Detector Simulation in the JUNO Experiment

MiaoYu on behalf of JUNO Collaboration

Institution of High Energy Physics, CAS, Beijing, China; Wuhan University, Wuhan, China;

Experiment Introduction

- The Jiangmen Underground Neutrino Observatory (JUNO) is a multi-purpose neutrino experiment with a 20 kton liquid scintillator detector [1].
- JUNO is now under civil construction in Jiangmen in southern China.
- JUNO has rich physical purposes:
  - Mass ordering and precision measurement of oscillation parameters with reactor neutrinos;
  - Supernova neutrino;
  - Solar neutrino;
  - Geo-neutrino;
  - Atmospheric neutrino;
  - Exotic searching, such as proton decay, dark matter
- A Geant4-based full detector simulation for JUNO has been built under the SNiPER framework

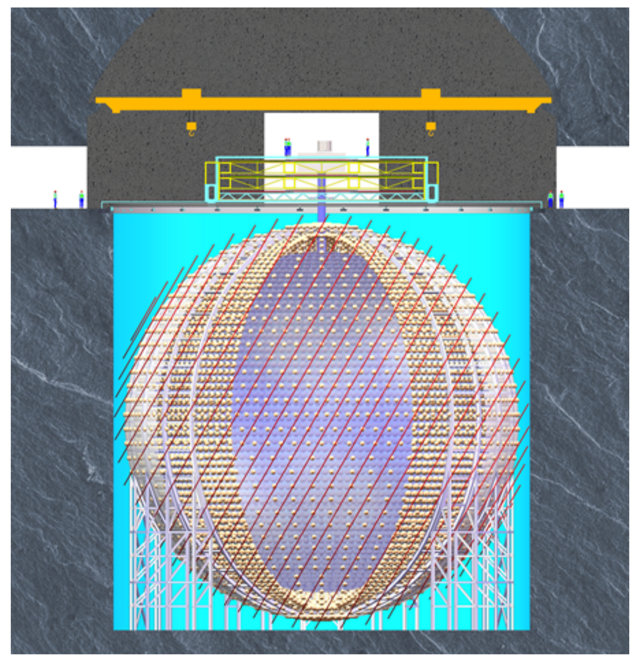

Fig. 1 JUNO Detector

Parallelized Simulation

- SNiPER Muster (Multiple SNiPER Task Scheduler) is a task-based scheduler.
- In parallelism mode, the SNiPER task could be executed based on TBB (Threading Building Block).
- Starting from Geant4 10.x, multi-threaded Geant4 applications enable event-level parallelism.
- The computing performance of the prototype looks promising.
- More work will be done to optimize the parallelized simulation.

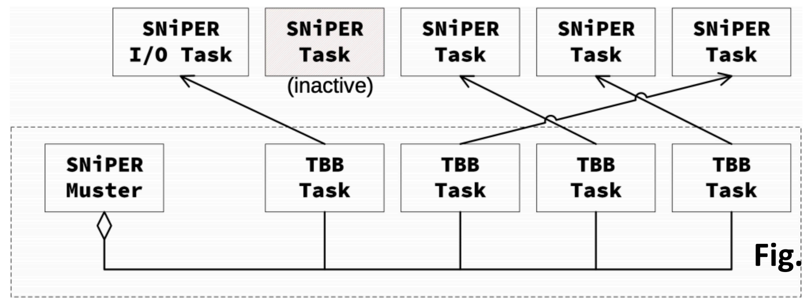

Fig. 5

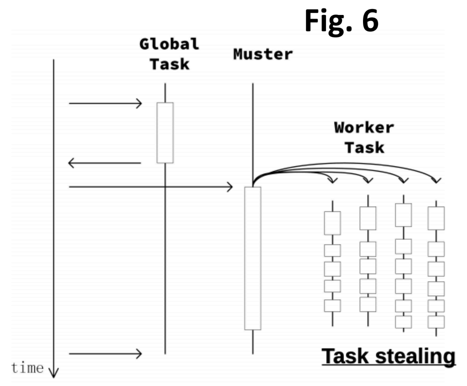

Fig. 6

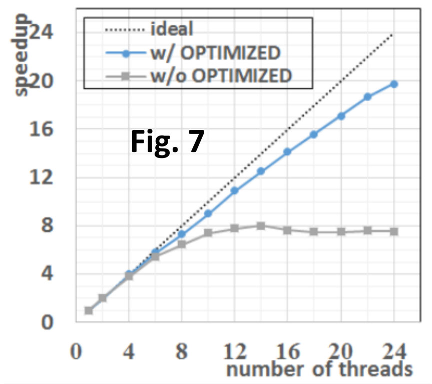

Fig. 7

Fig. 5 SNiPER Muster
Fig. 6 Flow chart of paralleled simulation
Fig. 7 performance

Electronics Simulation

- Thank to the specific features of SNiPER, a “pull” mode electronics simulation is proposed and implemented in JUNO, quite different with most of other experiments.
- Main features:
  - Easily handling time-correlated events, like IBD;
  - Convenient hit-level event-mixing;
  - Work in pipeline mode, good memory control.

