Activity Report 2014

Team RUNTIME

Efficient runtime systems for parallel architectures

RESEARCH CENTER
Bordeaux - Sud-Ouest

THEME
Distributed and High Performance Computing

Table of contents

Team RUNTIME

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

The RUNTIME project is in partnership with University of Bordeaux, CNRS and Bordeaux INP, in collaboration with LaBRI.

Creation of the Project-Team: 2004 October 07*, updated into Team:* 2014 January 01, end of the Team: 2014 December 31.

1. Members

Research Scientists

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

Faculty Members

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

Engineers

Nicolas Denoyelle [Inria]
Nathalie Furmento [CNRS]
Samuel Pitoiset [Inria, intern from Feb 2014 until Jun 2014, IJD engineer from Nov 2014 until Oct 2015]
James Tombi A Mba [Inria]

PhD Students

François Tessier [Inria, granted by ANR /13-MOEBUS-MERCIER project]
Paul-Antoine Arras [Inria, granted by ST MICROELECTRONICS (Grenoble 2) SAS]
Bertrand Putigny [Inria, until Mar 2014, granted by FP7 /13-MONTBLANC2 project]
Andra Hugo [Bordeaux INP]
Khan Muhammad Zaki Murtaza [ERCIM, Oct 2014]
Soufiane Baghdadi [Institut Telecom, until Oct 2014]
Emmanuel Cieren [granted by CEA]
Terry Cojean [Inria, from Oct 2014]
Christopher Haine [Inria, granted by FP7 /13-MONTBLANC2 project]
Suraj Kumar [Inria]
Pei Li [granted by Telecom Institute]
Jérôme Richard [Inria, from Nov 2014, granted by Inria IPL C2S@Exa]
Corentin Rossignon [granted by CIFRE TOTAL]
Emmanuelle Saillard [granted by CEA]
Marc Sergent [Inria, CEA]
Gregory Vaumourin [CEA]
Adele Villiermet [Inria, from Oct 2014, granted by ITEA Coloc]

Post-Doctoral Fellows

Lilia Ziane Khodja [Inria, granted by ANR -SONGS-CEPAGE project]

Thomas Ropars [Inria, granted by COLOC, ITEA project, from Dec 2014 to Nov. 2014]

Administrative Assistants

Sylvie Embolla [Inria]
Flavie Tregan [Inria, from Oct 2014]

Others

Pierre Celor [Univ. Bordeaux I, intern, from Apr 2014 until Jul 2014]
Clement Dussieux [Inria, intern, from Jun 2014 until Sep 2014]
Hamza El Moubarik [Inria, intern, from Feb 2014 until Jun 2014]
Rafaela Fernandes Silva [Inria, intern, from Jun 2014 until Aug 2014]
Romain Gary [Min. de l'Education Nationale, intern, Jun 2014]
Thibault Parpaite [Inria, intern, from Jun 2014 until Aug 2014]
Enguerrand Petit [Inria, intern, from Feb 2014 until Jun 2014]
Elisabeth Brunet [Institut Telecom]
Gilbert Grosdidier [CNRS]
Jean-Charles Papin [ENS Cachan]

2. Overall Objectives

2.1. Designing Efficient Runtime Systems

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

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

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

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

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

Our research project centers on three main directions:

Mastering large, hierarchical multiprocessor machines

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

Optimizing communication over high performance clusters

- Scheduling data packets over high speed networks
- New MPI implementations for Petascale computers
- Optimized intra-node communication
- Message passing over commodity networking hardware
- Influence of process placement on parallel applications performance

Integrating Communications and Multithreading

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

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

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

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

3. Research Program

3.1. Runtime Systems Evolution

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

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

Our research project centers on three main directions:

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

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

As quoted by Gao *et al.* [46], we believe it is important to expose domain-specific knowledge semantics to the various software components in order to organize computation according to the application and architecture. Indeed, the whole software stack, from the application to the scheduler, should be involved in the parallelizing, scheduling and locality adaptation decisions by providing useful information to the other components. Unfortunately, most operating systems only provide a poor scheduling API that does not allow applications to transmit valuable *hints* to the system.

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

This is also the reason why we explore new languages and compiler optimizations to better use domain specific information. We propose a new domain specific language, QIRAL, to generate parallel codes from high level formulations for Lattice QCD problems. QIRAL describes the formulation of the algorithms, of the matrices and preconditions used in this domain and generalizes languages such as SPIRAL used in auto-tuning library generator for signal processing applications. Lattice QCD applications require huge amount of processing power, on multinode, multi-core with GPUs. Simulation codes require to find new algorithms and efficient parallelization. So far, the difficulties for orchestrating parallelism efficiently hinder algorithmic exploration. The objective of QIRAL is to decouple algorithm exploration with parallelism description. Compiling QIRAL uses rewriting techniques for algorithm exploration, parallelization techniques for parallel code generation and potentially, runtime support to orchestrate this parallelism. Results of this work have been published in [12].

Following this effort, and through the combined analysis of the code behavior, at compile time and at runtime, MAQAO can then help users to better pinpoint and quantify performance issues in OpenMP codes, find load imbalance between threads, size of working sets, false sharing situations...

We proposed in [22] to combine static and dynamic dependence analysis for the detection of vectorization opportunities. MAQAO then estimates the potential gain that could be reached through vectorization and identifies the required code transformations, either by changing loop control or data layout.

Aside from greedily invading all these new cores, demanding HPC applications now throw excited glances at the appealing computing power left unharvested inside the graphical processing units (GPUs). A strong demand is arising from the application programmers to be given means to access this power without bearing an unaffordable burden on the portability side. Efforts have already been made by the community in this respect but the tools provided still are rather close to the hardware, if not to the metal. Hence, we decided to launch some investigations on addressing this issue. In particular, we have designed a programming environment named STARPU that enables the programmer to offload tasks onto such heterogeneous processing units and gives that programmer tools to fit tasks to processing units capability, tools to efficiently manage data moves to and from the offloading hardware and handles the scheduling of such tasks all in an abstracted, portable manner. The challenge here is to take into account the intricacies of all computation unit: not only the computation power is heterogeneous among the machine, but data transfers themselves have various behavior depending on the machine architecture and GPUs capabilities, and thus have to be taken into account to get the best performance from the underlying machine. As a consequence, STARPU not only pays attention to fully exploit each of the different computational resources at the same time by properly mapping tasks in a dynamic manner according to their computation power and task behavior by the means of scheduling policies, but it also provides a distributed shared-memory library that makes it possible to manipulate data across heterogeneous multicore architectures in a high-level fashion while being optimized according to the machine possibilities. In addition to this, the scheduling policy of STARPU has been modularized; this makes it easy to experiment with state of the art theoretical scheduling strategies. Last but not least, STARPU works over clusters, by extending the shared-memory view over the MPI communication library. This allows, with the same sequential-looking application source code, to tackle all architectures from small multicore systems to clusters of heterogeneous systems. We extended OpenCL capabilities by proposing to use, transparently, STARPU as an OpenCL device [23].

