IN PARTNERSHIP WITH:
CNRS

Ecole nationale supérieure d'électronique, informatique et radiocommunications de Bordeaux

Université de Bordeaux

Activity Report 2011

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: Grid Computing, High Performance Computing, Scheduling, Grid'5000, High Performance Communication, Multithreading

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]

Technical Staff

Nicolas Collin [Associate Engineer, INRIA, European Project grant]
Nathalie Furmento [Research Engineer, CNRS]
Yannick Martin [Associate Engineer, INRIA]
Cyril Roelandt [Associate Engineer, INRIA, ANR grant]
Ludovic Stordeur [Associate Engineer, INRIA]
Ludovic Courtès [Research Engineer, INRIA]
François Tessier [Associate Engineer, INRIA, ANR grant (until 10/2011)]
Sébastien Barascou [Associate Engineer, INRIA, ANR grant]

PhD Students

Paul-Antoine Arras [University of Bordeaux, STMicroelectronics-INRIA CIFRE]
Cédric Augonnet [University of Bordeaux, École Normale Supérieure de Lyon 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]
Julien Jaeger [University of Versailles, ANR grant]
Stéphanie Moreaud [University of Bordeaux]
Bertrand Putigny [University of Bordeaux, INRIA grant]
François Tessier [University of Bordeaux, MESR grant (since 10/2011)]

Post-Doctoral Fellow

Remi Sharrock [IPB until 08/2011]

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,OpenMP,compiler optimizations

The RUNTIME research project takes place within the context of high-performance computing. It seeks to explore the design, the implementation and the evaluation of novel mechanisms needed by **runtime systems** for parallel computers. *Runtime systems* are intermediate software layers providing parallel programming environments with specific functionalities left unaddressed by the 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
- understand network topology and application communication pattern to optimize process placement

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

- 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/ .
- Euro-Par is a major conference in parallel and distributed computing. It has been organized in Bordeaux from August 29 to September 2, 2011. It has featured 16 topics, 25 sessions and 12 workshops. 271 papers have been submitted and 81 papers have been accepted (29.9 %). Moreover 3 invited lectures have been given. 330 persons registered at either the conference or the workshops. The website is http://europar2011.bordeaux.inria.fr/. The conference chairs were Emmanuel Jeannot, Raymond Namyst and Jean Roman. The institutions involved in the organization were INRIA, the LaBRI, the CNRS and others.

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.* [59], 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 [14] 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 [7], 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 [61] 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

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 [6], 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 33 000 lines of C.
- http://runtime.bordeaux.inria.fr/hwloc/

5.3. 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 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

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 [58] or the recent iWarp [57] 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 [23], [10].
- 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, Nicolas Collin, Nathalie Furmento, Cyril Roelandt, 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.

- 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 higher than the number of processes ; it assumes that the topology is a tree and does not require valuation of the topology (e.g. communication speed) ; it implements different placement algorithms that are switched according to the input size.
- TREEMATCH is implemented as a load-balancer in Charm++ and as an tool for performing rank reordering in OpenMPI and MPICH-2 [37]

6. New Results

6.1. High-Performance Intra-node Collective Operations

Participants: Brice Goglin, Stéphanie Moreaud.

- KNEM is known to improve the performance of point-to-point intra-node MPI communication significantly [60], [18].
- 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 [35].
- 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.

6.2. 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 [11] (*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 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 significantly [34], [18].

6.3. High-Performance Point-to-Point Communications

Participants: Alexandre Denis, 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 [29] auto-tuning mechanisms for most parameters of a communication library: rendez-vous threshold, multi-rail ratio, optimization strategies.
- We have proposed a communication protocol [33] for InfiniBand that completely amortizes the cost of memory registration, through the use of a superpipeline that overlaps communication and memory copies. We have modeled the behavior of the network and proposed auto-tuning mechanism to adapt the protocol to the hardware properties.

6.4. Improve code-coupling performance in the SALOME platform

Participants: Alexandre Denis, Sébastien Barascou.

- SALOME platform is an open source software devlopped by EDF, CEA, and OpenCascade. It is an open simulation platform with pre-processing, post-processing, interoperability with CAD models, integration with computation kernels.
- YACS is the workflow engine used for code coupling applications in SALOME. It leverages CORBA for communications between kernels. We have ported [50] YACS atop PadicoTM, our communication platform for grids. It enables CORBA connections to use InfiniBand networks. Benchmarks show a significant improvement in code coupling performance.

6.5. Hardware topology-aware MPI applications

Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.

