

## A new method of The Monte Carlo Simulation based on Hit stream for the LHAASO

---

**Hanrong Wu\***

*Institute of High Energy Physics, CAS, Beijing, China.*

*E-mail: wuhr@ihep.ac.cn*

**Mingjun Chen**

*Institute of High Energy Physics, CAS, Beijing, China.*

*E-mail: mjchen@ihep.ac.cn*

**Zhiguo Yao**

*Institute of High Energy Physics, CAS, Beijing, China.*

*E-mail: yaozg@ihep.ac.cn*

*for LHAASO collaboration*

The Large High Altitude Air Shower Observatory (LHAASO) will be constructed at Mt. Haizi in Sichuan Province, China. Among several detector components of the LHAASO, the Water Cherenkov Detector Array (WCDA) is of great importance for low-to-middle energy gamma ray physics. The WCDA has an area of  $78,000 m^2$  in total, which is sub-divided into 3120 cells by curtains, with a PMT resided in each cell. Old simulation program for WCDA is based on event stream which is very popular, but produce huge memory consuming for large events, so this old method is probably not suitable for treatment of events piling-up like WCDA simulation. A new method of the simulation based on Hit stream is developed for the LHAASO-WCDA, which not only can resolve the problem of memory consuming up but also can do union-simulation for all the LHAASO detector array.

*35th International Cosmic Ray Conference ICRC2017*

*10-20 July, 2017*

*Bexco, Busan, Korea*

---

\*Speaker.

## 1. Introduction

Geant4 is a toolkit of detector simulation used in many high energy physics and high energy cosmic ray experiments. In the context of the Monte Carlo (MC) simulation of particle transport through a detector setup, control of data flow is important which can make simulation program efficiently run especially for very high energy cosmic ray events with large number of particles. Many detector simulation programs, like collider experiments CMS[1], are based on event stream which is very popular. In this method, an event forms a stream, flowing from the top to bottom, till the end of one event, a new event just starts, and at the top of the stream, as a generator, all particles in one event are stored in buffer memory, which will use mass memory. For the simulation of high energy cosmic ray event, like ARGO, LHAASO etc., there are millions of secondary particles and even more, which will be out of memory, and make the running program corrupt. So simulation based on event stream may probably be not suitable for treatment of events piling up, then a new method of the Monte Carlo simulation based on Hit stream is proposed. In this draft, firstly high energy cosmic ray event will be introduced. Secondly the new method of simulation based on hit stream will be explained and implementation of the method. At lastly, an example with validation will be given.

## 2. High energy cosmic ray event

Energy range of primary cosmic ray event simulation by LHAASO cover from 100 GeV to PeV, an event with energy of 100 TeV. The characteristic of high energy cosmic ray simulation is as following: High energy big shower events. The highest energy of cosmic ray discovered at present is up to  $10^{20}$  eV. One primary cosmic ray event with energy of 100 TeV includes secondary particles with number up to order  $10^6$ . Huge number of hits. A high energy air shower event can produce huge number of hits, and even a high energy secondary particle (like a hadron which produces cascade) also can produce huge number of hits occasionally falling into the detector.

## 3. Data Model

### 3.1 old fashion: event stream

Event stream named as Event Data Model (EDM) in collider experiments such as CMS is centered around the concept of an Event. The system requires that all data is only accessed through the Event. A data processing job is composed of a series of algorithms (e.g., a track finder or track fitter) that run in a particular order. The algorithms only communicate via data stored in the Event. So if there are huge particles or hits in an Event like high energy cosmic ray event, it will occupy huge memory and not suitable for suitable for treatment of events piling-up.

### 3.2 new method: hits stream

The new data model based on hit stream has been proposed, in this system, hits form a stream, flowing from the top to the bottom. There is no run and event concept. The new system is governed by several design goals:

solving the problem of memory depletion. There are following aspects that will bring memory cost: as a generator, plenty of number of particles from a high energy air shower event when they are in buffer at the same time will occupy huge memory; hits collection as the product of job processing, huge number of hits for a high energy air shower event, even, hits of a high energy secondary particle (e.g., a hadron) occasionally falling into the detector can use up the memory; Detailed implementation of every cells (there are many copies of the detector cells/units in one detector array) may bring additional memory cost.

