Updates of HPS MC

Tongtong Cao

November 19th, 2020
Online HPS Collaboration Meeting

Outline

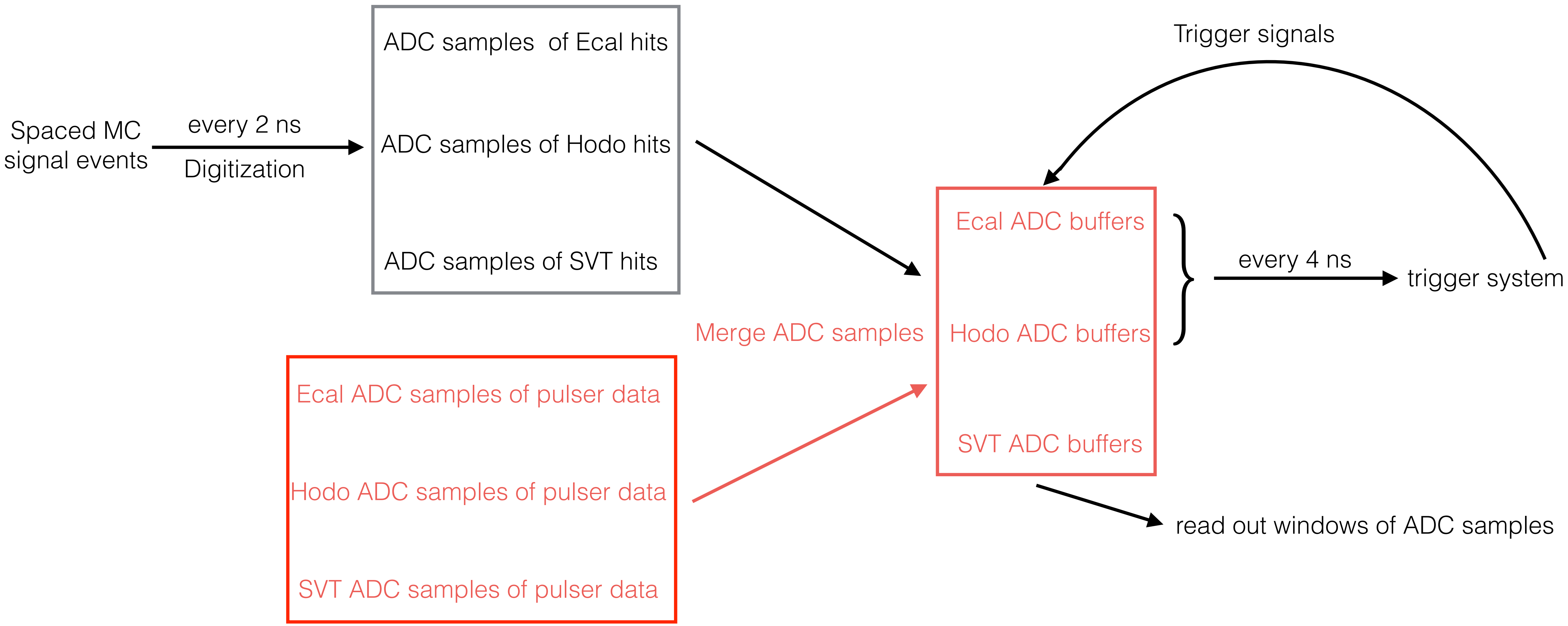
- Readout system update for beam background merging
- Job system update for hps-mc
- Software updates for MC of $mu^{+}mu^{-}$ trident
- Status of 2019 MC production
- Readiness for 2019 MC

Beam Background Merging: Introduction

- To take into account effects of beam background in MC, signal events (ap, rad, tritrig and wab) are required to be merged with beam background.
- For current MC processing,
    - Beam bunches are produced by EGS5+SLIC.
    - Signal events are produced by MadGraph+SLIC.
    - Produced signal events are filtered and spaced (interval between each two signal events = 250) firstly, and then merged with beam bunches event by event.
- It is time consuming to produce enough beam bunches to merge with spaced signal events. To significantly speed up MC production and improve quality of MC samples, we plan to merge pulser data as background with MC signal samples.

