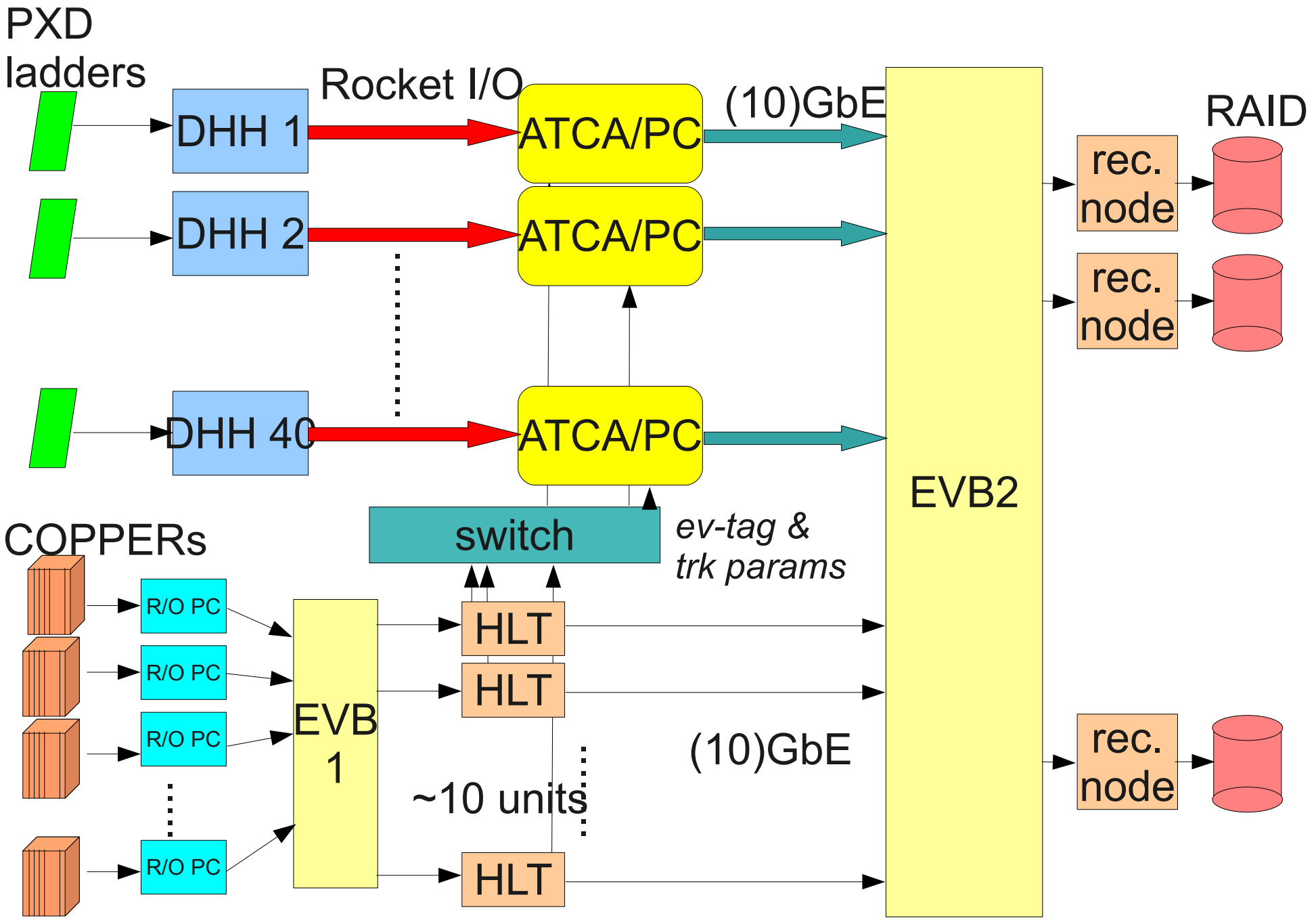
