PROCEEDINGS OF SCIENCE

Software framework and the reconstruction software of the DAMPE mission

Andrii Tykhonov∗,[1] **Chi Wang,**[2] **Xin Wu,**[1] **JingJing Zang**[3] †

[1]*DPNC, Université de Genève, CH-1211 Genève 4, Switzerland*
[2]*University of Science and Technology of China, 1129 Huizhou Ave, Baohe, Hefei, Anhui, China*
[3]*Purple Mountain Observatory, CAS,2 West Beijing Road, Nanjing 210008, China*
E-mail: andrii.tykhonov@cern.ch

An overview is given for the offline software framework and reconstruction software of the DAMPE (DArk Matter Particle Explorer) high energy astroparticle mission. DAMPE is a satellite mission in the framework of the Strategic Pioneer Research Program in Space Science of the Chinese Academy of Sciences, with a launch date scheduled for the end of 2015. The detector consists of a silicon-tungsten tracker-converter, comprising 12 layers of single-sided silicon-strip detectors, interleaved with 3 layers of tungsten converters, BGO calorimeter, and plastic scintillator, serving as anti-coincidence detector, and a layer of neutron detector in the bottom of the calorimeter. DAMPE analysis and reconstruction software is implemented based on the custom-made software framework, where the core software is written in C++, while the management part is done in Python. We take advantage of the boost-python libraries, whereby the bridge between the core and management part is done, allowing to fully exploit a computational power of modern CPUs, while keeping the framework flexible and easy to deploy. The building blocks of the framework are the algorithms, which are stacked together and configured in the job-option files. The geometry of the detector is implemented in the GDML format, through the direct conversion from the CAD drawings of the detector to the Geant4-compatible format. The data flow is handled by the dedicated input-output service, based on ROOT. The simulation algorithms are implemented with the Geant4 tool kit. In the heart of the reconstruction software lies the pattern recognition for the initial track finding, which is refined further by the track filtering algorithm, based on the adaptation of Kalman technique. The software has been extensively put on test during the beam test campaigns at CERN, in 2014-2015, proving its sustainability to a wide range of data processing challenges, encountered in a particle physics experiment.

The 34th International Cosmic Ray Conference,
30 July- 6 August, 2015
The Hague, The Netherlands

∗Speaker.
†For the DAMPE Collaboration

PoS(ICRC2015)1193

1. Introduction

DAMPE is a powerful satellite-borne telescope for high energy gamma-ray, electron and cosmic rays detection [1]. It consists of a double layer of plastic scintillator strips detector (PSD) that serves as anti-coincidence detector, followed by silicon-tungsten tracker-converter (STK), which is made of 6 tracking double layers; each consists of two layers of single-sided silicon strip detectors measuring the two orthogonal views perpendicular to the pointing direction of the apparatus. Three layers of tungsten plates with thickness of 1mm each are inserted in front of tracking layer 2, 3 and 4 for photon conversion. The STK is followed by an imaging calorimeter of about 31 radiation lengths thickness, made up of 14 layers of Bismuth Germanium Oxide (BGO) bars in a hodoscopic arrangement. A layer of neutron detectors is added to the bottom of the calorimeter. The total thickness of the BGO calorimeter and the STK makes DAMPE the deepest calorimeter ever used in space.

In this proceeding we discuss the framework of the offline software of DAMPE mission, and describe preliminary results on the reconstruction software. The overall performance of DAMPE software has been exercised during the beam-test campaigns at the CERN PS and SPS accelerators in 2014 and 2015, with the engineering-qualification model of satellite (EQM), as well as with the comsics data taking with the DAMPE flight model (FM). At the moment this proceeding being submitted, an active analysis campaign is underway at DAMPE collaboration, analysing the beam-test and cosmics data, using the software framework described here.

