High Level Trigger (HLT) and Trigger Software

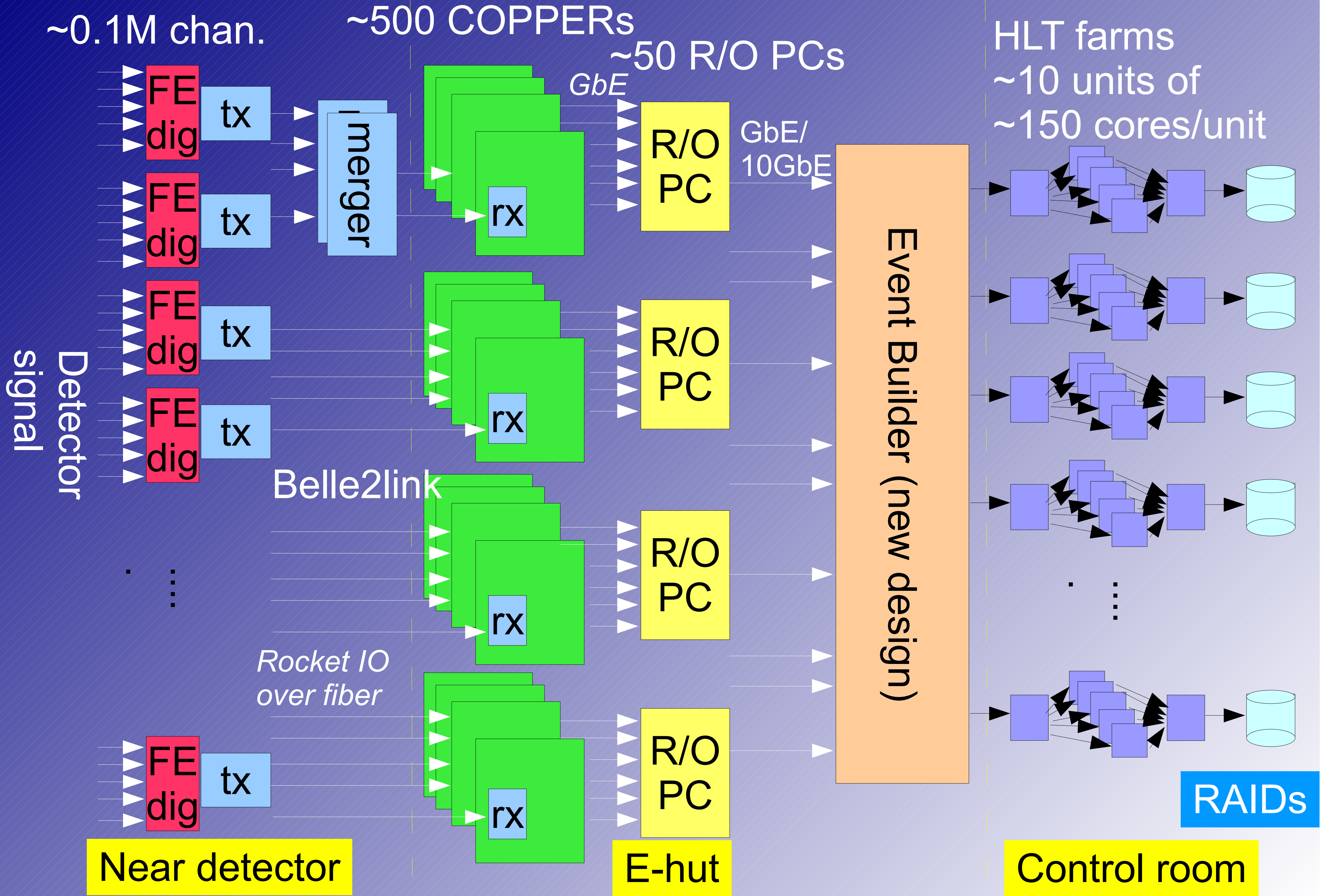
R. Itoh, KEK

Outline

1. HLT architecture
2. Role of HLT in Belle II DAQ system
3. Trigger software for HLT
4. Calibration constants and data base
5. Discussion