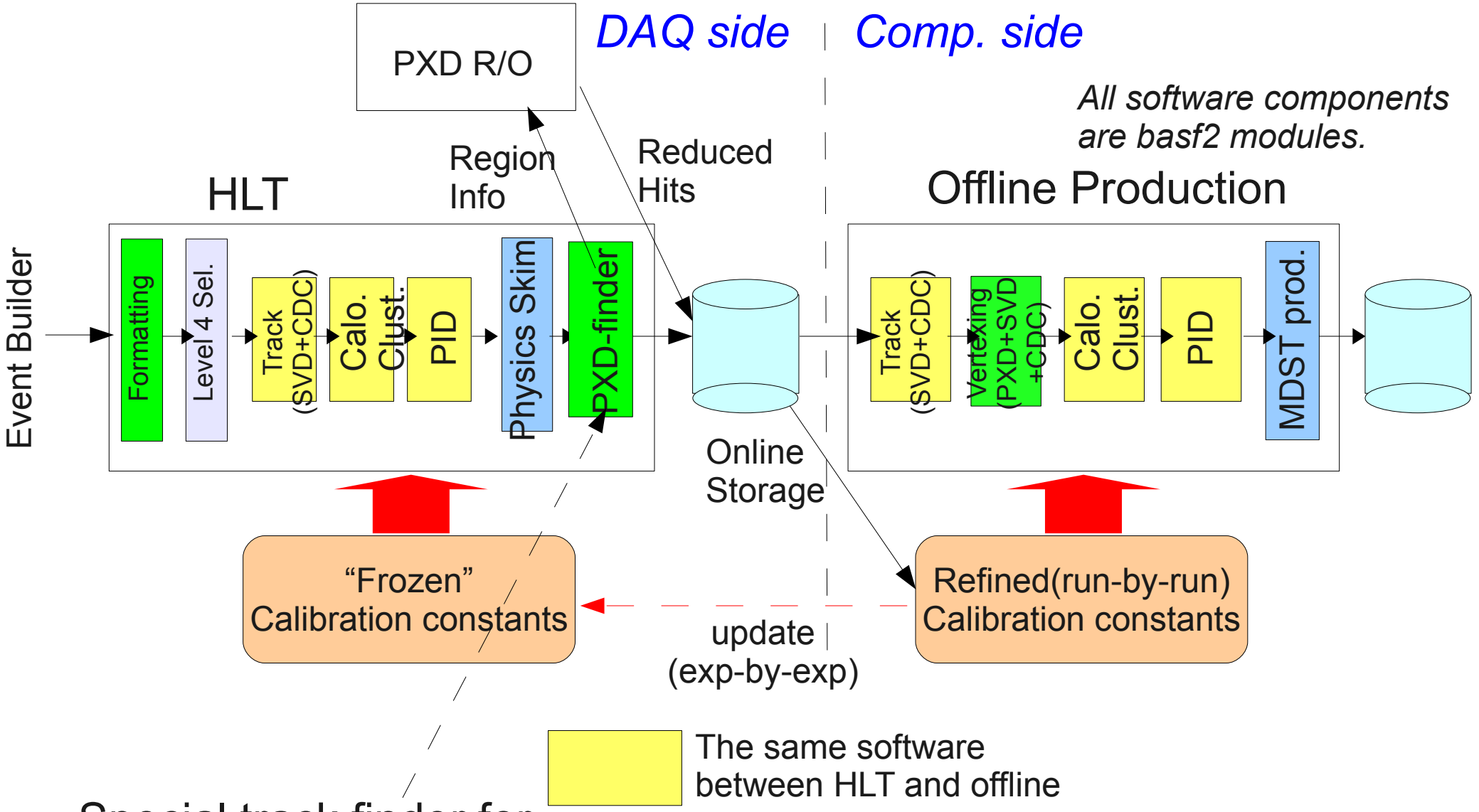
Role of HLT in PXD readout

