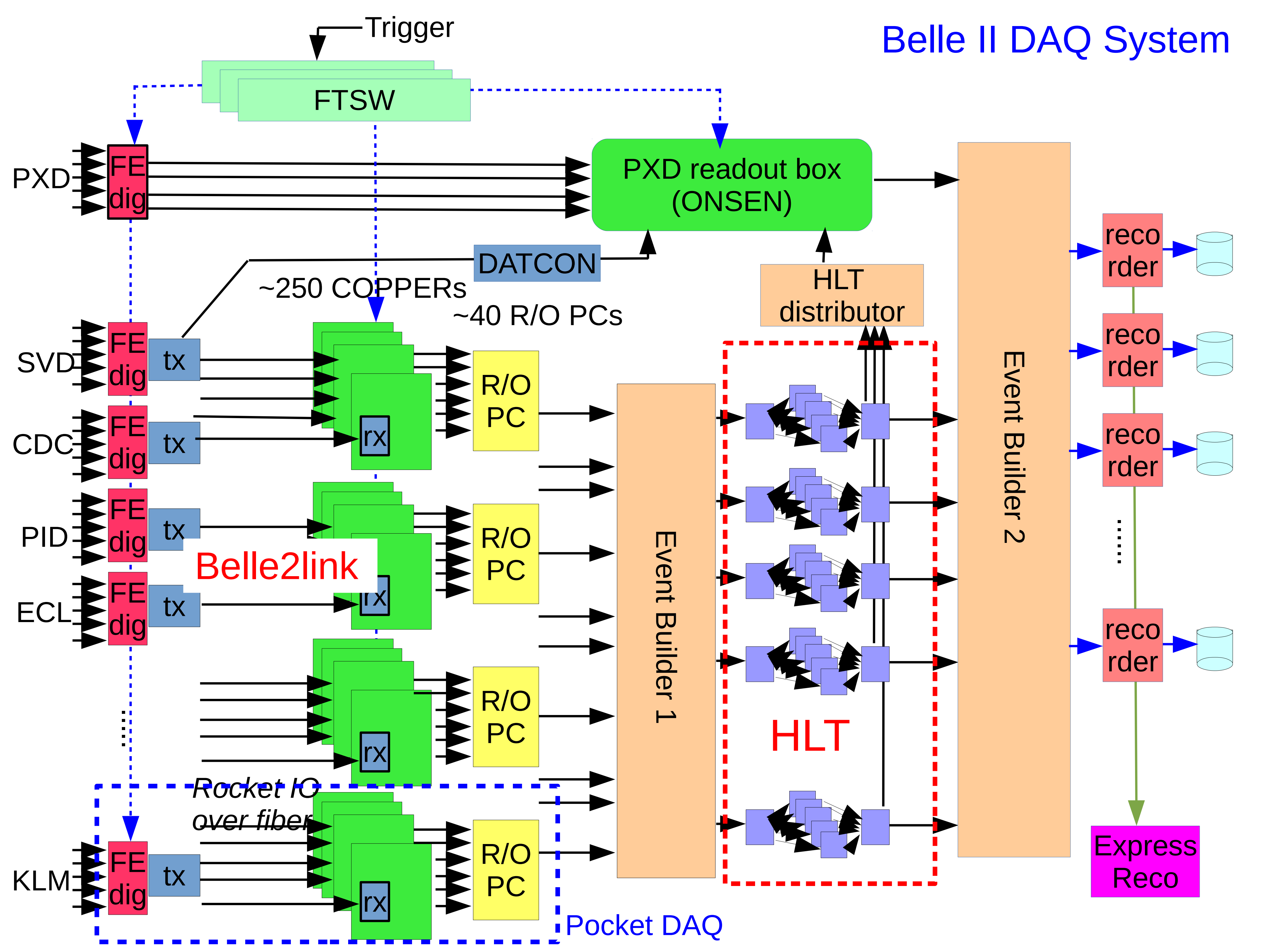
HLT Framework

R.Itoh, KEK



Trigger
Belle II DAQ System
FTSW
PXD
FE dig
PXD readout box (ONSEN)
DATCON
~250 COPPERs
~40 R/O PCs
HLT distributor
SVD
CDC
PID
ECL
KLM
tx
rx
R/O PC
Belle2link
Event Builder 1
Event Builder 2
HLT
reco rder
Express Reco
Rocket IO over fiber
Pocket DAQ

1. Introduction

Functionality of Belle II HLT

0. Final conversion of raw data into ROOT objects (except PXD)

1. Perform real time “full reconstruction” of events with data from all detectors (except for PXD) and Trigger decision.
   * Tracking with SVD+CDC
   * ECL clustering + energy reconstruction
   * PID with TOP+ARICH
   * Physics event reconstruction using all info.
   * Physics trigger decision
   -> Next talks

2. Data quality monitoring by live histogram transfer
   -> Separate talk.

3. RoI feedback to PXD

Estimation of required HLT processing power in DAQ design

- The required processing power of Belle II HLT was estimated based on our experience with RFARM used in the previous Belle experiment.

- The RFARM was equipped with 140 equivalent cores (number scaled to be equivalent to Belle II HLT cores) and the full event reconstruction was performed using the same offline software and scripts.

- The RFARM had been processing the maximum luminosity of Belle = 2.0×10^{34} with an 80-90% average CPU consumption.

- The luminosity of SuperKEKB at t=0 was assumed to be 1/4 of the design luminosity = 2.0×10^{35}, which is 10 times of Belle's maximum luminosity. Therefore the required number of cores at t=0 is estimated to be 140 * 10 = 1400 cores.

- Our plan is to have 5 HLT units at t=0 where one unit is equipped with 320 cores, and we will have 1600 cores in total with 5 units. It should be enough for t=0, if we scale our experience at Belle.

- For the beyond to reach the design luminosity, we will add more HLT units gradually. We assume Moore's law in the addition expecting to have more number of cores in one unit. So the reduction in total number of units is in the design and we expect the full luminosity of 8×10^{35} can be processed with 10-15 HLT units with 1600*4 = 6400 cores.

HLT processing rate in the original design

- At the full luminosity of 8x10^{35}, the physics rate for BBbar and Continuum events is 4kHz while 2kHz for "low multiplicity(=$\tau\tau$,$\mu\mu$, etc.)" events.

- The expected average L1 trigger rate at the full luminosity is 20kHz (Note: 30kHz is the peak maximum), and it is supposed to be processed with 6400 cores. So the required processing rate per core is ~3Hz resulting in the required average processing time per core to be ~0.3 sec.

