HLT Operation in Dress Rehearsal and Global Cosmic Ray Data Taking

Chunhua Li
The University of Melbourne
TRG/DAQ Workshop
2017-08-24

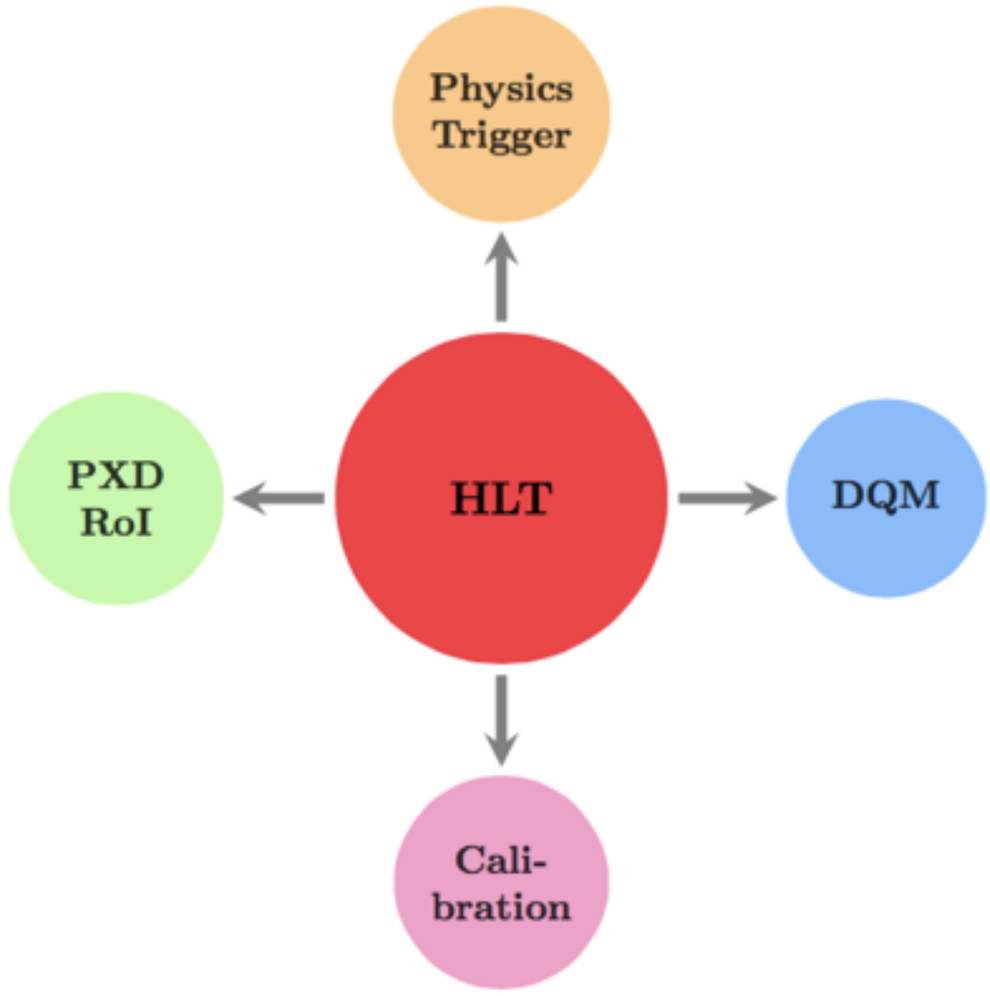
HLT

- Physics Trigger: suppress event rates from 30 kHz to 10 kHz

- PXD RoI: provide HLT trigger result and tracking information of SVD and CDC to calculate Region of Interest of PXD.

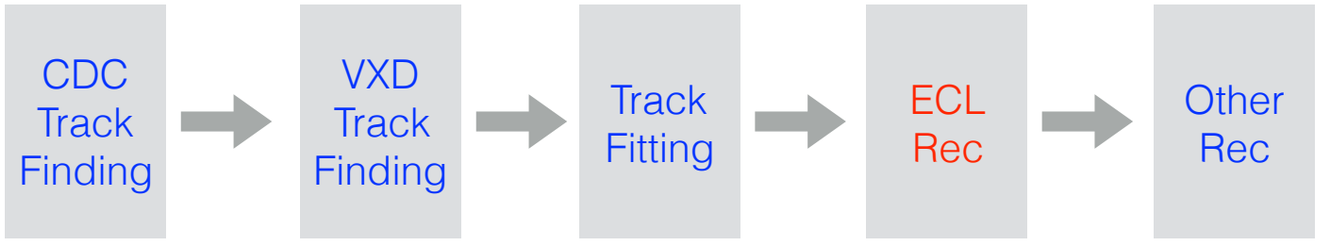
- Calibration: Flag samples for the calibration of detectors

- DQM: Information from Reconstruction for data quality monitoring

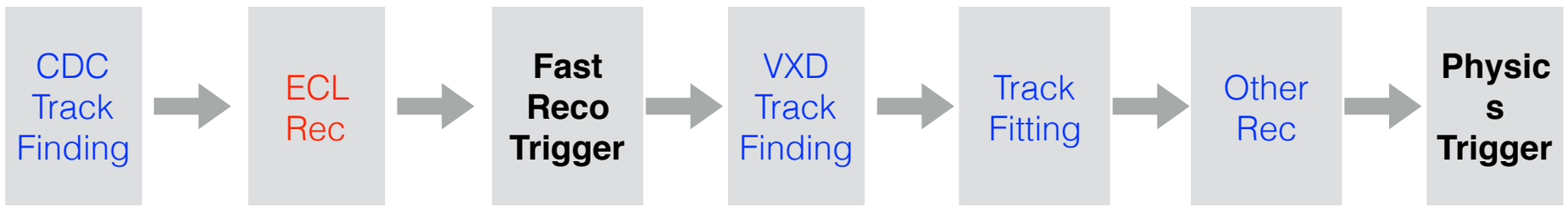


HLT Software

Standard Offline Reconstruction



HLT Standard Reconstruction and Trigger



See the details in Thomas Hauth's talk tomorrow

Local Database

- HLT server is inaccessible from outside KEK network
  - Download central database locally
  - conditionsdb download -c tagname LocalDB
  - all payloads is in LocalDB/database.txt

- The database is download to hlt/example/LocalDB temporarily in dress rehearsal and GCRT, and called in script with

```
basf2.reset_database()
basf2.use_local_database(ROOT.Belle2.FileSystem.findFile("hlt/examples/LocalDB/database.txt"))
```

- In realistic data taking, the database is frequently update with the real-time calibration, how to download the database need to be considered.

