Haute école d'ingénierie et d'architecture Fribourg
Hochschule für Technik und Architektur Freiburg

GPU optimization - Celeritas Specification

Computer science and Communication System (ISC), 2022-2023

Student	Simon Barras
Supervisors	Prof. Frédéric Bapst - HEIA-FR Prof. Jean Hennebert - HEIA-FR
Customer	Lawrence Berkeley National Laboratory (LBNL) Paolo Calafiura Julien Esseiva
Expert	Dr. Baptiste Wicht

v1.3
Berkeley, CA, USA, August 10, 2023

Version History

Version	Changes	Date
0.1	Adding the context 1 and the goals 2 chapter. These paragraphs haven't been reread because that is a WIP version	12.06.2023
0.2	Adding the activites 3 and the planning 4 chapter. These paragraphs haven't been reread because that is a WIP version	13.06.2023
1.0	First version of the specification	13.06.2023
1.1	Comments from Mr. Bapst and Mr. Hennebert were taken into account. Adding an image describing the ATLAS experiments in the context chapter 1 and an image of the schedule 4. Adding a secondary objective concerning different GPU architectures 2.	15.06.2023
1.2	Comments from Mr. Esseiva were taken into account. Correction and clarification of certain points in the context chapter 1 to bring it into line with the current status of the Celeritas project. Rewording the main objective 2 so as not to base the project's success on whether or not performance has improved.	15.06.2023
1.3	Comments from Mr. Calafiura were taken into account. Correction and clarification of certain points in the context chapter 1 to bring it into line with the current status of the Celeritas project and the physics behind it. Rewording the main objective 2 to better explain the difference between the measurement of the success of the wish of the team and the thesis.	19.06.2023

Contents

1 Context

This Bachelor thesis is done by Simon Barras and supervised by Frederic Bapst and Jean Hennebert. The customer Paolo Calafiura is a physicist and computer scientist at the Lawrence Berkeley National Laboratory (LBNL). To do this project, I am moving to Berkeley, California, for ten weeks. The goal is to improve the performance of the project Celeritas, which is a particle physics simulation software accelerated by GPUs.

1.1 Celeritas

The project Celeritas [1] is a particle physics simulation software that can be integrated with Geant4 [2] as a plugin library. It can also be used as a standalone application, with limited functionality for now. Geant4 is a toolkit for the simulation of the path of particles through matter. It is used for many detector simulation applications in particle physics, medical physics, and beyond. The goal of the Celeritas project is to develop GPU-accelerated versions of the most computationally intensive kernels of Geant4.

1.2 Physics simulation

Actually, the two main customers are CMS and ATLAS, both made their experiments at the European Organization for Nuclear Research (CERN) with the Large Hadron Collider (LHC) and run their simulation with Geant4. They are both using Geant4 and they didn't have committed to using Celeritas beyond an initial evaluation. Detector simulation is used to validate and calibrate the algorithms used to estimate the properties of the primary particles from the observed detector data. The main goal of the thesis will be to optimize a GPU-accelerated version of the Prince Dormand algorithm [3], a Runge-Kutta solver [4] for the differential equations governing the trajectory of particles in a non-uniform magnetic field. This work will improve the project Celeritas [1] which may replace Geant4 [2] in the future.

The A Toroidal LHC ApparatuS (ATLAS) experiment tracks the path of particles in the detector and produces coordinates points where particles traverse the sensors. Figure 1.1 represents this experiment.

