CURRENT WORKFLOW EXECUTION USING JOB SCHEDULING FOR THE NICA EXPERIMENTS

K.V. Gertsenberger [a], O.V. Rogachevsky

Laboratory of High Energy Physics, Joint Institute for Nuclear Research, 6 Joliot-Curie, Dubna, Moscow region, 141980, Russia

E-mail: [a] gertsen@jinr.ru

Simulated and experimental data processing is an important issue in modern high-energy physics experiments. High interaction rate and particle multiplicity in addition to the long sequential processing time of million events are the main reasons to parallelize event processing on distributed computing systems for the NICA experiments. The paper presents one of the directions of distributed data processing: job scheduling for user task distribution on computing clusters. The software and hardware environments being used for the current workflow execution are briefly noted. The current MPD-Scheduler system developed to simplify parallel execution of user ROOT macros for simulation, reconstruction and data analysis is described in details. The practical values of the speedup for simulated event processing in the MPD experiment are shown. The possible workflow management systems being under discussion for the NICA experiments are also noted.

Keywords: NICA collider, MPD experiment, BM@N, event data model, physics data processing, MPD-scheduler, job scheduling, distributed computing, batch systems

1. Software for the MPD and BM@N experiments of the NICA complex

The research program on heavy-ion collisions at the Nuclotron of the Joint Institute for Nuclear Research (JINR) in Dubna includes the following topics: study of the properties of nuclear matter in the region of the maximum baryonic density, investigation of the reaction dynamics and nuclear equations of state (EOS), study of the in-medium properties of hadrons, production of (multi)-strange hyperons at the threshold and search for hyper-nuclei. According to the program, the Nuclotron-based ion collider facility (NICA) [1] is being constructed as an ion accelerator complex to collide particles in the atomic mass range A=1–197 at a centre-of-mass energy up to 11 GeV for Au79+ and up to 27 GeV for protons.

Two interaction points are foreseen at the storage rings of the NICA collider for two detectors. The Multipurpose Detector (MPD) 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 Spin Physics Detector (SPD) is proposed for the realization of the spin physics program at NICA and focused on investigating the nucleon spin structure with high intensity polarized light nuclear beams. One of the main elements of the NICA first stage is a fixed target experiment with Nuclotron extracted beams – BM@N (Baryonic Matter at Nuclotron), whose technical runs are performed from spring 2015. The research program will be continued at higher energies with the MPD setup after putting the startup configuration of the NICA collider into operation in 2020. The commissioning of the design configuration of the NICA accelerator complex is foreseen in 2023.

The software for simulation, reconstruction and analysis of particle physics data is an essential part of each high-energy physics experiment. It should cover all stages, such as the simulation process of the particle interactions with media and detector materials, digitization – translating the interactions with the detectors into clusters of signals, reconstruction of the events and physics data analysis. The software and computing parts of experiments are responsible for the activities including design, evaluation and calibration of detectors; storing, access, reconstruction and analysis of the data; and development and maintenance of a distributed computing infrastructure for physicists engaged in these tasks. To support the BM@N and MPD experiments of the NICA complex, the BmnRoot and MpdRoot software are implemented in the programming language C++ and based on the ROOT [2] environment. To avoid developing the experiment software from scratch, the frameworks are built on the FairRoot [3] environment of the FAIR facility in the GSI Institute.

The BmnRoot and MpdRoot software provide a powerful tool for detector performance studies, event simulation, reconstruction of experimental or simulated data and following physics analysis of particle collisions registered by the BM@N and MPD detectors, respectively. MpdRoot and BmnRoot use a base part – the hierarchy of FairRoot classes to simplify simulation and data processing, for example, classes for describing detector geometries and magnetic field, task manager classes for data processing chains and for defining separate tasks. To investigate the feasibilities of the detectors for physics data analysis, a wide range of event generators are used with corresponding physics effects. The flexibility of the frameworks is gained through its modularity. The physics and detector parts could be written by many different groups. Using the same internal structure users can compare easily the real data with the simulation results at any time. An overview of event data processing via the BmnRoot and MpdRoot frameworks is presented in the next section.