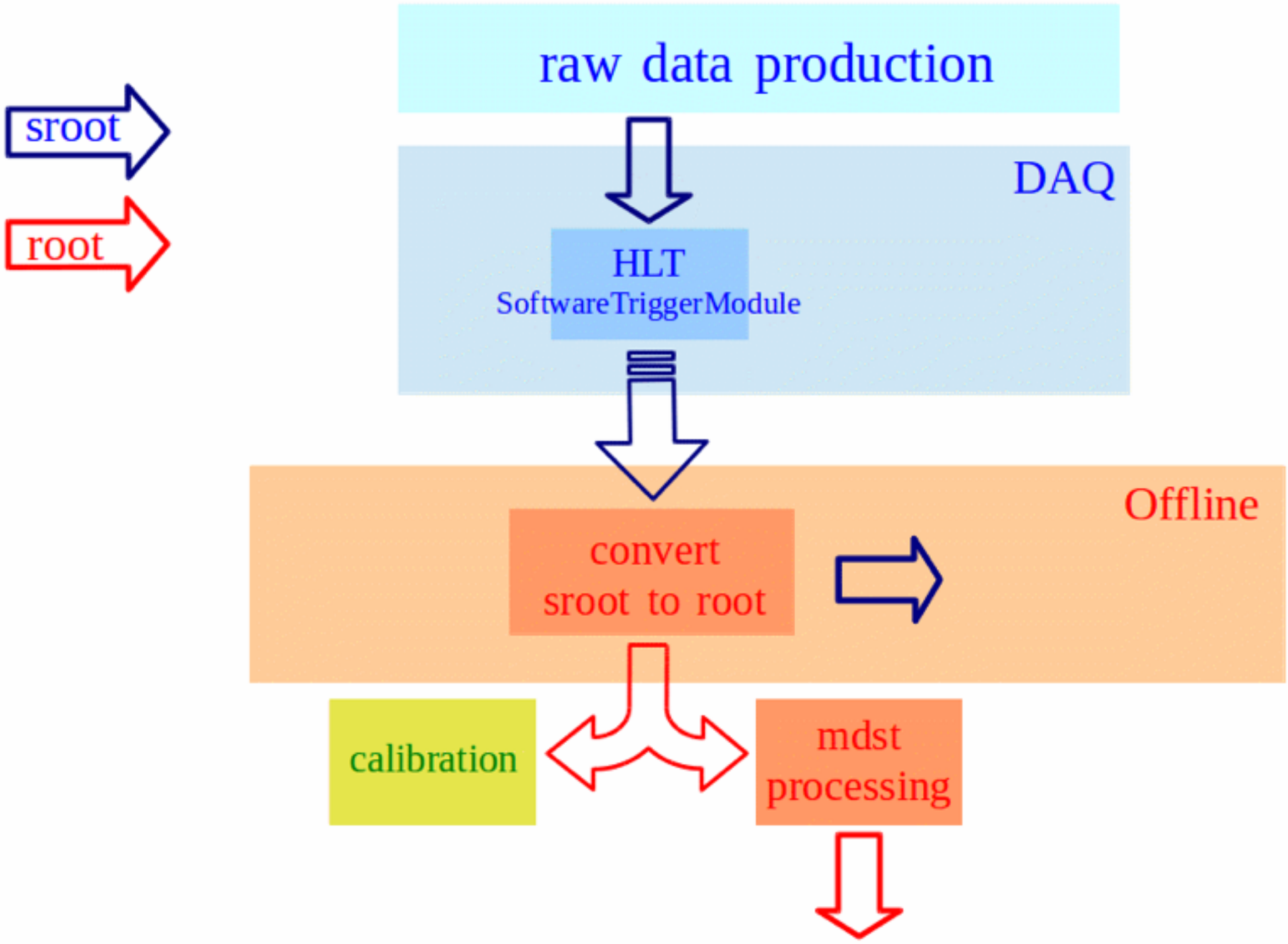
Software Debugging

- Smooth operation of HLT software is essential.