R.Itoh, KEK

PXD integration



Event Reconstruction Chain at Belle II



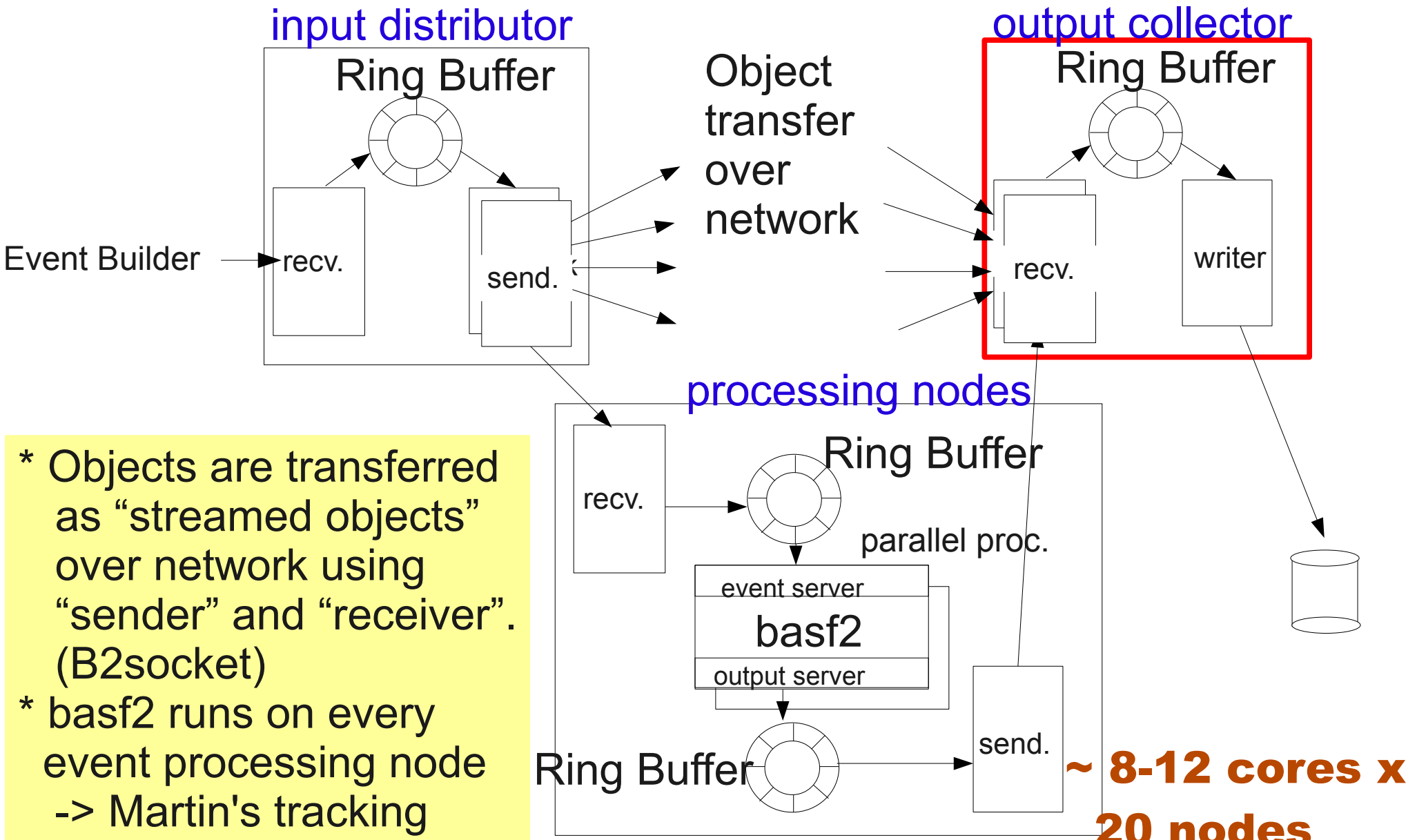
Special track finder for track-PXD hit association

