GPU based Level-3 Real Time Trigger for the Belle II High Level Trigger System

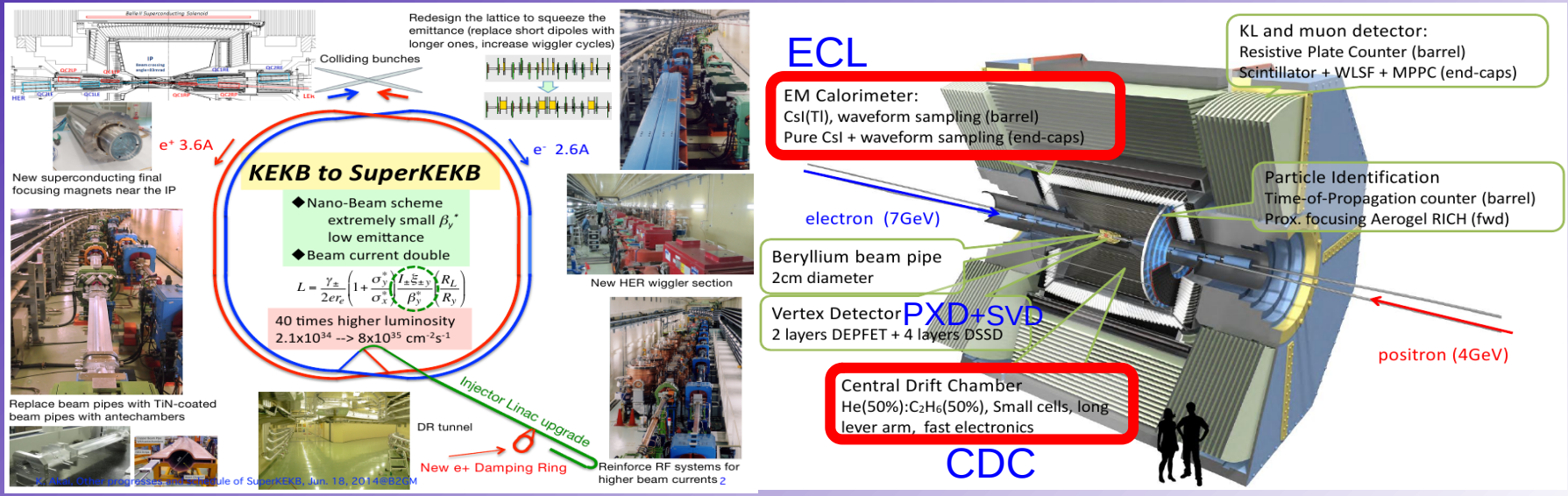
Ryosuke Itoh (KEK)
on behalf of
Belle II DAQ Group

Real Time 2026, La Biodola, Elba, Italy
May 25, 2026

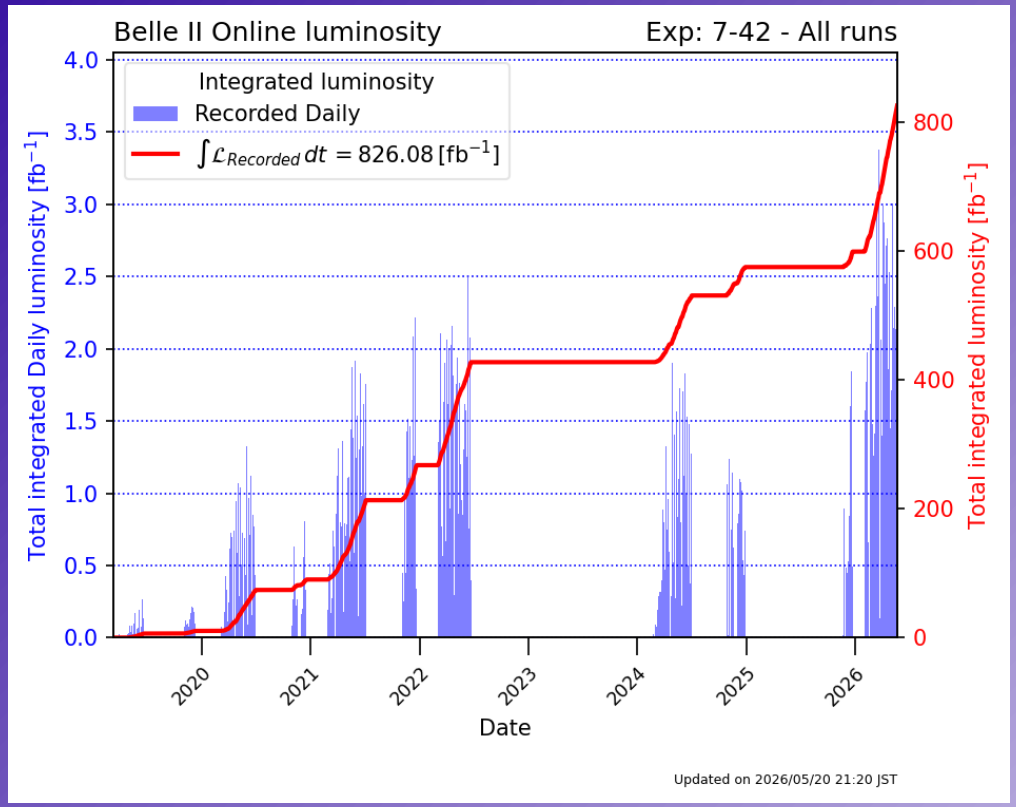
1. The Belle II experiment

- The Belle II experiment is a new generation B-factory experiment at KEK in Japan to search for New Physics(NP).

- A huge number of B mesons are produced in e^+e^- collisions in the SuperKEKB accelerator and NP is searched for in B meson decays by the Belle II detector complex.

