Event Feeding for Event Display and Dedicated Detector Monitoring

R.Itoh, KEK

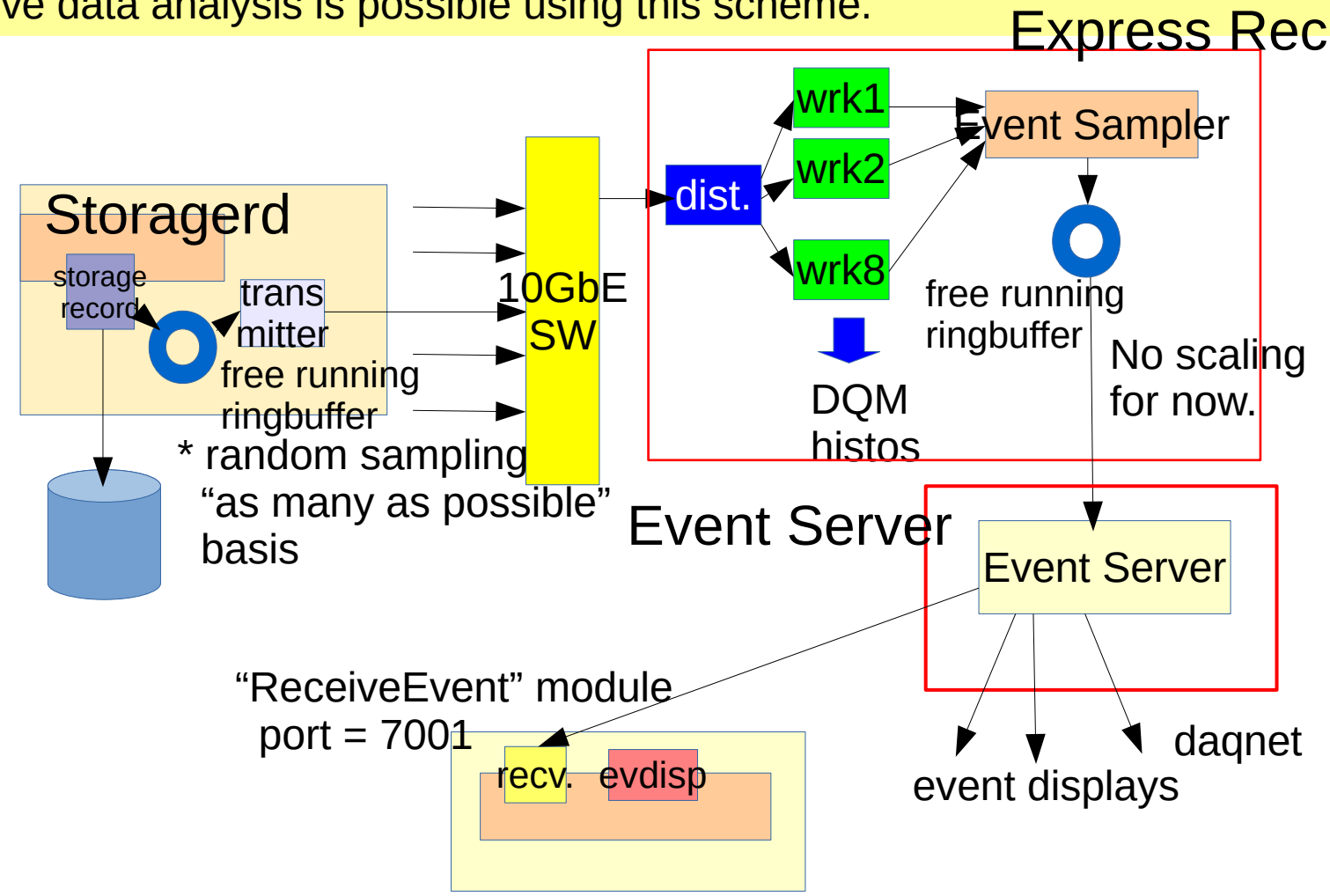
- Live event sampling and feeding is essential for the event display, and also useful for the detailed monitoring of each detector.

- The event sampling is implemented at the end of ExpressReco processing, which has the reconstruction objects in addition to raw data.
  -> Useful for event display and also detector monitoring.

- But there is a request to feed only raw data objects for detector monitoring.

Current Implementation

- For the detailed data monitoring and the live event display, an event sampling mechanism is implemented in ExpressReco.
- At the end of the processing in ExpressReco, the processed events are stored in a free running ring buffer so that they can be accessed through the event server process.
- The events are sent to the connected socket to the event server.
- The event display is implemented using this connection.
- Live data analysis is possible using this scheme.