- We have expanded our previous work dealing with MPI process placement. Indeed, our approach relied on tools and techniques which were outside the scope of the MPI standard itself. In order to allow the users to utilize our work in a portable way, we enhanced some routines of the MPI standard. We worked mainly with the MPICH2 implementation but we are also working on an OPEN MPI version as well.
- Instead of modifying the binding of the MPI processes onto the physical cores on the underlying architecture, we chose to create a new communicator for which the logical topology organization is optimized for the hardware. This work has been published in [37] and show interesting performance improvements for some class of MPI applications.
- The problem of process placement, which can be reduced to a NP-hard graph partitionning problem, can be dealt with several famous applications like Scotch or ParMETIS. To evaluate these solution with TREEMATCH, we ran several benchmarks using NAS Parellel Benchmarks and a real CFD application. On the one hand we study the quality of processes permutation (which will impact the execution time) and on the other hand the computation time of the permutation. These results will allow us to conclude about the pertinence of what graph partitioner can be used to bind processes on process units or to do a dynamic processes reordering

6.6. Mastering Heterogeneous Platforms

Participants: Cédric Augonnet, Olivier Aumage, Ludovic Courtès, Nathalie Furmento, Andra Hugo, Raymond Namyst, Samuel Thibault, Pierre-André Wacrenier.

- We continued our work on extending STARPU to master exploitation of Heterogeneous Platforms.
- We have extended the STARPU scheduler into managing *parallel tasks* which permit a better exploitation of CPUs and load balancing with GPUs.
- We have designed over STARPU a lightweight DSM over MPI, which permits to seamlessly execute STARPU applications over an MPI cluster of GPU-enhanced nodes.
- We have been developing a GCC plug-in which extends the C language with pragmas and attributes that make writing STARPU applications a lot easier.
- We have brought to STARPU support for automatically converting data between CPU and GPU formats (typically arrays of structures vs structures of arrays). We are now optimizing it.
- We have added an OpenCL interface to STARPU, SOCL [42], which permits to execute unmodified OpenCL applications over STARPU.
- We have introduced in STARPU theoretical bound support [27]: from a record of the set of tasks submitted by the application, STARPU uses linear programming to give the execution time of an ideal scheduling, which can then be compared with the actual results.
- We have continued collaboration with the University of Tennessee, Knoxville for STARPU support in the state-of-the art dense linera algebra library, Magma, in particular LU [26] and QR [27] factorizations. We have also collaborated with the University of Mons [41] and Linköping [32].
- Cédric Augonnet defended his PhD on STARPU [17].

6.7. Development of a flexible heterogeneous system-on-chip platform using a mix of programmable processing elements and hardware accelerators

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) 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 P2012 and aimed at meeting the latest codecs' requirements.‘
- This year, the privileged axis of research was directed towards dataflow-based models, which benefit from such strong, well-known properties as analyzability, schedulability and expressivity. Furthermore, dataflow programming has already been used extensively in DSP, yielding a number of dedicated software synthesis tools. We have proposed a first version of the programming model that will be evaluated later.

6.8. Sparse GMRES on heterogeneous platforms in oil extraction simulation

Participants: Olivier Aumage, Corentin Rossignon, Samuel Thibault.

- We started a study on sparse matrix factorization and system resolution on heterogeneous platforms in collaboration with Pascal Hénon from company Total, in the context of oil extraction simulation. Sparse matrix computations are notoriously difficult to efficiently run on heterogeneous platforms in the general case due to the irregular memory access patterns they generate.
- However, in the specific context of this study, Corentin Rossignon showed as part of his Master Thesis [56] that the sparsity layout of matrices generated by such oil extraction simulation problems can lead to a much higher level of efficiency on hetereogeneous platforms thanks when using a suitable sparse internal representation together with carefully written operators such as the sparse matrix-vector product together with the StarPU heterogeneous scheduler.
- Corentin Rossignon is now starting a Phd. Thesis in partnership with Total to build on these promising results.

6.9. Programming models for heterogeneous platforms

Participants: Olivier Aumage, Cyril Roelandt, Samuel Thibault, Ludovic Courtès.

