INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team runtime

Efficient runtime systems for parallel architectures

Bordeaux - Sud-Ouest

Theme : Distributed and High Performance Computing

Table of contents

1. Team

Research Scientists

Olivier Aumage [Junior Researcher, INRIA]
Alexandre Denis [Junior Researcher, INRIA]
Brice Goglin [Junior Researcher, INRIA]
Emmanuel Jeannot [Junior 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]

Technical Staff

Nicolas Collin [Associate Engineer, INRIA, European Project grant]
Nathalie Furmento [Research Engineer, CNRS]
Yannick Martin [Associate Engineer, INRIA]
Ludovic Stordeur [Associate Engineer, INRIA]
François Tessier [Associate Engineer, INRIA, ANR grant]

PhD Students

Cédric Augonnet [University of Bordeaux, École Normale Supérieure de Lyon grant]
François Broquedis [University of Bordeaux, MESR grant, until Aug. 2010]
Louis-Claude Canon [University of Bordeaux, MESR grant, until Aug. 2010]
Andres Charif-Rubial [University of Versailles, ANR grant]
Jérôme Clet-Ortega [University of Bordeaux, MESR grant]
François Diakhaté [CEA, University of Bordeaux]
Sylvain Henry [University of Bordeaux, MESR grant]
Julien Jaeger [University of Versailles, ANR grant]
Stéphanie Moreaud [University of Bordeaux]
Bertrand Putigny [University of Bordeaux, INRIA grant]

Post-Doctoral Fellows

Remi Sharrock [IPB, since Aug. 2010]
François Trahay [ATER, University of Bordeaux, until May 2010]

Administrative Assistant

Sylvie Embolla

2. Overall Objectives

2.1. Designing Efficient Runtime Systems

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 underlying operating system. Runtime systems can thus be seen as functional extensions of operating systems, but the boundary between them is rather fuzzy since runtime systems may actually contain specific extensions/enhancements to the underlying operating system (e.g. extensions to the OS thread scheduler). The increasing complexity of modern parallel hardware, making it more and more necessary to postpone essential decisions and actions (scheduling, optimizations) at run time, emphasizes the role of runtime systems.

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 deal with the increasing complexity of upcoming hardware architectures. While it is essential to understand – and somehow anticipate – the evolutions of hardware technologies (e.g. programmable network interface cards, multicore architectures, hardware accelerators), the most delicate task is to extract models and abstractions that will fit most of upcoming hardware features.

The originality of the runtime group lies in the fact that we address all these issues following a global approach, so as to propose complementary solutions to problems which may not seem to be linked at first sight. We actually realized, for instance, that we could greatly improve our communication optimization techniques by increasing the functionalities of the underlying core thread scheduler. This illustrates why most of our research efforts have consisted in cross-studying different topics, and have led to co-designing many software.

Our research project centers on three main directions:

Mastering large, hierarchical multiprocessor machines
- Thread scheduling over multicore machines
- Data management over NUMA architectures
- Task scheduling over GPU heterogeneous machines
- Exploring parallelism orchestration at compiler and runtime level
- Improved interactions between optimizing compiler and runtime

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

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.

3. Scientific Foundations

3.1. Runtime Systems Evolution

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.* [50], 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 [13] 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 [20], 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. For parallel stream languages, several languages has been proposed, dedicated to embedded applications (ArrayOL, Brook, BlockParallel, StreamIT [51] for instance). These languages either express explicitely the flow of data (such as StreamIT or Brook): code optimizations can more easily restructure this flow, but the expression is low-level and difficult to program. Other languages (such as ArrayOL or BlockParallel) rely on higher abstractions and flow is mainly expressed through dependences. The consequence is that it is difficult to extract in some complex flow restructuration the flow from the dependences, hence constraining de facto the expressivity of such languages or their optimizations. We have proposed an extension to StreamIT language [51] in [38] for a higher representation of structured streams, the language SLICES. One of the main issue here is to propose a high-level abstraction language, easy to use for the developper, preserving the semantics all along the optimization chain that leads to high performance code. The SLICES languages expresses flow reorganization in a manner that can be directly translated into a lower level representation. In [38], we introduced an internal representation of the stream programs, based on Cyclo-Static Dataflow Graphs, extending StreamIT,

tailored for the exploration of different communication and parallel expression of a program. More specifically, the language proposed, SJD, builds a graph representation of the dataflow, with specific nodes for stream reorganization or communication. The language is more expressive than StreamIT and can be obtained by transformation from a SLICES program. We have developped a set of optimization on such graph that explores different parallel versions of the code and flow reorganization to optimize some architectural constraint. In [39] we focused on the minimization of memory consumption of each processing node, and in [47] we shown that the same approach could generate multi-grain parallelism codes for heterogeneous architectures. Finally, the flow is not explicit as in usual imperative programs written in C/C++, we developped a tool called FADAlib[28] to automatically compute dataflow information and hence discover parallelism. This approach, relying on the polyedric model, works for any program and the method proposed here in this tool uses any information provided by the user through pragmas (such as invariants) to improve the quality of its analysis. The tool has been successfully integrated in a version of gcc/Graphite.

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 [34] 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. Panorama

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. 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 and NUMA memory nodes.
- 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 [30], from thread scheduling to placement of MPI processes (all major MPI implementations are being ported on top of it).
- 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 24 000 lines of C.
- http://runtime.bordeaux.inria.fr/hwloc/

5.2. KNem

Participants: Brice Goglin, Stéphanie Moreaud.

