IN PARTNERSHIP WITH:
CNRS

Université de Bordeaux

Activity Report 2012

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, GPU, Programming Languages

Creation of the Project-Team: October 07, 2004 .

1. Members

Research Scientists

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

Faculty Members

Raymond Namyst [University Bordeaux 1, Professor, Team Leader, HdR]
Denis Barthou [Professor, IPB, HdR]
Marie-Christine Counilh [Assistant Professor, University of Bordeaux]
Guillaume Mercier [Assistant Professor, IPB]
Samuel Thibault [Assistant Professor, University of Bordeaux]
Pierre-André Wacrenier [Assistant Professor, University of Bordeaux]

Engineers

Nicolas Collin [Associate Engineer, Inria, European Project grant]
Nathalie Furmento [Research Engineer, CNRS]
Cyril Roelandt [Associate Engineer, Inria, ANR grant]
Ludovic Stordeur [Associate Engineer, Inria, ANR grant]
Ludovic Courtès [Research Engineer, Inria]
Sébastien Barascou [Associate Engineer, Inria, ANR grant]

PhD Students

Paul-Antoine Arras [University of Bordeaux, STMicroelectronics-Inria CIFRE]
Cyril Bordage [University of Bordeaux, CEA grant]
Andres Charif-Rubial [University of Versailles, ANR grant]
Jérôme Clet-Ortega [University of Bordeaux, MESR grant]
Sylvain Henry [University of Bordeaux, MESR grant]
Andra Hugo [University of Bordeaux, MESR grant]
Bertrand Putigny [University of Bordeaux, Inria grant]
Corentin Rossignon [University of Bordeaux, TOTAL CIFRE]
François Tessier [University of Bordeaux, MESR grant]

Administrative Assistant

Sylvie Embolla

2. Overall Objectives

2.1. Designing Efficient Runtime Systems

parallel,runtime,environment,heterogeneity,SMP,multicore,NUMA,HPC,high-speed networks,protocols,MPI,scheduling,thread,O
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 5.7 is used for dynamic scheduling by the state-of-the art dense linear algebra library, Magma v1.1 http://icl.cs.utk.edu/magma/ .

3. Scientific Foundations

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.* [36], 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. In the ANR project PetaQCD, 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 are submitted to publication.

For parallel programs running on multicores, measuring reliable performance and determining performance stability is becoming a key issue: indeed, a number of hardware mechanisms may cause performance instability from one run to the other. Thread migration, memory contention (on any level of the cache hierarchy), scheduling policy of the runtime can introduce some variation, indenpendently of the program input. A speed-up is interesting only it corresponds to a performance that can be obtained through repeated execution of the application. Very few research efforts have been made in the identification of program optimization/runtime policy/hardware mechanisms that may introduce performance instability. We studied in [37] on a large set of OpenMP benchmarks performance variations, identified the mechanisms causing them and showing the need for better strategies for measuring speed-ups. Following this effort, we developped inside the tool MAQAO (Modular Assembler Quality Analyzer and Optimizer), the precise analysis of the interactions between OpenMP threads, through static analysis of binary codes and memory tracing. In particular, the influence of thread affinity is estimated and the tool proposes hints to the user to improve its OpenMP codes.

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.

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.

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

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 40 000 lines of C.
- http://runtime.bordeaux.inria.fr/hwloc/

5.3. 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 7000 lines of C. Its main contributor is Brice GOGLIN.
- http://runtime.bordeaux.inria.fr/knem/

5.4. Marcel

Participants: Olivier Aumage, Yannick Martin, Samuel Thibault.

- MARCEL is the two-level thread scheduler (also called N:M scheduler) of the PM^2 software suite.
- The architecture of MARCEL was carefully designed to support a large number of threads and to efficiently exploit hierarchical architectures (e.g. multicore chips, NUMA machines).
- MARCEL provides a *seed* construct which can be seen as a precursor of thread. It is only when the time comes to actually run the seed that MARCEL attempts to reuse the resources and the context of another, dying thread, significantly saving management costs.
- In addition to a set of original extensions, MARCEL provides a POSIX-compliant interface which thus permits to take advantage of it by just recompiling unmodified applications or parallel programming environments (API compatibility), or even by running already-compiled binaries with the Linux NPTL ABI compatibility layer.
- For debugging purpose, a trace of the scheduling events can be recorded and used after execution for generating an animated movie showing a replay of the execution.
- The MARCEL thread scheduling library is made of 80 000 lines of code.
- http://runtime.bordeaux.inria.fr/marcel/
- Marcel has been supported for 2 years (2009-2011) by the Inria ADT Visimar.