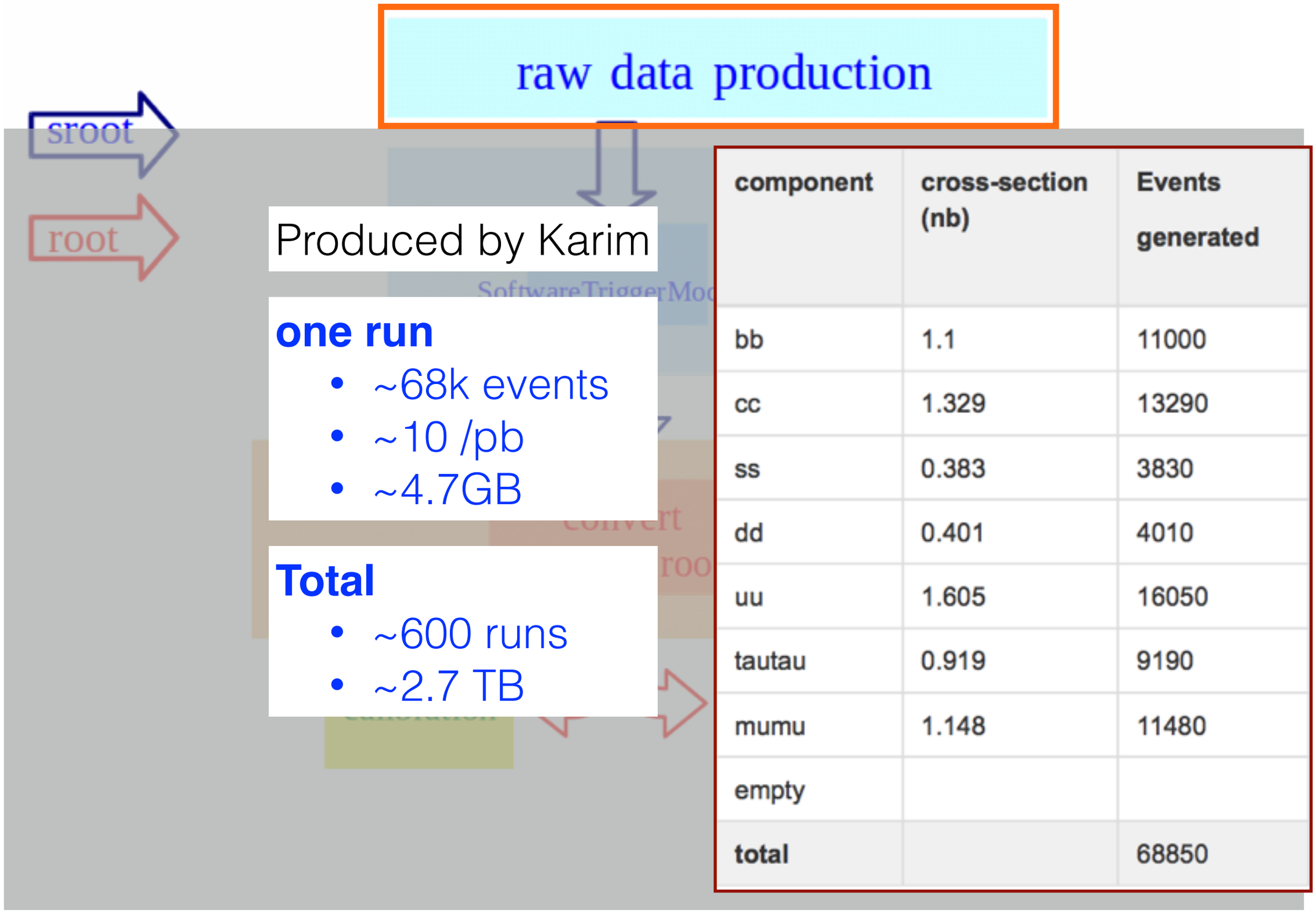
- Procedure of software debugging during dress rehearsal and GCRT
  - Debug the software on HLT test bench before implementing it online.
  - Create JIRA issue to the related people if some problems are found
  - Implement the patched software on HLT

- Crash handling module CrashHandlerModule (by C. Pulvermacher) was developed to make sure the software work smoothly even when the crashes happen during data taking.
- Not applied in dress rehearsal and GCRT due to large impact on the processing time.
- Further optimization for Phase2

Dress Rehearsal



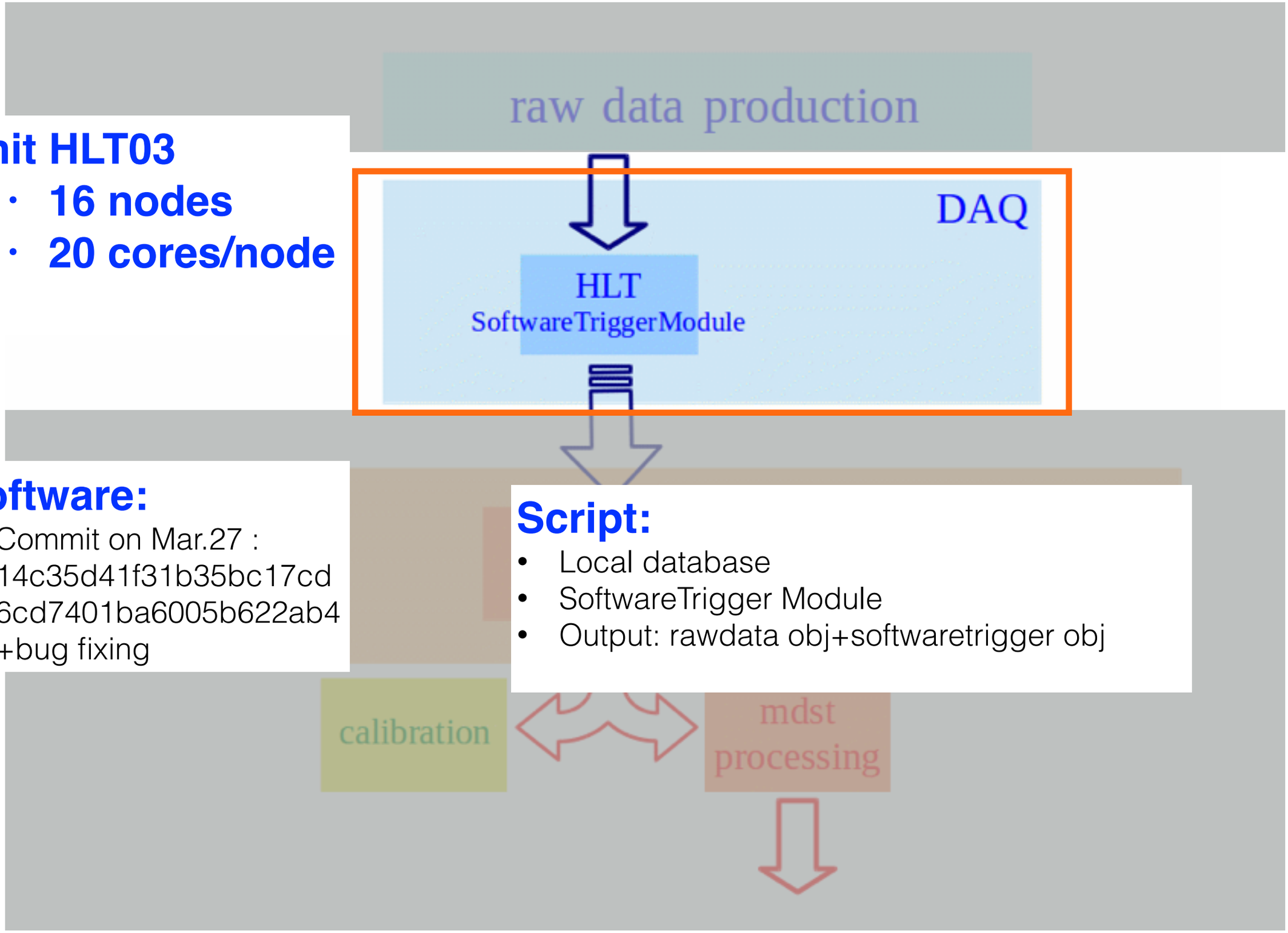
Round1 Dress Rehearsal



component	cross-section (nb)	Events generated
bb	1.1	11000
cc	1.329	13290
ss	0.383	3830
dd	0.401	4010
uu	1.605	16050
tautau	0.919	9190
mumu	1.148	11480
empty		
total		68850

