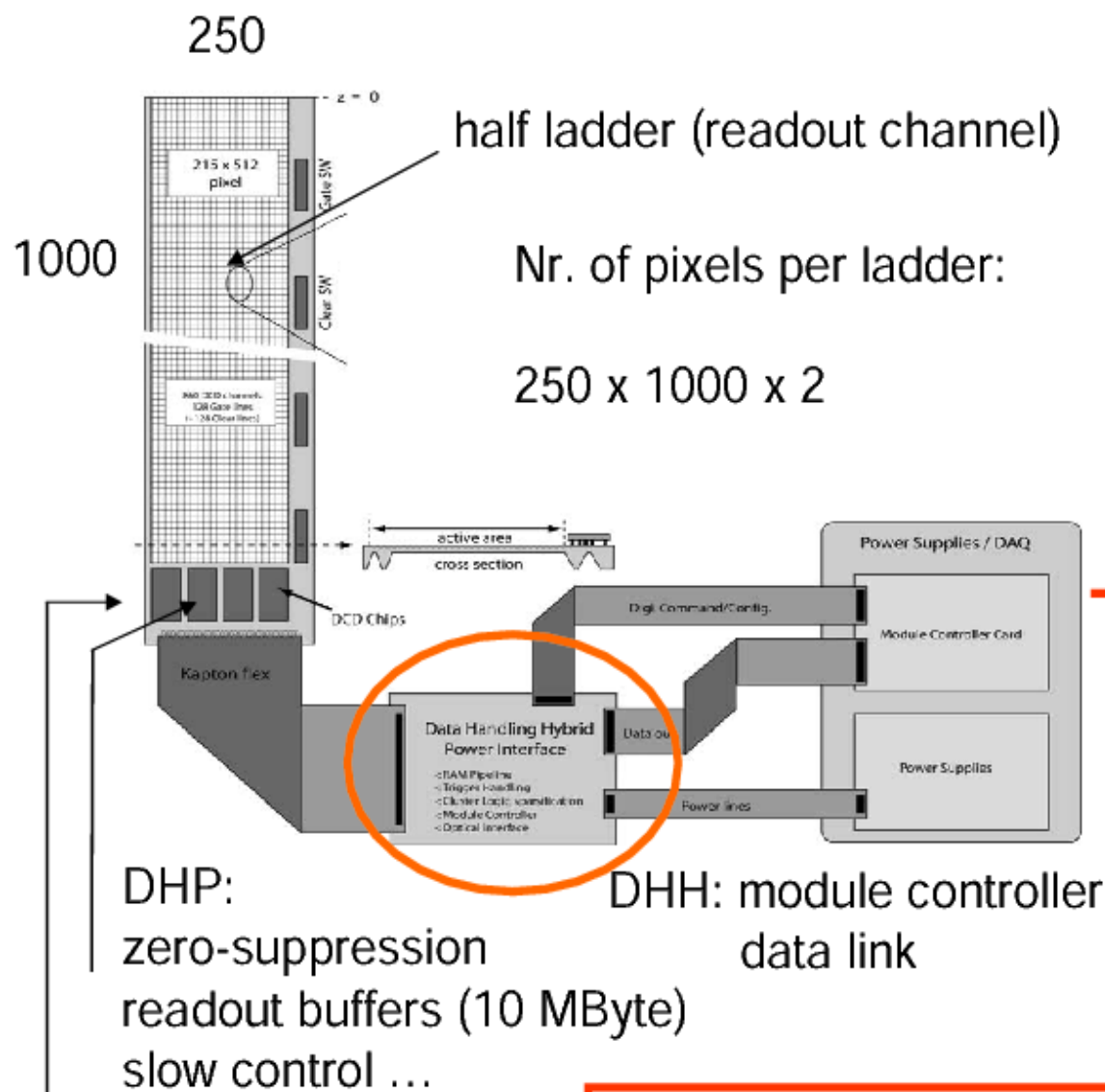


PXD data reduction using HLT processing results

R.Itoh, KEK

Overview of PXD DAQ Chain

Update: Nano beam option



- 40 half ladders:
10 Million pixels (px)
- 1-2% occupancy (?)
- 200 kpx on at any time
- 2 x 10⁵ px in each event
- 4 bytes per px (pos + ADC)
- 800 kB/event

Total evt size: ~ 1 MByte

Data Compression
(clustering):