- 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 between processes on the same node.
- Thanks to its kernel-based design, it 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 and they offer the ability dynamically deciding when and how to use it depending on the hardware characteristics [35].
- KNEM is composed of 7000 lines of C. Its main contributor is Brice GOGLIN.
- http://runtime.bordeaux.inria.fr/knem/

5.3. 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 83 000 lines of code.
- http://runtime.bordeaux.inria.fr/marcel/

5.4. ForestGOMP

Participants: Olivier Aumage, François Broquedis, 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/

5.5. Open-MX

Participants: Brice Goglin, Ludovic Stordeur.

- 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 [49] or the recent iWarp [48] 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 offers efficient data movement abilities thanks to copy offload abilities of modern hardware [24].
- 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 [23], [8].
- Brice GOGLIN and Ludovic STORDEUR are the main contributors 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: Cédric Augonnet, Nathalie Furmento, Samuel Thibault.

- STARPU typically makes it much easier for high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Cell processors: rather than handling low-level issues, programmers may concentrate on algorithmic concerns.
- Portability is obtained by the means of a unified abstraction of the machine. STARPU offers a unified offloadable task abstraction named codelet. Rather than rewriting the entire code, programmers can encapsulate existing functions within codelets. 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. In order to relieve programmers from the burden of explicit data transfers, 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.
- Given its expressive interface and portable scheduling policies, 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.
- http://runtime.bordeaux.inria.fr/StarPU/

5.7. 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/

5.8. 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, 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/

6. New Results

6.1. High-performance message passing over generic Ethernet hardware

Participants: Ludovic Stordeur, Brice Goglin.

- The OPEN-MX message passing stack (described in Section 5.5) offers a native message passing layer on any ETHERNET hardware. The API compatibility with the native Myrinet Express stack already enables existing parallel application to use OPEN-MX. Indeed, several legacy high-performance layers such MPICH2 or Open MPI run works transparently on top of OPEN-MX.
- OPEN-MX demonstrates significant performance improvement over traditional TCP implementations [24].

- OPEN-MX is also an interesting framework for studying next-generation hardware features that could help ETHERNET hardware becoming legacy in the context of high-performance computing. We exhibited some cache-inefficiency problems in the OPEN-MX receive stack that are inherited from the ETHERNET model. By adding OPEN-MX-aware packet filtering capabilities in the *Multiqueue* firmware of Myri-10G boards, we are able to control the location of the processing of the incoming OPEN-MX traffic. We extended this model by providing an automatic binding facility for user-space applications. This model enables the whole processing of each incoming OPEN-MX packet on the core that runs its target application, causing the overall cache efficiency to improve dramatically [23].

6.2. High-Performance Intra-node Communications

Participants: Brice Goglin, Stéphanie Moreaud.

- We showed in [9] that the major MPI implementations had severe performance problems for large-message intra-node communication. We thus extracted the optimized intra-node communication model out of OPEN-MX and created the KNEM driver so as to offer the same abilities to any existing MPI stack (see Section 5.2).
- We showed that KNEM indeed improves intra-node communication performance significantly. We described how each communication strategy performance depends on process placement [35] and proposed an optimized implementation of collective operation that take the machine topology into account [45].
- This work was initiated in the context of our collaboration with the MPICH2 team and is now also pursued with the OPEN MPI project.

6.3. I/O-Affinity-aware MPI Communications

Participants: Brice Goglin, Stéphanie Moreaud.

- We demonstrated in the past that the locality of I/O devices within modern computing nodes has the significant impact of the MPI communication performance [10] (*Non-Uniform I/O Access*, NUIOA).
- A first way to deal with such affinities would be to privilege I/O-intensive processes by placing them near the network interfaces. However, determining the communication-intensiveness may be tricky. Also, some applications have uniform communication patterns. The other way to deal with I/O affinities is to modify the implementation of communication operations given a predetermined task placement.
- We proposed a multirail strategy that takes these affinities into account when splitting MPI messages across multiple network interfaces. This strategy improves point-to-point communication by up to 15% and collective operations by up to 5% by privileging local interfaces [36].
- We also demonstrated that the implementation of collective operations should take I/O affinities into account. Deciding which steps and leaders should be involved in the algorithms based on NUIOA effects led us to improve broadcast performance by up to 10% [32].

6.4. Topology-aware High-Performance Computing

Participants: Brice Goglin, Emmanuel Jeannot, Guillaume Mercier, Samuel Thibault.

- The democratization of multicore processors and NUMA machines spreads complex and hierarchical architectures to the whole world of high-performance computing and even more. So far, the need to master the internal hardware topology was critical only to large shared-memory machines but now comes to smaller nodes and clusters as well.