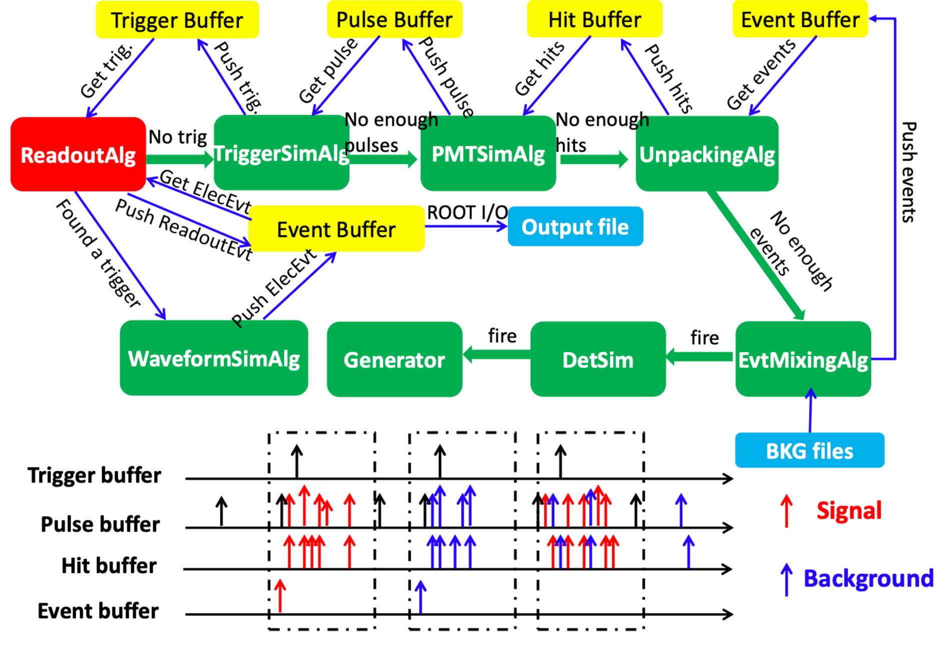

Fig. 13 workflow in electronics simulation

JUNO Offline Software

- SNiPER: Software for Non-collider Physics ExpeRiments developed by JUNO Collaboration.
- Offline: extensions of SNiPER and application for JUNO.
- External Libraries (EL): very frequently used software and tools.

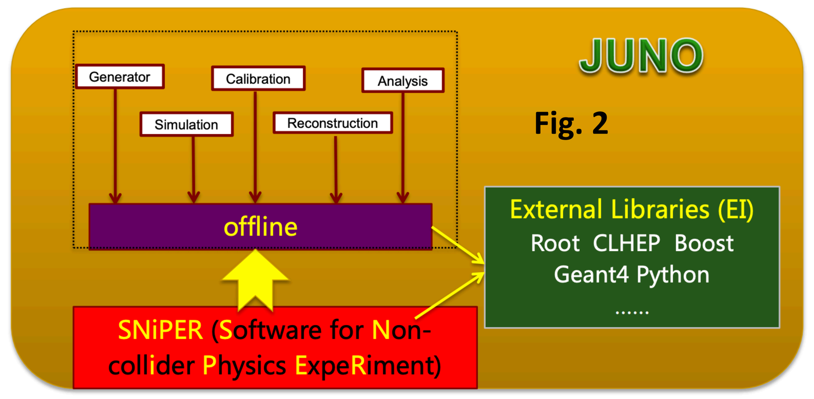

Fig. 2

- Key functionalities of framework:
  - Dynamically loading packages and elements
  - Flexible execution
  - Event management in memory
- Currently, the SNiPER is used in JUNO, LHAASO and nEXO experiments.

