Improved HLT Framework for Belle II Experiment

Ryosuke Itoh[1]*, *Anselm* Baur[2], *Seokhee* Park[1], *Mikihiko*Nakao[1], *Satoru*Yamada[1], *Soh Y* Suzuki[1], *Takuto* Kunigo[1], and *Dmytro* Levit[1]

[1]High Energy Accelerator Research Organization (KEK),
[2]Deutsches Elektronen-Synchrotron (DESY).

Abstract. The original Belle II HLT framework was formally upgraded replacing the old IPC based ring buffer with the ZeroMQ data transport to overcome the unexpected IPC locking problem. The new framework has been working stably in the beam run so far, but it lacks the capability to recover the processing fault without stopping the on-going data taking. In addition, the compatibility with the offline framework (**basf2**) was lost which was maintained in the original. In order to solve these, an improved core processing framework is developed based on original **basf2**, while keeping the existing ZeroMQ data transport between the servers unchanged. A new core framework **zmq-basf2** is developed with a lock-free 1-to-N and N-to-1 data transport using the ZeroMQ IPC socket so that it keeps a 100% compatibility with the original ring-buffer based framework. When a processing fault occurs, the affected faulty event is salvaged from the input buffer and sent directly to the output using the ZeroMQ broadcast. The terminated process is automatically restarted without stopping data taking. This contribution describes the detail of the improved Belle II HLT framework with the result of the performance test in the real Belle II DAQ data flow.

1 Introduction

The Belle II experiment[1] is a B-factory experiment at KEK in Japan aiming at the discovery of New Physics in B meson decays. The SuperKEKB accelerator[2] is designed to achieve the world's highest luminosity of $L = 6 \times 10^{35} cm^{-2} sec^{-1}$ in order to accumulate the integrated luminosity of $50ab^{-1}$ which corresponds to a 10^{11} B meson sample. The SuperKEKB accelerator and the Belle II detector are shown in Figure 1.

The commissioning of the SuperKEKB accelerator was started from February, 2016 for the accelerator tuning and the vacuum scrubbing (Phase 1). The first electron-positron collision was observed in April, 2018 and the pilot run was performed only without the vertex detector (Phase 2). The physics run with the vertex detector installed had started from March, 2019 and the data taking with a full DAQ configuration had been performed until June, 2022 (Phase 3) accumulating the total integrated luminosity of $428 fb^{-1}$ up to today.

Now the experiment is in the 1.5 year long shutdown period for the installation of upgraded pixel detector. The data taking will be resumed in December, 2023. Utilizing the shutdown period, various upgrades in the Belle II DAQ system is being performed. The improvement of the HLT framework is one of them.

*e-mail: ryosuke.itoh@kek.jp

2 Belle II DAQ System

Fig. 2 shows the schematic view of the Belle II DAQ system. The Belle II DAQ system is a conventional trigger driven data acquisition system. The Level 1 trigger generated by the global decision logic is distributed to the detector front-end by the multi-layered distribution logic (FTSW) through both optical and metal links[3].

The detector signals are digitized at the detector front-end located near the detector and transferred to the common readout cards (COPPER)[4] via the unified optical link (Belle2link)[5]. The Belle2link is a bi-directional link and it is also used to download slow control parameters to the detector front-end.

On each COPPER board, a Linux-operated CPU card is mounted as a daughter card, and the data formatting and the 1st level data reduction are performed. The COPPERs are now being replaced with new generation readout cards (PCIe40) where the software processing is ported to the firmware of FPGA (Intel Arria 10)[6].