On complex multicore, heterogeneous architectures, memory accesses often correspond in HPC application to performance bottlenecks. Indeed, either the code is memory bound, and restructuring data layout in order to take advantage of any reuse or spacial locality is essential. If the architecture has different types of memory (such as GPU with texture caches for instance), the code should exploit their features. Or the code is compute bound and in this case, SIMD vectorization represents the key for achieving high performance. Data structures may need to be changed in order to allow the compiler to automatically vectorize, or to efficiently vectorize. performance may only be reached only at the cost of data layout restructuration. In order to better optimize data layout and parallelization, we proposed performance model for the memory hierarchy [26], [12]. Compared to other existing models, this model takes into account the costs due to the coherence protocol, the contention and the capacity of caches. It is built on top of parallel micro-benchmark results and thus can adapt to a wide range of architectures, and it aggregates these benchmark results for large code performance prediction. This model has been applied with success to communications on shared memory machines [27]. For specific memory, we have explored the opportunities and benefits of data restructuration, in collaboration with CEA [31]. Finally, data restructuration for SIMDization have been explored through the performance tuning tool MAQAO [22].

Optimizing communications over high performance clusters and grids Using a large panel of mechanisms such as user-mode communications, zero-copy transactions and communication operation offload, the critical path in sending and receiving a packet over high speed networks has been drastically reduced over the years. Recent implementations of the MPI standard, which have been carefully designed to directly map *basic* point-to-point requests onto the underlying low-level interfaces, almost reach the same level of performance for very basic point-to-point messaging

requests. However more complex requests such as non-contiguous messages are left mostly unattended, and even more so are the irregular and multiflow communication schemes. The intent of the work on our NEWMADELEINE communication engine, for instance, is to address this situation thoroughly. The NEWMADELEINE optimization layer delivers much better performance on *complex* communication schemes with negligible overhead on basic single packet point-to-point requests. Through Mad-MPI, our proof-of-concept implementation of a subset of the MPI API, we intend to show that MPI applications can also benefit from the NEWMADELEINE communication engine.

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

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

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

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

Moreover, the design of distributed parallel code, integrating both MPI and OpenMP, is complex and error-prone. Deadlock situations may arise and are difficult to detect. We proposed an original approach, based on static (compile-time) analysis and runtime verification in order to detect deadlock situation but also to pinpoint the cause of such deadlock [28], [15].

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. New Software and Platforms

5.1. Common Communication Interface

Participant: Brice Goglin.

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

5.2. Hardware Locality

Participants: Brice Goglin, Samuel Thibault.

- *Hardware Locality* (HWLOC) is a library and set of tools aiming at discovering and exposing the topology of machines, including processors, cores, threads, shared caches, NUMA memory nodes and I/O devices.
- It builds a widely-portable abstraction of these resources and exposes it to the application so as to help them adapt their behavior to the hardware characteristics.
- HWLOC targets many types of high-performance computing applications [2] [20], from thread scheduling to placement of MPI processes. Most existing MPI implementations, several resource managers and task schedulers already use HWLOC.
- HWLOC is developed in collaboration with the OPEN MPI project. The core development is still mostly performed by Brice GOGLIN and Samuel THIBAULT from the RUNTIME team-project, but many outside contributors are joining the effort, especially from the OPEN MPI and MPICH2 communities.
- HWLOC is composed of 30 000 lines of C.
- http://www.open-mpi.org/projects/hwloc

5.3. Network Locality

Participant: Brice Goglin.

- *Netloc Locality* (NETLOC) is a library that extends hwloc to network topology information by assembling hwloc knowledge of server internals within graphs of inter-node fabrics such as Ethernet or Infiniband.
- NETLOC targets the same challenges as hwloc but focuses on a wider spectrum by enabling cluster-wide solutions such process placement [21].
- NETLOC is developed in collaboration with University of Wisconsin in LaCrosse and Cisco, within the OPEN MPI project.
- NETLOC is composed of 15 000 lines of C. It was recently merged in the HWLOC repository was better integration.
- http://netloc.org

5.4. KNem

Participant: Brice Goglin.

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

5.5. Open-MX

Participant: Brice Goglin.

- The OPEN-MX software stack is a high-performance message passing implementation for any generic ETHERNET interface.
- It was developed within our collaboration with Myricom, Inc. as a part of the move towards the convergence between high-speed interconnects and generic networks.
- OPEN-MX exposes the raw ETHERNET performance at the application level through a pure message passing protocol.
- While the goal is similar to the old GAMMA stack [45] or the recent iWarp [44] implementations, OPEN-MX relies on generic hardware and drivers and has been designed for message passing.
- OPEN-MX is also wire-compatible with Myricom MX protocol and interface so that any application built for MX may run on any machine without Myricom hardware and talk other nodes running with or without the native MX stack.
- OPEN-MX is also an interesting framework for studying next-generation hardware features that could help ETHERNET hardware become legacy in the context of high-performance computing. Some innovative message-passing-aware stateless abilities, such as multiqueue binding and interrupt coalescing, were designed and evaluated thanks to OPEN-MX [5].
- Brice GOGLIN is the main contributor to OPEN-MX. The software is already composed of more than 45 000 lines of code in the Linux kernel and in user-space.
- http://open-mx.gforge.inria.fr/