- We showed that a proper MPI processes binding policy within NUMA nodes induces significant impact for parallel application performance We proposed an automatic placement scheme that gathers information about the application communication patterns during a preliminary run so as to place processes according to their communication affinities and to the hardware characteristics such as shared caches or NUMA nodes. We developed a specific algorithm (called TREEMATCH) for matching the processes to the resources in order to reduce the communication cost of the application [33]. However, in order to be able to place the MPI processes onto the various computing cores, we need to acquire the most encompassing vision of the architecture. We have also developed an accelerated version of the algorithm for enabling on-the-fly decision. This version have been ported into the Charm++ framework as a dynamic load-balancer. TREEMATCH has also been integrated into the MPICH2 implementation. Indeed, the new topology interface enacted by the MPI Forum in September 2009 and implemented by MPICH2 is currently not taking into account the underlying hardware topology. To address this issue and in order to improve application performance, we expanded the current implementation by using both HWLOC and TREEMATCH. This work is preliminary and carried out with researchers from the University of Vienna that did propose the new topology MPI 2.2 interface.
- The HWLOC software (see Section 5.1) answers this problem by offering a detailed knowledge of the hardware in a portable and abstracted manner. We showed that HWLOC can help popular high-performance OPENMP or MPI software [20]. Indeed, scheduling OPENMP threads according to their affinities or placing MPI processes according to their communication patterns shows interesting performance improvement thanks to HWLOC. An optimized MPI communication strategy may also be dynamically chosen according to the location of the communicating processes in the machine and its hardware characteristics.

6.5. Static dimensioning of hybrid platform

Participants: Denis Barthou, Julien Jaeger, Emmanuel Jeannot.

Given a task graph modeling an application each of the task beings potentially executed on different type of resources (CPU, accelerator, GPU, etc.), we have targeted the problem of statically schedule such task graph onto the resources. We have proposed an algorithm (called SPAGHETTI) to determine the best possible allocation when an unlimited number of resources of each type is available. This algorithm then serves as a basis for determining the number of resources required to execute the application in a minimum amount of time. We are then able to modify the given mapping to explore different trade-offs between number of resources and the application execution time.

6.6. Fault tolerance in high-performance communication library

Participants: Alexandre Denis, François Trahay.

With the increase of the number of nodes in clusters, the probability of failures increases. We have studied the failures in the network stack for high performance networks. We have proposed [37] the design of several fault-tolerance mechanisms for communication libraries to detect failures and to ensure message integrity. We have implemented these mechanisms in the NEWMADELEINE communication library with a quick detection of failures in a portable way, and with fallback to available links when an error occurs. Our mechanisms ensure the integrity of messages without lowering too much the networking performance.

6.7. Programming Heterogeneous Platforms

Participants: Cédric Augonnet, Nathalie Furmento, Raymond Namyst, Samuel Thibault, Olivier Aumage.

The scientific community interests in exploiting accelerators like GPGPUs for scientific computations has been confirmed more and more during 2010. The top-ranked top500 machines are indeed gaining most of their computation power from NVIDIA or AMD GPUs. We have continued our efforts on our StarPU runtime system and collaborated with several other projects on the national, european, and international levels.

- The data management part of StarPU has been improved by the means of chained requests [27]. This permits to seamlessly support GPU-GPU transfers which are not yet directly supported by *e.g.* CUDA, but also support GPU-NIC transfers. More precisely, an MPI-like layer has been designed to nicely integrate StarPU tasks with MPI communications, completely keeping away the programmer from tedious dependency analysis and optimization between task scheduling and MPI requests.
- Task scheduling has been improved by taking data transfers into account [27]. Thanks to an auto-tuned performance model for data transfers, StarPU is now able to automatically make a compromise between load balancing and data transfer cost, by taking both task execution time and data transfer time into account during scheduling decisions. Results show that the resulting automatic data distribution yields almost as good performance as a hand-tune data distribution, without intervention from the programmer, and more importantly, independently from the target machine.
- A close collaboration with the University of Tennessee (UTK) permitted to integrate StarPU with both Magma and Plasma, which provide state-of-the art implementations of several linear algebra algorithms, notably the Cholesky, QR and LU factorizations. It took one week to develop a small layer which permits to use StarPU to integrate the best of both Magma and Plasma, i.e. the Magma kernels into the Plasma algorithm, for the Cholesky factorization [40], [26]. The same integration for the QR factorization then took just a few days, and permitted to extend the algorithm into a Communication-Avoidance variant [25]. Work is now being done on the LU factorization.
- Several collaborations have been started in the context of the ANR PRoHMPT project dedicated to programming heterogeneous platforms In particular, we have an ongoing collaboration with the CEA CESTA team on porting the Kiss3D application on GPU through the use of StarPU.
- A collaboration with the PIPS compiler has been started in the context of the ANR MEDIAGPU project: PIPS will have StarPU as one of its backend, so as to seamlessly support GPUs for applications compiled with PIPS.
- Several collaborations have been started in the context of the ANR/JST FP3C project. A collaboration with the team of Prof. Boku from the University of Tsukuba aims at the integration of XcalableMP and StarPU to seamlessly support grids and clusters of GPU-empowered machines. A collaboration with the team of Prof. Matsuoka from the University of Tokyo will make use StarPU as a runtime environment for their architecture-neutral domain-specific programming models.
- Several collaboration have been started in the context of the PEPPHER project. The Movidius industrial partner extended their pepphersim multicore simulator to provide an OpenCL layer which permits to integrate it with the OpenCL layer of StarPU. The University of Vienna started to work on work-stealing scheduling strategies within StarPU.

6.8. Efficient scheduling of OpenMP threads on NUMA machines

Participants: Olivier Aumage, François Broquedis, Nathalie Furmento, Brice Goglin, Raymond Namyst, Samuel Thibault, Pierre-André Wacrenier.

