The Multi-threaded Detector Simulation in JUNO

Peidong Yu (yupd@ihep.ac.cn)
on behalf of the JUNO collaboration

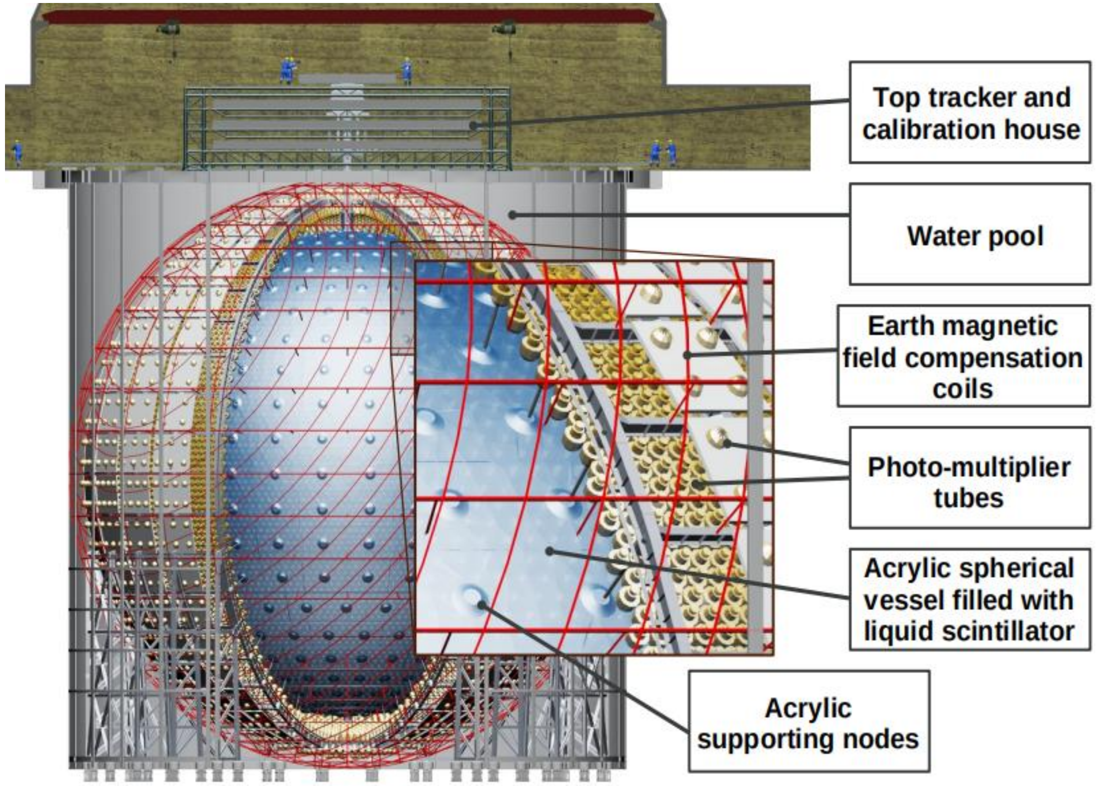
CHEP 2024, Krakow, Poland, 21–25 Oct 2024

Outline

- JUNO experiment
- Challenges and optimization in simulation
- Multi-threaded simulation software
- Summary

JUNO experiment

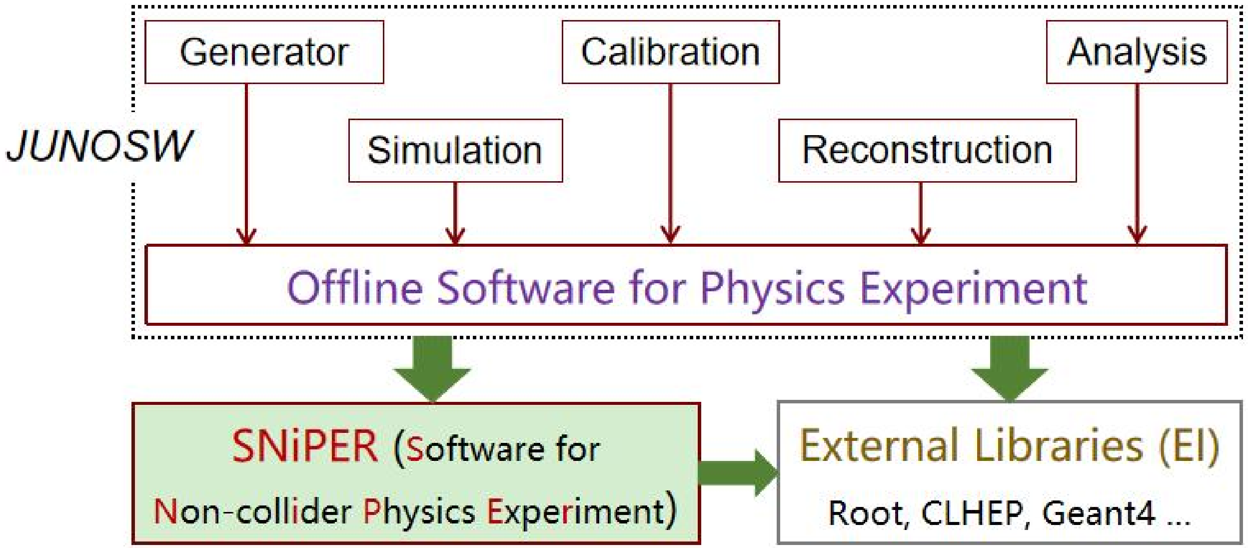
- **JUNO: Jiangmen Underground Neutrino Observatory**
  - Rich physics program [1]
    - Neutrino mass ordering and precise measurement of 3 oscillation parameters
    - Reactor neutrinos, supernova burst neutrinos, geo neutrinos, atmospheric neutrinos, and solar neutrinos
  - JUNO detector
    - 700 m deep underground
    - Central detector: 20 kton LS with 3%@1MeV of energy resolution.
    - Water Cerenkov detector and Top Tracker
  - Data taking expected in ~2025
  - Lifetime: 20+ years

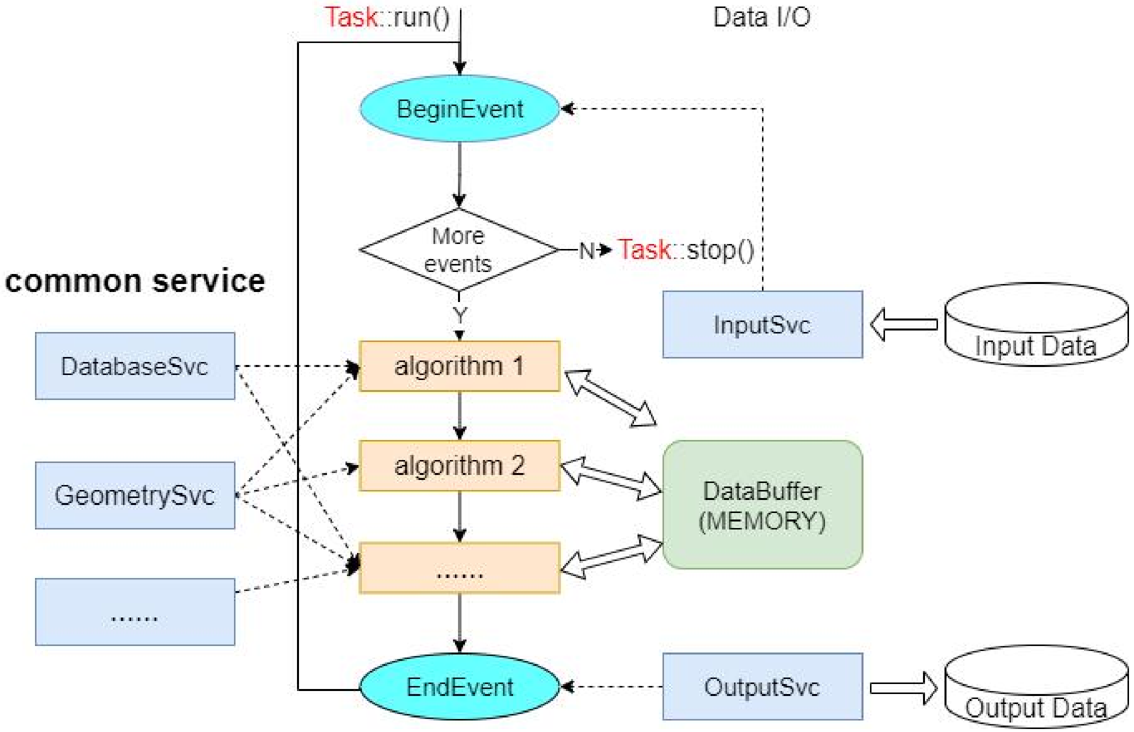


[1] JUNO Collaboration, Prog.Part.Nucl.Phys. 123 (2022) 103927

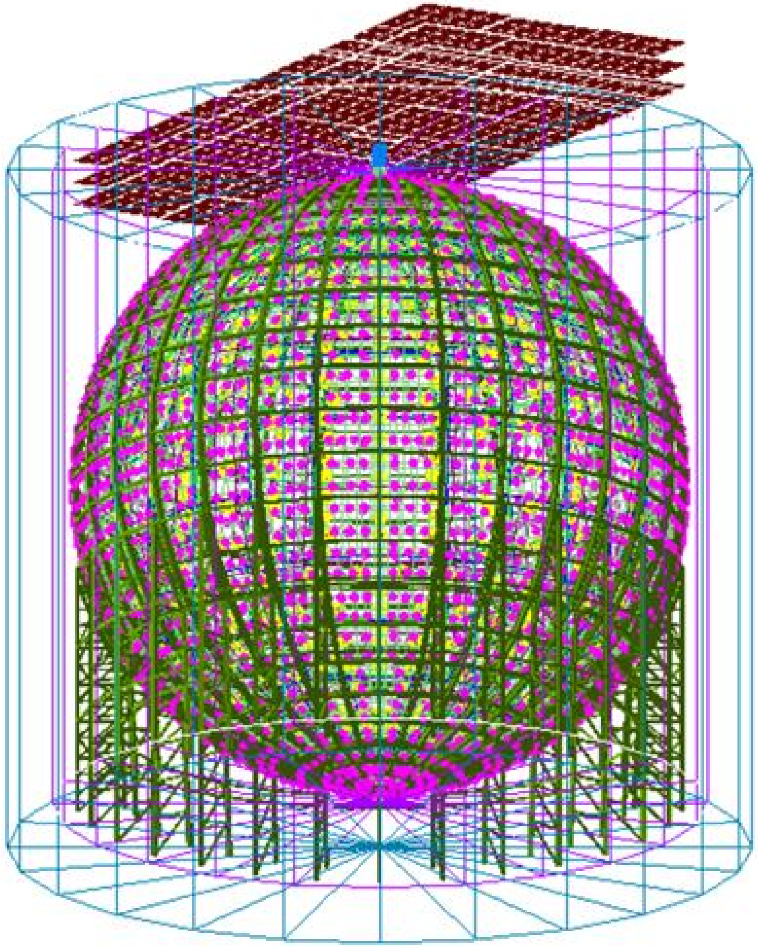
JUNOSW

- JUNOSW is the offline software
  - Implements the JUNO dependencies based on SNiPER
- SNiPER is a general purpose software framework used by several HEP exp.
  - No dependencies to a specific experiment, including JUNO