2. Workflow Execution in the NICA frameworks

The sequence of the main stages of event data processing in both BmnRoot and MpdRoot software is shown in Figure 1. Raw experimental data in a special binary format from a Data Acquisition System are digitized and converted into the ROOT format by a digitizer macro. The next step is the reconstruction of particle data, tracks and other parameters that are written to the DST file. The reconstruction algorithms restore the information on produced particles, their momenta, types, trajectories and other kinematic features from the information contained in the raw detector data. The last step of the event data processing is the physics analysis of the reconstructed data.

Before the experiment starts, the processing chain with simulation data containing the full information on the particles obtained by the event generators, such as UrQMD, QGSM is used. Then, the particle transport packages (usually, Geant 3 and 4) are used to transfer the particles through the detectors of the setup. While transporting particles, a detailed description of the detector geometries is used, and the tracks of all particles are passed through the medium and detector materials taking into account various physics effects and magnetic field. After simulating the passage of particles through the detectors, the data are converted into detector responses, and then the same steps are performed as for experimental data processing: event reconstruction and physics data analysis to evaluate the efficiency of the detectors or investigate the physics properties of nuclear matter. The experimental and Monte-Carlo data are stored in the ROOT files in a hierarchical tree view.

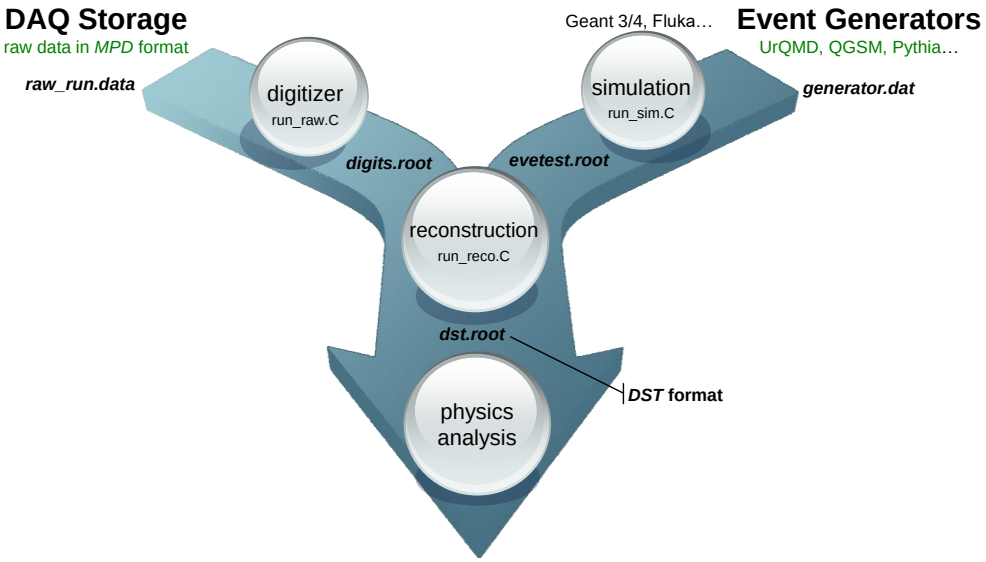

Figure 1. Data processing chains in the experiment frameworks

In central gold-gold collisions (with an impact parameter less than 3 fm) that are supposed to be used in the experiment, up to 1000 charged particles are produced at the NICA energies. In addition to the large multiplicity of events, it is necessary to take into account the high interaction rate (up to 7 kHz) in the future setup. The above conditions and the long sequential processing time of million events are the main reasons to use distributed computing systems for parallel processing of the NICA events. To parallelize the data processing on parallel architectures, various methods can be used. One of the directions for accelerating the processing of a large amount of data – a scheduling system for task distribution on distributed clusters is presented in the next section.

3. Current Job Execution at the NICA distributed platforms

For parallel data processing of the NICA events, different computing platforms are currently used at the Laboratory of Information Technologies (LIT) in JINR. A queue in the small part (200 processor cores) of the Tier-1 level centre in LIT was allocated to solve the task. The mass storage of Tier-1 is built on a dCache distributed file system; Torque/Maui is used as a scheduler. The core of the JINR computing infrastructure is the Central Information and Computer Complex (CICC) that possesses powerful computing tools, which, with the help of high-speed communication links, are integrated with information resources worldwide. A heterogeneous cluster “HybriLIT” [4] is a computing element of the CICC JINR, which allows developing parallel applications to solve a wide spectrum of mathematical resource-intensive problems using all opportunities of a multi-core component and computing accelerators: graphics processors NVidia and co-processors Intel Xeon Phi. The HybriLIT heterogeneous cluster containing 252 processor cores, EOS distributed file system and SLURM scheduler is also used for the NICA event processing.