Optimizing middle results store. As a generator, Air shower data such as CORSIKA[2] output are not necessary to be fully kept as many detectors are sparsely placed; Positions of particles relative to a detector cell recorded may shrink the data size with the same precision especially when a compression algorithm is applied; And more information on hits can be kept. More information rather than only the hits themselves may do help for some particular analysis and the detector optimization; The reading and storing of the information on the run and the event are not necessarily be taken care by the program developer in every step.

Ease of realization of detector simulation. The system split the simulation into several steps, leave the some mutable processes relying more on the realization in a later-on dedicated program and put the solid implementation depending little on detector realization but time-consuming in the first step, so that saving the CPU time.

Simplifying unified simulation of several kinds of detectors. In this system, the detector configuration, such as the geometrical position and the skeleton structure can be simulated in the first step, and the detailed detector simulations can be carried out in the second step with only a few detector cells (e.g., a few detectors per detector kind) residing in a simple box.

Parallel processing data. There is no event concept, and one event can be separated several batches which can be processed in different CPU at the same time.

## 4. Implementation

### 4.1 The Simulation Data Flow

Figure1 shows a simplified view of the processing stages in the simulation data flow. Input for simulation comes from event generators after CORSIKA simulation output which is named generator hits (Ghits). Data objects representing particles hitting a detector unit/cell delivered in batches, obtained by a simple tracking or geant4 simulation without real sensitive detectors is named cell hits (Chits).

Data objects representing Monte Carlo truth information from Chits by sensitive detector simulation is named detector hits (Dhits). After detector realization like adding-on with noise etc., Final hits (Fhits) can be gotten, and these hits can be used to do data reconstruction. These hits form a stream, flowing from the top to the bottom. Any two kind of consequential hits are connected by a "repeater", comprising of storage, processors and switches. In the GEANT4 nomenclature, actually Ghits and Chits are two steps of the particle generation; Dhits and Fhits are two steps of particle simulation. There is no run and event concept, run/event start, run/event end are just special hits, merged into the hits stream, to mark the separation of these hits; Hits are delivered in batches through the stream. The size of the batch can be controlled, but forced to be separated in the event gaps, and detector gaps (once the detector making sense).

## 4.2 Processing data

All hits are represented by int or float variables in different c++ classes, whose carries are by the means of `std::vector<class name>`, and stored in the repeater with root trees, which are well-designed with buffering and accessing model. The system defines a class HitsReader as interface code which can read different Hits information like Ghits, Chits or Dhits from user. In addition, a program converting CORSIKA output to root and a program sorting hits according to detector ID is implemented. The switches of the system are user codes, which can decide where to go by judging particle position and/or detector ID, and the entry positions of hits of all events are indexed and stored in a tree, enabling randomly accessing hits of any event.

## 5. Hit stream validation

The validation of the new data model based on hits stream was done on LHAASO-WCDA simulation. The WCDA has an area of 78,000 m<sup>2</sup> in total, which is sub-divided into 3120 cells by curtains, with a PMT resided in each cell. Firstly, Ghits from CORSIKA output as an input delivered in batches are read by simulation. A batch of hit is showed in the following program segment in Fig2. In the simulation of this step, PMT simulation was not done, a box including the PMT was selected as a sensitive detector recording the hit information saved as a Chit like position, momentum, time, detector ID, etc.. Secondly, Chits as an input for PMT sensitive detector simulation, in this step, only one detector cell was simulated and Dhits are output. And the realization of detector simulation can be done with Dhits and then data reconstruction. The simulation results using two different programs (event stream and hit stream) are shower in the following pictures. The energy of primary events are 2 TeV for comparison. From the Figure2, we can conclude that the results show a very good consistency from two programs.

the program based on event stream will break down because of memory depletion. But the new program based on hit stream run well, and memory depletion is resolved.