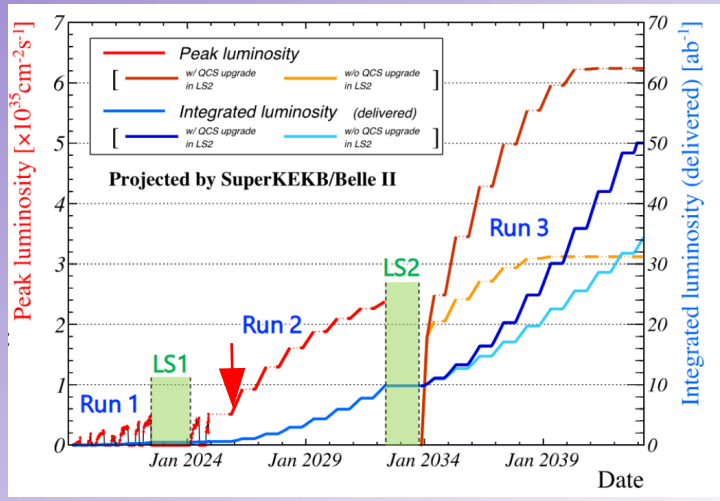


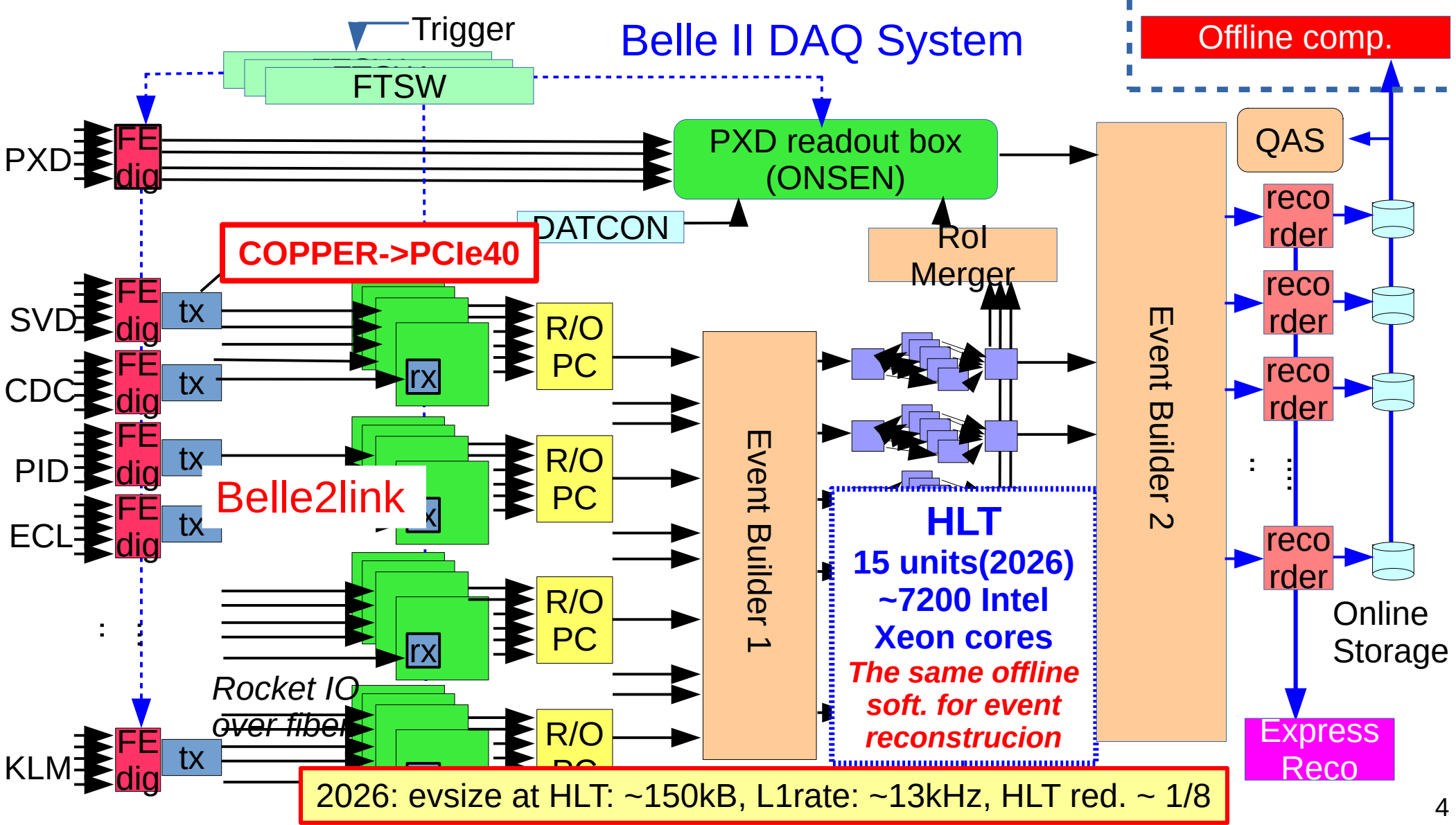
Belle II operation history

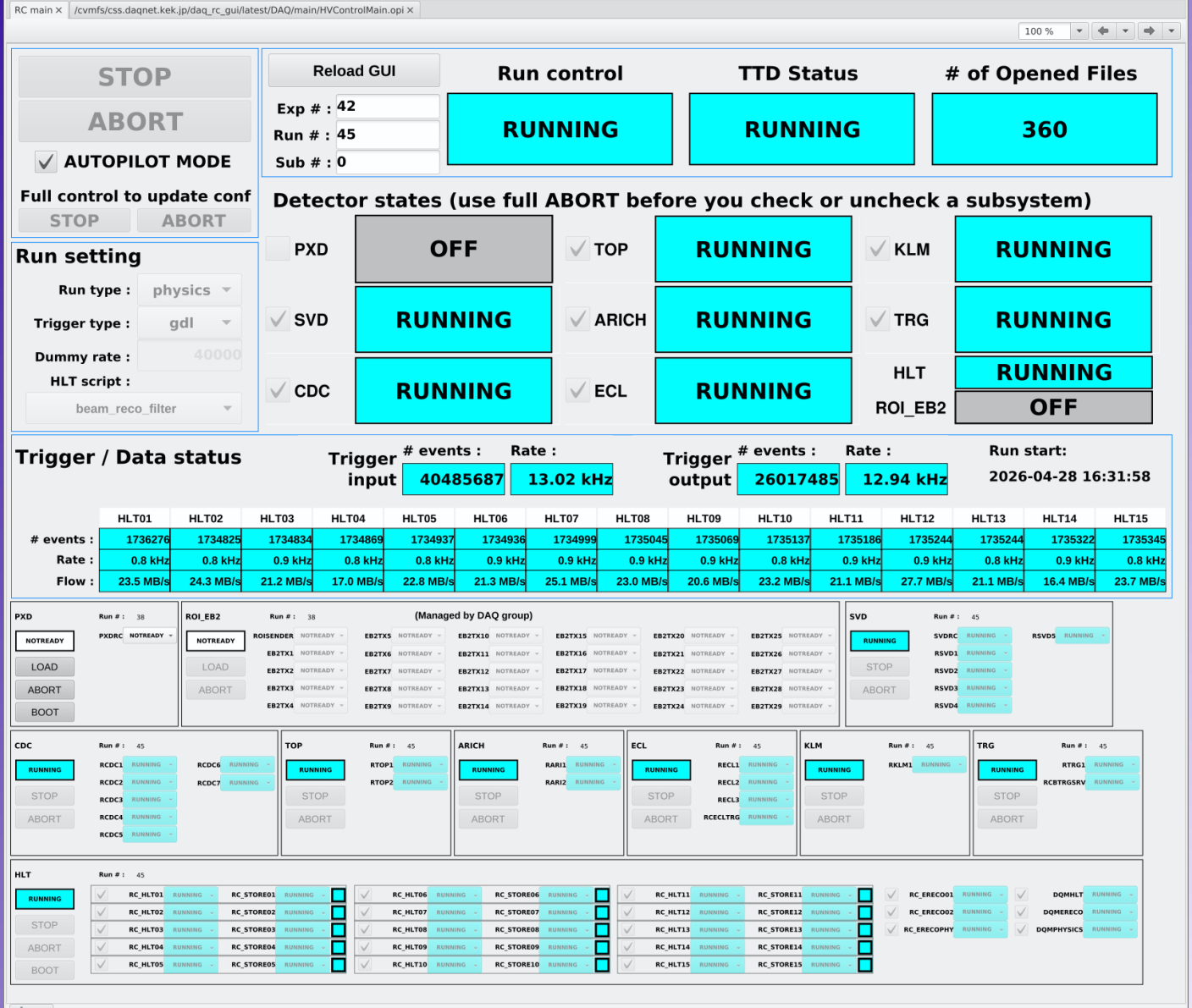


* Run 2 (Jan, 2024 -)
  L_{peak} = 5.24 x 10^{34} $cm^{-2}sec^{-1}$
  Integrated L ~ 830 fb^{-1}
  (Target luminosity 6 x $10^{35}cm^{-2}sec^{-1}$)

* Many difficulties in accelerator operation.
  -> Pixel Detector (PXD) is temporarily turned off.




Belle II DAQ System
Trigger
FTSW
Offline comp.
PXD
FE dig
PXD readout box (ONSEN)
DATCON
COPPER->PCIe40
RoI Merger
QAS
SVD
CDC
PID
ECL
KLM
tx
rx
R/O PC
Belle2link
Rocket IO over fiber
Event Builder 1
Event Builder 2
HLT
15 units(2026)
~7200 Intel Xeon cores
The same offline soft. for event reconstrucion
reco rder
Online Storage
Express Reco
2026: evsize at HLT: ~150kB, L1rate: ~13kHz, HLT red. ~ 1/8



PXD is turned off

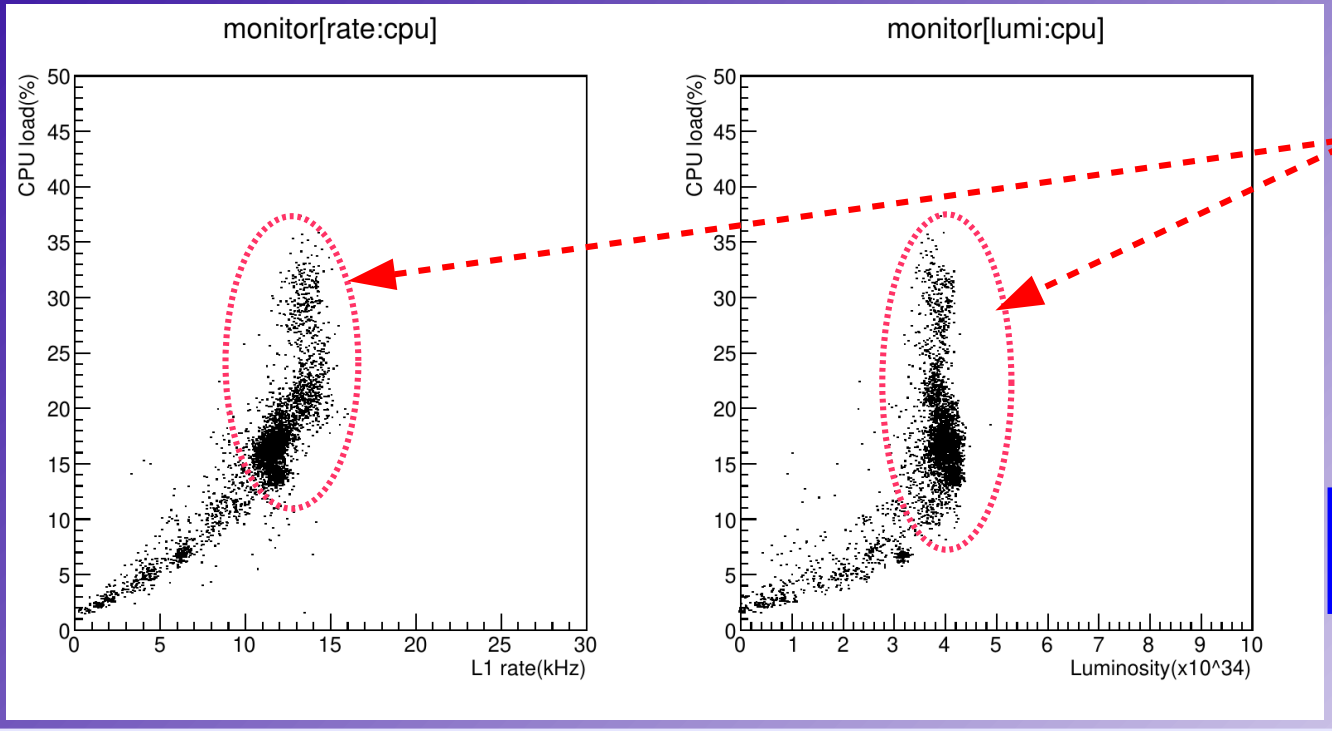
5

2. Motivation

- Current SuperKEKB status
  * Although the luminosity is still low, **the background is unexpectedly higher**.
  * This situation may persist as the luminosity continues to rise.

