing elements and hardware accelerators* from October 2011 to October 2014.

TOTAL TOTAL is granting the CIFRE PhD thesis of Corentin Rossignon on *Sparse GMRES on heterogeneous platforms in oil extraction simulation* from april 2012 to march 2015.

CEA CEA is granting the CIFRE PhD thesis of Emmanuelle Saillard (2012-2015) on *Static/Dynamic Analysis for the validation and optimization of parallel applications* and Grégory Vaumourin (2013-2016) on *Hybrid Memory Hierarchy and Dynamic data optimization for embedded parallel architectures*

CEA - REGION AQUITAINE CEA together with the Aquitaine Region Council is funding the PhD thesis of Marc Sergent (2013-2016) on *Scalability for Task-based Runtimes*.

8. Partnerships and Cooperations

8.1. Regional Initiatives

REGION AQUITAINE The Aquitaine Region Council is granting the PhD thesis of Andra Hugo about *Composability of parallel software over hybrid architectures*, from september 2011 to august 2014.

REGION AQUITAINE The Aquitaine Region Council is granting the PhD thesis of Bertrand Putigny about *Performance Models for Heterogeneous Parallel Architectures*.

REGION AQUITAINE - CEA The Aquitaine Region Council together with CEA is funding PhD thesis of Marc Sergent (2013-2016) on *Scalability for Task-based Runtimes* (See also Section Bilateral Grants with Industry)

8.2. National Initiatives

8.2.1. ANR

ANR COOP Multi-level Cooperative Resource Management (http://coop.gforge.inria.fr/).

ANR COSINUS 2009 Program, 12/2009 - 06/2013 (42 months)

Identification: ANR-09-COSI-001

Coordinator: Christian Pérez (Inria Rhône-Alpes)

Other partners: Inria Bordeaux, Inria Rennes, IRIT, EDF R&D.

Abstract: COOP aims at establishing generic cooperation mechanisms between resource management, runtime systems, and application programming frameworks to simplify programming models, and improve performance through adaptation to the resources.

ANR SOLHAR (http://solhar.gforge.inria.fr/doku.php?id=start).

ANR MONU 2013 Program, 2013 - 2016 (36 months)

Identification: ANR-13-MONU-0007

Coordinator: Inria Bordeaux/LaBRI

Other partners: CNRS-IRIT, Inria-LIP Lyon, CEA/CESTA, EADS-IW

Abstract: This project aims at studying and designing algorithms and parallel programming models for implementing direct methods for the solution of sparse linear systems on emerging computers equipped with accelerators. The ultimate aim of this project is to achieve the implementation of a software package providing a solver based on direct methods for sparse linear systems of equations. Several attempts have been made to accomplish the porting of these methods on such architectures; the proposed approaches are mostly based on a simple offloading of some computational tasks (the coarsest grained ones) to the accelerators and rely on fine hand-tuning of the code and accurate performance modeling to achieve efficiency. This project proposes an innovative approach which relies on the efficiency and portability of runtime systems, such as the StarPU tool developed in the runtime team (Bordeaux). Although the SOLHAR project will focus on heterogeneous computers equipped with GPUs due to their wide availability and affordable cost, the research accomplished on algorithms, methods and programming models will be readily applicable to other accelerator devices such as ClearSpeed boards or Cell processors.

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

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

Identification: ANR-11INFR01306

Coordinator: Martin Quinson (Inria Nancy)

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

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

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

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

Coordinator: Denis Trystram (Inria Rhône-Alpes)

Other partners: Inria Bordeaux.

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

8.2.2. Inria Project Lab

8.2.2.1. C2S@Exa - Computer and Computational Scienecs at Exascale

Participant: Olivier Aumage [RUNTIME project-team, Inria Bordeaux - Sud-Ouest].

Since January 2013, the team is participating to the C2S@Exa http://www-sop.inria.fr/c2s_at_exa Inria Project Lab (IPL). This national initiative aims at the development of numerical modeling methodologies that fully exploit the processing capabilities of modern massively parallel architectures in the context of a number of selected applications related to important scientific and technological challenges for the quality and the security of life in our society. At the current state of the art in technologies and methodologies, a multidisciplinary approach is required to overcome the challenges raised by the development of highly scalable numerical simulation software that can exploit computing platforms offering several hundreds of thousands of cores. Hence, the main objective of C2S@Exa is the establishment of a continuum of expertise in the computer science and numerical mathematics domains, by gathering researchers from Inria project-teams whose research and development activities are tightly linked to high performance computing issues in these domains. More precisely, this collaborative effort involves computer scientists that are experts of programming models, environments and tools for harnessing massively parallel systems, algorithmists that propose algorithms and contribute to generic libraries and core solvers in order to take benefit from all the parallelism levels with the main goal of optimal scaling on very large numbers of computing entities and, numerical mathematicians that are studying numerical schemes and scalable solvers for systems of partial differential equations in view of the simulation of very large-scale problems.

