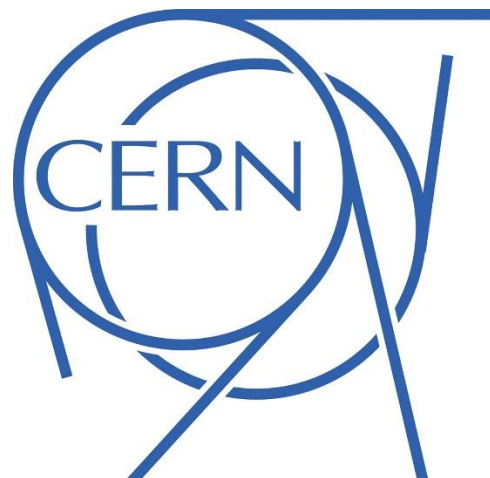




CERN

Summer Studentship Report

Analysis and improvement of the DIRAC CPU Performance Benchmarking tool

Student: Imane Iraoui

Studentship period: 14 June – 6 August

LHCb Experiment

Supervisors:

Federico Stagni

Alexandre Franck Boyer

Summer 2021

Table of Contents:

1. Abstract
2. CERN and the LHCb Experiment
 - a. CERN
 - b. LHCb
3. Project Description
 - a. Dirac Benchmark 12
 - b. Coding Part
 - c. Analysis Part
4. Conclusion
5. Acknowledgements

1. Summary

For the Summer 2021, I was selected to participate in CERN's summer student program. I got the chance to work on the LHCb experiment, specifically to work on the analysis and improvement of the DIRAC Benchmark 12.

Moreover, I got the chance to attend a series of lectures that were offered as part of the program. They covered a range of different topics: Particle Physics, Accelerators, Machine Learning... I also had the chance to visit LHCb virtually, since I was not physically present at CERN.

In this report, I will be discussing my work on DB12, a CPU Benchmarking Tool. In the first section, I will give some context about CERN and the LHCb experiment. Then, in section 2, I will describe the project in detail, to then discuss my work in section 3 and finally conclude the report.

2. CERN and the LHCb Experiment

2.1. CERN:

CERN, which is short for Organisation européenne pour la recherche nucléaire (European Organization for Nuclear Research) is a European scientific research organization based in Geneva on the Franco-Swiss border. It was established in 1954 by 12 founding members and currently has 23 member states.

CERN's main function is to provide the particle accelerators and other infrastructure needed for high-energy physics research. For this purpose, numerous experiments are being conducted at CERN, such as the LHCb experiment, the CAST experiment, and the AEGIC experiment, among others.

2.2. LHCb

The LHCb experiment, short for Large Hadron Collider beauty, is one of eight particle physics detector experiments collecting data at the LHC, which is the world's largest particle collider. The LHC's purpose is to allow physicists to test the predictions of various particle

physics theories, such as measuring the properties of the Higgs boson, searching for the family of new particles promised by supersymmetric theories, and other unsolved particle physics topics.

The initial focus of the LHC was to investigate the existence of the Higgs boson, and it was detected in July 2012. Some other discoveries included several composite particles (hadrons) like the χ_b (3P) bottomonium state, the first creation of a quark–gluon plasma, and the first observations of the very rare decay of the Bs meson into two muons ($B_s^0 \rightarrow \mu^+\mu^-$), which challenged the validity of existing models of supersymmetry.

3. Project Description:

3.1. CPU Benchmarking:

CPU Benchmarking allows for easy comparison between multiple CPUs by scoring their performance on a standardized series of tests, and it allows to construct theories about CPU performance. It is especially useful since it is necessary to know whether a certain task will have enough computing resources to be able to complete successfully before executing said task. It also helps to report the actual usage of computer resources and to be able to keep an accounting of the consumed resources.

The traditional way the CPU power has been estimated since 2007 is using the HEP-Spec06 benchmark, which was verified at the time to scale properly with a set of typical HEP applications. However, processors have evolved since then and the scaling with HS06 is no longer fulfilled. So, another benchmark tool is used which is DB12, the one my work revolves around.

3.1.1. DB12:

DB12, short for DIRAC Benchmark 12, is a fast benchmark tool designed to run within DIRAC and used for evaluating at run time the performance of the worker nodes.