- ForestGOMP (5.4) is now able to take thread/memory affinities into account while distributing the load on hierarchical architectures. It relies on the MAMI memory manager to allocate, bind or migrate memory buffers. Moreover ForestGOMP adopts a two-ways mechanism [20], [29] to decide how often the distribution needs to be updated. First, every time the application programmer updates the memory affinities, the bubble scheduler is called to check the current distribution. This approach may not be sufficient for irregular applications, so

- FORESTGOMP also provides a more dynamic mechanism based on hardware counters inspecting. The runtime checks the counters on a regular basis and infers the amount of remote memory accesses initiated from the current processor while defining a threshold from which FORESTGOMP will call the scheduler for checking the current distribution. These two approaches are complementary. Indeed, in some cases updates from the application programmer will not need the scheduler to rethink the current distribution. In other cases the programmer is able to roughly define which part of his application will work on which data, but cannot tell precisely when and how. Hardware counters can help reacting at the right time for these situations.
- In the context of the ANR PRoHMPT project, we started a study on the potential benefits from using two-level OpenMP parallelization inside the BigDFT application from CEA INAC team using our FORESTGOMP OpenMP runtime system.

7. Other Grants and Activities

7.1. Regional Initiatives

+ The RUNTIME team is member of the joint regional project with the CEPAGE INRIA team called : *Modélisation des performances pour plate-formes hétérogènes*. The goal of this project is to provide tractable models for modern parallel and large-scale platform and use these models to design algorithmic solutions enabling efficient use of such platforms (scheduling, communications, etc.).

7.2. National Initiatives

+ We participate to a research proposal to the ANR *Cosinus* program called "COOP" which was granted a three-year funding (dec. 2009 – dec. 2012). It 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. It involves academic partners and EDF R&D. (http://coop.gforge.inria.fr/)
+ We lead a research proposal to the ANR *Cosinus* program called "ProHMPT" which was granted a three-year funding (jan. 2009 – dec. 2011). It aims at focusing the joint research work of several teams about compilers, runtimes and libraries on programming heterogeneous platforms such as GPU and accelerators. It involves academic partners, companies (Bull, CAPS entreprise) and CEA teams. (http://runtime.bordeaux.inria.fr/prohmpt/)
+ We participate to a research proposal to the ANR *CONTINT* program called "MEDIAGPU" which was granted a three-year funding (jan. 2010 - dec. 2012). It 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 combinations of CPUs and GPUs (http://picoforge.int-evry.fr/projects/mediagpu/).
+ We participate to a research proposal to the ANR *Cosinus* program called "PetaQCD" which was granted a three-year funding (jan. 2009 - dec. 2011). It develops software architecture and methods for designing future machines for large sustained petaflop simulation of Lattice Quantic Chromo-Dynamics (https://www.petaqcd.org).

7.3. European Initiatives

+ *COST Action IC0805 ComplexHPC (Open European Network for High-Performance Computing in Complex Environments*

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. This Action gathers more than 20 countries and 30 partners in Europe. This Action runs for 4 years, may 2009– may 2013.

Emmanuel JEANNOT is the chair of this action. And we actively participate in the different working groups such as "*Efficient use of complex systems with an emphasis of computational library and communication library*"; "*Algorithms and tools for mapping and executing applications onto distributed and heterogeneous systems*" or "*Applications of hierarchical-heterogeneous systems.*"

Through this action we have setup collaborations and visits on STARPU with the University of Lisbon, Portugal and the University of Mons, Belgium.

7.4. International Initiatives

+ We established a collaboration with the OPEN MPI project in the context of development of the HWLOC software (see Section 5.1). This collaboration was also informally extended to the development of high-performance intra-node communication with OPEN MPI over our KNEM driver (see Section 5.2).
+ An associate team between our group and the MPICH2 development team (Argonne National Laboratory) was setup at the end of 2007 and finished this year. It has been favourably evaluated by all the reviewers. Our KNEM and HWLOC software (see Sections 5.2 and 5.1) were also integrated into MPICH2 and our TREEMATCH algorithm should follow.
+ The Runtime project is part of the joint laboratory that was setup between INRIA and University of Illinois Urbana-Champaign (UIUC) about Petascale Computing (http://jointlab.ncsa.illinois.edu/).
+ A PHC Sakura collaboration between the Runtime project and the group of Prof. Yutaka ISHIKAWA was accepted and has started at the beginning of 2009 and finishes in December 2010. Results have been submitted [37] for publication.
+ We participate to the joint ANR-JST project FP3C (*Framework and Programming for Post Petascale Computing*). The goal of this project is to contribute to establish software technologies, languages and programming models to explore extreme performance computing beyond petascale computing, on the road to exascale computing.

8. Dissemination

8.1. Scientific animation and expertise

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 and RUNTIME INRIA groups.

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

- EESI (*European Exascale Software Initiative*, since 2010) ;
- CEA/DAM (as a "scientific advisor" for the 2008-2010 period) ;
- CEA-EDF-INRIA School technical committee (since 2009) ;
- DGRI (French Ministry of Research since 2009) ;
- GENCI (http://www.genci.fr/?lang=en, since 2009) ;
- ORAP (http://www.irisa.fr/ORAP/, as the INRIA representative since 2010) ;
- ANR (Member of the "COSINUS" scientific committee since 2008).

Raymond NAMYST was member of the following PhD committees: Nicolas RICHART, Everton HERMANN (reviewer) and Fabrice DUPROS.