5.5. ForestGOMP

Participants: Olivier Aumage, Yannick Martin, Pierre-André Wacrenier.

- FORESTGOMP is an OPENMP environment based on both the GNU OPENMP run-time and the MARCEL thread library.
- It is designed to schedule efficiently nested sets of threads (derived from nested parallel regions) over hierarchical architectures so as to minimize cache misses and NUMA penalties.
- The FORESTGOMP runtime generates nested MARCEL bubbles each time an OPENMP parallel region is encountered, thereby grouping threads sharing common data.
- Topology-aware scheduling policies implemented by BUBBLESCHED can then be used to dynamically map bubbles onto the various levels of the underlying hierarchical architecture.
- FORESTGOMP allowed us to validate the BUBBLESCHED approach with highly irregular, fine grain, divide-and-conquer parallel applications.
- http://runtime.bordeaux.inria.fr/forestgomp/

5.6. 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 [35] or the recent iWarp [34] 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/

5.7. StarPU

Participants: Cédric Augonnet, Olivier Aumage, Nicolas Collin, Nathalie Furmento, Cyril Roelandt, Ludovic Stordeur, Samuel Thibault, Ludovic Courtès.

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Cell 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 a *reduction* mode, which permit to further optimize data management when results have to be reduced.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- 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.
- http://runtime.bordeaux.inria.fr/StarPU/

5.8. NewMadeleine

Participants: Alexandre Denis, François Trahay, Raymond Namyst.

- 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/

5.9. 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/

5.10. MAQAO

Participants: Denis Barthou, Andres Charif-Rubial.

- 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 and inserts 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 architecture and Itanium. This model analyzes performance of the predecoder, of the decoder and of the different pipelines of the x86 architecture, in particular for SSE 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 project "ProHMPT" from the ANR. A demo of MAQAO has been made in Jan. 2010 for SME/Inria days and in Nov. 2010 at SuperComputing, Inria Booth.
- http://www.maqao.org/

5.11. QIRAL

Participant: Denis Barthou.

- 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.

5.12. 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.
- 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. Mastering Heterogeneous Platforms

Participants: Cedric Augonnet, Olivier Aumage, Nicolas Collin, Ludovic Courtès, Nathalie Furmento, Sylvain Henry, Andra Hugo, Raymond Namyst, Cyril Roelandt, Corentin Rossignon, Ludovic Stordeur, Samuel Thibault, Pierre-André Wacrenier.

- We continued our work on extending StarPU to master exploitation of Heterogeneous Platforms.
- We have released version 1.0.0 of StarPU, now really considered a stable project that a lot of collaborators can base their work on.
- We have extended our lightweight DSM over MPI to support caching data [17], which dramatically reduces data transfers for classical applications.
- We have extended the StarPU scheduler to let the application provide several implementations of a function for the same architecture, implementation choice being performed by the scheduler according to actually measured performance, energy consumption, etc.
- We have collaborated with Computer Graphics research team in the MediaGPU project to make it possible to directly graphically render results from StarPU computations.
- Work has been initiated to integrate StarPU and SimGrid for the SONGS project, which will allow to simulate application execution on heterogeneous architectures, and thus easily experiment with scheduling strategies.
- We have extended StarPU with a protocol that permits to make it run with a master-slave model, which allowed to easily port it to the Intel SCC and Intel Xeon Phi processors, and will allow an easy load balancing support over MPI.
- We have extended StarPU to allow multiple parallel codes to run concurrently with minimal interference. Such parallel codes run within *scheduling contexts* that provide confined execution environments 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. We introduced a *hypervisor* that automatically expands or shrinks contexts using feedback from the runtime system (e.g. resource utilization).

