

# MANAGEMENT OF THE MONTE CARLO SIMULATION PROCESS IN THE MPD EXPERIMENT ON THE NICA CLUSTER\*

ŁUKASZ SAWICKI<sup>†</sup>, JAKUB ZIELIŃSKI

Faculty of Physics, Warsaw University of Technology, Warsaw, Poland

KRZYSZTOF DYGNAROWICZ

Faculty of Electronics and Information Technology  
Warsaw University of Technology, Warsaw, Poland

*(Received February 1, 2021)*

This paper shows the development of software aimed at automatization of the process of Monte Carlo simulation of event, and particle generation and propagation in Au–Au collisions in the MPD detector. It contains information about the tools used and a description of the software functioning.

DOI:10.5506/APhysPolBSupp.14.617

## 1. Introduction

During an internship at the Joint Institute for Nuclear Research [1] at the Laboratory of High Energy Physics, we created a software for the need of management of the generated data. These data were simulations of heavy-ion collisions inside the MPD (Multi-Purpose Detector) located at the NICA [2] (Nuclotron-based Ion Collider fAcility).

The major problem during the practice was to make the process of event generation and simulation of particles as automated as possible. We were running simulations and event generations for Au–Au collision in the NICA collider range of energy. It meant having to solve problems with marking, storing and using all the generated data.

This problem can be divided into four separate steps. Firstly, event generation, secondly event reconstruction using macros, then checking the correctness of received files and future data analysis.

---

\* Presented at NICA Days 2019 and IV MPD Collaboration Meeting, Warsaw, Poland, October 21–25, 2019.

<sup>†</sup> Corresponding author: [1.saw99@outlook.com](mailto:1.saw99@outlook.com)

## 2. Software

For this purpose, we had a NICA computing cluster with about 7 PB disc space to use at our disposal. Responsible for event generation was `UrQMD 3.4` [3] (Ultra-relativistic Quantum Molecular Dynamics) Monte Carlo simulation package. By using it, we generated most of our gathered data. In order to compare simulation models, we generated a small amount of data by using the `vHLLC` package (A 3 + 1 dimensional viscous hydrodynamic code for relativistic heavy-ion collisions). Event generations were executed for Au–Au collision at the energy from 4 GeV to 11 GeV. On the cluster, we had `MPDRoot` framework available containing MPD detector models. By using this framework, we were able to simulate the transition of particles created in the collision through the detector and we were able to see detector’s response to this event.

### 2.1. Software operating principle

For the purpose of realization of each step, we made a few `Bash` scripts and programs written in `C++`. The NICA cluster shares a folder named “data”, its size is about 7 PB without a quota limit. Thanks to the access to this folder, we were able to easily collect data coming from the event generation and detector response into one place. The full path to this folder is `/eos/nica/mpd/data`. In this location, every simulation model and each energy have their separate folder. Simulations for the `UrQMD` model were carried out for energies at the center-of-mass frame  $\sqrt{s_{NN}} = 4$  GeV, 7 GeV, 9 GeV and 11 GeV. For the `vHLLC` model, we were making simulations only for 7.7 GeV collisions, due to it being the only possible energy from NICA collider’s energy range we can choose from. The event generation and detector response simulation process were working according to this algorithm:

- In program’s `config` file set a number of events in one generation.
- Run program with an argument of energy and number of jobs to run (in the `vHLLC` model case we do not pass energy as an argument, because it is set in the `config` file).
- Program makes in the data folder a new folder with set energy and simulation model (if this name already exists, this step is skipped).
- Inside that folder, for every running job, the program is creating a folder with the name of running job. It is a unique `JOBID` number given by the queue system working on the cluster.
- Necessary macros and folders are placed in the created folder.
- Program runs event generation.

- Output file from every generation is passing through `runMC.C` macro which is simulating detector's response to a generated event and through `reco.C` macro which can reconstruct the event from information given by the previous macro. The final file ready for analysis is `mpddst.root`

The user can only set energy and the number of events for one simulation in the `config` file. The rest of the parameters is the same in every simulation. The user can modify the rest of the parameters by applying changes to the source code. However, it is then necessary to change names of folders, within the code. This is because it was our first attempt to create a program for running simulations in one command. We used a lot of the same parameters on purpose in order to collect as much data as possible for analysis. The source code and installation instruction are available on GitHub. The queue system on NICA cluster allows to run 200 jobs parallelly per user, which limited the possibility of producing data.

### *2.2. Data correctness checking*

The next stage is checking the correctness of generated files. Due to the complexity of the content of each file, file validation has been limited to checking if a file with the correct name exists and if its size on the disk is large enough. The correct size for every energy and number of events in one simulation is calculated by checking the size of a few generated files with various energies and the number of events in one simulation. Based on this, we can determine the right size of each file. The program knows the number of events in each simulation, because it is set in the `config` file (inputfile). It turns out that the file size is practically constant, and damaged files are at least a few percent smaller than the correct size. Following this, the program is able to determine which simulations ended correctly, and in the case of incorrect ones, it is able to determine in which place the error occurred. Paths to valid files are placed in a `txt` file, which is used in further collision analysis.

## **3. Summary**

During the 2-month internships, most of the time was devoted to learning how to use the cluster. First weeks were spent on writing the program and fixing its errors. The last 3 weeks of practice were devoted to data generation and analysis. During this time, we successfully reconstructed and analyzed 2,120,000 events. Together they had a size of 60 TB. The numbers of the reconstructed events for each energy are shown in Table I.

TABLE I

Number of reconstructed events for each energy.

UrQMD				vHLE
4 GeV	7 GeV	9 GeV	11 GeV	7.7 GeV
Reconstructed events				
203600	448346	593851	721652	151393

A typical job is 200 events for one simulation, this value was chosen because, by trial and error, it was noticed that a larger number of events results in more simulations ending with errors, and a smaller number of events reduces the amount of data produced during the day. 200 events were accepted as the most optimal number. Generation of so many events in one job and their reconstruction took about 24 h. Over 2 million events in the cluster were generated in 3 weeks. Using your own laptop for this would take a little over 10 years.

All data and images come from the presentation given at the conference. You can download the software from GitHub <https://github.com/Lukaszz99/work>

## REFERENCES

- [1] JINR, <http://www.jinr.ru/about-en/>, accessed: 28-08-2019.
- [2] NICA Collaboration (A. Sissakian, A.S. Sorin), *J. Phys. G: Nucl. Part. Phys.* **36**, 064069 (2009).
- [3] S.A. Bass *et al.*, *Prog. Part. Nucl. Phys.* **41**, 255 (1998).