Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

SHAPE Project AXESSIM - CINES Partnership: HPC for connected Objects

C. Girard[a], B. Weber[a], B. Cirou[b*], V. Cameo Ponz[b]

[a]*AxesSim, 1 rue Jean Sapidus, 67400 Illkirch-Graffenstaden, France, www.axessim.fr, bruno.weber@axessim.fr*
[b]*Centre Informatique National de l'Enseignement Supérieur (CINES), 950 rue St Priest, 34097 Montpellier, France*

Abstract

AxesSim has developed an electromagnetic simulation tool named TETA, which is optimized for harnessing several GPUs in parallel. It is based on the Discontinuous Galerkin method in the Time Domain (DGTD). This method allows to handle meshes of complex geometries and can model designs with high precision such as a full human body with its organs, clothes and surrounding environment. The targeted architecture is a CRAY XC50 (PizDaint) which is a supercomputer with Intel Broadwell CPUs and NVidia P100 GPUs. Both CPU and GPU are addressed via the OpenCL library. MPI is used for communication between accelerators and hosts.

Introduction

AxesSim, in collaboration with THALES and ONERA as part of the HOROCH project [1],[2], is conceiving a test bench for the electromagnetic design of connected objects. AxesSim is in charge of the simulation software tools attached to the test bench developed by THALES. The objective is to propose to vendors of connected objects measures and simulation tools for optimising the design. This requires the simulation of electromagnetic waves produced by small antennas and their interaction with biological tissues. Such simulations are complex and expensive. They require to take into account complex geometries. In order to answer to the client in a reasonable time, it is necessary to use highly optimized software, parallel computations and GPUs.

Through the SHAPE call for project launched by PRACE, AxesSim was granted 2000 node-hours on the Swiss supercomputer called PizDaint [3]. This choice allows the SME Axessim to access the latest GPU technologies on the most powerfull (25 PFlops) supercomputer in Europe and the third one in the world according to the TOP500 ranking [4] . PizDaint is a hybrid Cray XC40/XC50 system with 5320 nodes equipped with one Intel® Xeon® E5-2690 v3 @ 2.6GHz (12 cores, 64GB RAM) and NVIDIA® Tesla® P100 16GB and 1813 nodes equipped with two Intel® Xeon® E5-2695 v4 @ 2.1GHz (18 cores, 128GB RAM). The nodes are connected by the Aries proprietary interconnect from Cray, with a dragonfly network topology. We initially wanted to use these compute hours in order to perform simulations of connected objects (BLE antenna) close to the human body [2]. However, such a simulation requires about 2000 node-hours. Therefore, we finally only performed scalability tests on such a configuration, up to hundreds of GPUs in parallel.

* Corresponding author. *E-mail address*: cirou@cines.fr

Software Stack Compilation, Validation and Usage Through Jobs

The TETA [5], [6] simulation tool is developed by AxesSim in C++ and use the portable and standardized libraries OpenCL and MPI to express its parallelism. There is one OpenCL kernel for each terms of the Galerkin formula (mass, volume and stream) and the domain is partitioned across the MPI ranks. It has several dependencies: CMake, OpenCL, MPI, HDF5, Boost, GLM, Metis and Amelet-HDF, the open specification over HDF5 file format for recording and recovering computer data of electromagnetic simulation. Among these dependencies, CMake, OpenCL, MPI and HDF5 are installed on the supercomputer's system and have been used through the available modules. Most of them are loaded at login time via the `PrgEnv-*` modules. For this project, we had to use `PrgEnv-gnu` (gcc) because the compilation of Boost using `PrgEnv-cray` could not be completed during the compute hours' allocation time. Fortunately, the performance of the solver was not affected by this choice. To access GPUs, it is necessary to load the `daint-gpu` module. To activate OpenCL, it is necessary to load the `cudatoolkit` module.

Compiling the other modules and the TETA solver required some environment variables:

- `CRAY_CPU_TARGET=x86_64`: to indicate the targeted host processors;
- `CRAYPE_LINK_TYPE=dynamic`: to dynamically link the dependencies.

We also used the following compile options to fully optimise the resulting objects:

- General options: `-O3 -g -fPIC -std=c++11 -finline-functions`;
- Arch. dependent ones: `-march=core-avx2 -mtune=core-avx2 -mcmodel=medium`.