Emmanuel JEANNOT is member of the steering committee and the direction committee of the ADT Aladdin-G5K and serves as head of the Bordeaux site since October 2009.

Emmanuel JEANNOT was reviewer of the PhD dissertations of Alessio Merlo (University of Genoa, Italy) and Tram Truong Huu (Univ. of Nice). He was reviewer of the habilitation thesis of Thierry Monteil (Univ. of Toulouse). He was also member of the PhD committee of Remi Sharrock (Univ. Toulouse) and Benjamin Depardon (ENS-Lyon).

Emmanuel JEANNOT serves as reviewers of following journals/conferences: IEEE Trans. on Parallel and Dist. Syst., Future Generation Computer Systems, Parallel computing, Journal of Scheduling, Journal of Parallel and Distributed Systems and the heteropar 2010 conference.

Olivier AUMAGE is the head of the ANR ProHMPT project (Jan. 2009 – Dec. 2011).

Denis BARTHOU was member of the following PhD committees: Pablo DE OLIVEIRA, Stephane ZUCKERMANN, François BROQUEDIS.

Denis BARTHOU serves as an expert in the following institutions/initiatives:

- Exascale Research Lab.
- Cofecub: Comité Français d'Evaluation de la Coopération Universitaire et Scientifique avec le Brésil (since 2009)
- NSF G8 initiative: Performance and Introspection at Exascale

Denis BARTHOU serves as reviewers of following journals/conferences: ACM/IEEE PLDI, ACM Journal of Parallel and Distributed Computing, Compiler Construction Conference, Europar Conference, Smart Workshop.

8.2. Conference Committees

Emmanuel JEANNOT and Raymond NAMYST are chairs and organizers of the 17th International European Conference on Parallel and Distributed Computing (Euro-Par 2011)

In 2010, Raymond NAMYST was a program committee member of the following international conferences: EuroMPI, ICCN, HPCVirt, IWOMP, PMEA, A4MMC and MuCoCos.

Alexandre DENIS is member of the program committee of the RenPar conference (Rencontres Francophones du Parallelisme).

Emmanuel JEANNOT is member of the program committee of the 2011 International Conference of Parallel and Distributed Systems (IPDPS 2011) and the 13th IEEE International Conference on Computational Science and Engineering (CSE 2010). He is also member of the steering committee of the IEEE conference on cluster computing (Cluster).

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

Guillaume MERCIER is member of the program committee of the International High-Performance Computing Conference (HIPC 2010) and the IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2011). He also serves as a reviewer for the International Journal of High-Performance Computing Applications (IJHPCA).

Denis BARTHOU is member of the program committee of the Europar Conference.

8.3. Invitations

Raymond NAMYST and Cédric AUGONNET have been invited to spend one week respectively in November and May 2010 at the University of Illinois at Urbana-Champaign (USA). They have worked with the team of Prof. Wen Mei Hwu on the optimization of parallel applications for heterogeneous architectures equipped with GPU accelerators. This collaboration is supported by the INRIA-UIUC joint laboratory.

Cédric AUGONNET has been invited to spend one week in April 2010 and 3 weeks in July 2010 at the University of Tennessee in Knoxville. He has worked with the team of Professor Jack Dongarra on the port of the PLASMA and MAGMA linear algebra libraries on top of STARPU.

8.4. Seminars and invited talks

Raymond NAMYST was invited to give a talk at the WPSE'2010 International Workshop on Peta-Scale Computing Programming Environment, Languages and Tools (February 2010).

Alexandre DENIS gave a talk at the FP3C kickoff meeting about high performance communications on clusters of multicores (UVSQ, Versailles, Sep. 2010).

Alexandre DENIS gave a talk about Infiniband networking at the University of Tokyo (Tokyo, Dec. 2010).

Brice GOGLIN gave two talks about the HWLOC and KNEM software during the SuperComputing 2010 exhibition and conference (New Orleans, Nov. 2010).

Brice GOGLIN gave two presentations about high-performance computing and the Linux kernel at the Libre Software Meeting (Bordeaux, Jul. 2010).

Brice GOGLIN presented the Grid5000 initiative at the Grid day in University of Bordeaux (June 2010).

Brice GOGLIN explained the GIT source version control system at a SED seminar (Bordeaux, May 2010).

Emmanuel JEANNOT and Louis-Claude CANON gave a tutorial on performance analysis at the Grid'5000 spring school in Apr 2010.

Emmanuel JEANNOT and Raymond NAMYST gave two talks at the *Journée du calcul scientifique* organized by the CNRS in Nov. 2010 in Lyon.

Emmanuel JEANNOT gave an invited talk about MPI process placement at the CCSGC workshop (Ashville, NC, USA).

Samuel THIBAULT was invited to give a talk about STARPU at the GPU days of University of Mons (Mons, Nov. 2010).

Olivier AUMAGE, Jérôme CLET-ORTEGA and Pierre-André WACRENIER visited the Mescal team in Nov. 2010 to work on porting the BigDFT nanosimulation application based on the density functional theory on top of FORESTGOMP in the context of ANR Project ProHMPT.

Cédric AUGONNET gave two talks about the use of STARPU for 3D stencil kernels at the University of Illinois in Urbana-Champaign (Urbana, May 2010).

Cédric AUGONNET gave a talk about the use of STARPU for dense linear algebra during a "Friday Lunch" at the University of Tennessee in Knoxville (Knoxville, July 2010).