5.6. StarPU

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

- STARPU permits high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with GPGPUs or Xeon Phi processors.
- STARPU offers a unified offloadable task abstraction named codelet.In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architectures (e.g. one function for CUDA and one function for CPUs).
- STARPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine. A high-level data management library enforces memory coherency over the machine: before a codelet starts (e.g. on an accelerator), all its data are transparently made available on the compute resource.
- STARPU obtains portable performances by efficiently (and easily) using all computing resources at the same time.
- STARPU also takes advantage of the heterogeneous nature of a machine, for instance by using scheduling strategies based on auto-tuned performance models.
- STARPU can also leverage existing parallel implementations, by supporting *parallel tasks*, which can be run concurrently over the machine.
- STARPU provides *scheduling contexts* which can be used to partition computing resources. Scheduling contexts can be dynamically resized to optimize the allocation of computing resources among concurrently running libraries.
- STARPU provides integration in MPI clusters through a lightweight DSM over MPI.
- STARPU provides a scheduling platform, which makes it easy to implement and experiment with scheduling heuristics
- STARPU comes with a plug-in for the GNU Compiler Collection (GCC), which extends languages of the C family with syntactic devices to describe STARPU's main programming concepts in a concise, high-level way.
- STARPU has support for simulating the execution, by using the simgrid simulator, which allows to reproduce experiments on a remote system, or to even virtually modify the platform used to run the application
- STARPU has been extended to provide runtime support for the KLANG-OMP OpenMP compiler. StarPU's OpenMP runtime support is compliant with most OpenMP 3.0 constructs, and also supports new dependent tasks and accelerated targets constructs introduced by OpenMP 4.0.
- http://runtime.bordeaux.inria.fr/StarPU/

5.7. Klang-OMP

Participants: Olivier Aumage, Nathalie Furmento, Samuel Pitoiset, Samuel Thibault.

- The KLANG-OMP software is a source-to-source OpenMP compiler for languages C and C++. It is developed as part of the Inria development action "ADT K'STAR " jointly managed by Inria teams MOAIS (Inria Montbonnot) and RUNTIME (Inria Bordeaux - Sud-Ouest). The KLANG-OMP compiler translates OpenMP directives and constructs into API calls from the StarPU runtime system or the XKaapi runtime system (XKaapi is developed by the MOAIS team).
- The KLANG-OMP compiler is virtually fully compliant with OpenMP 3.0 constructs.
- The KLANG-OMP compiler supports OpenMP 4.0 dependent tasks and accelerated targets.

5.8. Kastors

Participants: Olivier Aumage, Nathalie Furmento, Samuel Pitoiset, Samuel Thibault.

- The KASTORS software is a suite of benchmarks for testing the performance of OpenMP compilers on codes making use of the new dependent tasks OpenMP 4.0 constructs. It is ported and maintained as part of the Inria development action "ADT K'STAR " jointly managed by Inria teams MOAIS (Inria Montbonnot) and RUNTIME (Inria Bordeaux - Sud-Ouest). It is constituted of well known computing kernels that have been ported on the OpenMP 4.0 dependent tasks model. The KASTORS suite has been introduced to the OpenMP community during the IWOMP 2014 conference [32].

5.9. NewMadeleine and PIOMan

Participant: Alexandre Denis.

- NEWMADELEINE is a 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.
- PIOMAN is a generic framework to be used by communication libraries, that brings seamless asynchronous progression of communication by opportunistically using available cores. It uses system threads and thus is composable with any runtime system used for multithreading.
- PIOMAN is closely integrated with the NEWMADELEINE communication library and PadicoTM.
- The reference software development branch of the NEWMADELEINE software consists in 60 000 lines of code. NEWMADELEINE is available on various networking technologies: Myrinet, Infiniband, Quadrics and ETHERNET. It is developed and maintained by Alexandre DENIS.
- http://pm2.gforge.inria.fr/newmadeleine/

5.10. 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://pm2.gforge.inria.fr/PadicoTM/

5.11. MAQAO

Participants: Denis Barthou, Olivier Aumage, Christopher Haine, James Tombi A Mba.

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

5.12. QIRAL

Participants: Denis Barthou, Olivier Aumage.

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

5.13. TreeMatch

Participants: Emmanuel Jeannot, Guillaume Mercier, François Tessier.

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

6. New Results

6.1. Highlights of the Year

- This year we started very large collaborations with the BULL/Atos company. WE started one European project, one PIA french project and one PhD thesis. The amount of Person Year funded with this project exceed 10. The research we will do with Bull covers resource management, process placement, platform modeling, application modeling, affinity abstraction.
- The StarPU software is used by CEA for automatically distributing linear algebra on their cluster of 144 hybrid nodes.

6.2. Task scheduling over heterogeneous architectures

We continued our work on extending StarPU to master exploitation of Heterogeneous Platforms through dynamic task scheduling, with a now-imminent release of StarPU 1.2.

We have improved the simulation support with SimGrid, to augment the accuracy of the simulated execution according to the hardware capabilities [30].

We have collaborated with various research projects to leverage the potential of StarPU. We have improved the support for the PaStiX and QR-Mumps sparse matrix solvers, thus obtaining competitive performance on CPUs and on CPUs+GPUs [25]. We have improved the MPI communication engine of StarPU to get better performance with the EADS hmatrix solver.

We have obtained very good performance and scalability with a Cholesky factorization distributed over a cluster of 144 heterogeneous nodes hosted at CEA.

We have studied the theoretical performance bound that can be achieved for the Cholesky factorization, reproduced the performance of a theorically optimal scheduled, shown that the classical HEFT heuristic is far from it, that more application-specific heuristics allow to get performance closer to the peak, and that the peak is not reachable with simple heuristics, because it requires non-trivial task order inversions.

In relationship with the ADT K'Star effort of building the Klang-OMP OpenMP compiler and putting together the Kastors benchmark suite, StarPU has been extended to provide an OpenMP-enabled runtime support for Klang-OMP. In particular, the StarPU OpenMP Runtime Support implements *preemptible* tasks required for OpenMP, using the concept of continuations, while maintaining interoperability with StarPU regular, non-blocking tasks, and while preserving the heterogeneous, performance model-based scheduling capabilities of StarPU.

The Klang-OMP C/C++ OpenMP compiler co-developed with Inria Team MOAIS enables plain OpenMP applications to run un-modified on top of the StarPU runtime system, thus significantly increasing the performance portability potential of StarPU.