8.2.2.2. MULTICORE - Large scale multicore virtualization for performance scaling and portability

Participant: Emmanuel Jeannot [RUNTIME project-team, Inria Bordeaux - Sud-Ouest].

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

8.3. European Initiatives

8.3.1. FP7 Projects

HPC-GA

Program: FP7 IRSES Marie-Curie

Project acronym: HPC-GA

Project title: High Performance Computing for Geophysics Applications

Duration: Jan 2012 - Dec 2014

Coordinator: Jean-François Méhaut (UJF, France)

Other partners: UFRGS, Inria, BCAM et UNAM.

Abstract: The design and implementation of geophysics applications on top of nowadays supercomputers requires a strong expertise in parallel programming and the use of appropriate runtime systems able to efficiently deal with heterogeneous architectures featuring many-core nodes typically equipped with GPU accelerators. The HPC-GA project aims at evaluating the functionalities provided by current runtime systems in order to point out their limitations. It also aims at designing new methods and mechanisms for an efficient scheduling of processes/threads and a clever data distribution on such platforms. The HPC-GA project is unique in gathering an international, pluridisciplinary consortium of leading European and South American researchers featuring complementary expertise to face the challenge of designing high performance geophysics simulations for parallel architectures.

MontBlanc2

Program: FP7 ICT-2013, Exascale Computing Platforms

Project acronym: MontBlanc2

Project title: European scalable and power efficient HPC platform based on low-power embedded technology

Duration: Oct 2013 - Nov 2016

Coordinator: Alex Ramirez (BSC, Spain)

Other partners: Inria, Bull, ST, ARM, Gnodal, Juelich, BADW-LRZ, HLRS, CNRS, CEA, CINECA, Bristol, Allinea

Abstract: The Mont-Blanc project aims to develop a European Exascale approach leveraging on commodity power-efficient embedded technologies. The project has developed a HPC system software stack on ARM, and will deploy the first integrated ARM-based HPC prototype by 2014, and is also working on a set of 11 scientific applications to be ported and tuned to the prototype system. The rapid progress of Mont-Blanc towards defining a scalable power efficient Exascale platform has revealed a number of challenges and opportunities to broaden the scope of investigations and developments. Particularly, the growing interest of the HPC community in accessing the Mont-Blanc platform calls for increased efforts to setup a production-ready environment. The Mont-Blanc 2 proposal has 4 objectives:

1. To complement the effort on the Mont-Blanc system software stack, with emphasis on programmer tools (debugger, performance analysis), system resiliency (from applications to architecture support), and ARM 64-bit support
2. To produce a first definition of the Mont-Blanc Exascale architecture, exploring different alternatives for the compute node (from low-power mobile sockets to special-purpose high-end ARM chips), and its implications on the rest of the system

3. To track the evolution of ARM-based systems, deploying small cluster systems to test new processors that were not available for the original Mont-Blanc prototype (both mobile processors and ARM server chips)
4. To provide continued support for the Mont-Blanc consortium, namely operations of the original Mont-Blanc prototype, the new small scale prototypes and hands-on support for our application developers

Mont-Blanc 2 contributes to the development of extreme scale energy-efficient platforms, with potential for Exascale computing, addressing the challenges of massive parallelism, heterogeneous computing, and resiliency. Mont-Blanc 2 has great potential to create new market opportunities for successful EU technology, by placing embedded architectures in servers and HPC..

8.3.2. Collaborations in European Programs, except FP7

COST ComplexHPC http://complexhpc.org

Program: COST Action IC0805

Project acronym: ComplexHPC

Project title: Open European Network for High-Performance Computing in Complex Environments

Duration: May 2009 - June 2013

Coordinator: Emmanuel Jeannot

Other partners: This Action gathers more than 20 countries and 30 partners in Europe.

Abstract: The goal of the Action is to establish a European research network focused on high performance heterogeneous computing in order to address the whole range of challenges posed by these new platforms including models, algorithms, programming tools and applications. The network will aim at contributing to exchange information, identify synergies and pursue common research activities, therefore reinforcing the strength of European research groups and the leadership of Europe in this field.

8.4. International Initiatives

8.4.1. Inria Associate Teams

MORSE Matrices Over Runtime Systems at Exascale

Inria Associate-Teams program: 2011-2016

Coordinator: Emmanuel Agullo (Hiepacs)

Parners: Inria (Runtime & Hiepacs), University of Tennessee Knoxville, University of Colorado Denver and KAUST.

