

# Mock Data Challenge for the MPD/NICA Experiment on the HybriLIT Cluster

Konstantin Gertsenberger<sup>1,\*</sup> and Oleg Rogachevsky<sup>1,2</sup>

<sup>1</sup>Laboratory of High Energy Physics, Joint Institute for Nuclear Research, Russia, Dubna

<sup>2</sup>Dubna State University, Russia, Dubna

**Abstract.** Simulation of data processing before receiving first experimental data is an important issue in high-energy physics experiments. This article presents the current Event Data Model and the Mock Data Challenge for the MPD experiment at the NICA accelerator complex which uses ongoing simulation studies to exercise in a stress-testing the distributed computing infrastructure and experiment software in the full production environment from simulated data through the physical analysis.

## 1 Introduction

According to the program of the Joint Institute for Nuclear Research on the creation of the ion accelerator complex for the range of collision energy  $\sqrt{s_{NN}}$  from 4 to 11 GeV, the Nuclotron-based ion collider facility (NICA) is being constructed. Two interaction points are foreseen at NICA for two detectors. The MultiPurpose Detector (MPD) [1] located at one of the two interaction points is optimized for a comprehensive study of the properties of hot and dense nuclear matter in heavy-ion collisions and search for the critical point of the phase transition to the quark–gluon plasma. The MPD experiment would be capable of unique measurements of heavy-ion collisions including high-precision tracking and particle identification as well as of very accurate event characterization.

The software for simulation, reconstruction and analysis of particle physics data is an essential part of each high-energy physics experiment. To support the MPD experiment, the MpdRoot [2] framework is under development. It is based on the ROOT environment [3] and the FairRoot framework [4] of the GSI Institute and serves for simulation of the MPD detector, the reconstruction of simulated and experimental data and the following physical analysis of particle collisions registered by the MPD detector. The MPD software covers all stages, such as the simulation process of the particle interactions with media and detector materials, digitization – translating the interactions with detectors into clusters of signals, reconstruction of events and physical data analysis.

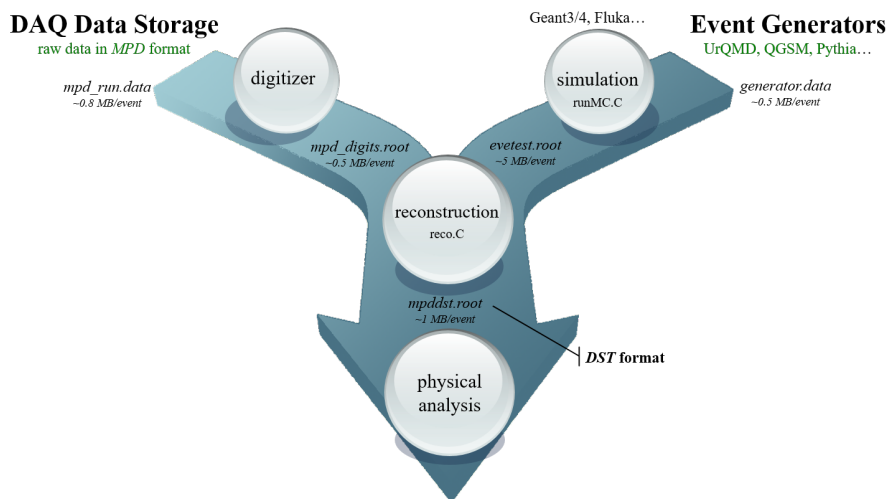
The Mock Data Challenge (MDC) is performed to simulate the offline processing of the data produced by the MPD detector before obtaining the first experimental data. Its main purpose is to exercise in a stress testing the distributed computing infrastructure and experiment software in the full production environment from simulated data through a physics analysis. The following sections of the article give an overview of how the MPD event data model is constructed and event data are processed in a distributed manner, and show the first mock data challenge for the MPD experiment.

