C2S@Exa
Computer and Computational Sciences at Exascale

Ph.D project proposal

High order discontinuous Galerkin solvers over runtimes for harnessing manycore systems

---

MOAIS project-team	Inria Grenoble - Rhône-Alpes research center
NACHOS project-team	Inria Sophia Antipolis - Méditerranée research center
RUNTIME project-team	Inria Bordeaux - Sud-Ouest research center
TONUS project-team	Inria Nancy - Grand Est research center

Contacts: Thierry Gautier, Stéphane Lanteri, Philippe Helluy and Raymond Namyst
E-mails: thierry.gautier@imag.fr, Stephane.Lanteri@inria.fr, helluy@math.unistra.fr, Raymond.Namyst@inria.fr

Keywords: high order method, discontinuous Galerkin, GPU, Xeon Phi, OpenMP, OpenCL, MPI, StarPU, Xkaapi.

The discontinuous Galerkin (DG) method was introduced in 1973 by Reed and Hill to solve the neutron transport equation. From this time to the 90's a survey of the DG methods would likely fit into one page. In the meantime, the finite volume approach has been widely adopted by computational fluid dynamics scientists and has now nearly supplanted classical finite difference and finite element methods in solving problems of non-linear convection. The success of the finite volume method is due to its ability to capture discontinuous solutions which may occur when solving non-linear equations or more simply, when convecting discontinuous initial data in the linear case. Let us first remark that DG methods share with finite volumes this property since a first order finite volume scheme can be viewed as a 0th order DG scheme. However a DG method may be also considered as a finite element one where the continuity constraint at an element interface is released. While it keeps almost all the advantages of the finite element method (large spectrum of applications, complex geometries, etc.), the DG method has other nice properties which explain the renewed interest it gains in various domains in scientific computing nowadays. In short, it is a compact support, high order finite element method (it relies on a high order interpolation of the problem unknowns within each cell of the mesh) that, from the algorithmic point of view, combines sparse and dense linear algebra operations. In particular, the DG method is naturally adapted to parallel computing and its compact nature is in favor of high computation to communication ratio especially when the interpolation order is increased.

The NACHOS project-team at Inria Sophia Antipolis - Méditerranée and the TONUS project-team at Inria Nancy - Grand Est are studying and developing high order DG methods for solving linear and non-linear systems of conservation laws and in particular those modeling electromagnetic wave propagation problems in various physical contexts [DLL14]-[LVD+14]-[CH13]-[CHJ13]. These teams have also conducted preliminary studies aiming at the algorithmic adaptation of these DG methods for exploiting heterogeneous parallel computing systems organized around multicore CPU chips and accelerators such as GPU and Intel Xeon Phi processors. The corresponding software developments are based on standard programming environments including MPI, OpenMP and OpenCL.

Traditional processors have reached architectural limits which heterogeneous multicore designs and hardware specialization (e.g. coprocessors, accelerators, etc.) intend to address. However, exploiting such massively parallel systems introduces numerous challenging issues at all levels, ranging from programming models and compilers to the design of scalable hardware solutions. The design of efficient runtime systems for these architectures is a critical issue. Indeed, due to the recent evolution of parallel architectures toward hierarchical, heterogeneous multicore machines, many research efforts have recently been devoted to the design of runtime systems able to provide programmers with portable techniques and tools to exploit such a complex hardware. Several mature programming environments have been adapted to better cope with the memory hierarchy exhibited by multicore machines (e.g. OpenMP, TBB), while new programming environments have been developed to deal with accelerator-based configurations (e.g. CUDA, OpenCL). Because of the lack of consensus regarding the definition of a standard programming model for such machines, an increasing number of HPC developers are manually combining multiple programming environments (e.g. MPI and OpenMP or MPI and OpenCL) to effectively make use of every underlying processing unit. Using hybrid parallelism to program heterogeneous machines is actually a clear trend, and it will probably become unavoidable in the future, through indirect hybridization.

The StarPU environment developed in the Runtime project-team at Inria Bordeaux - Sud-Ouest typically makes it much easier for high performance libraries or compiler environments to exploit heterogeneous multicore machines possibly equipped with accelerators: rather than handling low-level issues, programmers may concentrate on algorithmic concerns. Portability is obtained by the means of a unified abstraction of the machine. StarPU [ATNW11], offers a unified offloadable task abstraction named *codelet*. Rather than rewriting the entire code, programmers can encapsulate existing functions within codelets. In case a codelet may run on heterogeneous architectures, it is possible to specify one function for each architecture (e.g. one function for a GPU and one function for a traditional CPU). StarPU takes care to schedule and execute those codelets as efficiently as possible over the entire machine.
To bridge the gap between dynamic runtime systems and standard programming interfaces, an implementation of OpenCL was developped on top of StarPU. This implementation, named SOCL, enables applications to dynamically dispatch computation kernels over processing devices so as to maximize their utilization. OpenCL applications can incrementally make use of light extensions to automatically schedule kernels in a controlled manner on multi-device architectures.