DB12 gives a faster, better estimation of the CPU power for LHCb applications. It's also run from inside the DIRAC pilot before fetching a job, and so in combination with the CPU time left, which we can get by interrogating the batch system, we can get an idea of which jobs we can run.

DB12 however has some issues:

- It does not support Python 3, as it is entirely written in Python 2.
- Does not include CI/CD.
- Copy pasted, instead of imported, within DIRAC.
- Does not run well in multi-core environments: There is a function included in DB12 to run it on multiple cores in parallel, but it takes a lot of time to run and the accuracy of the scores is not certain. About 20% of the jobs fail because they run out of time on the Santos Dumont supercomputer for instance (See Figure 1). Currently when there are multi-cores DB12 is just run on one of the cores. This is not enough as DB12 should be run in parallel on every core and the lowest value should be used as reference, to make sure that jobs will not run out of time.

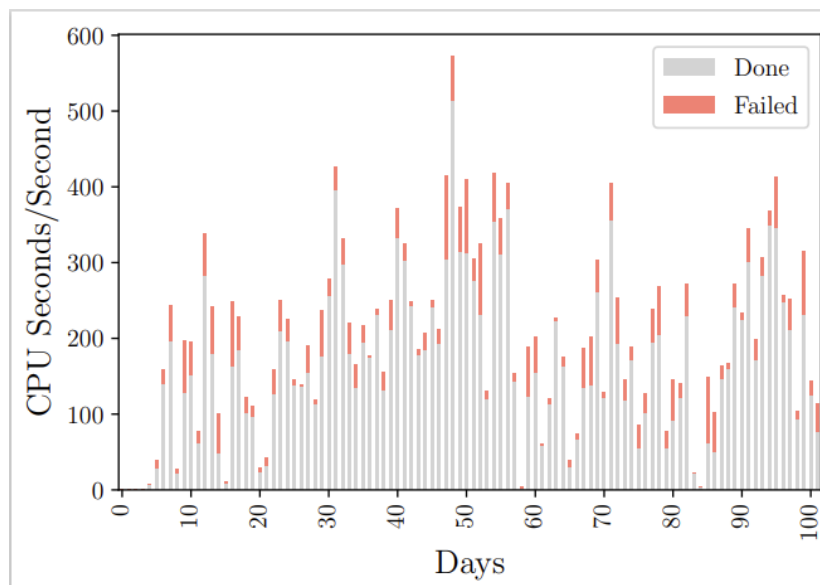


Figure 1: Number of jobs running in parallel on Santos Dumont (multi-core environment)

To fix these issues, my work consisted of two parts: a coding part and an analysis part.

3.1.2. Coding Part

First, since DB12 was entirely written in Python 2, I had to port it to Python 3 and make it support both Python 2 and Python 3. To do that, I rewrote the code in Python3 and mainly used

the `__future__` package to re-enable compatibility with Python 2. However, part of DB12 involves `xrange` that does not exist anymore in python3, and by running a few tests it was confirmed that python2 and python3 DB12 scores are a bit different, so the range and xrange difference still needs to be verified to make sure that it does not create any issues.

The second task I had to do was format the code using Pylint. Then, I had to develop a unit test mainly using Pytest. This unit test was to test whether:

- Each function used in DB12 produces the same output when given the same inputs multiple times.
- Each function produced the same output when run for a different number of iterations.
- The result produced is positive and not unreasonably high (i.e <100).

I was also tasked with developing an integration test, which consisted of comparing results from different versions of Python to make sure that they are the same within a specified threshold. The current implementation is not perfect because we compare scores that have been generated on different machines in parallel, and we are not sure that machines are completely identical. A better approach would be to execute DB12 with different python versions on a same machine using our own runner since the integration test fails sometimes.

To run those tests, I built a CI: this meant building a pipeline on Github Actions to run the tests and to check the format using Pylint.

To execute the code on different Python versions, I used Conda which helped create environments, one for each version. After the tests, I stored the execution results in GitLab artifacts to later use them in the integration test.