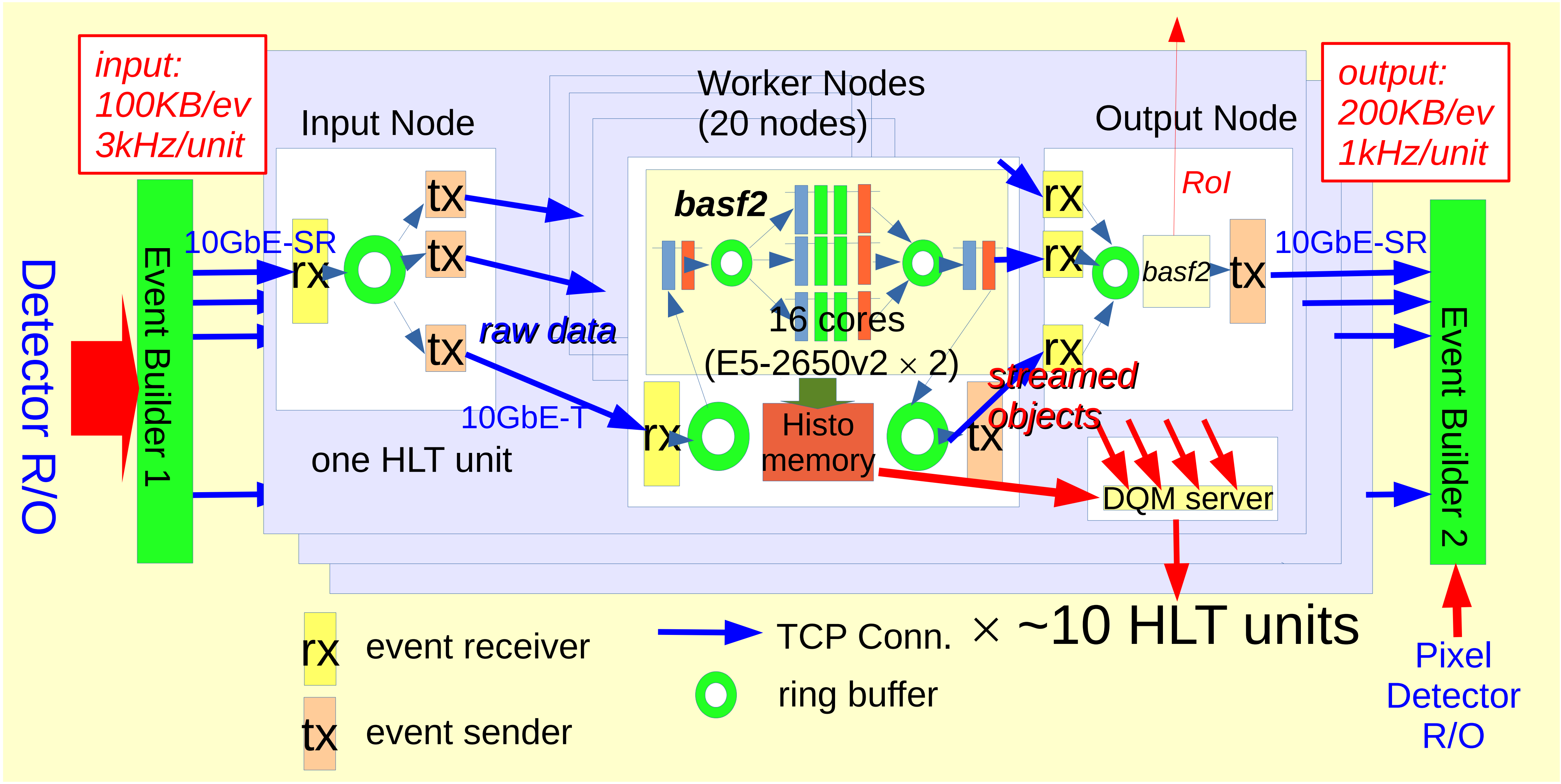
- The physics rate estimation shows, in the 3Hz rate, 20% are from BBbar and Continuum, 10% from "low multiplicity" events, and remaining 70% from other events including beam BGs and two photons.

- The processing time for "low multiplicity" and "other" events is expected to be significantly less than that for hadronic (=BBbar+Continuum) events.

- If we assume the processing time per core for one hadronic event (20%) to be 1.0 sec and 0.1 sec for "low multi" and other events (80%), the average processing time per event could be 0.28 sec, which satisfies 3Hz/core processing.

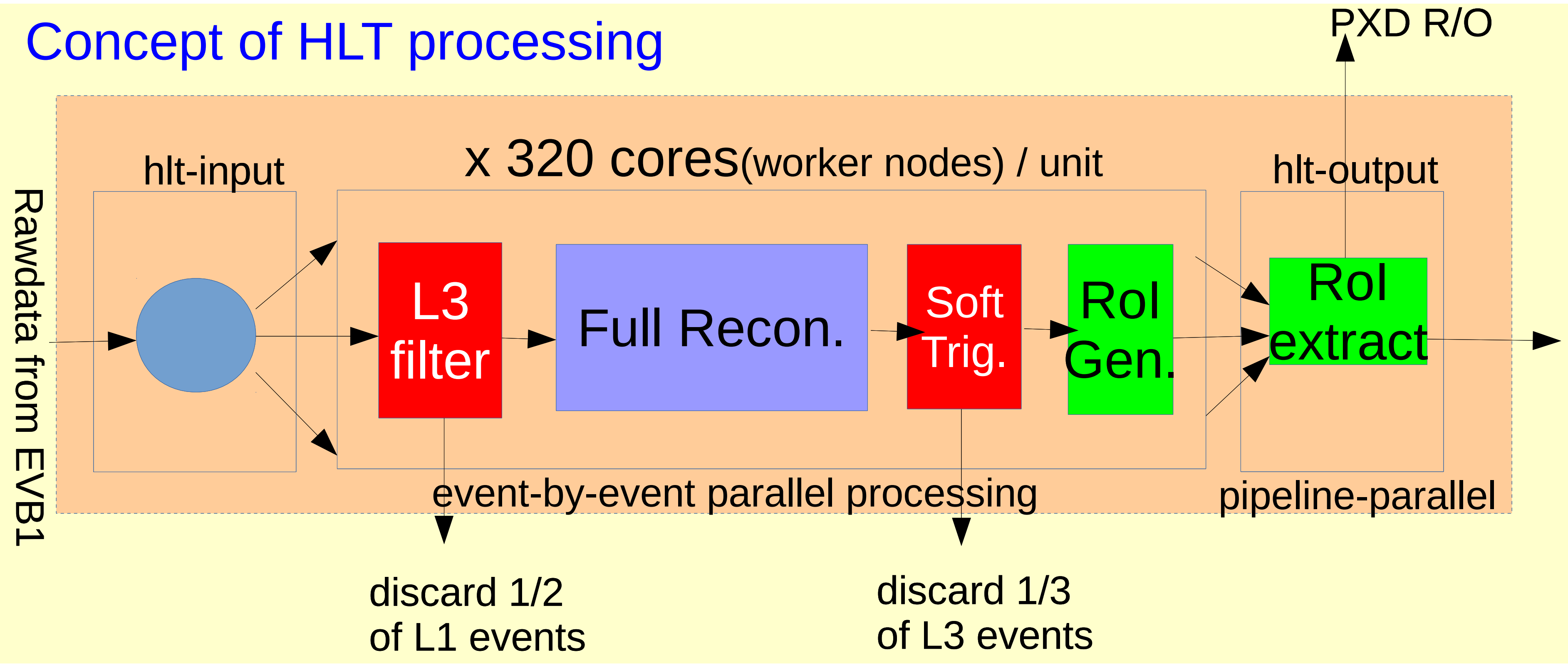
- The detail of the processing time is now being studied using the real Belle II software with the HLT configuration on the test bench. The results will be reported at next BPAC meeting.

Structure of Belle II High Level Trigger (HLT)



- 320 cores / unit (HLT1). Possible to have more cores/unit when adding more units. (following Moore's law.)
- Increase number of units gradually to keep up with the lum. improvement in accelerator.
- We need at least 6400 cores for the full luminosity, but could be more depending on the processing load of HLT reconstruction.