- As part of Project FP3C with Japan, we started a study on to explore the use of StarPU as possible target runtime system for the XcalableMP language and compiler developed by Prof. Sato's team from University of Tsukuba. XcalableMP is a pragma-based language designed for parallelising application on clusters of multicore processors. The compiler is responsible to expand XcalableMP pragma into complex work mapping, communication and data redistribution commands.
- The study of porting XcalableMP on top of StarPU was conducted by Cyril Roelandt during his Master Thesis [55], starting from the idea that computing node with one or more attached accelerating expansion cards can be seen as a distributed platform. The results of the study showed that on the one side, the power of the XcalableMP language itself is very interesting for the goal of simplifying the port of applications on hetereogeneous platforms. However, a current assumption of the XcalableMP model is that the compiler does not insert implicit commands and behaviour except at the exact location of pragma annotations, which limit the range of optimizations available to the dynamic scheduler and memory manager of StarPU. We will thus continue to collaborate with Prof. Sato's team within the FP3C to see how these limitations could be reduced or lifted when using XcalableMP with StarPU.

- In an effort to make it easier for C programmers to benefit from StarPU, the team-project has been working on extensions to the C language allowing important StarPU concepts to be expressed concisely. These C extensions are provided as a plug-in for the GNU Compiler Collection (GCC [1]), and is now distributed as part of StarPU.

  The GCC plug-in extends the syntax and semantics of C and related languages (C++, Objective-C) using *attributes* and *pragmas*. Attributes are used, for instance, to declare StarPU *tasks* and their *implementations* for the available targets (CPU, OpenCL, CUDA, etc.) Pragmas are used notably to provide programmers a way to describe data buffers that are passed to tasks, which in turn allows the StarPU run-time support to manage data transfers between main memory and GPUs as it sees fit. Finally, tasks are invoked like regular C functions.

  In addition to easing application development, the GCC plug-in, thanks to its higher-level view of the program structure, is able to report certain classes of errors at compile-time, which would otherwise lead to run-time errors.

  This project has been led by Ludovic Courtès of Inria's Development and Experimentation Department (SED) at Bordeaux, as part of a joint development action with the SED.

6.10. Parallel Concha

Participants: Olivier Aumage, Marie-Christine Counilh.

- Within the ADT Ampli project, we contributed to the Concha CFD library developed by R. Becker's Inria Team Concha in Pau. Together with R. Becker, E. Bergounioux and D. Trujillo from Concha Team, and François Rue from SED Bordeaux we designed and experimented with the MPI parallelization and the hybrid MPI+OpenMP parallelization of the library.
- The MPI parallelization is now finalized. The OpenMP level has been successfully tested on the Vanka smoother and is now being spread in the library. We will thus continue to contribute to this parallelization work, in particular with respect to the support of 3D simulation cases.

6.11. Scientific Application Analysis and Experiments

Participants: Olivier Aumage, Denis Barthou, Andres Charif-Rubial, François Tessier, Ludovic Stordeur.

- Within the context of the ANR ProHMPT project, we contributed a thorough analysis of hot spots, data structure usages and locality issues in memory accesses of an aerodynamics application from partner CEA CESTA.
- In accordance with these results, a new version of this application has been written by the CESTA Team with redesigned, locality-friendly data structures and simplified loop scheme. This new version perfoms much better than the previous one on both 2D and 3D cases.
- We also conducted tests about the port of selected kernels of this application on accelerated hetereogeneous platforms. The results of these tests were desappointing with the first version of the application due to the layout of the main data structures that led to a lot of memory transfers between the central memory and the accelerated memory.
- We are now working on conducting these experiments with the redesigned version of the application whose new data structures should dramatically reduce the amount of data transfers.

[1] See http://gcc.gnu.org/, for more information on GCC.

6.12. Virtualization of GPUs for OpenCL

Participants: Sylvain Henry, Alexandre Denis, Denis Barthou.

- We propose a new approach for OpenCL programming, using a unique virtual accelerator instead of using the physical accelerator. Placement on the real hardware is handled by the runtime instead of the user, improving productivity and performance scalability. This proposition relies on OpenCL standard but changes the way its API is used.
- We have shown on some simple examples how this approach, using StarPU as a runtime, enables executions with a better load balance and performance. We are working on how to generalize this to more complex benchmarks. This work has been presented in Renpar[42] workshop.

6.13. Automatically Adaptating Task Grain for Hybrid Architectures

Participants: Sylvain Henry, Alexandre Denis, Denis Barthou.

- Given a parallel task graph, a runtime such as StarPU can place each task on different hardware. However, there is still the need to adapt the number of tasks, the granularity of these tasks, according to the target hardware. Due to architectures with CPUs and GPUs, it is potentially interesting to have tasks of different granularities. We explore transformations that enable to either automatically split tasks into small ones, or given some user knowledge on the tasks, decide how and when to split a large task into small ones.
- This work starts from a high-level representation of the code, using an explicit data-flow graph.

6.14. Performance modeling for power consumption reduction on the SCC

Participants: Bertrand Putigny, Brice Goglin, Denis Barthou.