The next task consisted of building a CD to publish DB12 as a package on Pypi, using Github Actions, after each tag on Github. To do that, the structure of DB12 had to be changed to a more standard one. This meant including an `src` folder, a `setup.py` file, and a `tests` folder:

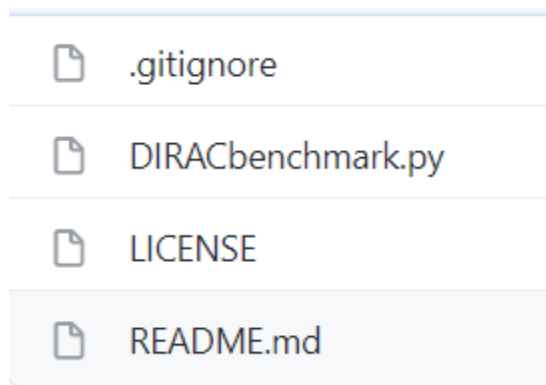


Figure 2: Old Structure of the DB12 Project

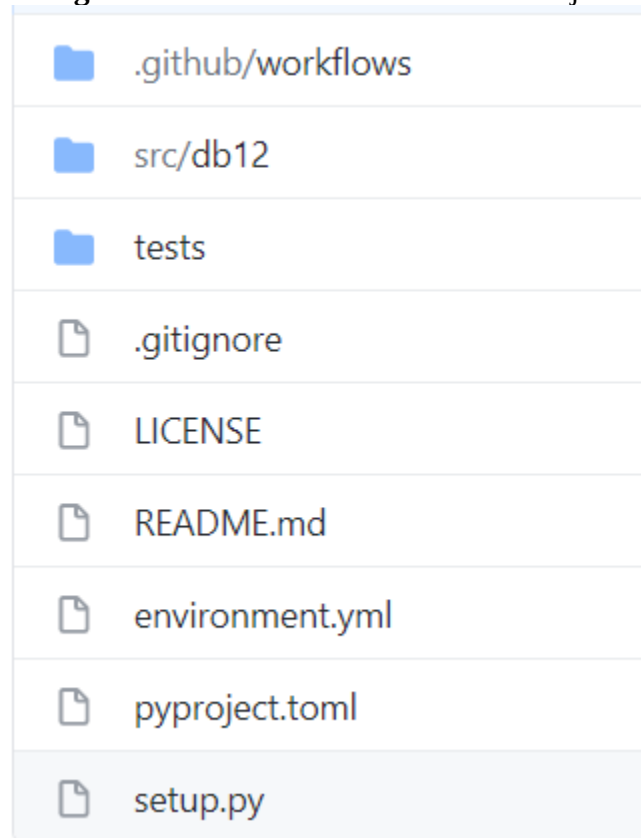


Figure 3: New Structure of the DB12 Project

And the last task of the coding part consists of using the DB12 package inside DIRAC by importing it instead of copy pasting it which is what is currently done.

3.1.3. Analysis Part

The second part is also divided into two parts: collecting data and analyzing data.

- To collect data, multiple jobs need to be submitted on many sites to collect data about CPU information, number of events processed, the duration, and the DB12 score (with different python versions, using a different number of iterations and on a different number of cores).
- The second step is analyzing the data that was collected. That was done by comparing DB12 scores per site, per number of iterations, per python version and per number of cores involved, in addition to also comparing DB12 scores with the number of events processed in a given duration.

4. What I Learned

Thanks to this opportunity at CERN, I had the chance to deepen my knowledge concerning Python, and I discovered and used many new tools that I did know about before. I also learned a lot about data analysis with Python which is an important skill to have as a computer science student.

I also learned a lot by attending the lectures, and I even ventured out of my field of interest, which is mainly Computer Science, and discovered a lot of things in the field of Physics.

5. Acknowledgements

I want to extend my gratitude to Alexandre Franck Boyer, my supervisor at CERN who helped me a lot throughout my time here. He guided me through any issues that I faced and was also very responsive to my emails and very encouraging and helpful. I also want to thank Federico Stagni, my supervisor as well, for providing useful feedback about the work and for all his answers to my questions. And lastly, I would like to thank the LHCb team and the Summer Student team for this amazing opportunity and for the organization which allowed a smooth run of the program.