The proceeding is structured as follows. In Section 2 we outline the overall structure of the offline software with the technologies being used. Next, in Section 3 we examine how the geometry model of DAMPE is implemented in the software, in particular the conversion from the CAD engineer drawings to the GDML (XML) model, which is used in the simulation and reconstruction. In section 4 the techniques for track pattern recognition implemented in DAMPE are described.

2. The DAMPE Offline Software Infrastructure

The structure of DAMPE offline software framework is driven by the GAUDI philosophy [2]. The core part is implemented in C++. The central component of framework is the algorithm base class. All algorithms are inherited from the base class, including algorithms for the conversion of binary data, simulation, and reconstructions. In addition to algorithms, there are core services, like the algorithm manager, input-output service (based on ROOT), and geometry manager. The latter one allows for reading DAMPE geometry from the XML database without using Geant4 [3] (or ROOT) libraries, reducing significantly the overhead of initialising the reconstruction and analysis jobs. Simulation is implemented as a dedicated algorithm, based on the Geant4 tool kit [3]. Not only sensitive detectors are included into simulation, but also all the supporting structures of the satellite are taken into account, owing to a custom technology based on CADMESH [4] (see Section 3). The sketch of the DAMPE offline software is shown in Fig. 1.

All DAMPE offline jobs are configured via job option files written in Python. The boost-python libraries [5] are exploited for accessing the C++ algorithms and services from Python. The algorithms can be easily stacked together inside a job option, forming a sequence, where the input-output service acts as a data buffer for communication between algorithms, and for reading