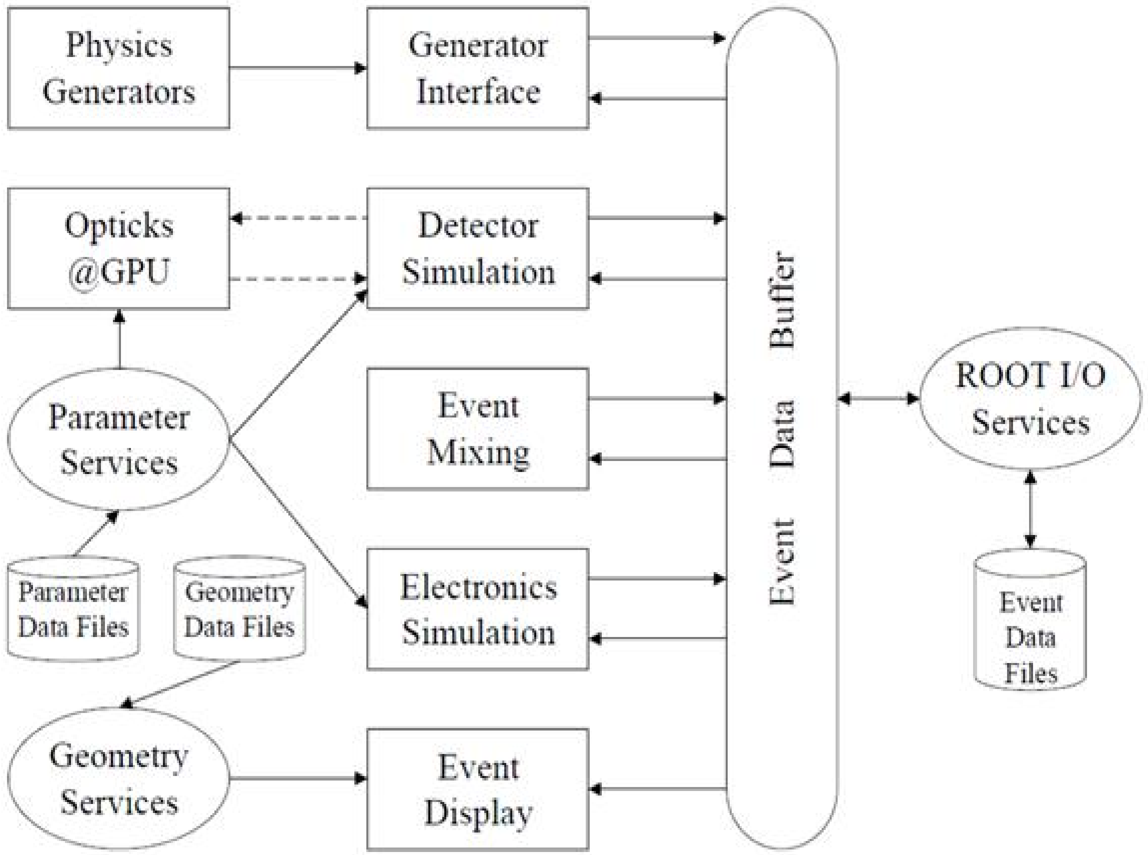




JUNO detector simulation software



The JUNO geometry of detector simulation



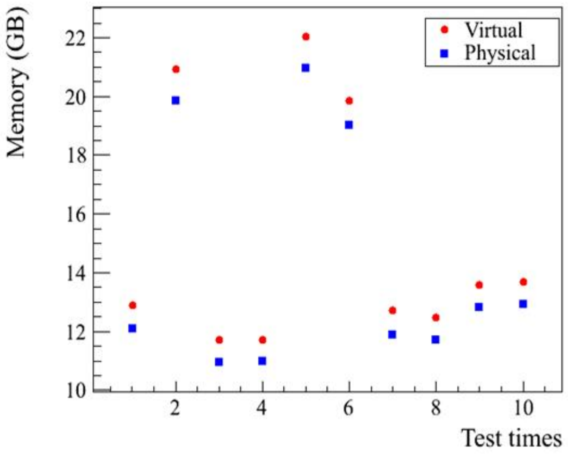
The structure of detector simulation[1]

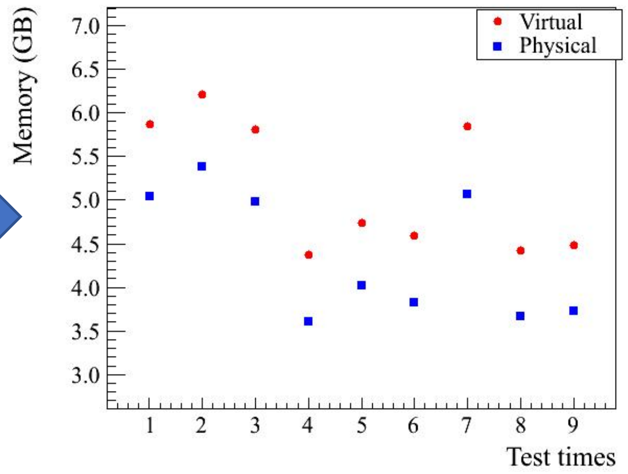
[1] Eur. Phys. J. C (2023) 83: 382， Erratum: Eur. Phys. J. C (2023) 83: 660

Optimization of DetSim memory consumption

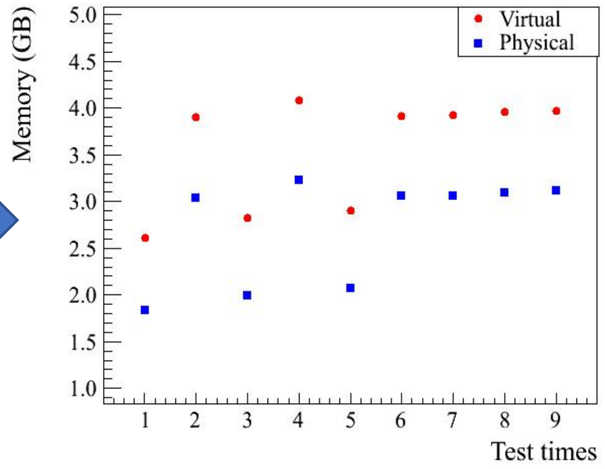
Full optical photon simulation implemented in JUNO offline software.
Large detector size and high energy deposition, millions of optical photons generated, high memory consumption.