- The HLT CPU load is higher even though the physics event rate is far below the design value.
  * Current HLT (with 15 units) already has a sufficient processing power to manage “physics rate” at the final design luminosity.
  * The main issue is
    the higher CPU consumption caused by the “unexpectedly high background”.

Current HLT performance



Higher CPU consumption than expected

Data taken in early 2026 runs

- Heavy noise hits caused by the background are contaminated in the taken events.
  -> Source of the unexpected higher CPU consumption.

Efficient supression of such noise hits/events is the key to mitigate the problem.

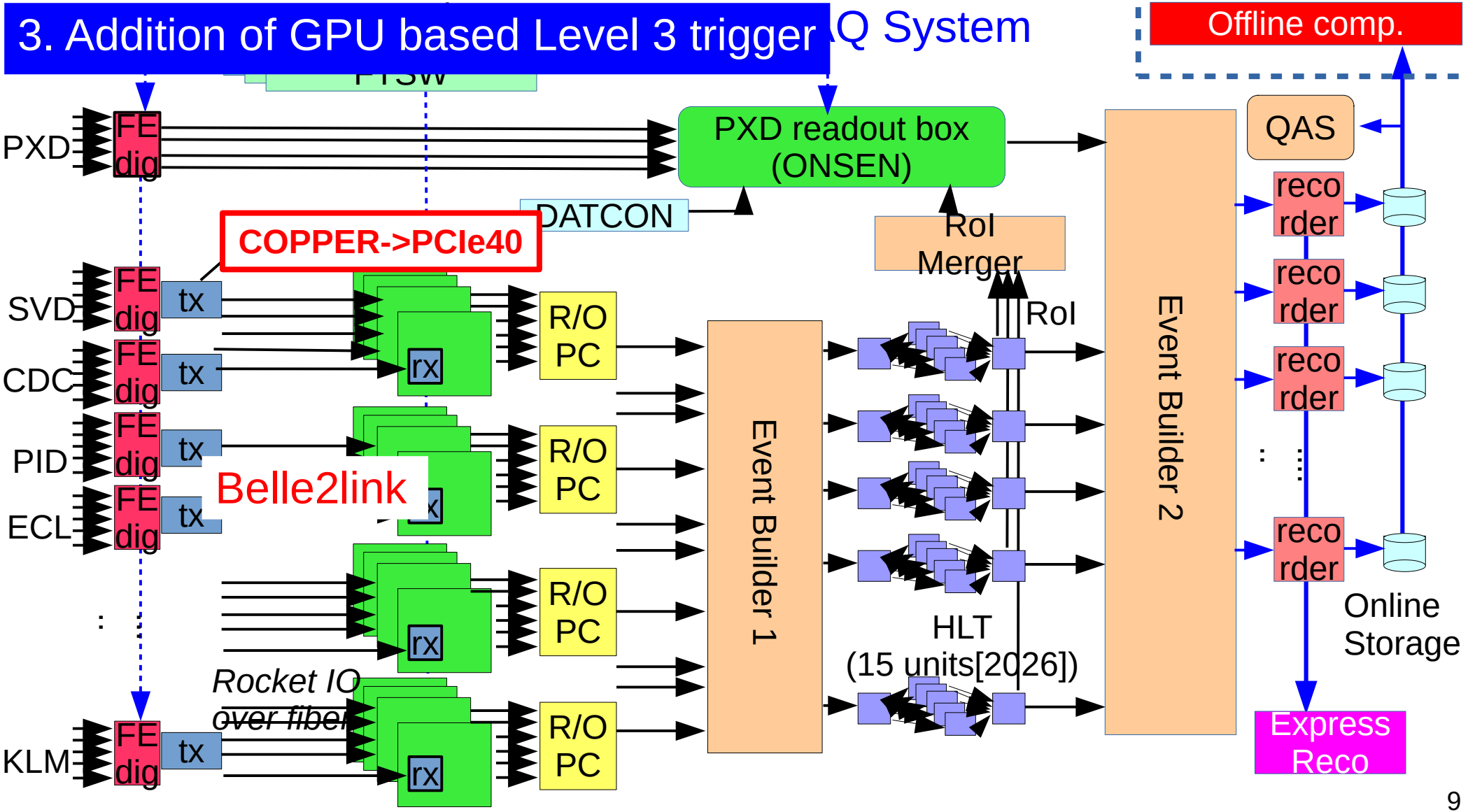
Current approaches for the mitigation

[Hardware]
* Continue to add more PC servers in HLT units : **baseline upgrade**
  - Difficult to prepare a large CPU power in a short period with a limited budget.
    (only up to 5-10% performance increase per year)

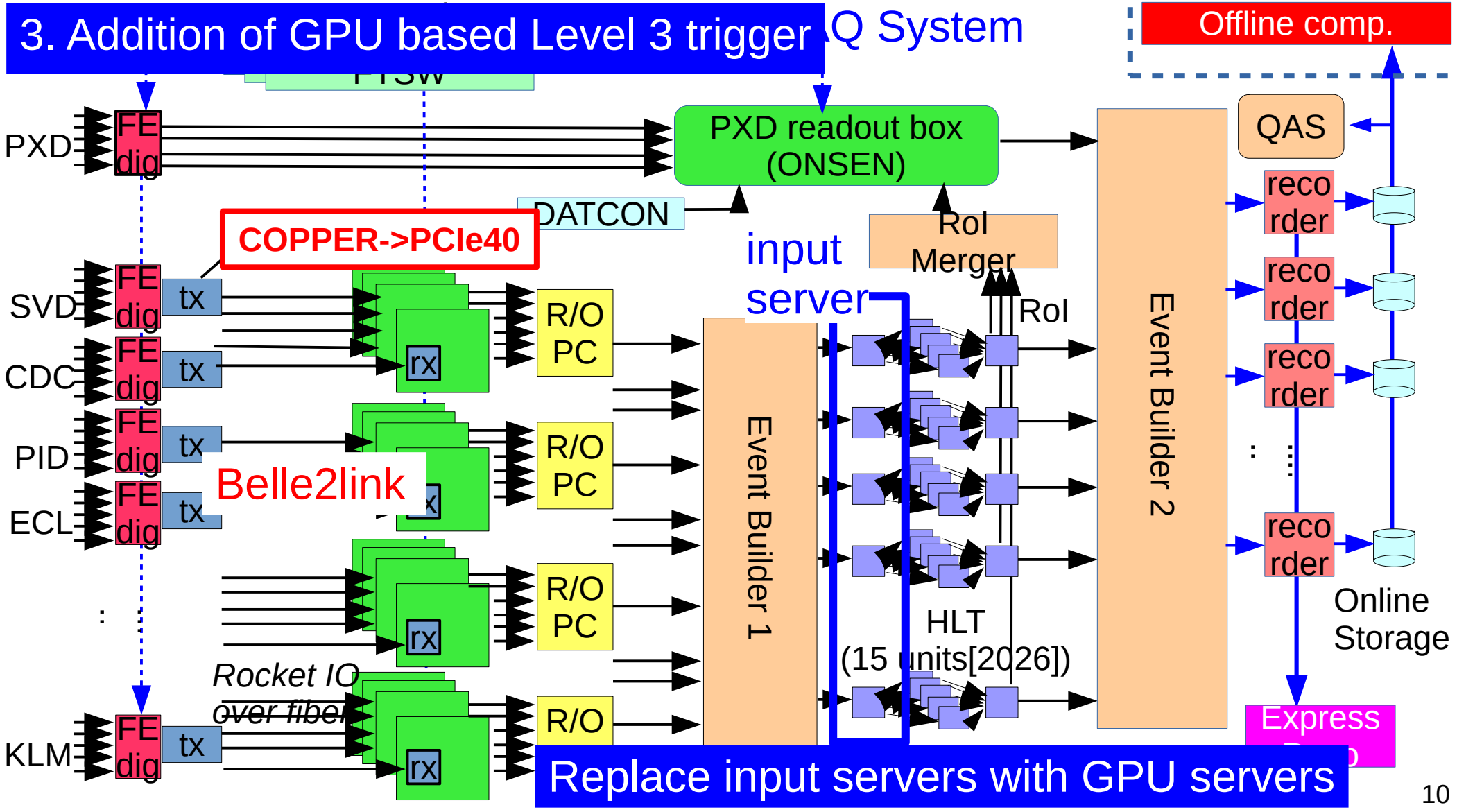
[Software]
* Signal “cleaning/filtering”
  -> already done. 10-20% reduction of CPU power.
* Implementation of software Level 3 trigger in HLT processing chain.
  -> being prepared

  => But still worry for the lack of CPU power.