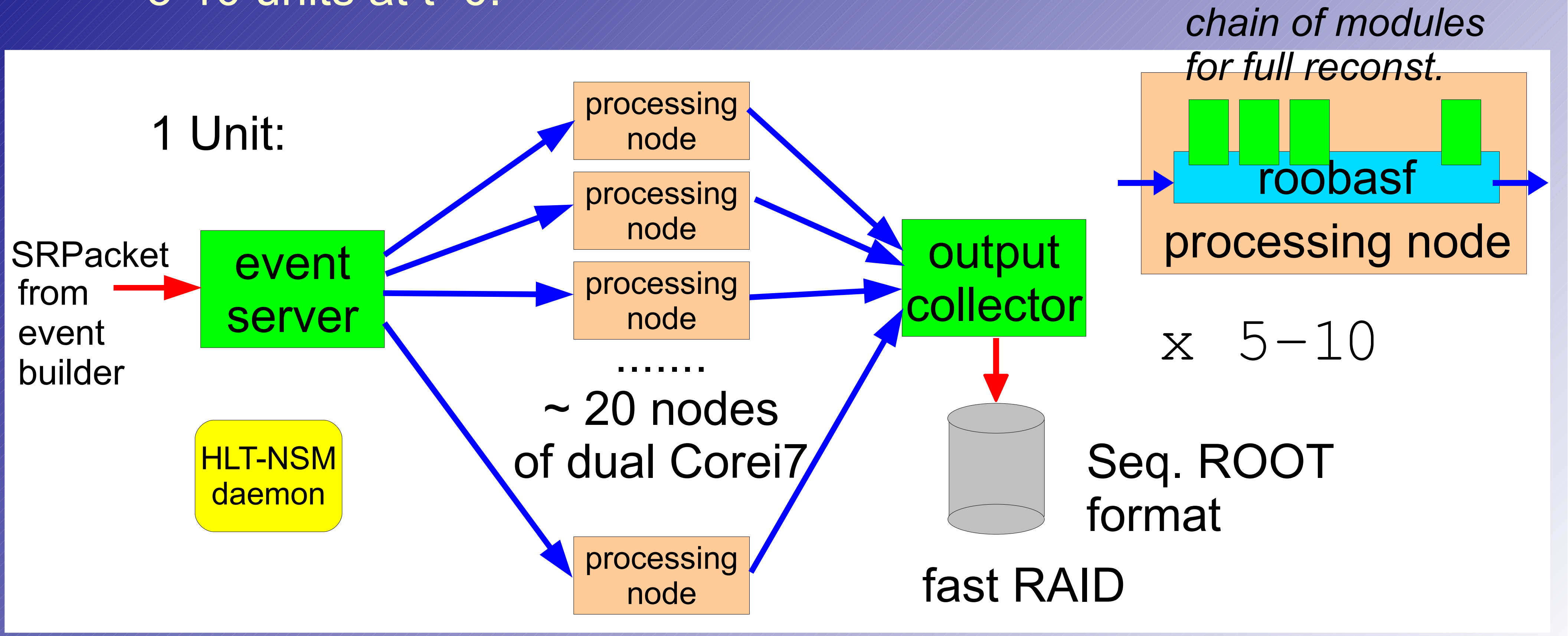
Global DAQ Design

* Timing dist. scheme is not included in this figure.



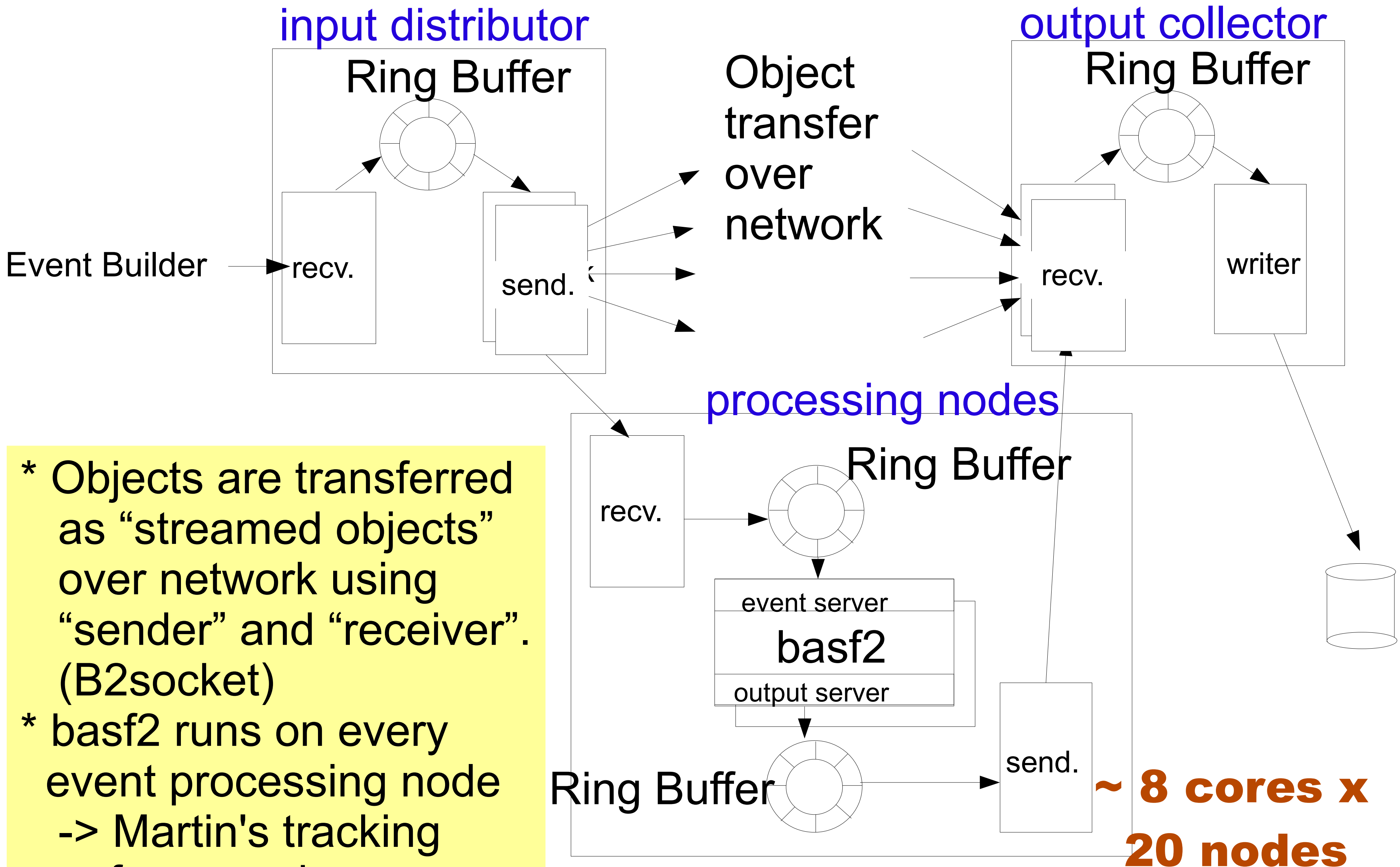
High Level Trigger (HLT) = RFARM@Belle

- Full event reconstruction chain identical to that in offline
- Massive parallel processing using a large number of processing nodes.
- Modularized construction to be scalable to the luminosity.
- 1 unit is supposed to process 2-3 x 10^34 luminosity.
  -> a module consists of ~20 nodes of dual Corei7(3.3GHz) servers
- ~5-10 units at t=0.