This system define 4 kinds of Hits: Ghits, Chits, Dhits, Fhits, all hits can store data information according to user's requirement and do some particular analysis and the detector optimization. This system split the simulation into two steps, in the first step, a particle is tracked to the box including PMT and water absorption is not considered, but about 95% or more time is spent on this step, and in the second step, PMT simulation and water absorption were simulated detailed. The results considering water absorption are shown in the Figure3. Results are again quite similar. The same as done with PMT optimization or other realization. So we can save much CPU time.

LHAASO has three kinds of detector array. In order to take full advantage of different detector array, unified observation is essential. This system provide possibilities to unified simulation of different kinds of array. Skeleton simulation can be done in the first step, and detailed simulation is done in the second step.

There is no event concept in this system, so one event can be assigned to several CPUs at the same time.

## 6. Conclusions

The new method of Monte Carlo simulation based on hit stream is developed. The problem

of memory depletion is resolved and make unified simulation be possible. This method can optimize data store of middle results and save more data information for analysis. The system makes realization of detector simulation more easily and makes parallel processing data be possible, which can save more CPU computing time.

## **7. Acknowledgements**

This work is supported by National Natural Science Foundation (NSFC) of China under contacts No. 11635011.

## **References**

- [1] C.D.Jones, M.Patern, J.Kowalkowski, L. Sexton-Kennedy, et al.,  
<https://indico.cern.ch/event/408139/contributions/979800/attachments/815724/1117731/FrameworkPaper.pdf>.
- [2] D. Heck et al., Report FZKA 6019 (1998).

