
GSoC Proposal: Improving performance of BioDynaMo using ROOT C++ Modules

30th March 2024

MENTORS: Vassil Vassilev, Lukas Breitwieser, Tobias Duswald and Fons Rademakers

AUTHOR: Isaac Morales Santana



INDEX

INDEX.....	2
CURRICULUM VITAE.....	3
Details.....	3
About myself.....	3
Academic Performance.....	3
Projects.....	3
ABSTRACT.....	4
EXPECTED RESULTS.....	5
OTHER GOALS.....	5
PROJECT TIMELINE.....	5
REFERENCES.....	8

CURRICULUM VITAE

Details

- **Name:** Isaac Morales Santana
- **E-Mail:** imorsan@correo.ugr.es
- **Timezone:** GMT+2
- **University:** Universidad de Granada (UGR)
- **Course:** Bachelor's Degree in Computer Engineering
- **GitHub:** [imorlxs \(Isaac Morales Santana\) \(github.com\)](https://github.com/imorlxs)
- **LinkedIn:** [Isaac Morales Santana | LinkedIn](#)

About myself

I am a second-year student at the **University of Granada (Spain)**. My areas of interest include **concurrent programming**, **embedded systems**, and **high-performance computing**. Since I was very young, I've been fascinated by open-source software, particularly Linux. I am a member of the **Google Developer Student Club** at my university. I have been programming in C++ for three years, although I had a keen interest in Arduino when I was younger. I have experience in C, C++, and Python.

I have **no major commitments** other than **GSoC** during the summer break, and I will give my best effort to complete the project idea within the given timeframe.

The only commitment I have is that I have final exams until June 24th. However, I will ensure that this does not hinder my participation in the project.

Academic Performance

Thanks to my passion for these subjects, I have achieved three honors in the courses: Methodology of Programming, Data Structures, and Operating Systems.

Projects

IMAGE MANIPULATION WITH DATA STRUCTURES IN C++.

- I developed a program capable of rotating, saving, and loading images, as well as adding transparency. The main functionality involves processing files containing aerial routes. When a user selects a route, the program prints a plane at each point of the route, adorned with the flag of the corresponding country. This project was undertaken as part of coursework in Data Structures. We used CMake to compile this project.

-
- You can view the code [here](#).

POKÉDLE RESOLVER IN PYTHON

- I developed Pokédle Resolver, a bot aimed at solving puzzles on Pokedle. Drawing inspiration from the LOLdleSolver project, I modified its code extensively to tailor it for solving Pokedle puzzles instead. This project significantly enhanced my proficiency in Python.
- You can view the code [here](#).

LANGUAGE DETERMINATOR IN C++

- During my studies in Programming Methodology, I developed an application that could analyze a text in an unknown language and determine its language of origin. To accomplish this, I trained the model using texts from known languages. It's important to note that the application can only identify languages that have been previously analyzed.
- Link to the project [here](#).

ABSTRACT

BioDynaMo, a powerful agent-based simulation platform, utilizes ROOT for functionalities like statistical analysis, random number generation, and IO. However, BioDynaMo's reflection system, which relies on efficient access to class information, can be optimized for improved performance. This project proposes upgrading to C++ modules within the ROOT framework to achieve this goal.

Let's break down the various components involved in this project:

- **ROOT:** ROOT is a framework for data processing, born at CERN, at the heart of the research on high-energy physics. It provides capabilities such as IO, a C++ interpreter, GUI, and math libraries. It uses object-oriented concepts and build-time modules to layer between components.
- **CMake:** CMake is a cross-platform build system used to control the software compilation process. It generates native build files for various platforms and simplifies the build process with platform-independent configuration files.
- **Genreflex:** ROOT Dictionary generator using REFLEX/GCCXML.
- **Rootcling:** Rootcling generates the dictionaries needed for performing I/O of classes.
- **ROOT C++ Modules.** Modules are a system to more efficiently handle used libraries in C++. Work like precompiled headers (PCHs) (Vassilev & Isemann, 2017)

EXPECTED RESULTS

- Rework the CMake rules to incorporate efficiently ROOT via FetchContent.
- Replace invocations of *genreflex* in favor of *rootcling*.
- Enable C++ modules in *rootcling*.
- Produce a comparison report.

OTHER GOALS.

In addition to the core objectives, we will address other important goals:

- Transitioning from wget/curl scripts to CMake's FetchContent module for external dependency management. (I am already working on that)

PROJECT TIMELINE

<i>Community Bonding Period</i>	
May 1 - May 26	Engage with the community. Establish regular meetings with the mentors. Set up the development environment. Set up external tools such as LLDB and Valgrind for debugging and profiling the code. Make documentation audits and submit improvements to it where necessary. Write a blog post announcing my project to the community on the compiler-research.org webpage. Development of a detailed project plan outlining specific tasks and milestones.
<i>Coding period begins</i>	
Week 1 27.05.2024 -02.06.2024	Add check conditions in CPPFunctionNumbaType to identify cuda kernel calls. Deliverable: Adding functionality
Week 2 03.06.2024-1 0.06.2024	Rework the CMake rules to incorporate efficiently ROOT via FetchContent. Deliverable: External packages do not require curl or wget to integrate at cmake configuration time.

Week 3 10.06.2024-1 6.06.2024	Further exploration of ROOT's C++ module functionality.
Week 4 17.06.2024-2 3.06.2024	(No activity due to finals week)
Week 5-6 24.06.2024-0 7.07.2024	Enable the C++ Modules in rootcling. Replace the indirection of invoking reflex from a script.
Week 7-8 08.07.2024-2 1.07.2024	Replace invocations of genreflex in favor of rootcling.
<i>Midterm Evaluations</i>	
Week 9 22.07.2024-2 8.07.2024	Making further changes in the code to improve the functionality, exception handling, bug removal.
Week 10 29.07.2024-0 4.08.2024	Most of the time will be consumed for rigorous testing and bug fixes.
Week 11 5.08.2024-11 .08.2024	Buffer Week
Week 12-13 12.08.2024-2 6.08.2024	Extended testing, developing documentation, presenting the work. Deliverable: test cases, demonstrated reduction of the binary sizes, blog post about the achieved results; presentation at the compiler-research.org team meeting.

Thank you so much for considering my proposal and for taking the time to review the effort being put forth. I am hopeful for the opportunity to work with **CERN** in the near future, anticipating numerous exciting and interesting developments ahead.

REFERENCES

(n.d.). CMake - Upgrade Your Software Build System. Retrieved March 31, 2024, from

<https://cmake.org/>

Chapter: *Cling*. (n.d.). CERN Root. Retrieved March 31, 2024, from

<https://root.cern.ch/root/html/doc/guides/users-guide/Cling.html>

GenreflexRoot6. (2017, November 9). Retrieved March 31, 2024, from

<https://twiki.cern.ch/twiki/bin/view/Main/GenreflexRoot6>

Vassilev, V., & Isemann, R. (2017, 07 17). *ROOT C++ modules*.

<https://indico.cern.ch/event/643728/contributions/2612822/attachments/1494074/232389>

[3/ROOTs_C_modules_status_report.pdf](#)