  We demonstrated the relevance of our approach using benchmarks invoking multiple high performance linear algebra kernels simultaneously on top of heterogeneous multicore machines. We showed that our mechanism can dramatically improve the overall application run time (-34%), most notably by reducing the average cache miss ratio (-50%).
- We have improved [15] the OpenCL implementation on top of StarPU (SOCL) to allow applications to use StarPU's scheduling contexts through OpenCL's contexts and to explicitly schedule some kernels to enhance performance. Moreover, SOCL fully supports the OpenCL ICD extension and can now be dynamically selected amongst other available platforms which makes it easier to use.
- We have continued collaborations on applications on top of StarPU with the University of Mons [14], the University of Vienna [20], the University of Linköping, the University of Tsukuba, TOTAL, the CEA INAC in Grenoble and the BRGM French public institution in Earth science applications.
- In a joint work with French SME company CAPS entreprise, as part of the ANR ProHMPT project, we have demonstrated a proof of concept framework enabling three kinds of pieces of applicative code — a native StarPU code, a Magma/StarPU code and a HMPP/StarPU code annotated with HMPP's directives — to integrate and cooperate together on a computation as a single coherent application.

- As part of the HPC-GA project, we initiated a preliminary study with University of Rio Grande do Sul (UFRGS), Brazil, to cooperate on the modeling of common computing kernel tasks and potentially making use of kernel models designed at UFRGS within the StarPU's task cost evaluation framework.
- As part of the partnership with Total, and in relationship with StarPU's task scheduling work, we have explored solutions to semi-automatically adapt the grain of elementary tasks to the available computing resources.

6.2. High-Performance Intra-node Collective Operations

Participant: Brice Goglin.

- KNEM is known to improve the performance of point-to-point intra-node MPI communication significantly [13].
- We designed an extended RMA interface in KNEM that suits the needs of point-to-point, collective and RMA operations.
- We showed that the native use of KNEM in MPI collective implementations enabled further optimization by combining the knowledge of collective algorithms with the mastering of KNEM region management and copies.
- This work was initiated in the context of our collaboration with the MPICH2 team and is now also pursued within the OPEN MPI project in collaboration with the University of Tennessee in Knoxville.

6.3. Process Placement and Topology-Aware Computing

Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.

- TREEMATCH's limitations have been addressed. In particular, it is now able to handle unbalanced physical topologies.
- TREEMATCH has been compared to various competitors. We carried out various experiments that showed that TREEMATCH outperforms other solutions based on graph partitioning or graph embedding. These experiments also showed the limitations of some existing solutions (Scotch for instance).
- TREEMATCH has been integrated into several major parallel programming environments. It is implemented as a load-balancer in Charm++ (François TESSIER made severals at UI Urbana Champaign) and is used to enhance topology management routines in Open MPI and MPICH2. It is indeed employed to allow rank reordering in functions such as `MPI_Dist_graph_create` for instance. This work started with a visit at UTK by Guillaume MERCIER.
- We set-up several collaborations: besides the collaboration with the Open MPI group, we also work with the CERFACS in order to speed-up existing CFD parallel applications developed by this group.

6.4. Thread placement and memory allocation on NUMA machines

Participant: Emmanuel Jeannot.

We have worked on optimizing the tiled Cholesky factorization on NUMA machine. We have designed a new symbolic technnique for allocating task and tiles at the same time called SMA (Symbolic Mapping and Allocation). SMA provide an optimal allocation in terms of point-to-point communication for the Cholesky factorization. We have studied some performance issues regarding the way threads are grouped and tiles are allocated in the memory. We have shown how to optimize thread placement and data placement in order to achieve performance gain up to 50% compared to state-of-the-art libraries such as Plasma or MKL. This work has been published in PAAP 2012 [25].

6.5. Scheduling for System On Chip

Participants: Paul-Antoine Arras, Emmanuel Jeannot, Samuel Thibault.

Today's embedded applications are increasingly demanding in terms of computational power, especially in real-time digital signal processing (DSP) where tight timing requirements are to be fulfilled. More specifically, when it comes to video decoding (e.g. H.264/AVC and HEVC) not only has it been almost impossible for some time to run such codecs on a stand-alone embedded processor, but it now also becomes quite impractical to execute them on homogeneous multicore platforms. In this context, STMicroelectronics is developing a scalable heterogeneous system-on-chip template called STHORM and aimed at meeting the latest codecs' requirements.

This year, we focused on the memory constraints embedded systems are subject to. As video coding is rather demanding in terms of storage capacity, we have proposed a method aimed at introducing the notion of memory into a class of widespread scheduling heuristics that exhibit both good performance and low complexity. Thanks to this technique, we achieved speedups over 20%.