Abstract: The Matrices Over Runtime Systems at Exascale (MORSE) associate team has vocation to design dense and sparse linear algebra methods that achieve the fastest possible time to an accurate solution on large-scale multicore systems with GPU accelerators, using all the processing power that future high end systems can make available. To develop software that will perform well on petascale and exascale systems with thousands of nodes and millions of cores, several daunting challenges have to be overcome both by the numerical linear algebra and the runtime system communities. With Inria Hiepacs, University of Tennessee, Knoxville and University of Colorado, Denver.

8.4.2. Inria International Labs

JLPC on Petascale Computing Inria joint-Lab

Coordinators: Franck Cappello and Marc Snir.

Other partners: Argonne National Lab, Inria, University of Urbanna Champaign.

Abstract: he Joint Laboratory is based at Illinois and includes researchers from Inria, Illinois' Center for Extreme-Scale Computation, and the National Center for Supercomputing Applications. It focuses on software challenges found in complex high-performance computers.

8.4.3. Participation in other International Programs

ANR-JST FP3C Framework and Programming for Post Petascale Computing.

ANR-JST 2010 Program, 01/09/2010 - 31/03/2014

Identification: ANR-10-JST-002

Coordinator: Serge Petiton (Inria Saclay)

Other partners: CNRS IRIT, CEA DEN Saclay, Inria Bordeaux, CNRS-Prism, Inria Rennes, University of Tsukuba, Tokyo Institute of Technology, University of Tokyo, Kyoto University.

Abstract: Post-petascale systems and future exascale computers are expected to have an ultra large-scale and highly hierarchical architecture with nodes of many-core processors and accelerators. That implies that existing systems, language, programming paradigms and parallel algorithms would have, at best, to be adapted. The overall structure of the FP3C project represents a vertical stack from a high level language for end users to low level architecture considerations, in addition to more horizontal runtime system researches.

HPC-GA High Performance Computing for Geophysics Applications (http://project.inria.fr/HPC-GA/)

European FP7 Programme, "Marie Curie" Action, PIRSES Scheme, 01/2012 - 12/2014 (36 months)

Identification: PIRSES-GA-2011-295217

Coordinator: Jean-François Méhaut (UJF)

Other Partners: Inria Grenoble, Inria Bordeaux, Basque Center for Applied Mathematics (BCAM, Bilbao, Spain), Federal University of Rio Grande do Sul (UFRGS, Porto Alegre, Brazil), Universidad Nacional Autónoma de México (UNAM, Mexico, Mexico), Bureau de Recherche Géologique et Minière (BRGM, Orléans, France), Grand Équipement National de Calcul Intensif (GENCI, France).

Abstract: The HPC-GA project is unique in gathering an international, pluridisciplinary consortium of leading European and South American researchers featuring complementary expertise to face the challenge of designing high performance geophysics simulations for parallel architectures: UFRGS, Inria, BCAM and UNAM. Results of this project will be validated using data collected from real sensor networks. Results will be widely disseminated through high-quality publications, workshops and summer-schools.

SEHLOC Scheduling evaluation in heterogeneous systems with hwloc

STIC-AmSud 2012 Program, 01/2013 - 12/2014 (24 months)

Coordinator: Brice Goglin

Other Partners: Universidad Nacional de San Luis (Argentina), Universidad de la Repúpublica (Uruguay).

Abstract: This project focuses on the development of runtime systems that combine application characteristics with topology information to automatically offer scheduling hints that try to respect hardware and software affinities. Additionally we want to analyze the convergence of the obtained performance from our algorithms with the recently proposed Multi-BSP model which considers nested levels of computations that correspond to natural layers of nowadays hardware architectures.

NextGN Preparing for Next Generation Numerical Simulation Platforms

PUF (Partner University Fund) - France USA, 01/2013 - 12-2016 (3 years)

Coordinator: Franck Capello, Marc Snir and Yves Robert

Other Partners: Inria, Argonne National Lab and University of Urbanna Chapaign

This PUF proposal builds on the existing successful joint laboratory between Inria and UIUC that has produced in past three years and half many top-level publications, some of which resulted in student awards; and several software packages that are making their way to production in Europe and USA. The proposal extends the collaboration to Argonne National Laboratory (ANL) and CNRS researchers who will bring their unique expertise and their skills to help addressing the scalability issue of simulation platforms.

9. Dissemination

9.1. Scientific Animation

Raymond NAMYST is the head of the LaBRI-CNRS “SATANAS” (*Runtime systems and algorithms for high performance numerical applications*) research team (about. 50 people) that includes the BACCHUS, HIEPACS, PHOENIX and RUNTIME Inria groups.