Concept of HLT processing



- One unit processes
  * ~ 2-3 kHz L1 rate (of total of 30kHz) with event size <100kB
  * 1/2 rate reduction with “Level 3” filter
    - Based on fast CDC tracking + ECL clustering.
    - Cut in the track |z| position and ECL energy sum
  * Full event reconstruction using all detector signals except PXD
  * Software trigger using physics event skim codes (Hadronic/tau event selection....) + Monitor trigger -> 1/3 reduction

Expected rate reduction : 1/3 = ~1 kHz/unit at output.

Trigger and Event Selection at HLT

- At the end of reconstruction chain, an event selection module is placed in basf2.

- The module summarizes the various reconstruction results and decides the event should be taken or not.

- The result of the decision is written to “HLTtag” object.

- A module to select objects to be sent to storage is placed right after the selection module.

- The module erases the unnecessary objects from DataStore by looking at HLTtag. If the HLTtag says the event to be discarded, all rawdata objects (RawXXXs) are erased.

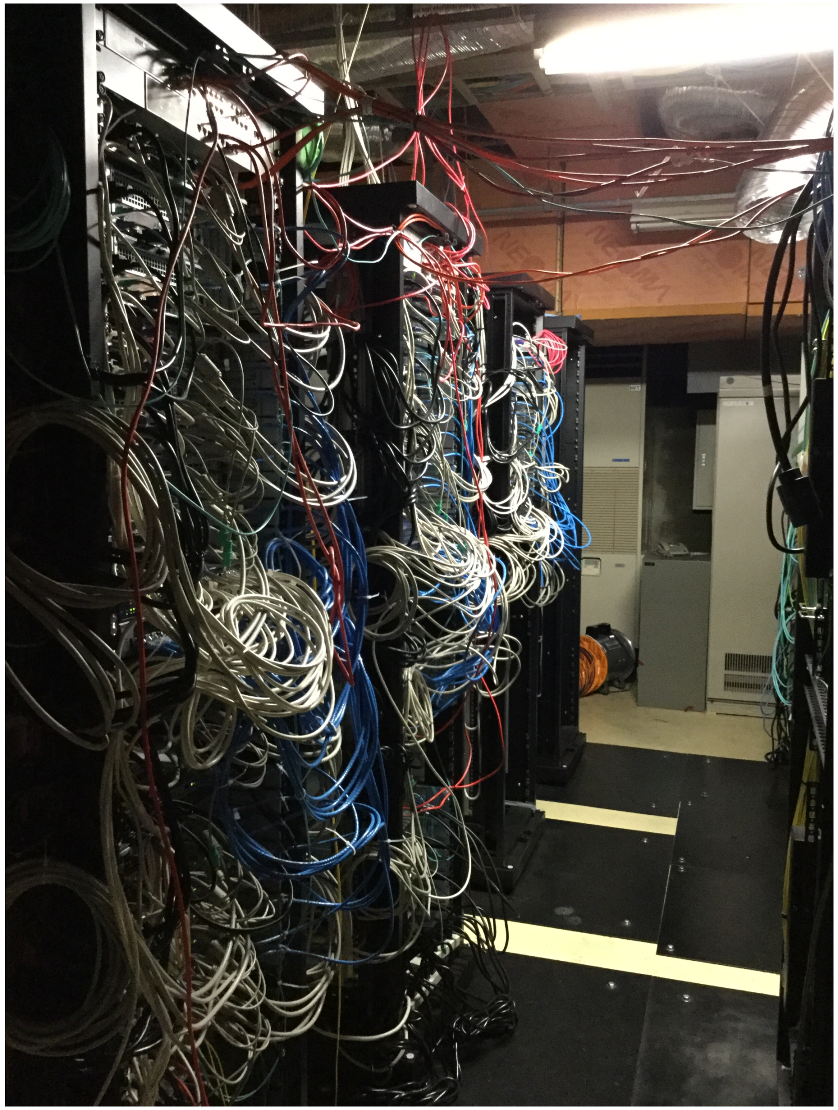
- However, at least, EventMetaData and trigger summary objects are left and sent to the storage even for the discarded event.

2. Recent status of HLT preparation

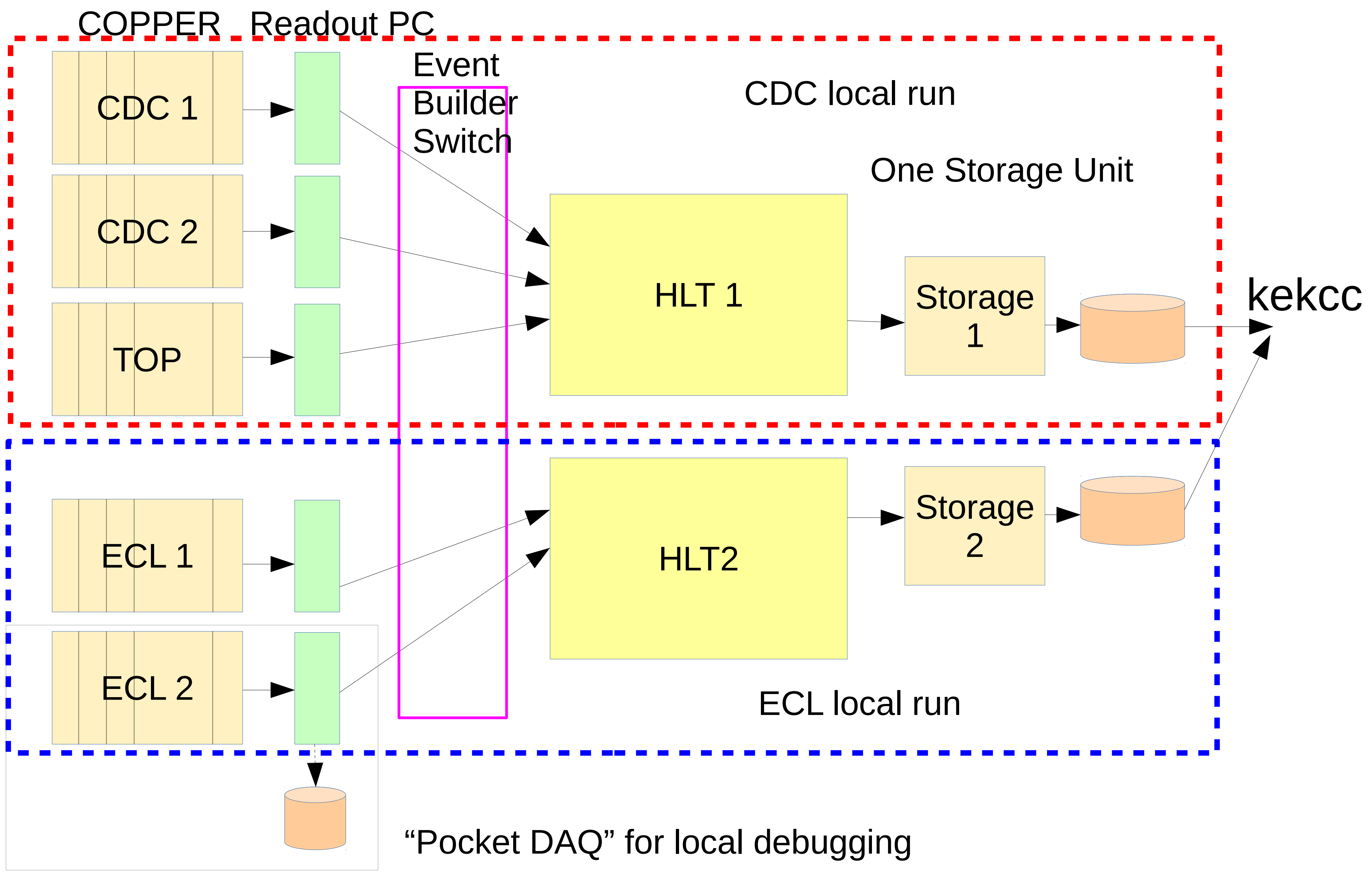
- 3 physical units are already built.
  * 1st unit
    - 20 worker nodes with 16 cores (3.2GHz(turbo))
    - Full data flow from EVB to Storage
    - Storage : 120TB
    - Direct connection to kekcc (tentative)
  * 2nd unit
    - 16 worker nodes with 20 cores (3.2GHz(turbo))
    - Full data flow from EVB to Storage
    - Storage : 160TB
  * 3rd unit
    - 16 worker nodes with 20 cores (3.2GHz(turbo))
    - No connection to EVB yet
    - Storage : 120TB