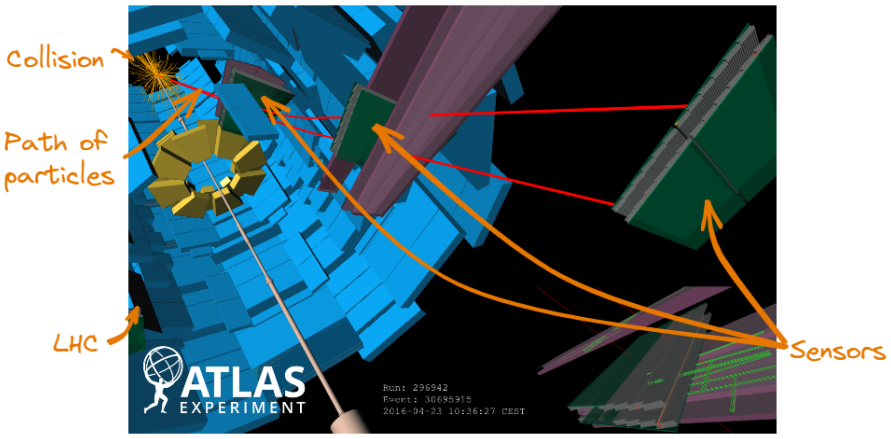

Figure 1.1: ATLAS experiment at CERN [5]

1.3 Lawrence Berkeley National Laboratory

The Lawrence Berkeley National Laboratory (LBNL) is a national laboratory in Berkeley, California. It is managed and operated by the University of California for the Department of Energy (DOE). The lab is situated in the hills of Berkeley and it is composed of many buildings and has a beautiful view of the San Francisco Bay (see Fig.1.2).

Figure 1.2: Lawrence Berkeley National Laboratory

The Physics and X-Ray Science Group, where the project is done, is situated in building 50f and the LBL ATLAS group, who are the customers of this project, are located in building 50.

1.4 The Need

Celeritas is already accelerated by GPUs, however, the team wants to improve the performance to be able to reduce the time of a simulation. In the current version of the code, each particle track is processed in parallel by one GPU thread, with no collaboration between threads. GPU profiling of the code shows that execution time is dominated by two kernels. The first one is handled by the interaction with the detector geometry to know where, in 3D space, the particle is situated and during the profiling, the library used is vecGeom [6]. The second kernel, which will be the focus of this Bachelor thesis project, is the computation of a differential equation using Dormand-Prince [3].

2 | Goals

The list of goals enumerates the different deliverables of the project and the main ones will be found in the list of milestones. The goals are described in the following sections using the SMART method [7].

To achieve the goal of this project, the profiling should show a performance improvement. However, the thesis can be a success even if the project doesn't meet the improvement. This is because we don't yet know whether thread synchronization will be more time-consuming than the original version. In addition, the kernel launch in Celeritas may have to be changed, and this could take up a considerable amount of optimization time. For the Bachelor thesis, it is sufficient to have a proof of concept that demonstrates that the enhancement is effective and deserves to be integrated. To measure the changes done during the internship, the profiler must be run before and after each step of the project.

2.1 Mandatory requirements

These requirements must be delivered at the end of the project.

2.1.1 Learn GPU programming

Before starting to work on the project, some things need to be learned and the goal here is to learn a new way of programming. To conclude this goal, no code will be produced except for exercises, but the important notions of GPU programming with CUDA will be synthesized using cheat sheets. To take advantage of the delay between the beginning of the Bachelor thesis and the beginning of the LBNL internship, this step will be done during this time.

2.1.2 Understand the project

To be able to improve the performance of the code, the first step is to understand the project and it's always better to understand the background: why it is needed, who will use it and which paradigm and tools are used.

To measure the performance gained, it is necessary to know where the project is at each step. To take a snapshot of the performance, a profiler can be run and this includes that we can compile and launch the project. This step will be done at the beginning of the project on-site.

2.1.3 Improve the performance

The main goal of the project is to improve the performance of the implementation of Dormand-Prince method [3]. This last mandatory requirement is the core of the thesis and the most important part of the project and it will require the knowledge gained in the first two steps to improve the performance.

To conclude this step, the code must compile, pass the unit test, and a profiler must be run to show the difference between the new and the legacy implementation of the DormandPrince method. To achieve this goal, the profiles must show an improvement, but this could meet some difficulties to be realized and integrated into the project. In all

cases, the failure of this goal doesn't mean the failure of the thesis if the documentation is correctly done and explains the results obtained and how is it possible or not to continue to this path. This step will be done after the first two steps and it will take the whole time left.