For high energy muon





--pmtsd-merge-twindow 1.0



--pmt-hit-type 2 --pmtsd-merge-twindow 1.0

1 The hits within the same time window(1ns) is merged into one hit

2 The compact hit type

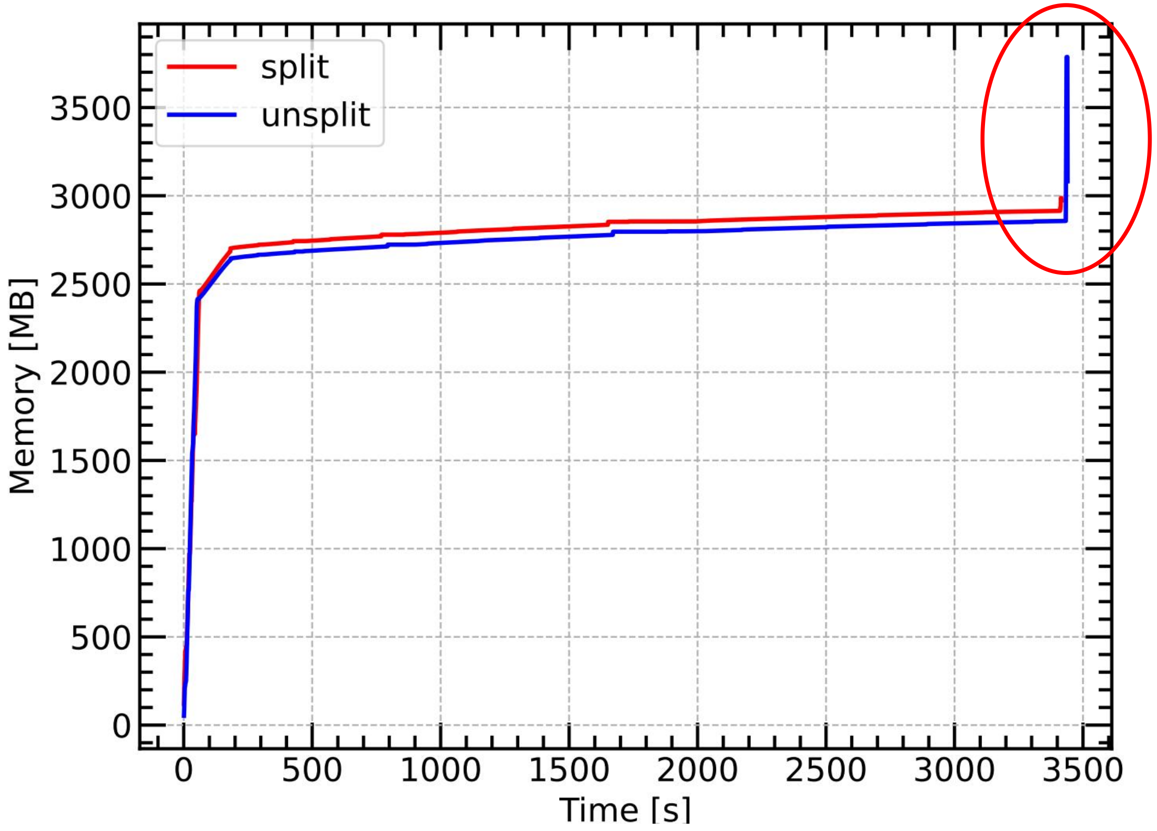
hittype1
248 bytes

→

hittype2
40 bytes

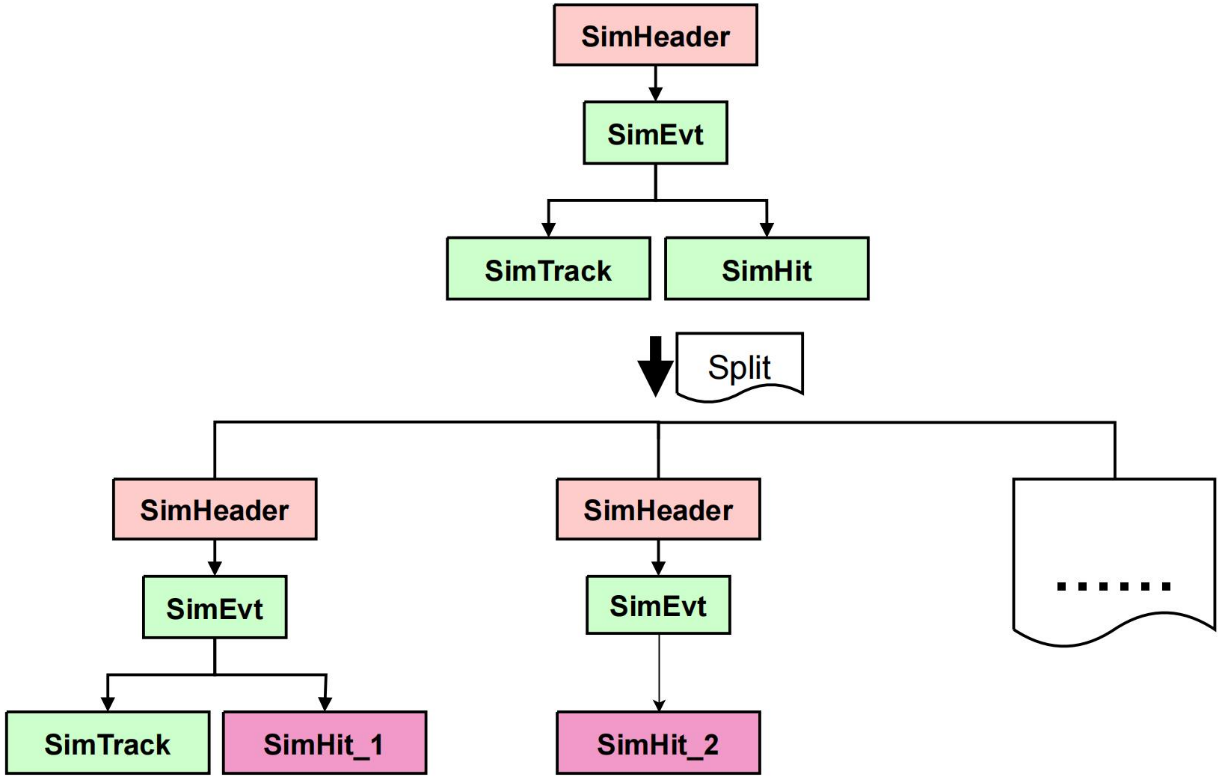
Optimization of DetSim memory consumption

Memory changes over time in muon simulation



Due to the compression algorithm of ROOT, memory will sharply increase during output

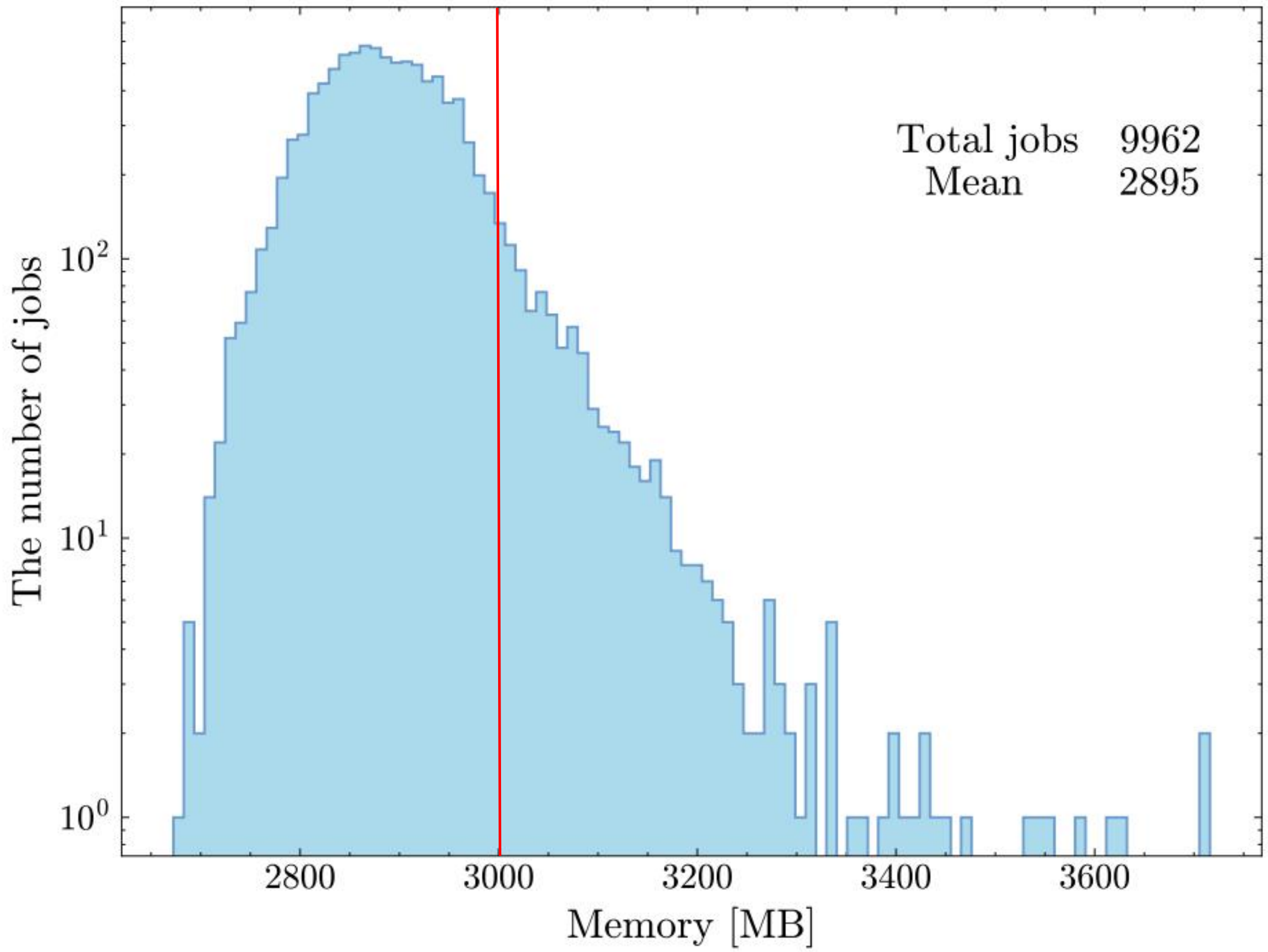
③Implementation of the DataModelWriterWithSplit method

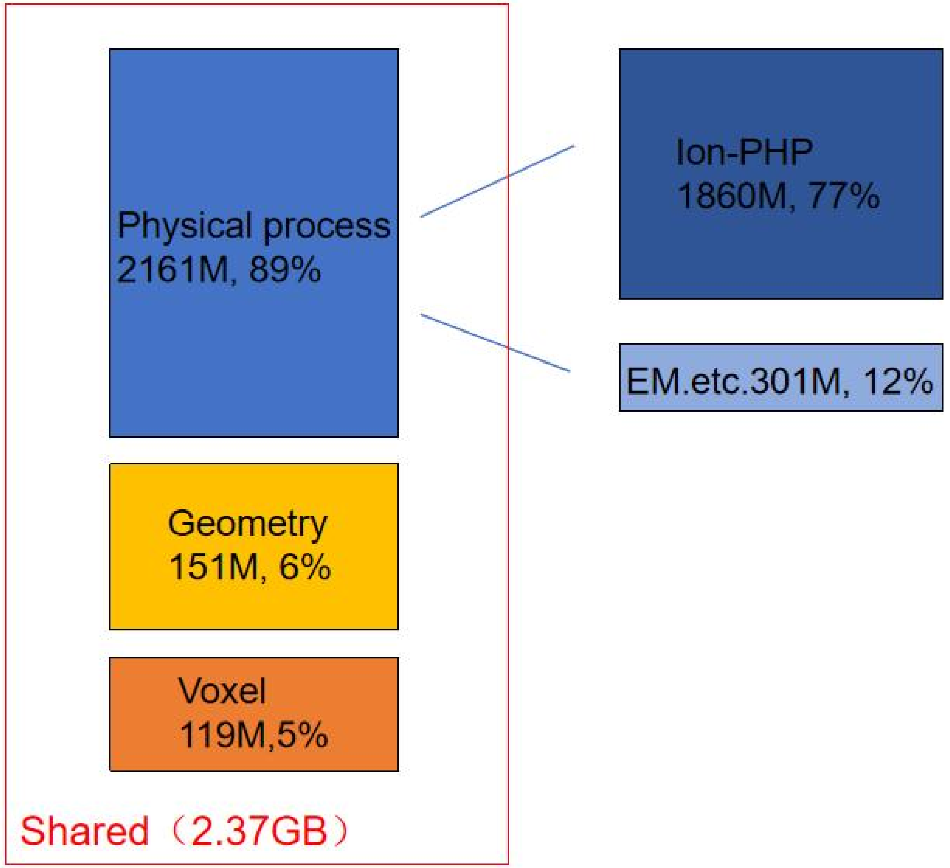


Split a collection of hits within an event into multiple parts to complete the output