- 1st and 2nd units are being used for the cosmic ray data taking
- 3rd unit is used for the software development/performance test.

3 HLT units implemented in “B3 server room” in Tsukuba hall



Two stream operation for cosmic ray run.



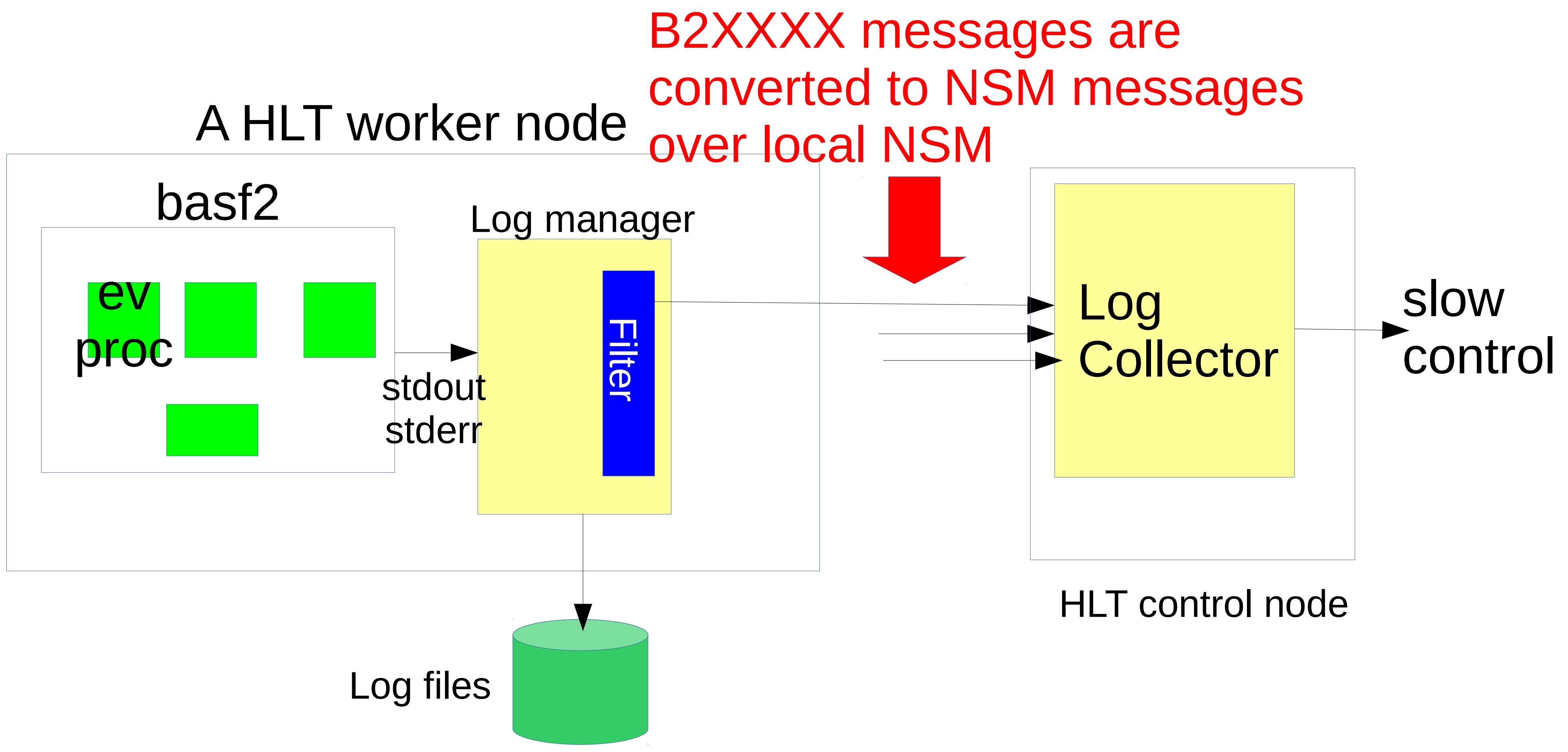
3. New development: Log message handling

- “basf2” processing is performed on all cores of the processing nodes using its parallel processing mechanism.

- The output of the processing (stdout/stderr) is redirected and logged into log files.

- The processing of a particular event may cause a trouble and stop the whole processing. In such cases, basf2 normally issues “B2ERROR” or “B2FATAL” message.

- These messages are trapped by HLT's log manager and redirected to slow control so that it can report the errors to slow control quickly.


B2XXXX messages are converted to NSM messages over local NSM
A HLT worker node
basf2
ev proc
Log manager
Filter
stdout stderr
Log Collector
slow control
HLT control node
Log files

