Data Reduction for Cherenkov Gamma-Ray Astronomy on Jetson TK1

A. Madonna¹,², M. Mastropietro¹,³, D. Bastieri²,⁴, S. Lombardi¹,³, L.A. Antonelli¹,³

1. INAF – Rome Astronomical Observatory, Monteporzio Catone (Rome), Italy
2. GPU Research Center, University of Padova, Italy
3. ASI Science Data Center, Rome, Italy
4. Dept. Physics and Astronomy, Univ. Padova and INFN, Padova, Italy

Abstract

A mini-array of ASTRI SST-2M Cherenkov telescopes will be deployed soon in a remote site far away from human activities to achieve optimal observation conditions for gamma-ray astronomy. In such a scenario, the capability of each telescope to process its own data before sending them to a central acquisition system provides a key advantage. We implemented the complete analysis chain required by a single telescope on a Jetson TK1 development board, overcoming the required real-time processing speed by more than a factor two, while staying within a very small power budget.

The ASTRI Project and Low-Power Computing

The ASTRI collaboration¹ (Astrofisica con Specchi a Tecnologia Replicante Italiana) is a flagship project of INAF (Italian National Institute for Astrophysics) funded by the Italian Ministry of Education, University and Research related to the next generation IACT (Imaging Atmospheric Cherenkov Telescope), within the framework of the CTA² (Cherenkov Telescope Array) International Observatory. The project aims at developing and deploying a full end-to-end prototype of dual-mirror Small Size Telescope (SST-2M) to observe gamma-ray photons through the Cherenkov radiation of their atmospheric showers; the prototype will be compliant with the requirements of the High Energy array of CTA.

IACTs are typically installed in isolated places in order to avoid luminous sky background. Such remote locations pose critical constraints on several aspects related to data processing, in particular: power consumption, data bandwidth, heat dissipation, compactness, etc.

The ASTRI Reconstruction Software

The ASTRI Reconstruction Software (or ARS, for short) is the software suite that implements all the tools and modules needed for the data reduction and analysis pipeline for both the SST-2M Prototype and the Mini-Array. It follows the general design and data model scheme that has been defined for the CTA and uses the Flexible Image Transport System (FITS) as the official data format. It is written in C++ and CUDA 7, targeting conventional x86 architectures as well as GPUs for accelerated parallel processing and ARM cores for low-power operation.

Pipeline implementation

The software receives input from the telescope in form of raw ADC counts (Data Level 0 or DLO), calibrates data (D1La), cleans signal noise and computes image parameters and moments (D1Lb). At this level, data size is reduced of about 70x with respect to the DLO input. Data is then processed by a machine learning module (based on Breiman’s Random Forests) that further estimates characteristics of the shower (D1Lc). The reconstructed single-detector events are merged together to yield array-level events (D1Lb), which are finally combined with Instrument Response Functions, providing a data product suitable for the Science Tools (DL3).

The above data model relies heavily on disk I/O. This can lead to significant overheads for the first two modules, until a drastic reduction in size is achieved at D1Lb. The CTA requirements define an On-line Analysis mode (real-time analysis), where reduced steps and accuracy are allowed in favor of a faster processing.

Since this mode is the one where an embedded solution would prove most effective, we produced a specific implementation for the complete on-line analysis of single-telescope data (DL0–DL1c, i.e. before the events from the whole array are merged together and processed as such). This procedure comprises:

• ADC Calibration (essentially an FMA operation)
• Image cleaning (two-pass threshold comparisons)
• Image momenta and shower parameters computation
• Event classification through pre-trained Random Forest models
Most notably, the processing between DLO and D1Lb (fully reduced data) has been reformulated in a “Unified Module” to minimize disk transfer time. The unified module is CUDA accelerated and runs on the GPU, while the subsequent classification is performed on multiple CPU cores.

Test Case

We carry out our tests on a dataset of 55000 simulated events (approximately 500MB). This corresponds to an estimated 110 seconds of nominal acquisition at 500kHz, or 55 seconds at the peak acquisition rate of 1kHz. This data sample is fully compliant with the format and interchange size agreed with the camera hardware team.

Introducing the Jetson TK1

We selected the NVIDIA Jetson TK1 as the platform for our embedded implementation. In the first version of the code, we made extensive use of mapped pinned memory for Zero Copy access. When working with integrated GPUs, this should always be a performance gain, because it avoids superfluous copies as integrated GPU and CPU memory are physically the same. However, we realized that using pinned memory led to very poor performance when operated on by the CPU portions of the code. An effective solution to this problem would be to make sure that the memory ranges that have to be accessed by the CUDA kernels using cudaHostRegister(). Unfortunately, this function is not supported on ARM platforms by the CUDA driver that ships with the Jetson devkit. We therefore resorted to managing Device memory the old-fashioned way, as if we were working with a discrete GPU. The performance penalty caused by pinned memory on Host code is such that this approach is faster nonetheless.

The next biggest factor limiting throughput was loading data from the Jetson eMMC. Outfitting the board with an external SATA HDD and subsequently an SSD significantly improved the overall execution times. The latest version of the code completes the single-telescope analysis chain on the 55000 events dataset in about 20 seconds (approximately 2750 events per second), thus achieving more than double the speed required by peak real-time processing. Power consumption remains limited, averaging at 7W during operation and idling at 3W (including power required by the SSD). We are also confident that a throughput of more than 3000 events/s is fully within reach with future improvements.

Conclusions

We successfully tested Jetson TK1 as an embedded module capable of carrying out the whole single telescope data processing pipeline including signal calibration, reduction and preliminary reconstruction. Leveraging its efficient SoC, we showed that it can process twice as much of the required data flow (more than 2x events/s) with a power consumption as low as 10 W. This makes Jetson TK1 a promising embedded processing module for the online data analysis of gamma-ray astronomy with Cherenkov telescopes.