GPU accelerated programing

Amaro Jr., Rafael A. N. and Igor Abritta
30/09/2020

Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora, MG

Summary

- Overview of last presentation;
- Solutions tested;
- What has been done so far;
- RAPIDS.

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

Last presentation

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

- We are starting to study the possibilities and potential of using GPU in the trigger processing part;
- A decision on the use of **DirectGMA (AMD)/GPUDirect (NVIDIA)** must be made, as it influences the choice of the readout electronics and computing hardware;
- The **code environment (C++,Python,etc)** and **GPU hardware** should also be chosen but it is not strongly related to the readout electronics.

Summary

- Overview of last presentation
- Solutions tested
- What's been done so far

Solutions tested

- C++ implementation of the G-DBSCAN algorithm;
- Python implementation of the G-DBSCAN;
- Python library for GPU Data Science (RAPIDS);
- All of the following work has been done with CUDA in mind, given that is the technology that we have available.

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

What has been done so far - C++

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

- We were not able to do much progress creating a C++ implementation of the G-DBSCAN algorithm ;
- Being new to parallel computing, I've decided to take some steps back and try something that would be faster to do and easier to adapt to current work;
- We still believe that a C++ implementation would be crucial for faster results in an online environment.

What has been done so far - Python

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

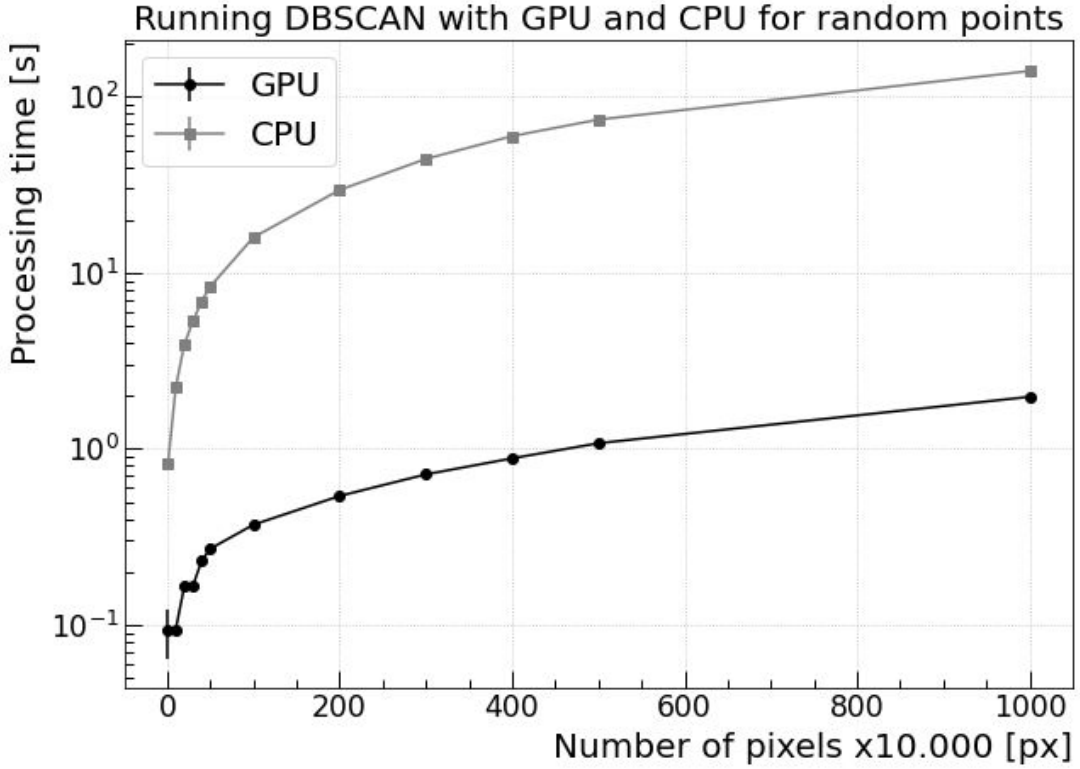
- The second approach was to implement the G-DBSCAN in python, but after some research we've decided to look somewhere for an already made implementation;
- We found RAPIDS:
  - RAPIDS is a python library that implements many Data Science methods in GPU using CUDA;
  - One of the already implemented methods is DBSCAN.