NSM messages at LogCollector

```
09:03:43.228 LOG<=EVP_HLTWK12 (2,1472601823) (20000) Processed:   1 runs,  35000 events
09:03:44.153 LOG<=EVP_HLTWK09 (2,1472601824) (20000) Processed:   1 runs,  35000 events
09:03:44.735 LOG<=EVP_HLTWK10 (2,1472601824) (20000) Processed:   1 runs,  36000 events
09:03:45.835 LOG<=EVP_HLTWK16 (2,1472601825) (20000) Processed:   1 runs,  35000 events
09:03:45.926 LOG<=EVP_HLTWK11 (2,1472601825) (20000) Processed:   1 runs,  36000 events
09:03:46.013 LOG<=EVP_HLTWK15 (2,1472601826) (20000) Processed:   1 runs,  34000 events
09:03:46.405 LOG<=EVP_HLTWK13 (2,1472601826) (20000) Processed:   1 runs,  37000 events
09:03:48.235 LOG<=EVP_HLTWK14 (2,1472601828) (20000) Processed:   1 runs,  37000 events
09:03:50.853 LOG<=EVP_HLTWK09 (2,1472601830) (20000) Processed:   1 runs,  36000 events
09:03:51.766 LOG<=EVP_HLTWK10 (2,1472601831) (20000) Processed:   1 runs,  37000 events
09:03:51.874 LOG<=EVP_HLTWK12 (2,1472601831) (20000) Processed:   1 runs,  36000 events
09:03:52.618 LOG<=EVP_HLTWK11 (2,1472601832) (20000) Processed:   1 runs,  37000 events
09:03:53.349 LOG<=EVP_HLTWK15 (2,1472601833) (20000) Processed:   1 runs,  35000 events
09:03:53.645 LOG<=EVP_HLTWK16 (2,1472601833) (20000) Processed:   1 runs,  36000 events
09:03:58.292 LOG<=EVP_HLTWK12 (2,1472601838) (20000) Processed:   1 runs,  37000 events
09:03:59.658 LOG<=EVP_HLTWK11 (2,1472601839) (20000) Processed:   1 runs,  38000 events
09:04:01.575 LOG<=EVP_HLTWK15 (2,1472601841) (20000) Processed:   1 runs,  36000 events
09:04:07.214 LOG<=EVP_HLTWK11 (2,1472601847) (20000) Processed:   1 runs,  39000 events
09:04:07.731 LOG<=COLLECTOR (2,1472601847) Processed:   1 runs,  50000 events
09:04:07.738 LOG<=COLLECTOR (2,1472601847) Processed:   1 runs,  50000 events
09:04:07.751 LOG<=COLLECTOR (2,1472601847) Processed:   1 runs,  50000 events
09:04:07.760 LOG<=COLLECTOR (2,1472601847) Processed:   1 runs,  50000 events
09:04:07.773 LOG<=COLLECTOR (2,1472601847) Processed:   1 runs,  50000 events
09:04:12.123 LOG<=COLLECTOR (2,1472601852) Processed:   1 runs,  50000 events
```

4. Performance test (of framework)

- Possible bottleneck in HLT processing performance is the load for ROOT object streaming and destreaming.

- In the 2014 DESY beamtest, object destreaming performance on the output node was the limiting factor, and the parallel processing of destreaming was introduced.

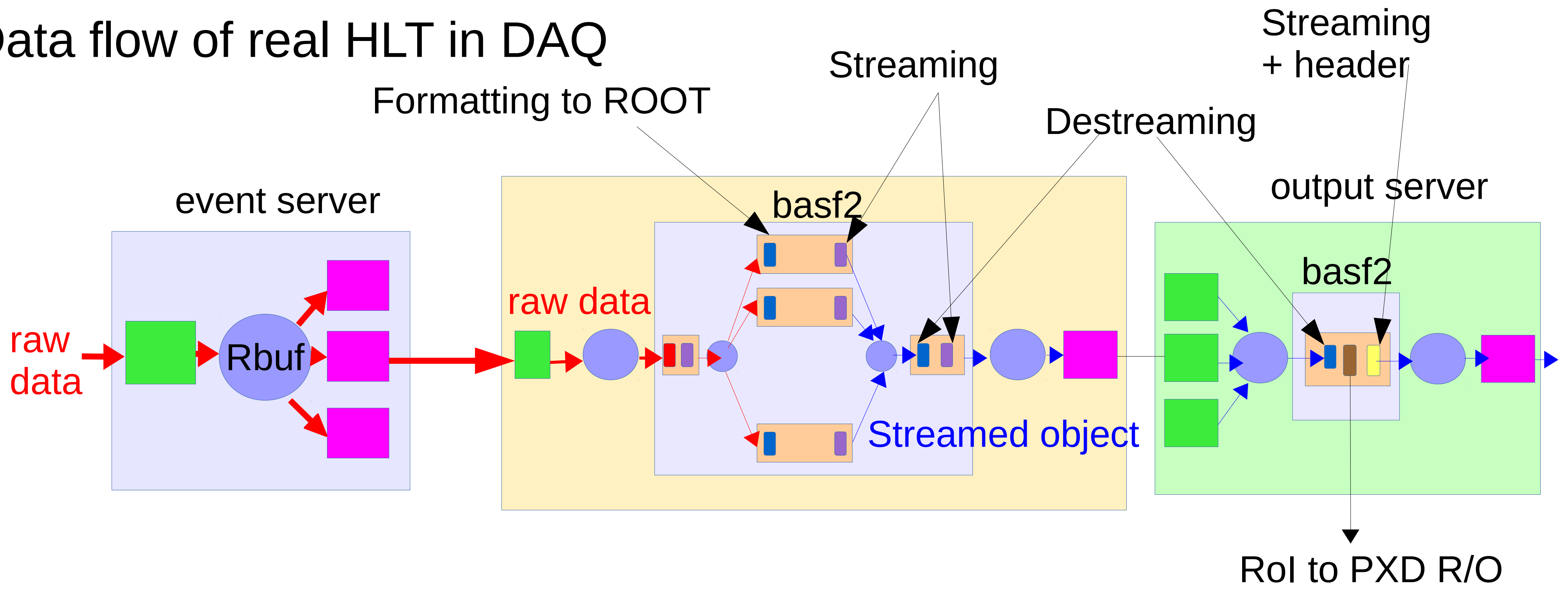
- We need to check the overall performance of data flow in HLT using the realistic benchmark.

- Input: RawSVD + RawCDC (+EventMetaData) generated by MC

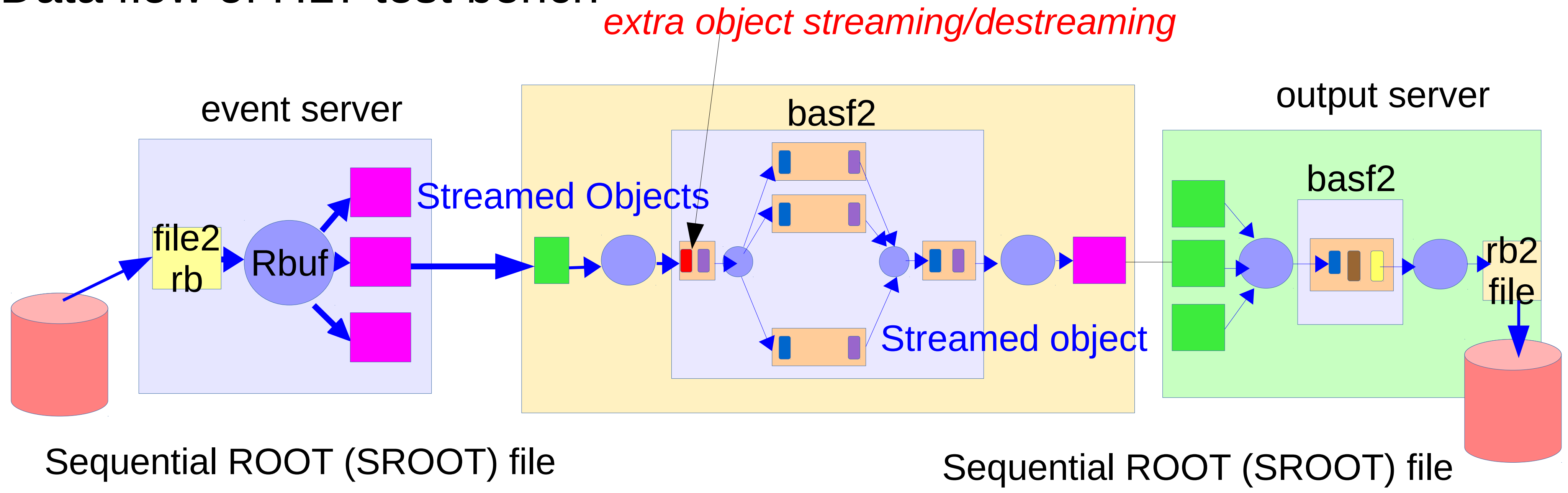
- Processing : Full tracking with SVD+CDC using the same offline code.