Memory consumption after optimization

Memory consumption during the Muon simulation： >3GB



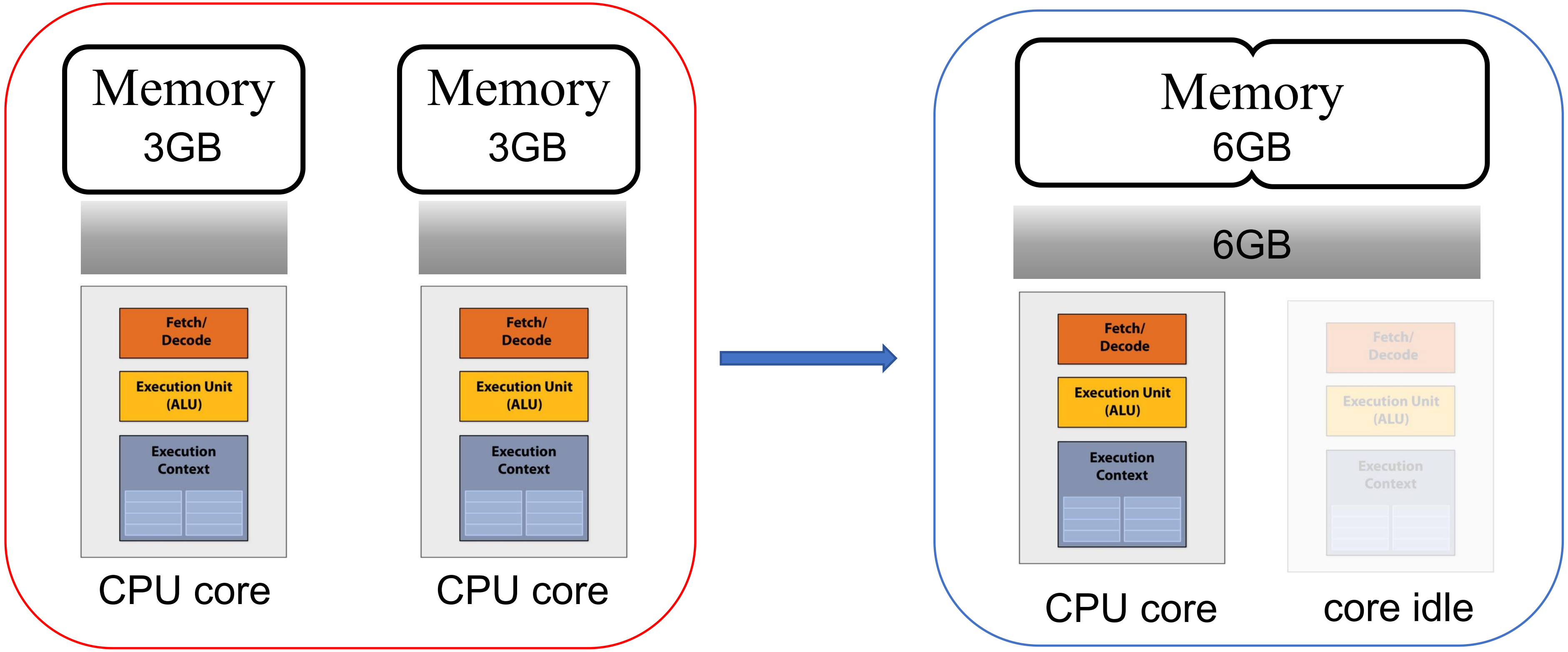


The memory usage after completing the initialization

Memory consumption vs data center configuration

Memory consumption

During the Muon simulation： >3GB



Data center configuration

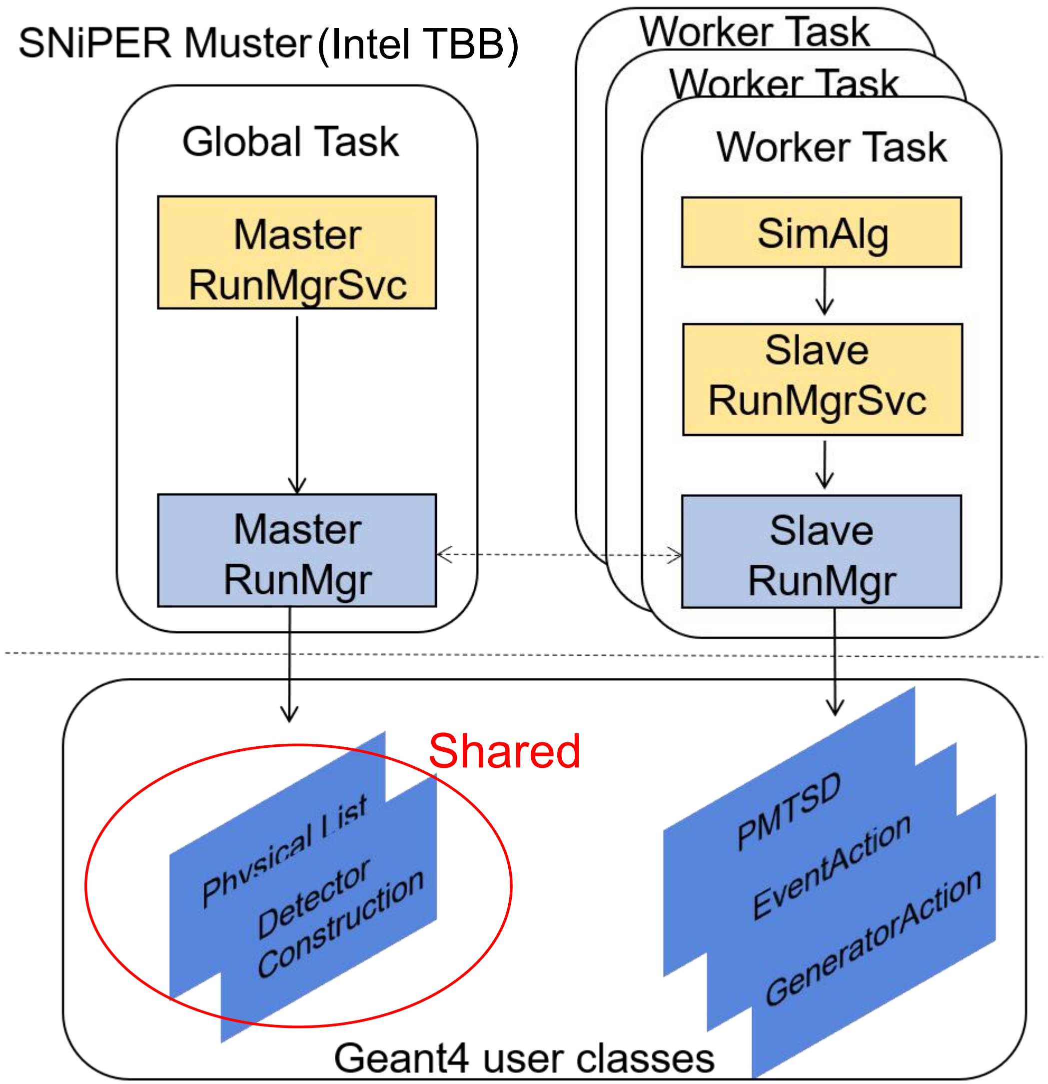
hep_sub -mem (>3000M) Muon.sh

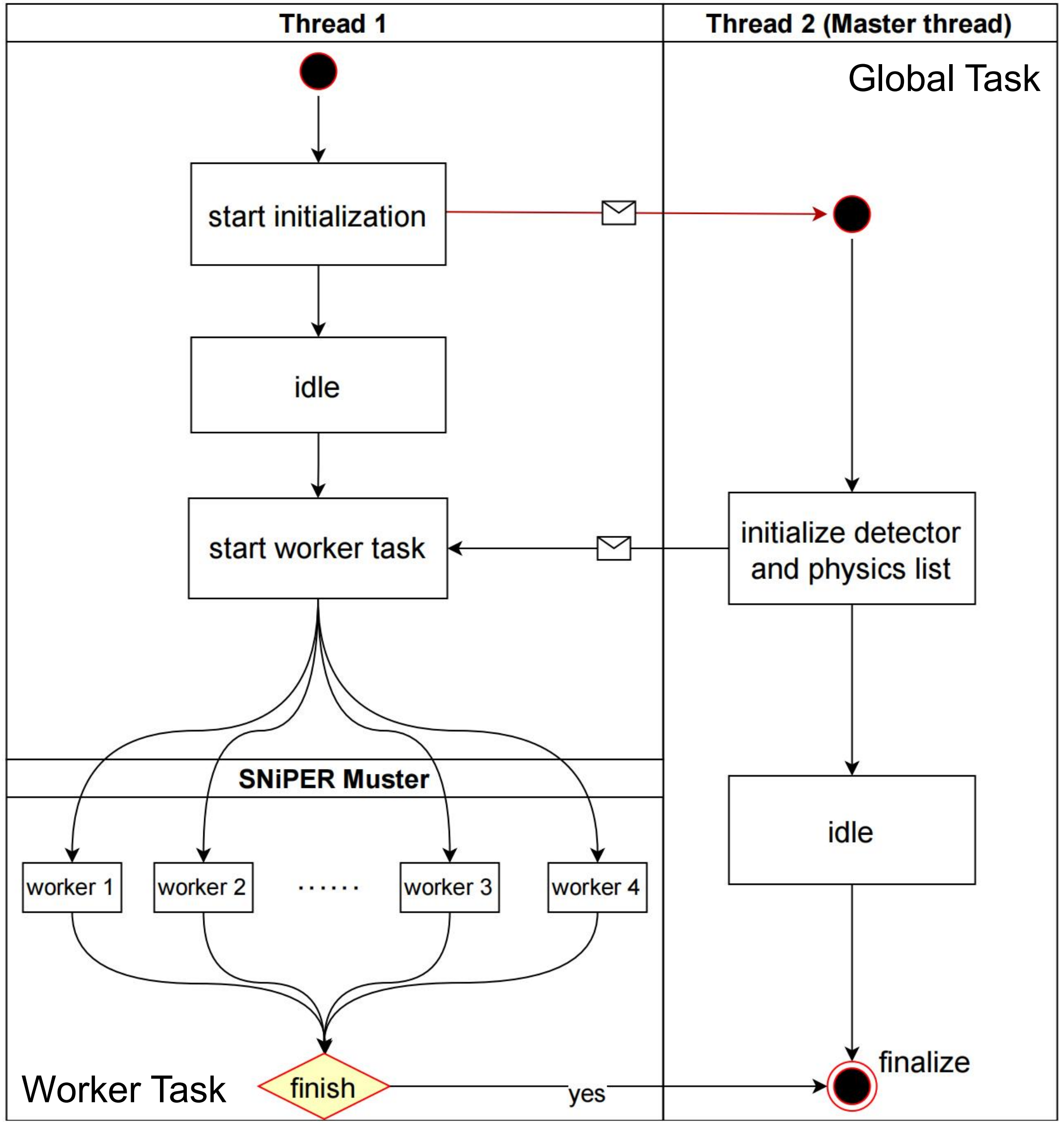
Allocate 3GB of memory for each CPU core

