The Application of SNiPER to the JUNO Simulation

Guofu Cao, Ziyan Deng, Xiao Fang, Xingtao Huang, Weidong Li, Tao Lin, Zhengyun You, Jiaheng Zou

CHEP 2016
10-14 October 2016
San Francisco Marriott Marquis
Outline

• Introduction
  • JUNO experiment
  • JUNO offline software and SNiPER

• JUNO Detector Simulation Framework
• JUNO Electronics and Digitization Simulation
• Conclusion
JUNO Experiment

- **Location:** Jiangmen, Guangdong, China
- **Status:** Under Construction

**Central detector**
- Acrylic sphere + 20kt Liquid Scin. + ~17,000 20” PMT + ~34,000 3” PMT
- Water Cherenkov ~2000 20” PMT

**AS:** Acrylic sphere; **SSLS:** stainless steel latticed shell

**AS:** ID35.4m; **SSLS:** ID40.1m; **D43.5m**

Mainly includes:

- Framework, Simulation, Calibration, Reconstruction

Software Development Environment

- Language: C++ and Python 2
- OS: Linux (Scientific Linux 6, gcc 4.4)
- Framework: SNiPER
- Simulation: Geant4 9.4
- Dependent: XercesC, CLHEP, HepMC
- Software management: CMT & SVN
- Installation Tool: junoeenv (bash)

~ 10 official versions have been released, the latest one is J16v2r1-Pre3
SNiPER [CHEP2015, ICHEP2016]
Software for Non-collider Physics Experiments

- Task, a lightweight application manager, controls Event Loop.
- Other tasks could be executed by Incident mechanism.

**User’s Application Layer**
- In an **User Algorithm**:
  1. get data from memory
  2. calculation
  3. put results back to memory
- No need to care where the data comes from
- No need to care where the data will go
- Prepare data to be processed
- features such as geometry ...
- Collect algorithm results

**Core Software Layer**
- I/O: disk, DB, network, grid...

**Python UI Layer**
- run a batch job or interactively debug a module

**Figure 1**: An overview of SNiPER

**Figure 3**: Conditional execution of algorithm subsets
Detector Simulation Framework

- Integrate SNiPER & Geant4 (9.x).

**Features:**
- Lightweight simulation framework.
- Easy to migrate from standalone application.
- Support both batch and interactive modes.
- Support Geant4’s macro files/commands in Python.

**Plan:**
- Supporting multi-threading Geant4 (10.x).
Geometry Management (1)

- GDML description including 3” PMTs, node size ~54k PMTs, ~330k nodes
- Geant4 generation and ROOT conversion time ~30 sec
- File size: GDML file 22 MB, ROOT file 3 MB

The Application of SNiPER to the JUNO Simulation
Geometry Management (2)

- Most of detector elements is constructed from code.
  - Each part could be configured and constructed at runtime.
  - Useful for detector design and calibration units.

- JUNO prototype detector could be setup easily using the same simulation software.
MC Truth and User Action

- Modular design of User Action.

- Event Data Model (ROOTIO)
  - Separation of Header and Event, for lazy loading.

- Design, Pluggable and configurable.

- Other MC Truth are saved in plain TTrees using RootWriter.
Electronics and Digitization Simulation

- “PULL” workflow (data-driven)
  - Starts from ReadoutAlg.
  - Using SNiPER’s Task+Incident to call TriggerSimAlg when buffer is empty.

- Handle event mixing and event splitting gracefully.
  - Such as IBD events (prompt and delay signals)

- Each part is designed and implemented as a module.

```python
python tut_det2elec.py \
  --input IBD:ibd1.root \
  --input U:u.root \
  --rate IBD:1.0 \
  --rate U:3.0
```
Data validation and production

“JunoTest Production” supports both validation and production.
“.ini file” is used for configuration.

- Full chains of simulation and reconstruction are ready.
- First round validation is on going.
- Data Production will start soon.

Validation.ini

[JChain]
seed = 42
evtxmax = 500
njobs = 1
tags = e+_0.0MeV e+_1.398MeV

Production.ini

[JChain]
seed = 42
evtxmax = 1000
njobs = 100
tags = e+_0.0MeV e+_1.398MeV
Conclusions

• JUNO uses SNiPER as underlying framework.
• Based on SNiPER, simulation software was developed.
  • Detector simulation framework integrates SNiPER and Geant4.
  • Electronics simulation supports “PULL” workflow, allows hit level mixing.
• The full data processing chain is completed.
• MC Data challenge is preparing, will start soon.