* Only for the HLT-triggered events

*already parallel in 200 * 10 cores*

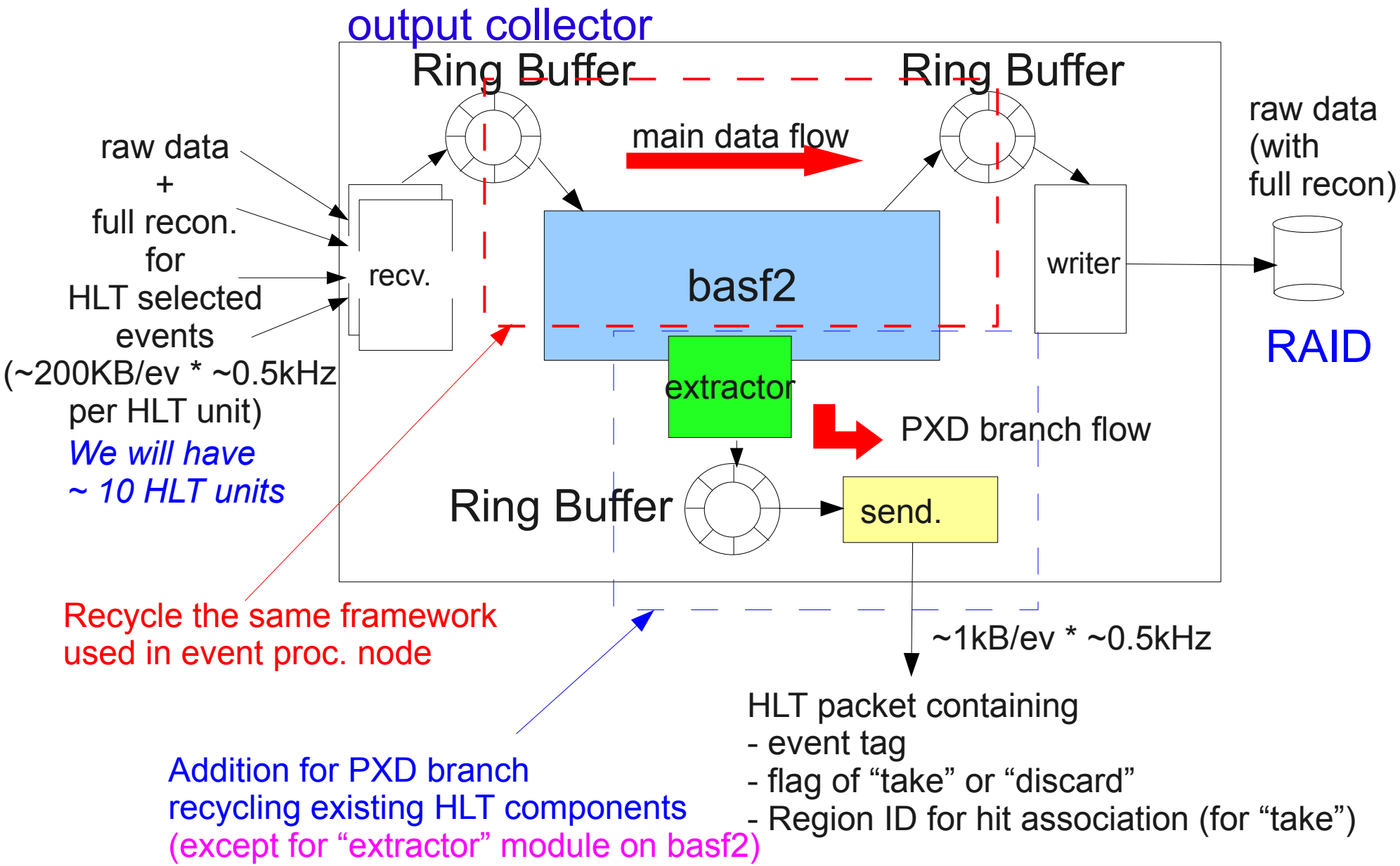
* Can make use of SVD+CDC raw data + full tracking results also

Software framework of HLT (1 unit) = pbasf2 for PC cluster



* Objects are transferred as “streamed objects” over network using “sender” and “receiver”. (B2socket)
* basf2 runs on every event processing node
  -> Martin's tracking framework runs “as is”.

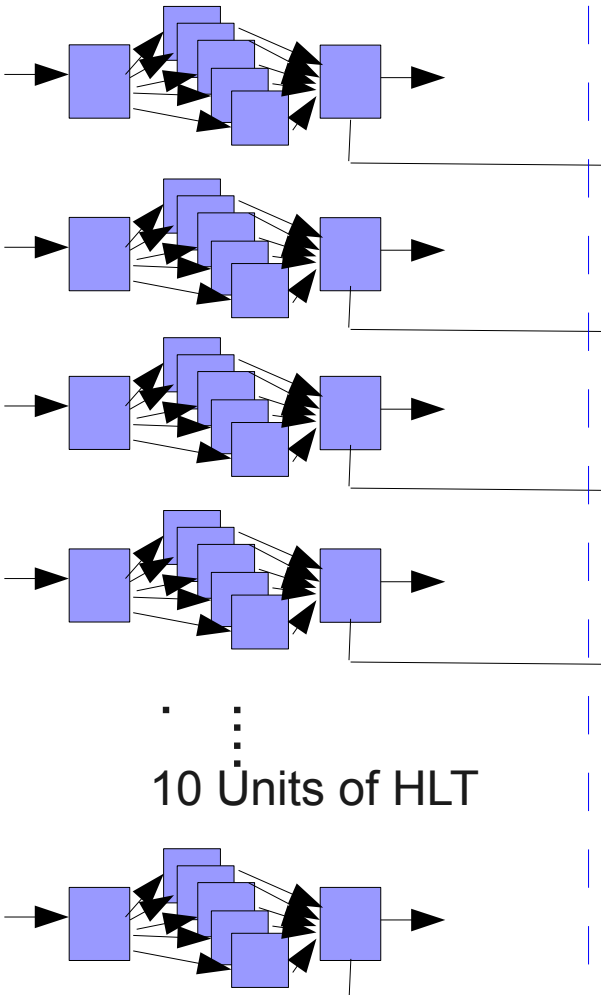
Modification for PXD data stream



“Design example”

Effort by HLT group

Effort by PXD-DAQ group



10 Units of HLT

1kB/ev * 2-3kHz / unit

Network Switch

~30MB/sec

PC

* Convert trk paremteres and ev tag in ROOT object to whatever you need (ex. area ID on PXD).
* Duplicate the record to number of receiving nodes and send them.