3. Addition of GPU based Level 3 trigger



3. Addition of GPU based Level 3 trigger



Choice of GPU for Belle II HLT

* A GPU houses >10,000 CUDA cores.
* Adding 1 GPU for each of 15 HLT input servers is enough for our purpose.

-> in total of 150,000 CUDA cores

* GPU with better performance in fp64 is desired but hopefully less dependent (assumption to rely mostly on fp32).
* Vector/Tensor cores are not supposed to be used for the planned processing.

RTX 5090 : 21,760 cores, 32GB, ~100TFLOPS(fp32)

RTX 5000Ada : 12,448 cores, 32GB, 61TFLOPS(fp32)

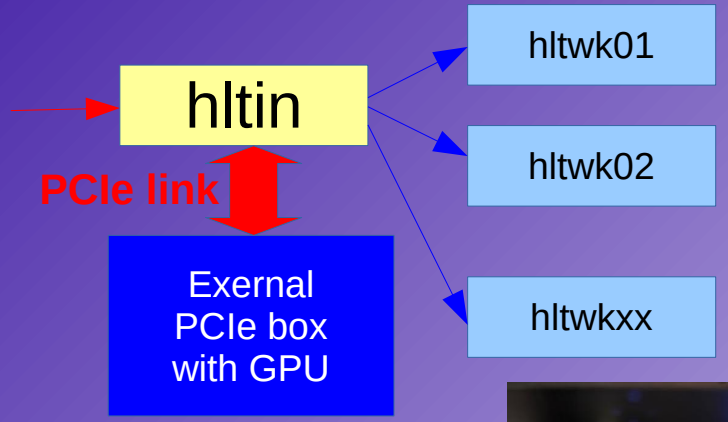
RTX 6000Ada : 18,176 cores, 48GB, 91.1TFLOPS(fp32)

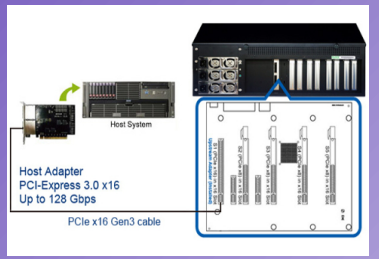
H100 : 14,592 cores, 80GB, 30TFLOPS(fp32)

A2 : 1,280 cores, 16GB, 4.5TFLOPS(fp32) : (used in test bench)

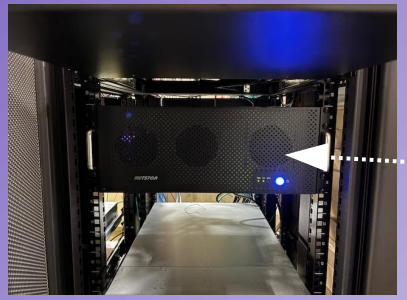
GPU Implementation in existing HLT unit (unit 15)

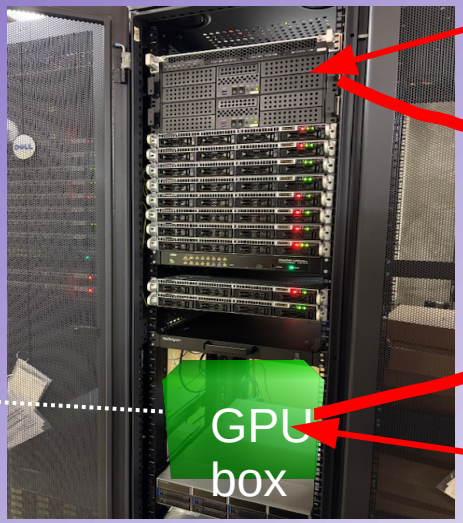
- Place GPU in an external rack-mount 4U unit (eGPU).
- Connect it to existing HLT input server via a special PCIe link.





netstor NA265A-G4





hltin

PCIe-link

GPU box

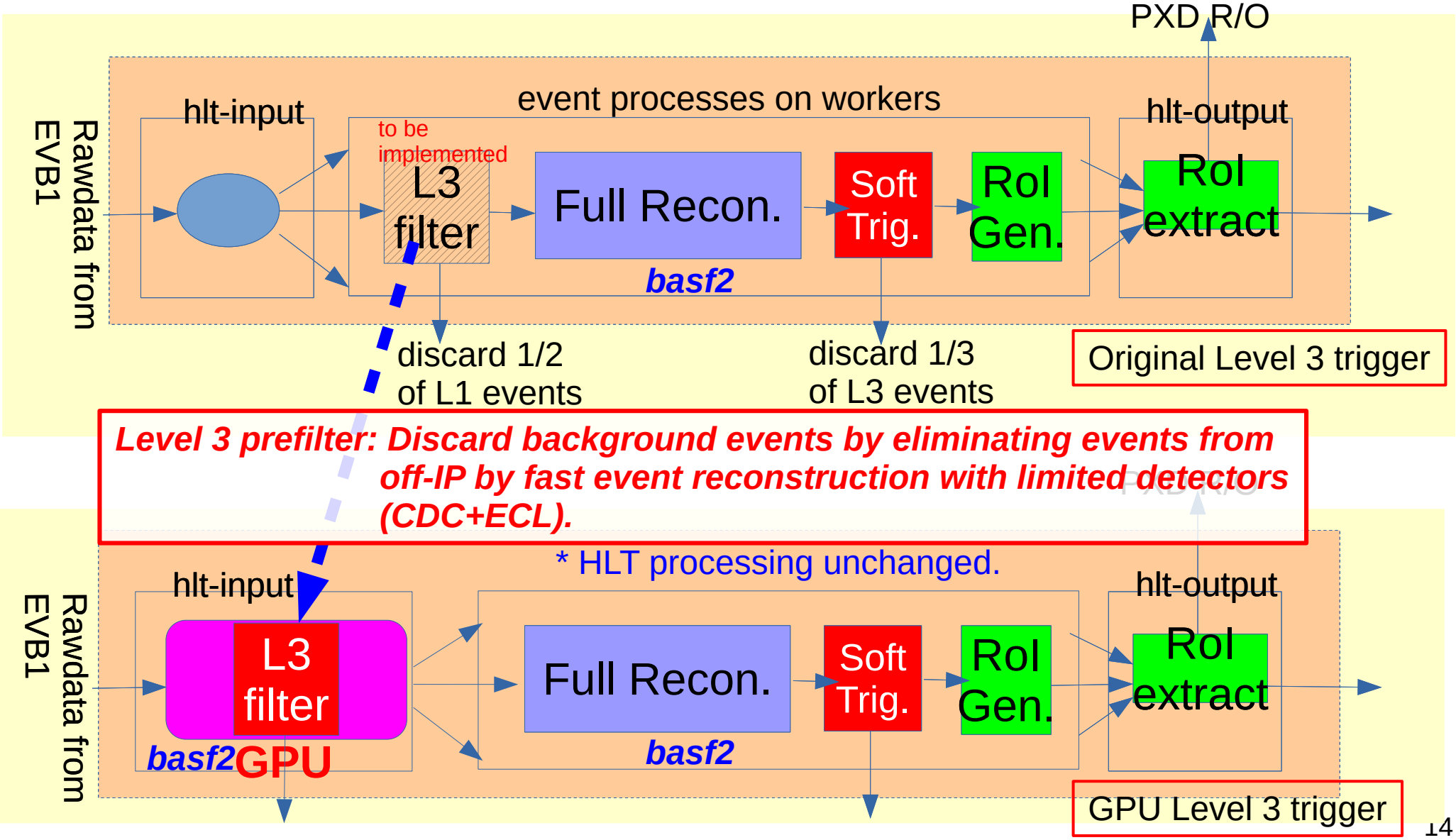
Design Policy

- Keep existing HLT as is.
  * No change in the HLT hardware/software.
    Current HLT software processing is not touched at all.

- Add GPU processing before the existing HLT framework as a part of the HLT system.
  * **Level 3 trigger software is implemented in GPU.**
  * Filter out background events before the actual HLT processing.

- Build GPU software development environment fully compatible with the Belle II software library coded mostly in C++.
    “Seamless software environment between CPU and GPU”

GPU-L3 project


PXD R/O
Rawdata from EVB1
hlt-input
event processes on workers
to be implemented
L3 filter
Full Recon.
Soft Trig.
RoI Gen.
hlt-output
RoI extract
basf2
discard 1/2 of L1 events
discard 1/3 of L3 events
Original Level 3 trigger
Level 3 prefilter: Discard background events by eliminating events from off-IP by fast event reconstruction with limited detectors (CDC+ECL).
* HLT processing unchanged.
Rawdata from EVB1
hlt-input
L3 filter
basf2GPU
Full Recon.
Soft Trig.
RoI Gen.
hlt-output
RoI extract
basf2
GPU Level 3 trigger

4. Seamless Software Development Environment

- The software on GPU is desired to be compatible with the existing Belle2 software coded mostly in C++.

- GPU is supposed to be programmed in CUDA.