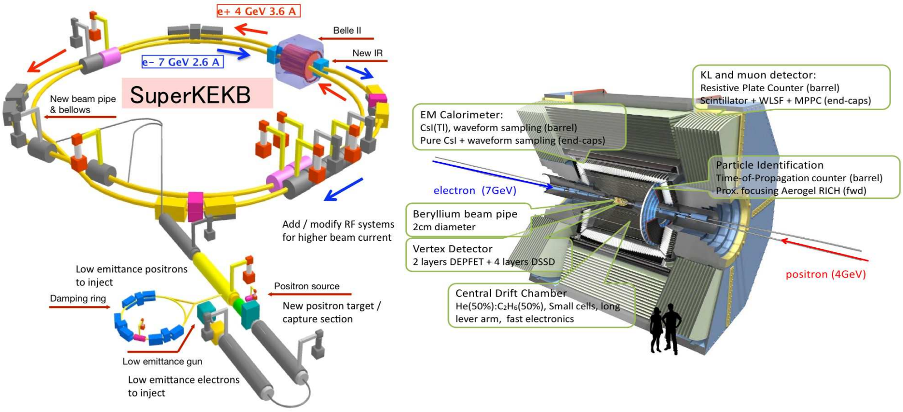
e+ 4 GeV 3.6 A
Belle II
New IR
e− 7 GeV 2.6 A
SuperKEKB
New beam pipe & bellows
Add / modify RF systems for higher beam current
Low emittance positrons to inject
Damping ring
Positron source
New positron target / capture section
Low emittance gun
Low emittance electrons to inject
KL and muon detector:
Resistive Plate Counter (barrel)
Scintillator + WLSF + MPPC (end-caps)
EM Calorimeter:
CsI(Tl), waveform sampling (barrel)
Pure CsI + waveform sampling (end-caps)
electron (7GeV)
Particle Identification
Time-of-Propagation counter (barrel)
Prox. focusing Aerogel RICH (fwd)
Beryllium beam pipe
2cm diameter
Vertex Detector
2 layers DEPFET + 4 layers DSSD
positron (4GeV)
Central Drift Chamber
He(50%):C2H6(50%), Small cells, long lever arm, fast electronics

Figure 1. The SuperKEKB accelerator and Belle II detector

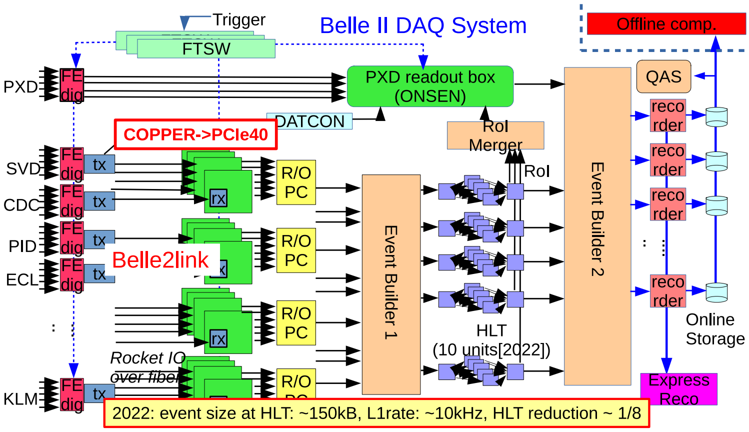

Figure 2. The Belle II Data Acquisition System

The processed data are sent to readout PCs via the GbE network. The event building is done in two steps. The first step is done on the readout PC which collects the event fragments from readout cards and formats into one event fragment of each subsystem. It is then sent to the network switch complex and distributed to one of High Level Trigger(HLT) units. A full event is built at the input node of a HLT unit.

The High Level Trigger (HLT) consists of multiple units of PC clusters. Currently 10 units are operated. One unit consists of an event distributor node, an output collector node, and up to 20 event processing nodes (workers). Each worker server houses multiple cores (16 to 48 physical cores) and in total of ~500 cores are equipped in one HLT unit. The event data are delivered to the worker nodes where event-by-event parallel processing is performed utilizing multi-cores by the HLT core framework. The processing results are collected again and sent to the 2nd event builder to merge with the pixel detector (PXD) readout.

The readout of PXD is handled in a separate way as shown in Fig. 3, since the event size is huge (up to 1 MB). The data from PXD are fed into a special readout system called ONSEN[7]. The tracks of charged particles reconstructed by the HLT are extrapolated to the surface of PXD sensors, and the region of interest (RoI) on the sensors are defined. ONSEN receives the RoIs from HLT through the RoI merger and only the hits inside RoIs are sent to the 2nd event builder switch and merged with the output from HLT to be recorded in the online storage. The data size is expected to be reduced by a factor of 10.

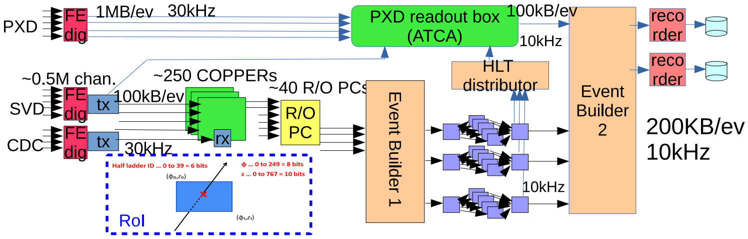

Figure 3. The scheme of RoI feedback to the PXD readout. The region of interest (RoI) in the PXD surface is defined for each of reconstructed tracks and only the detector hits in the RoI are sent.