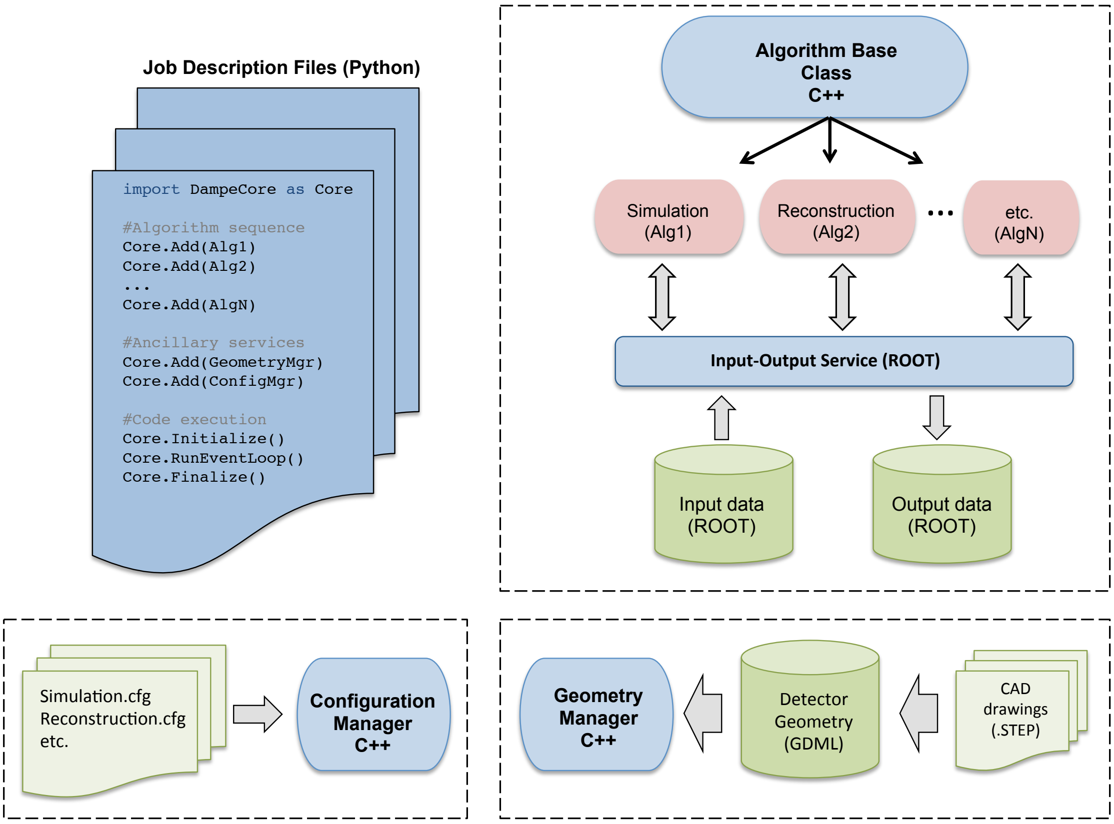

Figure 1: Block-diagram of the DAMPE offline software

(writing) the input (output) data to the ROOT files. The algorithm is not aware of whether data comes from the input file or as an output of preceding algorithm in a sequence, the only thing that matters it the name of the input "collection" of data and its type. Finally, although the algorithms and services can be fully configured via the job options, the configuration files are used as well, in order to store the configuration parameters that are shared by multiple algorithms, which makes versioning of the code easier to trace. In other words, each simulation (reconstruction) campaign can be "tagged" with one or more configuration files.

Although the majority of reconstruction jobs are usually run on the computing cluster, the whole DAMPE software framework can be also installed on a desktop computer with Linux or MacOS system on it. It requires ROOT, Boost libraries [5] and Geant4 (optional) to be installed beforehand. Also Scons [6] is required as a tool for compiling (assembling) the project. In case of analysis jobs, user does not require the whole DAMPE framework to be installed, instead a light-weight "Event" package is deployed. It contains the definitions of DAMPE data classes, and the corresponding ROOT dictionaries, allowing to read (analyze) the data either through compiled C++ code, CINT or in Python. The Event package requires only ROOT to be pre-installed on a user machine.

During the beam-test campaigns at CERN, 2014 and 2015, a fast data reprocessing has been

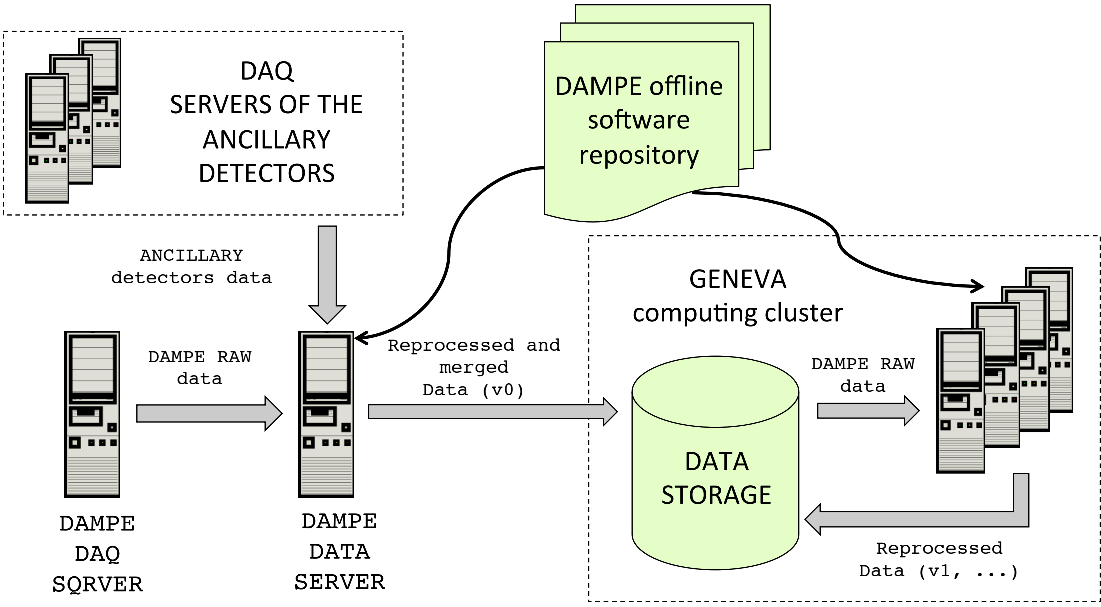

Figure 2: Block-diagram of data reconstruction procedure for the DAMPE beam-test campaigns at CERN.

