MULTITHREADED EVENT SIMULATION IN THE BMNROOT PACKAGE

M.M. Stepanova[1,a], A.V. Driuk[1], S.P. Mertz[2], K.V. Gertsenberger[2] and S.A. Nemnugin[1]

[1] *Saint Petersburg State University, 7-9 Universitetskaya emb., Saint Petersburg, Russia, 199034*

[2] *Joint Institute for Nuclear Research, 6 Joliot-Curie str., Dubna, Russia, 141980*

E-mail: [a] mstep@mms.nw.ru

The researches at the NICA accelerator complex (JINR, Joint Institute for Nuclear Research, Dubna) require efficient and fast software implementations of algorithms for modeling and reconstruction of the particle collision events. BmnRoot package developed for the BM@N experiment is based on the ROOT environment, Geant4 and the FairRoot object-oriented framework. BmnRoot includes tools for event simulation, reconstruction and data analysis modules. With the announcement of Geant4MT multithreading support in ROOT and FairRoot, the problem of modifying the BmnRoot code for multithreading simulation appeared. As a result, an event-driven parallelism model has been implemented for event simulation in BmnRoot. The stages of the performed work and the results of testing the implemented multithreaded version are presented.

Keywords: multithreading, Geant4, FairRoot, BmnRoot, BM@N, simulation

Margarita Stepanova, Andrei Driuk, Sergei Mertz, Konstantin Gertsenberger, Sergei Nemnugin

1. Introduction

A new accelerator complex NICA that it under construction at the Joint Institute for Nuclear Research (JINR, Dubna, Russia) consists of several experimental facilities. One of the main experiments of the NICA project is BM@N [2,3] (Baryonic Matter at Nuclotron). It is intended to study hot and dense baryonic matter in collisions of heavy ions with a fixed target.

For these studies one needs effective and fast software implementations of algorithms for simulation and reconstruction of the collision events. To solve the task, the BmnRoot framework [4] was developed for the BM@N experiment. BmnRoot includes necessary tools for investigation of the BM@N detector characteristics as well as for simulation, reconstruction and physics analysis of obtained data. Major components of the BmnRoot has been already implemented and the current challenge is optimization of the code and speeding up of the data processing, so the related work in different directions is conducted [5]. The presented study is intended to implement support of multithreaded event simulation in the BmnRoot framework. In order to reach the goal, the performance of multithreaded (MT) versions of external packages employed in the BmnRoot framework has been estimated, critical points in BmnRoot classes for implementation of the multithreaded simulation have been determined, and their modifications for MT have been made. Building, testing, debugging as well as valuation of scalability and efficiency of event simulation in the BmnRoot environment in MT mode have been also completed.

2. Support of multithreading in external packages used in BmnRoot

The BmnRoot framework has a module structure and is based on the object-oriented FairRoot framework [6], which is based on CERN ROOT environment [7]. The BmnRoot employs Geant4 transport package for event simulation [8]. Transfer to multithreaded mode is implemented by using the Geant4, Geant4 VMC, VMC in the ROOT, FairRoot, and BmnRoot packages on different levels of the employed hierarchy (Fig. 1). Prerequisites of the current work were the release of multithreaded versions Geant4MT and abstract interface for the particle transport package, named Geant4VMC, as well as multithreading support in the VMC ROOT and FairRoot packages.