Within the Session of the Committee of Plenipotentiaries of the Governments of the JINR Member States, presentation of the new supercomputer named after Nikolai Nikolaevich Govorun, who actively participated in the development of information technologies in JINR, was held in March 2017. The GOVORUN Supercomputer is a project aimed at sufficient acceleration of complex theoretical and experimental researches in the field of nuclear physics and condensed matter physics held at JINR including the NICA project. The supercomputer is based on the HybriLIT heterogeneous platform, and it has led to the increase of CPU and GPU performance. Currently, NICA data processing tasks are being actively executed on the Intel computing element of the supercomputer containing about 2 900 cores of Intel Xeon Gold processors and about 8 000 cores of Intel Xeon Phi co-processors. All required packages (FairSoft, FairRoot, MpdRoot, BmnRoot and others) are deployed and configured on the hardware platforms for distributed processing of the NICA events.

So, if users (for example, collaboration members) have many tasks or files to process, using these computing platforms can significantly speed up obtaining results. To distribute user jobs on clusters and run those in parallel, batch systems are used. The feature of the current computing platforms in JINR is that they provide different batch systems (Torque scheduler at the Tier-1 fragment, SLURM at the HybriLIT platform and Sun Grid Engine at the Laboratory of High Energy Physics). To solve this problem, at the current stage the MPD-Scheduler is developed in C++ language with the ROOT classes support as a submodule of the MpdRoot and BmnRoot frameworks to simplify the parallel execution of user tasks (ROOT macros) on cluster nodes without the knowledge of different batch systems. At present, it supports SLURM, Sun Grid Engine and Torque schedulers, and can work with the developed database of the experiments.

Jobs for multithreaded execution on a multicore machine or distributed execution on computing clusters are described and passed to the MPD-Scheduler as XML files (for example, to run in bash: $ *mpd-scheduler my_job.xml*). The XML description of a job starts with <*job*> and ends with the corresponding closing tag. The tag <*macro*> sets information about a ROOT macro being executed by the experiment software, and includes the following attributes: conventional name of the macro to use dependencies, macro path, global number of the start event and count of events to process for all input files, and additional arguments of the ROOT macro.

The tag <*file*> defines files to be processed by the above macro. The user specifies lists of input and output files, events to be processed, parallel mode, and whether resulting files will be merged. In addition to input file paths with possible regular expressions, the user can choose different file sources: text files containing a list of input files separated by new lines, output files of the previous macros or jobs, and a list of simulation or experimental files obtained from the experiment database according to the given criteria.

The tag <*run*> describes run parameters and allocated resources for parallel jobs, such as multithreaded execution on a local machine or distributed processing on a cluster, maximum count of processors allocated for the job, configuration and log files, job priority, selected queue and host names. Moreover, the MPD-Scheduler can execute not only ROOT macros but also arbitrary commands on remote nodes.

To execute a user job on the cluster, the MPD-Scheduler parses the job description and runs scripts via the batch system installed on the cluster. The latter defines free worker nodes and performs the data processing in parallel. When a worker node finishes its part, the state of the worker is changed to the free value and it can obtain another user job. The MPD-Scheduler also has the possibility to merge result files in the mode of partial file processing.

In general, an XML description for the MPD-Scheduler can contain more than one job. In this case, <*job*> tags are included in the common <*jobs*> tag, and dependencies can be set between the jobs, so that a job depending on another one will not start its execution until the latter ends. To set dependency between two jobs, the user can set the special job attribute ("*dependency*") assigned the name of another job.

To test the MPD data processing workflow, an XML description with three jobs was created for the MPD-Scheduler. It includes dependent tasks of MPD event simulation, reconstruction and physics analysis executed at the same time on the cluster. Femtoscopy was chosen as a physics analysis task of the chain. This test was performed on the GOVORUN Supercomputer, which stores MPD data on an EOS file system and uses a SLURM scheduler for distributing jobs. A list of input files consists of 95 simulation files obtained by the UrQMD-VHLLE generator for a collision energy

of 7 GeV. Figure 2 presents the chain and speedup of the MPD event data processing in dependence on the number of processor cores.