---

\*e-mail: gertsen@jinr.ru

## 2 Event Data Model in the MPD experiment

Fig. 1 provides an overview of the experimental data processing chain in the MpdRoot software. Raw data in a special *MPD* format using the TLV (type-length-value) architecture from the DAQ system are digitized and converted into ROOT format by an event builder. A reconstruction macro restores the information about the particles registered by the detectors, their tracks and other parameters. The target size of a reconstructed file is about 1 MB per central Au–Au event. Then physics analysis of the reconstructed data is performed to investigate the physical properties of the nuclear matter produced in the collisions. For the experimental data it is planned to use the customary two formats in the high-energy experiments: event summary data (ESD) containing the detailed output of the reconstruction, and analysis object data (AOD) with sufficient information for physics analyses. The



**Figure 1.** MpdRoot offline data processing chain

processing chain with simulation data containing the full information about the particles obtained by the event generators is used at this stage. After simulating the passage of the particles through the detectors using particle transport packages, the data are converted to detector responses. The MpdRoot simulation macro uses Geant 3 (or 4) as a particle transport package. The event data are stored in ROOT files in a hierarchical tree structure. The size of an event resulting from heavy-ion collisions is directly proportional to the multiplicity, which is a function of the interaction centrality. Furthermore, the amount of data taken in experimental runs varies with the trigger conditions. The approximate event sizes estimated in MpdRoot and shown in Fig. 1 correspond to central heavy ion collisions.

For the offline processing of the MPD data, a Unified Database [5] was developed on PostgreSQL database management system. In addition to other data, it also stores information on simulated files obtained by different event generators. We use a wide list of event generators with proper physics effects, for example, UrQMD and QGSM to provide realistic particle distributions or Hybrid UrQMD and Three Fluid Hydrodynamics to predict new phenomena in nuclear collisions. At present, the data storage contains about 34,000 simulated files of 1.2 TB. These initial simulation data were used for the Mock Data Challenge of the MPD experiment.

### 3 Distributed MPD event processing

It is assumed that the MPD experiment with an interaction rate up to 7 kHz will produce about twenty billion events per year. An event of a central Au + Au collision at the NICA energies contains up to 1,000 charged particles. The high interaction rate, the high particle multiplicity and the long sequential processing time of millions of events in MpdRoot are the main reasons to develop a distributed computing system for the MPD experiment.

There are two main approaches to the distributed MPD event processing now: the Parallel ROOT Facility (PROOF) tool of the ROOT environment to parallelize the processing of event trees in ROOT macros on parallel architectures and the MPD-Scheduler developed for user task distribution on cluster nodes.

The PROOF uses data independent parallelism following from the lack of event correlations to process different MPD events in parallel. The PROOF support was added to the MpdRoot software and the reconstruction code was rewritten according to the PROOF rules and classes.

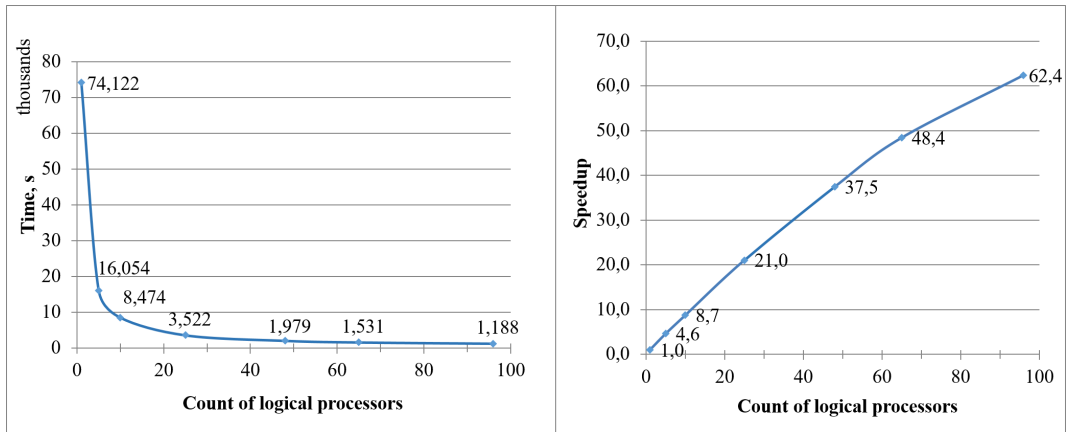
The MPD scheduling system was developed to distribute user jobs among computing clusters. MPD-Scheduler was implemented in C++ language with the ROOT classes support. It simplifies the parallel execution of user macros without asking for the knowledge of different batch systems. At present, it supports SLURM, Sun Grid Engine and Torque scheduling systems to run user jobs on cluster nodes in parallel and can use the Unified Database of the experiment. Jobs for multi-threading execution on one multi-core machine or distributed execution on clusters are described and passed to the MPD-Scheduler as XML files. To execute a user job on the cluster, the MPD-Scheduler parses the job description and runs scripts by the given batch system. It defines free workers of the cluster and performs data processing in parallel. To perform the Mock Data Challenge for the MPD experiment on the HybriLIT cluster, the MPD-Scheduler was used.