Unit HLT03

- 16 nodes
- 20 cores/node



Software:

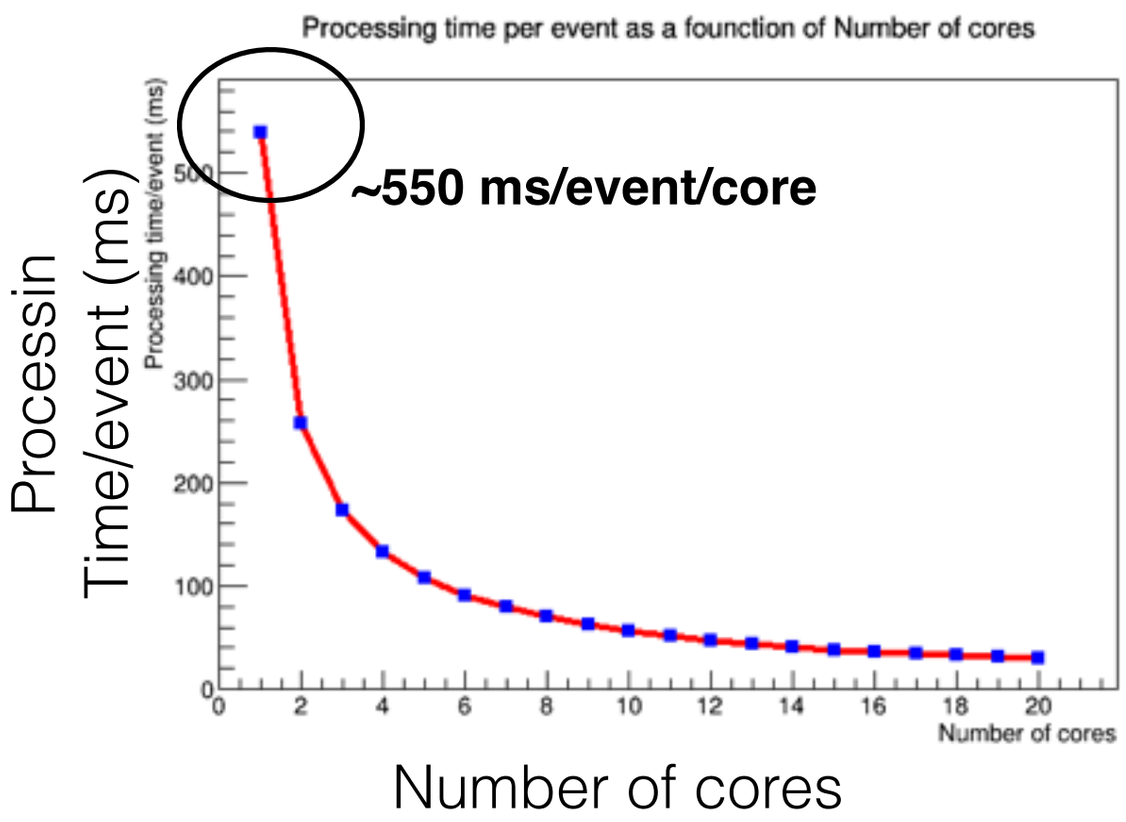
- Commit on Mar.27 : 14c35d41f31b35bc17cd6cd7401ba6005b622ab4
- +bug fixing

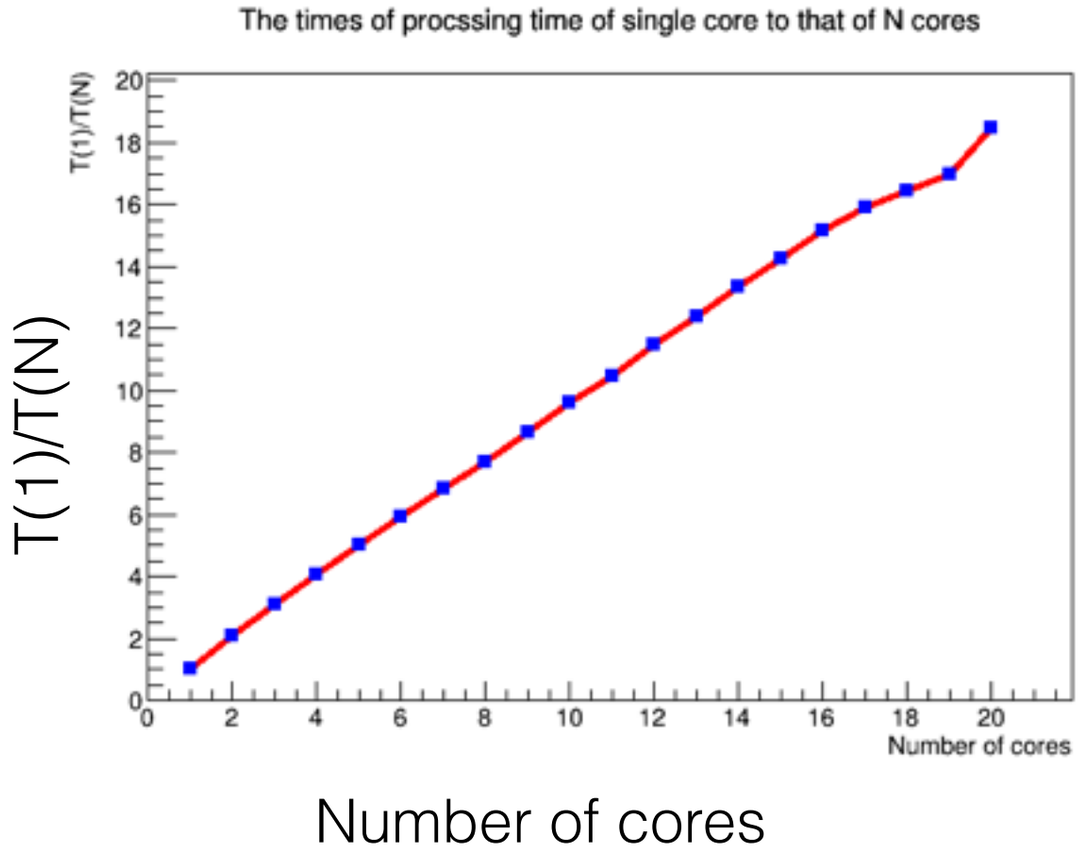
Script:

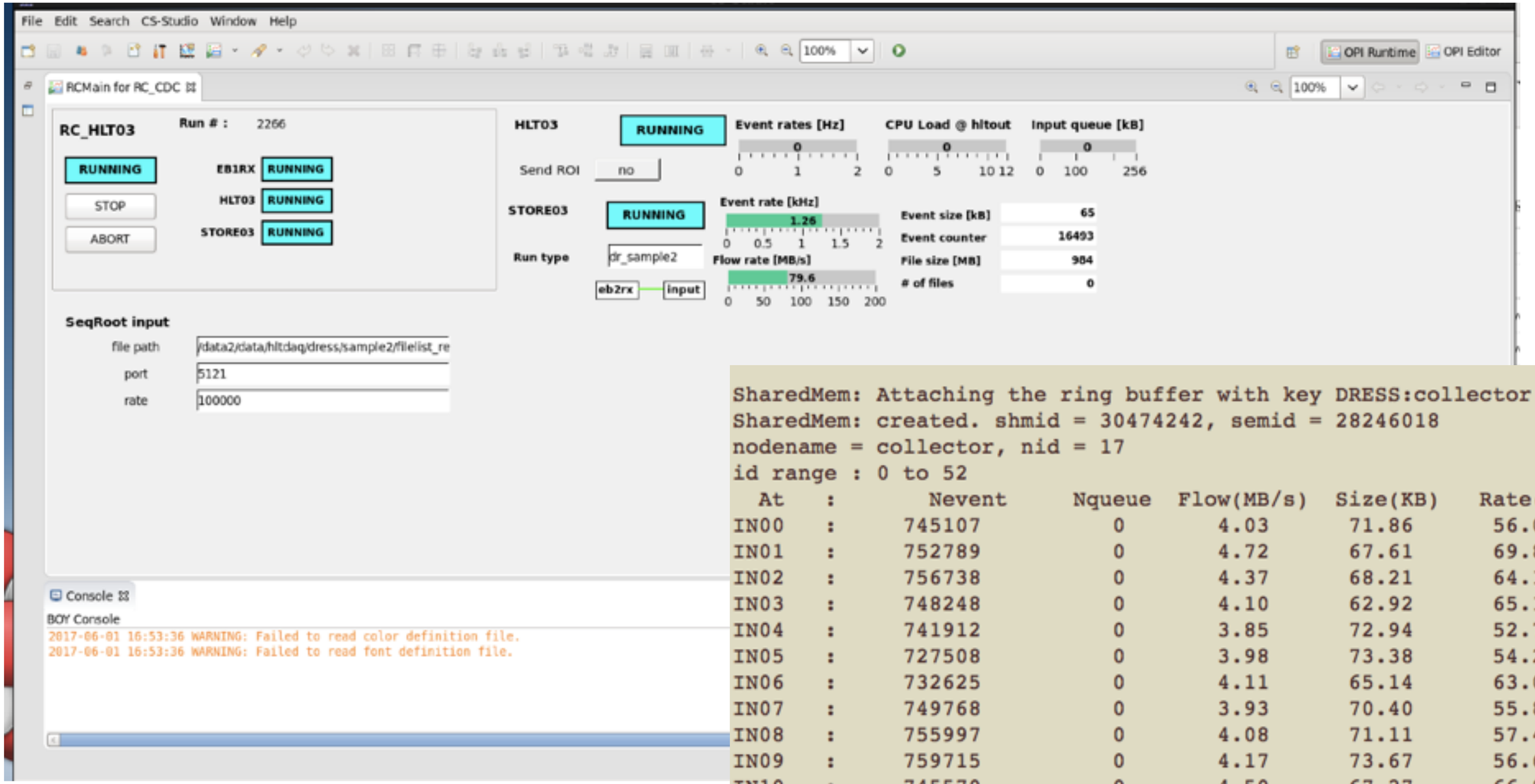
- Local database
- SoftwareTrigger Module
- Output: rawdata obj+softwaretrigger obj