- We build a model to predict performance of HPC code on the SCC ship. This model can predict runtime of regular code as well as power consumption for different frequency.
- This allows users to choose either to optimize power consumption, power efficiency or raw performance.
- This work has been published in an Intel Symposium [38].

6.15. Modeling cache coherence protocol overhead

Participants: Bertrand Putigny, Denis Barthou, Brice Goglin.

- We are building a fine grained cache model to understand common cache coherence issue.
- This model is built on a set of micro-benchmarks and can also be used to improve find some bottlenecks in memory bound code. Our set of micro-benchmarks can also be used as a test bed for new architectures [54].

6.16. Memory Performance Analysis and Tool for OpenMP codes

Participants: Andres Charif-Rubial, Denis Barthou.

- We propose a performance analysis of OpenMP codes, based on memory accesses and cache hierarchies.
- This analysis relies on memory traces for multi-threaded codes and on static analysis of binary code. Memory traces are obtained through MAQAO by static binary rewriting and are compressed online, building polyhedral iteration domains. The static analysis, mostly induction variable detection on binary code, provides the same analysis whenever possible, removing the need in some cases for dynamic instrumentation.
- The analysis focuses on a number of issues in multi-threaded executions: thread affinity issues, false sharing, cache pollution.
- This work is in collaboration with Exascale Computing Lab.

6.17. Data-layout Optimization for Stencil codes on multi-cores and GPUs

Participants: Julien Jaeger, Denis Barthou.

- We develop a new approach for stencil code generation, optimizing data-layout for multi-threaded, SIMD code on multicores and CUDA code on GPU. The transformation handles different stencil parameters, and memory constraints.
- The code generated reaches high levels of performance, outperforming related works for multicores and with similar performance on GPUs. This work is submitted to publication and was first presented in a workshop [52].

7. Contracts and Grants with Industry

7.1. Contracts with Industry

EDF R&D We participate to a contract between INRIA and EDF R&D which was granted a 6 month funding (apr. – sept. 2011). It aims at optimizing the communications of YACS, the workflow engine of the SALOME simulation platform, using our PadicoTM communication framework.

STMicroelectronics STMicroelectronics is paying 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 funded a study (apr. – sept. 2011) on porting sparse matrix computations and system resolution on heterogeneous platforms for oil extraction simulations.

8. Partnerships and Cooperations

8.1. National Initiatives

COOP 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/)

FP3C 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.

ProHMPT **Participants:** Cédric Augonnet, Olivier Aumage, Denis Barthou, Andres Charif-Rubial, Jérôme Clet-Ortega, Nathalie Furmento, Raymond Namyst, Ludovic Stordeur, François Tessier, Samuel Thibault, Pierre-André Wacrenier.

We lead a research proposal to the ANR *Cosinus* program called “ProHMPT” which was granted a three-year funding (jan. 2009 – jun. 2012). 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. Olivier AUMAGE is the head of the ANR ProHMPT project. (http://runtime.bordeaux.inria.fr/prohmpt/)

Hemera The runtime team is member of the large wigspan project Hémera started in 2010, that aims at demonstrating ambitious up-scaling techniques for large scale distributed computing by carrying out several dimensioning experiments on the Grid'5000 infrastructure, at animating the scientific community around Grid'5000 and at enlarging the Grid'5000 community by helping newcomers to make use of Grid'5000. It is not restricted to INRIA teams.

MEDIAGPU We participate to a research proposal to the ANR *CONTINT* program called "MEDIAGPU" which was granted a 30-month funding (jan. 2010 - jun. 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/).

8.2. European Initiatives

8.2.1. FP7 Project

8.2.1.1. PEPPHER

Title: Performance Portability and Programmability for Heterogeneous Many-core Architectures

Type: COOPERATION (ICT)

Defi: Computing Systems

Instrument: Specific Targeted Research Project (STREP)

Duration: 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)

See also: http://www.peppher.eu/

Abstract: PEPPHER will provide 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

8.2.2. Collaborations in European Programs, except FP7

Program: COST

Project acronym: ComplexHPC

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

Duration: may 2009 – may 2013

Coordinator: Emmanuel Jeannot

Other partners: 24 European Countries, 2 non-European counties.

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.

8.3. International Initiatives

8.3.1. INRIA Associate Teams

Morse The goal of Matrices Over Runtime Systems at Exascale (MORSE) project is 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. By designing a research framework for describing linear algebra algorithms at a high level of abstraction,the MORSE team will enable the strong collaboration between research groups in linear algebra and runtime systems needed to develop methods and libraries that fully benefit from the potential of future large-scale machines. Our project will take a pioneering step in the effort to bridge the immense software gap that has opened up in front of the High-Performance Computing (HPC) community.