Assign two cores to meet the memory requirements of the job

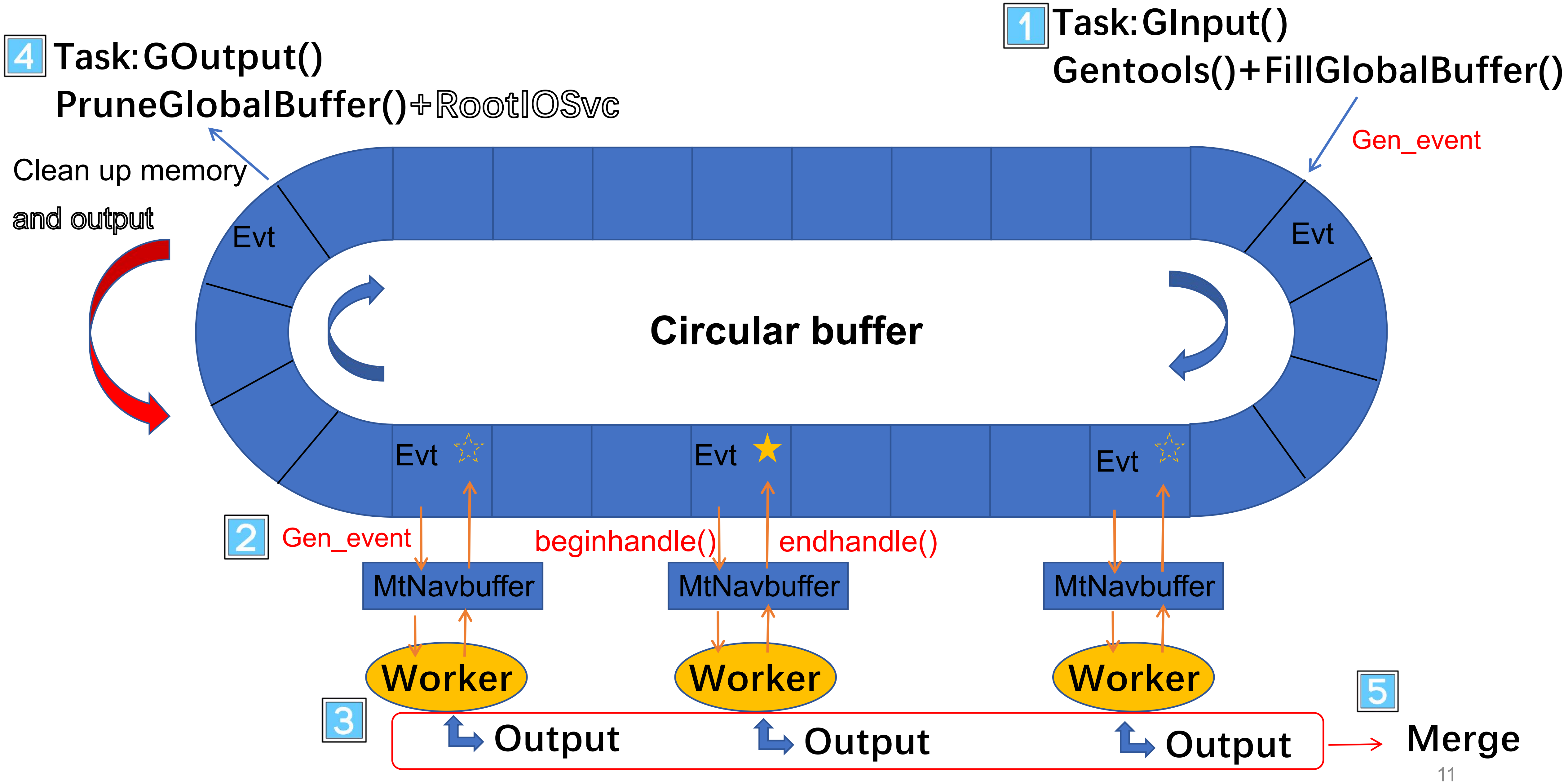
Multi-threaded detector simulation

The design of MT detector simulation





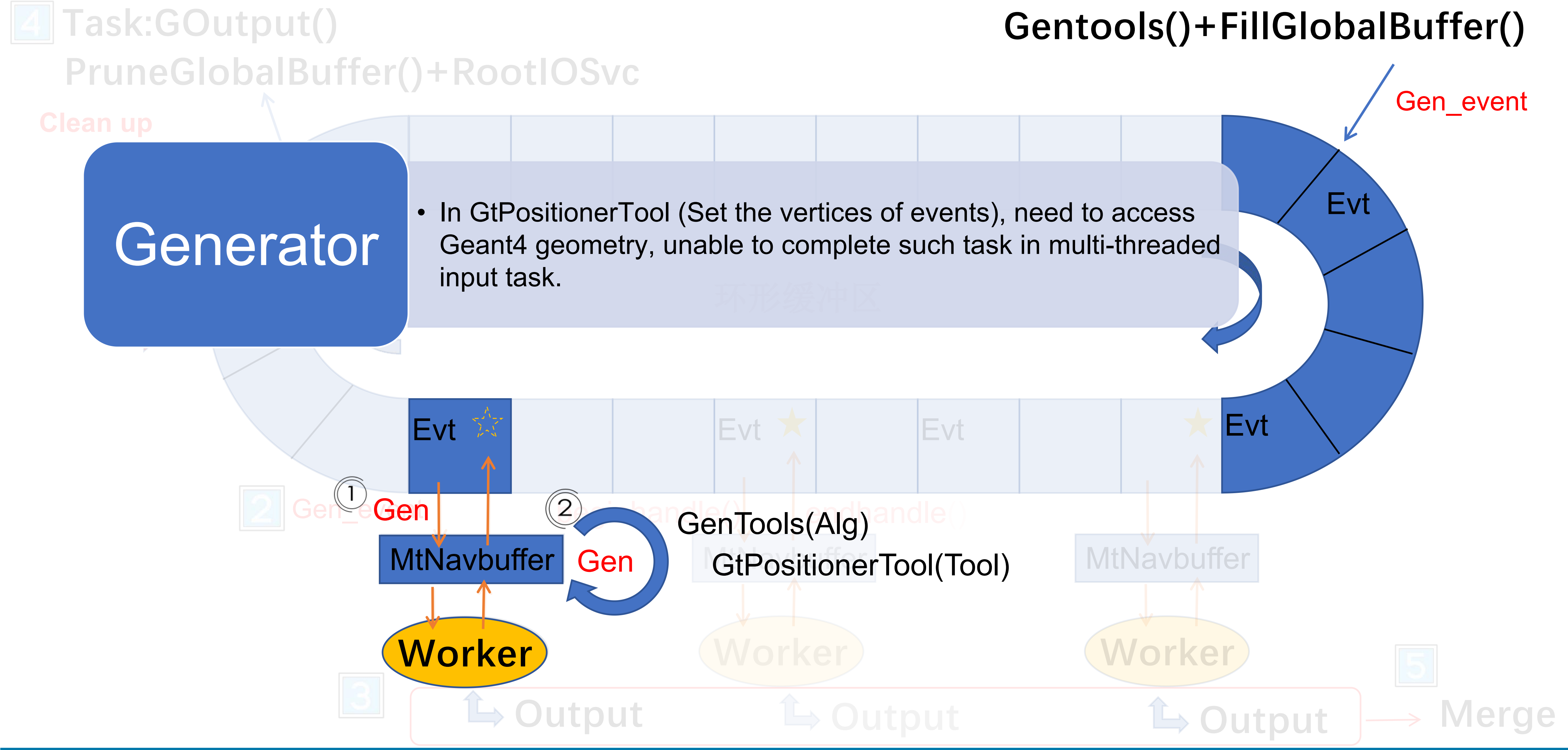
Multi-threaded detector simulation



Multi-threaded detector simulation

Event generation in multi-threaded simulation

1 Task:GInput()
Gentools()+FillGlobalBuffer()

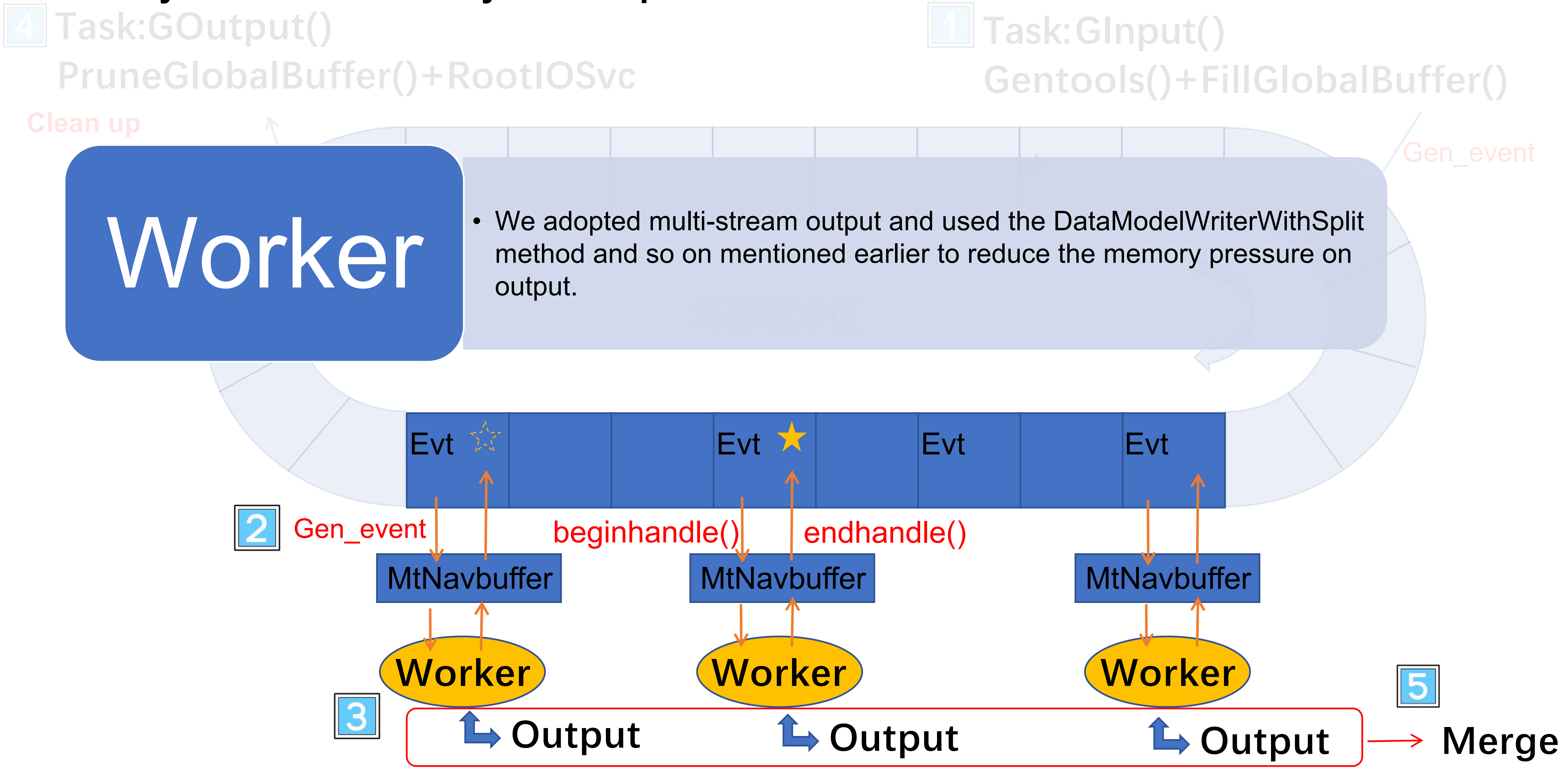


Multi-threaded detector simulation

Other ways to reduce memory consumption in multi-threaded detector simulation

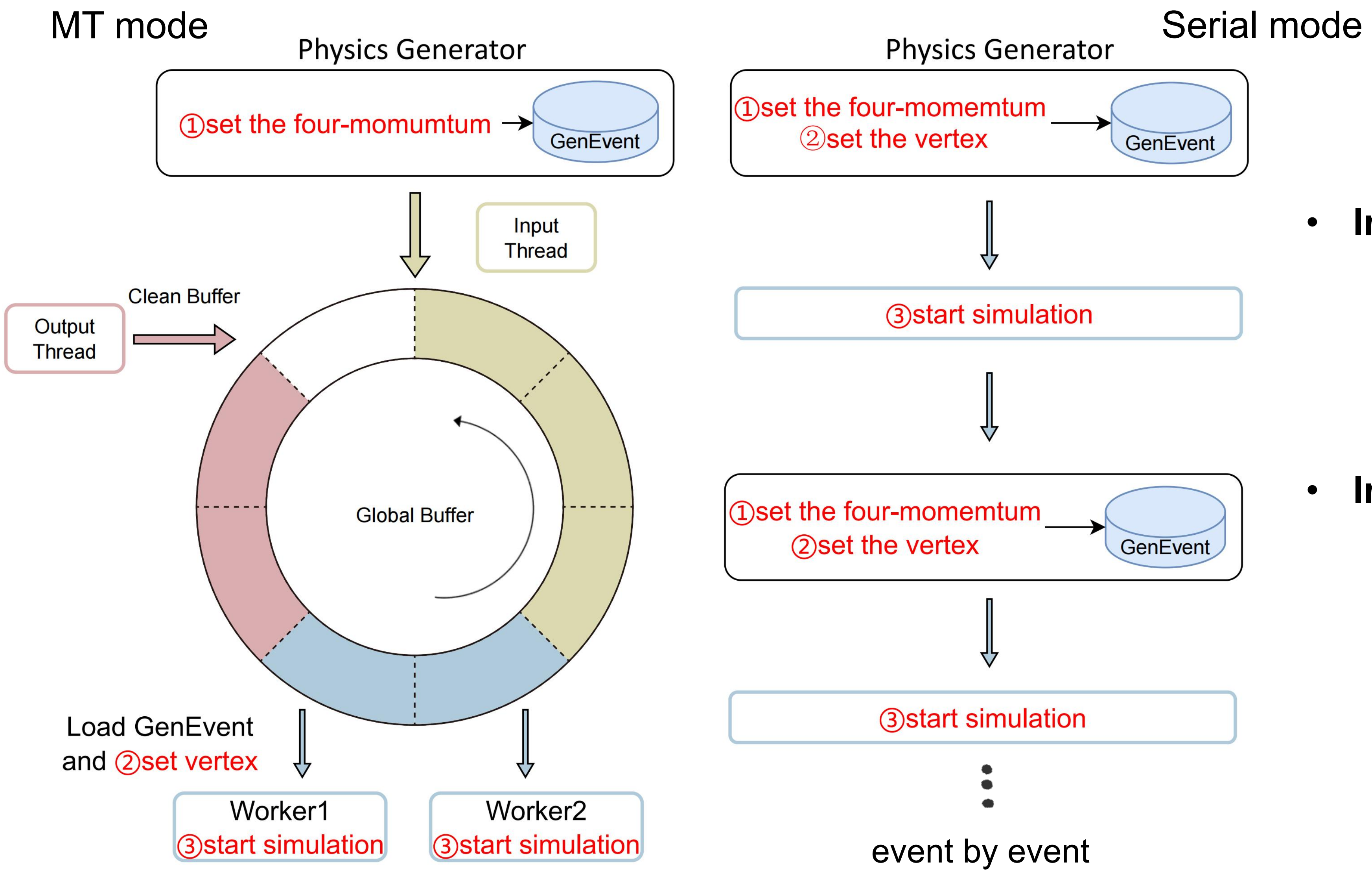
Worker