Cédric AUGONNET gave a tutorial on STARPU during the SAAHPC'10 workshop (Knoxville, July 2010).

8.5. Diffusion of the scientific culture

- 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 working group on Scientific Mediation.
- Brice GOGLIN presented the team's research work to the general public at the "Nuit des chercheurs" and to high-school student at the "Fête de la Science". He also presented research careers at the Aquitec student exhibition and at the "Ateliers métiers" day of the ENSEIRB-MATMECA engineering school.
- Emmanuel JEANNOT gave a talk at the *Unithé ou café* series at INRIA Bordeaux about the role of experiments in computer-science in May 2011.
- Denis BARTHOU and Andres CHARIF-RUBIAL made a presentation and demo of MAQAO during the meeting organized by INRIA between INRIA and SME (in Jan 2010).

8.6. Teaching

Most members of the RUNTIME team-project are involved in teaching either at University of Bordeaux or at the ENSEIRB-MATMECA engineering school. Their courses match the team's research areas, from computing architecture to distributed systems and from operating systems to parallel programming.

8.7. Miscellaneous

- Louis-Claude Canon defended his PhD thesis in Oct. 2010 [17].
- François Broquedis [16] and François Diakhaté [18] defended their PhD. thesis in Dec. 2010.
- Alin Dobre (Movidius) visited the Runtime team in Nov. 2010.
- Akihiro Nomura (University of Tokyo) visited the team from Oct. 2010 to Feb. 2011.
- Aleksandar Ilic (University of Lisbon) visited the team in May 2010.
- Sidi Mahmoudi (University of Mons) visited the team from May to June 2010.
- Élies Bergounioux from INRIA Team Concha visited the Runtime team in May 2010 in the context of the ADT AMPLI.
- Salvador Abreu (Universidade de Evora, Universidade Nova de Lisboa) visited the Runtime team in Apr. 2010 as a follow up to the former PHC Pessoa Grant from Egide.
- Vasco Pedro (Universidade de Evora) visited the Runtime team in June 2010 as a follow up to the former PHC Pessoa Grant from Egide.

9. Bibliography

Major publications by the team in recent years

[1] G. ANTONIU, L. BOUGÉ, P. HATCHER, M. MACBETH, K. MCGUIGAN, R. NAMYST. *The Hyperion system: Compiling multithreaded Java bytecode for distributed execution*, in "Parallel Computing", October 2001, vol. 27, p. 1279–1297.

[2] O. AUMAGE, L. BOUGÉ, A. DENIS, L. EYRAUD, J.-F. MÉHAUT, G. MERCIER, R. NAMYST, L. PRYLLI. *A Portable and Efficient Communication Library for High-Performance Cluster Computing (extended version)*, in "Cluster Computing", January 2002, vol. 5, n^{o} 1, p. 43-54.

[3] O. AUMAGE, E. BRUNET, N. FURMENTO, R. NAMYST. *NewMadeleine: a Fast Communication Scheduling Engine for High Performance Networks*, in "CAC 2007: Workshop on Communication Architecture for Clusters, held in conjunction with IPDPS 2007", Long Beach, California, USA, March 2007, Also available as LaBRI Report 1421-07 and INRIA RR-6085, http://hal.inria.fr/inria-00127356.

[4] O. AUMAGE, G. MERCIER. *MPICH/MadIII: a Cluster of Clusters Enabled MPI Implementation*, in "Proc. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2003)", Tokyo, IEEE, May 2003, p. 26–35.

[5] O. AUMAGE, G. MERCIER, R. NAMYST. *MPICH/Madeleine: a True Multi-Protocol MPI for High-Performance Networks*, in "Proc. 15th International Parallel and Distributed Processing Symposium (IPDPS 2001)", San Francisco, IEEE, April 2001, 51, Extended proceedings in electronic form only..

[6] 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.

[7] V. DANJEAN, R. NAMYST, R. RUSSELL. *Linux Kernel Activations to Support Multithreading*, in "Proc. 18th IASTED International Conference on Applied Informatics (AI 2000)", Innsbruck, Austria, IASTED, February 2000, p. 718-723.

[8] 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.

[9] B. GOGLIN. *High Throughput Intra-Node MPI Communication with Open-MX*, in "Proceedings of the 17th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP2009)", Weimar, Germany, IEEE Computer Society Press, February 2009, http://hal.inria.fr/inria-00331209.

[10] S. MOREAUD, B. GOGLIN. *Impact of NUMA Effects on High-Speed Networking with Multi-Opteron Machines*, in "The 19th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS 2007)", Cambridge, Massachussetts, November 2007, http://hal.inria.fr/inria-00175747.

[11] R. NAMYST. *Contribution à la conception de supports exécutifs multithreads performants*, Université Claude Bernard de Lyon, pour des travaux effectués à l'école normale supérieure de Lyon, December 2001, Habilitation à diriger des recherches.

[12] S. THIBAULT, F. BROQUEDIS, B. GOGLIN, R. NAMYST, P.-A. WACRENIER. *An Efficient OpenMP Runtime System for Hierarchical Architectures*, in "International Workshop on OpenMP (IWOMP)", Beijing,China, 6 2007, p. 148–159, http://hal.inria.fr/inria-00154502.

[13] 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.

[14] F. TRAHAY, E. 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.