8.3.2. INRIA International Partners

- The Runtime project is the representative of Inria within the *MPI Forum* which designs and maintains the *Message Passing Interface Standard* (http://www.mpi-forum.org).
- We established a collaboration with the OPEN MPI project in the context of development of the HWLOC software (see Section 5.2). This collaboration was also informally extended to the development of high-performance intra-node communication with OPEN MPI over our KNEM driver (see Section 5.3).
- Runtime is a member of the CCI project together with the Oak Ridge National Laboratory and several other american academic and industrial partners (http://www.cci-forum.org). See Section 5.1.
- 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/).

8.3.3. Visits of International Scientists

- Jan PERHAC from Trondheim University visited the runtime team as an ERCIM Fellow from March 7 to March 11. We worked on the Thor runtime system.
- Keisuke FUKUDA from Tokyo Tech visited from December 12th to Friday 16th, for the FP3C project, to port an FMM application on top of STARPU.
- Tetsuya ODAJIMA from University of Tsukuba, Japan visited the Runtime Team from September 2th to September 16th, for the FP3C Project, to integrate the XcalableMP language environment with StarPU.
- Satoshi OHSHIMA from Tokyo University visited from April 4th to April 15th, for the FP3C project, to work on FEM methods.

9. Dissemination

9.1. Animation of the scientific community

The Runtime project organized the Euro-Par 2011 conference in August in Bordeaux.

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

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

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

- 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) ;
- GENCI (http://www.genci.fr/?lang=en, since 2009) ;
- ORAP (http://www.irisa.fr/ORAP/, as the INRIA representative since 2010) ;

In 2011, Raymond NAMYST was co-chair of topic "Architecture and Networks" for the SuperComputing (SC) conference.

Raymond NAMYST has been reviewer for the PhD dissertation of Swann PERARNAU (University of Grenoble) and Christiane POUSA (University of Grenoble). He served as a member of the jury for the PhD defense of Souad KOLIAÏ (University of Versailles Saint Quentin) and Fabrice DUPROS (BRGM, Orleans).

Raymond NAMYST was a program committee member of the following international conferences: SC11, EuroMPI 2011, CASS 2011, PMEA 2011, A4MMC 2011.

Brice GOGLIN is member of the following program committees: SuperComputing 2012, EuroMPI 2011 and 2012, ISPAN 2011. He was also a reviewer for PLDI 2011 and CCGrid 2011 conferences, the CASS 2011 workshop, and the JPDC journal.

Denis BARTHOU has served as topic chair for Euro-par 2011 conference, has been member of the external review commitee of IEEE/ACM PLDI 2011, member of the program committee of SMART 2011.

Denis BARTHOU has been reviewer of the PhD dissertation of Guillaume Rizk (University of Rennes 1), Samir Ammenouche (University of Versailles Saint Quentin) and has been member of the qualifying exam for the PhD of Alexandre Duchateau (University of Illinois, Urbana Champaign).

Denis BARTHOU was involved in the reviewing process of one ANR project proposal for the call "'Infrastructures matérielles et logicielles pour la société numérique". He serves as expert in the Exascale Research Lab.

Guillaume MERCIER is member of the CCGrid 2011 program comittee and was involved in the reviewing process for the Computer and Fluids Journal.

Olivier AUMAGE was involved in the reviewing process of two ANR project proposals for the call "Infrastructures matérielles et logicielles pour la société numérique". He was also involved in the reviewing process of JPDC and Parallel Computing journals.

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 has been reviewer of the PhD dissertation of Alexandru Dobilla (Université de Franche-Comté) and Robbert Higgins (University College Dublin, Ireland).

Emmanuel JEANNOT served as reviewers of following journals: IEEE Trans. on Parallel and Dist. Syst., Parallel computing, Computing, Journal of Parallel and Distributed Computing.

Emmanuel JEANNOT is member of the steering committee of the IEEE cluster conference.

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

Emmanuel JEANNOT is member of the program committee of IPDPS 2012, heteropar 2011, PPAM, Cluster 2011 and Renpar 20 conferences.

Samuel THIBAULT is member of the program committee of HPCVirt. He was also involved in the reviewing process of the JPDC, SPE, and CCPE journals.