performed on a server in a control room (v0 reprocessing). It is required for quick validation of detector performance in the rapidly changing environment of the beam test. A dedicated tool was developed for shifters in the DAMPE beam-test control room, allowing to run DAMPE raw-data conversion and reconstruction jobs with a single command-line interface, as well as merging DAMPE data with the data coming from the ancillary detectors (beam telescope, trigger scintillators, etc.). The tool is a wrapper on top of the DAMPE software. Next, v0 and raw data were transferred to the data storage in Geneva. Afterwards, as the data were understood better, a refinement of the reconstruction algorithms has been performed, resulting in subsequent processing campaigns, performed in the Geneva computing cluster (v1, etc.). A sketch of reconstruction sequence is shown in Fig. 2. Similar data-processing organisation is foreseen for the flight data taking. Moreover, not only Geneva cluster but also computing clusters in China and INFN (Italy) are planned to be exploited.

3. The DAMPE Geometry and CAD to GEANT4 Conversion

In this Section we review the implementation of DAMPE geometry in the offline software. From the very beginning our goal was to create a routine that allows to input the detector CAD (Computer Aided Design) drawings, provided by mechanic engineers, directly into the physics simulation and reconstruction code. In this way one avoids mistakes due to human intervention in the implementation of geometry, and more important – it allows for faster iterations between engineers and physicists during the detector construction, since any modification of geometry can

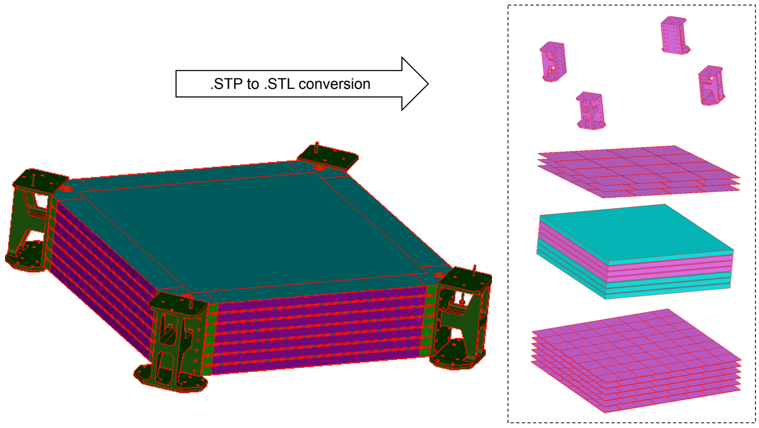

Figure 3: Cartoon illustrating conversion of engineering CAD model of the STK to GDML\XML geometry used by DAMPE simulation and reconstruction algorithms. The CAD drawing of the STK (.stp file) is exploded into a set of .stl files, each representing its own part of the geometry (from top to bottom in the figure): corner fits (aluminium), tungsten converters, aluminum honeycomb trays, silicon sensors. The .stl parts are converted further into tessellated structures using the CADMESH package [4]. Finally, the sensitive detectors are parsed with a custom software and the read-out silicon strip objects are created as simple geometries (not tessellated).

be immediately put on test in the physics simulation. It also automatizes versioning of the geometry, providing confidence that the simulated object corresponds to reality as much as possible.