[15] F. TRAHAY, A. DENIS, O. AUMAGE, R. NAMYST. *Improving Reactivity and Communication Overlap in MPI using a Generic I/O Manager*, in "EuroPVM/MPI, Recent Advances in Parallel Virtual Machine and Message Passing Interface", F. CAPPELLO, T. HERAULT, J. DONGARRA (editors), Lecture Notes in Computer Science, Springer, 2007, n° 4757, p. 170-177, http://hal.inria.fr/inria-00177167.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[16] F. BROQUEDIS. *De l'exécution d'applications scientifiques OpenMP sur architectures hiérarchiques*, Université Bordeaux 1, 351 cours de la Libération — 33405 TALENCE cedex, December 2010.

[17] L.-C. CANON. *Outils et algorithmes pour gérer l'incertitude lors de l'ordonnancement d'application sur plateformes distribuées*, Université Nancy 1, October 2010.

[18] F. DIAKHATÉ. *Contribution à l'élaboration de supports exécutifs exploitant la virtualisation pour le calcul hautes performances*, Université Bordeaux 1, 351 cours de la Libération — 33405 TALENCE cedex, December 2010.

Articles in International Peer-Reviewed Journal

[19] 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", 2010, http://hal.inria.fr/inria-00550877/en/, http://hal.inria.fr/inria-00550877.

[20] F. BROQUEDIS, N. FURMENTO, B. GOGLIN, P.-A. WACRENIER, R. NAMYST. *ForestGOMP: an efficient OpenMP environment for NUMA architectures*, in "International Journal of Parallel Programming", 2010, http://hal.inria.fr/inria-00496295.

[21] L.-C. CANON, O. DUBUISSON, J. GUSTEDT, E. JEANNOT. *Defining and Controlling the Heterogeneity of a Cluster: the Wrekavoc Tool*, in "Journal of Systems and Software", 2010, vol. 83, n° 5, p. 786-802, http://hal.inria.fr/inria-00438616.

[22] L.-C. CANON, E. JEANNOT. *Evaluation and Optimization of the Robustness of DAG Schedules in Heterogeneous Environments*, in "IEEE Transactions on Parallel and Distributed Systems", 04 2010, vol. 21 [*DOI :* 10.1109/TPDS.2009.84>], http://hal.inria.fr/inria-00430920/en/.

[23] B. GOGLIN. *NIC-assisted cache-efficient receive stack for message passing over Ethernet*, in "Concurrency and Computation: Practice and Experience", 2010, p. 1-15, http://hal.inria.fr/inria-00496301.

[24] B. GOGLIN. *High-Performance Message Passing over generic Ethernet Hardware with Open-MX*, in "Elsevier Journal of Parallel Computing", 2011, http://hal.inria.fr/inria-00533058.

International Peer-Reviewed Conference/Proceedings

[25] E. AGULLO, C. AUGONNET, J. DONGARRA, M. FAVERGE, H. LTAIEF, S. THIBAULT, S. TOMOV. *QR Factorization on a Multicore Node Enhanced with Multiple GPU Accelerators*, in "25th IEEE International Parallel & Distributed Processing Symposium", Anchorage États-Unis, 05 2011, to appear, http://hal.inria.fr/inria-00547614/en/, http://hal.inria.fr/inria-00547614.

[26] E. AGULLO, C. AUGONNET, J. DONGARRA, H. LTAIEF, R. NAMYST, J. ROMAN, S. THIBAULT, S. TOMOV. *Dynamically scheduled Cholesky factorization on multicore architectures with GPU accelerators.*, in "Symposium on Application Accelerators in High Performance Computing (SAAHPC)", Knoxville États-Unis, 07 2010, http://hal.inria.fr/inria-00547616/en/, http://hal.inria.fr/inria-00547616.

[27] C. AUGONNET, J. CLET-ORTEGA, S. THIBAULT, R. NAMYST. *Data-Aware Task Scheduling on Multi-Accelerator based Platforms*, in "16th International Conference on Parallel and Distributed Systems", Chine Shangai, Dec 2010, http://hal.inria.fr/inria-00523937.

[28] M. BELAOUCHA, D. BARTHOU, A. ELICHE, S.-A.-A. TOUATI. *FADAlib: an open source C++ library for fuzzy array dataflow analysis*, in "Intl. Workshop on Practical Aspects of High-Level Parallel Programming", Amsterdam, The Netherlands, May 2010, vol. 1, 2075—2084 [*DOI :* DOI: 10.1016/J.PROCS.2010.04.232], http://www.sciencedirect.com/science/article/B9865-506HM1Y-87/2/8ab9a2aea19e090399f478906c88f095.

[29] F. BROQUEDIS, O. AUMAGE, B. GOGLIN, S. THIBAULT, P.-A. WACRENIER, R. NAMYST. *Structuring the execution of OpenMP applications for multicore architectures*, in "International Parallel and Distributed Symposium (IPDPS 2010)", États-Unis Atlanta, Apr 2010, http://hal.inria.fr/inria-00441472.

[30] 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 "PDP 2010 - The 18th Euromicro International Conference on Parallel, Distributed and Network-Based Computing", Italie Pisa, Feb 2010, http://hal.inria.fr/inria-00429889.