The next step is to formalize an execution model on top of which a runtime software will be built. This implies specifying both the application requirements and modeling precisely the target platform, namely STHORM.

6.6. High-Performance Point-to-Point Communications

Participants: Alexandre Denis, Sébastien Barascou, Raymond Namyst.

- NEWMADELEINE is our communication library designed for high performance networks in clusters. We have worked on optimizations on low-level protocols so as to improve point-to-point performance.
- We have proposed a communication protocol [21] for InfiniBand that amortizes the cost of checksums as used by fault-tolerant MPI implementations. We have modeled the behavior of the network and proposed auto-tuning mechanisms to adapt the protocol to the hardware properties.
- This work was initiated in the context of the FP3C collaboration with the University of Tokyo.

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-CESTA CEA-CESTA is granting the CIFRE PhD thesis of Cyril Bordage on *Parallelization of fast multipole methods over hybrid CPU+GPU architectures* from october 2009 to november 2012.

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.

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 ProHMPT Programming Heterogeneous Multiprocessing Technologies (http://runtime.bordeaux.inria.fr/prohmpt/).

ANR COSINUS 2008 Program, 01/2009 - 06/2012 (42 months)

Identification: ANR-08-COSI-013

Coordinator: Olivier Aumage (Inria Bordeaux)

Other partners: CEA INAC, CEA CESTA, CAPS entreprise, Bull, UVSQ PRiSM, Inria Grenoble.

Abstract: ProHMPT aims at focusing the joint research work of several teams about compilers, runtimes and libraries as well as scientific application programmers on designing methods and tools for programming heterogeneous platforms such as GPU and accelerators.

Nomination: The project ProHMPT has been nominated for the first round of selection for the best ANR projects recently completed.

ANR MediaGPU Massive multimedia GPU-Based Processing (http://picoforge.int-evry.fr/projects/mediagpu/).

ANR CORD 2009 Program, 01/2010 - 12/2012 (36 months)

Identification: 2009-CORD-25-01

Coordinator: Pierre Pleven (Institut TELECOM)

Other partners: PLAY ALL, ATEME, HPC-Project, Inria Bordeaux

Abstract: The MediaGPU project will develop a software architecture and will review and adapt a number of classical multimedia algorithms, considering the latest advances offered by the new hardware architectures, such as Hybrid CPU+GPU and GPGPU. Initial key target applications are very large still images processing, high definition video encoding, video post-production, real-time geometry 3D synthesis.

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.

8.3. European Initiatives

8.3.1. FP7 Projects

PEPPHER FP7 Strep "Performance Portability and Programmability for Heterogeneous Many-core Architectures"

Specific Targeted Research Project (STREP), October 2010 - December 2012

Coordinator: Universität Wien (Austria)

Others partners: Chalmers Tekniska Högskola AB (Sweden), Codeplay Software Limited (United Kingdom), Intel GmbH (Germany), Linköpings Universitet (Sweden), Movidia Ltd. (Ireland), Universität Karlsruhe (Germany)

Abstract: PEPPHER aims at providing a unified framework for programming architecturally diverse, heterogeneous many-core processors to ensure performance portability. PEPPHER will advance state-of-the-art in its five technical work areas:

1. Methods and tools for component based software
2. Portable compilation techniques
3. Data structures and adaptive, autotuned algorithms
4. Efficient, flexible run-time systems
5. Hardware support for autotuning, synchronization and scheduling

8.3.2. Collaborations in European Programs, except FP7

COST ComplexHPC complexhpc.org

Program: COST

Project acronym: ComplexHPC

Project title: ComplexHPC

Duration: may 2009 – june 2013

Coordinator: Emmanuel Jeannot

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. This Action gathers more than 26 countries and 50 partners in Europe. The budget for the whole action and the four years is 380 000 euros.

8.4. International Initiatives

8.4.1. Inria Associate Teams

MORSE Matrices Over Runtime Systems at Exascale

Inria Associate-Teams program: 2011-2013

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. Participation In International Programs

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

ANR-JST 2010 Program, 03/2010 - 02/2013 (36 months)

Identification: ANR-10-JST-002

Coordinator: Serge Petiton (Inria Saclay)

Other partners: CNRS IRIT, CEA DEN Saclay, Inria Bordeaux, CNRSPrism, 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 - 12/2013 (12 months, renewable)

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.