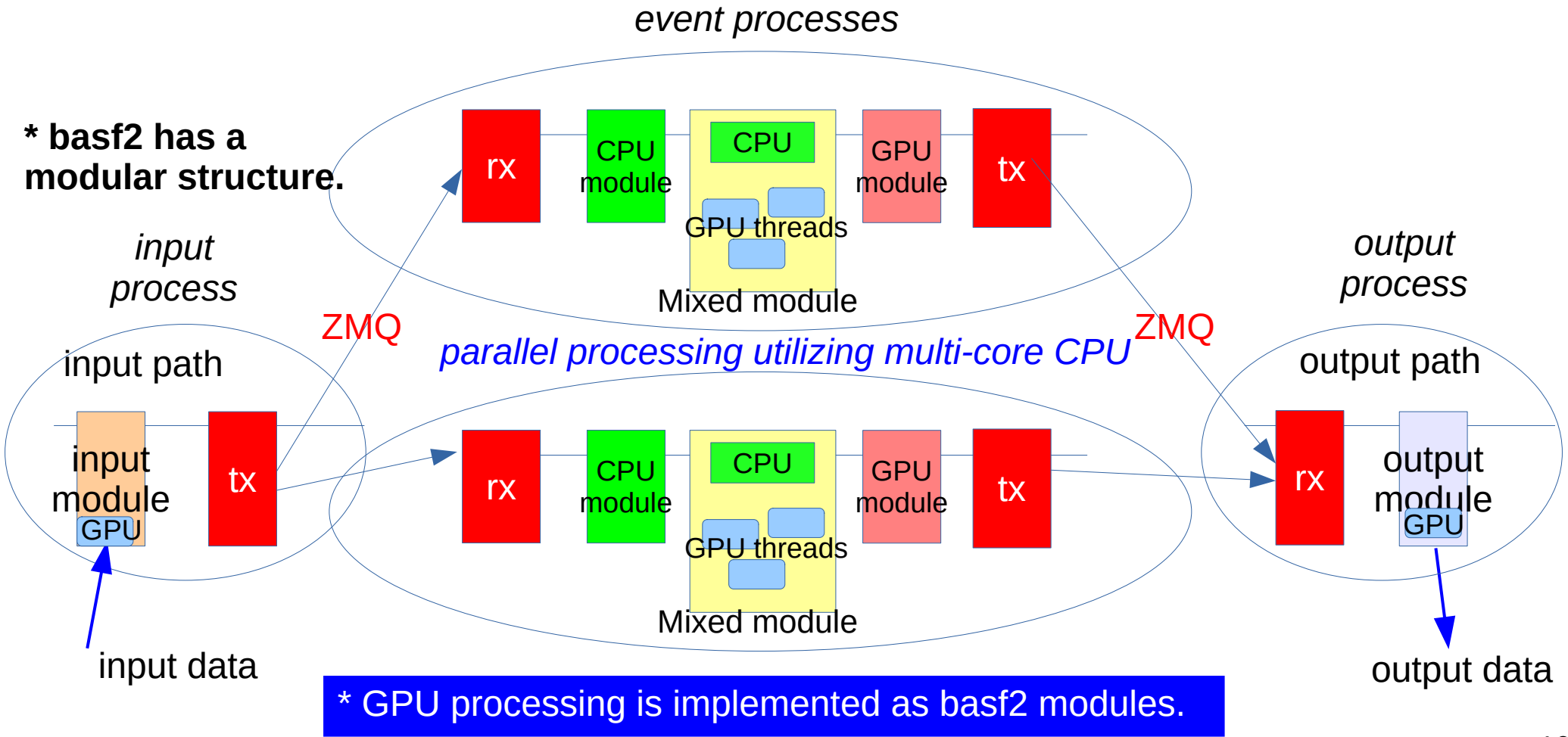
- The Belle2 software is developed using “SCons” and runs on a framework called “basf2” as a set of dynamically-linked modules.

- CUDA compiler (nvcc) was added to SCons build system.

- Development environment was built so that mixed C++ and CUDA codes can co-exist in Belle2 library and dynamically loaded in basf2.

=> Seamless execution of CUDA-coded GPU modules and C++-coded CPU modules on the same basf2 became possible.

basf2: Belle 2 event processing framework commonly used in online and offline with the parallel processing capability on multicore CPU.



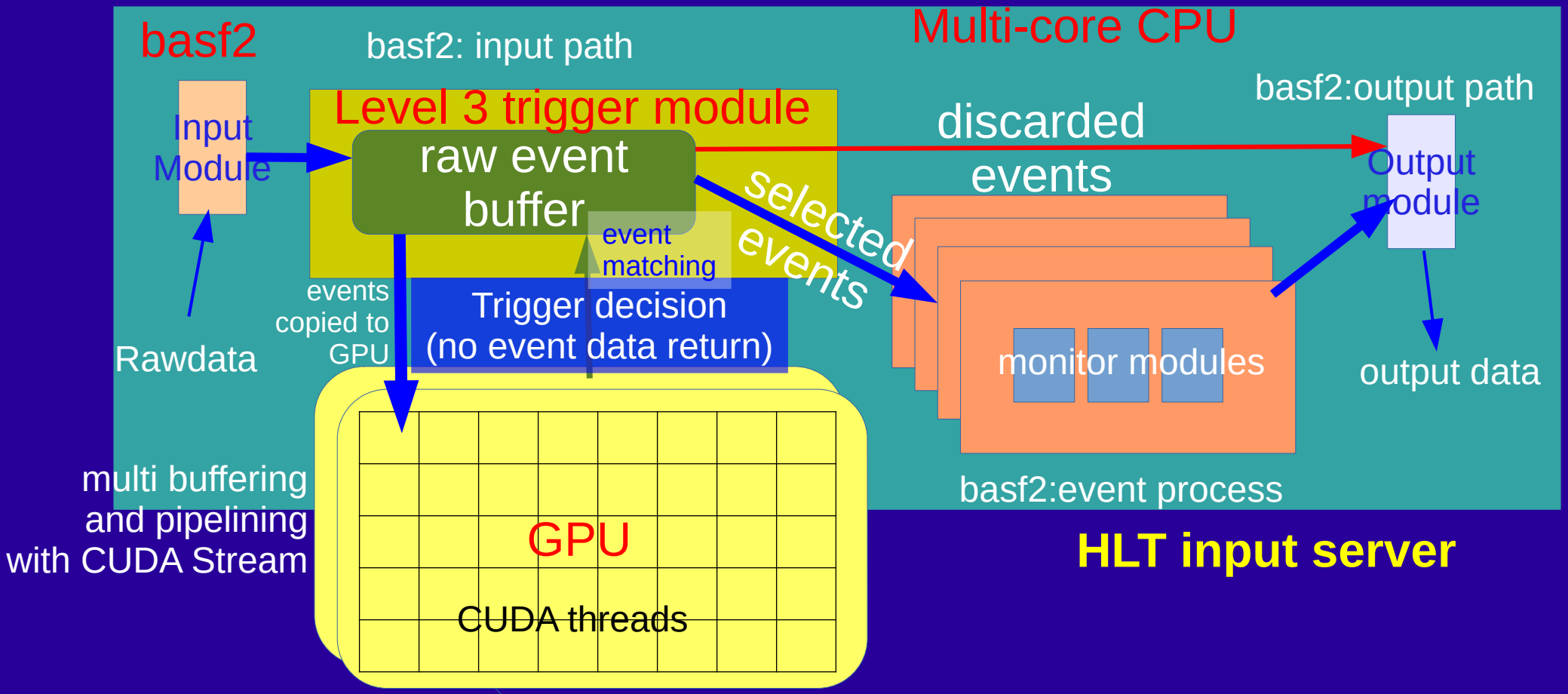
5. Data Flow for GPU

- Batch processing nature of GPU
  * GPU executes the same instruction chain for a group of different input events on a huge number of CUDA cores (threads) in parallel.
    -> “batch” processing of many events started at the same time.
  * Processing of one “batch”

1. buffer a bunch of events in the host memory.
2. copy them to GPU memory
3. start GPU batch processing and wait for the completion
4. transfer back results and compare with events in host memory
5. send chosen events to next basf2 module

- Pipelining is necessary to average out the bursty flow rate.
  ->”**CUDA Stream**” is utilized to implement multi-buffering.

Actual data flow for basf2+GPU in HLT input server



"Batch" processing is started when all thread buffers are filled.