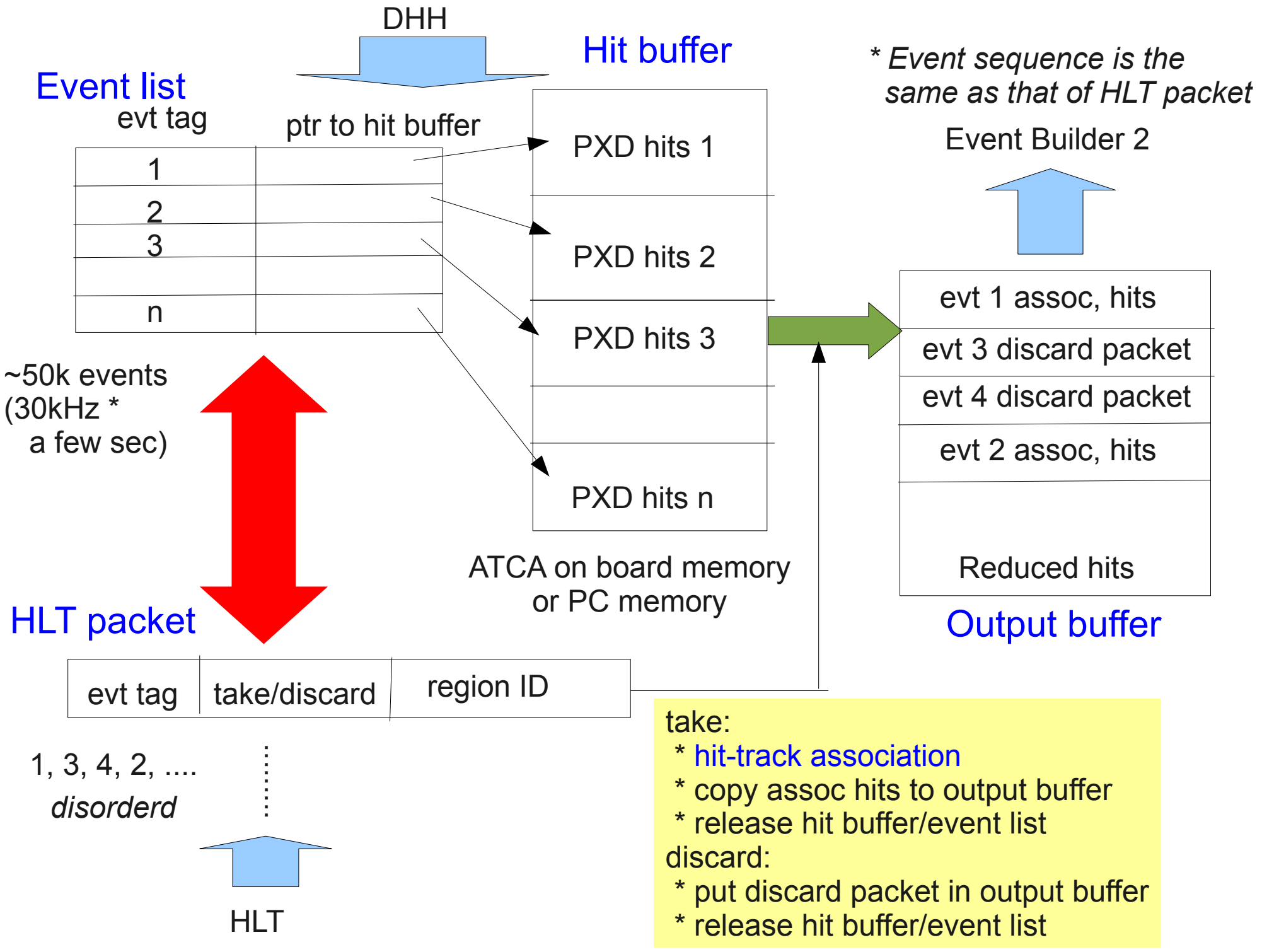
Network Switch

PXD readout nodes (ATCA CN or PC)