Raymond NAMYST was chairing the scientific committee of the ANR “Numerical Models” program for the 2011-2013 period.

Raymond NAMYST serves as an expert for the following initiatives/institutions:

- ETP4HPC (http://www.etp4hpc.eu/, in 2013) ;
- CEA/DAM (as a “scientific expert” for the 2008-2012 period) ;
- ORAP (ORganisation Associative du Parallélisme, since 2013) ;
- CEA-EDF-Inria School technical committee (since 2009) ;
- GENCI (http://www.genci.fr/?lang=en, since 2009) ;

Raymond NAMYST was a program committee member of the following international conferences: SC’13, PACT 2013, ROSS 2013, ICPP 2013, CASS 2013; CCGrid 2013, PPAM 2013.

Raymond NAMYST gave invited talks at the following international conferences/workshops: SC’13 (Birds of a Feather on "Dynamic Exascale Runtime Systems"), ORAP 2013 ("Programming Heterogeneous Architectures").

Samuel THIBAULT was a program committee member of IPDPS 2014

Brice GOGLIN was a program committee member of EuroMPI 2013, Hot Interconnects 2013, HIPC 2013.

Brice GOGLIN organized the CEA-EDF-Inria summer school on Programming Heterogeneous Parallel Architectures in Cadarache (June 2013).

Guillaume MERCIER was program committee member of EuroMPI 2013.

Olivier AUMAGE was reviewer for the ACM TACO journal and for the EuroPar 2013, ICPP 2013, ROSS 2013, and IPDPS 2014 conferences and workshops. He also reviewed one project submission for the 2013 ANR MN funding call. He is part of the Inria Bordeaux – Sud-Ouest committee for scientific event fundings.

Olivier AUMAGE was an invited teacher for the European COST-funded ComplexHPC school in Uppsala, Sweden. He also was an invited speaker at the International Conquest Workshop organized by the Theoretical Chemistry and Modeling Group of the Institute for Molecular Science from the University of Bordeaux.

Emmanuel JEANNOT was program committee member for: Euro-MPI 2013, CCGRID 2013, Heteropar 2013, PPAM 2013, IPDPS 2014.

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

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

Emmanuel JEANNOT was reviewer for the following journals: JPDC, IEEE TPDS, Parallel Computing.

Emmanuel JEANNOT has given an invited talk at the JLPC in Urbana and at the ComplexHPC Spring School 2013 on "Heterogeneous computing - impact on algorithms - ".

Denis BARTHOU was program committee member for: Euro-Par 2013, IPDPS 2013, PROPER 2013, UCHPC 2013. He is part of the Inria Bordeaux – Sud-Ouest committee for Young Researchers. He is member of the Governing Board of the LaBRI and of the board of directors of the Institut Polytechnique de Bordeaux (IPB). Denis BARTHOU is scientific expert for the Exascale Computing Research Laboratory (since 2009).

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

Members of RUNTIME project gave thousands of hours of teaching at University of Bordeaux and ENSEIRB-MATMECA engineering schools, covering a wide range of topics from basic use of computers and C programming to advance topics such as operating systems, parallel programming and high-performance runtime systems.

9.2.2. Supervision

PhD: Sylvain HENRY, Programming Models and Runtime Systems for Heterogeneous Architectures, 2013/11, Denis BARTHOU and Alexandre DENIS

PhD: Cyril BORDAGE, Parallélisation de la méthode multipôle sur architecture hybride, 2013/11, Raymond NAMYST and David GOUDIN (CEA CESTA)

PhD: Alexandre DUCHATEAU, Automatic Algorithm Derivation and Exploration in Linear Algebra for Parallelism and Locality, 2013/03, Denis BARTHOU and David PADUA (UIUC)

PhD in progress : Bertrand PUTIGNY, Modèles de performance pour l'ordonnancement sur architectures multicoeurs hétérogènes, 2010/11, Brice GOGLIN and Denis BARTHOU

PhD in progress : François TESSIER, Placement d'applications hybrides sur machine non-uniformes multicœurs, 2011/10 Emmanuel JEANNOT and Guillaume MERCIER

PhD in progress : Paul-Antoine ARRAS, Development of a Flexible Heterogeneous System-On-Chip Platform using a mix of programmable Processing Elements and harware accelerators. 2011/10, Emmanuel JEANNOT and Samuel THIBAULT

PhD in progress: Andra HUGO, Composability of parallel codes over heterogeneous platforms, 2013/10, Abdou GUERMOUCHE and Pierre-André WACRENIER and Raymond NAMYST