9.2. Seminars and invited talks

Raymond NAMYST gave a keynote speech at the International Conference On Preconditioning Techniques For Scientific And Industrial Applications (Preconditioning 2011) about “Programming heterogeneous, accelerator-based multicore machines:current situation and main challenges”.

Raymond NAMYST gave a keynote speech at the 9^{th} International Conference on Parallel Processing and Applied Mathematics (PPAM 2011) about “Programming heterogeneous, accelerator-based multicore machines: a runtime system’s perspective”.

Raymond NAMYST gave an invited talk at the 4th Workshop on UnConventional High Performance Computing (UCHPC 2011) about “programming heterogenous systems”.

Raymond NAMYST gave a lecture about “hybrid programming” at the 2011 CEA-EDF-INRIA school (Sophia Antipolis) about “Petaflop numerical simulation over hybrid parallel machines.

9.3. 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.

PhD & HdR:

PhD : Stéphanie MOREAUD, Mouvement de données et placement des tâches pour les communications haute performance sur machines hiérarchiques, Univ. Bordeaux, 12/10/2011, Brice GOGLIN and Raymond NAMYST

PhD : Cédric AUGONNET, Scheduling Tasks over Multicore machines enhanced with Accelerators: a Runtime System’s Perspective, Univ. Bordeaux, 09/12/2011, Samuel THIBAULT and Raymond NAMYST

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 : Jérôme CLET-ORTEGA, Exploitation efficace des architectures parallèles de type grappes de NUMA à l’aide de modèles hybrides de programmation, 2007/10, Raymond NAMYST and Guillaume MERCIER

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 : Andres CHARIF-RUBIAL, Performance analysis and tuning of memory accesses for multi-core codes, 2008/10, Denis BARTHOU and William JALBY (Université de Versailles Saint Quentin en Yvelines)

PhD in progress : Julien JAEGER, Iterative compilation for irregular applications, 2007/10, Denis BARTHOU

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 Le Barp)

PhD in progress: Corentin ROSSIGNON, Design of an object-oriented runtime system for oil reserve simulations on heterogeneous architectures, 2013/10, Olivier AUMAGE and Pascal HÉNON (TOTAL) and Raymond NAMYST and Samuel THIBAULT

9.4. 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 published two papers explaining multiprocessor operating systems [47] and affinities in modern computers [48] in Interstices.
- Brice GOGLIN presented the team's research work to one hundred high-school students at the "Fête de la Science".
- Stéphanie MOREAUD and Brice GOGLIN presented research careers at the Aquitec student exhibition.

10. 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, É. 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] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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^{o} 4757, p. 170-177, http://hal.inria.fr/inria-00177167.

Publications of the year

Doctoral Dissertations and Habilitation Theses

[17] C. AUGONNET. *Scheduling Tasks over Multicore machines enhanced with Accelerators: a Runtime System's Perspective*, Université Sciences et Technologies - Bordeaux I, December 2011.

[18] S. MOREAUD. *Mouvement de données et placement des tâches pour les communications haute performance sur machines hiérarchiques*, Université Sciences et Technologies - Bordeaux I, October 2011, http://hal.inria.fr/tel-00635651/en.

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, Special Issue: Euro-Par 2009", February 2011, vol. 23, p. 187–198 [*DOI :* 10.1002/CPE.1631], http://hal.inria.fr/inria-00550877.

[20] S. BENKNER, S. PLLANA, J. L. TRÄF, P. TSIGAS, U. DOLINSKY, C. AUGONNET, B. BACHMAYER, C. KESSLER, D. MOLONEY, V. OSIPOV. *PEPPHER: Efficient and Productive Usage of Hybrid Computing Systems*, in "IEEE Micro", 2011, vol. 31, n^{o} 5, p. 28-41 [*DOI :* 10.1109/MM.2011.67], http://hal.inria.fr/hal-00648480/en.

[21] 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", May 2011, http://hal.inria.fr/hal-00653477/en.

[22] 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.

[23] B. GOGLIN. *NIC-assisted cache-efficient receive stack for message passing over Ethernet*, in "Concurrency and Computation: Practice and Experience", 2011, vol. 23, n^{o} 2, p. 199-210 [*DOI :* 10.1002/CPE.1632], http://hal.inria.fr/inria-00496301/en.

[24] B. GOGLIN, J. SQUYRES, S. THIBAULT. *Hardware Locality: Peering under the hood of your server*, in "Linux Pro Magazine", July 2011, n^{o} 128, p. 28-33, http://hal.inria.fr/inria-00597961/en.