The slow control of the system is implemented based on two different frameworks. One is NSM2[8] which is a home grown framework used in the main DAQ component, while the other is the industrial standard EPICS[9] used in some of detector subsystems and in the SuperKEKB accelerator. A gateway between them is developed and a transparent environment is implemented. The user interface (GUI) is constructed using Control System Studio[10].

3 Evolution of Framework for Belle II HLT

3.1 RingBuffer HLT

The first HLT data flow framework is built based on the event data distribution over the home-grown ring buffer[11] implemented with the Linux IPC (Shared Memory and Semaphore). Fig. 4 shows the construction. The ring buffer is originally used in the offline analysis framework (**basf2**[12]) for the event-by-event parallel processing with multi-processes, and the same ring buffer is used for the event distribution over network to utilize multiple PC servers.

However, the unexpected “locking” of IPC is observed and also the removal of IPC is not properly done in some cases. In addition, the extraction of “RoI”s needs the unpacking

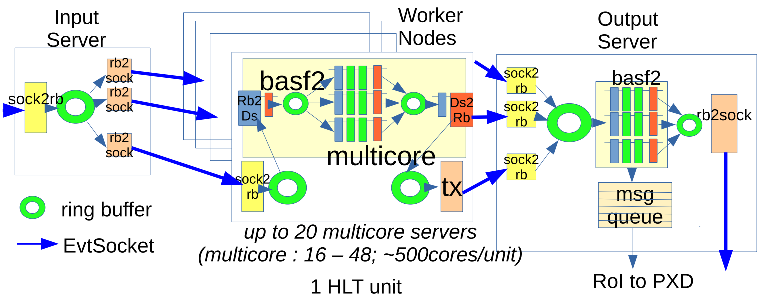

Figure 4. The original ring-buffer based HLT framework. The parallel processing is implemented based on the event distribution among processes over the home-grown ring buffer.

of streamed event objects which requires a large CPU consumption in the collector node. To overcome these problems, a new "lock free" HLT framework has been developed using ZeroMQ[13]

3.2 ZeroMQ HLT

Fig. 5 shows the data flow in the new HLT framework with the ZeroMQ data flow[14]. In each worker node, the core framework **hbasf2** is employed replacing the original **basf2**. The event data formatted in a ZeroMQ message are directly distributed to the each of worker process forked by **hbasf2** where the number of processes is up to the number of hyperthreading cores. The received message is destreamed to event objects and the processing chain is performed. The processed output are capsulized in a output ZeroMQ message and sent to collector node. At the same time, PXD RoIs are also capsulized as the 2nd message in the same ZeroMQ packet and stripped at the collector node to be sent to Onsen.

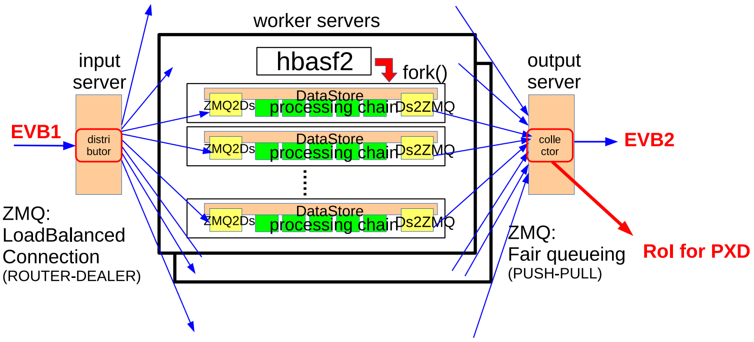

Figure 5. The new ZeroMQ based HLT framework. Event packets are directly distributed from a single input server (distributor) node to each of worker processes by ZeroMQ connections, and collected by an output server (collector) node in a reverse way.

This framework has been working stably, but it still has several problems. When one of the worker processes dies in the middle of processing (ex. segmentation fault), the event data is lost and the process is not restarted. This caused the loss of the processing power, and missing events (and RoI) in the output file. The other problem is that **hbasf2** cannot run offline with the parallel processing turned on. This means the multi-core processing cannot be utilized in the debugging of HLT software. The frequency of the crash of HLT software is supposed to be very rare and in order to reproduce the crash in offline, the parallel processing utilizing multi-core server is essential. Previous **basf2** framework has this functionality. In addition, the framework needs to have ZeroMQ connections from a single input server (distributor) to each of worker process and to a single output server (collector), the number of ZeroMQ connections to the distributor or collector becomes huge (O(1000)) and sometimes it causes the connection error.

To improve the situation, a new core framework running on each worker has been developed based on the original **basf2**.

4 Improved core HLT framework (zmq-basf2)

4.1 Design requirements

The new core framework named **zmq-basf2** is developed so that it can replace the existing **hbasf2** without any change in the external connection. The ring buffers in the original **basf2** are replaced with ZeroMQ connections.