PhD in progress: Corentin ROSSIGNON, Design of an object-oriented runtime system for oil reserve simulations on heterogeneous architectures, 2012/04, Olivier AUMAGE and Pascal HÉNON (TOTAL) and Raymond NAMYST and Samuel THIBAULT

PhD in progress: Emmanuelle SAILLARD, Analyse statique/dynamique/itérative pour la validation et l'amélioration des applications parallèles multi-modèles sur supercalculateur hybride de type cluster de CPUs/GPUs, 2012/10, Patrick CARRIBAULT (CEA/DAM), Denis BARTHOU

PhD in progress: Grégory VAUMOURIN, Hiérarchie mémoire hybride et gestion dynamique de données dans les architectures parallèles embarquées, 2013/10, Thomas DOMBEK (CEA/DACLE), Denis BARTHOU

PhD in progress: Soufiane BAGHDADI, Collaboration entre compilateur et support d'exécution pour les applications parallèles 2011/10, Elisabeth BRUNET (Telecom SudParis), Jean-François TRAHAY (Telecom SudParis) , Denis BARTHOU

PhD in progress: Marc SERGENT, Passage à l'échelle de moteur d'exécution à base de graphes de tâches, 2013/09, Olivier AUMAGE, David GOUDIN (CEA/CESTA), Samuel THIBAULT, Raymond NAMYST

PhD in progress: Suraj KUMAR, Stratégies d'ordonnancement dynamique pour l'algèbre linéaire dense, 2013/12, Emmanuel AGULLO, Olivier BEAUMONT, Samuel THIBAULT

PhD in progress: Pei LI, High-Performance Code Generation for Stencil Computations on Heterogeneous Multi-device Architectures, 2012/10, Raymond NAMYST, Elisabeth BRUNET (Telecom SudParis)

9.2.3. Juries

Raymond NAMYST was member of the PhD defense jury for the following candidates:

- Jean-Yves VET (University Pierre et Marie Curie Paris, reviewer)

Denis BARTHOU was member of PhD defense jury of the following candidates:

- Amira MENSI (Mines de Paris, reviewer)
- Yuryi KACHNIKOV (UVSQ, reviewer)
- Jose NOUDOHOUENOU (UVSQ, president)
- Jean-Marc GRATIEN (UJF, president)

9.3. Popularization

Brice GOGLIN is in charge of the diffusion of the scientific culture for the Inria Research Center of Bordeaux. He is also a member of the national Inria committee on Scientific Mediation. He gave numerous talks about high performance computing and research careers to general public audience and school student, as well as several radio and paper interviews about Inria's activities.

Brice GOGLIN and Bertrand PUTIGNY wrote two popularization papers about High-Performance Computing in the Interstices online journal.

Brice GOGLIN and François TESSIER presented research careers at the Aquitec student exhibition.

Paul-Antoine ARRAS, Sylvain HENRY, Bertrand PUTIGNY and François TESSIER gave a hands-on introduction to programming to 30 teenagers at the *Fete de la Science*. Brice GOGLIN was in charge of a numeric science popularization game during that same event.

Samuel THIBAULT presented his research work in a local elementary school.

10. Bibliography

Major publications by the team in recent years

[1] C. AUGONNET, S. THIBAULT, R. NAMYST, P.-A. WACRENIER. *StarPU: A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures*, in "Concurrency and Computation: Practice and Experience, Special Issue: Euro-Par 2009", February 2011, vol. 23, pp. 187–198 [*DOI :* 10.1002/CPE.1631], http://hal.inria.fr/inria-00550877

[2] F. BROQUEDIS, J. CLET-ORTEGA, S. MOREAUD, N. FURMENTO, B. GOGLIN, G. MERCIER, S. THIBAULT, R. NAMYST. *hwloc: a Generic Framework for Managing Hardware Affinities in HPC Applications*, in "Proceedings of the 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP2010)", Pisa, Italia, IEEE Computer Society Press, February 2010, pp. 180–186 [*DOI :* 10.1109/PDP.2010.67], http://hal.inria.fr/inria-00429889

[3] F. BROQUEDIS, N. FURMENTO, B. GOGLIN, P.-A. WACRENIER, R. NAMYST. *ForestGOMP: an efficient OpenMP environment for NUMA architectures*, in "International Journal on Parallel Programming, Special Issue on OpenMP; Guest Editors: Matthias S. Müller and Eduard Ayguadé", 2010, vol. 38, n^o 5, pp. 418-439 [*DOI :* 10.1007/S10766-010-0136-3], http://hal.inria.fr/inria-00496295