2.2 Optional requirements

These requirements are not mandatory but they could be a good addition to the project.

2.2.1 Portable performance

The purpose of Celeritas is to be run on all kinds of GPU and even on machines with just a CPU. During the optimization, the improvement will be checked on the Perlmutter [8] which uses Nvidia A100 with the architecture Ampere [9] and some improvement can be only effective to this kind of GPU. This goal is here to check if the improvement has a positive effect on other architectures and if it is not, to find a way to do that. To begin this step, the main goal needs to be finished.

2.2.2 Another performance improvement

If the performance of the Runge-Kutta method [4] is improved, another optimization can be done. This part goal will be discussed further in the project with the supervisors and the customers and it will be managed like the last mandatory goal. This step can be done multiple times if there is enough time.

3 Activities

Activities are all the tasks that need to be done to complete the project. These tasks come directly from the goals and the planning is based on the activities listed above.

All the tasks are not mentioning the documentation, but it's implicit that all the tasks will be reported in the final report.

3.1 Learn GPU programming

This step will launch the project and it can be without any knowledge of the project Celeritas, the objective is to learn the fundamentals of GPU programming.

3.1.1 Learn CUDA

First of all, the programming oriented GPU and the language CUDA are totally new, there is no course about that in the Bachelor program.

Lawrence Berkeley National Laboratory provides a course about CUDA [10] and some exercises [11] to learn the language.

3.1.2 Write cheat sheet

For each course, a cheat sheet will be produced to summarize the important notions and provide quick access to the information during the realization of the project. Not every course will produce a new one, an old cheat sheet can be improved.

3.2 Understand the project

The first step to do in the laboratory is to dive into the project to understand them. The discussion with the team and the customer will be important to catch the important points.

3.2.1 Compile the code

Celeritas is made to be run on HPC and it comes with a new tool to build it. This is important to understand the basis of Module [12] and Spack [13] to be able to compile the code. This task is the first one to do in the laboratory.

3.2.2 Launch job on perlmutter

Perlmutter [8] is the new HPC of NERSC. To launch a job on this kind of machine, it's not a simple command as on a personal computer, this requires a script that includes some parameters and the command to launch the code. This step must be done to have an example that can be adapted to Celeritas.

3.2.3 Profile the code

In order to measure the improvement, the code must be profiled on Perlmutter to know the performance's reference. This step includes running the application with the relevant input and having some basic knowledge of the Nvidia profiling tool. It is very

important to know the limit of the code and announce some objectives to reach before starting coding. To close this second goal of the project, a profile must be recorded and analyzed.

3.2.4 Background the project

Aside from the other task, this one will be done to understand the background of the project. It is not mandatory to drive this Bachelor thesis to success but it will help to understand what the other members of the team are doing and to provide a better view of the project in the final report.

3.3 Improve the performance

This is the goal of this Bachelor's thesis and to do that, several steps must be done to reach the objective.

3.3.1 Understand the code

First of all, it's important to understand how the code is working in order to not reinvent the wheel with a function that already exists. The objective is also to follow the guideline of the project to make easier understanding and maintainability for the team.

3.3.2 Understand Runge-Kutta

As the optimization is on the Runge-Kutta method [4], it's important to understand how it is working and how it's implemented in the code. Some analysis must be done to know where it's possible to improve the performance and how.

3.3.3 Implement the optimization

Once all the analysis is done, it's time to implement a new version of the Runge-Kutta that uses all the advantages of the GPU. This step will be done in several iterations to be sure that the code is working and that the performance is improved.

Some sub-tasks will appear under this one but they will be defined during the project.

4 Planning

To manage the project, the project tool from GitHub will be used. This allows us to link code with issues that represent activities and add better tracking of the project. The planning is available on my personal GitHub account [14].