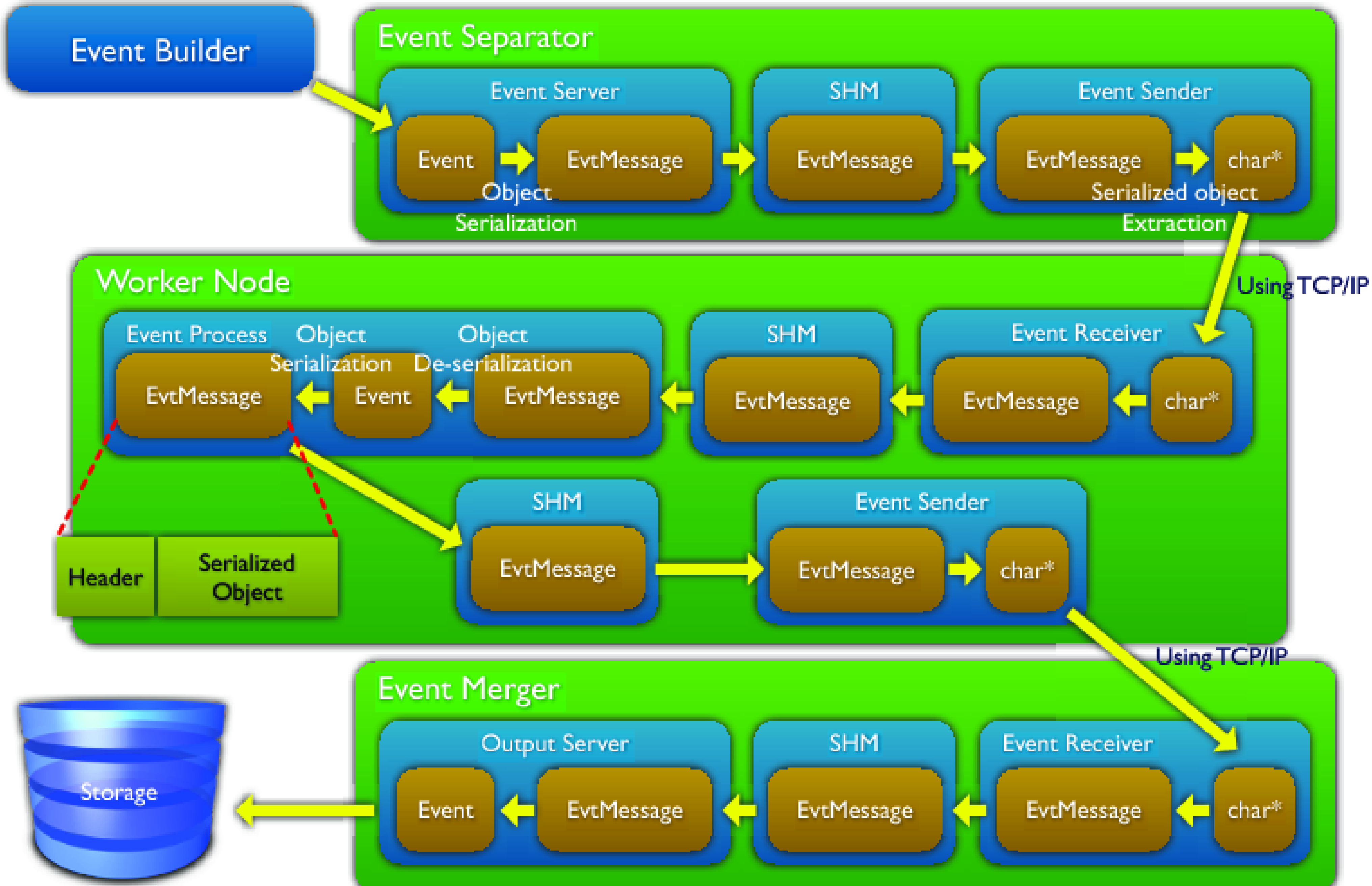
Development as a part of “roobasf” project

Software on 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".

Data Flow *based on pbasf2*



2. Role of HLT in Belle II DAQ

- Two aspects in the data reduction. One is the reduction of event size, and the other is the reduction of rate.

- The event size reduction is supposed to be performed on PXD readout processor and COPPERs/Readout PCs.

- The rate reduction after L1 is done only by HLT. We don't have level 2 trigger.

- Both reduction has close relation with the scheme to migrate PXD.

Event size / flow rate estimation

Table 13.1: Estimated average occupancy and data size and required number of subcomponents such as the number of Belle2Link for data transfer and number of COPPER modules. In addition to the listed subdetectors, trigger information is also planned to be read out in a similar way.

	#ch	occ. [%]	#link	/link [B/s]	#COPPER	ch size [B]	ev size [B]	total [B/s]	/COPPER [B/s]
PXD	8M	1	40	182M	—	4	320k	7.2G	—
SVD	243456	1.9	80	6.9M	80	4	18.5k	555M	6.9M
CDC	15104	10	300	0.6M	75	4	6k	175M	2.3M
BPID	8192	2.5	128	7.5M	8	16	4k	120M	15M
EPID	77760	1.3	138	0.87M	35	0.5	4k	120M	15M
ECL	8736	33	52	7.7M	13	4	12k	360M	30M
BKLM	21696	1	86	9.7M	6	8	2K	60M	10M
EKLM	16800	2	66	19.5M	5	4	1.4k	42M	8.4M

Note on event size estimation

- Event size estimation is still very preliminary.
  * The data size heavily depends on the detector occupancy, whose estimation is quite difficult for the moment.

  ex. PXD occupancy:
    - Previous optimistic assumption : 1% -> ~0.4MB/event
    - Recent concensus : 2% -> ~0.8MB/event

      + headers -> 1MB/event

  * The estimations of event size for other detectors also have large "errors".