“HLT packets”
Passed events:
evtag + track parameters
Discarded events:
evtag + “discarded” flag

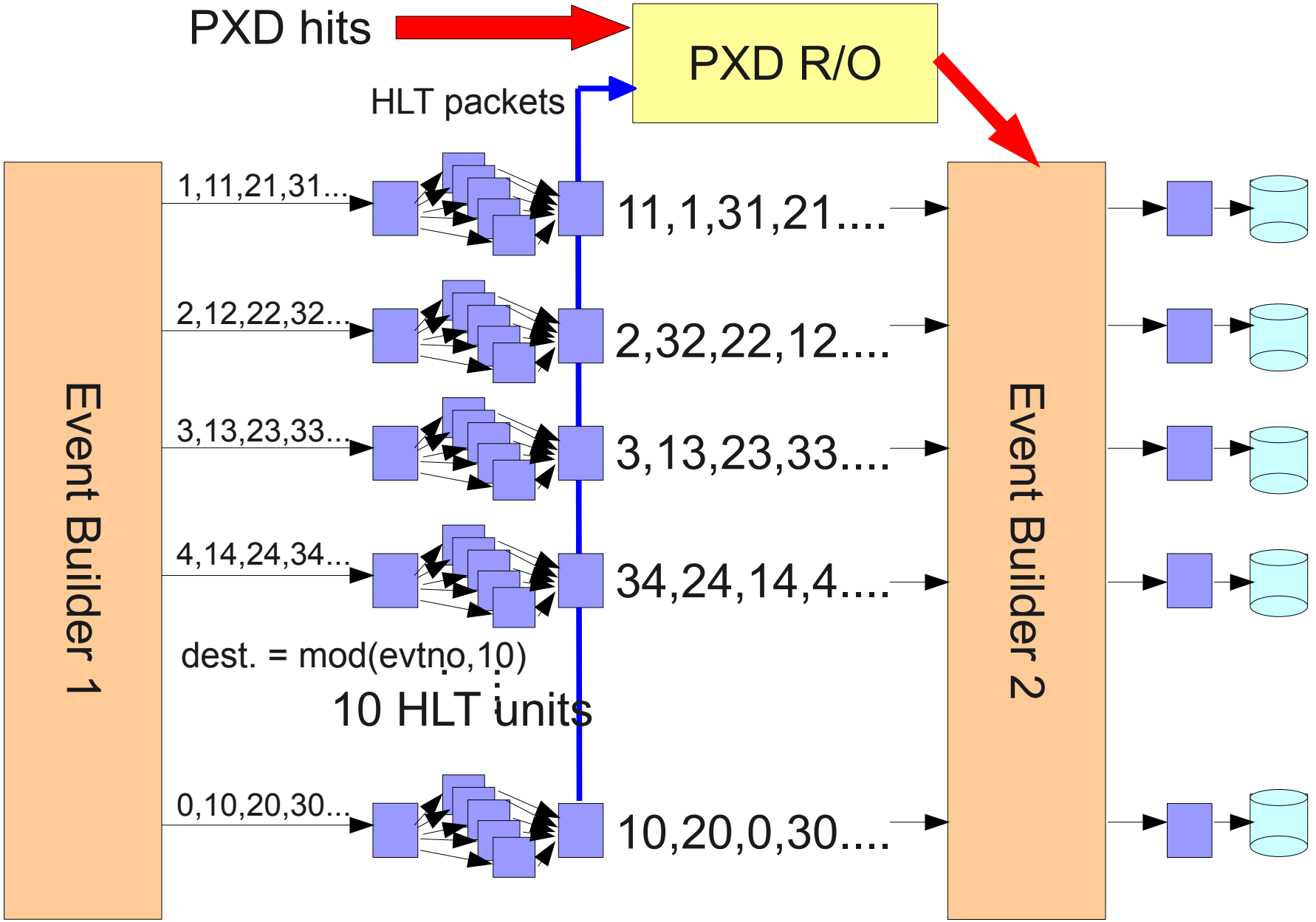
- “HLT packets” are sent to PXD R/O for all the events.
- HLT packets are disordered because of parallel processing

Processing framework in PXD R/O box for “baseline” and “backup” options



Idea on 2nd level event building (EVB2)