All the dependencies were built in shared linkage and release mode. The Boost 1.65.1 library has been built with the `gcc` toolset. To build the TETA solver with MPI, we used the MPICH library whose path is defined by the `MPICH_DIR` environment variable. We then used the associated pkgconfig:

```
$ export PKG_CONFIG_PATH=$MPICH_DIR/lib/pkgconfig:$PKG_CONFIG_PATH
```

And the following definitions:

- `MPI_CXX_COMPILER=CC`: default compiler loaded by the GNU environment;
- `MPI_CXX_LIBRARIES=$(pkg-config --libs-only-l mpich)`: MPICH libraries;
- `MPI_CXX_INCLUDE_PATH=$MPICH_DIR/include`: MPICH headers.

Finally, we managed to build the TETA solver and all its dependencies successfully: all the `ctest`s passed, except those using OpenCL devices due to the simultaneous usage of an older OpenCL library by Boost. In order to test the OpenCL implementation on the supercomputer, we used a slurm script to run a small simulation on several GPUs.

Here is an example of such a slurm script launched on two GPUs:

```
#!/bin/bash
#SBATCH -p normal
#SBATCH -n 2
#SBATCH -N 2
#SBATCH --tasks-per-node 1
#SBATCH --constraint=gpu
#SBATCH -t 01:00:00

module swap PrgEnv-cray PrgEnv-gnu
export CRAY_CPU_TARGET=x86_64
export CRAYPE_LINK_TYPE=dynamic
module load daint-gpu
module load cudatoolkit

export HDF5_USE_FILE_LOCKING=FALSE
export MPICH_MAX_THREAD_SAFETY=multiple


export LIBRARY_PATH=$MPICH_DIR/lib:path/to/our/libs:$LIBRARY_PATH
export LD_LIBRARY_PATH=$MPICH_DIR/lib:path/to/our/libs:$LD_LIBRARY_PATH

srun path/to/teta-solver [command line options]
```

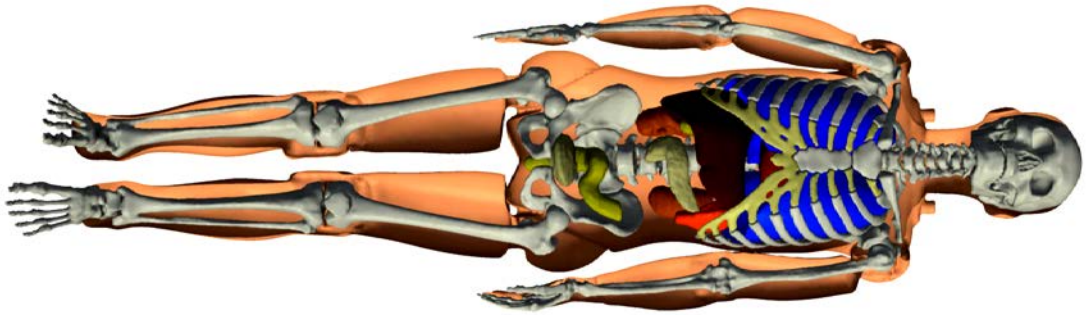

Figure 1: *Full human body model meshed with 6 million tetrahedrons.*

As described in the above script, first we force HDF5 to not lock the files thanks to `HDF5_USE_FILE_LOCKING` [7] and then we specify that every threads in the simulation will do MPI communications by setting `MPICH_MAX_THREAD_SAFETY` [8] environment variables. Otherwise, opening a HDF5 file in write mode is impossible, as well as the use we make of MPICH: allow communications between all nodes by multiple threads. This model of script worked for all our scalability tests. The only parameters we have varied are:

- `n`: the number of tasks (GPUs);
- `N`: the number of nodes (=`n`, in our case);
- `t`: the estimated simulation time.

Unable to launch a complete simulation, we thus carried out tests of scalability of our TETA solver in order to evaluate its performances during a parallel execution. We performed these tests on a complete human body configuration, with skeleton and organs.

Test Case and First Results

The test case we used to evaluate the scalability of the TETA solver involves a complete human body, with skeleton and organs, and a Bluetooth antenna. The numerical human model is an X-ray scan made by the IRCAD of Strasbourg [9] from the PBU-60 model produced by Kyoto Kagaku Inc. This model comprises 12 organs and is composed of 6 million tetrahedrons (Figure 1). Our solver is designed to handle hexahedral meshes, so we cut those 6 million tetrahedrons into 24 million hexahedrons. The chosen antenna is of LTCC technology commonly used for connected objects. The scaling factor between the smallest and largest mesh elements is of the order of 800.