Software performance on HLT

Test
- 20 cores (node 1, hlt03)
- ~2000 BB events



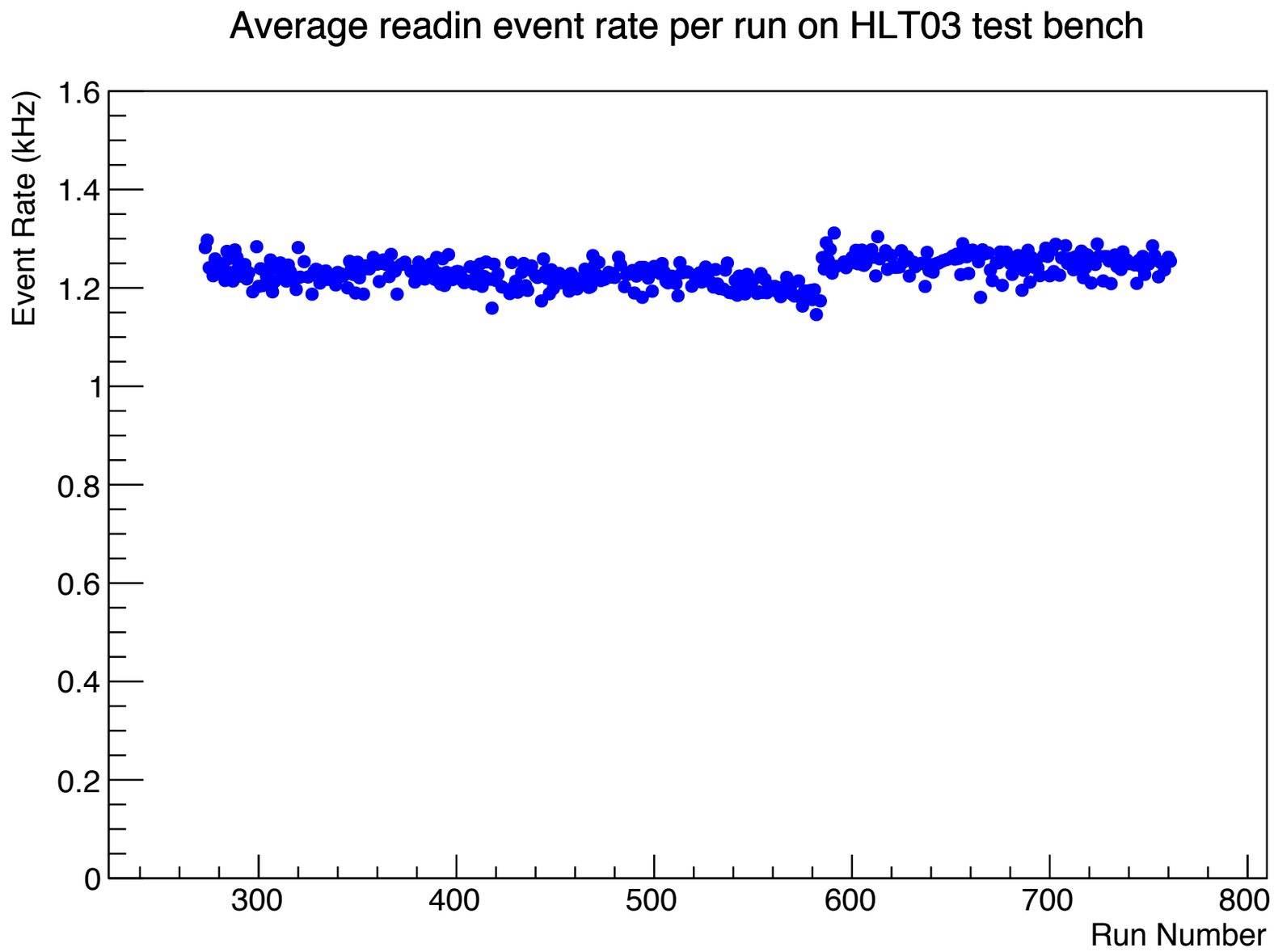





```
SharedMem: Attaching the ring buffer with key DRESS:collector
SharedMem: created. shmid = 30474242, semid = 28246018
nodename = collector, nid = 17
id range : 0 to 52
  At   :       Nevent     Nqueue  Flow(MB/s)  Size(KB)   Rate(Hz)
IN00   :       745107          0        4.03     71.86      56.08
IN01   :       752789          0        4.72     67.61      69.80
IN02   :       756738          0        4.37     68.21      64.13
IN03   :       748248          0        4.10     62.92      65.13
IN04   :       741912          0        3.85     72.94      52.78
IN05   :       727508          0        3.98     73.38      54.25
IN06   :       732625          0        4.11     65.14      63.02
IN07   :       749768          0        3.93     70.40      55.88
IN08   :       755997          0        4.08     71.11      57.42
IN09   :       759715          0        4.17     73.67      56.60
IN10   :       745570          0        4.50     67.27      66.87
IN11   :       762351          0        4.18     72.94      57.37
IN12   :       750590          0        4.73     63.39      74.55
IN13   :       765237          0        4.32     64.83      66.66
IN14   :       756646          0        4.04     70.85      56.97
IN15   :       754456          0        4.12     71.78      57.34
OUT    :     12005257          0       71.15     71.39     996.69
```

HLT03 test bench

ABORT → **LOAD** → **START**



- ~1.2 kHz read in rate with one hlt unit
- 70% events are from hadronic processes with six tracks on average

- ~10 hours smooth running to process all the raw data

HLT operation in GCRT

```
##########
# Local DB specification
##########
basf2.reset_database()
basf2.use_local_database(ROOT.Belle2.FileSystem.findFile("hlt/examples/LocalDB/database.txt"))

##########
# Path definitions
##########
# main path
main_path = basf2.create_path()
# crash handling path
crashsafe_path = basf2.create_path()

##########
# Input
##########
# Input from ringbuffer (for raw data)
input = basf2.register_module('Raw2Ds')
input.param("RingBufferName", argvs[1])

main_path.add_module(input)
```

Local DB
Download central DB with “conditionsdb download -c production LocaDB”

```
# HistoManager for real HLT
histoman = basf2.register_module('DqmHistoManager')
histoman.param("Port", int(argvs[3]))
histoman.param("Port", 9991)
histoman.param("DumpInterval", 180)
histoman.param("WriteInterval", 180)

main_path.add_module(histoman)

# Raw data unpackers
add_unpackers(crashsafe_path, components=components)

# cosmic reconstruction
add_cosmics_reconstruction(crashsafe_path, components=components)

# hlt trigger modules for test
add_fast_reco_software_trigger(crashsafe_path)
add_hlt_software_trigger(crashsafe_path)
add_calibration_software_trigger(crashsafe_path)

if enable_graceful_crash_handling:
    crashhandler = main_path.add_module('CrashHandler', path=crashsafe_path)
    # in case of crashes, save the event and continue normally with the
    # following modules
```