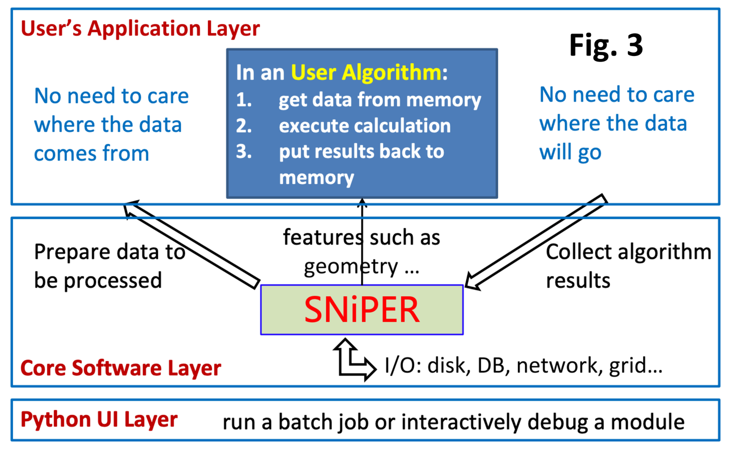

Fig. 3

Fig. 2 JUNO Offline Software Framework
Fig. 3 How to work with SNiPER

Fast Simulation for Cosmic Rays

- Muons arriving at JUNO liquid scintillator have mean energy 215 GeV and yield large amount of photons. The full simulation with Geant4 is extremely CPU consuming.
- Voxel method: dividing the liquid scintillator sphere into voxels and building the connections between the visible energy in a voxel and the response of the PMTs.
- Fast simulation shows coincident results with full simulation.
- Fast simulation achieves a speedup ratio of around 50 over the full simulation.

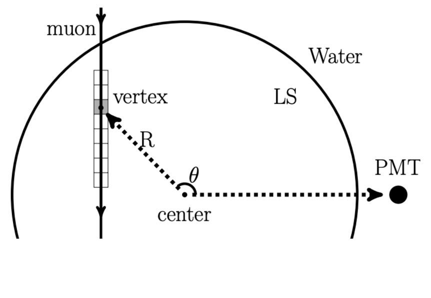

Fig. 8 Mapping E_{vis} to the response of PMTs

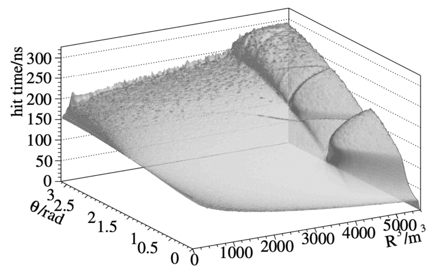

Fig. 9 Mean hit time profiles

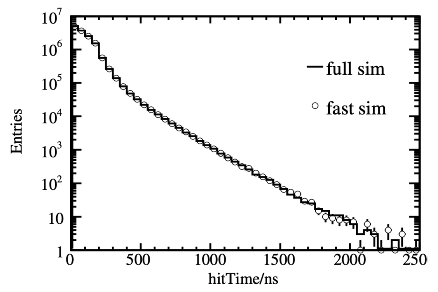

Fig. 10 Hit time of PMTs

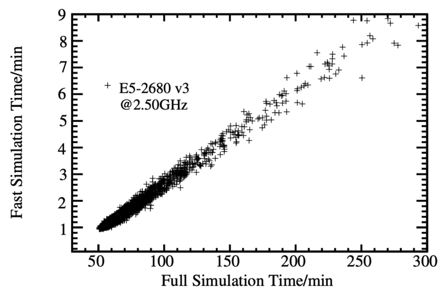

Fig. 11 Simulation time

Visualization

- ELAINA: Event Live Animation with unIty for Neutrino Analysis, is a visualization method based on Unity engine [7].
- ELAINA provides better visual effects compared with the traditional ROOT-based event display.
- ELAINA is independent of any offline software, which makes it convenient to run on local computers.
- ELAINA has great potential to develop more powerful functions and applications such as online monitoring and VR programs.

