PhD subject :
Innovative programming models for the development of scientific applications on exascale architectures.

This thesis subject is suited for students with a master's degree in computer science or scientific computing with a strong high-performance computing component.

PhD supervisor : Mathieu Lobet
PhD director : Edouard Audit
Laboratory : Maison de la Simulation, CEA Saclay (CEA Saclay, 91191 Gif-sur-Yvette Cedex)
Contact : Mathieu Lobet (mathieu.lobet@cea.fr)

Context

Supercomputers are becoming more and more heterogeneous. Computing nodes consist of several CPU sockets coupled to several accelerators, most often GPGPUs (General Purpose Graphics Processing Units). GPU accelerators were first introduced in super-computers in the early 2010s and now accounts for almost 50% of the 100 most powerful supercomputers in the world. This trend has become more pronounced in recent years as GPU technologies for computing and artificial intelligence have become increasingly sophisticated, both in terms of hardware (performance, energy consumption, on-board memory) and software (maturity of programming models, development of libraries, etc.). While NVIDIA has long dominated the High-Performance Computing (HPC) GPU market, other players such as AMD and Intel are now diversifying the available technologies. These new architectures are notably favoured by the race to Exascale, which aims to achieve computing power of 10^{18} arithmetic operations per second.

The democratisation of GPUs, especially for scientific computing, has been accompanied by the emergence of new programming models. For simplicity, we distinguish between three major categories: relatively low-level languages provided by manufacturers or supported by some of them (CUDA, OpenCL, etc.), directive-based models (OpenACC, OpenMP) and high-level abstraction models (Kokkos, RAJA, Alpaka, SYCL, etc.). These models can be compared using several criteria such as performance (relative to the peak power of the machine), portability (ability of the code to adapt to multiple architectures with minimal effort), model maturity, ease of learning, ease of production, maintainability, modularity and more. With the advent of multiple GPU architectures, portability has become a priority issue in the choice of a model for modernising or writing code. It allows a single implementation to run on many architectures, both CPU and GPU, but also ensure compatibility with future technologies. Developers, often scientists, thus minimise rewriting efforts in favour of better productivity.

High-level abstraction programming models allow, as their name suggests, the abstraction of data structures, memory management and parallelism on the targeted architectures. To do this, they use low-level languages in the background (CUDA backends, for example) and the power of the latest C++ standards (especially C++17) in the foreground. They allow us to propose portable and high-performance implementations in a few lines of code. Some of them have been developed for almost 10 years but are still relatively little used in the scientific community on production codes. Nevertheless, many of them are now reaching maturity. They are pushed and supported by most manufacturers.

Goals

The primary objective of this thesis is to explore these new programming models in the context of developing scientific applications for future exascale architectures. The model that interests us most is SYCL developed by the Khronos Group. It has the advantage of being adopted today by many manufacturers (AMD, Intel, XILINX and others) and of being supported by many parallel software libraries. It is one of the main models proposed by Intel within its DPC++ implementation for programming its future GPUs (Ponte Vecchio) but also its future FPGA boards (Agilex).

The goal of the candidate will be to explore the SYCL implementation of several scientific computing kernels. One of the first kernels envisaged corresponds to the "Particle-In-Cell" method used for plasma simulation in several codes at CEA. Other kernels from codes developed or supported at the Maison de la Simulation may be explored (astrophysics, molecular dynamics, fluid dynamics). SYCL and the implemented kernels will be tested on systems under development for the European Exascale. A partnership with SiPearl and Intel in the framework of the EoCoE-III project will allow the solution to be tested on heterogeneous nodes coupling the ARM Rhea processor and Ponte Vecchio GPUs. Future architectures from other manufacturers will also be tested progressively with the arrival of future European and French machines.

The second objective of the thesis will be to explore the programming of Field Programmable Gate Array (FPGA) accelerators in a scientific context. With the aim of performance and energy efficiency, the FPGA architecture could become a relevant accelerator technology alongside CPUs and GPUs and perhaps equip future post-exascale machines. Many manufacturers are pushing in this direction, even if we are still in the experimental phase in an HPC context. Intel is again a front-line player and has recently launched a new accelerator card (Agilex) that can be programmed using the SYCL-based DPC++ programming model for industrial and scientific applications. Through a partnership between the CEA and Intel, the candidate will have access to Agilex boards in order to study the performance of scientific kernels on this type of architecture.

For each objective, the model will be evaluated according to various criteria to be defined more precisely during the thesis, such as real portability, performance compared to other programming models, ease of use, maturity, complexity of the resulting code, etc. Depending on his or her affinity and progress, the successful candidate will be able to choose the kernels or architectures on which to strengthen his or her studies. However, the aim is not to develop a new scientific code.

This thesis is on a topic that is suitable for a candidate who wishes to conduct a research project in computer science and then pursue a career in the research area as well as in industry.

Skills

- Significant knowledge of software programming in modern C++
- Knowledge of parallel programming (GPU accelerator programming is a plus)
- Interest in carrying research activities
- Good ability to work in English (written and spoken)
- Interest in applied mathematics and numerical simulation is a plus