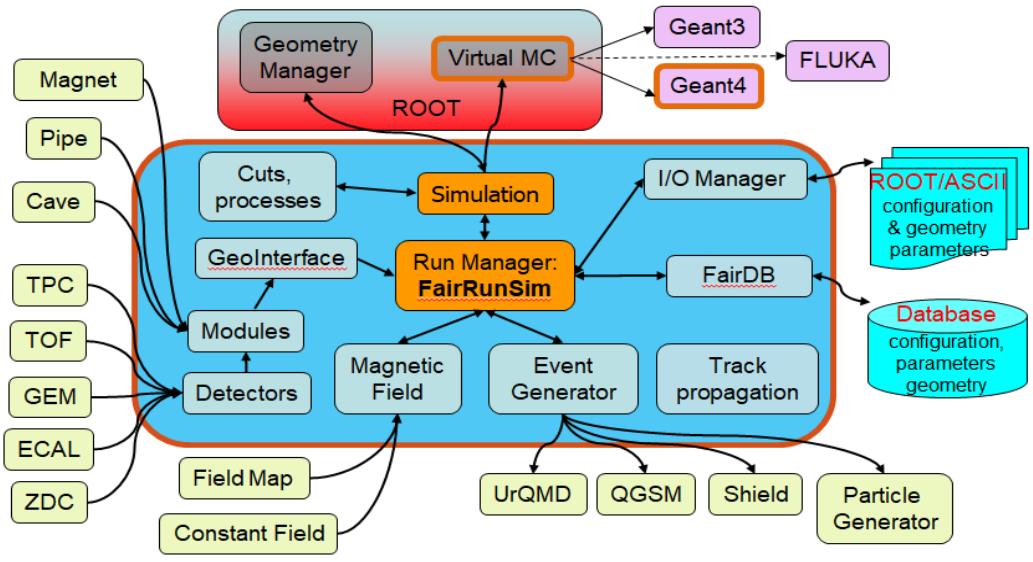

Figure 1. Structure of the event simulation module in BmnRoot

2.1 Geant4 MT

Geant is a set of tools for simulating of particle transport through detectors and media of an experimental setup. The multithreading support in the package started with version 10.03. Geant4 parallelization uses the POSIX standard. The Geant4MT provides parallelism on the event level. Events are independent, therefore they can be simulated by independent threads in parallel. The

parallelism model is master-worker in which one or more threads are responsible of performing the simulation, while a separate control flow manages their work (Fig. 2).

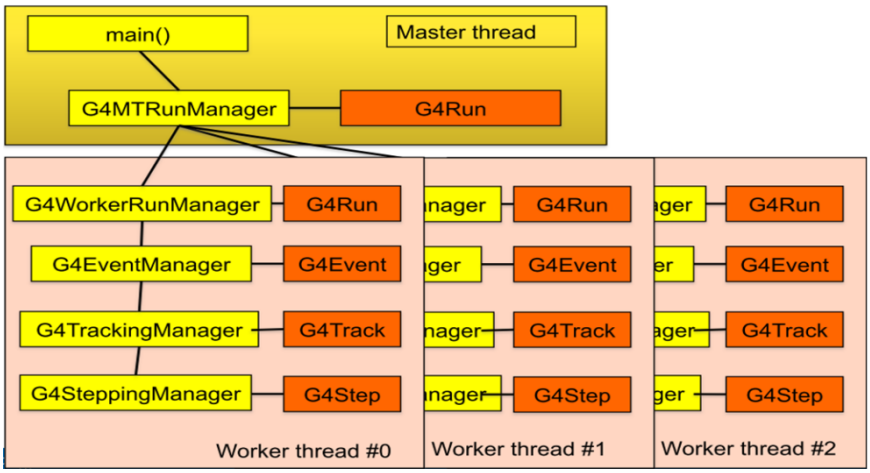

Figure 2. General scheme of the multithreading Geant4MT application

Similar to the sequential version of Geant4, master and workers are represented by instances of classes inheriting from G4RunManager: the G4MTRunManager class represents the master model, while G4WorkerRunManager instances represent worker models. User should create only one instance G4MTRunManager, which instantiates and controls one or more G4WorkerRunManager instances. Before starting the simulation, the master generates seeds for each events. This operation guarantees reproducibility. After construction and tuning the threads, each worker thread is responsible for creation its own G4Run instance and simulation of a subset of particle collision events [8].

Thread safety has been achieved by using the local thread storage (TLS). This allows using the code without blockings by increasing memory size and by a small increase in usage of processor time.

2.2 Geant4 VMC (MT)

Virtual Monte Carlo (VMC) represents a single interface for various transport codes. VMC for Geant4 has the MT support, starting with version 3.0. In 2019 the VMC package was removed from the ROOT environment. This package, as well as the Geant4VMC, is available from a separate repository [9,10].

2.3 FairRoot framework

The FairRoot is an object-oriented environment for simulation, reconstruction, and analysis of data, based on the ROOT environment. It provides basic classes that allow fast construction of classes for experimental data processing and solving other software problems in high energy physics. For example, using the virtual Monte-Carlo concept, one can perform simulation with use of the Geant3 or Geant4 without modification of the code or experimental setup geometry. Main modifications in FairRoot, implemented for the multi-threading support are the following [11]:

- Class FairMCApplication: CloneForWorker() and InitOnWorker() functions have been added.
- Class FairModule: a new virtual function, which must be overridden to clone objects for each worker thread, has been added: FairModule* CloneModule() const.
- Class FairRootManager has been expanded: automatic creation and writing output files in the ROOT format for each thread has been made.
- Classes FairRunSim, FairStack, FairRootManager and FairLogger have been modified.