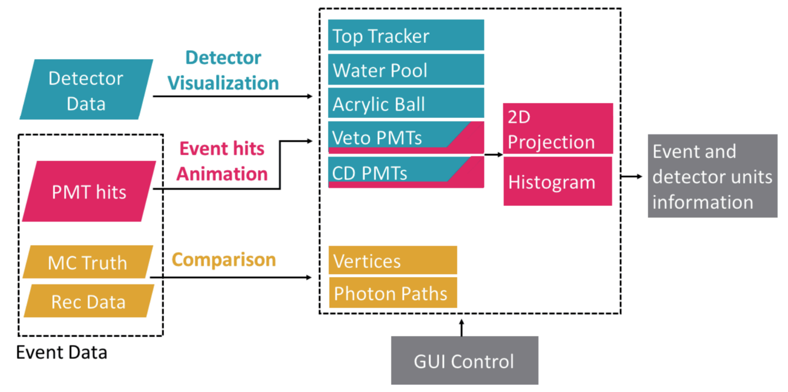

Fig. 13 Structure and data flow in ELAINA

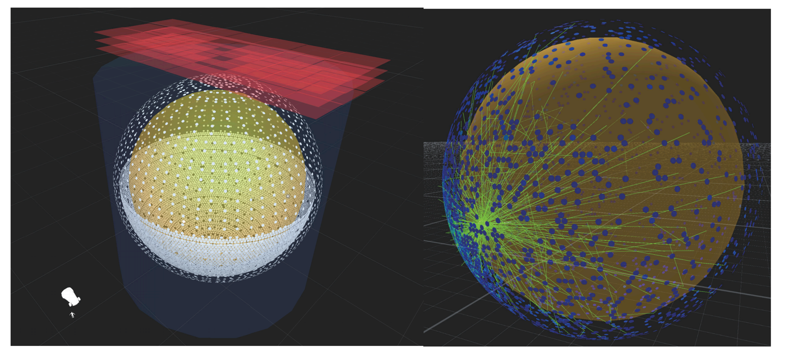

Fig. 14 Visualization of JUNO detector and event

Detector Simulation

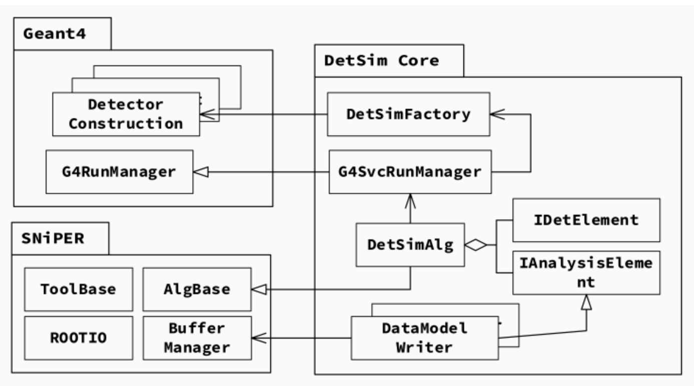

Fig. 4 Interface of Detector Simulation

- The modules of generator, geometry management, GEANT4-level detector simulation, background mixing and electronics simulation have been implemented at JUNO.
- Detector simulation framework is implemented to integrate SNiPER and Geant4 [3].
- Features:
  - Lightweight
  - Easy to migrate from standalone application.
  - Support both batch and interactive modes.
  - Support Geant4's macro file/commands in Python.
  - Modular design of user action.
- A parallelized simulation software [4] is developed in JUNO.
- A fast simulation method [5] has been established for cosmic muons.
- A GPU based simulation tool (Opticks) [6] has been developed for tracking optical photons.

Opticks

- In JUNO simulation, large amount of optical photons will cause huge memory and time expense, such as muon simulation.
- Opticks enables Geant4-based simulations to benefit from effectively zero time and zero CPU memory optical photon simulation, due to massive parallelism made accessible by NVIDIA OptiX.
- Collecting and copying “gensteps” rather than photons avoiding allocation of CPU memory for the photons, only photons that reach sensors requiring CPU memory allocation.
- >1000 faster than CPU optical photon simulation.

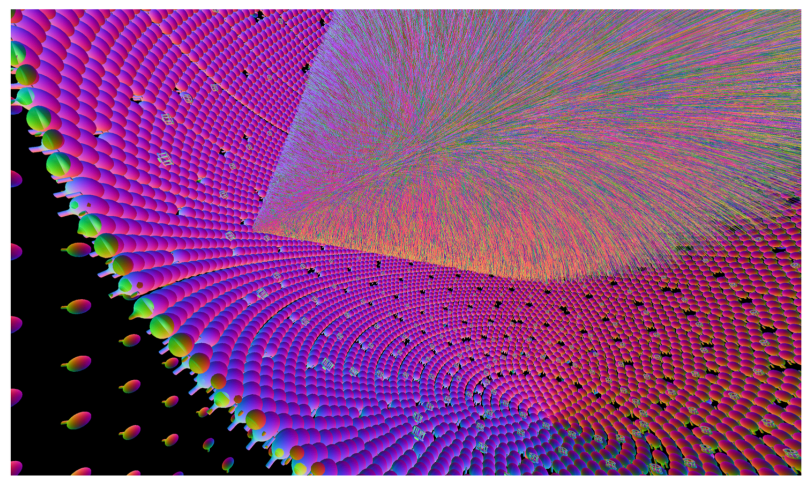

Fig. 12 Optical photons from a 200 GeV muon crossing the JUNO liquid scintillator

Summary

- SNiPER has been developed for non-collider experiments.
- JUNO simulation framework has been designed and developed, and further optimization is important for physical performance improvement.
- Several methods such as fast simulation and opticks have been used to overcome the bottleneck due to large amount of optical photon simulation.

Reference

[1] JUNO collaboration, “Neutrino physics with JUNO”, Journal of Physics G: Nuclear and Particle Physics, Volume 43, Number 3
[2] J.H. Zou et al., J. Phys. Conf. Ser. 664, no. 7, 072053 (2015)
[3] Geant4 Collaboration, J. Phys. Conf. Ser. 513, 022005 (2014)
[4] arXiv:1710.07150
[5] arXiv:1602.00056
[6]https://bitbucket.org/simoncblyth/opticks
[7] arXiv: 1812.05304

miaoyu@ihep.ac.cn

Neutrino 2020 Poster

June 22-July 2, 2020