6. Implementation of Level 3 trigger code

- “Reference” Level 3 trigger software is now being implemented on GPU by recycling the old code developed in C++ for the practice of code porting to GPU and performance evaluation.
  <- *Consisting of fast CDC tracking and ECL clustering.*

Various tips in the porting of C++ code to CUDA

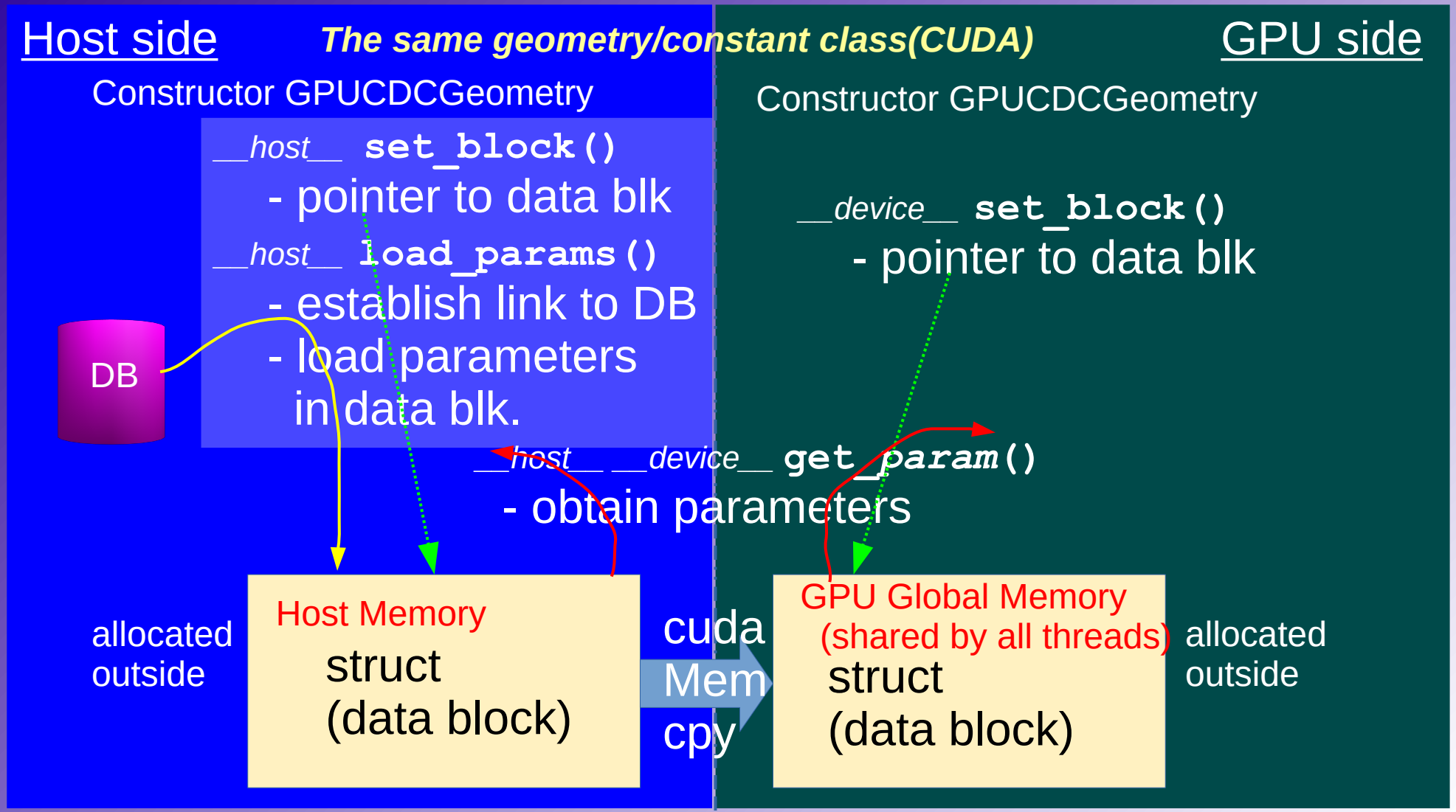
- Database access from GPU is implemented by copying the parameter memory block initialized on host from the database to GPU’s global memory.

- Dynamic object creation by C++ new() is replaced with a private GPU memory management using the “placement new()” so that the objects are created in the GPU’s global memory.

- STL (such as std::vector) / ROOT functions are replaced with home-grown libraries.

- “Double precision” floating point operations are replaced with single precision ones so as to maximize the GPU performance.

Geometry/Constants DB access from GPU



Current status of GPU-L3 project

Item	Status	Remark
Seamless software environment	Done and working	Still some problem (-j 1 is necessary...)
Data transport from host to GPU (and reverse)	Done and working	Pipelined data flow is implemented/tested.
Raw data unpacking (CDC/ECL/TRG)	CDC -> done ECL/TRG -> not yet	channel->wire conversion with DB
Geometry database access	CDC -> done ECL-> next	Full CDC geometry is available on GPU
CDC Track Finding/Fitting	In progress	Compilation passed Debugging in progress
ECL clustering	not yet	
TRG processing	not yet	
Level 3 decision	not yet	

7. Summary

- A significant shortage in the processing power of Belle II HLT is forseen because of the unexpected high background.

- The addition of GPU in HLT is being planned by porting existing C++ based reconstruction software.

- A seamless development environment between CPU and GPU has been implemented in Belle2 software library.

- The pipelined data flow scheme for GPU on basf2 framework has been developed and established.

- C++ based Level3 trigger software is being ported to GPU.

- The performance will be studied in beam from this fall.

Backup Slides

Application of Level 3 reconstruction

- Noise hits within the physics events also cause a high CPU consumption in the event processing.

- The results of Level 3 tracking can be used for the reduction of such noise hits.

- The tracking results are sent to main HLT processing as “Region of Interest”. The hits outside the “RoI” are discarded in the main tracking so as to reduce the CPU consumption.

Validation of GPU code in offline

- The insertion of GPU based L3 trigger may affect on the physics quality and it has to be validated in MC simultation.

- But it is difficult to prepare GPUs in offline computing facility for the MC generation.

- CUDA can generate the binary for CPU and GPU simultaneously from the same source code. (__host__ __device__ function).

- Compare the results between GPU and CPU execution and if the difference is within a tolerable range, we can rely on the CPU code only for MC.

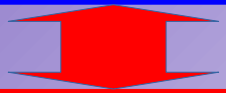
Machine learning approach for Level 3 trigger

- Current design of Level 3 code is based on the conventional CDC tracking + ECL clustering approach (reference design).

- But new algorithm can be developed using the machine learning technique. Study is on-going in Belle II (Shandong U).

- The GPU-capable Belle II software environment can be utilized for the development.

- Possibility to utilize vector/tensor cores in GPU in this case.



However, the development and validation of new code will take significantly longer time...

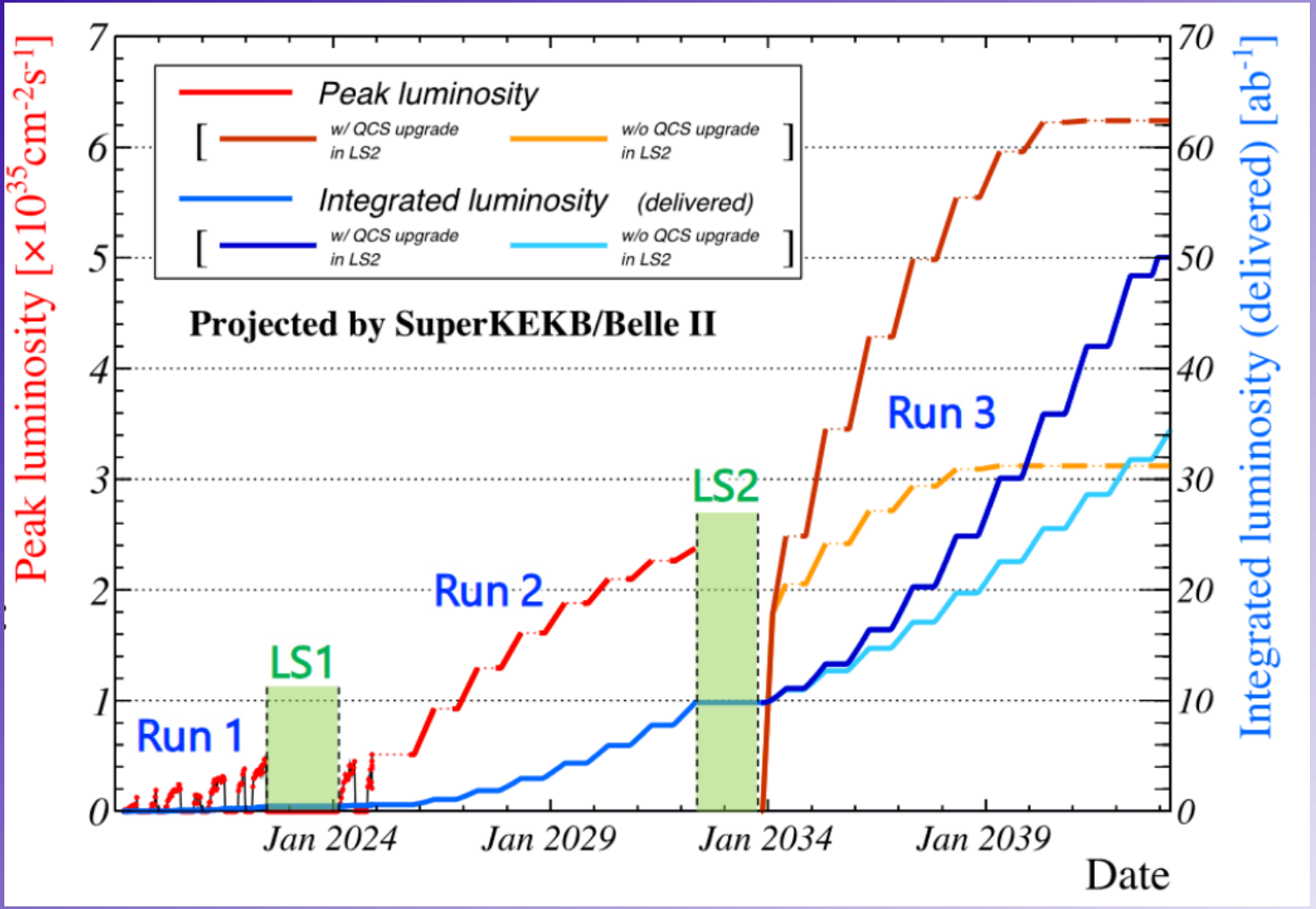
Coding Notes:

- CUDA is mostly compatible with gcc and existing C++ codes can be compiled by “nvcc” without modifications.
  * But compiler version should be consistent.