[25] E. JEANNOT, E. SAULE, D. TRYSTRAM. *Optimizing Performance and Reliability on Heterogeneous Parallel Systems: Approximation Algorithms and Heuristics*, in "Journal of Parallel and Distributed Computing", 2012, vol. 72, n^{o} 2, p. 268 – 280 [*DOI :* 10.1016/J.JPDC.2011.11.003].

International Conferences with Proceedings

[26] E. AGULLO, C. AUGONNET, J. DONGARRA, M. FAVERGE, J. LANGOU, H. LTAIEF, S. TOMOV. *LU Factorization for Accelerator-based Systems*, in "9th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 11)", Sharm El-Sheikh, Egypt, June 2011, http://hal.inria.fr/hal-00654193/en.

[27] 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, United States, May 2011, http://hal.inria.fr/inria-00547614/en.

[28] S. Benkner, S. Pllana, J. Larsson Träff, P. Tsigas, A. Richards, R. Namyst, B. Bachmayer, C. Kessler, D. Moloney, P. Sanders. *The PEPPHER Approach to Programmability and Performance Portability for Heterogeneous many-core Architectures*, in "ParCo", Ghent, Belgique, 2011, http://hal.inria.fr/hal-00661320.

[29] É. Brunet, F. Trahay, A. Denis, R. Namyst. *A sampling-based approach for communication libraries auto-tuning*, in "IEEE International Conference on Cluster Computing", Austin, United States, September 2011, http://hal.inria.fr/inria-00605735/en.

[30] L.-C. Canon, E. Jeannot. *MO-Greedy: an extended beam-search approach for solving a multi-criteria scheduling problem on heterogeneous machines*, in "International Heterogeneity in Computing Workshop", Anchorage, United States, September 2011, http://hal.inria.fr/hal-00653724/en.

[31] L.-C. Canon, E. Jeannot, J. Weissman. *A Scheduling and Certification Algorithm for Defeating Collusion in Desktop Grids*, in "International Conference on Distributed Computing Systems", Minneapolis, United States, July 2011, http://hal.inria.fr/hal-00653493/en.

[32] U. Dastgeer, C. Kessler, S. Thibault. *Flexible runtime support for efficient skeleton programming on hybrid systems*, in "International conference on Parallel Computing (ParCo)", Gent, Belgium, August 2011, http://hal.inria.fr/inria-00606200/en.

[33] A. Denis. *A High-Performance Superpipeline Protocol for InfiniBand*, in "Euro-Par 2011", Bordeaux, France, E. Jeannot, R. Namyst, J. Roman (editors), Lecture Notes in Computer Science, Springer, August 2011, vol. 6853, p. 276-287, http://hal.inria.fr/inria-00586015/en.

[34] B. Goglin, S. Moreaud. *Dodging Non-Uniform I/O Access in Hierarchical Collective Operations for Multicore Clusters*, in "CASS 2011: The 1st Workshop on Communication Architecture for Scalable Systems, held in conjunction with IPDPS 2011", Anchorage, United States, May 2011, 7p, http://hal.inria.fr/inria-00566246/en.

[35] T. Ma, G. Bosilca, A. Bouteiller, B. Goglin, J. Squyres, J. Dongarra. *Kernel Assisted Collective Intra-node MPI Communication Among Multi-core and Many-core CPUs*, in "40th International Conference on Parallel Processing (ICPP-2011)", Taipei, Taiwan, Province Of China, September 2011, http://hal.inria.fr/inria-00602877/en.

[36] A. Mazouz, S.-A.-A. Touati, D. Barthou. *Analysing the Variability of OpenMP Programs Performances on Multicore Architectures*, in "Fourth Workshop on Programmability Issues for Heterogeneous Multicores (MULTIPROG-2011)", Heraklion, Greece, Held in conjunction with: the 6th International Conference on High-Performance and Embedded Architectures and Compilers (HiPEAC), 2011, 14, http://hal.inria.fr/inria-00637957/en.

[37] G. Mercier, E. Jeannot. *Improving MPI Applications Performance on Multicore Clusters with Rank Reordering*, in "EuroMPI", Santorini, Italy, Springer Verlag, September 2011, vol. 6960, p. 39-49 [*DOI :* 10.1007/978-3-642-24449-0], http://hal.inria.fr/hal-00643151/en.

[38] B. PUTIGNY, B. GOGLIN, D. BARTHOU. *Performance modeling for power consumption reduction on SCC*, in "4th Many-core Applications Research Community (MARC) Symposium", Potsdam, Germany, H. PLATTNER (editor), December 2011, http://hal.inria.fr/hal-00649635/en.