[4] D. Buntinas, G. Mercier, W. Gropp. *Implementation and Shared-Memory Evaluation of MPICH2 over the Nemesis Communication Subsystem*, in "Recent Advances in Parallel Virtual Machine and Message Passing Interface: Proc. 13th European PVM/MPI Users Group Meeting", Bonn, Germany, September 2006

[5] B. Goglin, N. Furmento. *Finding a Tradeoff between Host Interrupt Load and MPI Latency over Ethernet*, in "Proceedings of the IEEE International Conference on Cluster Computing", New Orleans, LA, IEEE Computer Society Press, September 2009, http://hal.inria.fr/inria-00397328

[6] B. Goglin. *High-Performance Message Passing over generic Ethernet Hardware with Open-MX*, in "Journal of Parallel Computing", February 2011, vol. 37, n^{o} 2, pp. 85-100 [*DOI :* 10.1016/j.parco.2010.11.001], http://hal.inria.fr/inria-00533058/en

[7] S. Thibault, R. Namyst, P.-A. Wacrenier. *Building Portable Thread Schedulers for Hierarchical Multiprocessors: the BubbleSched Framework*, in "EuroPar", Rennes,France, ACM, 8 2007, http://hal.inria.fr/inria-00154506

[8] F. Trahay, É. Brunet, A. Denis, R. Namyst. *A multithreaded communication engine for multicore architectures*, in "CAC 2008: Workshop on Communication Architecture for Clusters, held in conjunction with IPDPS 2008", Miami, FL, IEEE Computer Society Press, April 2008, http://hal.inria.fr/inria-00224999

Publications of the year

Articles in International Peer-Reviewed Journals

[9] 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", December 2013, LPT-Orsay-13-142, http://hal.inria.fr/hal-00926513

[10] B. Goglin, S. Moreaud. *KNEM: a Generic and Scalable Kernel-Assisted Intra-node MPI Communication Framework*, in "Journal of Parallel and Distributed Computing", February 2013, vol. 73, n^{o} 2, pp. 176-188 [*DOI :* 10.1016/j.jpdc.2012.09.016], http://hal.inria.fr/hal-00731714

[11] E. Jeannot. *Symbolic Mapping and Allocation for the Cholesky Factorization on NUMA machines: Results and Optimizations*, in "International Journal of High Performance Computing Applications", 2013, vol. 27, n^{o} 3, pp. 283–290, http://hal.inria.fr/hal-00921611

[12] E. Jeannot, G. Mercier, F. Tessier. *Process Placement in Multicore Clusters: Algorithmic Issues and Practical Techniques*, in "IEEE Transactions on Parallel and Distributed Systems", May 2013, http://hal.inria.fr/hal-00921605

International Conferences with Proceedings

[13] G. Antoniu, T. Boku, C. Calvin, P. Codognet, M. Dayde, N. Emad, Y. Ishikawa, S. Matsuoka, K. Nakajima, H. Nakashima, R. Namyst, S. Petiton, T. Sakurai, M. Sato. *Towards exascale with the ANR-JST japanese-french project FP3C (Framework and Programming for Post- Petascale Computing)*, in "9th International Conference on Computer Science and Information Technologies", Yerevan, Armenia, 2013, http://hal.inria.fr/hal-00922754

[14] P.-A. Arras, D. Fuin, E. Jeannot, A. Stoutchinin, S. Thibault. *List Scheduling in Embedded Systems under Memory Constraints*, in "SBAC-PAD'2013 - 25th International Symposium on Computer

Architecture and High-Performance Computing", Porto de Galinhas, Brazil, J. GUERRERO (editor), IEEE Computer Society, October 2013, http://hal.inria.fr/hal-00906117

[15] O. AUMAGE, D. BARTHOU, C. HAINE, T. MEUNIER. *Detecting SIMDization Opportunities through Static/Dynamic Dependence Analysis*, in "PROPER - 6th Workshop on Productivity and Performance - 2013", Aachen, Germany, September 2013, http://hal.inria.fr/hal-00858004

[16] A. CHARIF-RUBIAL, D. BARTHOU, C. VALENSI, S. SAMEER, A. MALONY, W. JALBY. *MIL : A language to build program analysis tools through static binary instrumentation*, in "High Performance Computing", India, 2013, pp. 206-215, http://hal.inria.fr/hal-00920875

[17] A. DUCHÂTEAU, D. PADUA, D. BARTHOU. *Hydra: Automatic algorithm exploration from linear algebra equations*, in "Code Generation and Optimization", Shenzhen, China, 2013, pp. 1-10, http://hal.inria.fr/hal-00920869