Converting CAD drawings into Geant4 compatible format, GDML (Geometry Description Markup Language) [3], is a long-standing problem which has been attacked many times with no permanent success. The most widespread solution is based on conversion of CAD drawings into tessellated structures [4]. However, it suffers reasonable computing overhead due to necessity of dealing with complex geometries at the simulation level. In case of DAMPE we improve this technique by replacing sensitive parts of the detector (silicon sensors) with simple rectangular shapes. Silicon detectors can be parsed out from the geometry and split further into particular read-out strips. In doing so we bypass the problem of computing overhead with tessellated solids, while keeping the geometry exactly the same as in the provided CAD drawing. The CAD to GDML conversion is illustrated in Fig. 3. Before converting CAD geometry to GDML, an intermediate step is taken to explode the model into parts, where each part corresponds to one single material. Once those parts are converted into corresponding GDML modules, proper materials are assigned to them. Finally, an XML file with the description of sensitive detectors (their positions) is created. The later one is used as an input to reconstruction jobs, providing a mapping between read-out channel identifiers in electronics and real x, y, z positions of the hits. The 3D model of DAMPE visualised with Geant4 is shown in Fig. 4.

4. The DAMPE Reconstruction Software

In this Section we describe the current status of reconstruction software of DAMPE. It can be

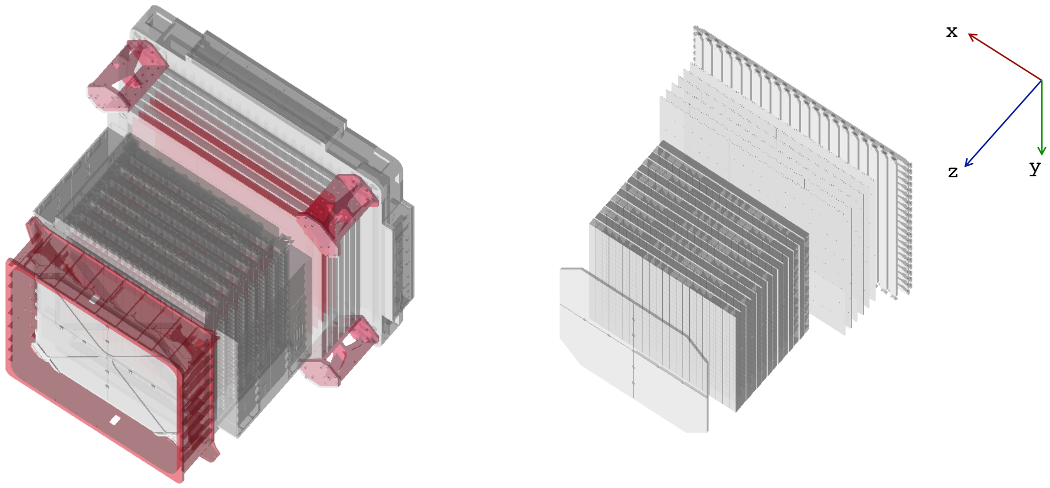

Figure 4: GDML geometry of DAMPE as it appears in the GEANT4 viewer: left - whole satellite payload, right - detector elements only (sensitive volumes).

divided in two parts: shower reconstruction in the calorimeter and track reconstruction in the STK. Those two form the objects that are used for further identification of photons, electrons etc. One of the goals of shower reconstruction in the calorimeter is to provide an initial particle direction, which is then passed to the STK reconstruction algorithms, e.g. as a seed of Kalman tracking algorithm. First, a centre-of-gravity position is evaluated for the calibrated energy deposits in each layer of calorimeter, along the z-axis (in the coordinate system of DAMPE, z-axis is the direction perpendicular to the STK layers, pointed from the STK to the calorimeter, see Fig 4). Then a set of points is formed in both xz and yz projections in the calorimeter, and are fitted with a straight line. This line gives a preliminary direction of a particle in DAMPE.