8.5. International Research Visitors

8.5.1. Visits of International Scientists

8.5.1.1. Internships

Satoshi OHSHIMA visited us in September and October 2012, and accelerated the FEM application of the University of Tokyo execution by using STARPU.

9. Dissemination

9.1. Scientific Animation

Raymond NAMYST is vice-chair of the Research and Training Department in Mathematics and Computer Science (UFR Math-Info) of the University of Bordeaux 1. He is also a member of the Scientific Committee of the University of Bordeaux 1

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 chairs the scientific committee of the ANR "Numerical Models" program for the 2011-2013 period.

Raymond NAMYST was the coordinator of the chapter about runtime systems within the "Software Eco-System" Workgroup of EESI (*European Exascale Software Initiative*).

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

- EESI (*European Exascale Software Initiative*, since 2010) ;
- CEA/DAM (as a "scientific expert" for the 2008-2012 period) ;
- 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'12, MSEPT 2012, ROSS 2012, ISPA 2012, CASS 2012.

Raymond NAMYST gave invited talks at the following international workshops: CCDSC'12 (Dareizé), COMPEEF'12 (Grenoble), Torrents'12 (Toulouse), FGPS'175 (Mons).

Samuel THIBAULT was a program committee member of the following conferences: IPDPS 2013, Renpar 2013

Brice GOGLIN was a program committee member of SC'12 technical program and posters, Hot Interconnect 2012 and EuroMPI 2012. He was also a member of the SC'12 ACM Student Research Competition jury.

Guillaume MERCIER was program committee member of EuroMPI 2012, ICPADS 2012 and CCGrid 2012. He was also a reviewer for the IEEE TPDS and FGCS journals.

Olivier AUMAGE was reviewer for the IEEE TPDS journal and for the SC12, IPDPS 2012, ROSS, DATE 2013 and RenPar/Compas conference and workshops. He is part of the Inria Bordeaux – Sud-Ouest committee for scientific event fundings.

Emmanuel JEANNOT was program committee member for: ISPA 2012, HPDC 2012, PDGC 2012, HiPC 2012, Heteropar 2012, renpar 21, Realis-01, CCGrid2013 and IPDPS 2013.

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, CEJCS.

Emmanuel JEANNOT has given an invited talk and CCDSC 2012 (Darize, France), the 7th scheduling workshop (Pittsburgh USA), CCMSE 2012 (La Manga, Spain), and the Inria-Joint-Lab meeting at Argonne National Lab.

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 & HdR :

PhD: Jérôme CLET-ORTEGA, Exploitation efficace des architectures parallèles de type grappes de NUMA à l'aide de modèles hybrides de programmation, 2012/04, Raymond NAMYST and Guillaume MERCIER

PhD: Andres CHARIF-RUBIAL, On code performance analysis and optimization for multicore architectures, 2012/02, Denis BARTHOU and William JALBY (Université de Versailles Saint Quentin en Yvelines)

PhD in progress : Julien JAEGER, Source-to-source transformations for irregular and multithreaded code optimization, 2012/02, Denis BARTHOU

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 : Sylvain HENRY, Modèles de programmation et systèmes d'exécution pour architectures hétérogènes, 2009/10, Denis BARTHOU and Alexandre DENIS

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: Cyril BORDAGE, Parallélisation de la méthode multipôle sur architecture hybride, 2012/10, Raymond NAMYST and David GOUDIN (CEA CESTA)

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

9.2.3. Juries

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

- Marcio CASTRO (University of Grenoble, reviewer)
- Vincent PICHON (University of Lyon, reviewer)
- Marc PALYART (CEA-CESTA, Bordeaux, reviewer)
- Andres CHARIF RUBIAL (University of Versailles, president)

Samuel THIBAULT was member of the PhD defense jury for the following candidates:

- Vincent BOULOS (University of Grenoble, examinator)

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

- Mohamed Slim BOUGUERRA (University of Grenoble, reviewer)
- Cristian KLEIN (ENS Lyon, reviewer)
- Jan-Christian MEYER (NTNU, Trondhiem, Norway, opponent)

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 generac public audience and school student, as well as several radio and paper interviews about Inria's activities.