The following figure 4.1 shows the planning of the project. On August 4th, the report needs to be finished and sent to the HEIA-FR but the internship will continue until the end of August 11th. Normally, the Bachelor thesis begins directly at the LBNL but due to the time required to get a visa, the plane has been delayed and the project has started in Switzerland. This week will be used to finish the work on-site and to present the work done to the team. Exceptionally, the Bachelor's thesis end date can be delayed up to one week due to the disagreement to obtain a visa.

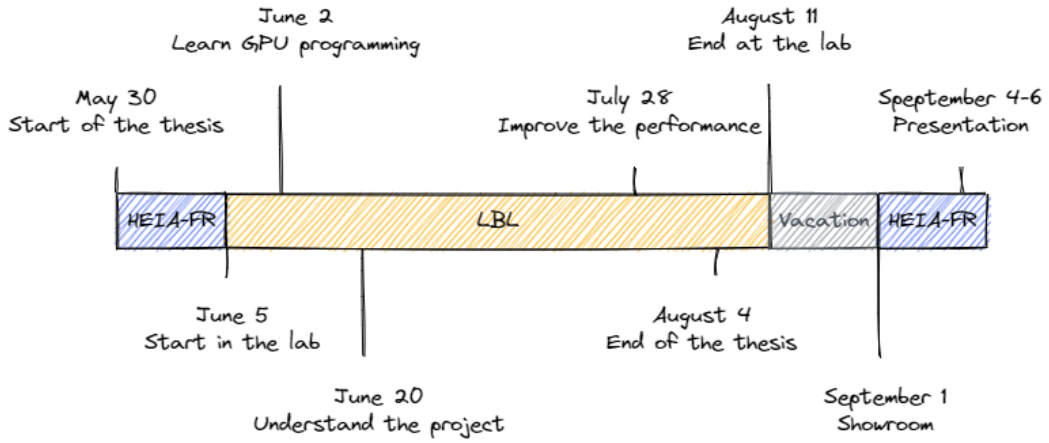

Figure 4.1: Timeline of the project

On 1st of September, there is a showroom at the HEIA-FR where all the students will present their work. The final presentation will be held between the 4th and the 7th of September.

The detailed planning is available on my GitHub but the figure 4.2 is an overview of the planning.

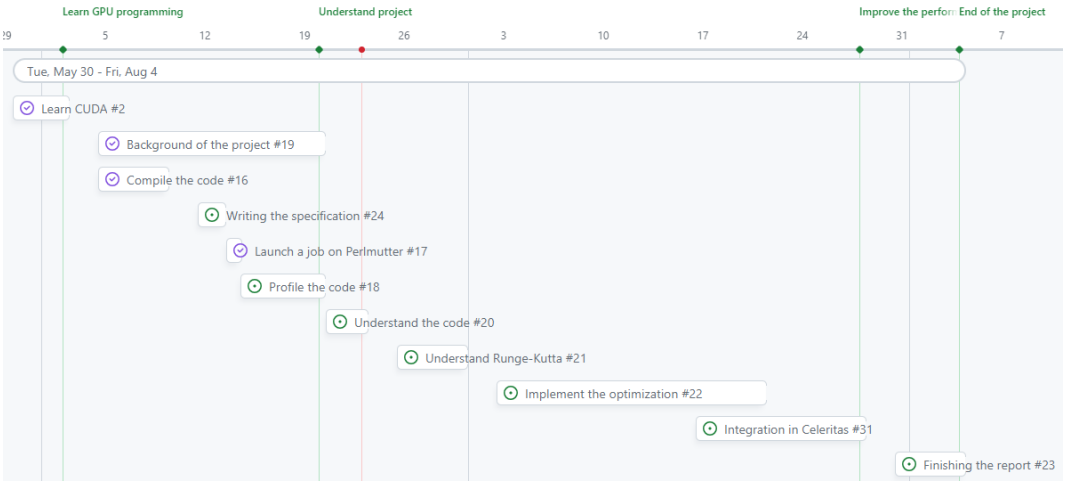

Figure 4.2: Planning

4.1 Milestones

The milestones represent the main steps of the project and they are used to track the progress of each step and set a list of activities to do. The milestones are shown in the project timeline 4.1 and in the GANTT chart 4.2 in green.