- Free running ring buffer is to ensure the smooth event sampling not to disturb the data taking.

- Currently “as many as possible” basis sampling is performed.

- The events in output ring buffer are fed to “event server” which accepts connections from external application.

- Multiple connections can be accepted by the event server.

- One of the apps is the event display.

Event Display script

- Event display is implemented using the standard offline event display.

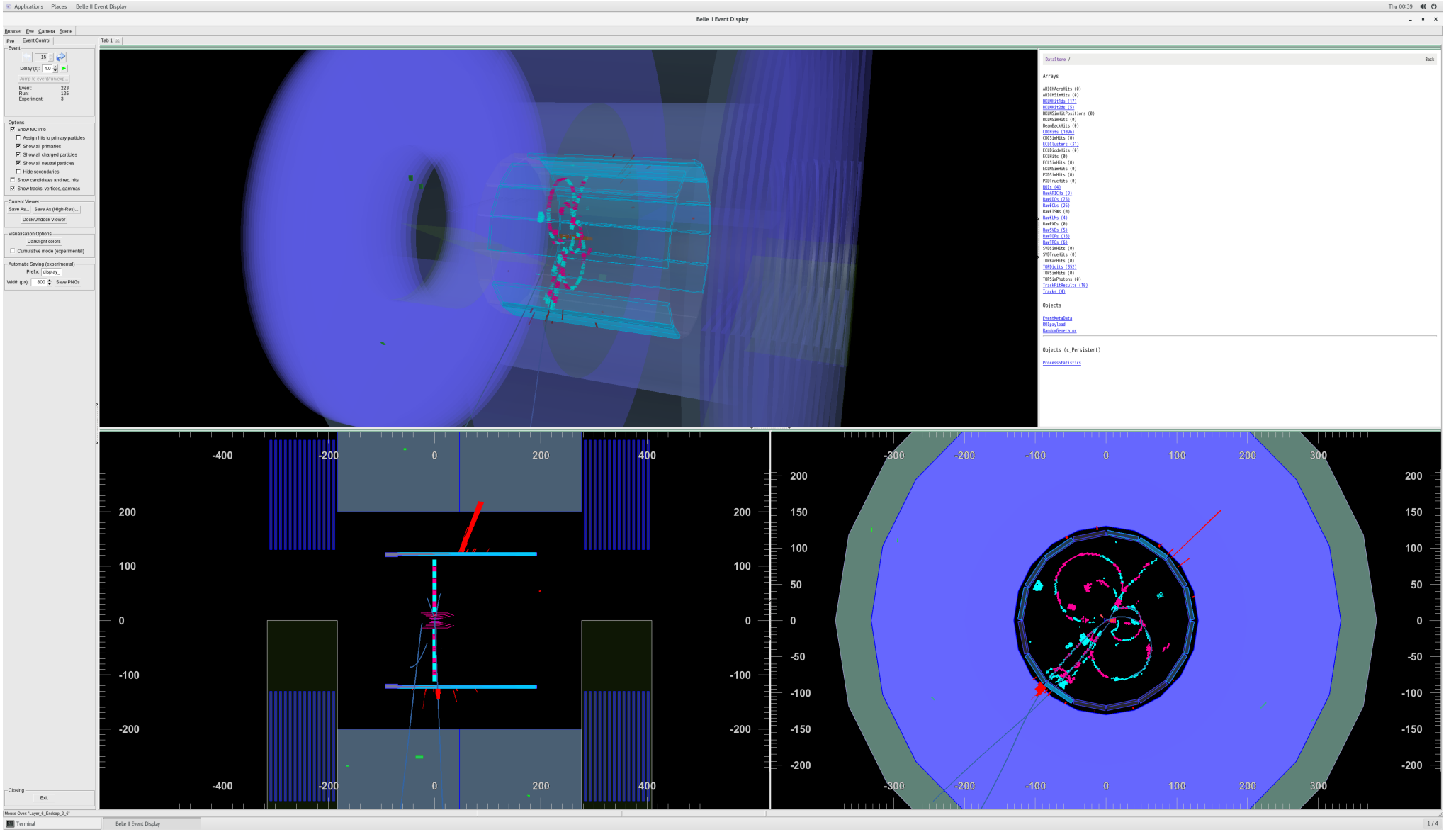
- It has “asynchronous” mode just to pick up the latest event from the event stream.