The track reconstruction in DAMPE is done as follows. First, the track seed is formed either with a blind-seed finding algorithm, which uses STK information only, or the seed is taken from the shower direction reconstructed in the calorimeter. Depending on conditions, one or both seeding algorithms are used. As soon as track seed(s) are created, they are reconstructed (filtered) further using the adaptation of Kalman algorithm [7]. For the calorimeter-seed approach, the shower direction is projected onto the closest layer of the STK with the corresponding covariance matrix, either infinite, or the one evaluated from the calorimeter position and angular resolution as a function of energy. If the hit is found within a reasonable window around the projected position, a seed is formed and a track is reconstructed further using the Kalman filter. If the resulting track is of insufficient quality (the χ^2-test or the number of hits in the track does not fulfil the corresponding threshold values) the procedure is repeated with the second-closest layer of the STK, and if the sufficiently good track is not found yet – it is repeated with the third one. Then, if the track is finally retrieved, the whole procedure is repeated again with the first point of previous track being removed from the list of available points. Up to 10 such iterations are allowed. Finally, the same procedure is repeated with the 3 furthermost layers of STK in the opposite direction (towards calorimeter). Once a set of tracks is formed, ghost tracks are eliminated by looping over all tracks and removing those which are crossed by other tracks. If two tracks cross each other, the one which has lower number of hits or lower χ^2 is removed. The track "forks" in the direction towards calorimeter are