Beam Background Merging: Scheme of New Readout System

The readout system is emulation of the DAQ system



Spaced MC signal events

every 2 ns

Digitization

ADC samples of Ecal hits

ADC samples of Hodo hits

ADC samples of SVT hits

Ecal ADC samples of pulser data

Hodo ADC samples of pulser data

SVT ADC samples of pulser data

Merge ADC samples

Ecal ADC buffers

Hodo ADC buffers

SVT ADC buffers

every 4 ns

trigger system

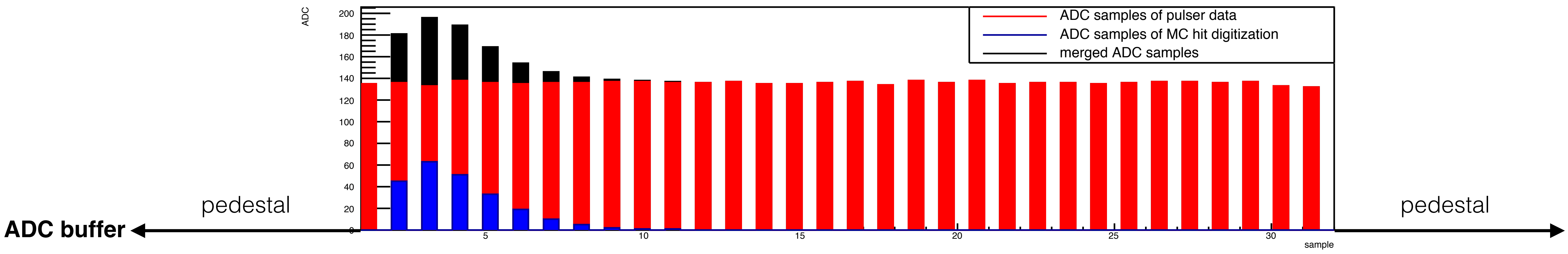
Trigger signals

read out windows of ADC samples

Beam Background Merging: How to merge

- If ADC samples of a channel in buffers is directly from pulser data, noise and pedestal are not be added.
- If ADC samples of a channel in buffers from merging of MC sample and pulser data, noise and pedestal are not be added
- If ADC samples of a channel in buffers is directly from MC sample, noise and pedestal are added.

ADC samples for a hodoscope channel



Beam Background Merging: Effects of Merging on Trigger

- 1000 tritrig events of 2019 MC are used for tests:
    - With beam background merging (run new system with singles3 trigger): 105 triggered events
    - Without beam background merging (run old system with singles3 trigger): 103 triggered events
- Comparing event numbers between two files from output of the two readout systems, where event numbers in the files are identical to event numbers in SLIC files.
    - Event numbers of **101** triggered events are the same. Note: readout output could be different between events with the same event number. For events with merging, ADC samples could be changed due to merging, or extra channels from pulser data are read out
    - 4 triggered events exist in the file with merging, but do not in the file without merging
        - 3 events have no hodoscope hits in MC data
        - 1 event has a hodoscope hit in MC data, but energy is too small to generator singles3 trigger
        - After merging, hits from pulser data are added into the trigger system, and triggers are produced for such events
    - 2 triggered events exist in the file without merging, but do not in the file with merging: due to pulser data merging, distributions of ADC samples in buffers changes comparing to without merging. Further, hit distributions of Ecal and hodoscope after pulse integration could change at some clock-cycles. For these two events, geometry matching fails after pulser data merging.

Beam Background Merging: Status of software updates

- Updates for Ecal and hodoscope have been done.

- Preliminary tests show new Ecal and hodoscope digitization drivers and the trigger system work well.

- Update for the SVT readout driver needs to be supported by SVT experts. Further tests will be taken after the update.

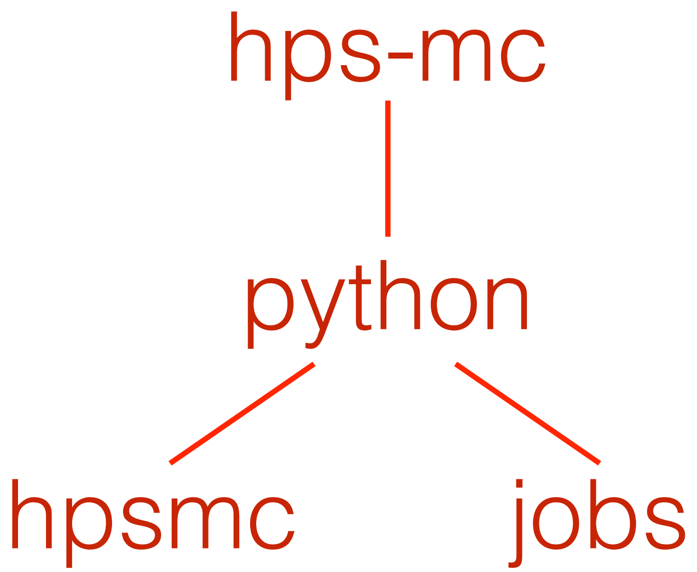
- All new drivers are developed in a new package org.hps.digi of hps-java so that the old readout system keeps available.

