Parallelized JUNO simulation software based on SNiPER

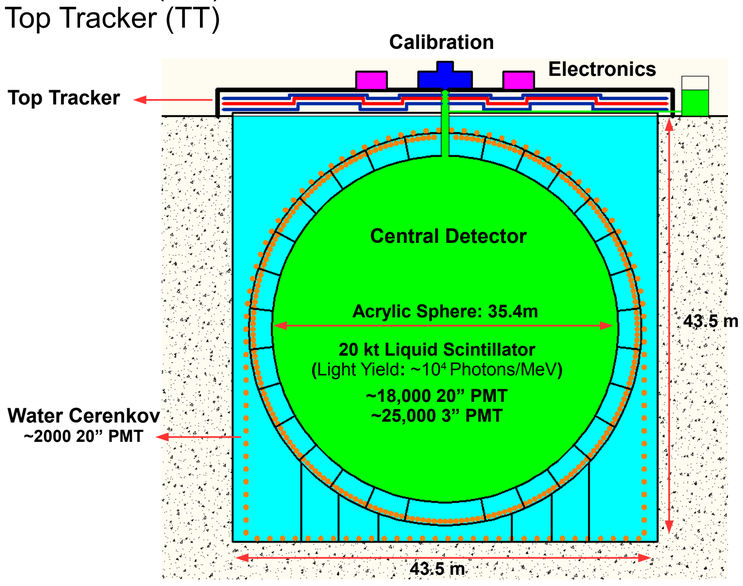
Tao Lin[1,†], Jiaheng Zou[1,†], Weidong Li[1], Ziyan Deng[1], Guofu Cao[1], Xingtao Huang[2], Zhengyun You[3]
(On behalf of JUNO Collaboration)

[1]Institute of High Energy Physics, Beijing [2]Shandong University, Jinan [3]Sun Yan-sen University, Guangzhou
[†]Email: lintao@ihep.ac.cn, zoujh@ihep.ac.cn

Introduction

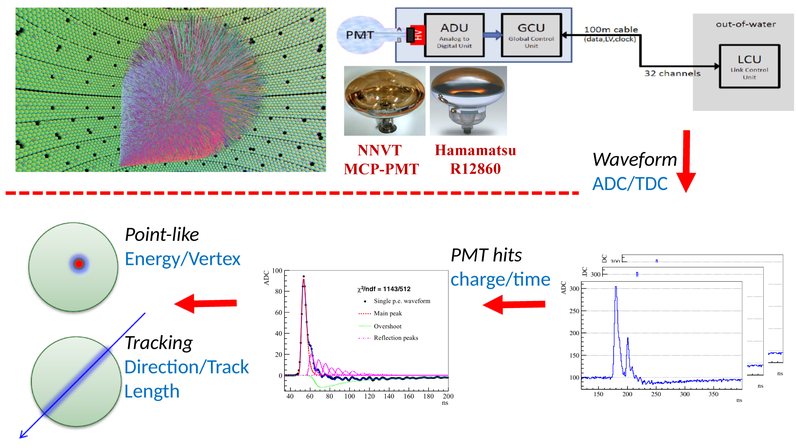
The JUNO (Jiangmen Underground Neutrino Observatory) [1,2] is a multipurpose neutrino experiment which is designed to determine neutrino mass hierarchy and precisely measure oscillation parameters. It will be located in southern China about 53 km away from Yangjiang and Taishan nuclear power plants.

- Central Detector (CD)
- Water Pool (WP)
- Top Tracker (TT)



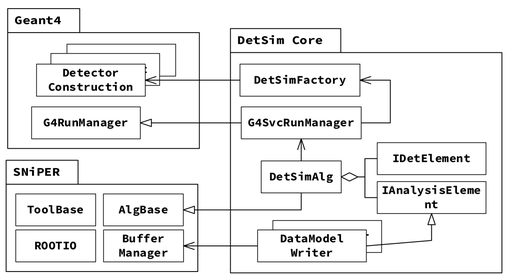
Offline Software

JUNO Offline Software is developed based on SNiPER [3]. It consists a full chain of data processing, including physics generator, detector simulation, electronics simulation, calibration and reconstruction.



Detector Simulation Framework

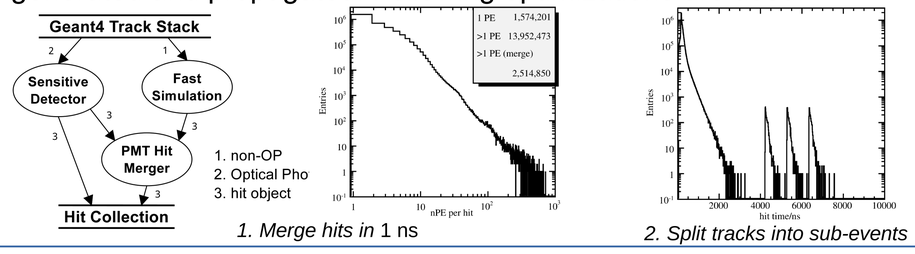
Detector simulation framework is implemented to integrate SNiPER and Geant4 (9.4) [4].



Features:

- Lightweight simulation framework.
- Easy to migrate from standalone application.
- Support both batch and interactive modes.
- Support Geant4's macro files/commands in Python.
- Modular design of User Action.

For simulation of cosmic ray muons, millions of optical photons are generated and propagated. Following optimizations are used.