4.2 Issues

The issues are representing the tasks that need to be done to complete a milestone. They can be detailed with a checklist to specify the steps to follow and they can be linked to a pull request to keep a reference to the code that solves the issue. The GANTT chart 4.2 shows the main issues like a Gantt diagram.

List of Figures

References

[1] celeritas-project. *celeritas*. https://github.com/celeritas-project/celeritas; accessed 17-June-2023. [GitHub repository; Commit c8db3fc; Celeritas is a new Monte Carlo transport code designed for high-performance simulation of high-energy physics detectors.] 2023.

[2] S. Agostinelli et al. "Geant4—a simulation toolkit". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8. URL: https://www.sciencedirect.com/science/article/pii/S0168900203013688.

[3] Wikipedia contributors. *Dormand-Prince method — Wikipedia, The Free Encyclopedia*. [Online; accessed 17-June-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=Dormand%E2%80%93Prince_method&oldid=1138052375.

[4] Apr. 2023. URL: https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods.

[5] CERN. *2016 physics season starts at the LHC*. URL: https://home.cern/news/news/accelerators/2016-physics-season-starts-lhc.

[6] *VecGeom/VecGeom: The new geometry library for ROOT*. URL: https://gitlab.cern.ch/VecGeom/VecGeom.

[7] Kat Boogaard. *How to write SMART goals (with examples)*. https://www.atlassian.com/blog/productivity/how-to-write-smart-goals; accessed 17-June-2023. 2023.

[8] Nersc. *Using perlmutter*. URL: https://docs.nersc.gov/systems/perlmutter/.

[9] May 2023. URL: https://en.wikipedia.org/wiki/Ampere_(microarchitecture).

[10] Oak Ridge. *CUDA Training Series*. URL: https://www.olcf.ornl.gov/cuda-training-series/.

[11] Oak Ridge. *Simbarras/cuda-training-series: Training materials associated with Nvidia's cuda training series (www.olcf.ornl.gov/cuda-training-series/). to train my Bachelor thesis*. URL: https://github.com/simbarras/cuda-training-series.

[12] *Modules documentation*. URL: https://modules.readthedocs.io/en/latest/.

[13] *Spack documentation*. URL: https://spack.io/.

[14] Simon Barras (Simbarras). *GitHub Project: Bachelor Thesis*. URL: https://github.com/users/simbarras/projects/3/views/1.

Acronyms

ANL Argonne National Laboratory.

API Application Programming Interface.

ATLAS A Toroidal LHC ApparatuS.

BNL Brookhaven National Laboratory.

CERN European Organization for Nuclear Research.

CMS Compact Muon Solenoid.

CPU Central Processing Unit.

CUDA Compute Unified Device Architecture.

DOE Department of Energy.

FNAL Fermi National Accelerator Laboratory.

GPU Graphics Processing Unit.

HEIA-FR Haute Ecole d'Ingénierie et d'Architecture de Fribourg.

HEP High Energy Physics.

HPC High Performance Computing.

LBNL Lawrence Berkeley National Laboratory.

LHC Large Hadron Collider.

NERSC National Energy Research Scientific Computing Center.

ORNL Oak Ridge National Laboratory.

OS Operating System.

RKDP Runge Kutta Dormand Prince.

SDK Software Development Kit.

SIMD Single Instruction Multiple Data.

SIMT Single Instruction Multiple Thread.

SISD Single Instruction Single Data.

SM Streaming Multiprocessor.

SMART Specific, Measurable, Achievable, Relevant, Time-bound.