### 4 MPD Mock Data Challenge on the HybriLIT cluster

The Mock Data Challenge aims at providing large-scale simulation production, exercising the full spectrum of the experiment software and hardware from simulation through to physics analysis. It is good for stress-testing of the distributed computing infrastructure of the experiment, identifies potential issues before first data, helps to estimate offline computing requirements, checks data quality and prepares quick turnaround from data taking to publication.

The Mock Data Challenge was performed on the HybriLIT cluster [6]. The HybriLIT heterogeneous cluster is a computation component of the Multifunctional Information and Computing Center of the Laboratory of Information Technologies of JINR, which contains a multi-core component and computation accelerators: NVIDIA graphic processors and Intel Xeon Phi co-processors. It is designed for parallel computations asked by a wide range of tasks arising in the scientific and applied research conducted by JINR. The HybriLIT cluster stores the MPD data on the distributed file system NFS and uses SLURM to distribute jobs among the cluster nodes. All external packages for MpdRoot and MpdRoot itself were installed and configured on the cluster for parallel event data processing. The distributed HybriLIT NFS storage contains files with 3 million simulated events obtained by the LAQGSM generator for the MPD experiment.

An XML description of the job used for the MPD mock data processing was created for the MPD-Scheduler, which distributes the processing of the files among cluster nodes by a SLURM batch system. The job includes dependent tasks of simulation, reconstruction and physics analysis executed at the same time on the HybriLIT cluster. Femtoscopy was chosen as a physics analysis task for reconstructed data. The list of input files consists of three hundred twenty simulated files for the MPD collision energies from 5 to 11 GeV. Fig. 2 presents the time and speedup of the MPD mock data processing in terms of the number of processor cores used.



**Figure 2.** The dependence of the time (left graph) and the speedup (right graph) of the Mock Data Challenge on the number of processor cores

As a result, the MPD Mock Data Challenge was successfully performed on the HybriLIT cluster. All of the steps were successfully completed, and software and hardware errors did not occur during the execution. Now the HybriLIT storage includes four versions of the reconstructed data for the generated events, depending on the simulation settings.

## 5 Conclusions

The MpdRoot environment with required packages (Fairsoft, ROOT/PROOF, MpdRoot, the MPD-Scheduler and others) was deployed on the HybriLIT cluster for MPD data processing. Two methods were implemented to process event data of the MPD experiment on distributed systems in parallel: using the PROOF system and the MPD-Scheduler. The MPD-Scheduler tool was developed to automate running MpdRoot macros concurrently and it was used in the MPD Mock Data Challenge (*simulation-reconstruction-analysis* chain) to test the software and computing infrastructure of the HybriLIT cluster. Practical values of time and speedup for MPD event processing were obtained.

## Acknowledgement

The authors would like to thank the HybriLIT team for the opportunity to use the computational resources of the HybriLIT cluster and support.

## References

- [1] MPD Collaboration, *The MultiPurpose Detector – MPD. Conceptual Design Report* (JINR, Dubna, Russia, 2012) 259 pp.
- [2] K. Gertsenberger, S. Merts, O. Rogachevsky, and A. Zinchenko, *European Physical Journal A* **52**, 8, p. 5 (2016)
- [3] CERN, *ROOT – An Object-Oriented Data Analysis Framework* (CERN, 2013) 628 pp.
- [4] M. Al-Turany, D. Bertini, T. Karabowicz, et al., *Journal of Physics : Conference Series* **396**, 022001 (2012)
- [5] K. Gertsenberger, *Physics of Particles and Nuclei Letters* **13**, 5, 634–639 (2016)
- [6] HybriLIT, *Heterogeneous computing cluster HybriLIT of LIT JINR* (JINR, Dubna, Russia, 2016)