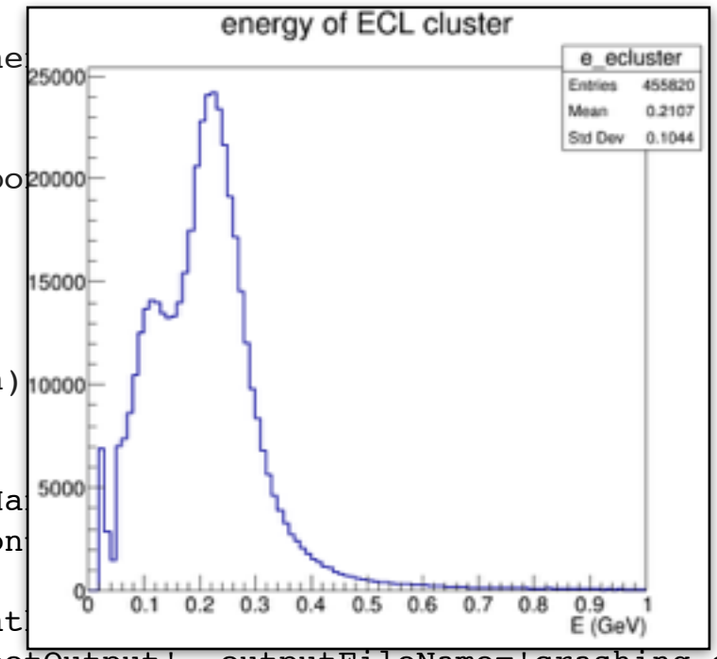
Unpacker

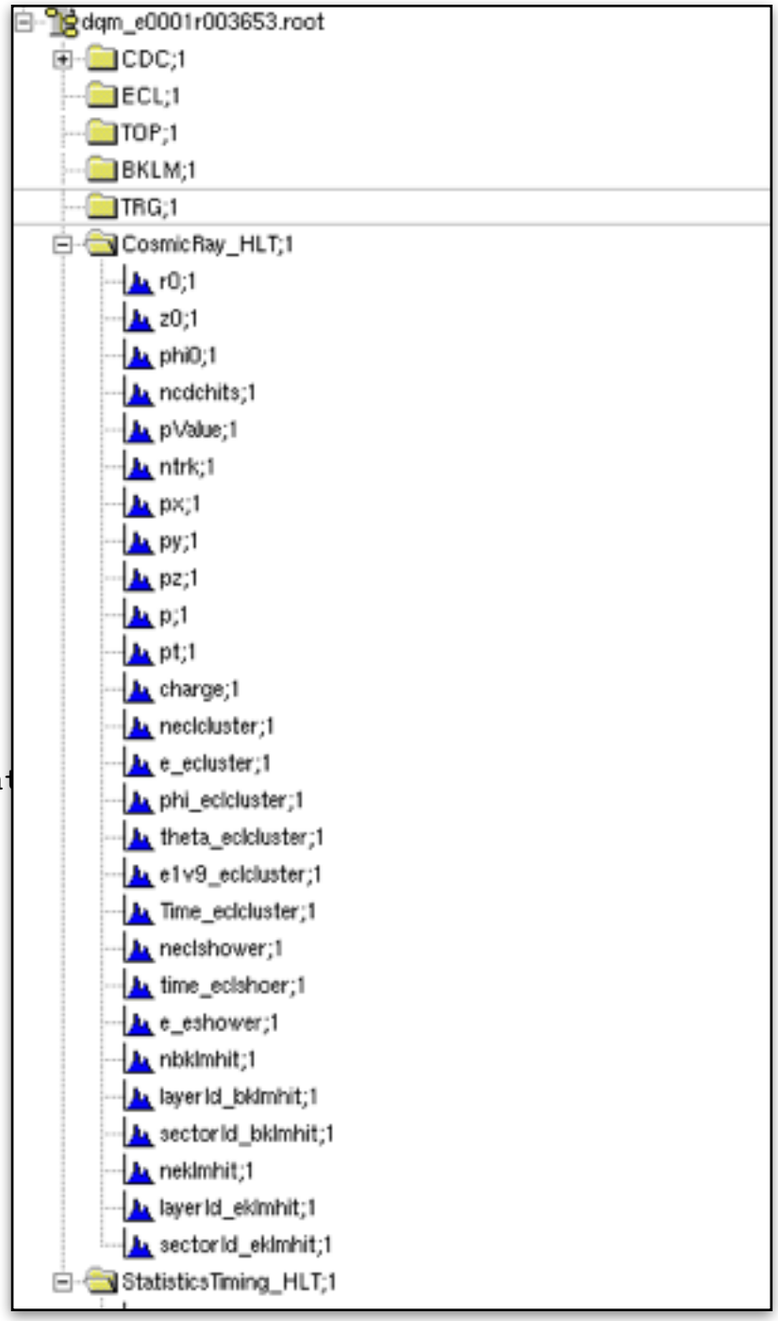
Cosmic ray reconstruction

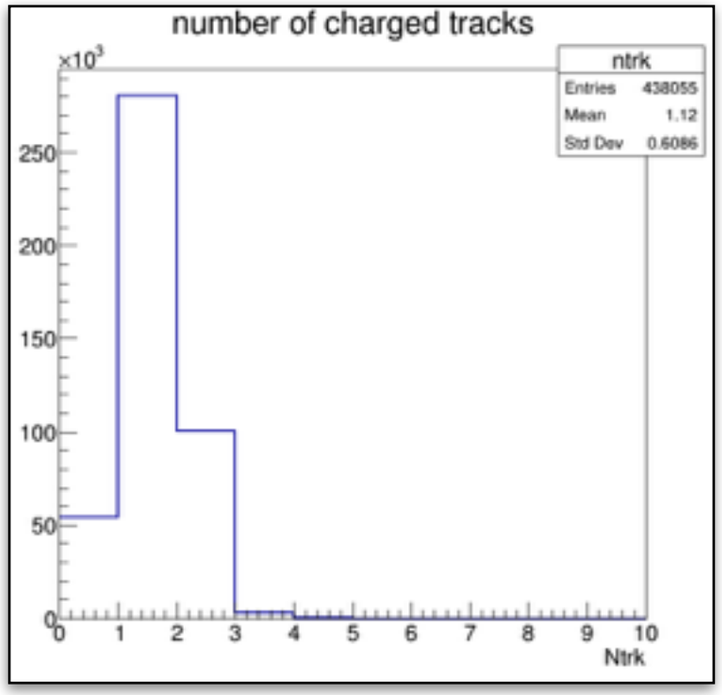
HLT trigger algorithm (no filter)

DQM of sub-detectors and HLT

```
# HistoManager for real HLT
histoman = basf2.register_module('DqmHistoManager')
histoman.param("Port", int(argvs[3]))
histoman.param("Port", 9991)
histoman.param("DumpInterval", 180)
histoman.param("WriteInterval", 180)

main_path.add_module(histoman)

# Raw data unpackers
add_unpackers(crashsafe_path, components=compone

# cosmic reconstruction
add_cosmics_reconstruction(crashsafe_path, compo

# hlt trigger modules for test
add_fast_reco_software_trigger(crashsafe_path)
add_hlt_software_trigger(crashsafe_path)
add_calibration_software_trigger(crashsafe_path)

if enable_graceful_crash_handling:
    crashhandler = main_path.add_module('CrashHa
    # in case of crashes, save the event and con
    # following modules
    save_crashing_events_path = basf2.create_pat
    save_crashing_events_path.add_module('SeqRootOutput', outputFileName='crashing_event
    crashhandler.if_false(save_crashing_events_path, basf2.AfterConditionPath.CONTINUE)
    crashhandler.set_log_level(basf2.LogLevel.WARNING)
else:
    main_path.add_path(crashsafe_path)

add_cosmic_dqm(main_path, components=components)
cosmic_hltdqm(main_path)

# Output to RingBuffer
output = basf2.register_module("Ds2Rbuf")
output.param("RingBufferName", argvs[2])

# Specification of output objects
output.param("saveObjs", saveobjs)

main_path.add_module(output)

##########
# Other utilities
##########
progress = basf2.register_module('Progress')
```