[18] S. HENRY. *ViperVM: a Runtime System for Parallel Functional High-Performance Computing on Heterogeneous Architectures*, in "2nd Workshop on Functional High-Performance Computing (FHPC'13)", Boston, United States, September 2013, http://hal.inria.fr/hal-00851122

[19] A.-E. HUGO, A. GUERMOUCHE, R. NAMYST, P.-A. WACRENIER. *Composing multiple StarPU applications over heterogeneous machines: a supervised approach*, in "Third International Workshop on Accelerators and Hybrid Exascale Systems", Boston, United States, May 2013, http://hal.inria.fr/hal-00824514

[20] A.-E. HUGO. *Le problème de la composition parallèle : une approche supervisée*, in "RenPAR - 21e Rencontres Francophones du Parallélisme (2013)", Grenoble, France, January 2013, http://hal.inria.fr/hal-00773610

[21] E. JEANNOT, E. MENESES, G. MERCIER, F. TESSIER, G. ZHENG. *Communication and Topology-aware Load Balancing in Charm++ with TreeMatch*, in "IEEE Cluster 2013", Indianapolis, United States, IEEE, September 2013, http://hal.inria.fr/hal-00851148

[22] P. LI, E. BRUNET, R. NAMYST. *High Performance Code Generation for Stencil Computation on Heterogeneous Multi-device Architectures*, in "HPCC-15th IEEE International Conference on High Performance Computing and Communications", Zhangjiajie, China, IEEE Computer Society, 2013, http://hal.inria.fr/hal-00925481

[23] A. MAZOUZ, S.-A.-A. TOUATI, D. BARTHOU. *Dynamic Thread Pinning for Phase-Based OpenMP Programs*, in "The Euro-Par 2013 conference", Aachen, Germany, F. WOLF, B. MOHR, D. AN MEY (editors), Lecture Notes in Computer Science, Springer, August 2013, vol. 8097, pp. 53-64 [*DOI :* 10.1007/978-3-642-40047-6_8], http://hal.inria.fr/hal-00847482

[24] T. ODAJIMA, T. BOKU, M. SATO, T. HANAWA, Y. KODAMA, R. NAMYST, S. THIBAULT, O. AUMAGE. *Adaptive Task Size Control on High Level Programming for GPU/CPU Work Sharing*, in "The 2013 International Symposium on Advances of Distributed and Parallel Computing (ADPC 2013)", Vietri sul Mare, Italy, December 2013, http://hal.inria.fr/hal-00920915

[25] S. OHSHIMA, S. KATAGIRI, K. NAKAJIMA, S. THIBAULT, R. NAMYST. *Implementation of FEM Application on GPU with StarPU*, in "SIAM CSE13 - SIAM Conference on Computational Science and Engineering 2013", Boston, United States, SIAM, February 2013, http://hal.inria.fr/hal-00926144

[26] C. Rossignon, H. Pascal, O. Aumage, S. Thibault. *A NUMA-aware fine grain parallelization framework for multi-core architecture*, in "PDSEC - 14th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing - 2013", Boston, United States, May 2013, http://hal.inria.fr/hal-00858350

[27] E. Saillard, P. Carribault, D. Barthou. *Combining Static and Dynamic Validation of MPI Collective Communication*, in "EuroMPI 2013", Madrid, Spain, September 2013, pp. 117-122 [*DOI :* 10.1145/2488551.2488555], http://hal.inria.fr/hal-00920901

National Conferences with Proceedings

[28] P.-A. Arras, D. Fuin, E. Jeannot, A. Stoutchinin, S. Thibault. *Ordonnancement de liste dans les systèmes embarqués sous contrainte de mémoire*, in "ComPAS'13 / RenPar'21 - 21es Rencontres francophones du Parallélisme", Grenoble, France, Inria Grenoble, January 2013, http://hal.inria.fr/hal-00772854

[29] E. Jeannot, G. Mercier, F. Tessier. *TreeMatch : Un algorithme de placement de processus sur architectures multicœurs*, in "RenPAR - 21e Rencontres Francophones du Parallélisme", Grenoble, France, January 2013, http://hal.inria.fr/hal-00773254

[30] C. Rossignon. *Optimisation du produit matrice-vecteur creux sur architecture GPU pour un simulateur de reservoir*, in "ComPAS'13 / RenPar'21 - 21es Rencontres francophones du Parallélisme", Grenoble, France, Inria Grenoble, 2013, http://hal.inria.fr/hal-00773571

Scientific Books (or Scientific Book chapters)

[31] T. Hoefler, E. Jeannot, G. Mercier. *An Overview of Process Mapping Techniques and Algorithms in High-Performance Computing*, in "High Performance Computing on Complex Environments", E. Jeannot, J. Zilinskas (editors), Wiley, 2014, pp. 65–84, To be published, http://hal.inria.fr/hal-00921626