6.3. Modeling hierarchical platform memory performance with microbenchmarks

Bertrand Putigny developed a new memory performance model based on micro-benchmarks during his PhD. He transforms parallel codes such as OpenMP into memory access skeleton before predicting memory buffer states in caches and using benchmarks outputs to predict the runtime. This model successfully predict the performance behavior of several memory-bound kernels [26].

We also used this model to study the impact of memory caches on the performance on intra-node MPI communication [27].

6.4. Static modeling of clusters of multicore and heterogeneous nodes

We improved the hwloc software to better manage clusters of nodes. This first includes the management of HPC node I/O devices by providing easy ways to retrieve the locality of GPUs and network interfaces. A scalable global view of clusters can be built by factorizing the common topology information that is usually shared by many similar nodes [20]. Finally the topology of the network assembling all these nodes can be exposed in a generic technology-independent manner using the new netloc tool [21] that is now part of hwloc.

6.5. Multithreaded communications

We have proposed a full rewrite of the PIOMAN software, to make it rely on system threads rather than on the now obsolete MARCEL thread scheduler. It makes it more portable, composable with any runtime system used for multithreading, and more scalable. We have shown [19][18] that it features good properties with regard to asynchronous communication progression and multithreaded communications in applications.

6.6. Toplogy-aware load balancing in Charm++

Charm++ implements a fine-grained paradigm based on migratable computing objects. This programming model is designed to run large-scale experiments and provide a dynamic load balancing system to optimize it. Our previous Charm++ load balancer designed for communication-bound applications was improved to scale on large platforms. More precisely, we worked on the network awareness of this algorithm by using LibTopoMap. Our topology-aware load balancing algorithm was also restructured to be parallel and distributed. These enhancements were validated on the Blue Waters supercomputer at Urbana-Champaign, IL. Finally, We have begun to carry out experiments on real application modeling seismic wave propagation.

6.7. Topology-aware ressource allocation

On the one hand SLURM already provides topology aware placement techniques to promote the choice of group of nodes that are placed on the same network level, connected under the same network switch or even placed close to each other so as to avoid long distance communications. On the other hand users can map tasks in a parallel application to the physical processors on the chosen nodes, based on the communication topology.

Our goal is to take in account, in SLURM, placement process, hardware topology, and application communication pattern too. We have implemented a new selection option for the cons_res plugin in SLURM 2.6.5. In this case the usually best fit algorithm used to choose nodes is replaced by Treematch, an algorithm to find the best placement among the free nodes list in light of a given application communication matrix. Tests and evaluation of this feature are in progress.

6.8. Scheduling of dynamic streaming applications on hybrid embedded MPSoCs

The work on the dataflow scheduler has continued so as to improve it: it is now simper and more efficient. Moreover, an H.264 video decoder implementation from STMicroelectronics has been ported onto the developed execution model to conduct more significant experiments. This application exhibits a higher level of complexity and variability, which is the reason why it is well suited for assessing the scheduler's reactivity. Furthermore, an important groundwork has been carried out to enable software support for parts of the application, which enlarges considerably the design space and allow to benefit from better flexibility. In parallel, some earlier work on list scheduling under memory constraints has been extended and published in an international journal [11].

6.9. Performance model for multithreaded applications on multi-core processors

Concerning data locality, researches have shown a tradeof in groupement strategy for process mapping. We have to deal with balanced improvement of several aspects such as threads synchronizations or resource exploitation. Weighting those criterias can only be achieved according to a certain knowledge of both the application and the machine.

Thus, we are working on modeling threads affinity and weights on machines topology to improve a placement method based on the TreeMatch algorithm using new metrics. Several experiences have lead us to the conclusion that it is very hard to identify the key hints and to understand application needs.

Consequently, we are developping a visual tool which displays hardware counters aggregated and mapped on the system topology to identify dynamically those hardware narrows during execution, and understand processes placement effects on them. We hope to achieve a better comprehension of process placement consequences on resources usage by applications.

7. Bilateral Contracts and Grants with Industry

7.1. Bilateral Grants with Industry

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

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

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

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

8. Partnerships and Cooperations

8.1. Regional Initiatives

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

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

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

8.2. National Initiatives

8.2.1. ANR

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

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

Identification: ANR-13-MONU-0007

Coordinator: Inria Bordeaux/LaBRI

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

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

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

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

Identification: ANR-11INFR01306

Coordinator: Martin Quinson (Inria Nancy)

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

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

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

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

Coordinator: Denis Trystram (Inria Rhône-Alpes)

Other partners: Inria Bordeaux.

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

8.2.2. ADT - Inria Technological Development Actions

ADT K'Star (http://kstar.gforge.inria.fr/#!index.md)

Participants: Olivier Aumage, Nathalie Furmento, Samuel Pitoiset, Samuel Thibault.

Inria ADT Campaign 2013, 10/2013 - 9/2015 (24 months)

Coordinator: Thierry Gautier (team MOAIS, Inria Montbonnot) and Olivier Aumage (team RUNTIME, Inria Bordeaux - Sud-Ouest)

Abstract: The Inria action ADT K'Star is a joint effort from Inria teams MOAIS and RUNTIME to design the KLANG-OMP source-to-source OpenMP compiler to translate OpenMP directives into calls to the API of MOAIS and RUNTIME respective runtime systems (XKaapi for MOAIS, StarPU for RUNTIME).

8.2.3. IPL - Inria Project Lab

C2S@Exa - Computer and Computational Sciences at Exascale **Participant:** Olivier Aumage.

Inria IPL 2013 - 2017 (48 months)

Coordinator: Stéphane Lantéri (team Nachos, Inria Sophia)

Since January 2013, the team is participating to the C2S@Exa http://www-sop.inria.fr/c2s_at_exa Inria Project Lab (IPL). This national initiative aims at the development of numerical modeling methodologies that fully exploit the processing capabilities of modern massively parallel architectures in the context of a number of selected applications related to important scientific and technological challenges for the quality and the security of life in our society. This collaborative effort involves computer scientists that are experts of programming models, environments and tools for harnessing massively parallel systems, algorithmists that propose algorithms and contribute to generic libraries and core solvers in order to take benefit from all the parallelism levels with the main goal of optimal scaling on very large numbers of computing entities and, numerical mathematicians that are studying numerical schemes and scalable solvers for systems of partial differential equations in view of the simulation of very large-scale problems.