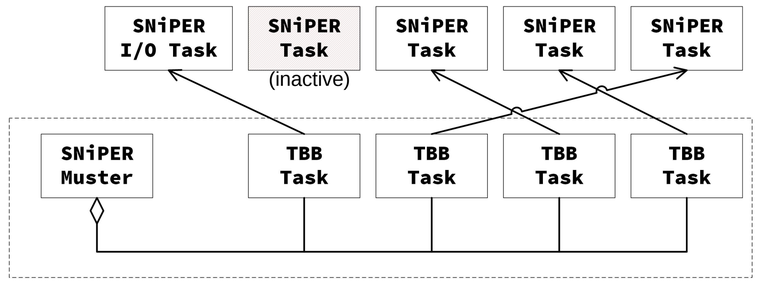


1. Merge hits in 1 ns

2. Split tracks into sub-events

SNiPER Muster

SNiPER **Muster** (Multiple SNiPER Task Scheduler) is a task-based scheduler. The SNiPER task is configured by user and could be run in sequential mode. For the parallelism mode, the SNiPER Task could be executed by an underlying worker, which is based on Intel TBB (Threading Building Blocks). Because the decouple of SNiPER Task and underlying implementation, we can use other implementations, such as pthread.

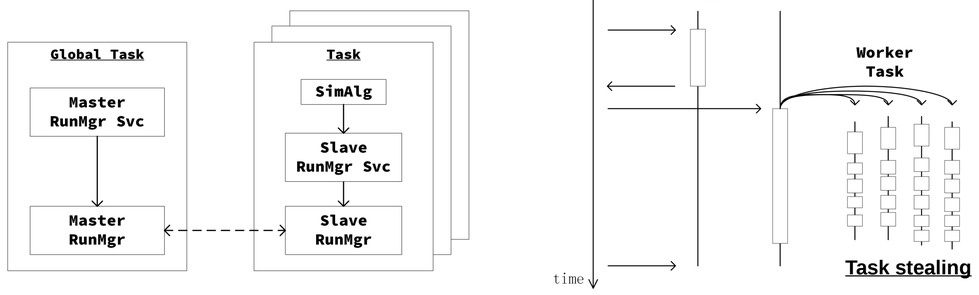


Based on "task stealing", events could be dispatched dynamically.

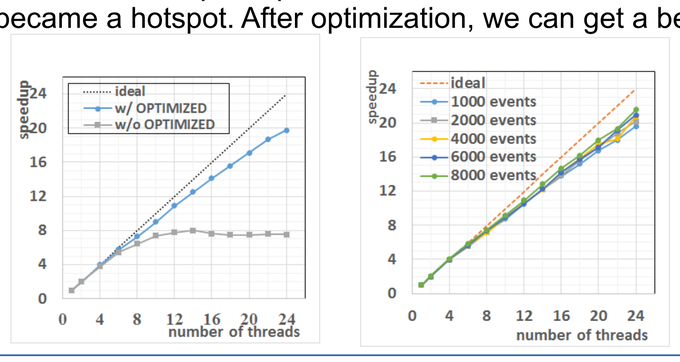
Note: to keep locality and achieve better performance, a SNiPER task will bind to a specific TBB task by default.

SNiPER Muster based Simulation Framework

Starting from Geant4 10.x, multi-threaded Geant4 applications enable event-level parallelism [5]. With the evolution of Geant4, it makes possible to run simulation with Intel TBB, MPI and so on [6].



Due to a lot of optical photons, mutex used in Geant4's material properties table became a hotspot. After optimization, we can get a better performance.

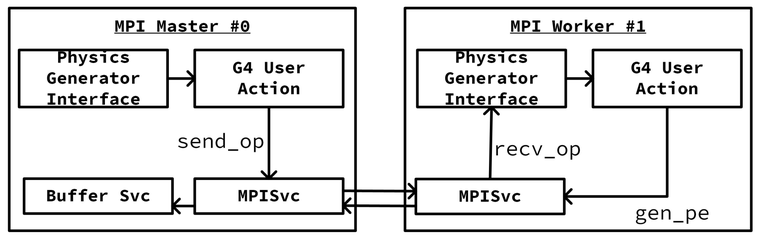


- CPU: Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz (2 sockets, 12 cores per socket).
- Memory: 64 GB
- OS: Scientific Linux 6.5; GCC: 4.9.4
- TBB: tbb2017_20160916oss
- Geant4: 10.03.p01
  - G4EmStandardPhysics + G4OpticalPhysics
- Particle: 2.2 MeV gamma at center.

Summary and Plans

In this poster, we present an event-level parallelism simulation framework based on Geant4 10.x and SNiPER Muster. Benefit from the multi-threading support and thread safe of Geant4 10.x, we integrate Geant4 application into the framework. Even though SNiPER Muster is based on Intel TBB, SNiPER could still work together with pthread. The software is optimized to achieve a good performance.

To speedup events such as cosmic ray muons, we are investigating how to use track-level parallelism in our simulation framework. Our current prototype is using MPI to dispatch tracks into different work nodes (Rank #0 is master).



Reference

[1] Z. Djurcic *et al.* [JUNO Collaboration], arXiv:1508.07166
[2] F. An *et al.* [JUNO Collaboration], J. Phys. G **43**, no. 3, 030401 (2016)
[3] J.H. Zou *et al.*, J. Phys. Conf. Ser. **664**, no. 7, 072053 (2015)
[4] T. Lin *et al.* [JUNO Collaboration], arXiv:1702.05275
[5] G. Cosmo [Geant4 Collaboration], J. Phys. Conf. Ser. **513**, 022005 (2014).
[6] A. Dotti *et al.*, arXiv:1605.01792