Brice GOGLIN, François TESSIER and Bertrand PUTIGNY presented the team's research work to one hundred high-school students at the "Fête de la Science".

Brice GOGLIN and Bertrand PUTIGNY presented research careers at the Aquitec student exhibition.

Samuel THIBAULT was an invited speaker for a public round table about Author rights and HADOPI.

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, p. 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, p. 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, p. 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, p. 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

Doctoral Dissertations and Habilitation Theses

[9] A. CHARIF-RUBIAL. *On code performance analysis and optimization for multicore architectures*, Université de Versailles Saint-Quentin, 2012.

[10] J. CLET-ORTEGA. *Exploitation efficace des architectures parallèles de type grappes de NUMA à l'aide de modèles hybrides de programmation*, Université Sciences et Technologies - Bordeaux I, April 2012, http://hal.inria.fr/tel-00773007.

[11] J. JAEGER. *Source-to-source transformations for irregular and multithreaded code optimization*, Université de Versailles Saint-Quentin, 2012.

Articles in International Peer-Reviewed Journals

[12] A. BENOIT, L.-C. CANON, E. JEANNOT, Y. ROBERT. *Reliability of task graph schedules with transient and fail-stop failures: complexity and algorithms*, in "Journal of Scheduling", 2012, vol. 15, n^{o} 5, p. 615-627 [*DOI :* 10.1007/s10951-011-0236-y], http://hal.inria.fr/hal-00763343.

[13] 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, p. 176-188 [*DOI :* 10.1016/j.jpdc.2012.09.016], http://hal.inria.fr/hal-00731714.

Articles in National Peer-Reviewed Journals

[14] S. MAHMOUDI, P. MANNEBACK, C. AUGONNET, S. THIBAULT. *Traitements d'Images sur Architectures Parallèles et Hétérogènes*, in "Technique et Science Informatiques", 2012, http://hal.inria.fr/hal-00714858.

[15] H. SYLVAIN, A. DENIS, D. BARTHOU. *Programmation unifiée multi-accélérateur OpenCL*, in "Techniques et Sciences Informatiques", 2012, vol. 31, n^{o} 8-9-10, p. 1233-1249 [*DOI :* 10.3166/TSI.31.1233-1249], http://hal.inria.fr/hal-00772742.

Invited Conferences

[16] C. BORDAGE. *Parallelization on Heterogeneous Multicore and Multi-GPU Systems of the Fast Multipole Method for the Helmholtz Equation Using a Runtime System*, in "ADVCIMP12", Barcelone, Spain, IARIA, September 2012, p. 90-95, http://hal.inria.fr/hal-00773114.

International Conferences with Proceedings

[17] C. AUGONNET, O. AUMAGE, N. FURMENTO, R. NAMYST, S. THIBAULT. *StarPU-MPI: Task Programming over Clusters of Machines Enhanced with Accelerators*, in "The 19th European MPI Users' Group Meeting (EuroMPI 2012)", Vienna, Austria, J. L. TRÄFF, S. BENKNER, J. DONGARRA (editors), LNCS, Springer, 2012, vol. 7490, http://hal.inria.fr/hal-00725477.

[18] D. BARTHOU, G. GROSDIDIER, M. KRUSE, O. PENE, C. TADONKI. *QIRAL: A High Level Language for Lattice QCD Code Generation*, in "Programming Language Approaches to Concurrency and Communication-centric Software Workshop", 2012, To appear.

[19] D. BARTHOU, G. GROSDIDIER, M. KRUSE, O. PÈNE, C. TADONKI. *QIRAL: A High Level Language for Lattice QCD Code Generation*, in "European Joint Conferences on Theory and Practice of Software (ETAPS)", Tallin, Estonia, Electronic Proceedings in Theoretical Computer Science, 2012, p. 37-43 [*DOI :* 10.4204/EPTCS], http://hal.inria.fr/hal-00666885.

[20] S. BENKNER, E. BAJROVIC, E. MARTH, M. SANDRIESER, R. NAMYST, S. THIBAULT. *High-Level Support for Pipeline Parallelism on Many-Core Architectures*, in "Europar - International European Conference on Parallel and Distributed Computing - 2012", Rhodes Island, Greece, August 2012, http://hal.inria.fr/hal-00697020.