MULTICORE - Large scale multicore virtualization for performance scaling and portability

Participants: Emmanuel Jeannot, Denis Barthou [RUNTIME project-team, Inria Bordeaux - Sud-Ouest].

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

8.3. European Initiatives

8.3.1. FP7 & H2020 Projects

8.3.1.1. Mont-Blanc 2

Type: FP7

Defi: Special action

Instrument: Integrated Project

Objectif: Exascale computing platforms, software and applications

Duration: October 2013 - September 2016

Coordinator: Alex Ramirez (UPC)

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

Inria contact: Denis Barthou

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

- To complement the effort on the Mont-Blanc system software stack, with emphasis on programmer tools (debugger, performance analysis), system resiliency (from applications to architecture support), and ARM 64-bit support

- To produce a first definition of the Mont-Blanc Exascale architecture, exploring different alternatives for the compute node (from low-power mobile sockets to special-purpose high-end ARM chips), and its implications on the rest of the system
- To track the evolution of ARM-based systems, deploying small cluster systems to test new processors that were not available for the original Mont-Blanc prototype (both mobile processors and ARM server chips)
- To provide continued support for the Mont-Blanc consortium, namely operations of the original Mont-Blanc prototype, the new small scale prototypes and hands-on support for our application developers

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

8.3.1.2. HPC-GA

Type: FP7

Defi: NC

Instrument: International Research Staff Exchange Scheme

Objectif: NC

Duration: January 2012 - December 2014

Coordinator: Jean-François Méhaut (UJF, France)

Partner: UFRGS, Inria, BRGM, BCAM et UNAM.

Inria contact: Jean-François Mehaut

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

8.3.2. Collaborations in European Programs, except FP7 & H2020

Program: **ITEA2**

Project acronym: COLOC

Project title: The Concurrency and Locality Challenge

Duration: November 2014 - November 2017

Coordinator: BULL

Other partners: BULL SA (France); Dassault Aviation (France) ; Enfeild AB (Sweden); Scilab entreprise (France); Teratec (France); Inria (France); Swedish Defebnse Research Agency - FOI (France); UVSQ (France).

Abstract: The COLOC project aims at providing new models, mechanisms and tools for improving applications performance and supercomputer resources usage taking into account data locality and concurrency.

——

Program: **COST**

Project acronym: NESUS

Project title:Network for Ultrascale Computing

Duration: April 2014 - April 2018

Coordinator: University Carlos III de Madrid

Other partners: More than 35 European Countries.

Abstract: Ultrascale systems are envisioned as large-scale complex systems joining parallel and distributed computing systems that will be two to three orders of magnitude larger that today's systems. The EU is already funding large scale computing systems research, but it is not coordinated across researchers, leading to duplications and inefficiencies. The goal of the NESUS Action is to establish an open European research network targeting sustainable solutions for ultrascale computing aiming at cross fertilization among HPC, large scale distributed systems, and big data management. The network will contribute to glue disparate researchers working across different areas and provide a meeting ground for researchers in these separate areas to exchange ideas, to identify synergies, and to pursue common activities in research topics such as sustainable software solutions (applications and system software stack), data management, energy efficiency, and resilience. Some of the most active research groups of the world in this area are members of this proposal. This Action will increase the value of these groups at the European-level by reducing duplication of efforts and providing a more holistic view to all researchers, it will promote the leadership of Europe, and it will increase their impact on science, economy, and society.

8.4. International Initiatives

8.4.1. Inria International Labs

JLPC Inria joint-Lab on Extreme Scale Computing:

Coordinators: Franck Cappello and Marc Snir.

Other partners: Argonne National Lab, Inria, University of Urbanna Champaign, Tokyo Riken, Jülich Supercomputing Center, Barcelona Supercomputing Center.

Abstract: The Joint Laboratory is based at Illinois and includes researchers from Inria, and the National Center for Supercomputing Applications, ANL, Riken, Jülich, and BSC. It focuses on software challenges found in extreme scale high-performance computers.

8.4.2. Inria Associate Teams

MORSE Matrices Over Runtime Systems at Exascale

Inria Associate-Teams program: 2011-2016

Coordinator: Emmanuel Agullo (Hiepacs)

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

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

8.4.3. Inria International Partners

8.4.3.1. Informal International Partners

We collaborate with the following team.

- INESC-ID, Lisbon, Portugal on appplication moldeling.
- UWLAX (Wisconsin) works with us on network topology modeling;
- we collaborate with ICL at University of Tennessee on instrumenting MPI applications and modeling platforms (works on HWLOC take place in the context of the OPEN MPI consortium) and MPI and process placement
- On the industrial side collaborate with Cisco Systems about network topologies and platform models and Intel on modeling many-core platforms and BULL on memory hierarchy modeling.
- ETH Zurich (Switzerland), on topology mapping;
- PPL (U. Illinois at Urbana Champaign) on topology-aware load-balancing (through the Inria-Urbana-Argonne Joint Lab).
- University of Tokyo and Riken on the adaptation of MPI and runtime systems to MIC processors.
- Oak Ridge National Laboratory on high-performance network programming interfaces.

8.4.4. Participation In other International Programs

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

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

Identification: ANR-10-JST-002

Coordinator: Serge Petiton (Inria Saclay)

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

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

SEHLOC Scheduling evaluation in heterogeneous systems with hwloc

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

Coordinator: Brice Goglin

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

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

NextGN Preparing for Next Generation Numerical Simulation Platforms

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

Coordinator: Franck Capello, Marc Snir and Yves Robert

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

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

8.5. International Research Visitors

8.5.1. Visits of International Scientists

8.5.1.1. Internships

- Malik Muhammad Zaki Murtaza Khan from Dept. of Computer and Information Science (IDI), Norwegian University of Science and Technology, Trondheim, Norway visited us for one week in October.

9. Dissemination

9.1. Promoting Scientific Activities

9.1.1. Scientific events organisation

9.1.1.1. General chair, scientific chair

Emmanuel JEANNOT was a program chair of Heteropar 2014.