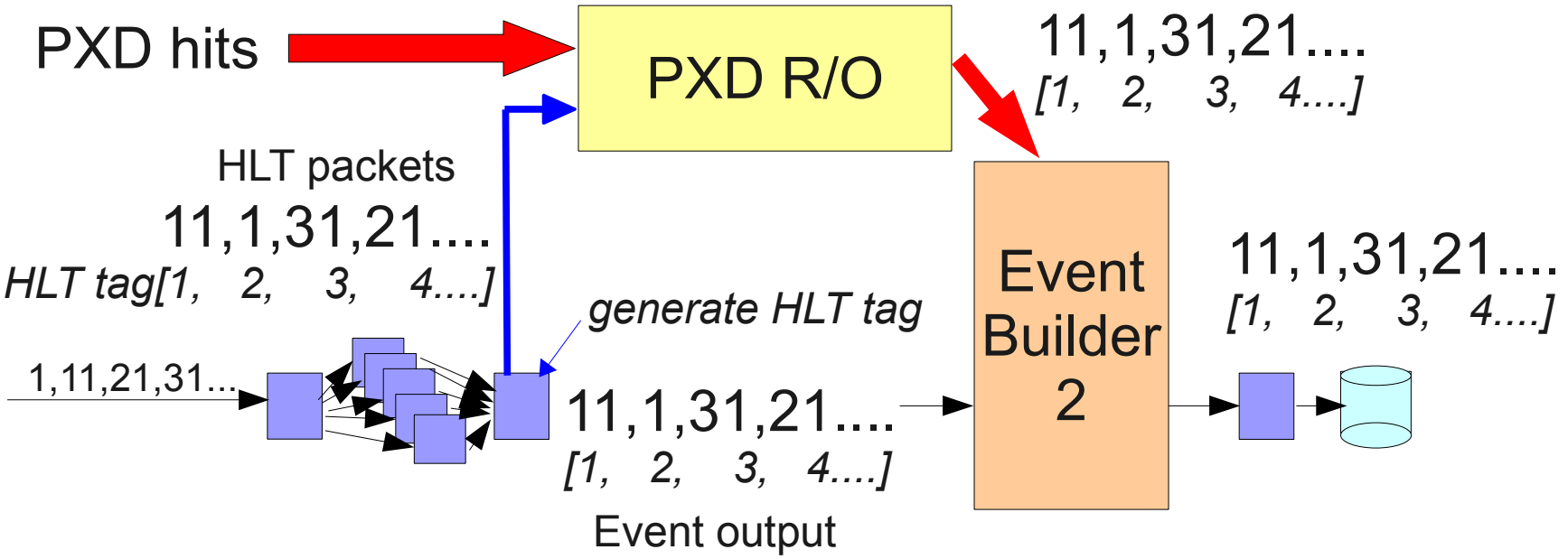
- Event sequence from HLT is disordered.
- Do we need to “sort” events for the 2nd level event building?