Job System Updates of hps-mc: Introduction

- Source of hps-mc: https://github.com/JeffersonLab/hps-mc
- HPS MC production is a complicated proceed with several steps in chains. The job system is developed to let users conveniently take MC production by implementation of commands.
- The system can work for
    - single job
    - batch jobs with four modes:
        - Local mode for local machine
        - Pool mode (run a set of jobs using Python's multiprocessing module) for local machine
        - Auger mode for JLab batch machine
        - LSF mode for SLAC LSF machine
- Thank Jeremy for hard efforts on the job system updates.

Job System Updates of hps-mc: Packages

- Packages for the job system are in hps-mc/python, including
  - package hpsmc:
    - run_param.py: parameter setup for beam, target and ap mass
    - component.py: a base class for components
      - generators.py: son classes for generator components
      - tools.py: son classes for tool components
    - job.py: a class for running single job
    - batch.py: a class for running batch jobs, and four son classes for four different modes
    - job_template.py: a class for producing a json format parameter file with a job list
    - other support scripts
  - package jobs: various job scripts; each job script sets a MC chain with application of generator and tool components


hps-mc
python
hpsmc
jobs

Job System Updates of hps-mc: How to run

- Environment setup: path-of-dir/hps-mc/install/bin/hps-mc-env.sh (hps-mc-env.csh)
- Commands: command scripts are in hps-mc/scripts
    - single job: hps-mc-job run [other-args] [-c configuration-file] [job-script] [parameter-file]
    - batch jobs: hps-mc-batch [mode] [other-args] [-c configuration-file] [job-script] [parameter-file]
- Configuration-file:
    - Includes setup for the Job class and configurations for packages of generators and tools
    - Software installation directories in a configuration file need to be specified by users
    - Multiple configuration files are available in commands, such as -c configuration-file-1 -c configuration-file-2
- job-script: a job script name in hps-mc/python/jobs
- parameter-file:
    - A json format file
    - Setup of job id, input files, output files, directories of input and output files, and parameters for generators and tools;
    - Includes one job for single job, while includes a job list for batch jobs.
    - For batch jobs, use command hps-mc-job-template to apply the class job_template.py for production of a job list using a jinja format template, a jason format file for variable substitutions and input file list(s)
- Examples for application: hps-mc/examples includes examples for application of the job system

Software updates for MC of mu^+mu^- Trident

- MadGrah: In MG4/rad, particle ids for lepton pair are 611 and -611, which are distinguishable with id of beam electron. To generate rad with mu^+mu^- trident, we just need to change such lepton mass as muon mass in the parameter card.

- EGS5: A program *lhe_rad_mu.f* is built to process *mu* trident events with proper setup for building truth information of mother particles.

- Stdhep tools: The tool *add_mother_full_truth* is updated to be applicable for rad_mu events from EGS5.

Status of 2019 Production

- All information about 2019 MC production is well documented in https://confluence.slac.stanford.edu/display/hpsg/2019+MC+Samples

- Samples for the first pass of 2019 MC are produced with 120 nA beam and 20 um target (Cameron has been producing some samples with other setup):

  - pass0 samples: pure or background merged samples for ap, rad, tritrig and wab are produced with singles or pulse triggers; besides, a FEE sample is produced.

  - pass0b samples: after a new filter specified for singles trigger is developed and applied, samples for ap-beam, rad-beam, tritrig-beam and wab-beam with singles trigger are produced.

  - pass0d samples: tritrig-beam samples with looser kinematic limits in generator are produced.

  - For $mu^{+}mu^{-}$ trident, pass0 sample for rad-beam with pulse trigger is produced.

- Samples have been being used for analyses: reach study, calibration, software tests, etc.

Readiness for 2019 MC

- With completion of alignment, calibration and detector geometry update, more accurate parameters for beam and target needs to be extracted and applied into particles from generators before detector simulation

- Complete readout system update for beam background merging

- Update for biasing in detector simulation

- Handle dead/hot channels

- Investigate to use hps-mc on OSG grid resources