[21] A. DENIS, F. TRAHAY, Y. ISHIKAWA. *High performance checksum computation for fault-tolerant MPI over InfiniBand*, in "the 19th European MPI Users' Group Meeting (EuroMPI 2012)", Vienna, Austria, J. L. TRÄFF, S. BENKNER, J. DONGARRA (editors), LNCS, Springer, September 2012, vol. 7490, http://hal.inria.fr/hal-00716478.

[22] A. DUCHATEAU, D. PADUA, D. BARTHOU. *Hydra: Automatic Algorithm Exploration from Linear Algebra Equations*, in "ACM/IEEE Intl. Symp. on Code Optimization and Generation", Shenzhen, China, IEEE Computer Society, February 2013, To appear.

[23] 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.

[24] J. JAEGER, D. BARTHOU. *Automatic efficient data layout for multithreaded stencil codes on CPUs and GPUs*, in "IEEE Intl. High Performance Computing Conference", Pune, India, December 2012, To appear.

[25] E. JEANNOT. *Performance Analysis and Optimization of the Tiled Cholesky Factorization on NUMA Machines*, in "PAAP 2012 - IEEE International Symposium on Parallel Architectures, Algorithms and Programming", Taipei, Taiwan, Province Of China, IEEE, December 2012, http://hal.inria.fr/hal-00772790.

[26] C. KESSLER, U. DASTGEER, S. THIBAULT, R. NAMYST, A. RICHARDS, U. DOLINSKY, S. BENKNER, J. L. TRÄFF, S. PLLANA. *Programmability and Performance Portability Aspects of Heterogeneous Multi-/Manycore Systems*, in "DATE-2012 conference on Design, Automation and Test in Europe", Dresden, Germany, IEEE CS Press, March 2012, p. 1403–1408.

National Conferences with Proceeding

[27] 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.

[28] 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.

[29] C. ROSSIGNON. *Optimisation du produit matrice-vecteur creux sur architecture GPU pour un simulateur de réservoir*, in "ComPAS'13 / RenPar'21 - 21es Rencontres francophones du Parallélisme", Grenoble, France, INRIA GRENOBLE (editor), 2013, http://hal.inria.fr/hal-00773571.

Scientific Books (or Scientific Book chapters)

[30] P. DE OLIVEIRA CASTRO, S. LOUISE, D. BARTHOU. *DSL Stream Programming on Multicore Architectures*, in "Programming Multi-core and Many-core Computing Systems", Parallel and Distributed Computing, Wiley-Blackwell, February 2013, To appear.

Research Reports

[31] D. BALOUEK, A. CARPEN AMARIE, G. CHARRIER, F. DESPREZ, E. JEANNOT, E. JEANVOINE, A. LÈBRE, D. MARGERY, N. NICLAUSSE, L. NUSSBAUM, O. RICHARD, C. PÉREZ, F. QUESNEL, C. ROHR, L. SARZYNIEC. *Adding Virtualization Capabilities to Grid'5000*, Inria, July 2012, n^{o} RR-8026, 18, http://hal.inria.fr/hal-00720910.

[32] F. DESPREZ, G. FOX, E. JEANNOT, K. KEAHEY, M. KOZUCH, D. MARGERY, P. NEYRON, L. NUSSBAUM, C. PÉREZ, O. RICHARD, W. SMITH, G. VON LASZEWSKI, J. VÖCKLER. *Supporting Experimental Computer Science*, March 2012, n^{o} Argonne National Laboratory Technical Memo 326, http://hal.inria.fr/hal-00720815.

[33] F. DESPREZ, G. FOX, E. JEANNOT, K. KEAHEY, M. KOZUCH, D. MARGERY, P. NEYRON, L. NUSSBAUM, C. PÉREZ, O. RICHARD, W. SMITH, G. VON LASZEWSKI, J. VÖCKLER. *Supporting Experimental Computer Science*, Inria, July 2012, n^{o} RR-8035, 29, http://hal.inria.fr/hal-00722605.

References in notes

[34] 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.

[35] 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.

[36] 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.

[37] A. MAZOUZ, S.-A.-A. TOUATI, D. BARTHOU. *Study of Variations of Native Program Execution Times on Multi-Core Architectures*, in "Intl. IEEE Workshop on Multi-Core Computing Systems", Krakow, Poland, IEEE Computer Society, February 2010, 919—924.