The Xkaapi environment developed in the Moais project-team at Inria Grenoble - Rhône-Alpes is a runtime system for data-flow task programming on multi-CPU and multi-GPU architectures, which supports a data-flow task model and a locality-aware work stealing scheduler. XKaapi [GFJ+13] enables task multi-implementation on CPU or GPU and multi-level parallelism with different grain sizes. A XKaapi program is a sequential code complemented with annotations or runtime calls to create tasks. Parallelism is explicit, while the detection of synchronizations is implicit: the dependencies between tasks and the memory transfers are automatically managed by the runtime. The runtime creates a system thread for each computational resource to be used.

In the recent years, the Nachos and Tonus project-teams have developed their own software suites implementing high order DG methods formulated on unstructured meshes for simulating various applications involving the propagation of electromagnetic waves in interaction with complex devices and media. These two software suites share the fact that the basis mathematical model that has to be solved is the system of Maxwell equations in three dimensions (four dimensions when time is added). Different parallelissation strategies have been adopted by the two software (MPI+OpenCL and MPI+OpenMP). The current common objective of these two teams is to exploit heterogeneous parallel computing platforms in a flexible way (in particular with respect to the virtualization of the underlying architecture characteristics) while maximizing the floating-point performance and scalability on manycore systems. In this context, the general goal of the present *inter-disciplinary* Ph.D project is to exploit the capabilities of the StarPU and XKaapi runtime systems for harnessing modern manycore systems in the context of very large-scale simulations of complex electromagnetic wave propagation problems using DG-based numerical methodologies. This study will be conducted in close collaboration with researchers from the Moais, Nachos, Runtime and Tonus project-teams. In practice, this project will be comprise several facets aiming at:

- Tuning performance of numerical algorithms (DG kernels in the present case) to the hardware characteristics of maycore nodes (Xeon Phi and GPU cards) through the specification and development of a dedicated auto-tuning tool.
- Proposing and studying possible extensions to the OpenCL and OpenMP standards for improving the flexibility in combining MIMD and SIMD parallel programming paradigms.
- Leveraging the performances of massively parallel heterogenous systems through the appropriate, task graph-based, interfacing of the DG-based simulation software with the above-mentioned runtime systems.
- Demonstrating the benefits of the methodological achievements through the realization of large-scale numerical simulations of complex electromagnetic wave propagation problems.

For this Ph.D project, applications are invited from recently graduated students in scientific computing with a major in computer science. Knowledge of the principles of high performance computing is required; a first practical experience in this domain would be an asset. Strong software programming ability in a Linux environment is mandatory.

References

[ATNW11] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier. StarPU: a unified platform for task scheduling on heterogeneous multicore architectures. *Concur. and Comput: Pract. Exp.*, 23(2):187–198, 2011.

[CH13] A. Crestetto and P. Helluy. Resolution of the Vlasov-Maxwell system by PIC discontinuous Galerkin method on GPU with OpenCL. In *CEMRACS'11: Multiscale Coupling of Complex Models in Scientific Computing*, volume 38, pages 257–274, France, 2013.

[CHJ13] A. Crestetto, P. Helluy, and J. Jung. Numerical resolution of conservation laws with OpenCL. *ESAIM: Proceedings*, 40:51–62, 2013.

[DLL14] C. Durochat, S. Lanteri, and R. Léger. A non-conforming multi-element DGTD method for the simulation of human exposure to electromagnetic waves. *Int. J. Numer. Model., Electron. Netw. Devices Fields*, 2014. To appear.

[GFJ+13] T. Gautier, L. Ferreira, V. Joao, N. Maillard, and B. Raffin. XKaapi: a runtime system for data-flow task programming on heterogeneous architectures. In *27th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, Boston, Massachusetts, USA, 2013.

[LVD+14] R. Léger, J. Viquerat, C. Durochat, C. Scheid, and S. Lanteri. A parallel non-confoming multi-element DGTD method for the simulation of electromagnetic wave interaction with metallic nanoparticles. *J. Comp. Appl. Math.*, 2014. To appear.