- We adopted multi-stream output and used the DataModelWriterWithSplit method and so on mentioned earlier to reduce the memory pressure on output.



Multi-threaded detector simulation

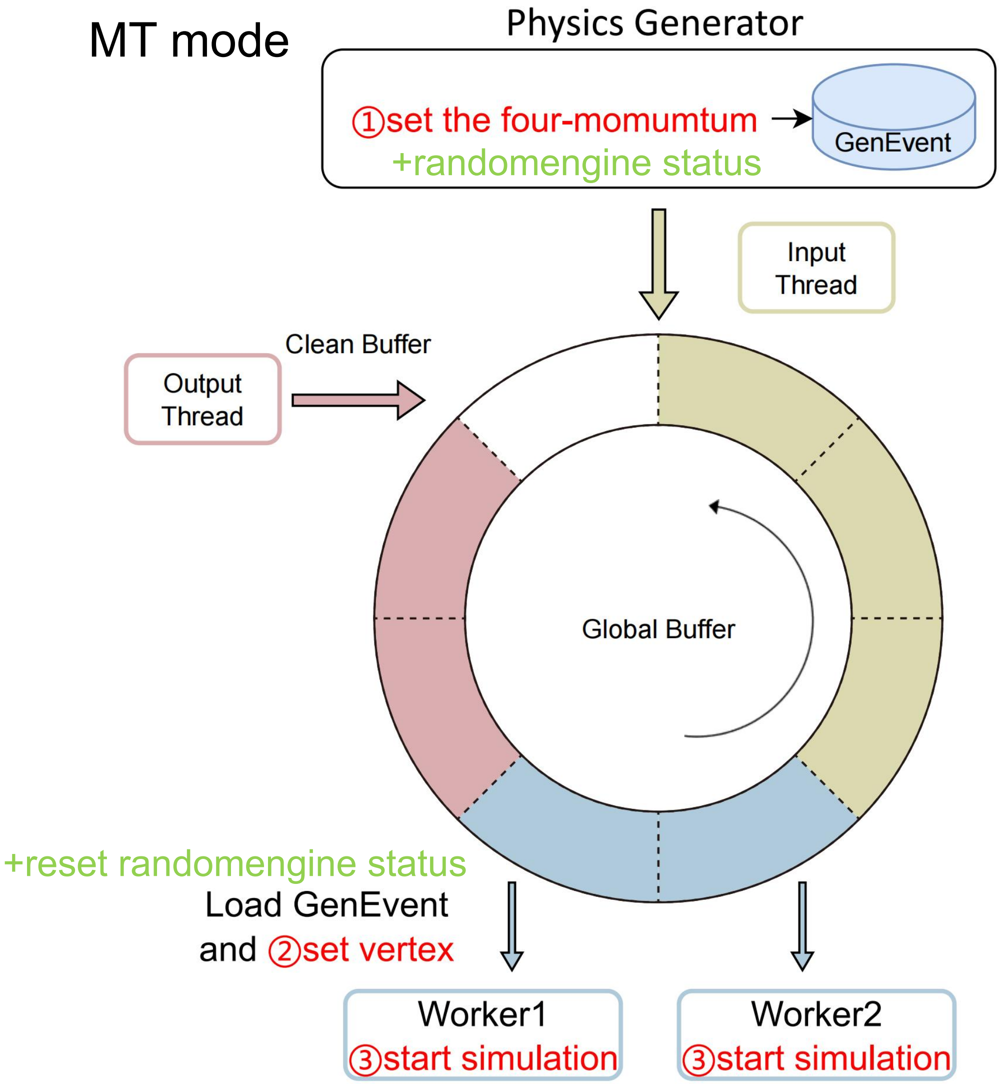
The unity of results between multi-threading and single-threading

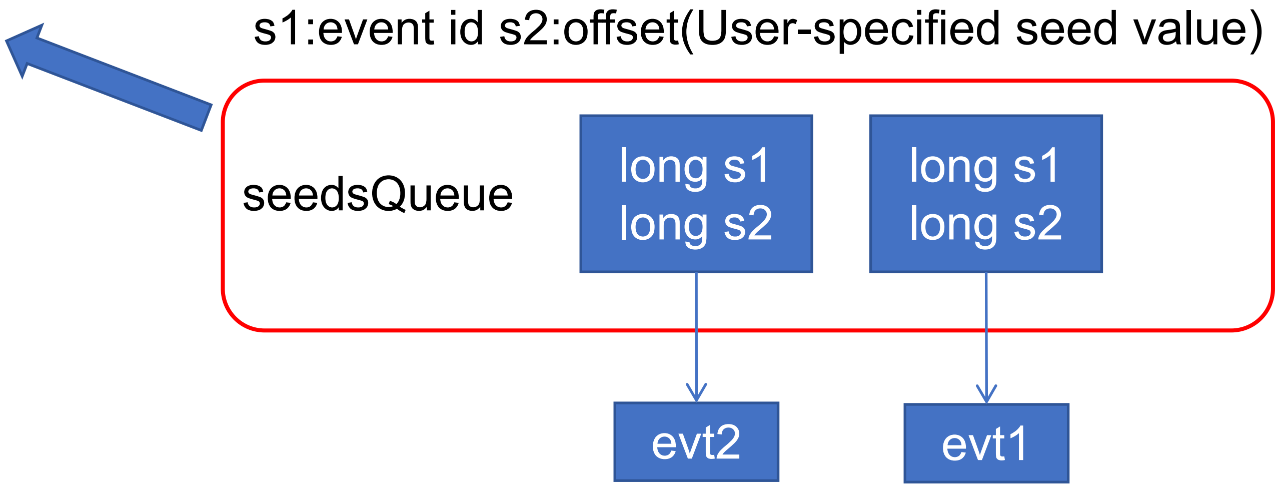


- **In serial mode**
  - Set the seed value at the beginning of the program
  - The result can be reproducible
- **In MT mode**
  - Each worker has its own unique random number sequence
  - The result is not reproducible
  - Inconsistent with single-threaded results.