→ 1/2

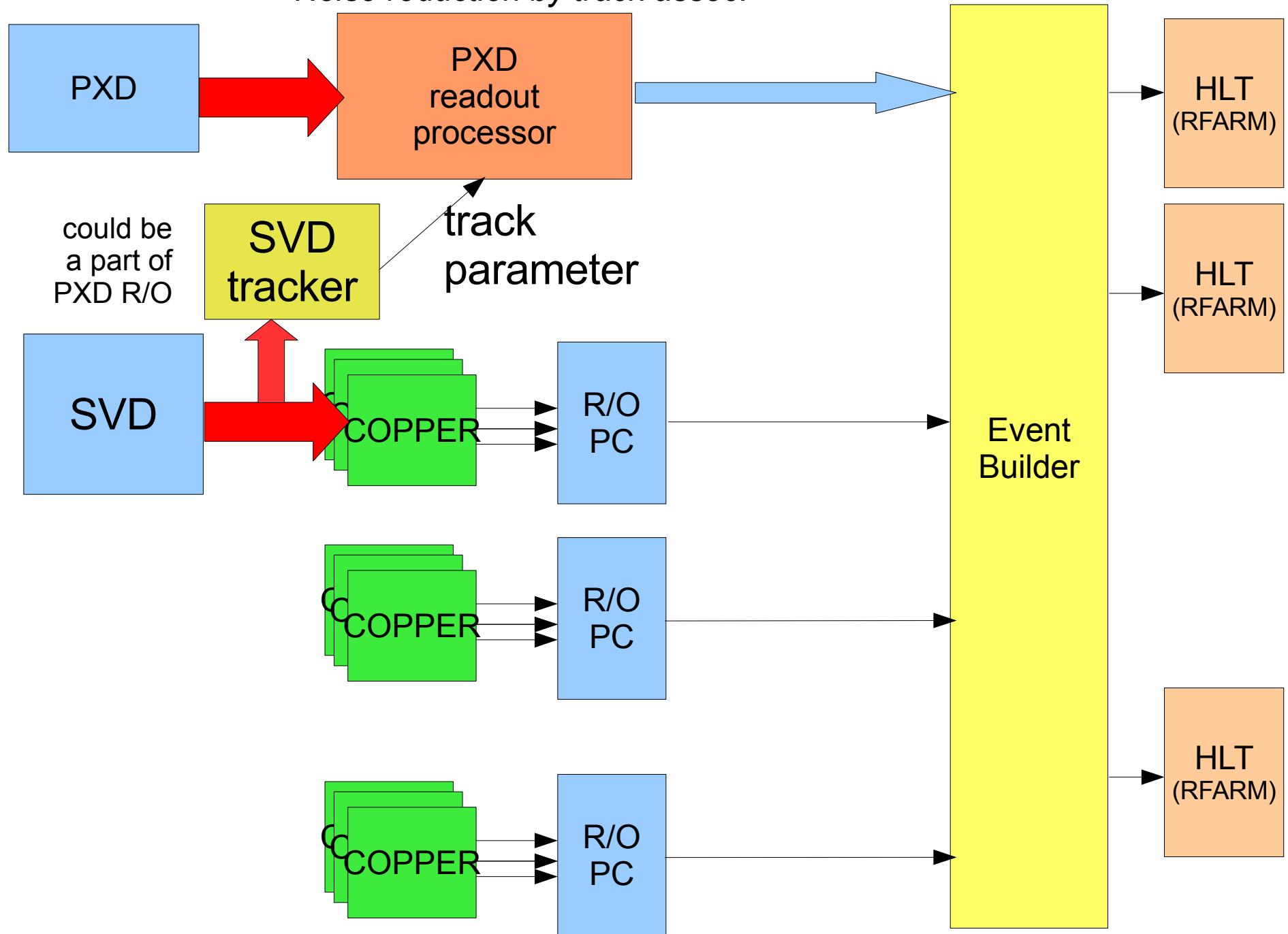
- The data size of 1MB/event is too huge when compared with other detectors.
- After the compression by clustering (using fast SVD track info), the reduction factor is estimated to be a half or so -> 500kB/event
- The expected total data size from other detectors is 100-200 kB/event and the PXD data size is still huge after the compression.
- Data flow rate to event builder could become quite unbalanced which may cause trouble in back-end processing.
 - * Maybe subdivision of PXD data flow into ladder-by-ladder can fix this problem, but we need larger network switch for the event building.
- Some worries on real time SVD-only tracking considering our bad experience of Belle's SVD1.5 level trigger



Better idea?

Pixel Integration: Option 1)

Noise reduction by track assoc.



Two approaches to reduce the data flow rate from PXD

1. The reduction of event size

- Noise-hit reduction by hit-track matching
 - * This is being studied by PXD group already
 - * The reduction factor could be $1/2$ - $1/10$ (?)

2. The reduction of rate

- The event selection at HLT can reduce the data rate down to $1/5$ to $1/10$ by using physics-level even skimming as the software trigger.

Belle's experience:

Cut in event vertex + ECL E sum (L4) : $1/2$

hadronic+low-mult. skim : $1/4$ (after L4)