[39] F. TRAHAY, F. RUE, M. FAVERGE, Y. ISHIKAWA, R. NAMYST, J. DONGARRA. *EZTrace: a generic framework for performance analysis*, in "IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)", Newport Beach, CA, United States, May 2011, Poster Session, http://hal.inria.fr/inria-00587216/en.

[40] S. YI, E. JEANNOT, D. KONDO, D. P. ANDERSON. *Towards Real-Time, Volunteer Distributed Computing*, in "11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2011)", Newport Beach, CA, United States, 2011, http://hal.inria.fr/hal-00654691/en.

National Conferences with Proceeding

[41] S. MAHMOUDI, P. MANNEBACK, C. AUGONNET, S. THIBAULT. *Détection optimale des coins et contours dans des bases d'images volumineuses sur architectures multicœurs hétérogènes*, in "Rencontres francophones du parallélisme", Saint-Malo, France, May 2011, http://hal.inria.fr/inria-00606195/en.

[42] H. SYLVAIN. *Programmation multi-accélérateurs unifiée en OpenCL*, in "RenPAR'20", Saint Malo, France, May 2011, http://hal.inria.fr/hal-00643257/en.

Scientific Books (or Scientific Book chapters)

[43] P. VICAT-BLANC PRIMET, B. GOGLIN, R. GUILLIER, S. SOUDAN. *Computing Networks: From Cluster to Cloud Computing*, Wiley-ISTE, May 2011, http://hal.inria.fr/inria-00590739/en.

[44] P. DE OLIVEIRA CASTRO, S. LOUISE, D. BARTHOU. *Programming Multi-core and Many-core Computing Systems*, Wiley-Blackwell, 2012, To Appear.

Books or Proceedings Editing

[45] E. JEANNOT, R. NAMYST, J. ROMAN (editors). *Euro-Par 2011 Parallel Processing - 17th International Conference, Euro-Par 2011, Bordeaux, France, August 29 - September 2, 2011, Proceedings, Part I*, Lecture Notes in Computer Science, Springer, 2011, vol. 6852.

[46] E. JEANNOT, R. NAMYST, J. ROMAN (editors). *Euro-Par 2011 Parallel Processing - 17th International Conference, Euro-Par 2011, Bordeaux, France, August 29 - September 2, 2011, Proceedings, Part II*, Lecture Notes in Computer Science, Springer, 2011, vol. 6853.

Scientific Popularization

[47] B. GOGLIN. *De votre boulangerie à un système d'exploitation multiprocesseur*, in "Interstices", February 2011, http://hal.inria.fr/inria-00566232/en.

[48] B. GOGLIN. *Et plus vite si affinités...*, in "Interstices", June 2011, http://hal.inria.fr/inria-00604025/en.

[49] R. NAMYST. *Virtualization of Hybrid Architectures*, in "Super-computers: at the frontiers of extreme computing", November 2011.

Other Publications

[50] S. Barascou. *Optimisation des communications pour les calculs parallèles avec SALOME/YACS et PadicoTM*, Université Sciences et Technologies - Bordeaux I, September 2011, http://hal.inria.fr/hal-00652882/en.

[51] A.-E. Hugo. *Composabilité de codes parallèles sur architectures hétérogènes*, Université Sciences et Technologies - Bordeaux I, 2011, http://hal.inria.fr/inria-00619654/en.

[52] J. Jaeger, D. Barthou. *Stencils sur CPU et GPU*, December 2011, Quatrième rencontres de la communauté française de compilation, Saint-Hippolyte, France.

[53] R. Namyst. *Programming heterogeneous, accelerator-based multicore machines:current situation and main challenges*, May 2011, Invited Talk, http://hal.inria.fr/inria-00590670/en.

[54] B. Putigny, D. Barthou, B. Goglin. *Modélisation du coût de la cohérence de cache pour améliorer le tuilage de boucles*, December 2011, Quatrième rencontres de la communauté française de compilation, Saint-Hippolyte, France.

[55] C. Roelandt. *Association de modèles de programmation pour l'exploitation de clusters de GPUs dans le calcul intensif*, Université Sciences et Technologies - Bordeaux I, June 2011.

[56] C. Rossignon. *Étude du GMRES dans un code de simulation de réservoir*, Université Sciences et Technologies - Bordeaux I, June 2011.

References in notes

[57] 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.

[58] 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.

[59] 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.

[60] B. Goglin, S. Moreaud. *KNEM: a Generic and Scalable Kernel-Assisted Intra-node MPI Communication Framework*, in "Journal of Parallel and Distributed Computing", 2012, Submitted.

[61] 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.