- Output: RawSVD + RawCDC + Tracking objects (Tracks, TrackFitResults)

Data flow of real HLT in DAQ



Data flow of HLT test bench


extra object streaming/destreaming
event server
basf2
output server
file2 rb
Rbuf
Streamed Objects
basf2
Streamed object
rb2 file
Sequential ROOT (SROOT) file
Sequential ROOT (SROOT) file

* CPU consumption on all cores was confirmed to be close to 100%.
-> No bottleneck by object streaming seen.

[mixed rate]

IN00	:	18269	0	0.71	40.14	17.78
IN01	:	22235	0	0.94	38.63	24.45
IN02	:	30451	0	1.10	38.52	28.43
IN03	:	24394	0	0.91	39.24	23.29
IN04	:	28513	0	1.04	39.18	26.50
IN05	:	99851	0	0.98	39.01	25.04
IN06	:	104195	0	1.08	38.75	27.76
IN07	:	61452	0	1.08	39.38	27.50
IN08	:	40518	0	0.90	39.22	23.01
IN09	:	92953	0	1.05	38.47	27.21
IN10	:	102719	0	1.00	38.88	25.61
IN11	:	58679	0	1.00	38.38	26.06
IN12	:	71747	0	1.00	39.22	25.57
IN13	:	34697	0	1.08	38.95	27.67
IN14	:	35099	0	1.04	38.84	26.73
IN15	:	13367	0	1.05	38.78	27.11

Ave. event size : 39kB/ev (in 34kB)
Processing time / event / core = 0.54 sec.
Processing rate / node (20core) ~ 26Hz ==> 416Hz/HLT unit

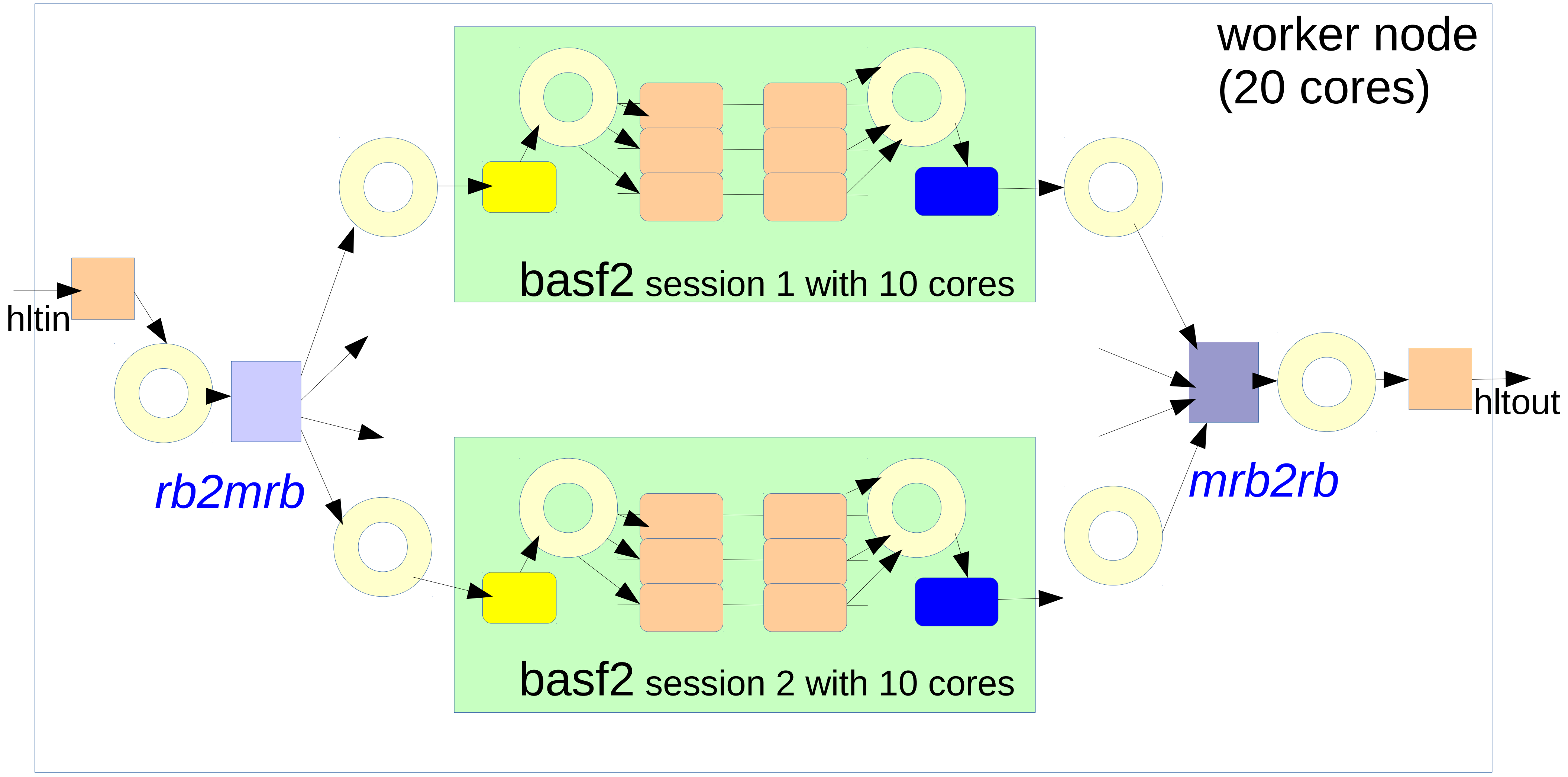
[tautau rate]

At	:	Nevent	Nqueue	Flow(MB/s)	Size(KB)	Rate(H
IN00	:	17941	0	3.54	28.46	124.30
IN01	:	21756	0	3.39	28.46	118.95
IN02	:	29955	0	3.95	28.22	139.79
IN03	:	23984	0	4.68	28.12	166.48
IN04	:	28046	0	3.88	28.26	137.24
IN05	:	99381	0	4.99	28.13	177.42
IN06	:	103704	0	4.02	28.43	141.33
IN07	:	60974	0	2.71	28.60	94.73
IN08	:	40055	0	3.99	28.38	140.77
IN09	:	92463	0	3.95	28.39	138.96
IN10	:	102285	0	4.22	28.57	147.71
IN11	:	58210	0	4.59	28.12	163.06
IN12	:	71302	0	4.70	28.10	167.42
IN13	:	34207	0	4.21	28.34	148.61
IN14	:	34649	0	4.08	28.30	144.30
IN15	:	12897	0	4.58	28.28	161.81

Event size : 28kB
Average rate : ~140Hz/node -> 2.8kHz/unit

* However, there could be a bottleneck by object streaming
  if the event processing time becomes shorter.