- Functions which run on GPU are marked with a special header key __global__ or __device__.

- A basf2 module can be compiled with nvcc where C++ codes on host and CUDA codes on GPU are mixed.

- The compiled objects are linkable as a shared object which can be dynamically loaded as a basf2 module.

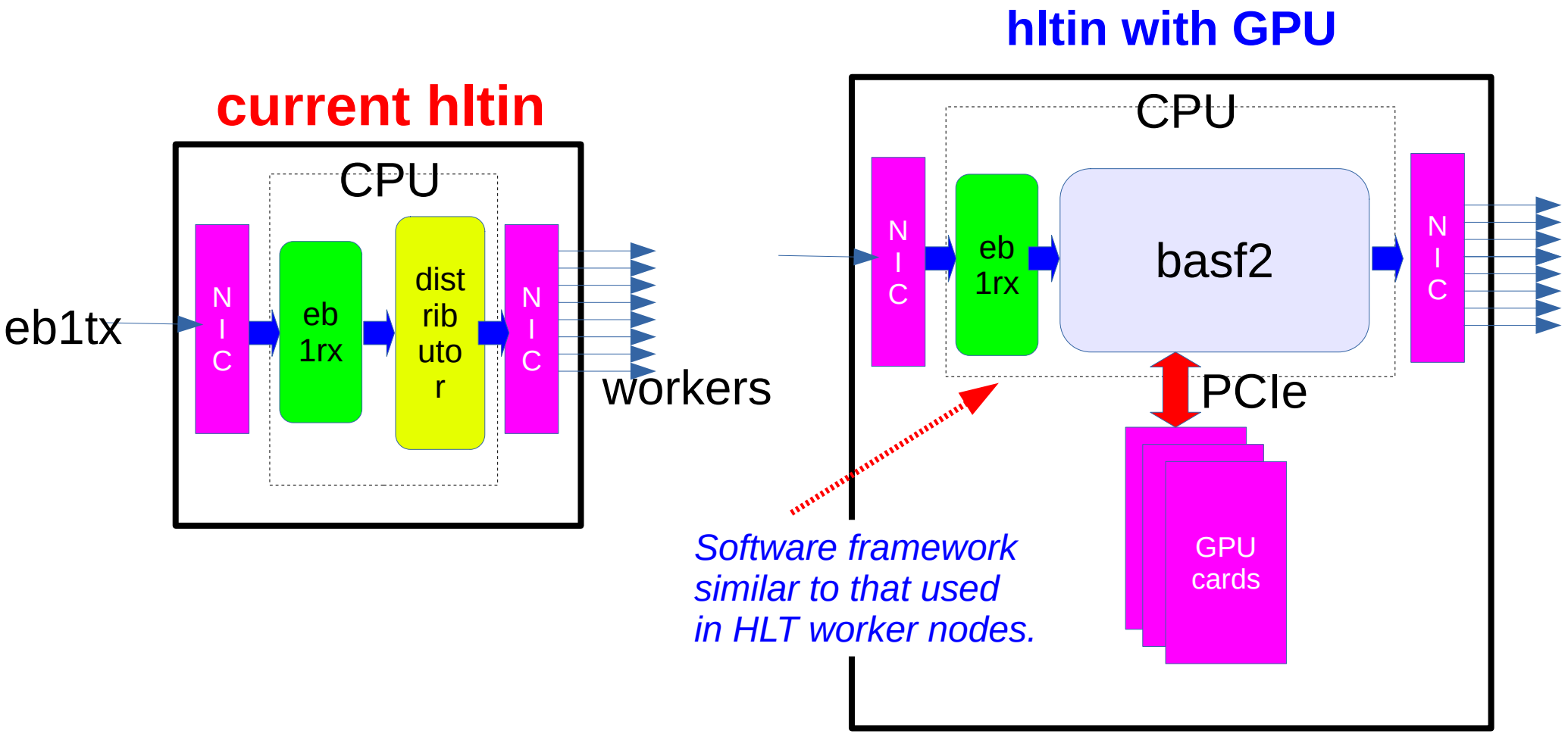

Peak luminosity
w/ QCS upgrade in LS2
w/o QCS upgrade in LS2
Integrated luminosity (delivered)
w/ QCS upgrade in LS2
w/o QCS upgrade in LS2
Projected by SuperKEKB/Belle II
Peak luminosity [×10^35 cm^-2 s^-1]
Integrated luminosity (delivered) [ab^-1]
Run 1
LS1
Run 2
LS2
Run 3
Jan 2024
Jan 2029
Jan 2034
Jan 2039
Date

Number of physical cores in HLT in 2025c-2026ab run

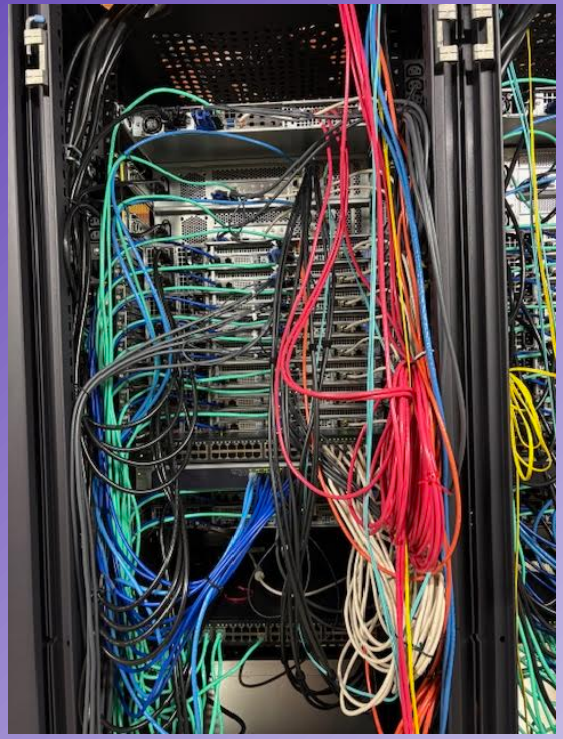
HLT01: (already renovated)
28 cores * 1 + 36 * 2 + 40 * 3 + 48 cores * 5 = 460 cores/u
HLT02-05
20 cores * 16 + 36 cores * 2 + 40 cores * 2 = 472 cores/u -> 1888
HLT06-10
28 cores * 12 + 36 cores * 2 + 40 cores * 2 = 488 cores/u -> 2440
HLT11-12, 14-15
48 cores * 10 = 480 cores/u -> 1920
HLT13 (reserved as a test bench)
16 cores * 3 + 20 cores * 6 + 48 cores * 8 = 452 cores/u -> 552

64 x “20 core” servers were purchased in 2014-2015
-> Renovation (rep. with >=48 core servers) is planned from 2025

In total of 7260 cores (c.f. design:6400 cores)


current hltin
CPU
NIC
eb 1rx
distributor
NIC
eb1tx
workers
hltin with GPU
CPU
NIC
eb 1rx
basf2
NIC
PCIe
GPU cards
Software framework
similar to that used
in HLT worker nodes.

HLT15



Test bench system

recycled from another purpose

```
itohepyc% lscpu
Architecture:             x86_64
  CPU op-mode(s):         32-bit, 64-bit
  Address sizes:          48 bits physical, 48 bits virtual
  Byte Order:             Little Endian
CPU(s):                   96
  On-line CPU(s) list:    0-95
Vendor ID:                AuthenticAMD
  Model name:             AMD EPYC 7643 48-Core Processor
    CPU family:           25
    Model:                1
    Thread(s) per core:   2
    Core(s) per socket:   48
    Socket(s):            1
    Stepping:             1
    Frequency boost:      enabled
    CPU(s) scaling MHz:   63%
    CPU max MHz:          3640.9170
    CPU min MHz:          1500.0000
    BogoMIPS:             4599.87
```

```
itohepyc% nvidia-smi
Thu Jan 30 13:59:22 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 565.57.01              Driver Version: 565.57.01      CUDA Version: 12.7     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA A2                      Off |   00000000:81:00.0 Off |                    0 |
|  0%   33C    P8              5W /   60W |       1MiB /  15356MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
```