    -> We tentatively take 100kB/event in total from detectors exept for PXD as the "maximum" size.

- Numbers shown in Table 13.1 are some kind of "typical" numbers.

-> We take 1MB for PXD and 100kB for others as the worst case raw data size in our DAQ design. (Safety factor of 2)

Event size reduction

1. Frontend readout except for PXD
   - By FPGA processing in FEE of each detector
     * Zero suppression
     * Feature extraciton from wave form sampling
   - Current estimation of event size is for after these processing.
     -> 100kB/ev at most.
   - Possibility of further reduction by software processing on COPPERs and Readout PCs. But we don't consider this in our current design to be conservative.

2. PXD data size reduction
   - Raw data size : 1MB/event (2% occupancy + headers)
   - The idea of data size reduction method is:
     * Select hits only around associated tracks in readout box
   - The track reconstruction:
     Full software tracking with SVD+CDC on HLT
     -> fine tracking -> reduction factor of 10 (MC study)
     -> ~100kB/ev. (could be ~200kB/ev. if red. fac is only a half.)

Rate reduction

0. L1 trigger rate
  - Estimation of typical rate : 20kHz at $8x10^{35}$ luminosity.
  - Maximum (average) rate assumption : 30kHz <- take this number

1. Frontend readout except for PXD
   - Output rate from FFE is the same as L1 trigger rate.
     -> 100kB/ev * 30kHz = 3GB/sec data flow

2. PXD rate
   - Rate reduction by HLT trigger (Option 2)
     * HLT rate reduction factor : 1/5
     -> 100kB/ev * 6kHz = 0.6GB/sec

Estimation of reduction rate by HLT

Experience at Belle

- Two level reduction

a) "Level 4" selection (= Conventional HLT trigger software)
  * Cut in event vertex obtained using fast tracking
  * Cut in total energy sum of calorimeter
  - Reduction rate is dependent on the beam condition
    Typical reduction factor ~ 50% (2006 beam condition)

b) "Physics skim"
  * Physics level event selection using full reconstruction results.
  * Almost 100% of physics analysis use so-called "hadronBJ" and "low multiplicity skims + some scaled monitor events.

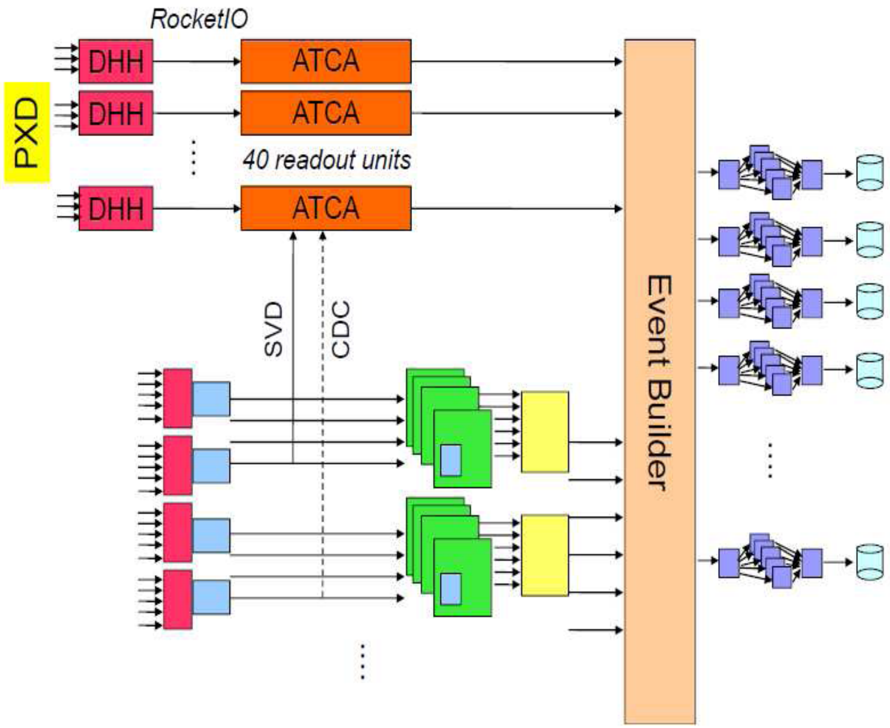
HadronBJ	: 14.2%
Low mult. ($\tau\tau$, 2photon)	: 9.6%
Monitor events (ee,$\mu\mu$...)	: ~1%
Total	: ~25% of L4 passed events