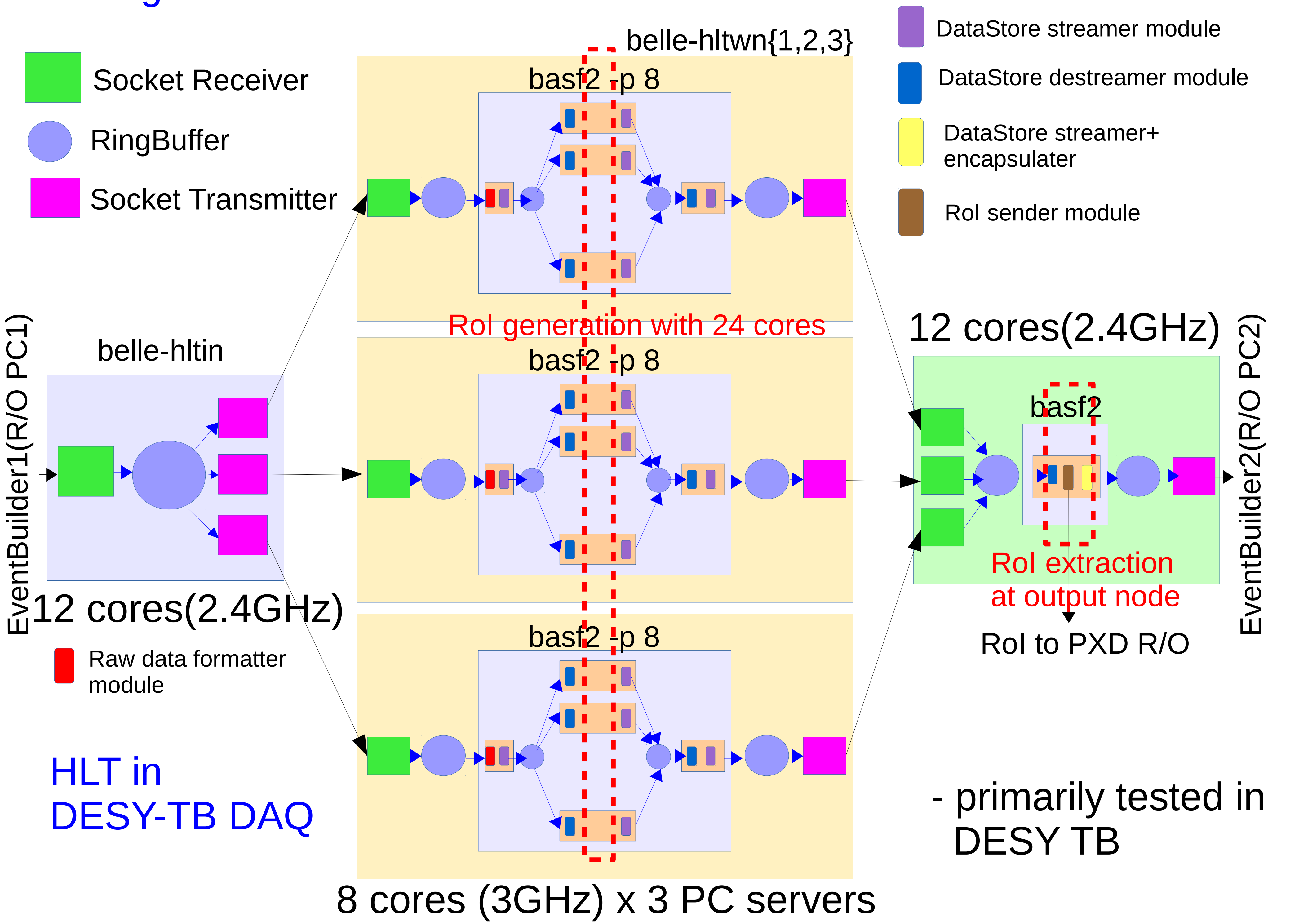
Parallelizing object streaming/destreaming using multiple basf2 session



* Advantage of the use of basf2's parallel processing is to reduce the memory consumption. (fork() with "copy on write" implemented in Linux kernel)

* We need to adjust the number of basf2 sessions considering the bottleneck and memory usage.

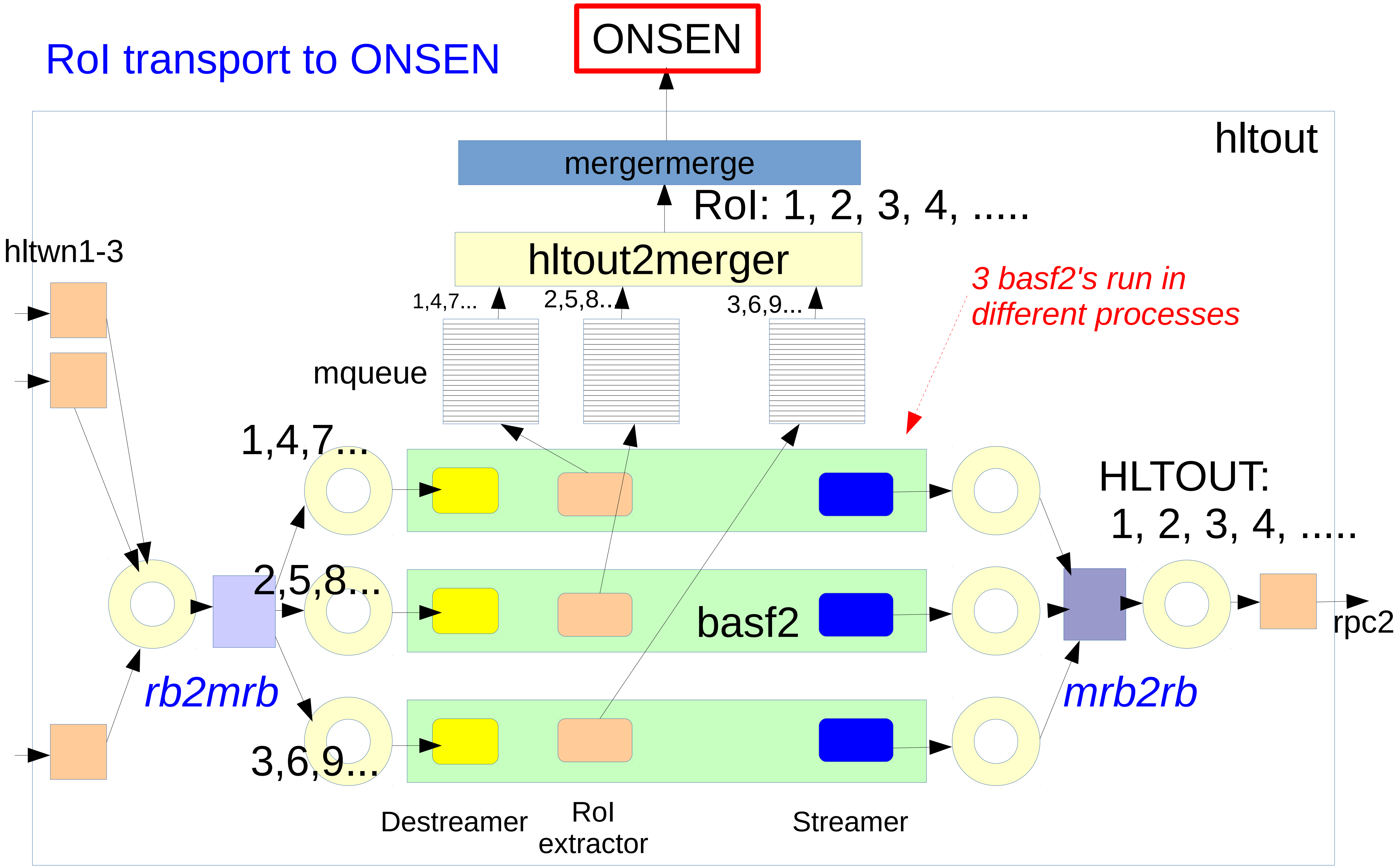
5. RoI generation for PXD data reduction