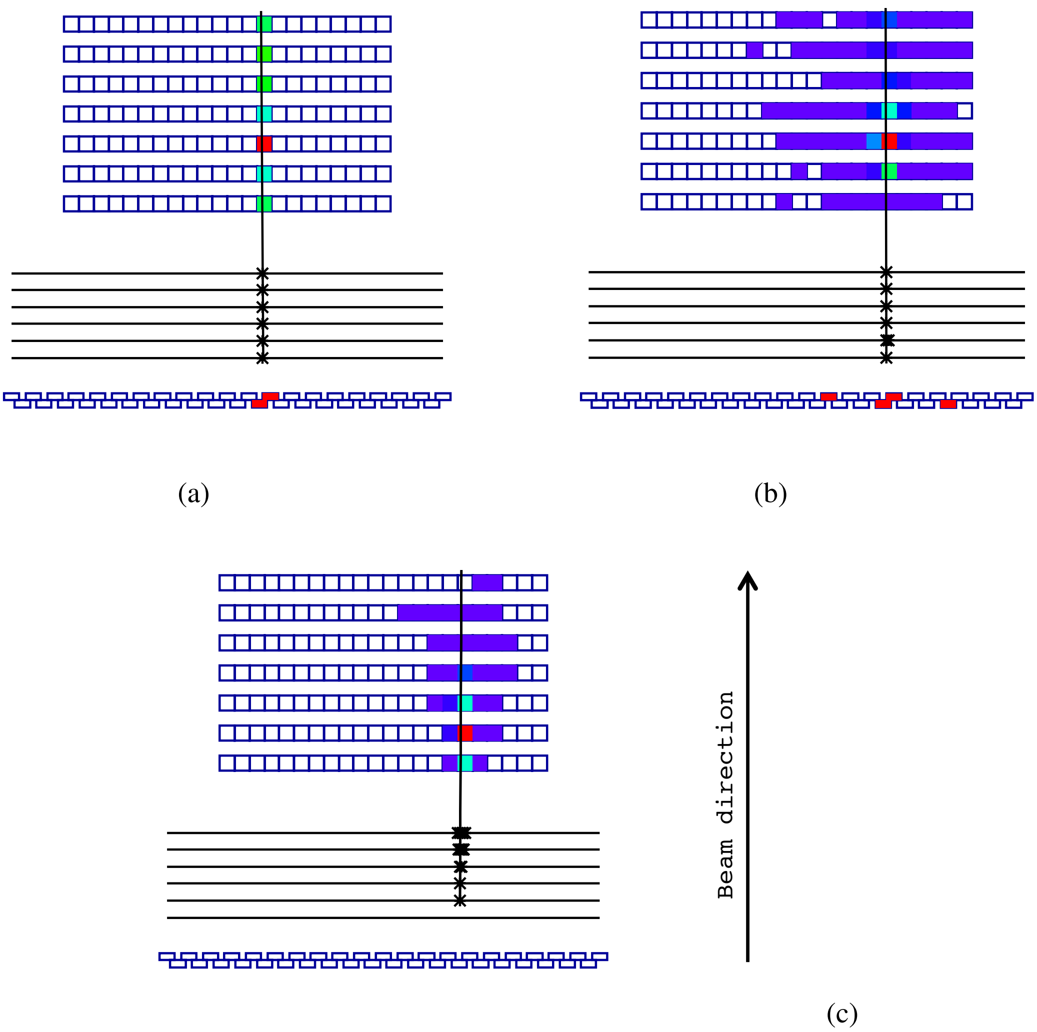

Figure 5: Typical display of reconstructed (a) muon (150 GeV), (b) electron (150 GeV) and (c) photon (10 GeV) event candidate in the DAMPE beam test at the SPS accelerator at CERN, 2015. Sensitive volumes are shown for the BGO calorimeter, STK and PSD respectively. Energy deposits in BGO are displayed using the coloured squares (each square corresponds to one BGO bar), where red ones correspond to bars with highest energy contribution. STK hits (clusters) are shown with black crosses, and reconstructed tracks are shown with black solid lines. The xz projection is used in all event displays.

allowed, while those which point in the opposite direction are considered as “crosses” and treated correspondingly. The procedure for the blind-seed algorithm is similar, with the only exception that the seeds are formed from a set of three x, y points forming a line. All combinations of three planes out of six in the STK are looped over for this algorithm.

Performance of the reconstruction algorithms was evaluated in the preliminary beam-test data analysis. The results on position and angular resolution are close to those estimated with the Monte-Carlo simulations [8], however more detailed studies are now underway. In particular, at the moment this paper being submitted, the alignment of the flight model of STK is performed with the comics data, using the track reconstruction algorithms described above. In Fig. 5 one can examine typical proton, electron and photon events as they appear in DAMPE after the reconstruction.

5. Conclusions

We present the status of offline software of the DAMPE satellite mission. This includes the framework for simulation, reconstruction, analysis and other types of jobs. The reconstruction algorithms have proven to work well, with performance similar to the one expected from Monte-Carlo simulation. DAMPE features a unique approach for the geometry implementation, with the software geometry model being obtained directly from the CAD drawings of the detector. The framework has been extensively exercised during the beam test data-taking campaigns and with the comics data, and the analysis of those data is now underway based on the framework.

References

[1] J. Chang, in *The 7th international workshop " the Dark Side of the Universe" (DSU)* 2011.

[2] B. Hegner, P. Mato and D. Piparo, *Evolving LHC data processing frameworks for efficient exploitation of new CPU architectures*, [`10.1109/NSSMIC.2012.6551463`].

[3] S. Agostinelli *et. al.*, *Geant4 – a simulation toolkit*, *Nucl. Instrum. Meth. A* **506, 3** (2003) 250. [`http://dx.doi.org/10.1016/S0168-9002(03)01368-8`].

[4] Christopher M Poole *et. al.*, *A CAD Interface for GEANT4*, [`physics.med-ph/1105.0963`].

[5] D. Abrahams, R. W. Grosse-Kunstleve, *Building Hybrid Systems with Boost.Python*, *Boost Consulting*, 2003, [`http://www.boost.org/doc/libs/1_31_0/libs/python/doc/PyConDC_2003/bpl.html`].

[6] S. Knight, *SCons Design and Implementation*, [`http://legacy.python.org/workshops/2002-02/papers/16/index.htm`].

[7] R. Fruhwirth, *Application of Kalman filtering to track and vertex fitting*, *Nucl. Instrum. Meth. A* **262** (1987) 444. [`DOI: 10.1016/0168-9002(87)90887-4`].

[8] V. Gallo *et. al.*, *The test results of the Silicon Tungsten Tracker of DAMPE*, in proceeding of 34*th* ICRC, **1199** (2015).