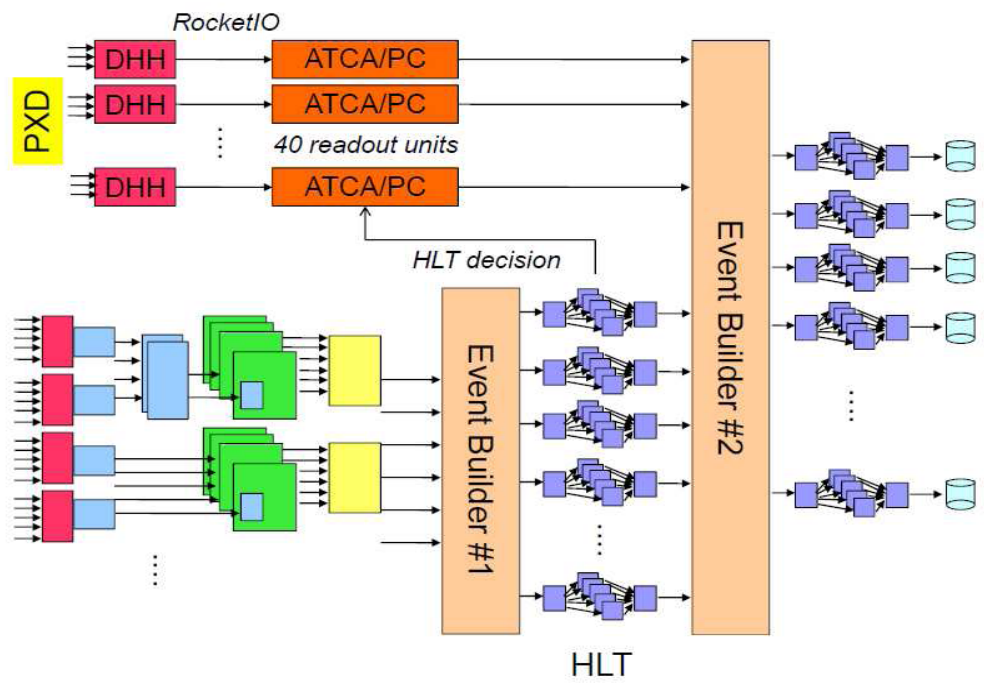
2004 experience

➡ Order of 1/10 reduction at HLT is possible in some cases.

Assume 1/5 reduction as a realistic number.

PXD integration





"Option 1"

- Noise reduction based on self-tracking by FPGA processing in ATCA modules with SVD (and CDC) data provided separately from DAQ data flow.

"Option 2 or 2' "

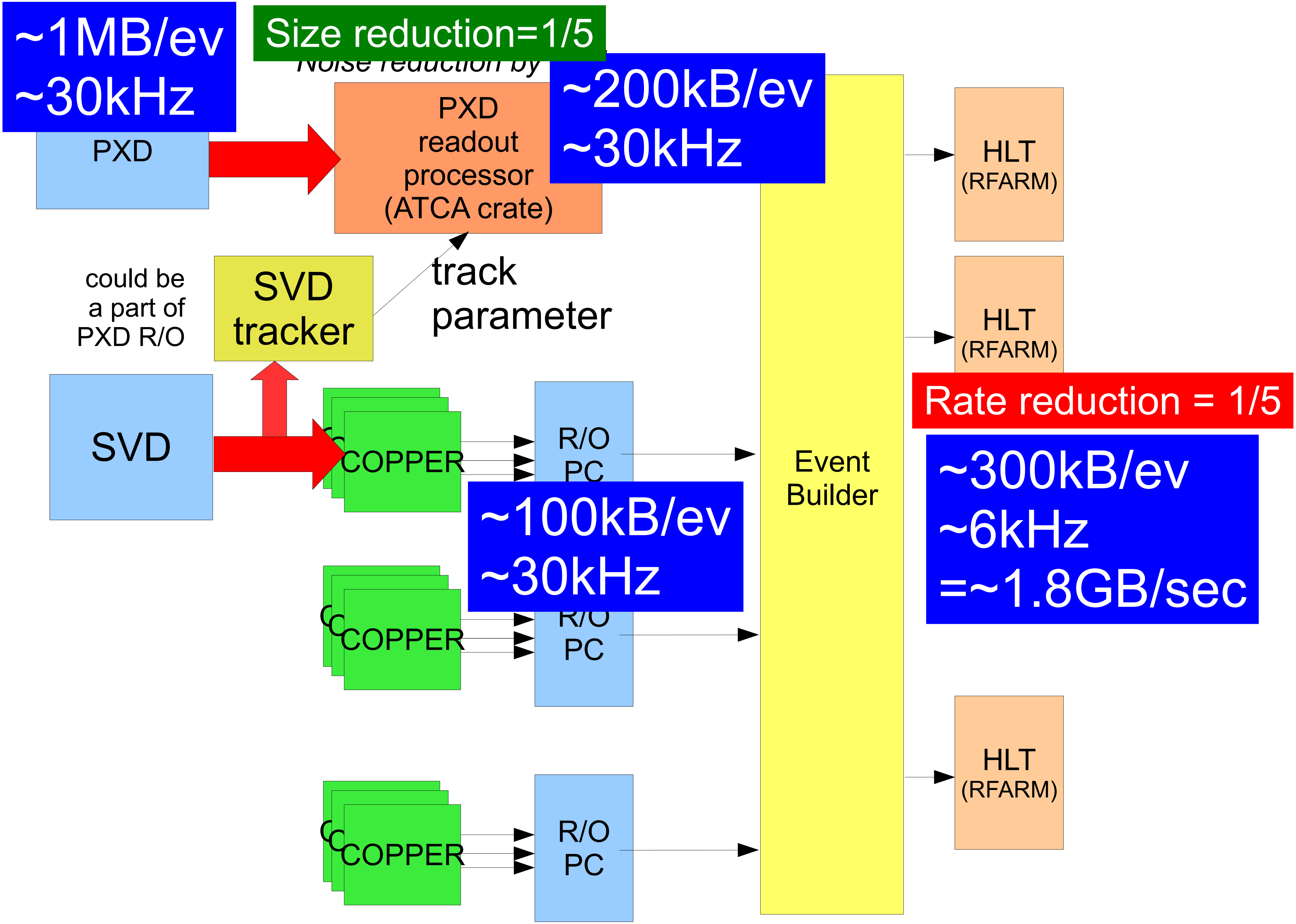
- Track reconstruction is done on HLT using CDC and SVD data.
- Track info for HLT accepted events are sent to PXD readout processor.
- Noise reduction using the track info is performed on PXD readout processor.