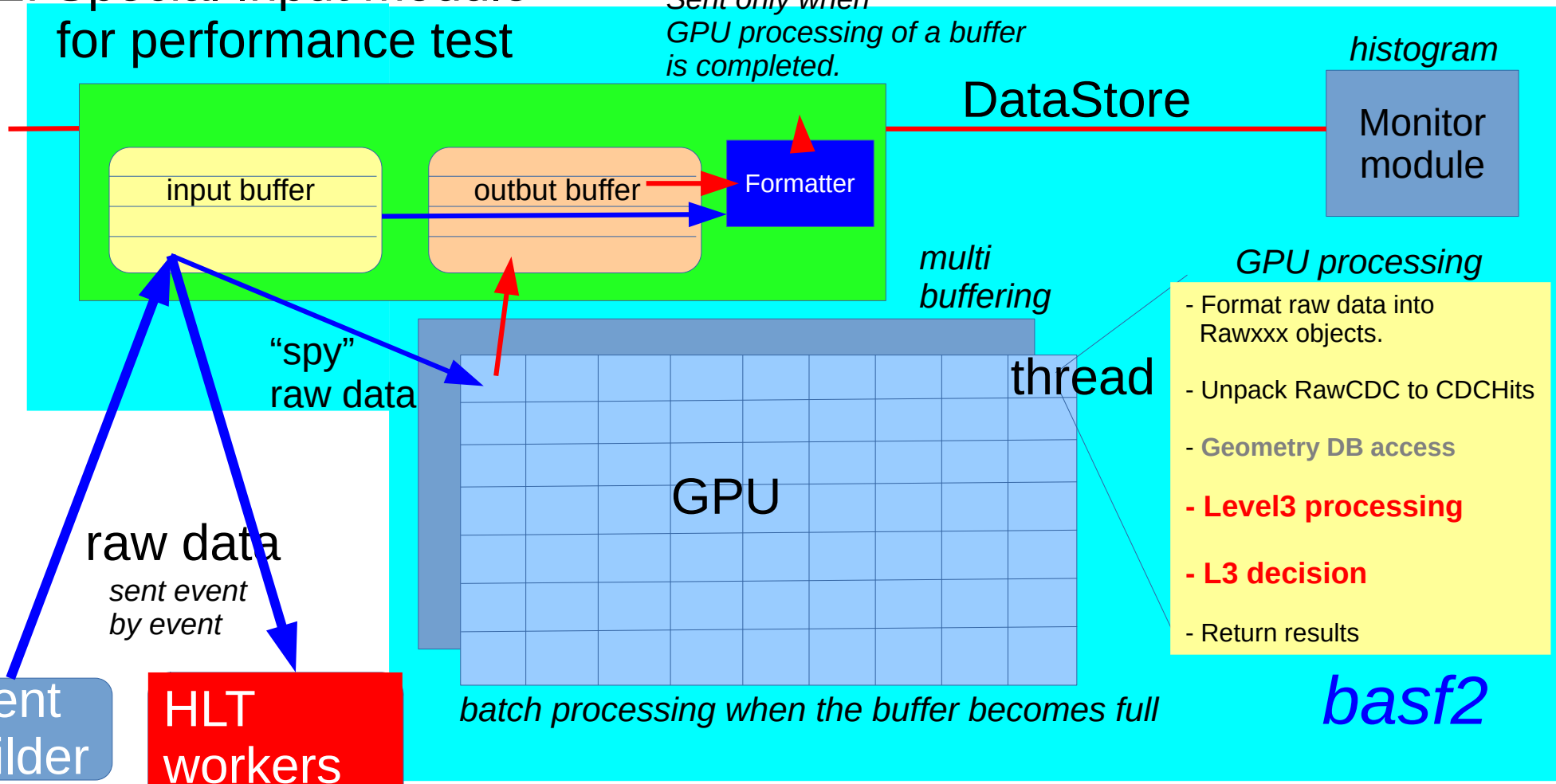
CPU : AMD EPYC 7643
(48 cores/96 threads)

```
itohepyc% cat /etc/redhat-release
Rocky Linux release 9.5 (Blue Onyx)
```

GPU : NVIDIA A2 Tensor
(1280 CUDA cores;
10 RT cores;
40 Tensor cores)

basf2: Special Input module for performance test

HLT15:hltin



* Normal HLT data flow is not disturbed by this performance test.

Cost comparison between 1xPC server and 1xGPU

(1 PC server(Xeon Gold 6336Yx2) vs 1 GPU(RTX6000Ada))

🧠 Comparison: Xeon Gold 6336Y ×2 vs NVIDIA RTX 6000 Ada

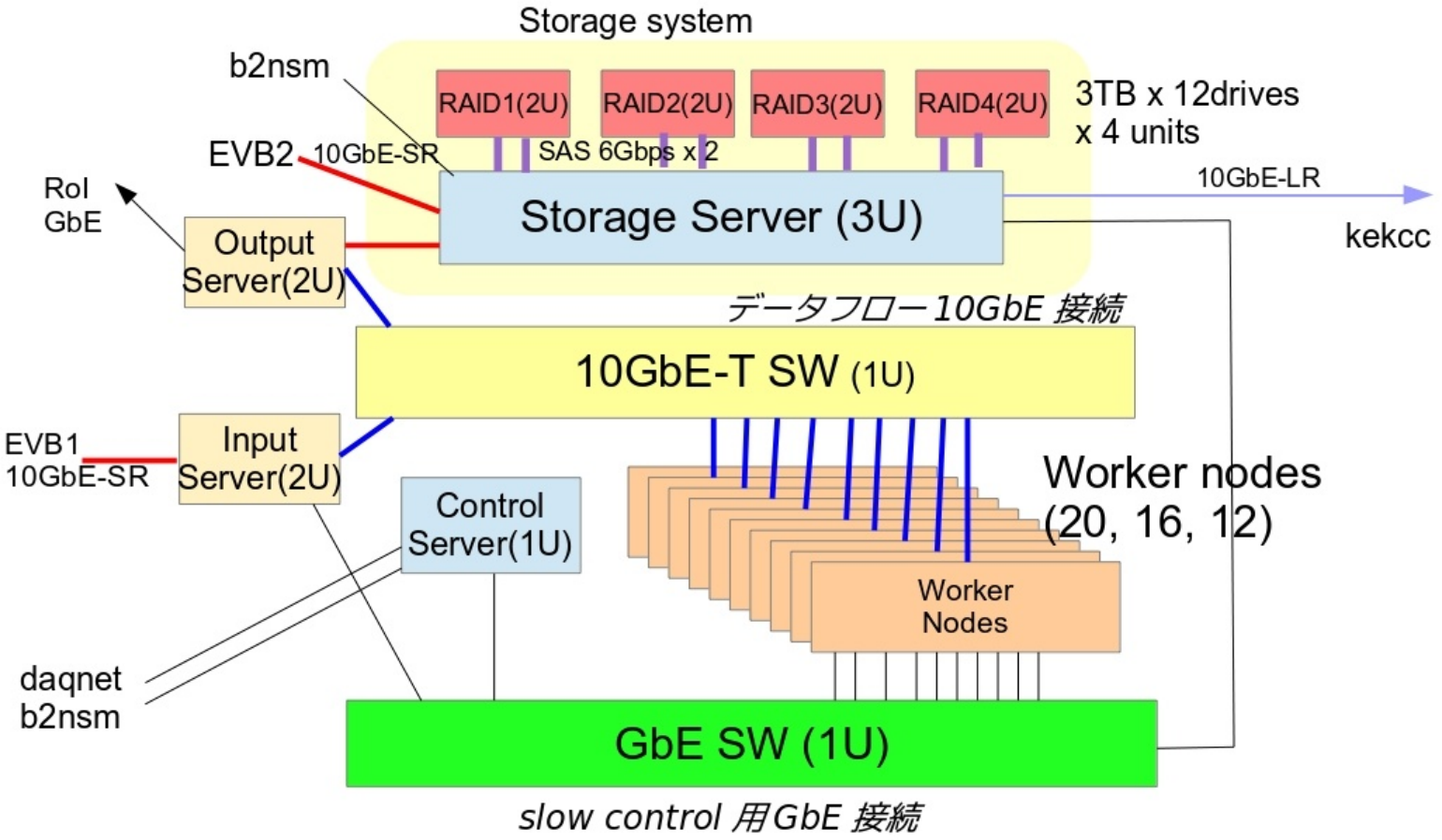
Category	Xeon Gold 6336Y ×2	NVIDIA RTX 6000 Ada
Architecture	Ice Lake-SP (10nm)	Ada Lovelace
CPU/GPU Cores	**48 cores / 96 threads** (24C×2)	**18,176 CUDA cores**
Clock Speed	2.4 GHz (Base), 3.6 GHz (Turbo)	~2.5 GHz (Boost)
FP64 (Double Precision)	**~1.1–1.4 TFLOPS** (combined)	**~2.0 TFLOPS**
FP32 (Single Precision)	**~3–4 TFLOPS** (combined, effective)	**91.1 TFLOPS**
Memory Capacity	Up to **2TB+ DDR4 ECC**	**48GB GDDR6 ECC**
Memory Bandwidth	**~200 GB/s** (DDR4 x2 CPUs)	**960 GB/s**
Power Consumption (TDP)	**~410W** (total for 2 CPUs)	**~300W**
PCIe Interface	PCIe Gen 4.0 ×64 total (2 CPUs)	PCIe Gen 4.0 ×16
Price (JPY)	**¥1,600,000–2,000,000** (CPU, RAM, motherboard, etc.)	**¥850,000–900,000**

- The processing power/GPU core is a bit lower than that of CPU, but number is incredibly more!
- >30 times faster performance (fp32).

* Processing power can be doubled only with a cost of 1-year HLT reinforcement!

- But fp64 performance is not so much....

Actual Implementation of HLT/Storage unit



Equipped in a single rack