9.1.1.2. Member of the steering committee

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

9.1.2. Scientific events selection

9.1.2.1. Member of the conference program committee

Brice GOGLIN was a program committee member of ICCCN 2014, EuroMPI/ASIA 2014, CARLA 2014, HiPC 2014, Hot Interconnects 2014 and Cluster 2014. Emmanuel JEANNOT was a program committee member of CCGRID'2014, EuroMPI/ASIA 2014, HiPC 2014, Compas 2014. Samuel THIBAULT was a program committee member of IPDPS 2014, HIPC 2014 and MuCoCos 2014. Alexandre DENIS was a program committee member of HiPC 2014, Heteropar 2014, and Realis.

9.1.2.2. Reviewer

The members of the team reviewe numerous papers for various internationcal conferences such as IPDPS, Super-Computing, Euro-Par, ICPP

9.1.3. Journal

9.1.3.1. Member of the editorial board

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

9.1.3.2. Reviewer

Emmanuel JEANNOT was reviewer of IEEE TPDS, Parallel Computing, JPDC, IJPP. Samuel THIBAULT was reviewer for IJHPC.

9.1.4. Scientific project selection

9.1.4.1. Reviewer

Olivier AUMAGE reviewed a project proposal for the CORE 2014 call of Luxembourg's FNR research funding agency.

9.2. Teaching - Supervision - Juries

9.2.1. Teaching

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

9.2.2. Supervision

HDR: Brice Goglin, Towards generic Communication Mechanisms and better Affinity Management in Clusters of Hierarchical Nodes [9], 2014/04.

PhD: Bertrand Putigny, Benchmark-driven Approaches to Performance Modeling of Multi-Core Architectures [10], 2014/03, Denis Barthou and Brice Goglin.

PhD: Andra Hugo , Composability of parallel codes over heterogeneous platforms, 2014/12, Abdou Guermouche and Pierre-André Wacrenier and Raymond Namyst.

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: Corentin Rossignon , Design of an object-oriented runtime system for oil reserve simulations on heterogeneous architectures, 2012/04, Olivier Aumage and Pascal Hénon (TOTAL) and Raymond Namyst and Samuel Thibault

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

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

PhD in progress: Soufiane Baghdadi , Collaboration entre compilateur et support d'exécution pour les applications parallèles 2011/10, Elisabeth Brunet (Telecom SudParis), Jean-François Trahay (Telecom SudParis) , Denis Barthou

PhD in progress: Marc Sergent , Passage à l'échelle de moteur d'exécution à base de graphes de tâches, 2013/09, Olivier Aumage , David Goudin (CEA/CESTA), Samuel Thibault , Raymond Namyst

PhD in progress: Suraj Kumar , Stratégies d'ordonnancement dynamique pour l'algèbre linéaire dense, 2013/12, Emmanuel Agullo , Olivier Beaumont , Samuel Thibault

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

PhD in progress: Christopher Haine, Estimating efficiency and automatic restructuration of data layout, 2014/01, Olivier Aumage, Denis Barthou

PhD in progress: Jérôme Richard, Conception of a software component model with task scheduling for many-core based parallel architecture, application to the Gysela5D code, 2014/11, Christian Perez (LIP/ENSL), Julien Bigot (Maison de la Simulation), Olivier Aumage, Guillaume LATU (IRFM).

9.2.3. Juries

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

- Cédric Valensi (UVSQ, President)

Brice GOGLIN was member of the PhD defense jury of the following candidates:

- Sylvain Didelot (UVSQ, Reviewer)
- Robert Rey Exposito (Universidade Da Coruna, Examinator)

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

- Georges Markomanolis (ENS-Lyon, Reviewer)
- Aleksandar Ilic (Univ. Of Porto, Reviewer)
- Sergio Aldea Lopez (Univ. Of Valladolid, Reviewer)
- Sabastien Valat (UVSQ, Examinator)
- Cristian Ruiz (Univ. Of Grenoble, President)

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

- Florence Monna (LIP6, Examinator)

9.3. Popularization

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

Brice GOGLIN gave talks about Software releases and Source version control with GIT in internal Inria seminars.

Samuel THIBAULT gave a talk about the structure of Internet and questions of security at "Unithé ou Café"

Olivier AUMAGE gave a talk at Seminar Modeling, at the Maison de la Simulation on the StarPU Runtime System.

Runtime organized a 2-days PRACE Advanced Training Center session where several member of the team gave talks about programming heterogeneous parallel architectures with tools such as StarPU and hwloc.

10. Bibliography

Major publications by the team in recent years

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

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

[3] F. BROQUEDIS, N. FURMENTO, B. GOGLIN, P.-A. WACRENIER, R. NAMYST. *ForestGOMP: an efficient OpenMP environment for NUMA architectures*, in "International Journal on Parallel Programming, Special Issue on OpenMP; Guest Editors: Matthias S. Müller and Eduard Ayguadé", 2010, vol. 38, n^{o} 5, pp. 418-439 [*DOI :* 10.1007/s10766-010-0136-3], http://hal.inria.fr/inria-00496295

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

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

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

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

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

Publications of the year

Doctoral Dissertations and Habilitation Theses

[9] B. GOGLIN. *Towards generic Communication Mechanisms and better Affinity Management in Clusters of Hierarchical Nodes*, Université de Bordeaux, April 2014, Habilitation à diriger des recherches, https://tel.archives-ouvertes.fr/tel-00979512

[10] B. PUTIGNY. *Benchmark-driven Approaches to Performance Modeling of Multi-Core Architectures*, Université Sciences et Technologies - Bordeaux I, March 2014, https://tel.archives-ouvertes.fr/tel-00984791

Articles in International Peer-Reviewed Journals

[11] P.-A. ARRAS, D. FUIN, E. JEANNOT, A. STOUTCHININ, S. THIBAULT. *List Scheduling in Embedded Systems Under Memory Constraints*, in "International Journal of Parallel Programming", November 2014 [*DOI :* 10.1007/s10766-014-0338-1], https://hal.inria.fr/hal-01087067