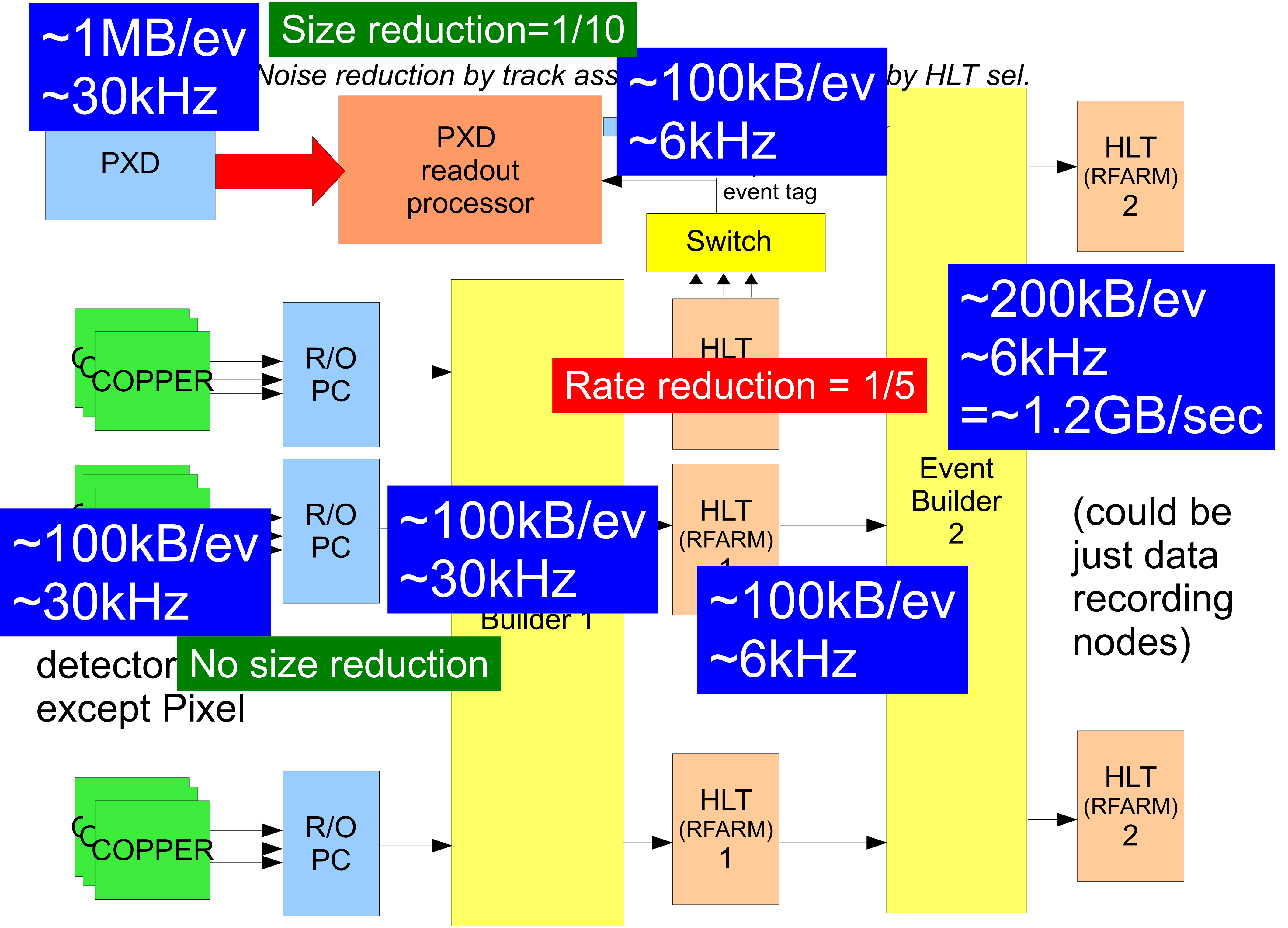
- HLT plays a major role in option 2 and 2'.

- The track parameters are calculated using the offline tracking software with full SVD+CDC data (i.e. Martin Heck's tracking framework).
  <- The same offline reconstruction code is supposed to run on HLT.
  => Very precise hit-track association
    -> Possible to narrow association region as small as possible
    Reduction factor of 1/10 is in scope.

- The reduction factor of HLT is expected to be at most 1/10
  (1/5 is assumed for low-background environment, though)
  <- estimation based on Belle's RFARM (HLT)

Reduction factor of 1/100 is possible in some cases!


~1MB/ev
~30kHz
Size reduction=1/5
Noise reduction by
PXD
PXD
readout
processor
(ATCA crate)
~200kB/ev
~30kHz
HLT
(RFARM)
could be
a part of
PXD R/O
SVD
tracker
track
parameter
HLT
(RFARM)
Rate reduction = 1/5
SVD
COPPER
R/O
PC
Event
Builder
~300kB/ev
~6kHz
=~1.8GB/sec
~100kB/ev
~30kHz
COPPER
R/O
PC
HLT
(RFARM)
COPPER
R/O
PC


~1MB/ev
~30kHz
Size reduction=1/10
Noise reduction by track ass
~100kB/ev
~6kHz
by HLT sel.
PXD
PXD readout processor
event tag
Switch
HLT (RFARM) 2
COPPER
R/O PC
HLT
Rate reduction = 1/5
~200kB/ev
~6kHz
=~1.2GB/sec
Event Builder 2
~100kB/ev
~30kHz
R/O PC
~100kB/ev
~30kHz
HLT (RFARM) 1
~100kB/ev
~6kHz
(could be just data recording nodes)
Builder 1
detector
No size reduction
except Pixel
COPPER
R/O PC
HLT (RFARM) 1
HLT (RFARM) 2

3. Trigger Software for HLT

- Use the event reconstruction software which is exactly the same as those used in the offline reconstruction

- The software trigger code = “Physics skim” code
  * Hadronic event selection for *B/D* physics
  * “Low multi” skim for tau physics and NP search

- Pre-selection software using fast-tracking (Level-3 like) is required to reduce CPU load on HLT (or 3D CDC HW trigger).