The main feature of new framework is the recovery from the fault of event processing in a worker process. The recovery mechanism consists of two functionalities. One is to take backup of every event before it is passed to the event processes which may crash, and the other is to restart a new event process when a process crashes. At the time the backed up event is transferred directly to the output process with a bad event tag.

4.2 Implementation

Fig. 6 shows the global structure of the new framework **zmq-basf2**. In the parallel processing mode, it consists of 4 kinds of processes. The input process receives the event data delivered from the HLT distributor node and dispatches them to multiple worker processes as ZeroMQ packets. Each worker process receives the event and performs the event processing chain. The resulting output are then collected by the output process as other ZeroMQ packets and sent to the collector node. The input and output data transport is fully compatible with that of **hbasf2** so as not to change the external framework. The monitor process forks out the worker processes and monitors their status.

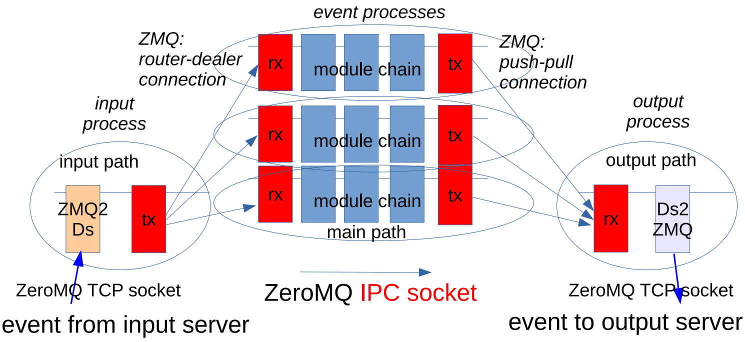

Figure 6. The new core parallel processing framework (**zmq-basf2**). The events are distributed (collected) to (from) multiple worker processes via ZeroMQ IPC sockets replacing the ring buffer.

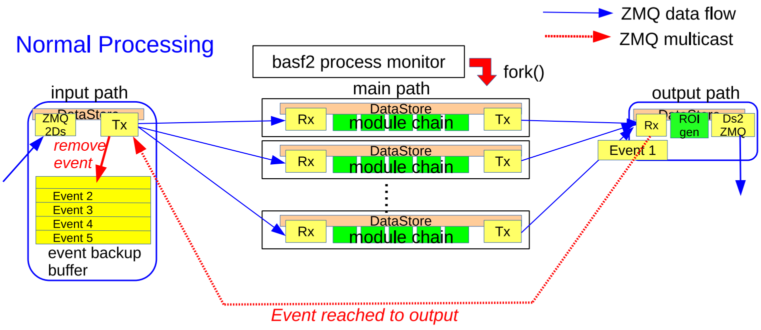

Figure 7. The data flow in **zmq-basf2** in normal processing. When the processing output of an event reaches the output process, the event is removed from the buffer in the input process.

The connections inside **zmq-basf2** are implemented using the IPC socket of ZeroMQ. Each ZeroMQ client is implemented with two connections. One is the unidirectional data socket replacing the IPC ring buffer in **basf2**. It is a load-balanced 1-to-N connection with the ZMQ ROUTER-DEALER link for input or a fair queued N-to-1 connection with the ZMQ PUSH-PULL link for output. The other is the multicast monitor socket. It consists of two sockets with ZMQ PUB and SUB properties. A set of call-backs are hooked to the socket and they are called when the corresponding message is received.

4.3 Recovery from processing fault

Fig. 7 shows the event processing in the normal case. The events received by the input process are backed up in the buffer and sent to each worker process. When the output process receives the processing output of the event, it tells the completion to the input process using the monitor socket and the input process removes the event from the buffer.

When an event causes a crash in a worker process, the monitor process detects the crash and issues the event salvage request to the input process through the monitor socket. The

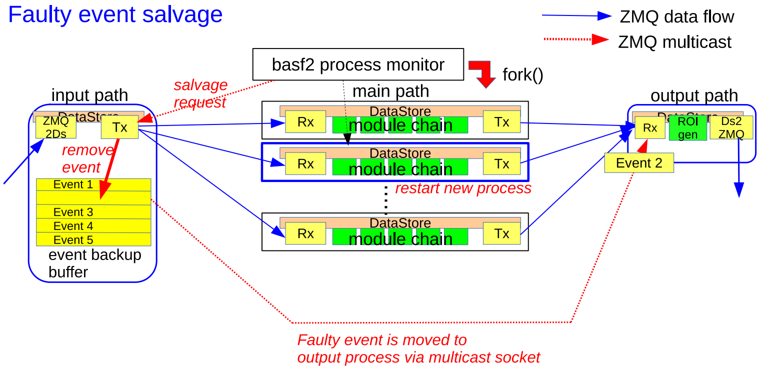

Figure 8. The recovery from process crash in **zmq-basf2**. When a worker process crashes, it is notified to the input process and the faulty event is transferred to the output process directly by the ZeroMQ multicalst, and a new worker process is forked out.