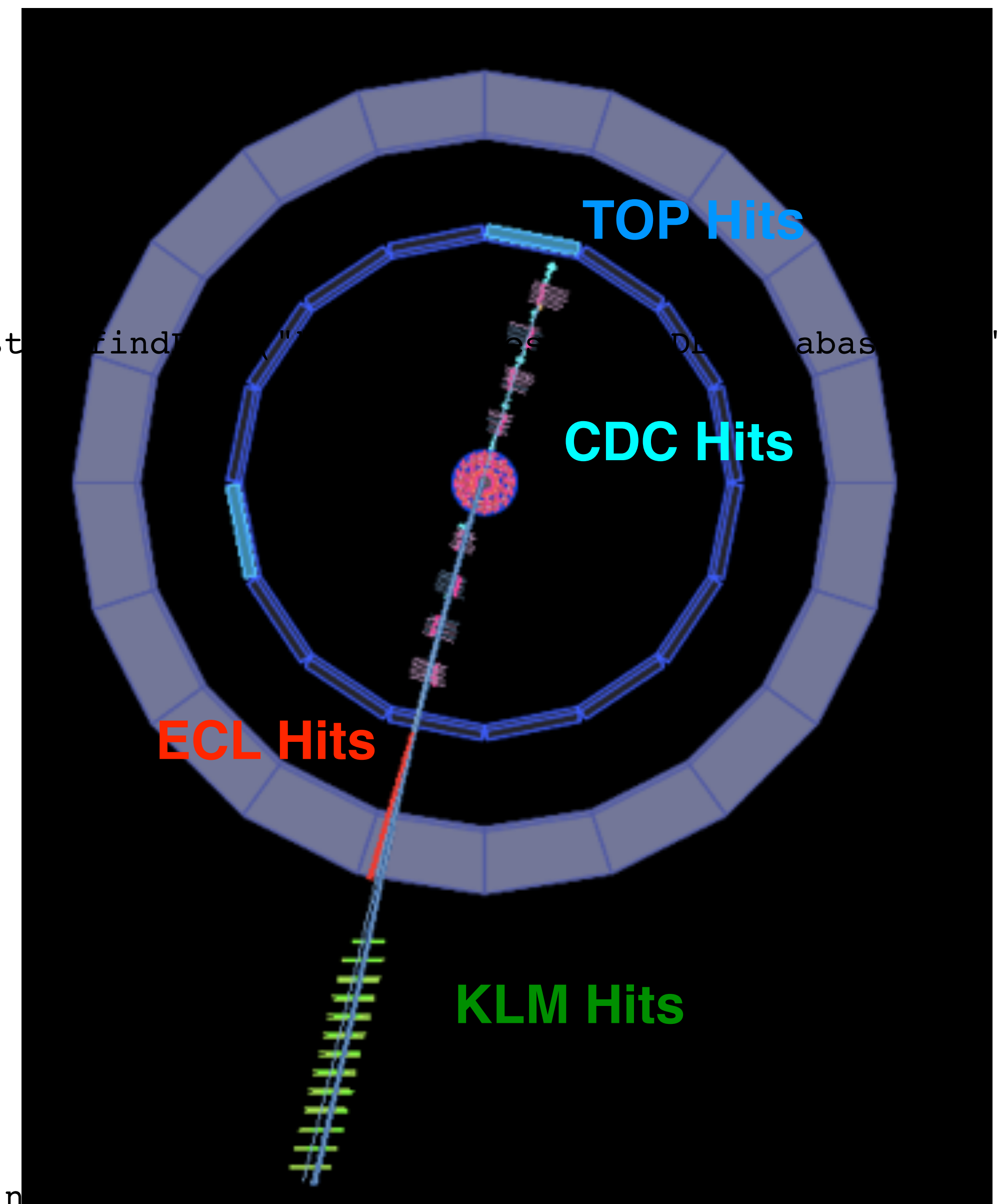
```
# Components to be processed
components = ["CDC", "ECL", "TOP", "BKLM", "EKLM", "TRG"]

# Objects to be sent to storage
saveobjs = ['EventMetaData',
  'RawCDCs', 'RawTOPs', 'RawECLs', 'RawKLMs',
  'RawFTSWs', 'RawTRGs',
  'SoftwareTriggerResult',
  'Tracks', 'TrackFitResults', 'CDCHits', 'TOPDigits',
  'ECLClusters',
  'BKLMDigits', 'BKLMHit1ds', 'BKLMHit2ds',
  'EKLMDigits', 'EKLMHit1ds', 'EKLMHit2ds']
```

configure detector components

Raw data objects

Reconstructed data objects for online event display

```
# Crash handler switch
enable_graceful_crash_handling = False

##########
# Local DB specification
##########
basf2.reset_database()
basf2.use_local_database(ROOT.Belle2.FileSyst
'))

##########
# Path definitions
##########
# main path
main_path = basf2.create_path()
# crash handling path
crashsafe_path = basf2.create_path()

##########
# Input
##########
# Input from ringbuffer (for raw data)
input = basf2.register_module('Raw2Ds')
input.param("RingBufferName", argvs[1])

main_path.add_module(input)

# HistoManager for real HLT
histoman = basf2.register_module('DqmHistoMan
histoman.param("Port", int(argvs[3]))
histoman.param("Port", 9991)
histoman.param("DumpInterval", 180)
histoman.param("WriteInterval", 180)
```



Summary

- Dress rehearsal (DAQ part) on HLT03 test bench
  - Round 1 shows the HLT implementation is stable
  - Round 2
    - a larger data samples including low multiplicity processes will be run
    - updated software

- Cosmic ray data taking
  - HLT software is implemented in GCRT with stable operation

- Things to be done before phase2 running
  - Database, crash handling, software configuration handling etc.
  - Collaboration with the groups e.g. software, database, calibration.