[12] D. BARTHOU, O. BRAND-FOISSAC, O. PENE, G. GROSDIDIER, R. DOLBEAU, C. EISENBEIS, M. KRUSE, K. PETROV, C. TADONKI. *Automated Code Generation for Lattice Quantum Chromodynamics and beyond*, in "Journal of Physics: Conference Series", 2014, vol. 510, 11 p. , LPT-Orsay-13-142 [*DOI :* 10.1088/1742-6596/510/1/012005], https://hal.inria.fr/hal-00926513

[13] A. HUGO, A. GUERMOUCHE, P.-A. WACRENIER, R. NAMYST. *Composing multiple StarPU applications over heterogeneous machines: A supervised approach*, in "The International Journal of High Performance Computing Applications", February 2014, vol. 28, pp. 285 - 300 [*DOI :* 10.1177/1094342014527575], https://hal.inria.fr/hal-01101045

[14] E. JEANNOT, G. MERCIER, F. TESSIER. *Process Placement in Multicore Clusters: Algorithmic Issues and Practical Techniques*, in "IEEE Transactions on Parallel and Distributed Systems", April 2014, vol. 25, n° 4, pp. 993- 1002 [*DOI :* 10.1109/TPDS.2013.104], https://hal.inria.fr/hal-01109978

[15] E. SAILLARD, P. CARRIBAULT, D. BARTHOU. *PARCOACH: Combining static and dynamic validation of MPI collective communications*, in "International Journal of High Performance Computing Applications", 2014 [*DOI :* 10.1177/1094342014552204], https://hal.archives-ouvertes.fr/hal-01078762

International Conferences with Proceedings

[16] M. ALANIZ, S. NESMACHNOW, B. GOGLIN, S. ITURRIAGA, V. GIL COSTA, M. PRINTISTA. *MBSPDiscover: An Automatic Benchmark for MultiBSP Performance Analysis*, in "First HPCLATAM - CLCAR Joint Latin American High Performance Computing Conference", Valparaiso, Chile, Communications in Computer and Information Science (CCIS), Springer, October 2014, vol. 485, pp. 158-172, https://hal.inria.fr/hal-01062528

[17] D. BARTHOU, E. JEANNOT. *SPAGHETtI: Scheduling/Placement Approach for Task-Graphs on HETerogeneous archItecture*, in "Euro-Par", Lisboa, Portugal, LNCS, August 2014, vol. 8632, pp. 174 - 185 [*DOI :* 10.1007/978-3-319-09873-9_15], https://hal.archives-ouvertes.fr/hal-01100948

[18] A. DENIS. *pioman: a Generic Framework for Asynchronous Progression and Multithreaded Communications*, in "IEEE International Conference on Cluster Computing (IEEE Cluster)", Madrid, Spain, September 2014, https://hal.inria.fr/hal-01064652

[19] A. DENIS. *pioman: a pthread-based Multithreaded Communication Engine*, in "Euromicro International Conference on Parallel, Distributed and Network-based Processing", Turku, Finland, March 2015, https://hal.inria.fr/hal-01087775

[20] B. GOGLIN. *Managing the Topology of Heterogeneous Cluster Nodes with Hardware Locality (hwloc)*, in "International Conference on High Performance Computing & Simulation (HPCS 2014)", Bologna, Italy, IEEE, July 2014, https://hal.inria.fr/hal-00985096

[21] B. GOGLIN, J. HURSEY, J. M. SQUYRES. *netloc: Towards a Comprehensive View of the HPC System Topology*, in "Fifth International Workshop on Parallel Software Tools and Tool Infrastructures (PSTI 2014)", Minneapolis, United States, IEEE, September 2014, https://hal.inria.fr/hal-01010599

[22] C. HAINE, O. AUMAGE, P. ENGUERRAND, D. BARTHOU. *Exploring and Evaluating Array Layout Restructuration for SIMDization*, in "The 27th International Workshop on Languages and Compilers for Parallel Computing (LCPC 2014)", Hillsboro, United States, Intel Corporation, September 2014, https://hal.inria.fr/hal-01070467

[23] S. HENRY, A. DENIS, D. BARTHOU, M.-C. COUNILH, R. NAMYST. *Toward OpenCL Automatic Multi-Device Support*, in "Euro-Par 2014", Porto, Portugal, F. SILVA, I. DUTRA, V. S. COSTA (editors), Springer, August 2014, https://hal.inria.fr/hal-01005765

[24] A.-E. HUGO, A. GUERMOUCHE, P.-A. WACRENIER, R. NAMYST. *A runtime approach to dynamic resource allocation for sparse direct solvers*, in "2014 43rd International Conference on Parallel Processing", Minneapolis, United States, September 2014 [*DOI :* 10.1109/ICPP.2014.57], https://hal.inria.fr/hal-01101054

[25] X. LACOSTE, M. FAVERGE, P. RAMET, S. THIBAULT, G. BOSILCA. *Taking advantage of hybrid systems for sparse direct solvers via task-based runtimes*, in "HCW'2014 workshop of IPDPS", Phoenix, United States, IEEE, May 2014, https://hal.inria.fr/hal-00987094

[26] B. PUTIGNY, B. GOGLIN, D. BARTHOU. *A Benchmark-based Performance Model for Memory-bound HPC Applications*, in "International Conference on High Performance Computing & Simulation (HPCS 2014)", Bologna, Italy, IEEE, July 2014, https://hal.inria.fr/hal-00985598

[27] B. PUTIGNY, B. RUELLE, B. GOGLIN. *Analysis of MPI Shared-Memory Communication Performance from a Cache Coherence Perspective*, in "PDSEC - The 15th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing, held in conjunction with IPDPS", Phoenix, AZ, United States, IEEE, May 2014, https://hal.inria.fr/hal-00956307

[28] E. SAILLARD, P. CARRIBAULT, D. BARTHOU. *Static Validation of Barriers and Worksharing Constructs in OpenMP Applications*, in "IWOMP", Salvador, Brazil, September 2014, pp. 73 - 86 [*DOI :* 10.1007/978-3-319-11454-5_6], https://hal.archives-ouvertes.fr/hal-01078759

[29] M. SERGENT, S. ARCHIPOFF. *Modulariser les ordonnanceurs de tâches : une approche structurelle*, in "ComPAS 2014 : conférence en parallélisme, architecture et systèmes", Neuchâtel, Switzerland, P. FELBER, L. PHILIPPE, E. RIVIERE, A. TISSERAND (editors), April 2014, https://hal.inria.fr/hal-00978364