For migration of applications inherited from the FairRoot base classes, it is necessary to implement the new TVirtualMCApplication functions, which are used to clone objects of simulation classes in worker threads.

2.4 Base Packages: MT support

Table 1 shows the versions of the basic packages that have been tested, the minimal versions containing the MT support and the most actual versions.

Table 1. Versions of external packages with MT support

Package	Min version	Used version	Last version
Geant4	10.03.p01	10.05.p01	10.07.p02
Geant4VMC	3.0	4.0.p1	5.3
ROOT	6/09/04	6/16/00	6.24.06
FairRoot	18.2.0	18.2.0	18.6.4

To enable support for multithreaded simulation in the BmnRoot package, during the initial configuration of the base packages one needs to change only the value of default option to "compile Geant4 in multi-threaded mode" [4]. The number of threads can be changed in event simulation either by assigning the value of the environment variable G4FORCENUMBEROFTHREADS or directly in the C++ code: runManager->SetNumberOfThreads(8).

Presented work was accomplished on the NICA cluster node (LHEP, JINR, Dubna) with the following configuration: 2x Intel Xeon E5-2697a, DDR4-2400, 2.6GHz, 32(64) cores, 512G RAM, CentOS7, EOS Storage.

2.5 Testing multithreaded event simulation

For testing the new multithreaded mode for simulation of the particle collision events, an example from FairRoot employed in modeling of Rutherford's experiment, was used. Ion and BOX event generators and geometry of the setup consisting of CAVE, Target, and RutherfordDetector modules, were used. As follows from Fig. 3, for different number of events (nEvents = 10^4, 10^5, 10^6), an observed speedup depends on the number of threads. It reaches a maximum value of the order of 2.6 for 6 cores and decreases after. It should be noted that such a result is observed only with a sufficiently large number of events, or high multiplicity of events.

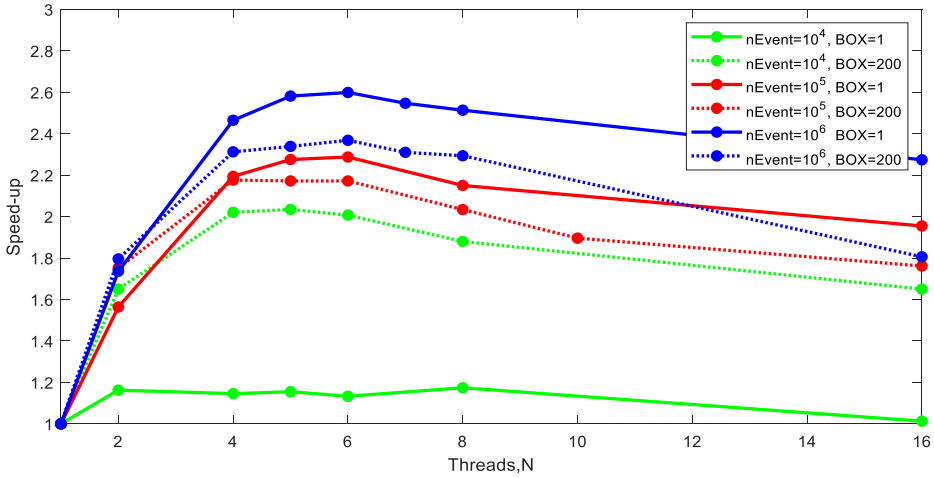

Figure 3. Dependence of speedup on the number of the threads performing simulation

3. Integration of the multithreaded support into BmnRoot

At the first stage, the BmnRoot code was analyzed and the key classes that require modifications to support multithreaded simulation were found. The following modifications of the source code were made:

- The CloneModule function and copying constructors were added to the classes of all BM@N detectors (BmnFD, BmnBd, BmnCSC, BmDch, BmnEcal, BmnSilicon, BmnMWPC, CbmSts, BmnTOF, BmnTOF1, BmnZdc, BmnPsd,BmnRecoil).
- CbmSts class: the methods for registering particles in threads were added.
- CbmStack class: cloning of the objects in threads was implemented.

Simulation in the BmnRoot framework is launching in the MT mode. During this work, a serious decay in efficiency instead of expected acceleration with growing number of threads was found. This is related with reading operations from the file of real map of magnetic field in the BM@N experiment. Substitution of reading from the BmnFieldMap() file for use of constant magnetic field BmnFieldfConst() became a temporary solution of this problem.