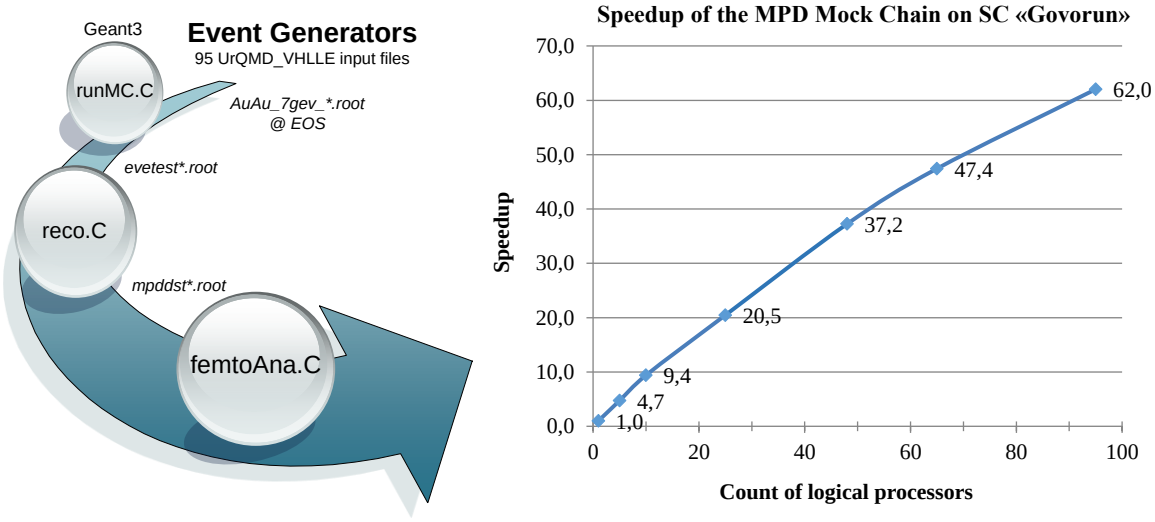

Figure 2. An example of MPD processing chain and speedup on the GOVORUN Supercomputer

4. Acknowledgement

The authors would like to thank the HybriLIT team for the support and LIT JINR for the opportunity to use the resources of the HybriLIT heterogeneous platform and GOVORUN Supercomputer (LIT, JINR), where computations were held. The work was funded by the Russian Foundation for Basic Research (RFBR) grant according to the research project 18-02-40102.

5. Conclusion

The current computing clusters for the NICA data processing contain various data storages based on dCache, EOS and GlusterFS distributed file systems, and various batch systems, such as SLURM, Sun Grid Engine and Torque. The new system for distributed job execution – the MPD-Scheduler was developed to simplify running user macros on the clusters in parallel. All external packages and the MpdRoot and BmnRoot frameworks including the MPD-Scheduler were installed and configured on the computing platforms for the NICA event processing. To test the deployed software infrastructure, the simulation–reconstruction–analysis chain was performed on the GOVORUN Supercomputer via the MPD-Scheduler. The detailed information on the described systems is presented on the technical web-site *mpd.jinr.ru* in the «Computing» section. To process a future huge amount of NICA events on worldwide distributed sites, three workload management systems are under discussion and investigation now: the ALFA framework of the GSI Institute, the DIRAC "interware" primarily used in the BES III and LHCb experiments, and the PanDA system originated from the ATLAS experiment.

References

[1] NICA White paper. Searching for a QCD mixed phase at the Nuclotron-based ion collider facility. Available at: mpd.jinr.ru/wp-content/uploads/2016/04/WhitePaper_10.01.pdf (accessed 21.11.2018).

[2] Rene Brun, Fons Rademakers. ROOT – An Object Oriented Data Analysis Framework // Proceedings AIHENP'96 Workshop, Nucl. Inst. & Meth. in Phys. Res. A. 389. 1997. pp. 81-86.

[3] Al-Turany M. et al. The FairRoot framework // Journal of Physics: Conference Series. Vol. 396, Part 2. 2012. P. 10.

[4] Heterogeneous platform "HybriLIT". Available at: http://hybrilit.jinr.ru (accessed 21.11.2018).