Multi-threaded detector simulation

The unity of results between multi-threading and single-threading

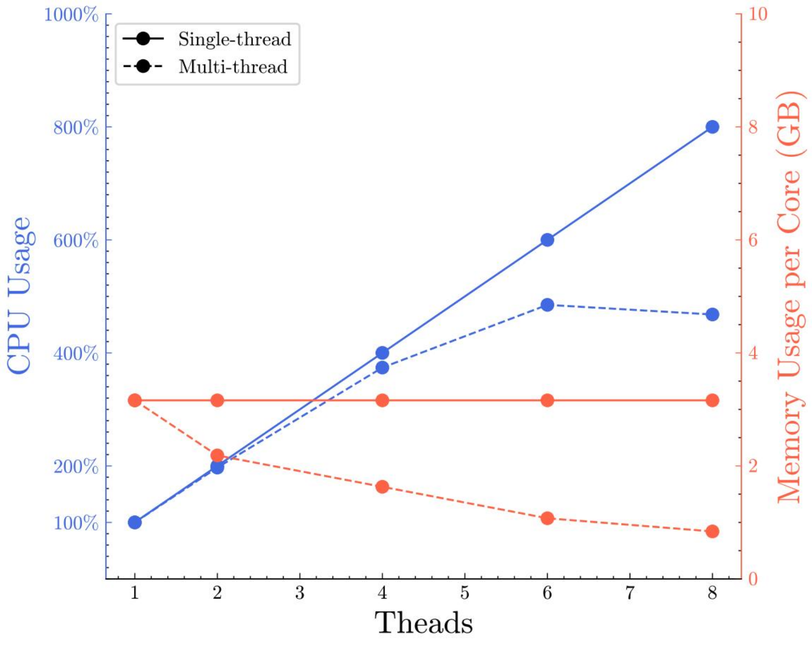




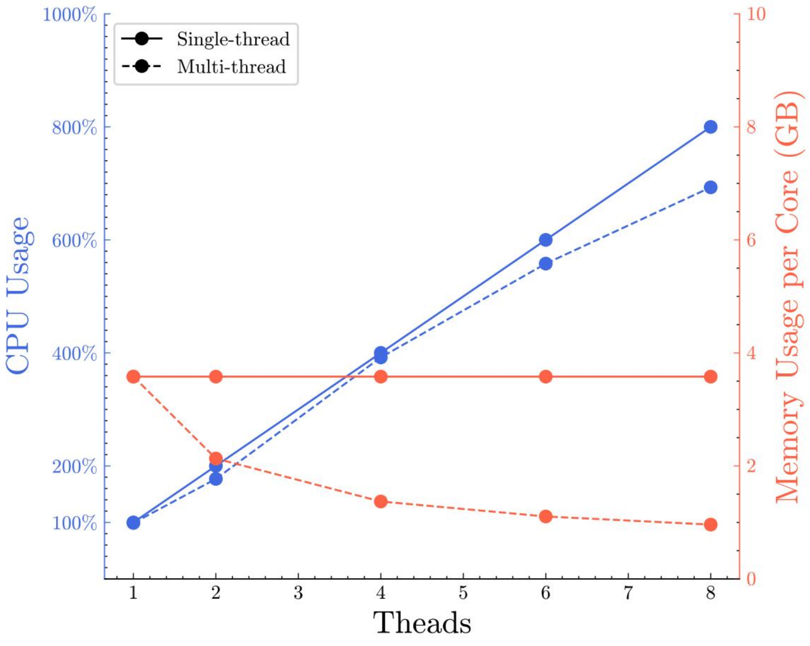
- **Management of random number**
  - Use eventid and offset value to determine the random number sequence of events
  - Reset the random number sequence before setting the four-momentum
  - After completing the setting, save the status of the random engine :std::vector
  - Ensured consistency of results between multi-threading and single-threading

Multi-threaded detector simulation performance

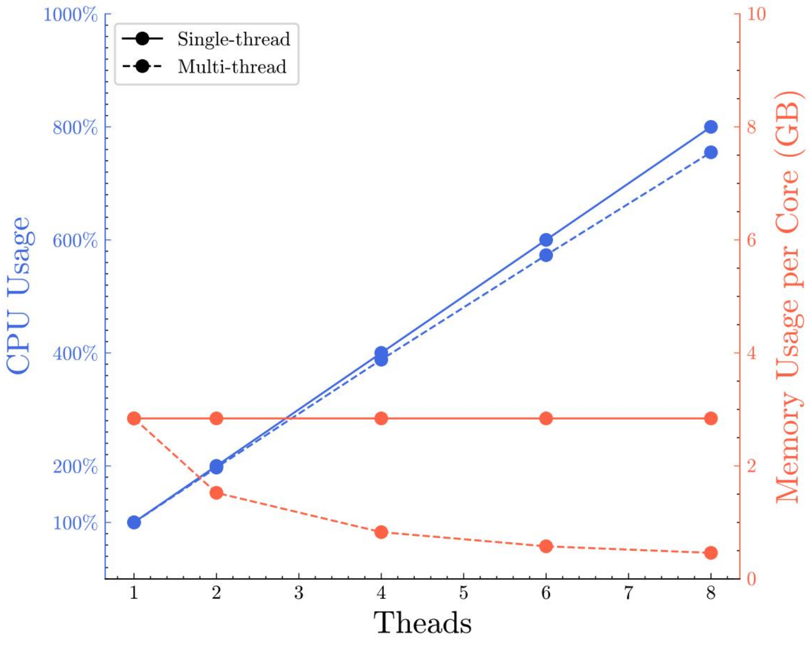
100evts Muon



200evts Muon

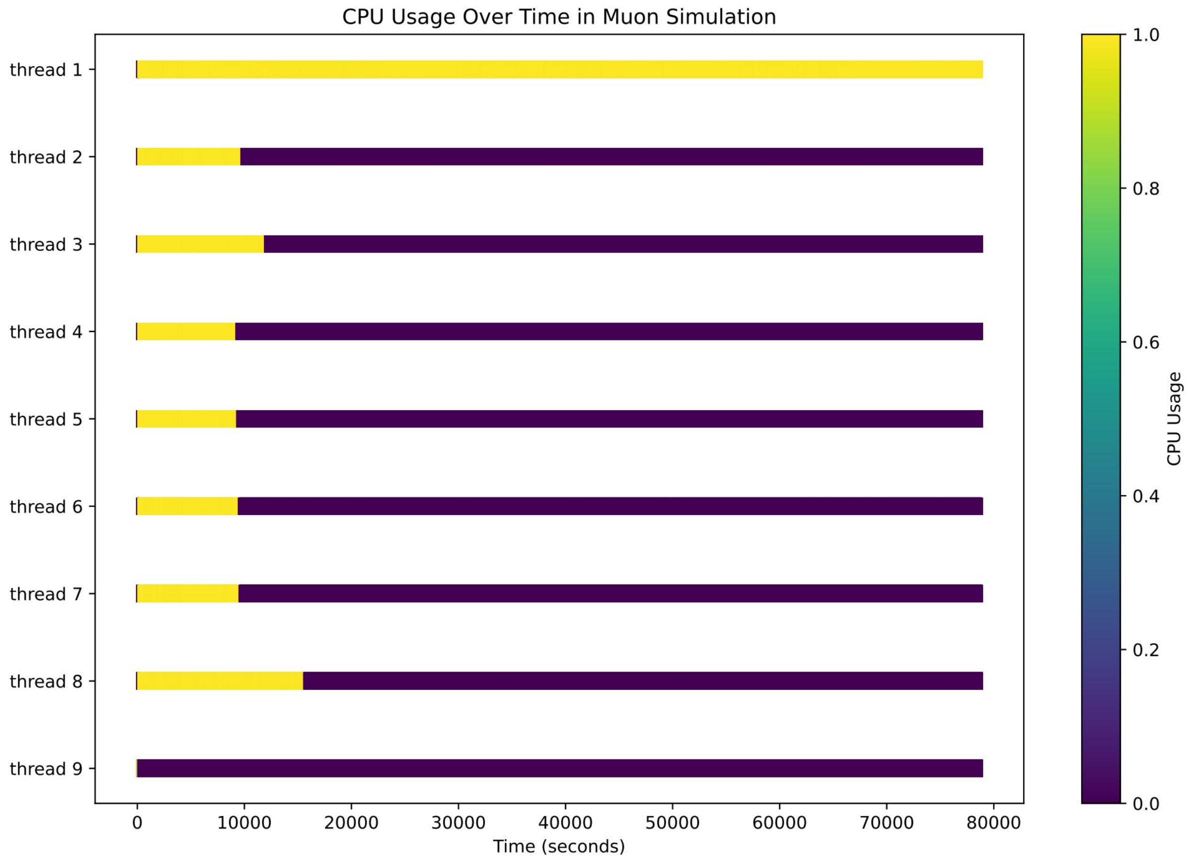


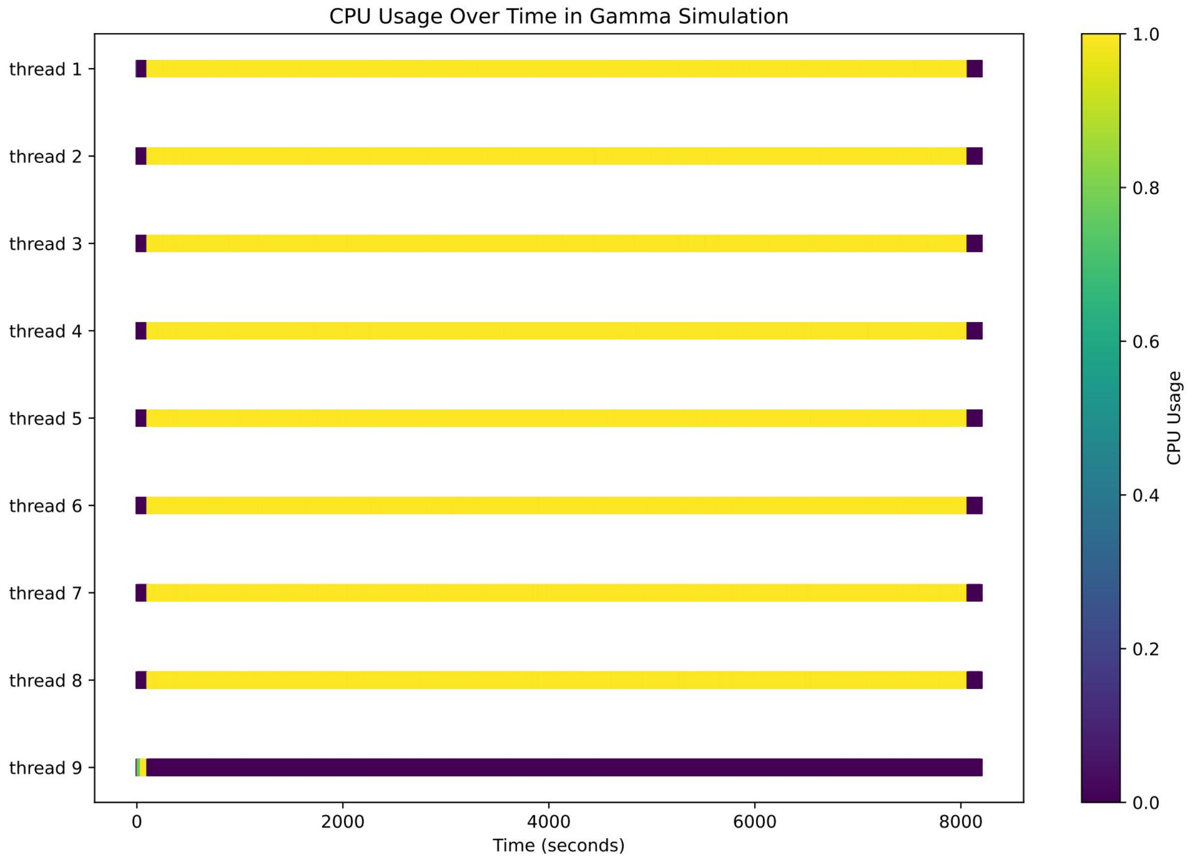
500k evts 1MeV Gamma



- **Performance testing**
  - For 4 threads, ~100% efficiency can be achieved
  - Efficiency decreases significantly when more than four threads are used
  - For events with evenly distributed energy deposition, almost linear acceleration can be achieved
  - Memory consumption has reached the expected target, less than 3GB per core