First results for simulation of particle collision events in the BM@N experiment were obtained with use of the Ion and BOX event generators and simplified geometry and consisting the following set of detectors: CAVE, MAGNET, STS, ECAL, and ZDC.

Dependencies of speedup on the thread number at different numbers of events were determined (Fig. 4). Solid lines represent speedup for full modeling time. The dotted line shows the dependences of the speedup over the averaged running time of the threads. The results demonstrate a good speedup and scalability, with performance gains as the number of events processed increases.

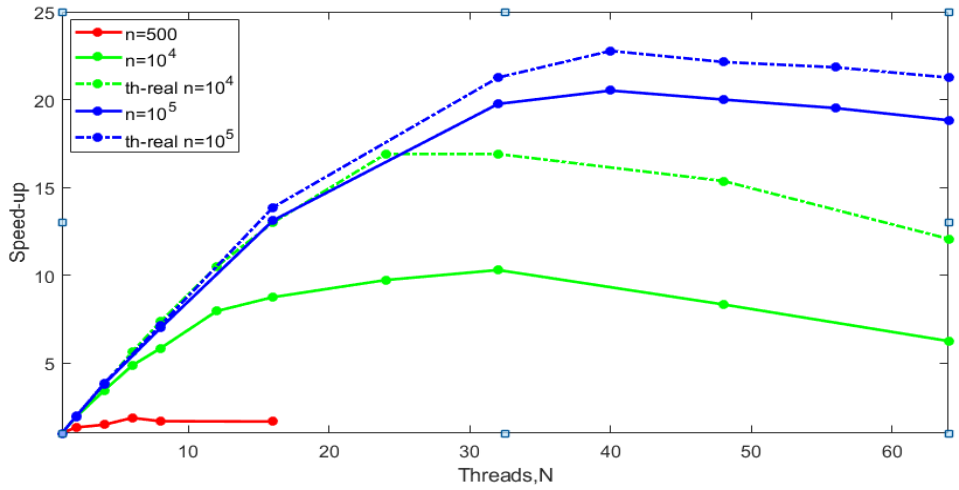

Figure 4. Scalability of events modeling in BM@N experiment with uniform magnetic field

4. Conclusion

The possibility of implementing a multithreaded simulation mode of events in the BM@N experiment of the NICA project in the BmnRoot environment is demonstrated. Simulations in the BmnRoot are based on parallel Geant4 MT transport code. Estimates of Geant4 MT scalability have been obtained for Rutherford experiment. Classes of the BmnRoot code that are involved in multithreaded simulation are defined. Problems of usage of Geant4 MT in the BmnRoot package are found and their solutions are proposed.

Currently, modification of the simulation code in the BmnRoot environment is required. It includes, in particular, a modification of the software implementation of all detectors for parallel event processing, support for reading a real magnetic field from a file, support for reading of primary particles from a file, modification of some event generators as well as resolving of possible problems with thread safety, for example, merge of output files into a single resulting file.

The results presented in the paper indicate good scalability of the simulation under certain conditions. Taking into account the full set of detectors and supporting reading from a file of real magnetic field values can degrade scalability, but the present study indicates the promise of using multithreading in the BmnRoot.

5. Acknowledgement

This work was supported by Russian Foundation for Basic Research, grant 18-02-40104 mega.

References

[1] Official site of NICA accelerator complex. http://nica.jinr.ru (accessed 23.09.2021)

[2] Baranov D et al. 2018 KnE Energy Phys. 3 291–6

[3] Gertsenberger K, Merts S P, Rogachevsky O V et al. 2016 Eur. Phys. J. A 52 214

[4] BmntRoot installation. https://bmn.jinr.ru/software-installation/ (accessed 23.09.2021)

[5] Driuk A V, S P Merts, S A Nemnyugin, V A Roudnev, M M Stepanova, A A Iufryakova. Journal of Physics:Conference Series 1690 (2020) 012066

[6] Fairroot. https://fairroot.gsi.de (accessed 23.09.2021)

[7] CERN ROOT. https://root.cern.ch (accessed 23.09.2021)

[8] Geant4 User's Guide for ToolkitDevelopers. https://Geant4.web.cern.ch/support/user_documentation (accessed 23.09.2021)

[9] I Hrivnacova. EPJ Web of Conferences 245, 02005 (2020) (CHEP 2019)

[10] vmc-project. https://vmc-project.github.io/download/ (accessed 23.09.2021)

[11] I Hrivnacova. EPJ Web of Conferences 214, 02018 (2019) (CHEP 2018)