[31] L.-C. CANON, E. JEANNOT, J. WEISSMAN. *A Dynamic Approach for Characterizing Collusion in Desktop Grids*, in "24th IEEE International Parallel and Distributed Processing Symposium - IPDPS 2010", États-Unis Atlanta, IEEE, 2010, p. 1-12, http://hal.inria.fr/inria-00441256.

[32] B. GOGLIN, S. MOREAUD. *Dodging Non-Uniform I/O Access in Hierarchical Collective Operations for Multicore Clusters*, in "1st Workshop on Communication Architecture for Scalable Systems, help in conjunction with IPDPS", Anchorage, AK, May 2011, Submitted.

[33] E. JEANNOT, G. MERCIER. *Near-Optimal Placement of MPI processes on Hierarchical NUMA Architectures*, in "Euro-Par 2010 Parallel Processing Europar", Ischia Italie, P. D'AMBRA, M. R. GUARRACINO, D. TALIA (editors), Lecture Notes on Computer Science, Springer, 08 2010, vol. 6272, p. 199-210, http://hal.inria.fr/inria-00544346.

[34] 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.

[35] S. MOREAUD, B. GOGLIN, D. GOODELL, R. NAMYST. *Optimizing MPI Communication within large Multicore nodes with Kernel assistance*, in "Workshop on Communication Architecture for Clusters, held in conjunction with IPDPS 2010", États-Unis Atlanta, Apr 2010, 7 p., http://hal.inria.fr/inria-00451471.

[36] S. MOREAUD, B. GOGLIN, R. NAMYST. *Adaptive MPI Multirail Tuning for Non-Uniform Input/Output Access*, in "The 17th European MPI Users Group conference", Allemagne Stuttgart, 2010, http://hal.inria.fr/inria-00486178.

[37] F. TRAHAY, A. DENIS, Y. ISHIKAWA. *A Generic and High Performance Approach for Fault Tolerance in Communication Library*, in "1st Workshop on Communication Architecture for Scalable Systems, help in conjunction with IPDPS", Anchorage, AK, May 2011, Submitted.

[38] P. DE OLIVEIRA CASTRO, S. LOUISE, D. BARTHOU. *A multidimensional array slicing DSL for Stream Programming*, in "Intl. IEEE Workshop on Multi-Core Computing Systems", Krakow, Poland, IEEE Computer Society, February 2010, p. 913–918.

[39] P. DE OLIVEIRA CASTRO, S. LOUISE, D. BARTHOU. *Reducing Memory Requirements of Stream Programs by Graph Transformations*, in "IEEE Intl. Conf. on High Performance Computing and Simulation", Caen, France, IEEE Computer Society, June 2010, 171—180.

Scientific Books (or Scientific Book chapters)

[40] E. AGULLO, C. AUGONNET, J. DONGARRA, H. LTAIEF, R. NAMYST, S. THIBAULT, S. TOMOV. *Faster, Cheaper, Better - a Hybridization Methodology to Develop Linear Algebra Software for GPUs*, in "GPU Computing Gems", WEN-MEI W. HWU (editor), Morgan Kaufmann, 09 2010, vol. 2, http://hal.inria.fr/inria-00547847/en/, http://hal.inria.fr/inria-00547847.

[41] B. GOGLIN. *Réseaux rapides et stockage distribué dans les grappes de calculateurs*, Editions Universitaires Européennes, 07 2010, http://hal.inria.fr/inria-00533064.

[42] P. VICAT-BLANC PRIMET, R. GUILLIER, S. SOUDAN, B. GOGLIN. *Réseaux de calcul - des grappes aux nuages de calcul*, Hermès Science - Lavoisier, 09 2010, http://hal.inria.fr/inria-00533072.

Research Reports

[43] C. AUGONNET, S. THIBAULT, R. NAMYST. *StarPU: a Runtime System for Scheduling Tasks over Accelerator-Based Multicore Machines*, INRIA, Mar 2010, RR-7240, http://hal.inria.fr/inria-00467677.

[44] L.-C. CANON, E. JEANNOT, J. WEISSMAN. *A Scheduling Algorithm for Defeating Collusion*, INRIA, Oct 2010, RR-7403, http://hal.inria.fr/inria-00524493.

[45] T. MA, G. BOSILCA, A. BOUTEILLER, B. GOGLIN, J. SQUYRES, J. DONGARRA. *Kernel Assisted Collective Intra-node Communication Among Multicore and Manycore CPUs*, INRIA, 12 2010, http://hal.inria.fr/inria-00544872/en/.

Other Publications

[46] A. BENOIT, L.-C. CANON, E. JEANNOT, Y. ROBERT. *On the complexity of task graph scheduling with transient and fail-stop failures*, http://hal.inria.fr/hal-00457511.

[47] P. DE OLIVEIRA CASTRO, S. LOUISE, D. BARTHOU. *Automatic Mapping of Stream Programs on Multicore Architectures*, July 2010, Workshop on Compilers for Parallel Computing.

References in notes

[48] 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.

[49] 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.

[50] 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.

[51] M. I. GORDON, W. THIES, M. KARCZMAREK, J. LIN, A. S. MELI, A. A. LAMB, C. LEGER, J. WONG, H. HOFFMANN, D. MAZE, S. AMARASINGHE. *A stream compiler for communication-exposed architectures*, in "Proceedings of the 10th international conference on Architectural support for programming languages and operating systems", New York, NY, USA, ASPLOS-X, ACM, 2002, p. 291–303, http://doi.acm.org/10.1145/605397.605428.