RAPIDS

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

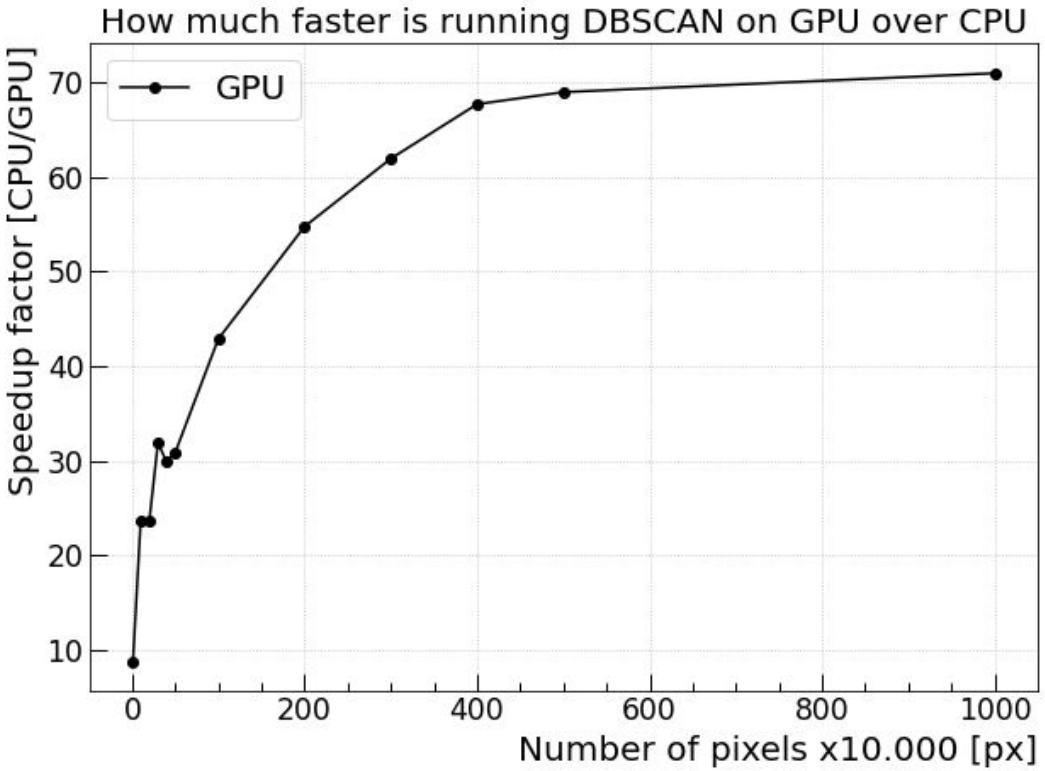
- Running on colab during preliminary tests in order to get experience with CUDA and GPU processing;
- Now for the next steps we'll move the codebase to GAP01 for some more robust testing.
- RAPIDS is a good option so far because applying it's accelerated DBSCAN method with experiment data is very easy

RAPIDS results



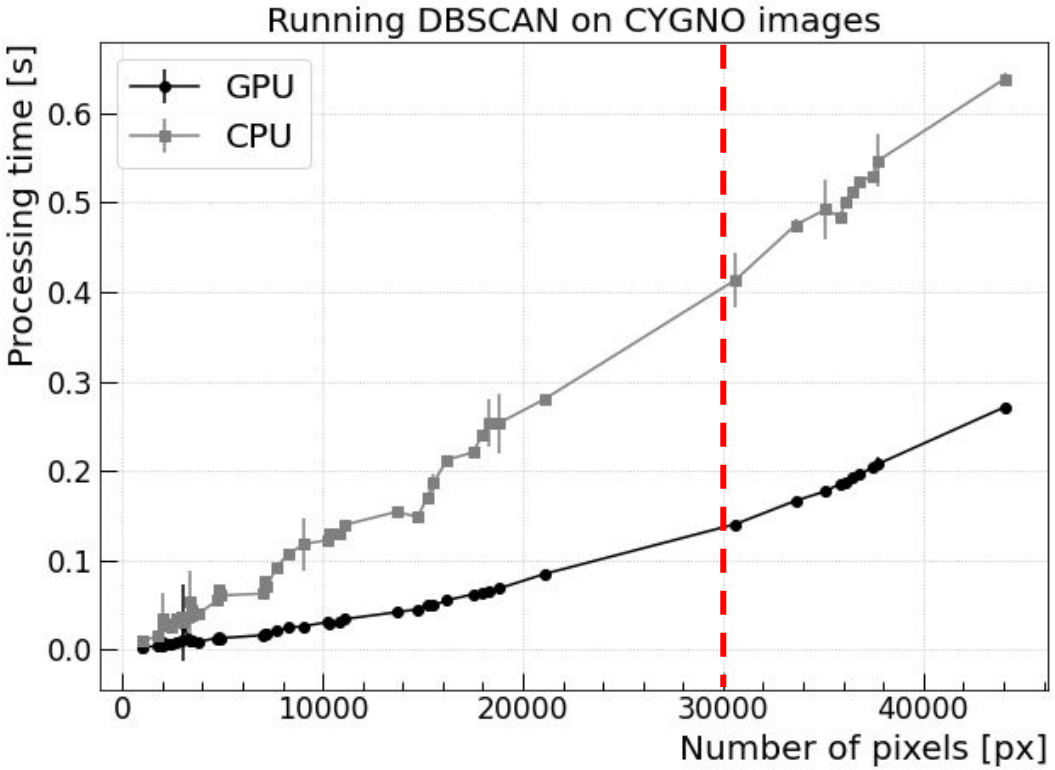
- Randomly distributed noise

RAPIDS results



- Randomly distributed noise
- With colab the speed apparently caps at 70x, but given the right machine could be even better

RAPIDS results



- Run 2089
- 10 images analyzed
- It was used the default preprocessing (pedmap, zero suppression, median filter and noise reductor);
- And scanning over the **nsigma (from 3 to 1.5)** in order to get different number of pixels.
- GPU is 3x faster for 30k pixels

Next steps

UNIVERSIDADE FEDERAL DE JUIZ DE FORA

- As mentioned, our next step will be moving the codebase to GAP01 for further testing
- Stress test the RAPIDS library with experiment data