Books or Proceedings Editing

[32] E. Jeannot, J. Zvilinskas (editors). , *High Performance Computing on Complex Environments*, Wiley, 2014, 499 p. , to be published, http://hal.inria.fr/hal-00921619

Research Reports

[33] D. Barthou, G. Grosdidier, K. Petrov, M. Kruse, C. Eisenbeis, O. Pène, O. Brand-Foissac, C. Tadonki, R. Dolbeau. , *Automated Code Generation for Lattice QCD Simulation*, Inria, December 2013, n^o RR-8417, 13 p. , http://hal.inria.fr/hal-00918812

[34] L. Courtès. , *C Language Extensions for Hybrid CPU/GPU Programming with StarPU*, Inria, April 2013, n^o RR-8278, 25 p. , http://hal.inria.fr/hal-00807033

[35] S. Henry, D. Barthou, A. Denis, R. Namyst, M.-C. Counilh. , *SOCL: An OpenCL Implementation with Automatic Multi-Device Adaptation Support*, Inria, August 2013, n^o RR-8346, 18 p. , http://hal.inria.fr/hal-00853423

[36] E. Jeannot, G. Mercier, F. Tessier. , *Process Placement in Multicore Clusters: Algorithmic Issues and Practical Techniques*, Inria, March 2013, n^o RR-8269, 32 p. , http://hal.inria.fr/hal-00803548

[37] X. LACOSTE, M. FAVERGE, P. RAMET, S. THIBAULT, G. BOSILCA. , *Taking advantage of hybrid systems for sparse direct solvers via task-based runtimes*, Inria, January 2014, n^{o} RR-8446, 25 p. , http://hal.inria.fr/hal-00925017

[38] A. ROUSSEAU, A. DARNAUD, B. GOGLIN, C. ACHARIAN, C. LEININGER, C. GODIN, C. HOLIK, C. KIRCHNER, D. RIVES, E. DARQUIE, E. KERRIEN, F. NEYRET, F. MASSEGLIA, F. DUFOUR, G. BERRY, G. DOWEK, H. ROBAK, H. XYPAS, I. ILLINA, I. GNAEDIG, J. JONGWANE, J. EHREL, L. VIENNOT, L. GUION, L. CALDERAN, L. KOVACIC, M. COLLIN, M.-A. ENARD, M.-H. COMTE, M. QUINSON, M. OLIVI, M. GIRAUD, M. DORÉMUS, M. OGOUCHI, M. DROIN, N. LACAUX, N. ROUGIER, N. ROUSSEL, P. GUITTON, P. PETERLONGO, R.-M. CORNUS, S. VANDERMEERSCH, S. MAHEO, S. LEFEBVRE, S. BOLDO, T. VIÉVILLE, V. POIREL, A. CHABREUIL, A. FISCHER, C. FARGE, C. VADEL, I. ASTIC, J.-P. DUMONT, L. FÉJOZ, P. RAMBERT, P. PARADINAS, S. DE QUATREBARBES, S. LAURENT. , *Médiation Scientifique : une facette de nos métiers de la recherche*, March 2013, 34 p. , http://hal.inria.fr/hal-00804915

Scientific Popularization

[39] B. GOGLIN. *Les réseaux pour le calcul haute performance : facteur, livreur ou déménageur ?*, in "Interstices", December 2013, http://hal.inria.fr/hal-00915723

[40] B. GOGLIN, B. PUTIGNY. *Idée reçue: Comparer la puissance de deux ordinateurs, c'est facile !*, in "Interstices", April 2013, http://hal.inria.fr/hal-00816422

References in notes

[41] P. BALAJI, H.-W. JIN, K. VAIDYANATHAN, D. K. PANDA. *Supporting iWARP Compatibility and Features for Regular Network Adapters*, in "Proceedings of the Workshop on Remote Direct Memory Access (RDMA): Applications, Implementations, and Technologies (RAIT); held in conjunction with the IEEE International Confer ence on Cluster Computing", Boston, MA, September 2005

[42] G. CIACCIO, G. CHIOLA. *GAMMA and MPI/GAMMA on GigabitEthernet*, in "Proceedings of 7th EuroPVM-MPI conference", Balatonfured, Hongrie, Lecture Notes in Computer Science, Springer Verlag, Septembre 2000, vol. 1908

[43] G. R. GAO, T. STERLING, R. STEVENS, M. HERELD, W. ZHU. *Hierarchical multithreading: programming model and system software*, in "20th International Parallel and Distributed Processing Symposium (IPDPS)", April 2006