Such a simulation requires 11 days of intense parallel computing on 8 NVidia GeForce GTX 1080 Ti GPUs in order to obtain 10ns of physical time. By performing the calculation, this corresponds to more than 2000 node-hours. The computation power of the P100 being equivalent to that of the GTX 1080 Ti, we concluded that we did not have enough node-hours to perform a full simulation. The occupied RAM varies according to the chosen spatial interpolation order (which gives the number of approximation nodes in each mesh element): an interpolation at order 1 requires 40GB. 100GB are required at order 2 and 200GB at order 3 with the same mesh.

We performed the scalability tests for an interpolation order varying from 1 to 3 and on a growing number of GPUs involved. The results are shown in Figure 2 [10]. The reference time used for each interpolation order is the one obtained for the minimum number of GPUs needed to perform the simulation, according to the required RAM.

#GPUs	Order 1	Order 2	Order 3
4	1 (*)	-	-
8	0.98	1 (*)	-
16	0.96	0.98	1 (*)
32	0.91	0.94	0.97
64	0.84	0.89	0.92
128	0.75	-	-
256	0.61	-	-

Figure 2: *Solver's efficiency on a growing number of GPUs.*
(*) Simulation giving the reference time for the current column.

We find that the efficiency on a few GPUs is very good (close to 1). These results also show, for example at order 1, that on 256 GPUs, the efficiency of the solver is of 0.61. This means that, on 256 GPUs, the machine time required to perform the simulation mentioned above is about 3500 node-hours. But the restitution time of the results (or human time) is about 13 hours. These results are logical since, the more we divide the problem, the more MPI exchanges between GPUs have to be processed.

Before that, we already estimated, thanks to scaling results obtained on other computers, that the minimum GPU memory load in order to remain really efficient (above 0.9) was about 1.5GB per GPU. The results obtained on PizDaint confirm this assertion once again. In addition, before now we had never exceeded the use of 16 GPUs in parallel, and thanks to these compute hours, we could test our solver on a large amount of GPUs.

Conclusion

The calculation of the radiation of electromagnetic waves near or in the human body requires sufficient precision in the simulated model. This need for precision induces simulation sizes of several tens, or even hundreds, of GB, and simulation times of several hours, or even days, depending on the power of the addressed GPUs.

For industrial application purposes, it is necessary to reduce this calculation time. Thanks to the calculation hours granted by PRACE, we have demonstrated by these scalability tests that the TETA solver is able to meet the demands of industry. Our solver is able to parallelize calculations on a large number of GPUs while remaining sufficiently efficient.

Recent work on local time step schemes would theoretically reduce this computing time to a few hours or even minutes. However, for the moment, this is only a conjecture. In addition, such good scalability results of the TETA solver with such a method are not yet assured.

References

[1] AxesSim. HOROCH project. 2015-2018. https://www.axessim.fr/horoch-project-one-year-old-first-highly-promising-results.html.
[2] A. Guéna *et al.* Electromagnetic analysis: Radiated emission of IoT applications close to an anthropomorphic mannequin. *EMC EUROPE - 2017 International Symposium on Electromagnetic Compatibility*, Angers, 2017, pp. 1-5. doi: 10.1109/EMCEurope.2017.8094767
[3] Swiss National Supercomputing Centre, PizDaint, https://www.cscs.ch/computers/piz-daint/.
[4] TOP500. https://www.top500.org.
[5] B. Weber, P. Helluy, T. Strub. Optimisation d'un algorithme Galerkin Discontinu en OpenCL appliqué à la simulation en électromagnétisme. *NUMELEC 2017 - 9th European Conference on Numerical Methods in Electromagnetics*, Nov 2017, Paris, France. pp.1-2. 〈https://numelec2017.sciencesconf.org/〉. 〈hal-01666352〉
[6] AxesSim. TETA solver. https://www.axessim.fr/discontinuous-galerkin-optimizations-numelec-2017.html.

[7] HDF5_USE_FILE_LOCKING: https://github.com/ALPSCore/ALPSCore/issues/348.
[8] MPICH_MAX_THREAD_SAFETY: PizDaint support.
[9] IRCAD: http://www.les-haras-biocluster.com.
[10] N. Muot *et al.* Implémentation massivement parallèle d'une méthode Galerkin Discontinue appliquée à l'électromagnétisme en domaine temporel – Application aux interactions onde / corps humain. *EMC EUROPE - 2018 International Symposium on Electromagnetic Compatibility*, 2018 (validated by the peer, published soon).

Acknowledgements

This work was financially supported by the PRACE project funded in part by the EU's Horizon 2020 Research and Innovation programme (2014-2020) under grant agreement 730913.