input process transfers the faulty event to the output process through the monitor socket and the event is sent to the collector node without the processing results, and the event is removed from the buffer. At the same time, the monitor process forks out a new worker process to recover the processing power. The scheme is shown in Fig. 8.

5 Test of zmq-basf2 HLT

zmq-basf2 has been implemented in one of real HLT units replacing the previous **hbasf2** and is being tested in the high rate test and cosmic ray run of Belle II DAQ. The data flow is confirmed to work stably in the real DAQ operation although the commissioning detectors are limited. The typical event size is 100k bytes/ev and the input rate is up to 10kHz, which is close to the limit of processing flow per 1 HLT units without the full reonstruction chain. The observed rate is 9.0kHz and it is confirmed that the data flow with the new framework can achieve the enough performance exceeding the design (3 kHz/unit).

The recovery from the processing fault is tested by inserting the segmentation fault artificially once per 10 events in the processing chain. The output record is monitored in the recorded file and the output objects are checked to identify the event is salvaged. The test confirms that the worker process is restarted when the segmentation fault occurs, and the faulty event is properly recorded in the output without the processing results.

6 Summary and Plan

A new core framework for the Belle II HLT is developed to be prepared for the recovery from the unexpected processing fault. It is built by replacing the old IPC based ring buffer with the ZeroMQ IPC socket in the original **basf2** framework. The event recovery is implemented using the ZeroMQ multicast. The core framework **zmq-basf2** is integrated in the existing external ZeroMQ-based HLT data transport among processing nodes, and tested in the actual data acquisition. The operation of the framework is checked with single HLT unit and the performance is confirmed to satisfy the requirement. A further test with multiple HLT units is being performed in the on-going Belle Ii Run2 commissioning and the new framework is planned to be used in the beam run starting from December 2023.

References

[1] Z. Dolezal and S. Uno (ed.), “Belle II Technical Design Report”, KEK Report 2010-1 (2010).

[2] Y. Ohnishi, *et al*, “Accelerator design at SuperKEKB”, Prog. Theor. Exp. Phys. 2013, 03A001 (2013).

[3] M. Nakao, “Timing Distribution for the Belle II Data Acquisition System”, JINST 7, C01028 (2011).

[4] T. Higuchi, *et al*, “Modular Pipeline Readout Electronics For The Superbelle Drift Chamber”, IEEE Trans. Nucl. Sci. **52**, 1912 (2005).

[5] D. Sun, Z-A. Liu, J. Zhao, H. Xu, “Belle2Link: a Global data Readout and Transmission for Belle II Experiment at KEK”, Phys. Procedia, 37, 1933-1939 (2012).

[6] Q-D. Zhou, S. Yamada, P. Robbe, D. Charlet, R. Itoh, M. Nakao, S. Y. Suzuki, T. Kunigo, *et al*, “PCI-Express Based High-Speed Readout for the Belle II DAQ Upgrade”, IEEE Trans. Nucl. Sci. **68:8**, 1818-1825 (2021).

[7] T. Gessler, W. Kuhn, *et al*, “The ONSEN Data Reduction System for the Belle II Pixel Detector”, IEEE Trans. Nucl. Sci. **62:3**, 1149-1154 (2015).

[8] M. Nakao, S. Y. Suzuki, “Network shared memory framework for the Belle data acquisition control system”, IEEE Trans. Nucl. Sci. **47:2**, 267-271, 2000.

[9] https://epics.anl.gov/index.php .

[10] http://controlsystemstudio.org .

[11] R. Itoh, N. Braun, C. Li, *et al*, “The Performance of Belle II High Level Trigger in the First Physics Run”, EPJ Web Conf. 245, 01040 (2020)

[12] R. Itoh, S. Lee, *et al*, “Implementation of parallel processing in basf2 framework for Belle II”, J. Phys. Conf. Ser. **396**, 022026 (2012).

[13] http://zeromq.org .

[14] M. Prim, N. Braun, Y. Guan, O. Hartbrich, R. Itoh, *et al*, “Design and Performance of the Belle II High Level Trigger”, PoS ICHEP2020, 769, (2021).