HLT in DESY-TB DAQ

- primarily tested in DESY TB

RoI transport to ONSEN



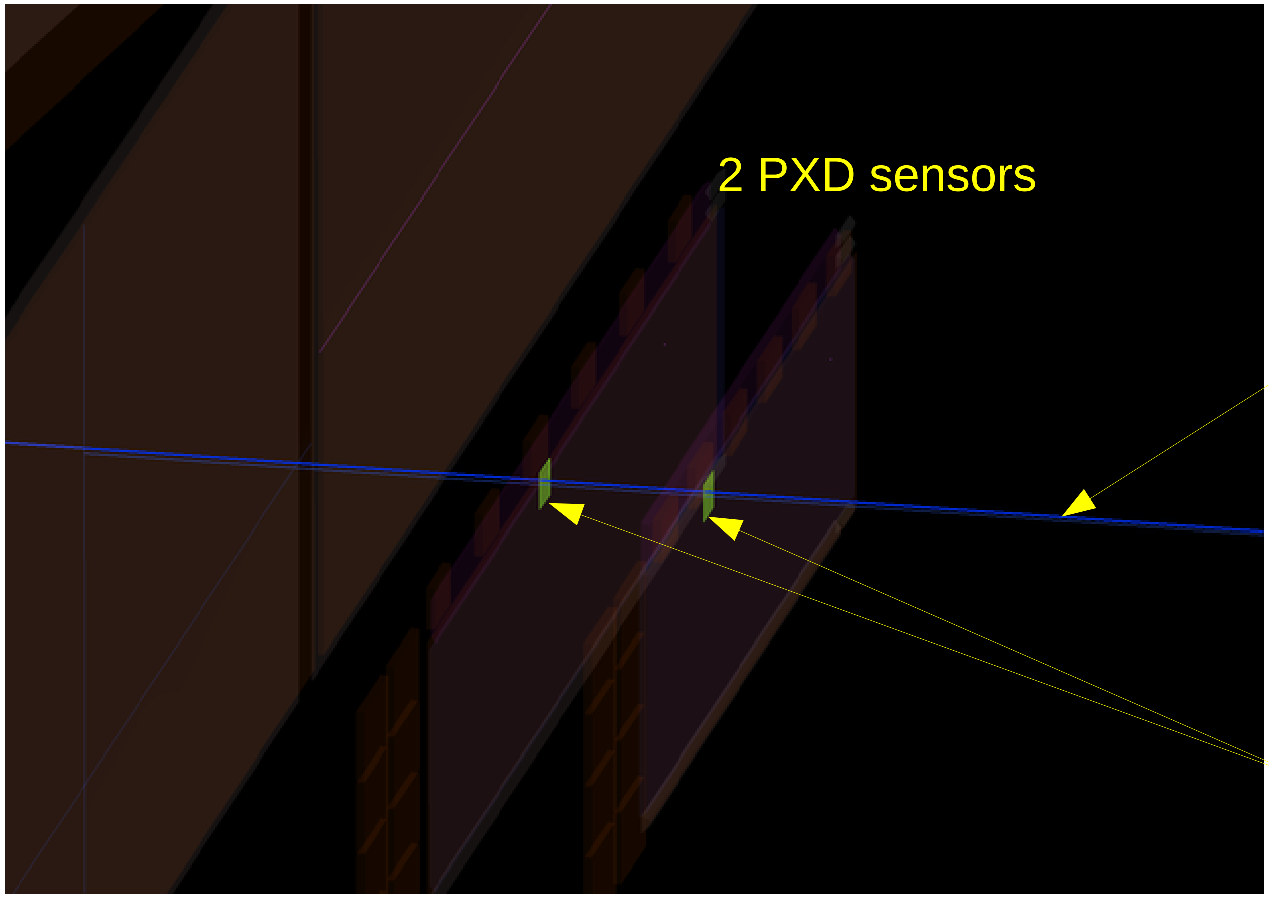
* rb2mrb, mrb2rb, and hltout2merger distribute/pick up records in turn to/from ringbuffers/mqueues in the same order.

Script used for tracking+RoI generation

```
# Unpack SVD
main.add_module(SVDUNPACK)        <- SVD unpacker
main.add_module(SVDDIGISORTER)    <- SVD digit sorter
main.add_module(SVDCLUST)         <- SVD cluster

# Track finding and fitting
main.add_module(vxdtf)      <- VXDTF
main.add_module(trackfitter) <-GenFitter

# ROI finding
main.add_module(roiprod) #standard ROI production   <- ROI by PXDDataReduction
main.add_module(roipayload) #payload for standard ROI production <- ROI payload
#main.add_module(roiprod_debug) # dummy ROI production
#main.add_module(roiprod_debug1) # dummy ROI production full frame ROI L1
#main.add_module(roiprod_debug2) # dummy ROI production full frame ROI L2
#main.add_module(roipayload_debug) #payload for dummy ROI production


# DQM
main.add_module(svddqm)          <- SVD DQM
main.add_module(vxdtf_dqm)       <- VXDTF DQM
#main.add_module(analyzer)
main.add_module(trackfit_dqm)    <- GenFit DQM

main.add_module(output)
main.add_module(progress)
#main.add_module(pcol)
```

Could be a good starting example of real HLT script

Real Time Event Display



track reconstructed by HLT

RoIs obtained from tracks

RoI generation-extraction-transport by HLT is confirmed to work!

6. Remaining issue: the robustness of HLT processing

- We are already very much aware of the importance of robustness in HLT processing in our experience at last DESY beam test, where we had a lot of HLT crashes caused by the segmentation faults in the tracking software.

- As a final solution to this problem, we started to develop a "self recovery" mechanism in basf2 framework. It detects various faults which cause aborts of the processing and restart the processing from next event automatically, while sending the problematic event to output with the "bad event" tag.

- Such a mechanism was already implemented in Belle's BASF and RFARM framework (that was the reason why we could operate RFARM stably in Belle), and we have enough experience and skill to implement.

- We are planning to complete the development by the time of next DESY beam test in December to be tested there. We will report the results at next BPAC meeting.

7. Summary

1. 3 HLT units are already installed and working.

2. The performance of real data processing is being tested using one of the unit with MC-generated raw data.

3. The performance bottleneck by object streaming/destreaming is foreseen, although it is not observed in the current test in the BB and tautau tracking reconstruction.

4. A mechanism to parallelize object streaming by running multiple basf2 sessions has been implemented to avoid the bottleneck. If the bottleneck is seen, we will turn on this mechanism.

5. The RoI feed-back scheme from HLT to ONSEN was already tested at DESY-TB and confirmed to work. Further test is scheduled in Dec.

6. A mechanism to recover from processing failure of one problematic event is required for the stable operation. Being developed.