Multi-threaded detector simulation performance

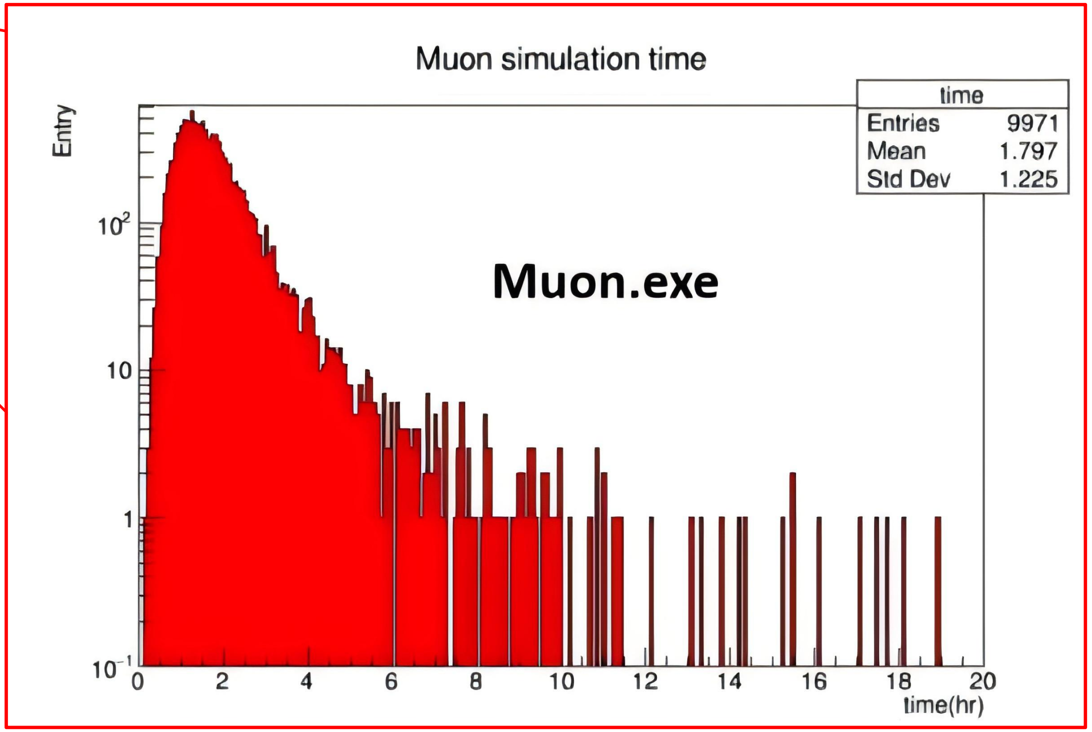




Causes of decreased CPU utilization efficiency

- Different energy deposition in LS event by event, resulting a significant difference in simulation time
- At the end of the program, other threads need to wait for the completion of simulating a high-energy event in one thread.

Multi-threaded detector simulation performance

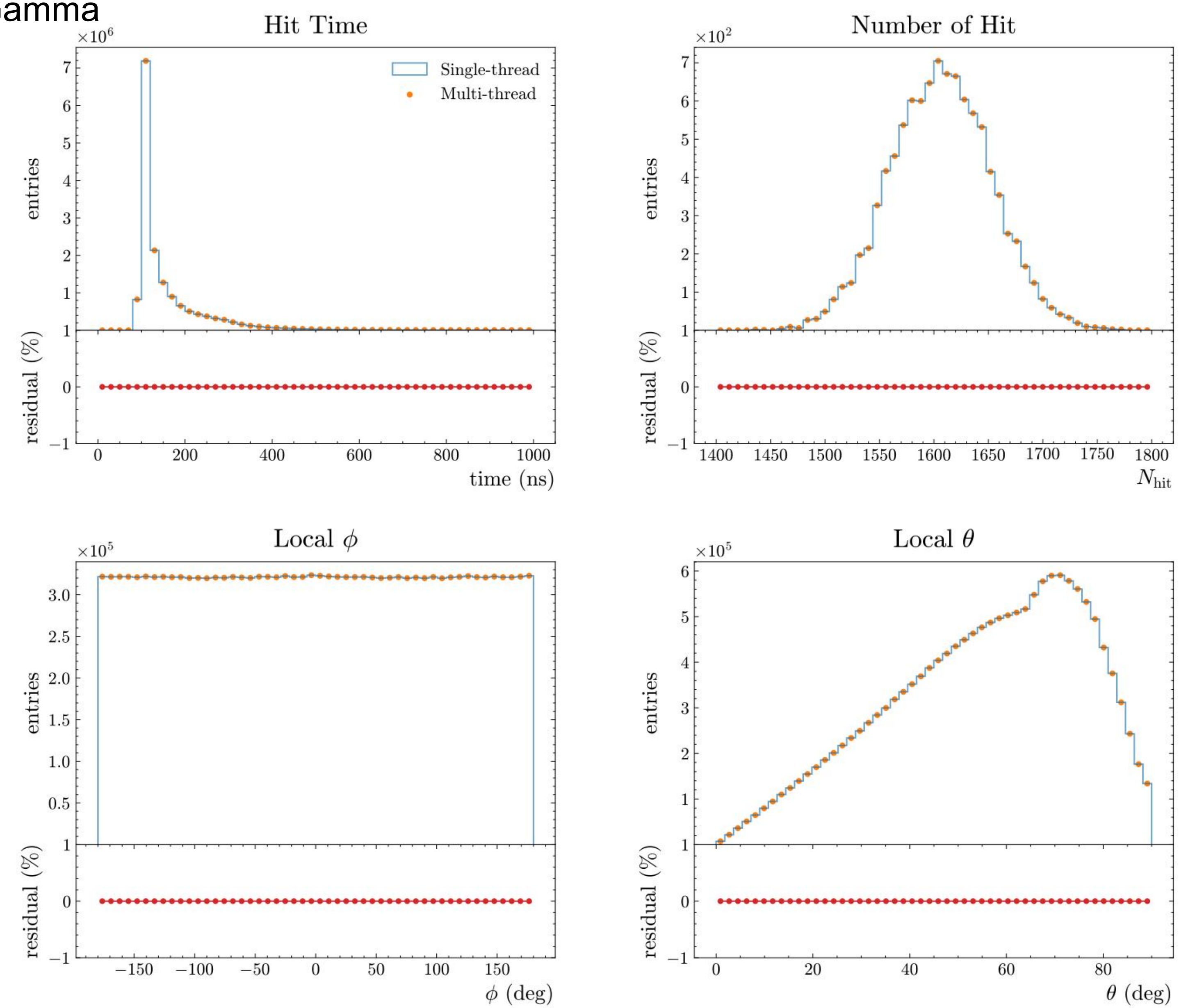


Possible solutions

- **Make a pre-select,** for events with energy deposition higher than a certain value, save the seed first, do not implement optical photon simulation, and then simulate this type of events later, for example, **based on Opticks@GPU**

Multi-threaded detector simulation performance

Test sample：100k evts 1MeV Gamma

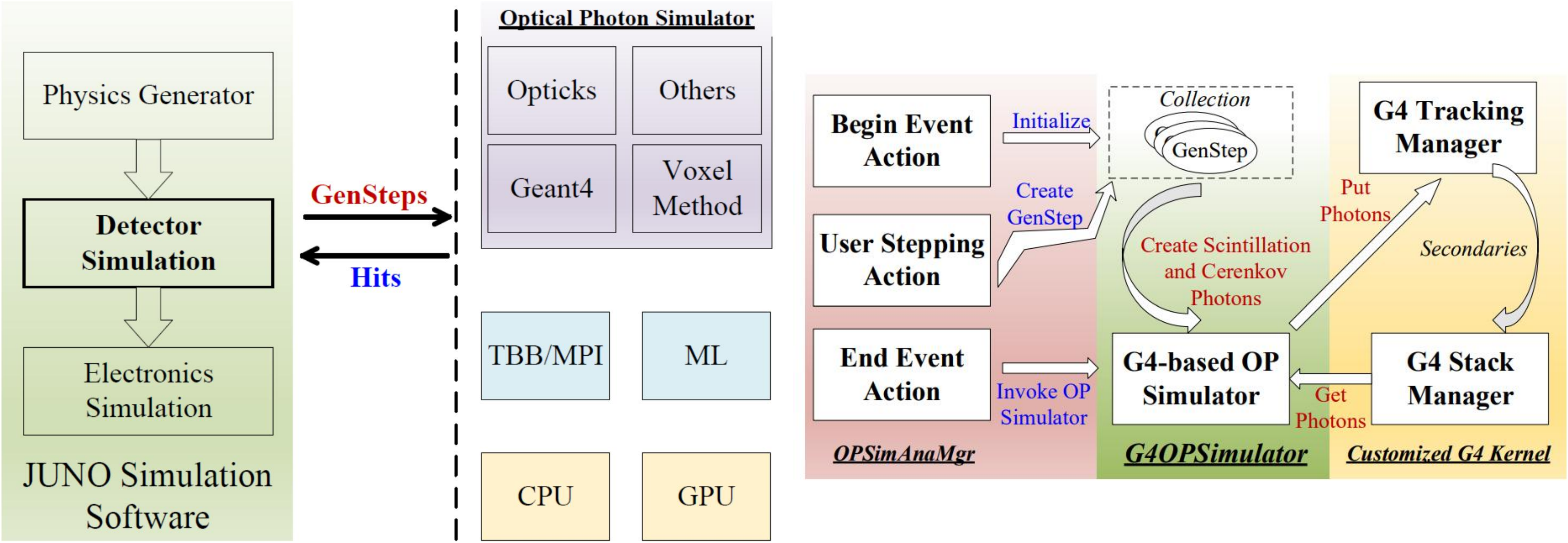


The result is consistent

Summary

- **Development of multi-threaded simulation software**
  - SNiPER Muster and Geant4 run manager kernel are integrated
  - Global buffer is used in simulation to maintain thread synchronization
  - The setting of the random number engine in multi-threading has been completed, ensuring consistency between the results of multi-threading and single-threading
  - Reduce memory pressure of multiple stream outputs using the DataModelWriterWithSplit method and so on
- **Performance of multi-threaded detector simulation software**
  - Ensured consistency of results between multi-threading and single-threading
  - Achieved a close to linear speedup when the simulation time of events is relatively uniform
  - The memory per core has been reduced to below 3GB, solving the problem of wasted computing resources
  - Prepared for large-scale MC production for JUNO

Backup



OP simulator[1]

[1]J.Phys.Conf.Ser. 2438 (2023) 1, 012078