Estimated reduction : 1/5

* Need a close collaboration with Comp/Soft group.
  - Not only on software, but also on HLT architecture (ex. access to constant database, etc.)
  => Main reason why I come here!

* The processing latency is a critical issue for PXD integration to feed back reconstruted track informations to PXD readout processor.

Why the same offline reconstruction software on HLT?

- Since the trigger software on HLT is supposed to be the real “physics skim”, the reconstruction should be 100% compatible with the one in offline

- We don't want to introduce additional systematics in the HLT event selection by using different recon soft.

↑

Critical for the study of rare decays sensitive to trigger efficiency.

HLT latency

* The design of HLT->PXD interface heavily depends on the HLT latency, in particular, the buffering depth for PXD data flow.

* Current assumption is "5 sec. at most".

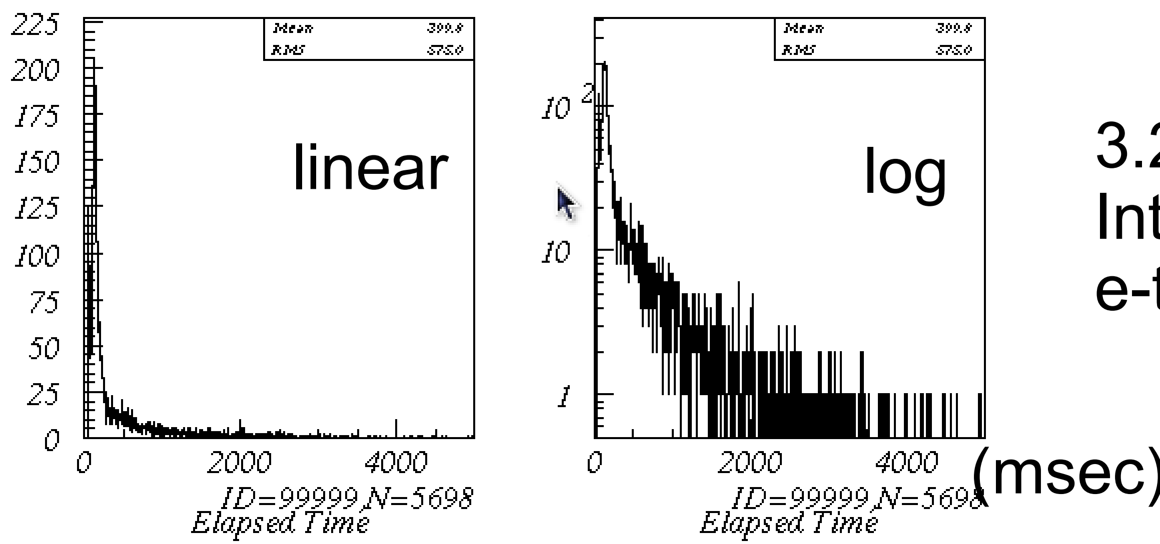
* With the assumptions of
    - Typical occupancy : 2%
    - Maximum L1 rate : 30kHz
        => Data flow per DHH = 600 MB/sec

  The buffer depth is required to be
        600MB/sec * 5 sec = 3GB per DHH.

* Considering the safety margin of ~50%, the buffer size should be ~5GB for a single DHH.
    -> Resulting in 5GB * 40 DHHs = 200GB in total.

Estimation of HLT latency

- Measurement using Belle's RFARM(=HLT) with current Belle reconstruction code.
- The processing time for full event reconstruction (incl. both full tracking + energy clustering) is measured for "L4 passed" events. (Exp.57, ~5000 events)



3.2GHz
Intel Xeon,
e-time / core.

- 5 sec. latency seems to be a reasonable assumption even though we take into the account the possibility of longer reconstruction time (~50% slower, for example.)
- 2.6 % of events takes more than 5 sec.
  -> Under investigation by Iwasaki-san
  Could be "junk events"?