[30] L. STANISIC, S. THIBAULT, A. LEGRAND, B. VIDEAU, J.-F. MÉHAUT. *Modeling and Simulation of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures*, in "Euro-par - 20th International Conference on Parallel Processing", Porto, Portugal, Euro-Par 2014, LNCS 8632, Springer International Publishing Switzerland, August 2014, pp. 50-62, https://hal.inria.fr/hal-01011633

[31] G. VAUMOURIN, D. THOMAS, G. ALEXANDRE, D. BARTHOU. *Specific Read Only Data Management for Memory Hierarchy Optimization*, in "EWiLi 2014 - Workshop Embed With Linux", Lisboa, Portugal, J. BOUKHOBZA, J. P. DIGUET, P. FICHEUX, J. RUFINO, F. SINGHOFF (editors), Proceedings of the Embed With Linux 2014 Workshop, November 2014, vol. Vol-1291, Session 2, https://hal.archives-ouvertes.fr/hal-01090218

[32] P. VIROULEAU, P. BRUNET, F. BROQUEDIS, N. FURMENTO, S. THIBAULT, O. AUMAGE, T. GAUTIER. *Evaluation of OpenMP Dependent Tasks with the KASTORS Benchmark Suite*, in "IWOMP - 10th International Workshop on OpenMP", Salvador, Brazil, France, Springer, September 2014, pp. 16 - 29 [*DOI :* 10.1007/978-3-319-11454-5_2], https://hal.inria.fr/hal-01081974

Conferences without Proceedings

[33] E. JEANNOT, G. MERCIER, F. TESSIER. *Matching communication pattern with underlying hardware architecture*, in "6th European Conference on Computational Fluid Dynamics", Barcelona, Spain, July 2014, https://hal.inria.fr/hal-01087611

Scientific Books (or Scientific Book chapters)

[34] P. DE OLIVEIRA CASTRO, S. LOUISE, D. BARTHOU. *DSL Stream Programming on Multicore Architectures*, in "Programming multi-core and many-core computing systems", John Wiley and Sons, 2014, chapter 12, https://hal.archives-ouvertes.fr/hal-00952318

[35] T. HOEFLER, E. JEANNOT, G. MERCIER. *An Overview of Process Mapping Techniques and Algorithms in High-Performance Computing*, in "High Performance Computing on Complex Environments", E. JEANNOT, J. ŽILINSKAS (editors), Wiley, June 2014, pp. 75-94, https://hal.inria.fr/hal-00921626

[36] L. LOPEZ, J. ŽILINSKAS, A. COSTAN, R. G. CASCELLA, G. KECSKEMETI, E. JEANNOT, M. CANNATARO, L. RICCI, S. BENKNER, S. PETIT, V. SCARANO, J. GRACIA, S. HUNOLD, S. L. SCOTT, S. LANKES, C. LENGAUER, J. CARRETERO, J. BREITBART, M. ALEXANDER. *Euro-Par 2014: Parallel Processing Workshops, Part I*, Lecture Note In Computer Science, Springer, December 2014, vol. 8805, https://hal.inria.fr/hal-01110069

[37] L. LOPEZ, J. ŽILINSKAS, A. COSTAN, R. G. CASCELLA, G. KECSKEMETI, E. JEANNOT, M. CANNATARO, L. RICCI, S. BENKNER, S. PETIT, V. SCARANO, J. GRACIA, S. HUNOLD, S. L. SCOTT, S. LANKES, C. LENGAUER, J. CARRETERO, J. BREITBART, M. ALEXANDER. *Euro-Par 2014: Parallel Processing Workshops, Part II*, Lecture Note In Computer Science, Springer, December 2014, vol. 8806, https://hal.inria.fr/hal-01110071

Books or Proceedings Editing

[38] E. JEANNOT, J. ŽILINSKAS (editors). *High Performance Computing on Complex Environments*, Wiley, June 2014, 512 p. , https://hal.inria.fr/hal-00921619

Research Reports

[39] C. AUGONNET, O. AUMAGE, N. FURMENTO, S. THIBAULT, R. NAMYST. *StarPU-MPI: Task Programming over Clusters of Machines Enhanced with Accelerators*, May 2014, n° RR-8538, https://hal.inria.fr/hal-00992208

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

[41] L. STANISIC, S. THIBAULT, A. LEGRAND, B. VIDEAU, J.-F. MÉHAUT. *Modeling and Simulation of a Dynamic Task-Based Runtime System for Heterogeneous Multi-Core Architectures*, March 2014, n° RR-8509, https://hal.inria.fr/hal-00966862

[42] A. TATE, A. KAMIL, A. DUBEY, A. GRÖSSLINGER, B. CHAMBERLAIN, B. GOGLIN, C. EDWARDS, C. J. NEWBURN, D. PADUA, D. UNAT, E. JEANNOT, F. HANNIG, T. GYSI, H. LTAIEF, J. SEXTON, J. LABARTA, J. SHALF, K. FÜRLINGER, K. O'BRIEN, L. LINARDAKIS, M. BESTA, M.-C. SAWLEY, M. ABRAHAM, M. BIANCO, M. PERICÀS, N. MARUYAMA, P. H. J. KELLY, P. MESSMER, R. B. ROSS, R. CLEDAT, S. MATSUOKA, T. SCHULTHESS, T. HOEFLER, V. J. LEUNG. *Programming Abstractions for Data Locality*, PADAL Workshop 2014, April 28–29, Swiss National Supercomputing Center (CSCS), Lugano, Switzerland, November 2014, 54 p. , https://hal.inria.fr/hal-01083080

Scientific Popularization

[43] E. AGULLO, O. AUMAGE, M. FAVERGE, N. FURMENTO, F. PRUVOST, M. SERGENT, S. THIBAULT. *Overview of Distributed Linear Algebra on Hybrid Nodes over the StarPU Runtime*, February 2014, SIAM Conference on Parallel Processing for Scientific Computing, https://hal.inria.fr/hal-00978602

References in notes

[44] 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

[45] 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

[46] 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