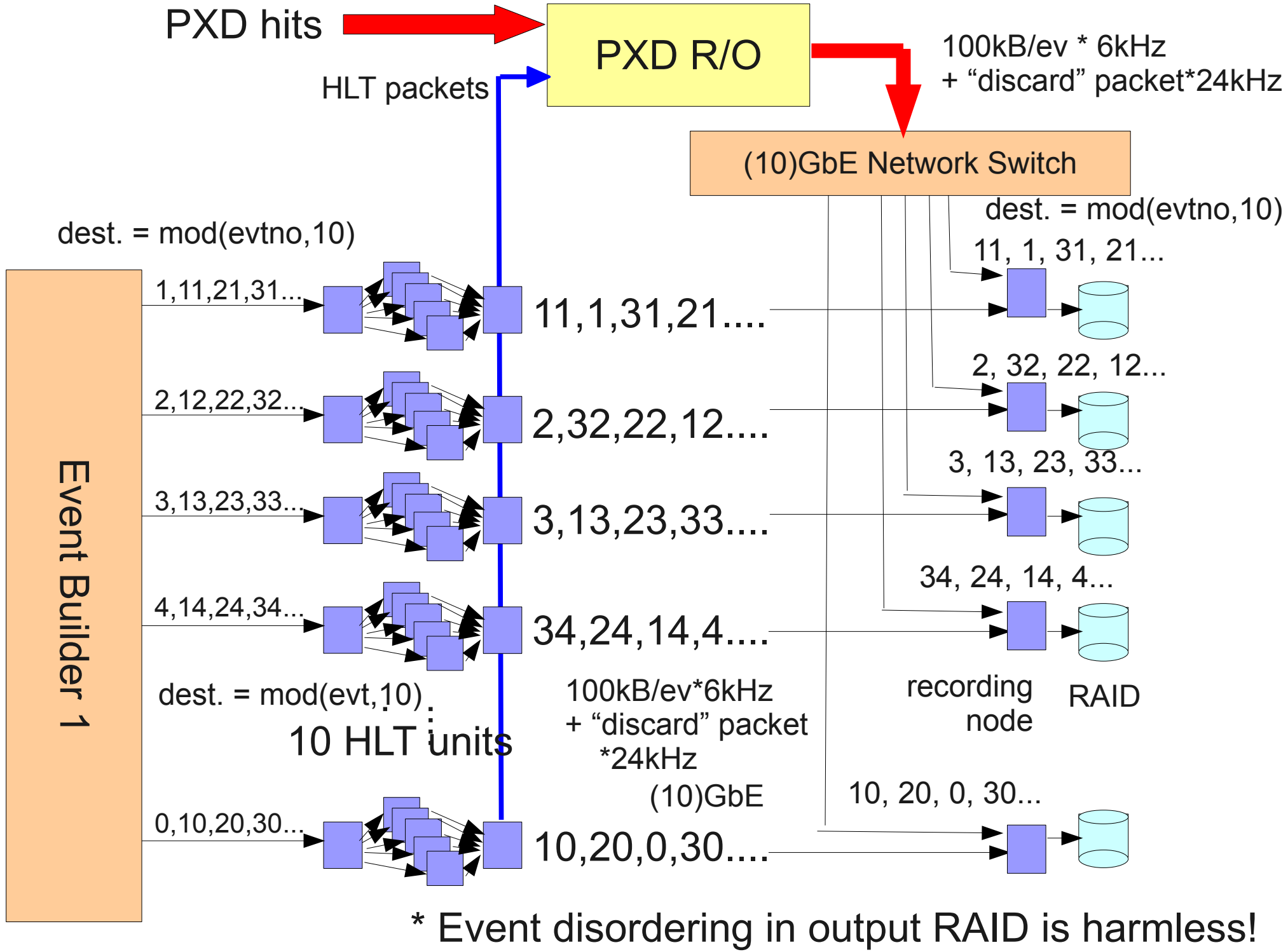


- HLT sends the event output to EVB2 and HLT packets to PXD R/O in the same order, although event tag is disordered.
  -> Idea: place new “HLT tag” both in “event output” and “HLT packet” and build event using the tag at EVB2.

- Processing on PXD R/O does not break the ordering of HLT packets.

→ No sorting is required although the event tag is disordered in recording RAID.




PXD hits
PXD R/O
HLT packets
100kB/ev * 6kHz
+ “discard” packet*24kHz
(10)GbE Network Switch
dest. = mod(evtno,10)
dest. = mod(evtno,10)
11, 1, 31, 21...
Event Builder 1
1,11,21,31...
11,1,31,21....
2,12,22,32...
2,32,22,12....
2, 32, 22, 12...
3,13,23,33...
3,13,23,33....
3, 13, 23, 33...
4,14,24,34...
34,24,14,4....
34, 24, 14, 4...
dest. = mod(evt,10)
10 HLT units
100kB/ev*6kHz
+ “discard” packet
*24kHz
(10)GbE
recording
node
RAID
0,10,20,30...
10,20,0,30....
10, 20, 0, 30...
* Event disordering in output RAID is harmless!

Summary

- HLT is supposed to provide track information to PXD readout box for the track-hit association (baseline and PC options).

- The “event disordering problem” can be solved for “baseline” and “PC” options in some cases.

- However, in case of “challenge” option where another parallel processing in ATCA card is expected, the event ordering may be broken in ATCA card differently from HLT's.
  -> Needs more consideration on event disordering.....

- Waiting for the decision on PXD readout box.

Backup Slides

Do we really need 4GB/DHH for data buffering?

- Average HLT latency for an event is expected to be less than 1 sec as shown in previous meeting.

- Thanks to parallel processing, the HLT packets for events with shorter processing latency come faster. Not necessary to wait for the 5 sec latency for one particular problematic event.

- The PXD hit buffer is supposed to be released immediately after the hit-track association (if the algorithm shown in previous slide is used).

We don't have to buffer every event for 5sec.

* It seems the required buffer size is much smaller than 30kHz*5sec*600MB/sec.
  <- This size is necessary only when 5 sec latency is consumed by all of ~2000 cores in HLT at one time. ==> very rare!
* We need a simple MC study to estimate the actual size.
* Sophisticated memory management scheme is required on FPGA in case of Option 12, anyway.