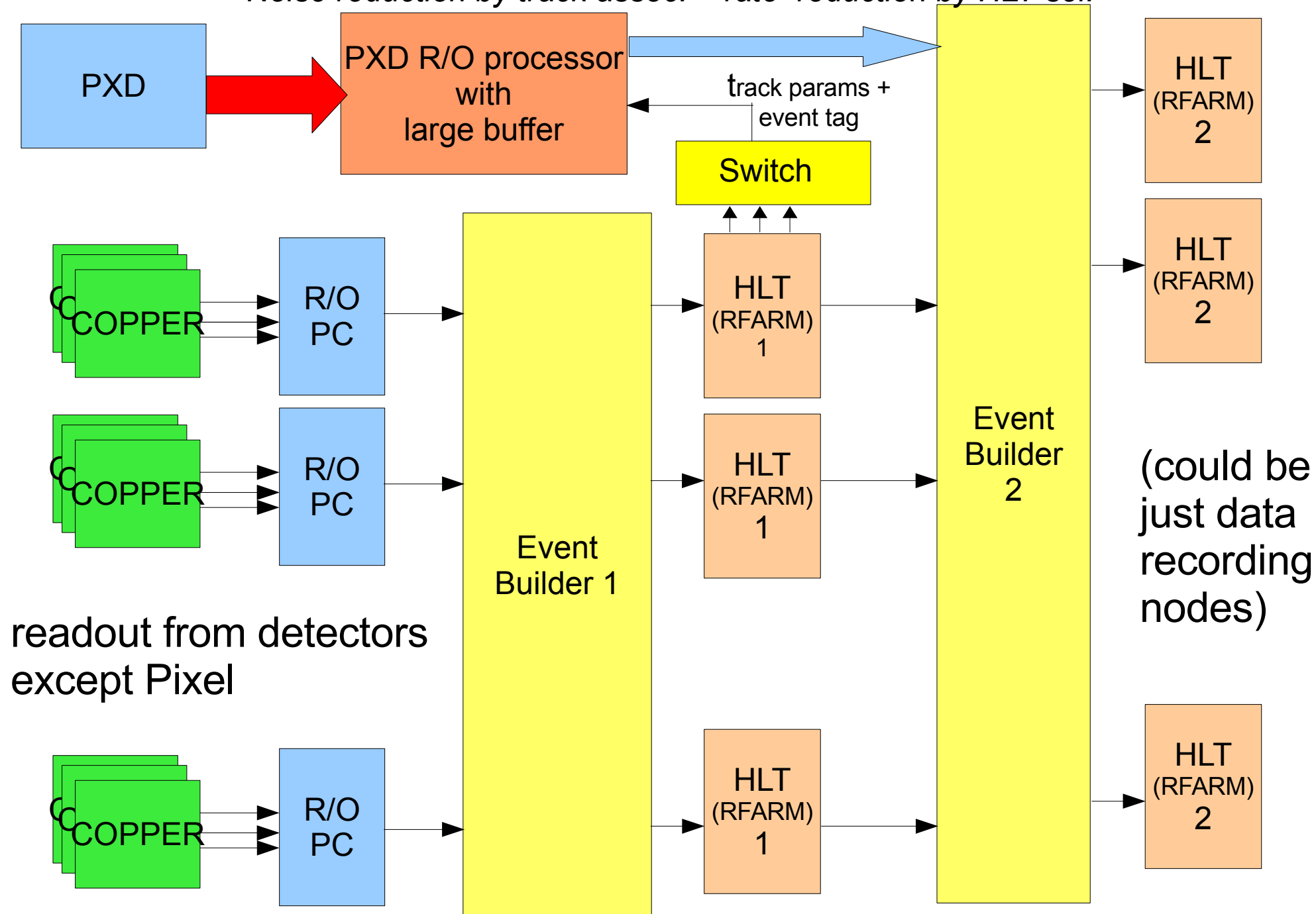
= total reduction factor : $1/8$

- We don't need PXD data for HLT event selection

By applying both of them to PXD data flow, the data flow rate can be reduced to $\sim 1/20$ or less.

Pixel Integration: Option 2)

Noise reduction by track assoc. + rate reduction by HLT sel.



Processing at PXD readout processor

1. Buffer whole PXD data up to a few seconds (HLT latency).
2. Receive event tag from HLT which passed HLT event selection together with the track parameters in the event
3. Noise reduction by track association.
4. Send reduced PXD data to 2nd event builder for the selected events only. Discard data for unselected events.

- Advantage to Option 1)

- * Precision of track parameter is much better
 - <- full tracking using offline-level reconstruction
 - * No special treatment of SVD data flow is necessary.
 - > Simple data stream
 - * Powerful data reduction both for the size and rate.
 - > taking the reduction factor 1/3 in size and 1/8 in rate
- PXD flow rate : $1\text{MB} * 20\text{kHz} / 24 = 830\text{MB/sec}$
- * Further reduction before recording is possible by further PXD data processing with full recon at 2nd HLT.

Critical paths in this approach

1. Need to implement a large buffer memory in PXD readout processor.

- * According to Iwasaki-san, it is possible to shorten the processing time of an event on HLT (=HLT latency) down to a few sec (say, 3 sec. at max.).

=> $1\text{MB (event size)} * 20\text{kHz (typical rate)} * 3\text{ sec} = 60\text{GB in total}$

- * If we use 12 ATCA boards for PXD readout, one board should contain at least 5GB. Desired to have ~10GB/board.

2. The event sequence is dis-ordered after HLT processing because of parallel processing nature.

- * PXD processor and Event builder 2 have to be prepared for the disordered event sequence.

- * If it is not possible, a mechanism to reorder the event sequence after HLT processing have to be implemented.

 - > This may require additional HLT latency resulting in a necessity of larger buffer memory (at least double size).