- The simplest “normal.py”

```
path.add_module("ReceiveEvent", Host="ereco01", Port=7001)

path.add_module("Gearbox")
path.add_module("Geometry")

path.add_module("AsyncDisplay", showCDCHits=True,
                showARICHHits=True, showRecoTracks=False)
```


Applications
Places
Belle II Event Display
Thu 00:39
Belle II Event Display
Browser
Eve
Camera
Scene
Eve
Event Control
Tab 1
Event
Delay (s):
Jump to event/run/exp...
Event:
Run:
Experiment:
Options
Show MC info
Assign hits to primary particles
Show all primaries
Show all charged particles
Show all neutral particles
Hide secondaries
Show candidates and rec. hits
Show tracks, vertices, gammas
Current Viewer
Save As...
Save As (High-Res)...
Dock/Undock Viewer
Visualisation Options
Dark/light colors
Cumulative mode (experimental)
Automatic Saving (experimental)
Prefix:
Width (px):
Save PNGs
Closing
Exit
DataStore /
Back
Arrays
Objects
Objects (c_Persistent)
Mouse Over: "Layer_6_Endcap_2_6"
Terminal
Belle II Event Display
1 / 4

Objects to be transferred to event display

```
# Always store those objects
ALWAYS_SAVE_OBJECTS = ["EventMetaData", "SoftwareTriggerResult", "TRGSummary",
"ROIpayload", "SoftwareTriggerVariables"]
# Objects to be left on output
RAWDATA_OBJECTS = ["RawCDCs", "RawSVDs", "RawPXDs", "RawTOPs", "RawARICHs",
"RawKLMs", "RawECLs", "RawFTSWs", "RawTRGs",
                   "ROIs"]
# Objects which will be kept after the ExpressReconstruction, for example for the
Event Display
PROCESSED_OBJECTS = ['Tracks', 'TrackFitResults',
                     'SVDClusters', 'PXDClusters',
                     'CDCHits', 'TOPDigits', 'ARICHHits',
                     'ECLClusters',
                     'BKLMHit1ds', 'BKLMHit2ds',
                     'EKLMHit1ds', 'EKLMHit2ds',
                     'SoftwareTriggerResult']
```

If other objects are necessary to show some particular items in the event display, let us know.

Event display with event selection

- Two types of event selection can be turned on using different script for basf2.

1) mumu only
   evdisp_hltskim_mumu_only.py

2) Hadronic event only
   evdisp_hltskim_hadrons_only.py

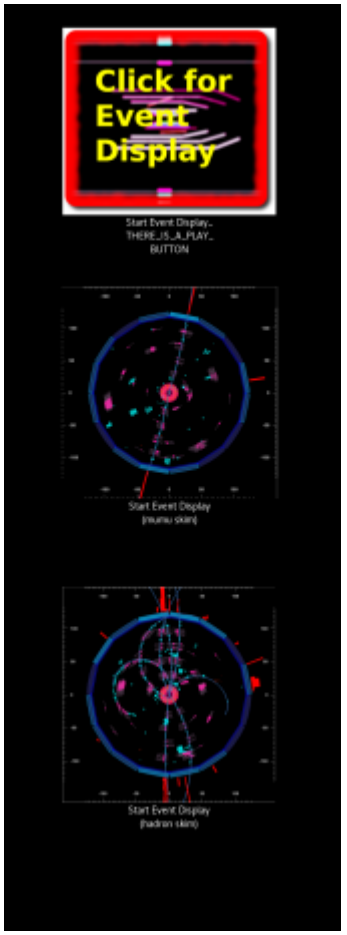
- They are implemented by looking at HLT tag

```
path = basf2.Path()
path.add_module("ReceiveEvent", Host="ereco01", Port=7001)

path.add_module("Gearbox")
path.add_module("Geometry")
add_mumu_skim(path)
#path.add_module(DisplayHLTTags())

path.add_module("AsyncDisplay", showCDCHits=True, showARICHHits=True,
                showRecoTracks=False)
```

They can be switched by clicking corresponding icons on the event display screen.



Event feeding to other apps

- Sampled events could be useful for the detailed debugging of detector.

- Reconstructed objects are not necessary for the debugging.
  ->Removing them from sampled event stream is preferred.

- An additional event server is being prepared for the purpose and will be tested soon.
  (Simple test was already performed.)

- I will announce it when ready with a simple description on Confluence.

Express Reco