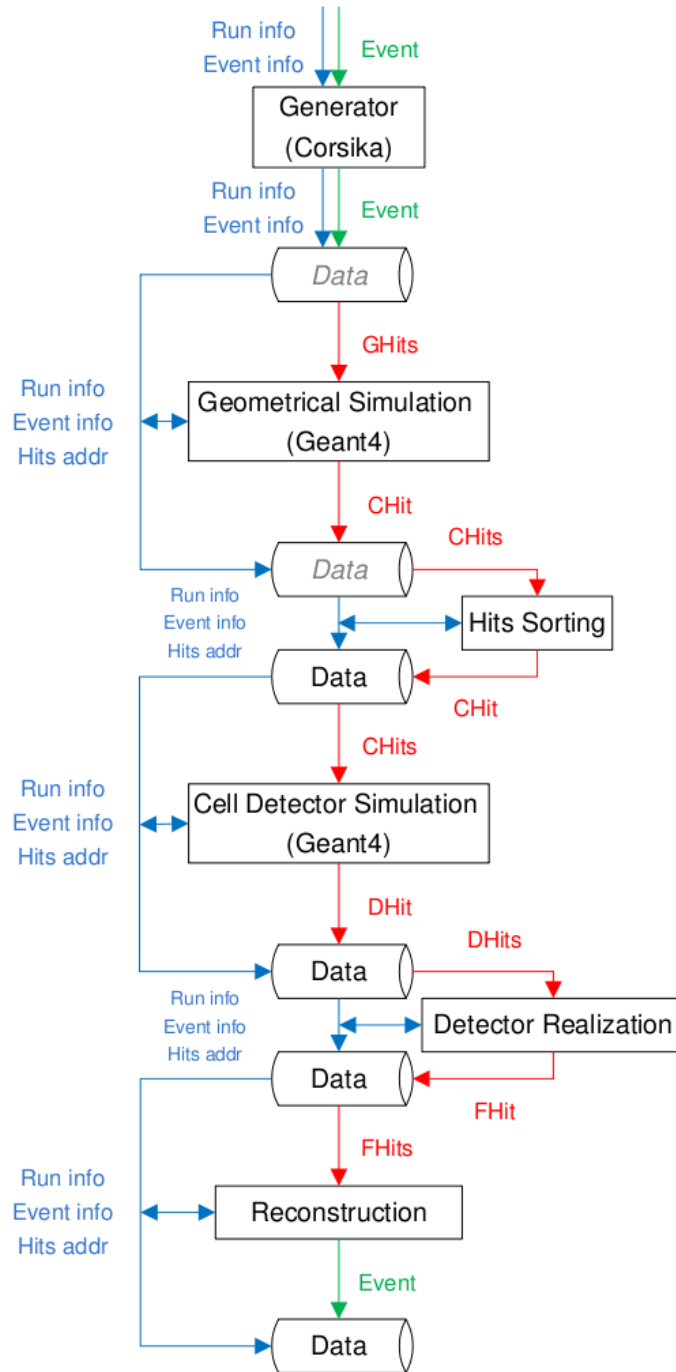
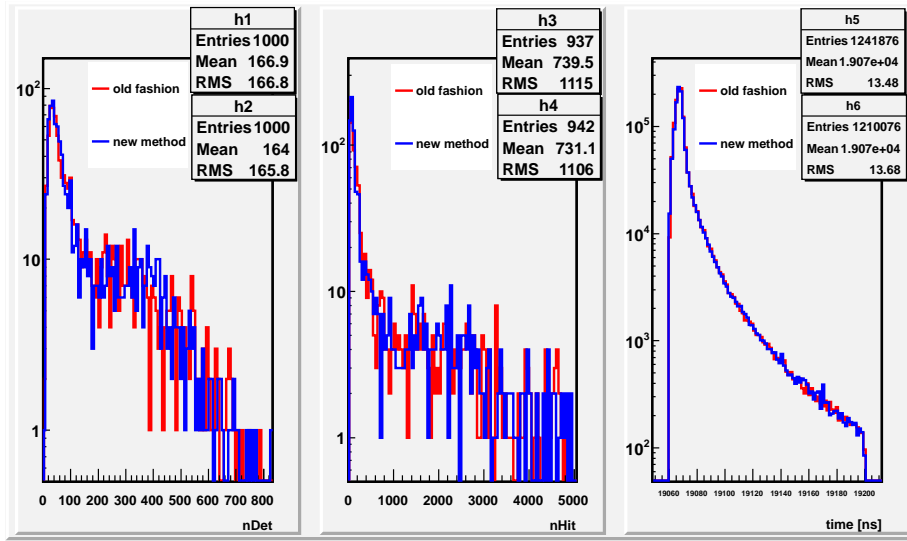
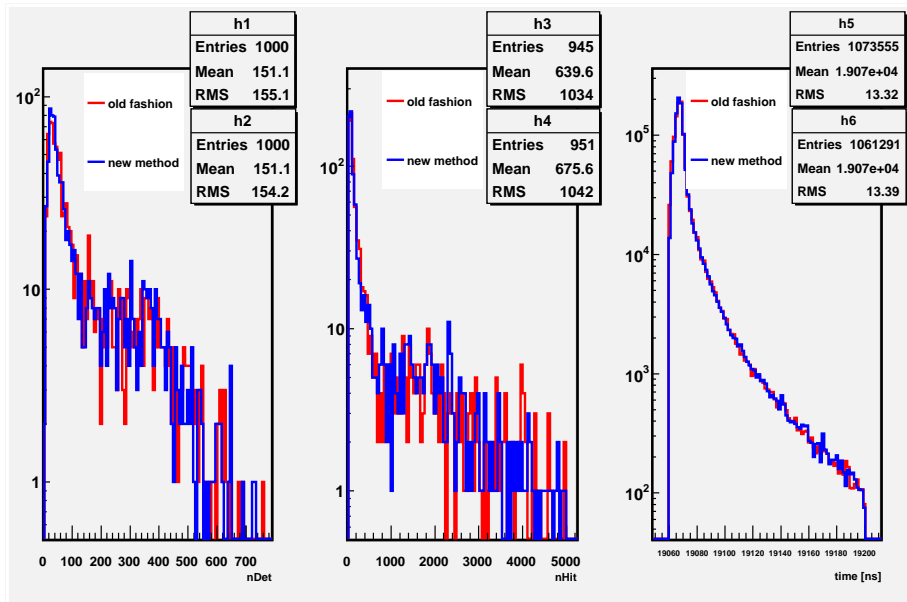


Figure 1: Data flow of new method.



**Figure 2:** Distribution of simulation without considering water absorption from two programs, red for old one and blue for new one. Left, PMT number distribution; Middle, Hit number distribution; Right, time distribution.



**Figure 3:** Distribution of simulation considering water absorption from two programs, red for old one and blue for new one. Left, PMT number distribution; Middle, Hit number distribution; Right, time distribution.