Composition of HLT trigger software

Experience with Belle RFARM and offline DST production

- 2 step selection

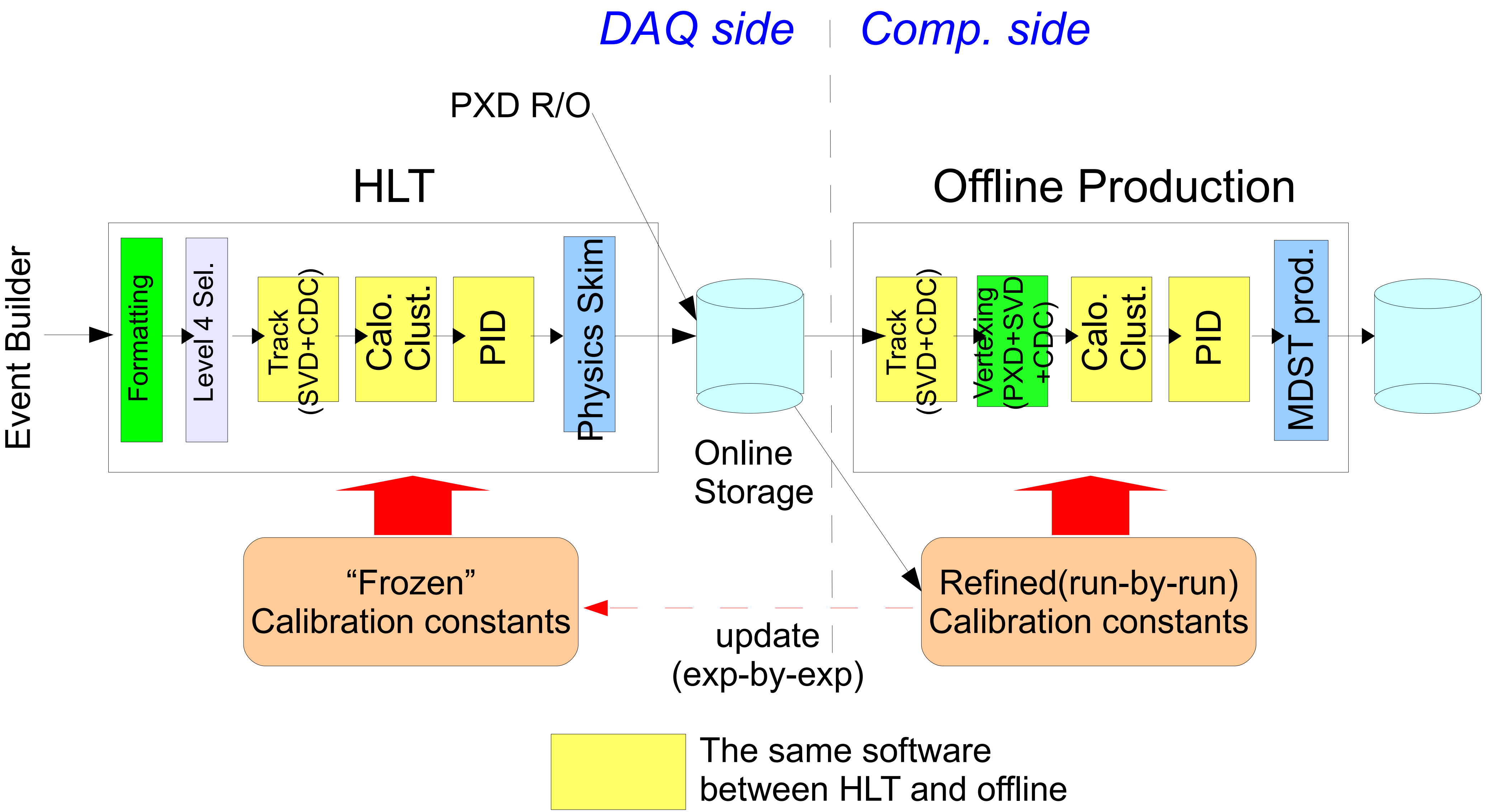
1) "Level 4" event selection
   * Fast CDC tracking + ECL clustering
     -> Rough event selection with a cut in
        * Event vertex position (d~ +-5cm)
        * Energy sum > 1 GeV
     -> Could be moved to L1 TRG in Belle II (3D track trigger, etc)

2) Full offline event reconstruction chain
   * Tracking using SVD + CDC, Kalman-filtering
   * ECL clustering
   * PID (eID, muID, atc_pid)
   * "Frozen" (or "course") calibration constants are used.

3) Physics skim (HadronBJ, LowMult, Monitor)
   * Currently done in offline at Belle.

Desired to have a similar software structure in Belle II HLT.

Expected Reconstruction Chain at Belle II



4. Calibration Constants and Database

- HLT uses “frozen” (= pre-determined, updated exp-by-exp basis) constants for reconstruction.

- The constants are assumed to be extracted from database which is common with offline.

- However, at the beginning of a run, many processing nodes access the database node simultaneously which results in a heavy network traffic.
  -> “run start” may become very slow.

- Need to think about the cacheing mechanism of database.
  -> can be a common problem with offline production.

Belle's RFARM:
- Database (postgres) is installed in each of processing node and database access is closed in the node -> no network access.
- The constants are updated at every change of exp number. (or on request, done during accelerator maintenance day.)

5. Discussion

1. Strategy of HLT software
   * The same offline recon code on HLT

   * Physics skim as the trigger software

   * Level 4 filter

2. Placement of reconstruction software components
   * Tracking : how to merge PXD data in tracking?
     <- It becomes available after HLT (in option 2,2').
     - HLT : SVD+CDC track finding/fitting
     - Offline : Track finding/fitting w/ fine constants.
       Add PXD for track refinment + vertexing

   * ECL clustering

   * PID

3. HLT latency (~ elapsed time for reconstruction per event)
  * Shorter latency is preferable for PXD integration (smaller buffering size)

  * 5 sec. at most is currently assumed based on Belle's experience.

4. Output stream
  * What to be left in “Online Storage”
    - raw data + formatted “hits”
    - Selected reconstruction results
    - Result of skim (<- some info for discarded events)

  * Multiple streams corresponding to number of HLT units.
    In addition, separate streams(files) for each event class?
    (HadronBJ, LowMult.......)

  * Size of Online Storage
    - Belle's case : The disk size is determined so that it can hold rawdata+DST for a week.
      -> Reprocessing or Data copy has to be done within a week.

5. Database access

* What is the Belle II database engine?
  -> postgres? MySQL? Other?

* Separation between geometry and calibration constants

* It might be better to have a local copy on each HLT processing node to minimize start-up time of HLT.
  -> Update policy/method of database.

* Better method? Database caching on each of local node. Support by database engine (postgres? MySQL?)
  -> Strategy of database caching in offline?